# Table of contents

1 Foreword .................................................................................................................................................. 5  
  1.1 Notes on the documentation.............................................................................................................. 5  
  1.2 Safety instructions ................................................................................................................................ 6  
  1.3 Notes on information security ............................................................................................................. 7  

2 Introduction................................................................................................................................................ 8  

3 LON ............................................................................................................................................................. 9  
  3.1 Supported SNVTs.................................................................................................................................. 10  
  3.2 Configuration software KS2000.......................................................................................................... 15  

4 Programming ............................................................................................................................................... 18  
  4.1 POU s .................................................................................................................................................. 18  
    4.1.1 FB_LON_KL6401 .......................................................................................................................... 27  
    4.1.2 Read .......................................................................................................................................... 28  
    4.1.3 Send .......................................................................................................................................... 197  
    4.1.4 Error codes ............................................................................................................................... 460  
  4.2 DUTs..................................................................................................................................................... 470  
    4.2.1 Enums ........................................................................................................................................ 476  
    4.2.2 Structures ................................................................................................................................... 551  
  4.3 Resources ........................................................................................................................................... 622  
    4.3.1 Globale_Variablen_LON............................................................................................................. 622  
  4.4 Integration into TwinCAT ..................................................................................................................... 622  
    4.4.1 KL6401 with CX5120 .................................................................................................................. 622  

5 Appendix .................................................................................................................................................. 630  
  5.1 Automatic send................................................................................................................................... 630  
  5.2 dwErrorKL ........................................................................................................................................... 631  
  5.3 Samples ............................................................................................................................................. 631  
  5.4 SNVT-variables (OFF) ...................................................................................................................... 631  
  5.5 Support and Service............................................................................................................................ 633
1 Foreword

1.1 Notes on the documentation

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with applicable national standards.

It is essential that the documentation and the following notes and explanations are followed when installing and commissioning the components.

It is the duty of the technical personnel to use the documentation published at the respective time of each installation and commissioning.

The responsible staff must ensure that the application or use of the products described satisfy all the requirements for safety, including all the relevant laws, regulations, guidelines and standards.

Disclaimer

The documentation has been prepared with care. The products described are, however, constantly under development.

We reserve the right to revise and change the documentation at any time and without prior announcement.

No claims for the modification of products that have already been supplied may be made on the basis of the data, diagrams and descriptions in this documentation.

Trademarks

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered trademarks of and licensed by Beckhoff Automation GmbH.

Other designations used in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

Patent Pending

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents:

with corresponding applications or registrations in various other countries.

EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany

Copyright

© Beckhoff Automation GmbH & Co. KG, Germany.

The reproduction, distribution and utilization of this document as well as the communication of its contents to others without express authorization are prohibited.

Offenders will be held liable for the payment of damages. All rights reserved in the event of the grant of a patent, utility model or design.
1.2 Safety instructions

Safety regulations

Please note the following safety instructions and explanations!
Product-specific safety instructions can be found on following pages or in the areas mounting, wiring, commissioning etc.

Exclusion of liability

All the components are supplied in particular hardware and software configurations appropriate for the application. Modifications to hardware or software configurations other than those described in the documentation are not permitted, and nullify the liability of Beckhoff Automation GmbH & Co. KG.

Personnel qualification

This description is only intended for trained specialists in control, automation and drive engineering who are familiar with the applicable national standards.

Description of symbols

In this documentation the following symbols are used with an accompanying safety instruction or note. The safety instructions must be read carefully and followed without fail!

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="danger" alt="DANGER" /></td>
<td>Serious risk of injury! Failure to follow the safety instructions associated with this symbol directly endangers the life and health of persons.</td>
</tr>
<tr>
<td><img src="warning" alt="WARNING" /></td>
<td>Risk of injury! Failure to follow the safety instructions associated with this symbol endangers the life and health of persons.</td>
</tr>
<tr>
<td><img src="caution" alt="CAUTION" /></td>
<td>Personal injuries! Failure to follow the safety instructions associated with this symbol can lead to injuries to persons.</td>
</tr>
<tr>
<td><img src="note" alt="NOTE" /></td>
<td>Damage to the environment or devices Failure to follow the instructions associated with this symbol can lead to damage to the environment or equipment.</td>
</tr>
</tbody>
</table>

- **Tip or pointer**
  This symbol indicates information that contributes to better understanding.
1.3 Notes on information security

The products of Beckhoff Automation GmbH & Co. KG (Beckhoff), insofar as they can be accessed online, are equipped with security functions that support the secure operation of plants, systems, machines and networks. Despite the security functions, the creation, implementation and constant updating of a holistic security concept for the operation are necessary to protect the respective plant, system, machine and networks against cyber threats. The products sold by Beckhoff are only part of the overall security concept. The customer is responsible for preventing unauthorized access by third parties to its equipment, systems, machines and networks. The latter should be connected to the corporate network or the Internet only if appropriate protective measures have been set up.

In addition, the recommendations from Beckhoff regarding appropriate protective measures should be observed. Further information regarding information security and industrial security can be found in our https://www.beckhoff.com/secguide.

Beckhoff products and solutions undergo continuous further development. This also applies to security functions. In light of this continuous further development, Beckhoff expressly recommends that the products are kept up to date at all times and that updates are installed for the products once they have been made available. Using outdated or unsupported product versions can increase the risk of cyber threats.

To stay informed about information security for Beckhoff products, subscribe to the RSS feed at https://www.beckhoff.com/secinfo.
2 Introduction

The Tc2_LON library is an extensive TwinCAT PLC library for data exchange with LON devices. The communication with these devices is realized via SNVTs (Standard Network Variable Types). These SNVTs are defined in the LONMark (see also LONMARK and LONMARK Network Variables). For each SNVT there is a function block for sending and another function block for receiving.

This library should only be used in conjunction with a KL6401 (LON master terminal).

The SNVT should be configured with the KS2000 [15] in the terminal.

The SNVTs are linked with a LON configuration tool (e.g. LonMaker from Echelon). This tool is not provided by Beckhoff.

The user of this library requires basic knowledge of the following:

- TwinCAT XAE
- KS2000
- PC and network knowledge
- Structure and properties of the Beckhoff Embedded PC and its Bus Terminal system
- Technology of LON devices / LON configuration tools
- Relevant safety regulations for building technical equipment

This software library is intended for building automation system partners of Beckhoff Automation GmbH & Co. KG. The system partners operate in the field of building automation and are concerned with the installation, commissioning, expansion, maintenance and service of measurement, control and regulating systems for the technical equipment of buildings.

The Tc2_LON library is usable on all hardware platforms that support TwinCAT 3.1 or higher.

Hardware documentation in the Beckhoff information system: KL6401.
Each LON device provides network variables (SNVTs) for communication with other devices. The required variables are selected based on the device documentation or the Xif file (machine-readable representation of all used SNVTs) and entered in the KL6401 via the KS2000. The KS2000 can now be used to create another Xif file. Both Xif files are read and combined with a LON configuration tool (not provided by Beckhoff). For further information please refer to the respective tool.

For each SNVT entered in the KL6401, a suitable function block has to be programmed on the PLC side. These function blocks can then be used to access the SNVTs of the devices.

### How the KL6401 functions

**Sending**

The KL6401 sends single SNVT variables. This means that an SNVT variable sent to the KL6401 is sent to the LON network individually. Only when this has successfully been sent can the next SNVT variable be transferred to the KL6401.

**Receiving**

The KL6401 has 2 buffers, the telegram buffer and the index buffer.
The input variables of a LON telegram that has been received are placed in the telegram buffer. The index number of the incoming telegram is also entered into the index buffer.

The TwinCAT LON function block evaluates the index buffer, transfers input variables that have been received to the application program, and removes them from the index buffer. If the function block does not read the data out of the telegram buffer quickly enough, it can save up to 62 entries (i.e. the maximum number of SNVT variables).

If a telegram that has already been received (which the function block has not yet transferred to the application program, and which therefore still has an entry in the index buffer) is received from the KL6401 again, then a new entry is not made in the index buffer, but the input variables in the telegram buffer are updated.

### 3.1 Supported SNVTs

General information on SNVTs can be found online under [http://types.lonmark.org/](http://types.lonmark.org/). The KL6401 supports the following SNVTs (LonMark data types).

<table>
<thead>
<tr>
<th>SNVT ID</th>
<th>SNVT NAME</th>
<th>Read</th>
<th>Send</th>
<th>Additional information</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SNVT_amp</td>
<td>Read [33]</td>
<td>Send [202]</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>SNVT_amp_mil</td>
<td>Read [34]</td>
<td>Send [203]</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>SNVT_angle</td>
<td>Read [35]</td>
<td>Send [204]</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>SNVT_angle_vel</td>
<td>Read [36]</td>
<td>Send [206]</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>SNVT_btu_kilo</td>
<td>Read [36]</td>
<td>Send [207]</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>SNVT_btu_mega</td>
<td>Read [37]</td>
<td>Send [209]</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>SNVT_char_ascii</td>
<td>Read [38]</td>
<td>Send [210]</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>SNVT_count</td>
<td>Read [39]</td>
<td>Send [211]</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>SNVT_count_inc</td>
<td>Read [40]</td>
<td>Send [213]</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>SNVT_date_cal</td>
<td>Read [41]</td>
<td></td>
<td>SNVT outdated, not implemented</td>
</tr>
<tr>
<td>11</td>
<td>SNVT_date_day</td>
<td>Read [41]</td>
<td>Send [214]</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>SNVT_date_time</td>
<td>Read [42]</td>
<td></td>
<td>SNVT outdated, not implemented</td>
</tr>
<tr>
<td>13</td>
<td>SNVT_elec_kwh</td>
<td>Read [42]</td>
<td>Send [215]</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>SNVT_elec_whr</td>
<td>Read [42]</td>
<td>Send [217]</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>SNVT_flow</td>
<td>Read [43]</td>
<td>Send [218]</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>SNVT_flow_mil</td>
<td>Read [44]</td>
<td>Send [219]</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>SNVT_length</td>
<td>Read [45]</td>
<td>Send [221]</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>SNVT_length_kilo</td>
<td>Read [46]</td>
<td>Send [222]</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>SNVT_length_micr</td>
<td>Read [47]</td>
<td>Send [224]</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>SNVT_length_mil</td>
<td>Read [48]</td>
<td>Send [225]</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>SNVT_lev_cont</td>
<td>Read [48]</td>
<td>Send [226]</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>SNVT_lev_disc</td>
<td>Read [49]</td>
<td></td>
<td>SNVT outdated, not implemented</td>
</tr>
<tr>
<td>23</td>
<td>SNVT_mass</td>
<td>Read [49]</td>
<td>Send [228]</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>SNVT_mass_kilo</td>
<td>Read [50]</td>
<td>Send [229]</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>SNVT_mass_mega</td>
<td>Read [51]</td>
<td>Send [231]</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>SNVT_mass_mil</td>
<td>Read [52]</td>
<td>Send [232]</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>SNVT_power</td>
<td>Read [53]</td>
<td>Send [233]</td>
<td></td>
</tr>
<tr>
<td>SNVT ID</td>
<td>SNVT NAME</td>
<td>Read</td>
<td>Send</td>
<td>Additional information</td>
</tr>
<tr>
<td>---------</td>
<td>-------------------</td>
<td>-------</td>
<td>--------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>28</td>
<td>SNVT_power_kilo</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>SNVT_ppm</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>SNVT_press</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>SNVT_res</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>SNVT_res_kilo</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>SNVT_sound_db</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>SNVT_speed</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>SNVT_speed_mil</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>SNVT_str_asc</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>SNVT_str_int</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>SNVT_telcom</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>SNVT_temp</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>SNVT_time_passed</td>
<td></td>
<td></td>
<td>SNVT outdated, not implemented</td>
</tr>
<tr>
<td>41</td>
<td>SNVT_vol</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>SNVT_vol_kilo</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>SNVT_vol_mil</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>SNVT_volt</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>SNVT_volt_dbmv</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>46</td>
<td>SNVT_volt_kilo</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>SNVT_volt_mil</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>SNVT_amp_f</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>49</td>
<td>SNVT_angle_f</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>SNVT_angle_vel_f</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>SNVT_count_f</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>52</td>
<td>SNVT_count_inc_f</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>53</td>
<td>SNVT_flow_f</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>54</td>
<td>SNVT_length_f</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>55</td>
<td>SNVT_lev_cont_f</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>56</td>
<td>SNVT_mass_f</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>57</td>
<td>SNVT_power_f</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>58</td>
<td>SNVT_ppm_f</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>59</td>
<td>SNVT_press_f</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>SNVT_res_f</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>61</td>
<td>SNVT_sound_db_f</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>62</td>
<td>SNVT_speed_f</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>63</td>
<td>SNVT_temp_f</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>SNVT_time_f</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>65</td>
<td>SNVT_vol_f</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>66</td>
<td>SNVT_volt_f</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>67</td>
<td>SNVT_btu_f</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>68</td>
<td>SNVT_elec_whr_f</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>69</td>
<td>SNVT_config_src</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>SNVT ID</td>
<td>SNVT NAME</td>
<td>Read</td>
<td>Send</td>
<td>Additional information</td>
</tr>
<tr>
<td>---------</td>
<td>--------------------</td>
<td>--------</td>
<td>--------</td>
<td>------------------------------------------------------------</td>
</tr>
<tr>
<td>70</td>
<td>SNVT_color</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>71</td>
<td>SNVT_grammage</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>72</td>
<td>SNVT_grammage_f</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>73</td>
<td>SNVT_file_req</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>74</td>
<td>SNVT_file_status</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>75</td>
<td>SNVT_freq_f</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>76</td>
<td>SNVT_freq_hz</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>77</td>
<td>SNVT_freq_kilohz</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>78</td>
<td>SNVT_freq_milhz</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>79</td>
<td>SNVT_lux</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>SNVT_ISO_7811</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>81</td>
<td>SNVT_lev_percent</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>82</td>
<td>SNVT_multiplier</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>83</td>
<td>SNVT_state</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>84</td>
<td>SNVT_time_stamp</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>85</td>
<td>SNVT_zerospan</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>86</td>
<td>SNVT_magcard</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>87</td>
<td>SNVT_elapsed_tm</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>88</td>
<td>SNVT_alarm</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>89</td>
<td>SNVT_currency</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>90</td>
<td>SNVT_file_pos</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>91</td>
<td>SNVT_multidiv</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>92</td>
<td>SNVT_obj_request</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>93</td>
<td>SNVT_obj_status</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>94</td>
<td>SNVT_preset</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>95</td>
<td>SNVT_switch</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>96</td>
<td>SNVT_trans_table</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>97</td>
<td>SNVT_override</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>98</td>
<td>SNVT_pwr_fact</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>99</td>
<td>SNVT_pwr_fact_f</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>SNVT_density</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>101</td>
<td>SNVT_density_f</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>102</td>
<td>SNVT_rpm</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>103</td>
<td>SNVT_hvac_emerg</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>104</td>
<td>SNVT_angle_deg</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>105</td>
<td>SNVT_temp_p</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>106</td>
<td>SNVT_temp_setpt</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>107</td>
<td>SNVT_time_sec</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>108</td>
<td>SNVT_hvac_mode</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>109</td>
<td>SNVT_occupancy</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>110</td>
<td>SNVT_area</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>111</td>
<td>SNVT_hvac_overid</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>SNVT ID</td>
<td>SNVT NAME</td>
<td>Read</td>
<td>Send</td>
<td>Additional information</td>
</tr>
<tr>
<td>---------</td>
<td>----------------------</td>
<td>------</td>
<td>------</td>
<td>------------------------</td>
</tr>
<tr>
<td>112</td>
<td>SNVT_hvac_status</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>113</td>
<td>SNVT_press_p</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>114</td>
<td>SNVT_address</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>115</td>
<td>SNVT_scene</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>116</td>
<td>SNVT_scene_cfg</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>117</td>
<td>SNVT_setting</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>118</td>
<td>SNVT_evap_state</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>119</td>
<td>SNVT_therm_mode</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>120</td>
<td>SNVT_defr_mode</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>121</td>
<td>SNVT_defr_term</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>122</td>
<td>SNVT_defr_state</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>123</td>
<td>SNVT_time_min</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>124</td>
<td>SNVT_time_hour</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>125</td>
<td>SNVT_ph</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>126</td>
<td>SNVT_ph_f</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>127</td>
<td>SNVT_chr_status</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>128</td>
<td>SNVT_tod_event</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>129</td>
<td>SNVT_smo_obscur</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>130</td>
<td>SNVT_fire_test</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>131</td>
<td>SNVT_temp_ror</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>132</td>
<td>SNVT_fire_init</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>133</td>
<td>SNVT_fire_indc.te</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>134</td>
<td>SNVT_time_zone</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>135</td>
<td>SNVT_earth_pos</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>136</td>
<td>SNVT_reg_val</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>137</td>
<td>SNVT_reg_val_ts</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>138</td>
<td>SNVT_vol_t_ac</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>139</td>
<td>SNVT_amp_ac</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>143</td>
<td>SNVT_turbidity</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>144</td>
<td>SNVT_turbidity_f</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>145</td>
<td>SNVT_hvac_type</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>146</td>
<td>SNVT_elec_kwh_l</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>147</td>
<td>SNVT_temp_diff_p</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>148</td>
<td>SNVT_ctrl_req</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>149</td>
<td>SNVT_ctrl_resp</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>150</td>
<td>SNVT_ptz</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>151</td>
<td>SNVT_privacyzone</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>152</td>
<td>SNVT_pos_ctrl</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>153</td>
<td>SNVT_enthalpy</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>154</td>
<td>SNVT_gfci_status</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>155</td>
<td>SNVT_motor_state</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>156</td>
<td>SNVT_pumpset_mn</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>SNVT ID</td>
<td>SNVT NAME</td>
<td>Read</td>
<td>Send</td>
<td>Additional information</td>
</tr>
<tr>
<td>---------</td>
<td>----------------------</td>
<td>-------</td>
<td>-------</td>
<td>------------------------</td>
</tr>
<tr>
<td>157</td>
<td>SNVT_ex_control</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>158</td>
<td>SNVT_pumpset_sn</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>159</td>
<td>SNVT_pump_sensor</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>160</td>
<td>SNVT_abs_humid</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>161</td>
<td>SNVT_flow_p</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>162</td>
<td>SNVT_dev_c_mode</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>163</td>
<td>SNVT阀模式</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>164</td>
<td>SNVT_alarm_2</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>165</td>
<td>SNVT_state_64</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>166</td>
<td>SNVT_nv_type</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>167</td>
<td>SNVT_ent_opmode</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>168</td>
<td>SNVT_ent_state</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>169</td>
<td>SNVT_ent_status</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>170</td>
<td>SNVT_flow_dir</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>171</td>
<td>SNVT阀模式</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>172</td>
<td>SNVT阀模式</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>173</td>
<td>SNVT_dev_status</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>174</td>
<td>SNVT阀模式</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>175</td>
<td>SNVT阀模式</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>176</td>
<td>SNVT阀模式</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>177</td>
<td>SNVT阀模式</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>178</td>
<td>SNVT阀模式</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>179</td>
<td>SNVT阀模式</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>180</td>
<td>SNVT阀模式</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>181</td>
<td>SNVT阀模式</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>182</td>
<td>SNVT阀模式</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>183</td>
<td>SNVT阀模式</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>184</td>
<td>SNVT阀模式</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>185</td>
<td>SNVT阀模式</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>186</td>
<td>SNVT阀模式</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>187</td>
<td>SNVT阀模式</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>188</td>
<td>SNVT阀模式</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>189</td>
<td>SNVT阀模式</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>190</td>
<td>SNVT阀模式</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>191</td>
<td>SNVT阀模式</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>192</td>
<td>SNVT阀模式</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>193</td>
<td>SNVT阀模式</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>194</td>
<td>SNVT阀模式</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>195</td>
<td>SNVT阀模式</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>196</td>
<td>SNVT阀模式</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>197</td>
<td>SNVT阀模式</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
<tr>
<td>198</td>
<td>SNVT阀模式</td>
<td>Read</td>
<td>Send</td>
<td></td>
</tr>
</tbody>
</table>
### 3.2 Configuration software KS2000

**Commissioning procedure**

1. Install the configuration software KS2000, version 4.3.0.39 or higher.
2. Plug the KL6401 into your Bus Coupler, and terminate the K-Bus with a KL9010.
3. Log on. Should the dialog box **Create new XML Device Files** be open, cancel at this point.
4. The terminal can be configured under **Settings** ( ).

![Configuration software KS2000](image)
5. Enter the SNVTs in the table (②).

Use ⑦ to add a variable and ⑧ to remove a selected variable.

The column Id cannot be edited. It contains the NV index. This index is also required as input variable "wNVIndex" for the PLC function blocks [⑨].

6. Enter any text in column Nv Name. These descriptions are required in the LON configuration tool (not provided by Beckhoff) for identifying the respective variables.

7. Select the required SNVT in column Snvt Type.

8. In column Dir (Direction) you can specify whether the variable is sent (out) or received (in).

9. Enter any project name (click on PROJECTNAME ③).

10. Edit the program ID (Prog ID ③). Do this by clicking Program ID, and selecting a Program ID. This program ID may only be issued once in your LON project. A maximum of 256 (0-255) program IDs can be used. This means that you can use 256 different configurations in one project.

LON terminals with the same configuration also have the same program ID.

11. Press the button ⑤ to create the Xif file. This file is required in the LON configuration tool (not provided by Beckhoff).

12. Use the button ⑥ to save the configuration in a BLC file. This file can be used to load the configuration if a terminal is replaced or to use the same configuration in another terminal.

The data can now be written to the terminal.

13. Use the buttons [>>>] or [<<<] to select the required slot, then use the button WRITE to write the data to the terminal. The slot may not be activated (green) during this process.

The terminal is delivered with active slot 0 (factory setting). This slot contains a fixed configuration and cannot be modified. To transfer other values a different slot has to be configured and activated. Only one slot at a time is active.

The project must be activated after the download has successfully been completed.

14. Press the button SET ACTIVE PROJECT.

15. De-energies the KL6401.

After switching on the KL6401 (switching the power supply on again) the desired configuration is active.

Description of the editing functions

① Deletes the table

② Opens an existing SNVT configuration file (*.BLC) for the KL6401

③ Saves the current configuration as a BLC file

④ Edits an SNVT variable

⑤ Inserts an SNVT variable

⑥ Deletes an SNVT variable
Creates an XIF file for a LON configuration tool (not provided by Beckhoff)

Opens the Help

Info
# 4 Programming

## 4.1 POUs

### Function blocks

<table>
<thead>
<tr>
<th>POU</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FB_LON_KL6401</td>
<td>Send / receive function block</td>
</tr>
</tbody>
</table>

### Read

<table>
<thead>
<tr>
<th>POUs</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FB_READ_001_SNVT_amp</td>
<td>Electric current (Amperes)</td>
</tr>
<tr>
<td>FB_READ_002_SNVT_amp_mil</td>
<td>Electric current (milliAmperes)</td>
</tr>
<tr>
<td>FB_READ_003_SNVT_angle</td>
<td>Angular distance (radians)</td>
</tr>
<tr>
<td>FB_READ_004_SNVT_angle_vel</td>
<td>Angular velocity (radians/second)</td>
</tr>
<tr>
<td>FB_READ_005_SNVT_btu_kilo</td>
<td>Thermal energy (kilo-Btus)</td>
</tr>
<tr>
<td>FB_READ_006_SNVT_btu_mega</td>
<td>Thermal energy (mega-Btus)</td>
</tr>
<tr>
<td>FB_READ_007_SNVT_char_ascii</td>
<td>ASCII character (8-bit ASCII character)</td>
</tr>
<tr>
<td>FB_READ_008_SNVT_count</td>
<td>Absolute count (units)</td>
</tr>
<tr>
<td>FB_READ_009_SNVT_count_inc</td>
<td>Increment count (units (delta))</td>
</tr>
<tr>
<td>FB_READ_011_SNVT_date_day</td>
<td>Day of week (day names)</td>
</tr>
<tr>
<td>FB_READ_013_SNVT_elec_kwh</td>
<td>Electric energy (kiloWatt-hours)</td>
</tr>
<tr>
<td>FB_READ_014_SNVT_elec_whr</td>
<td>Electric energy (Watt-hours)</td>
</tr>
<tr>
<td>FB_READ_015_SNVT_flow</td>
<td>Flow volume (liters/second)</td>
</tr>
<tr>
<td>FB_READ_016_SNVT_flow_mil</td>
<td>Flow volume (milliliters/second)</td>
</tr>
<tr>
<td>FB_READ_017_SNVT_length</td>
<td>Length (meters)</td>
</tr>
<tr>
<td>FB_READ_018_SNVT_length_kilo</td>
<td>Length (kilometers)</td>
</tr>
<tr>
<td>FB_READ_019_SNVT_length_micr</td>
<td>Length (micrometers (microns))</td>
</tr>
<tr>
<td>FB_READ_020_SNVT_length_mil</td>
<td>Length (millimeters)</td>
</tr>
<tr>
<td>FB_READ_021_SNVT_lev_cont</td>
<td>Continuous level (% of full level)</td>
</tr>
<tr>
<td>FB_READ_023_SNVT_mass</td>
<td>Mass (grams)</td>
</tr>
<tr>
<td>FB_READ_024_SNVT_mass_kilo</td>
<td>Mass (kilograms)</td>
</tr>
<tr>
<td>FB_READ_025_SNVT_mass_mega</td>
<td>Mass (metric tons)</td>
</tr>
<tr>
<td>FB_READ_026_SNVT_mass_mil</td>
<td>Mass (milligrams)</td>
</tr>
<tr>
<td>FB_READ_027_SNVT_power</td>
<td>Power (Watts)</td>
</tr>
<tr>
<td>FB_READ_028_SNVT_power_kilo</td>
<td>Power (kiloWatts)</td>
</tr>
<tr>
<td>FB_READ_029_SNVT_ppm</td>
<td>Concentration (ppm)</td>
</tr>
<tr>
<td>FB_READ_030_SNVT_press</td>
<td>Pressure (gauge) (kiloPascals)</td>
</tr>
<tr>
<td>FB_READ_031_SNVT_res</td>
<td>Electric resistance (Ohms)</td>
</tr>
<tr>
<td>FB_READ_032_SNVT_res_kilo</td>
<td>Electric resistance (kiloOhms)</td>
</tr>
<tr>
<td>FB_READ_033_SNVT_sound_db</td>
<td>Sound level (dB)</td>
</tr>
<tr>
<td>FB_READ_034_SNVT_speed</td>
<td>Linear velocity (meters/second)</td>
</tr>
<tr>
<td>FB_READ_035_SNVT_speed_mil</td>
<td>Linear velocity (meters/second)</td>
</tr>
<tr>
<td>FB_READ_036_SNVT_str_asc</td>
<td>Character string</td>
</tr>
<tr>
<td>FB_READ_037_SNVT_str_int</td>
<td>Wide character string with locale code</td>
</tr>
<tr>
<td>POUs</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>FB_READ_038_SNVT_telcom[62]</td>
<td>Telecomm states</td>
</tr>
<tr>
<td>FB_READ_039_SNVT_temp[63]</td>
<td>Temperature (degrees Celsius)</td>
</tr>
<tr>
<td>FB_READ_041_SNVT_vol[64]</td>
<td>Volume (liters)</td>
</tr>
<tr>
<td>FB_READ_042_SNVT_vol_kilo[65]</td>
<td>Volume (kiloliters)</td>
</tr>
<tr>
<td>FB_READ_043_SNVT_vol_mil[66]</td>
<td>Volume (milliliters)</td>
</tr>
<tr>
<td>FB_READ_044_SNVT_volt[66]</td>
<td>Electric voltage (Volts)</td>
</tr>
<tr>
<td>FB_READ_045_SNVT_volt_dbmv[67]</td>
<td>Electric voltage (dB microVolts)</td>
</tr>
<tr>
<td>FB_READ_046_SNVT_volt_kilo[68]</td>
<td>Electric voltage (kiloVolts)</td>
</tr>
<tr>
<td>FB_READ_047_SNVT_volt_mil[69]</td>
<td>Electric voltage (milliVolts)</td>
</tr>
<tr>
<td>FB_READ_048_SNVT_amp_f[70]</td>
<td>Electric current (Amperes)</td>
</tr>
<tr>
<td>FB_READ_049_SNVT_angle_f[71]</td>
<td>Angular distance (radians)</td>
</tr>
<tr>
<td>FB_READ_050_SNVT_angle_vel_f[72]</td>
<td>Angular velocity (radians/second)</td>
</tr>
<tr>
<td>FB_READ_051_SNVT_count_f[72]</td>
<td>Absolute count (units)</td>
</tr>
<tr>
<td>FB_READ_052_SNVT_count_inc_f[73]</td>
<td>Increment count (units (delta))</td>
</tr>
<tr>
<td>FB_READ_053_SNVT_flow_f[74]</td>
<td>Flow volume (liters/second)</td>
</tr>
<tr>
<td>FB_READ_054_SNVT_length_f[75]</td>
<td>Length (meters)</td>
</tr>
<tr>
<td>FB_READ_055_SNVT_lev_cont_f[76]</td>
<td>Continuous level</td>
</tr>
<tr>
<td>FB_READ_056_SNVT_mass_f[77]</td>
<td>Mass (grams)</td>
</tr>
<tr>
<td>FB_READ_057_SNVT_power_f[78]</td>
<td>Power (Watts)</td>
</tr>
<tr>
<td>FB_READ_058_SNVT_ppm_f[78]</td>
<td>Concentration (ppm)</td>
</tr>
<tr>
<td>FB_READ_059_SNVT_press_f[79]</td>
<td>Pressure (gauge) (Pascals)</td>
</tr>
<tr>
<td>FB_READ_060_SNVT_res_f[80]</td>
<td>Electric resistance (Ohms)</td>
</tr>
<tr>
<td>FB_READ_061_SNVT_sound_db_f[81]</td>
<td>Sound level (dBSpl)</td>
</tr>
<tr>
<td>FB_READ_062_SNVT_speed_f[82]</td>
<td>Linear velocity (meters/second)</td>
</tr>
<tr>
<td>FB_READ_063_SNVT_temp_f[83]</td>
<td>Temperature (degrees Celsius)</td>
</tr>
<tr>
<td>FB_READ_064_SNVT_time_f[84]</td>
<td>Elapsed time (seconds)</td>
</tr>
<tr>
<td>FB_READ_065_SNVT_vol_f[84]</td>
<td>Volume (liters)</td>
</tr>
<tr>
<td>FB_READ_066_SNVT_volt_f[85]</td>
<td>Electric voltage (Volts)</td>
</tr>
<tr>
<td>FB_READ_067_SNVT_btu_f[86]</td>
<td>Thermal energy (Btus)</td>
</tr>
<tr>
<td>FB_READ_068_SNVT_elec_whr_f[87]</td>
<td>Electric energy (Watt-hours)</td>
</tr>
<tr>
<td>FB_READ_069_SNVT_config_src[88]</td>
<td>Configuration source</td>
</tr>
<tr>
<td>FB_READ_070_SNVT_color[89]</td>
<td>CIELAB color (L*,a*,b)</td>
</tr>
<tr>
<td>FB_READ_071_SNVT_grammage[90]</td>
<td>Grammage (grams/sq meter)</td>
</tr>
<tr>
<td>FB_READ_072_SNVT_grammage_f[90]</td>
<td>Grammage (grams/sq meter)</td>
</tr>
<tr>
<td>FB_READ_073_SNVT_file_req[91]</td>
<td>File request</td>
</tr>
<tr>
<td>FB_READ_074_SNVT_file_status[92]</td>
<td>File status</td>
</tr>
<tr>
<td>FB_READ_075_SNVT_freq_f[93]</td>
<td>Frequency (Hertz)</td>
</tr>
<tr>
<td>FB_READ_076_SNVT_freq_hz[94]</td>
<td>Frequency (Hertz)</td>
</tr>
<tr>
<td>FB_READ_077_SNVT_freq_kilohz[95]</td>
<td>Frequency (kiloHertz)</td>
</tr>
<tr>
<td>FB_READ_078_SNVT_freq_milhz[96]</td>
<td>Frequency (Herzt)</td>
</tr>
<tr>
<td>FB_READ_079_SNVT_lux[96]</td>
<td>Illumination (lux)</td>
</tr>
<tr>
<td>FB_READ_081_SNVT_lev_percent[97]</td>
<td>Percentage level (% of full level)</td>
</tr>
<tr>
<td>FB_READ_082_SNVT_multiplier[98]</td>
<td>Multiplier</td>
</tr>
<tr>
<td>POUs</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------</td>
</tr>
<tr>
<td>FB_READ_083_SNVT_state [99]</td>
<td>State vector</td>
</tr>
<tr>
<td>FB_READ_084_SNVT_time_stamp [100]</td>
<td>Time stamp</td>
</tr>
<tr>
<td>FB_READ_085_SNVT_zerospan [101]</td>
<td>Zero and span</td>
</tr>
<tr>
<td>FB_READ_086_SNVT_magcard [102]</td>
<td>ISO 7811 (40 hexadecimal digits)</td>
</tr>
<tr>
<td>FB_READ_087_SNVT_elapsed_tm [103]</td>
<td>Elapsed time</td>
</tr>
<tr>
<td>FB_READ_088_SNVT_alarm [103]</td>
<td>Alarm status</td>
</tr>
<tr>
<td>FB_READ_089_SNVT_currency [104]</td>
<td>Currency</td>
</tr>
<tr>
<td>FB_READ_090_SNVT_file_pos [105]</td>
<td>File position</td>
</tr>
<tr>
<td>FB_READ_091_SNVT_muldiv [106]</td>
<td>Multiply/Divide</td>
</tr>
<tr>
<td>FB_READ_092_SNVT_obj_request [107]</td>
<td>Object request</td>
</tr>
<tr>
<td>FB_READ_093_SNVT_obj_status [108]</td>
<td>Object status</td>
</tr>
<tr>
<td>FB_READ_094_SNVT_preset [109]</td>
<td>Preset</td>
</tr>
<tr>
<td>FB_READ_095_SNVT_switch [109]</td>
<td>Switch</td>
</tr>
<tr>
<td>FB_READ_096_SNVT_trans_table [110]</td>
<td>Translation table</td>
</tr>
<tr>
<td>FB_READ_097_SNVT_override [111]</td>
<td>Override code</td>
</tr>
<tr>
<td>FB_READ_098_SNVT_pwr_fact [112]</td>
<td>Power factor</td>
</tr>
<tr>
<td>FB_READ_099_SNVT_pwr_fact_f [113]</td>
<td>Power factor (multiplier)</td>
</tr>
<tr>
<td>FB_READ_100_SNVT_density [114]</td>
<td>Density (kilograms/cubic meter)</td>
</tr>
<tr>
<td>FB_READ_101_SNVT_density_f [115]</td>
<td>Density (kilograms/cubic meter)</td>
</tr>
<tr>
<td>FB_READ_102_SNVT_rpm [115]</td>
<td>Angular velocity (revolutions/minute (RPM))</td>
</tr>
<tr>
<td>FB_READ_103_SNVT_hvac_emerg [116]</td>
<td>HVAC emergency mode</td>
</tr>
<tr>
<td>FB_READ_104_SNVT_angle_deg [117]</td>
<td>Angular distance(degrees)</td>
</tr>
<tr>
<td>FB_READ_105_SNVT_temp_p [118]</td>
<td>Temperature (degrees Celsius)</td>
</tr>
<tr>
<td>FB_READ_106_SNVT_temp_setpt [119]</td>
<td>Temperature</td>
</tr>
<tr>
<td>FB_READ_107_SNVT_time_sec [120]</td>
<td>Elapsed time (seconds)</td>
</tr>
<tr>
<td>FB_READ_108_SNVT_hvac_mode [121]</td>
<td>HVAC mode</td>
</tr>
<tr>
<td>FB_READ_109_SNVT_occupancy [121]</td>
<td>Occupancy</td>
</tr>
<tr>
<td>FB_READ_110_SNVT_area [122]</td>
<td>Area (square meters)</td>
</tr>
<tr>
<td>FB_READ_111_SNVT_hvac_ovenid [123]</td>
<td>HVAC override</td>
</tr>
<tr>
<td>FB_READ_112_SNVT_hvac_status [124]</td>
<td>HVAC status</td>
</tr>
<tr>
<td>FB_READ_113_SNVT_press_p [125]</td>
<td>Pressure</td>
</tr>
<tr>
<td>FB_READ_114_SNVT_address [126]</td>
<td>Neuron address</td>
</tr>
<tr>
<td>FB_READ_115_SNVT_scene [127]</td>
<td>Scene control</td>
</tr>
<tr>
<td>FB_READ_116_SNVT_scene_cfg [127]</td>
<td>Scene configuration</td>
</tr>
<tr>
<td>FB_READ_117_SNVT_setting [128]</td>
<td>Setting control</td>
</tr>
<tr>
<td>FB_READ_118_SNVT_evap_state [129]</td>
<td>Evaporator state</td>
</tr>
<tr>
<td>FB_READ_119_SNVT_therm_mode [130]</td>
<td>Thermostat mode</td>
</tr>
<tr>
<td>FB_READ_120_SNVT_defr_mode [131]</td>
<td>Defrost mode</td>
</tr>
<tr>
<td>FB_READ_121_SNVT_defr_term [132]</td>
<td>Defrost termination</td>
</tr>
<tr>
<td>FB_READ_122_SNVT_defr_state [133]</td>
<td>Defrost state</td>
</tr>
<tr>
<td>FB_READ_123_SNVT_time_min [133]</td>
<td>Elapsed time</td>
</tr>
<tr>
<td>FB_READ_124_SNVT_time_hour [134]</td>
<td>Elapsed time</td>
</tr>
<tr>
<td>FB_READ_125_SNVT_ph [135]</td>
<td>Acidity (pH)</td>
</tr>
<tr>
<td>POUs</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>FB_READ_126_SNVT_ph_f [136]</td>
<td>Acidity (pH)</td>
</tr>
<tr>
<td>FB_READ_127_SNVT_chlr_status [137]</td>
<td>Chiller status</td>
</tr>
<tr>
<td>FB_READ_128_SNVT_tod_event [138]</td>
<td>Time of day event</td>
</tr>
<tr>
<td>FB_READ_129_SNVT_smo_obscur [139]</td>
<td>Smoke obscuration</td>
</tr>
<tr>
<td>FB_READ_130_SNVT_fire_test [139]</td>
<td>Fire test request</td>
</tr>
<tr>
<td>FB_READ_131_SNVT_temp_ror [140]</td>
<td>Temperature rate of change/rise</td>
</tr>
<tr>
<td>FB_READ_132_SNVT_fire_init [141]</td>
<td>Fire initiator type</td>
</tr>
<tr>
<td>FB_READ_133_SNVT_fire_indcte [142]</td>
<td>Fire indicator type</td>
</tr>
<tr>
<td>FB_READ_134_SNVT_time_zone [143]</td>
<td>Time zone descriptor</td>
</tr>
<tr>
<td>FB_READ_135_SNVT_earth_pos [144]</td>
<td>Earth position</td>
</tr>
<tr>
<td>FB_READ_136_SNVT_reg_val [145]</td>
<td>Register value</td>
</tr>
<tr>
<td>FB_READ_137_SNVT_reg_val_ts [145]</td>
<td>Register value</td>
</tr>
<tr>
<td>FB_READ_138_SNVT_volt_ac [146]</td>
<td>Voltage in alternating current</td>
</tr>
<tr>
<td>FB_READ_139_SNVT_amp_ac [147]</td>
<td>Amperage in alternating current</td>
</tr>
<tr>
<td>FB_READ_143_SNVT_turbidity [148]</td>
<td>Turbidity</td>
</tr>
<tr>
<td>FB_READ_144_SNVT_turbidity_f [149]</td>
<td>Turbidity</td>
</tr>
<tr>
<td>FB_READ_145_SNVT_hvac_type [150]</td>
<td>HVAC unit type</td>
</tr>
<tr>
<td>FB_READ_146_SNVT_elec_kwh_l [151]</td>
<td>Electric energy</td>
</tr>
<tr>
<td>FB_READ_147_SNVT_temp_diff_p [151]</td>
<td>Temp difference</td>
</tr>
<tr>
<td>FB_READ_148_SNVT_ctrl_req [152]</td>
<td>Control request</td>
</tr>
<tr>
<td>FB_READ_149_SNVT_ctrl_resp [153]</td>
<td>Control response</td>
</tr>
<tr>
<td>FB_READ_150_SNVT_ptz [154]</td>
<td>Camera PTZ</td>
</tr>
<tr>
<td>FB_READ_151_SNVT_privacyzone [155]</td>
<td>Privacy zone</td>
</tr>
<tr>
<td>FB_READ_152_SNVT_pos_ctrl [156]</td>
<td>Position control</td>
</tr>
<tr>
<td>FB_READ_153_SNVT_enthalpy [157]</td>
<td>Enthalpy (kiloJoules/kg)</td>
</tr>
<tr>
<td>FB_READ_154_SNVT_gfci_status [157]</td>
<td>GFCI status type</td>
</tr>
<tr>
<td>FB_READ_155_SNVT_motor_state [158]</td>
<td>Motor state</td>
</tr>
<tr>
<td>FB_READ_156_SNVT_pumpset_mn [159]</td>
<td>Pumpset</td>
</tr>
<tr>
<td>FB_READ_157_SNVT_ex_control [160]</td>
<td>Exclusive control</td>
</tr>
<tr>
<td>FB_READ_158_SNVT_pumpset_sn [161]</td>
<td>Pumpset sensor</td>
</tr>
<tr>
<td>FB_READ_159_SNVT_pump_sensor [162]</td>
<td>Pump sensor</td>
</tr>
<tr>
<td>FB_READ_160_SNVT_abs_humid [163]</td>
<td>Absolute humidity</td>
</tr>
<tr>
<td>FB_READ_161_SNVT_flow_p [163]</td>
<td>Flow volume</td>
</tr>
<tr>
<td>FB_READ_162_SNVT_dev_c_mode [164]</td>
<td>Device control mode</td>
</tr>
<tr>
<td>FB_READ_163_SNVT_valve_mode [165]</td>
<td>Valve mode</td>
</tr>
<tr>
<td>FB_READ_164_SNVT_alarm_2 [166]</td>
<td>Alarm status 2</td>
</tr>
<tr>
<td>FB_READ_165_SNVT_state_64 [167]</td>
<td>State vector</td>
</tr>
<tr>
<td>FB_READ_166_SNVT_nv_type [168]</td>
<td>Network variable type</td>
</tr>
<tr>
<td>FB_READ_168_SNVT_ent_opmode [169]</td>
<td>Entry operation mode</td>
</tr>
<tr>
<td>FB_READ_169_SNVT_ent_state [169]</td>
<td>Entry state</td>
</tr>
<tr>
<td>FB_READ_170_SNVT_ent_status [170]</td>
<td>Entry status</td>
</tr>
<tr>
<td>FB_READ_171_SNVT_flow_dir [171]</td>
<td>Flow direction</td>
</tr>
<tr>
<td>FB_READ_172_SNVT_hvac_satsts [172]</td>
<td>HVAC saturation status</td>
</tr>
</tbody>
</table>
### POUs

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device status</td>
</tr>
<tr>
<td>Device fault states</td>
</tr>
<tr>
<td>Device maintenance</td>
</tr>
<tr>
<td>Date event</td>
</tr>
<tr>
<td>Scheduler value</td>
</tr>
<tr>
<td>Rail-Audio Controller Control</td>
</tr>
<tr>
<td>Rail-Audio Controller Request</td>
</tr>
<tr>
<td>Absolute count</td>
</tr>
<tr>
<td>Clothes Washer Command</td>
</tr>
<tr>
<td>Clothes Washer-Management Status</td>
</tr>
<tr>
<td>Clothes Washer Status</td>
</tr>
<tr>
<td>Clothes Washer Alarm</td>
</tr>
<tr>
<td>Value multiplier</td>
</tr>
<tr>
<td>Switch with scene and setting control.</td>
</tr>
<tr>
<td>Log status (hundredths of second)</td>
</tr>
<tr>
<td>Precision timestamp.</td>
</tr>
<tr>
<td>Log file transfer request.</td>
</tr>
<tr>
<td>Log file transfer status.</td>
</tr>
<tr>
<td>Log status request.</td>
</tr>
<tr>
<td>Enthalpy difference (kJ/kg)</td>
</tr>
<tr>
<td>Electrical current (milliAmperes)</td>
</tr>
<tr>
<td>Time hour (hours)</td>
</tr>
<tr>
<td>Lamp Status.</td>
</tr>
<tr>
<td>Environment.</td>
</tr>
<tr>
<td>Geographic Location.</td>
</tr>
</tbody>
</table>

### Send

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electric current (ampere)</td>
</tr>
<tr>
<td>Electric current (milliampere)</td>
</tr>
<tr>
<td>Angular distance (radian)</td>
</tr>
<tr>
<td>Angular velocity (radian / second)</td>
</tr>
<tr>
<td>Thermal energy (kiloBtu)</td>
</tr>
<tr>
<td>Thermal energy (megaBtu)</td>
</tr>
<tr>
<td>ASCII character (8-bit ASCII character)</td>
</tr>
<tr>
<td>Absolute number (items)</td>
</tr>
<tr>
<td>Increment counter (items(delta))</td>
</tr>
<tr>
<td>Day of the week</td>
</tr>
<tr>
<td>Electric energy (kW / hour)</td>
</tr>
<tr>
<td>Electric energy (Watt / hour)</td>
</tr>
<tr>
<td>Volume flow (liters / second)</td>
</tr>
<tr>
<td>Function blocks</td>
</tr>
<tr>
<td>------------------------------------</td>
</tr>
<tr>
<td>FB_SEND_016_SNVT_flow_mil</td>
</tr>
<tr>
<td>FB_SEND_017_SNVT_length</td>
</tr>
<tr>
<td>FB_SEND_018_SNVT_length_kilo</td>
</tr>
<tr>
<td>FB_SEND_019_SNVT_length_micr</td>
</tr>
<tr>
<td>FB_SEND_020_SNVT_length_mil</td>
</tr>
<tr>
<td>FB_SEND_021_SNVT_lev_cont</td>
</tr>
<tr>
<td>FB_SEND_023_SNVT_mass</td>
</tr>
<tr>
<td>FB_SEND_024_SNVT_mass_kilo</td>
</tr>
<tr>
<td>FB_SEND_025_SNVT_mass_mega</td>
</tr>
<tr>
<td>FB_SEND_026_SNVT_mass_mil</td>
</tr>
<tr>
<td>FB_SEND_027_SNVT_power</td>
</tr>
<tr>
<td>FB_SEND_028_SNVT_power_kilo</td>
</tr>
<tr>
<td>FB_SEND_029_SNVT_ppm</td>
</tr>
<tr>
<td>FB_SEND_030_SNVT_press</td>
</tr>
<tr>
<td>FB_SEND_031_SNVT_res</td>
</tr>
<tr>
<td>FB_SEND_032_SNVT_res_kilo</td>
</tr>
<tr>
<td>FB_SEND_033_SNVT_sound_db</td>
</tr>
<tr>
<td>FB_SEND_034_SNVT_speed</td>
</tr>
<tr>
<td>FB_SEND_035_SNVT_speed_mil</td>
</tr>
<tr>
<td>FB_SEND_036_SNVT_str_asc</td>
</tr>
<tr>
<td>FB_SEND_037_SNVT_str_int</td>
</tr>
<tr>
<td>FB_SEND_038_SNVT_telcom</td>
</tr>
<tr>
<td>FB_SEND_039_SNVT_temp</td>
</tr>
<tr>
<td>FB_SEND_041_SNVT_vol</td>
</tr>
<tr>
<td>FB_SEND_042_SNVT_vol_kilo</td>
</tr>
<tr>
<td>FB_SEND_043_SNVT_vol_mil</td>
</tr>
<tr>
<td>FB_SEND_044_SNVT_volt</td>
</tr>
<tr>
<td>FB_SEND_045_SNVT_volt_dbmv</td>
</tr>
<tr>
<td>FB_SEND_046_SNVT_volt_kilo</td>
</tr>
<tr>
<td>FB_SEND_047_SNVT_volt_mil</td>
</tr>
<tr>
<td>FB_SEND_048_SNVT_amp_f</td>
</tr>
<tr>
<td>FB_SEND_049_SNVT_angle_f</td>
</tr>
<tr>
<td>FB_SEND_050_SNVT_angle_vel_f</td>
</tr>
<tr>
<td>FB_SEND_051_SNVT_count_f</td>
</tr>
<tr>
<td>FB_SEND_052_SNVT_count_inc_f</td>
</tr>
<tr>
<td>FB_SEND_053_SNVT_flow_f</td>
</tr>
<tr>
<td>FB_SEND_054_SNVT_length_f</td>
</tr>
<tr>
<td>FB_SEND_055_SNVT_lev_cont_f</td>
</tr>
<tr>
<td>FB_SEND_056_SNVT_mass_f</td>
</tr>
<tr>
<td>FB_SEND_057_SNVT_power_f</td>
</tr>
<tr>
<td>FB_SEND_058_SNVT_ppm_f</td>
</tr>
<tr>
<td>FB_SEND_059_SNVT_press_f</td>
</tr>
<tr>
<td>FB_SEND_060_SNVT_res_f</td>
</tr>
<tr>
<td>Function blocks</td>
</tr>
<tr>
<td>-----------------</td>
</tr>
<tr>
<td>FB_SEND_061_SNVT_sound_db_f</td>
</tr>
<tr>
<td>FB_SEND_062_SNVT_speed_f</td>
</tr>
<tr>
<td>FB_SEND_063_SNVT_temp_f</td>
</tr>
<tr>
<td>FB_SEND_064_SNVT_time_f</td>
</tr>
<tr>
<td>FB_SEND_065_SNVT_vol_f</td>
</tr>
<tr>
<td>FB_SEND_066_SNVT_volt_f</td>
</tr>
<tr>
<td>FB_SEND_068_SNVT_elec_whr_f</td>
</tr>
<tr>
<td>FB_SEND_069_SNVT_config_src</td>
</tr>
<tr>
<td>FB_SEND_070_SNVT_color</td>
</tr>
<tr>
<td>FB_SEND_071_SNVT_grammage</td>
</tr>
<tr>
<td>FB_SEND_072_SNVT_grammage_f</td>
</tr>
<tr>
<td>FB_SEND_073_SNVT_file_req</td>
</tr>
<tr>
<td>FB_SEND_074_SNVT_file_status</td>
</tr>
<tr>
<td>FB_SEND_075_SNVT_freq_f</td>
</tr>
<tr>
<td>FB_SEND_076_SNVT_freq_hz</td>
</tr>
<tr>
<td>FB_SEND_077_SNVT_freq_kilohz</td>
</tr>
<tr>
<td>FB_SEND_078_SNVT_freq_milhz</td>
</tr>
<tr>
<td>FB_SEND_079_SNVT_lux</td>
</tr>
<tr>
<td>FB_SEND_081_SNVT_lev_percent</td>
</tr>
<tr>
<td>FB_SEND_082_SNVT_multiplier</td>
</tr>
<tr>
<td>FB_SEND_083_SNVT_state</td>
</tr>
<tr>
<td>FB_SEND_084_SNVT_time_stamp</td>
</tr>
<tr>
<td>FB_SEND_085_SNVT_zerospan</td>
</tr>
<tr>
<td>FB_SEND_086_SNVT_magcard</td>
</tr>
<tr>
<td>FB_SEND_087_SNVT_elapsed_tm</td>
</tr>
<tr>
<td>FB_SEND_088_SNVT_alarm</td>
</tr>
<tr>
<td>FB_SEND_089_SNVT_currency</td>
</tr>
<tr>
<td>FB_SEND_090_SNVT_file_pos</td>
</tr>
<tr>
<td>FB_SEND_091_SNVT_muldiv</td>
</tr>
<tr>
<td>FB_SEND_092_SNVT_obj_request</td>
</tr>
<tr>
<td>FB_SEND_093_SNVT_obj_status</td>
</tr>
<tr>
<td>FB_SEND_094_SNVT_preset</td>
</tr>
<tr>
<td>FB_SEND_095_SNVT_switch</td>
</tr>
<tr>
<td>FB_SEND_096_SNVT_trans_table</td>
</tr>
<tr>
<td>FB_SEND_097_SNVT_override</td>
</tr>
<tr>
<td>FB_SEND_098_SNVT_pwr_fact</td>
</tr>
<tr>
<td>FB_SEND_099_SNVT_pwr_fact_f</td>
</tr>
<tr>
<td>FB_SEND_100_SNVT_density</td>
</tr>
<tr>
<td>FB_SEND_101_SNVT_density_f</td>
</tr>
<tr>
<td>FB_SEND_102_SNVT_rpm</td>
</tr>
<tr>
<td>FB_SEND_103_SNVT_hvac_emerg</td>
</tr>
<tr>
<td>FB_SEND_104_SNVT_angle_deg</td>
</tr>
<tr>
<td>Function blocks</td>
</tr>
<tr>
<td>-------------------------------------</td>
</tr>
<tr>
<td>FB_SEND_105_SNVT_temp_p [339]</td>
</tr>
<tr>
<td>FB_SEND_106_SNVT_temp_setpt [340]</td>
</tr>
<tr>
<td>FB_SEND_107_SNVT_time_sec [341]</td>
</tr>
<tr>
<td>FB_SEND_108_SNVT_hvac_mode [343]</td>
</tr>
<tr>
<td>FB_SEND_109_SNVT_occupancy [344]</td>
</tr>
<tr>
<td>FB_SEND_110_SNVT_area [345]</td>
</tr>
<tr>
<td>FB_SEND_111_SNVT_hvac_overid [347]</td>
</tr>
<tr>
<td>FB_SEND_112_SNVT_hvac_status [348]</td>
</tr>
<tr>
<td>FB_SEND_113_SNVT_press_p [349]</td>
</tr>
<tr>
<td>FB_SEND_114_SNVT_address [351]</td>
</tr>
<tr>
<td>FB_SEND_115_SNVT_scene [352]</td>
</tr>
<tr>
<td>FB_SEND_116_SNVT_scene_cfg [353]</td>
</tr>
<tr>
<td>FB_SEND_117_SNVT_setting [355]</td>
</tr>
<tr>
<td>FB_SEND_118_SNVT_evap_state [356]</td>
</tr>
<tr>
<td>FB_SEND_119_SNVT_therm_mode [357]</td>
</tr>
<tr>
<td>FB_SEND_120_SNVT_defr_mode [359]</td>
</tr>
<tr>
<td>FB_SEND_121_SNVT_defr_term [360]</td>
</tr>
<tr>
<td>FB_SEND_122_SNVT_defr_state [361]</td>
</tr>
<tr>
<td>FB_SEND_123_SNVT_time_min [363]</td>
</tr>
<tr>
<td>FB_SEND_124_SNVT_time_hour [364]</td>
</tr>
<tr>
<td>FB_SEND_125_SNVT_ph [365]</td>
</tr>
<tr>
<td>FB_SEND_126_SNVT_ph_f [367]</td>
</tr>
<tr>
<td>FB_SEND_127_SNVT_chlr_status [368]</td>
</tr>
<tr>
<td>FB_SEND_128_SNVT_tod_event [369]</td>
</tr>
<tr>
<td>FB_SEND_129_SNVT_smo_obscur [371]</td>
</tr>
<tr>
<td>FB_SEND_130_SNVT_fire_test [372]</td>
</tr>
<tr>
<td>FB_SEND_131_SNVT_temp_ror [373]</td>
</tr>
<tr>
<td>FB_SEND_132_SNVT_fire_init [375]</td>
</tr>
<tr>
<td>FB_SEND_133_SNVT_fire_indcte [376]</td>
</tr>
<tr>
<td>FB_SEND_134_SNVT_time_zone [377]</td>
</tr>
<tr>
<td>FB_SEND_135_SNVT_earth_pos [379]</td>
</tr>
<tr>
<td>FB_SEND_136_SNVT_reg_val [380]</td>
</tr>
<tr>
<td>FB_SEND_137_SNVT_reg_val_ts [381]</td>
</tr>
<tr>
<td>FB_SEND_138_SNVT_volt_ac [383]</td>
</tr>
<tr>
<td>FB_SEND_139_SNVT_amp_ac [384]</td>
</tr>
<tr>
<td>FB_SEND_143_SNVT_turbidity [385]</td>
</tr>
<tr>
<td>FB_SEND_144_SNVT_turbidity_f [387]</td>
</tr>
<tr>
<td>FB_SEND_145_SNVT_hvac_type [388]</td>
</tr>
<tr>
<td>FB_SEND_146_SNVT_elec_kwh_l [389]</td>
</tr>
<tr>
<td>FB_SEND_147_SNVT_temp_diff_p [391]</td>
</tr>
<tr>
<td>FB_SEND_148_SNVT_ctrl_req [392]</td>
</tr>
<tr>
<td>FB_SEND_149_SNVT_ctrl_RESP [393]</td>
</tr>
<tr>
<td>FB_SEND_150_SNVT_ptz [395]</td>
</tr>
<tr>
<td>Function blocks</td>
</tr>
<tr>
<td>------------------------------------------------------</td>
</tr>
<tr>
<td>FB_SEND_151_SNVT_privacyzone [396]</td>
</tr>
<tr>
<td>FB_SEND_152_SNVT_pos_ctrl [397]</td>
</tr>
<tr>
<td>FBSEND_153_SNVT_enthalpy [399]</td>
</tr>
<tr>
<td>FBSEND_154_SNVT_gfci_status [400]</td>
</tr>
<tr>
<td>FBSEND_155_SNVT_motor_state [401]</td>
</tr>
<tr>
<td>FBSEND_156_SNVT_pumpset_mn [403]</td>
</tr>
<tr>
<td>FBSEND_157_SNVT_ex_control [404]</td>
</tr>
<tr>
<td>FBSEND_158_SNVT_pumpset_sn [405]</td>
</tr>
<tr>
<td>FBSEND_159_SNVT_pump_sensor [407]</td>
</tr>
<tr>
<td>FBSEND_160_SNVT_abs_humid [408]</td>
</tr>
<tr>
<td>FBSEND_161_SNVT_flow_p [409]</td>
</tr>
<tr>
<td>FBSEND_162_SNVT_dev_c_mode [411]</td>
</tr>
<tr>
<td>FBSEND_163_SNVT_valve_mode [412]</td>
</tr>
<tr>
<td>FBSEND_164_SNVT_alarm_2 [413]</td>
</tr>
<tr>
<td>FBSEND_165_SNVT_state_64 [415]</td>
</tr>
<tr>
<td>FBSEND_166_SNVT_nv_type [416]</td>
</tr>
<tr>
<td>FBSEND_168_SNVT_ent_opmode [417]</td>
</tr>
<tr>
<td>FBSEND_169_SNVT_ent_state [419]</td>
</tr>
<tr>
<td>FBSEND_170_SNVT_ent_status [420]</td>
</tr>
<tr>
<td>FBSEND_171_SNVT_flow_dir [421]</td>
</tr>
<tr>
<td>FBSEND_172_SNVT_hvac_satsts [423]</td>
</tr>
<tr>
<td>FBSEND_173_SNVT_dev_status [424]</td>
</tr>
<tr>
<td>FBSEND_174_SNVT_dev_fault [425]</td>
</tr>
<tr>
<td>FBSEND_175_SNVT_dev_maint [427]</td>
</tr>
<tr>
<td>FBSEND_176_SNVT_date_event [428]</td>
</tr>
<tr>
<td>FBSEND_177_SNVT_sched_val [429]</td>
</tr>
<tr>
<td>FBSEND_180_SNVT_sblind_state [431]</td>
</tr>
<tr>
<td>FBSEND_181_SNVT_rac_ctrl [432]</td>
</tr>
<tr>
<td>FBSEND_182_SNVT_rac_req [433]</td>
</tr>
<tr>
<td>FBSEND_183_SNVT_count_32 [435]</td>
</tr>
<tr>
<td>FBSEND_184_SNVT_clothes_w_c [436]</td>
</tr>
<tr>
<td>FBSEND_185_SNVT_clothes_w_m [437]</td>
</tr>
<tr>
<td>FBSEND_186_SNVT_clothes_w_s [439]</td>
</tr>
<tr>
<td>FBSEND_187_SNVT_clothes_w_a [440]</td>
</tr>
<tr>
<td>FBSEND_188_SNVT_multiplier_s [441]</td>
</tr>
<tr>
<td>FBSEND_189_SNVT_switch_2 [443]</td>
</tr>
<tr>
<td>FBSEND_190_SNVT_color_2 [444]</td>
</tr>
<tr>
<td>FBSEND_191_SNVT_log_status [445]</td>
</tr>
<tr>
<td>FBSEND_192_SNVT_time_stamp_p [447]</td>
</tr>
<tr>
<td>FBSEND_193_SNVT_log_fx_request [448]</td>
</tr>
<tr>
<td>FBSEND_194_SNVT_log_fx_status [449]</td>
</tr>
<tr>
<td>FBSEND_195_SNVT_log_request [451]</td>
</tr>
<tr>
<td>FBSEND_196_SNVT_enthalpy_d [452]</td>
</tr>
</tbody>
</table>
Function blocks | Description
--- | ---
FB_SEND_197_SNVT_amp_ac_mil | Electric current (milliampere)
FB_SEND_198_SNVT_time_hour_p | Time in hours
FB_SEND_199_SNVT_lamp_status | Lamp status
FB_SEND_200_SNVT_environment | Environment
FB_SEND_201_SNVT_geo_loc | Geographical location

4.1.1 FB_LON_KL6401

**Application**

This function block is used for sending / receiving LON SNVTs via Bus Terminal KL6401. An instance of this block is required for each terminal. One instance can send or receive up to 62 SNVTs.

The FB must be tied to the send/receive blocks via the VAR_IN_OUT structure stLON_Com.

**Restrictions:**

- This function block may only be called once for each KL6401!
- It must be called in the same task as the send and receive function blocks!
- In the PLC project this block may only be called once per cycle!

**VAR_INPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bResetMaxBuffer</td>
<td>Deletes the value rMaxBuffer for the maximum utilization of the send buffer.</td>
</tr>
<tr>
<td>bDisabled</td>
<td>TRUE = deselection of the block.</td>
</tr>
</tbody>
</table>

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bReady</td>
<td></td>
</tr>
<tr>
<td>bBusy</td>
<td></td>
</tr>
<tr>
<td>bTimeOut</td>
<td></td>
</tr>
<tr>
<td>bError</td>
<td></td>
</tr>
<tr>
<td>eError</td>
<td></td>
</tr>
<tr>
<td>dwErrorKL</td>
<td></td>
</tr>
<tr>
<td>byLONState</td>
<td></td>
</tr>
<tr>
<td>stKL6401</td>
<td></td>
</tr>
<tr>
<td>arrNeuronID</td>
<td></td>
</tr>
<tr>
<td>arrProgID</td>
<td></td>
</tr>
<tr>
<td>byActBuffer</td>
<td></td>
</tr>
<tr>
<td>rActBuffer</td>
<td></td>
</tr>
<tr>
<td>rMaxBuffer</td>
<td></td>
</tr>
</tbody>
</table>
bReady: The initialization is complete.
bBusy: The output is TRUE as long as values are sent.
bTimeOut: Timeout during initialization.
bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.
eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.
dwErrorKL: Error ID [631] of the function block. In this case the variable eError has the value "eKL6401_Error". bError goes TRUE at the same time.
byLONState: LON status.
stKL6401: Structure for the configuration (for future applications) (see ST_KL6401 [589]).
arrNeuronID: Neuron ID of the terminal (supported from firmware 4C).
arrProgID: Program ID of the terminal (supported from firmware 4C).
wFW: Terminal firmware (supported from firmware 4C).
byActBuffer: Number of instructions in the send buffer.
rActBuffer: Current utilization of the send buffer in percent.
rMaxBuffer: Maximum utilization of the send buffer in percent. The value can be cancelled with the input variable bResetMaxBuffer.

VAR_IN_OUT
stParameter_IN : ST_LON_Parameter_IN_36B;
stParameter_OUT : ST_LON_Parameter_OUT_36B;
stLON_Com : ST_LON_Communication;

stParameter_IN : Input variable of the hardware (see ST_LON_Parameter_IN_36B [585]).
stParameter_OUT : Output variable of the hardware (see ST_LON_Parameter_OUT_36B [586]).
stLON_Com : This structure is used to link the function block with the send/receive function blocks (see ST_LON_Communication [587]). The actual sending and receiving is realized in this function block. The send/receive blocks only deal with preparation/evaluation of the data.

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2 Read

<table>
<thead>
<tr>
<th>Function blocks</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FB_READ 001 SNVT_amp [33]</td>
<td>Electric current (ampere)</td>
</tr>
<tr>
<td>FB_READ 002 SNVT_amp_mil [34]</td>
<td>Electric current (milliampere)</td>
</tr>
<tr>
<td>FB_READ 003 SNVT_angle [35]</td>
<td>Angular distance (radian)</td>
</tr>
<tr>
<td>FB_READ 004 SNVT_angle_vel [36]</td>
<td>Angular velocity (radian / second)</td>
</tr>
<tr>
<td>FB_READ 005 SNVT_btu_kilo [36]</td>
<td>Thermal energy (kiloBtu)</td>
</tr>
<tr>
<td>Function blocks</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------</td>
</tr>
<tr>
<td>FB_READ_006_SNVT_btu_mega[37]</td>
<td>Thermal energy (megaBtu)</td>
</tr>
<tr>
<td>FB_READ_007_SNVT_char_ascii[38]</td>
<td>ASCII character (8-bit ASCII character)</td>
</tr>
<tr>
<td>FB_READ_008_SNVT_count[39]</td>
<td>Absolute number (items)</td>
</tr>
<tr>
<td>FB_READ_009_SNVT_count_inc[40]</td>
<td>Increment counter (items(delta))</td>
</tr>
<tr>
<td>FB_READ_011_SNVT_date_day[41]</td>
<td>Day of the week</td>
</tr>
<tr>
<td>FB_READ_013_SNVT_elec_kwh[42]</td>
<td>Electric energy (kWh)</td>
</tr>
<tr>
<td>FB_READ_014_SNVT_elec_whr[42]</td>
<td>Electric energy (Wh)</td>
</tr>
<tr>
<td>FB_READ_015_SNVT_flow[43]</td>
<td>Volume flow (liters / second)</td>
</tr>
<tr>
<td>FB_READ_016_SNVT_flow_mil[44]</td>
<td>Volume flow (milliliters / second)</td>
</tr>
<tr>
<td>FB_READ_017_SNVT_length[45]</td>
<td>Length (meter)</td>
</tr>
<tr>
<td>FB_READ_018_SNVT_length_kilo[46]</td>
<td>Length (kilometer)</td>
</tr>
<tr>
<td>FB_READ_019_SNVT_length_micr[47]</td>
<td>Length (micrometer)</td>
</tr>
<tr>
<td>FB_READ_020_SNVT_length_mil[48]</td>
<td>Length (millimeter)</td>
</tr>
<tr>
<td>FB_READ_021_SNVT_lev_cont[48]</td>
<td>Continuous change in value (% from maximum level)</td>
</tr>
<tr>
<td>FB_READ_023_SNVT_mass[49]</td>
<td>Mass (gram)</td>
</tr>
<tr>
<td>FB_READ_024_SNVT_mass_kilo[50]</td>
<td>Mass (kilogram)</td>
</tr>
<tr>
<td>FB_READ_025_SNVT_mass_mega[51]</td>
<td>Mass (metric ton)</td>
</tr>
<tr>
<td>FB_READ_026_SNVT_mass_mil[52]</td>
<td>Mass (milligram)</td>
</tr>
<tr>
<td>FB_READ_027_SNVT_power[53]</td>
<td>Power (watt)</td>
</tr>
<tr>
<td>FB_READ_028_SNVT_power_kilo[54]</td>
<td>Power (kilowatt)</td>
</tr>
<tr>
<td>FB_READ_029_SNVT_ppm[54]</td>
<td>Concentration (ppm)</td>
</tr>
<tr>
<td>FB_READ_030_SNVT_press[55]</td>
<td>Pressure (overpressure) (kilopascal)</td>
</tr>
<tr>
<td>FB_READ_031_SNVT_res[56]</td>
<td>Electric resistance (ohm)</td>
</tr>
<tr>
<td>FB_READ_032_SNVT_res_kilo[57]</td>
<td>Electric resistance (kiloohm)</td>
</tr>
<tr>
<td>FB_READ_033_SNVT_sound_db[58]</td>
<td>Sound pressure level (dB)</td>
</tr>
<tr>
<td>FB_READ_034_SNVT_speed[59]</td>
<td>Linear velocity (meters / second)</td>
</tr>
<tr>
<td>FB_READ_035_SNVT_speed_mil[60]</td>
<td>Linear velocity (millimeters / second)</td>
</tr>
<tr>
<td>FB_READ_036_SNVT_str_asc[60]</td>
<td>String</td>
</tr>
<tr>
<td>FB_READ_037_SNVT_str_int[61]</td>
<td>Wide Character String</td>
</tr>
<tr>
<td>FB_READ_038_SNVT_telcom[62]</td>
<td>Telephone status (telephone status name)</td>
</tr>
<tr>
<td>FB_READ_039_SNVT_temp[63]</td>
<td>Temperature (°C)</td>
</tr>
<tr>
<td>FB_READ_041_SNVT_vol[64]</td>
<td>Volume (liter)</td>
</tr>
<tr>
<td>FB_READ_042_SNVT_vol_kilo[65]</td>
<td>Volume (kiloliter)</td>
</tr>
<tr>
<td>FB_READ_043_SNVT_vol_mil[66]</td>
<td>Volume (milliliter)</td>
</tr>
<tr>
<td>FB_READ_044_SNVT_volt[66]</td>
<td>Electrical voltage (volt)</td>
</tr>
<tr>
<td>FB_READ_045_SNVT_volt_dbmv[67]</td>
<td>Electrical voltage (dB microvolt)</td>
</tr>
<tr>
<td>FB_READ_046_SNVT_volt_kilo[68]</td>
<td>Electrical voltage (kilovolt)</td>
</tr>
<tr>
<td>FB_READ_047_SNVT_volt_mil[69]</td>
<td>Electrical voltage (millivolt)</td>
</tr>
<tr>
<td>FB_READ_048_SNVT_amp_f[70]</td>
<td>Electric current (ampere)</td>
</tr>
<tr>
<td>FB_READ_049_SNVT_angle_f[71]</td>
<td>Angular distance (radian)</td>
</tr>
<tr>
<td>FB_READ_050_SNVT_angle_vel_f[72]</td>
<td>Angular velocity (radian / second)</td>
</tr>
<tr>
<td>FB_READ_051_SNVT_count_f[72]</td>
<td>Absolute number (items)</td>
</tr>
<tr>
<td>FB_READ_052_SNVT_count_inc_f[73]</td>
<td>Increment counter (items(delta))</td>
</tr>
<tr>
<td>Function blocks</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------</td>
</tr>
<tr>
<td>FB_READ_053_SNVT_flow_f</td>
<td>Volume flow (liters / second)</td>
</tr>
<tr>
<td>FB_READ_054_SNVT_length_f</td>
<td>Length (meter)</td>
</tr>
<tr>
<td>FB_READ_055_SNVT_lev_cont_f</td>
<td>Continuous change in value (% from maximum level)</td>
</tr>
<tr>
<td>FB_READ_056_SNVT_mass_f</td>
<td>Mass (gram)</td>
</tr>
<tr>
<td>FB_READ_057_SNVT_power_f</td>
<td>Power (watt)</td>
</tr>
<tr>
<td>FB_READ_058_SNVT_ppm_f</td>
<td>Concentration (ppm)</td>
</tr>
<tr>
<td>FB_READ_059_SNVT_press_f</td>
<td>Pressure (overpressure) (pascal)</td>
</tr>
<tr>
<td>FB_READ_060_SNVT_res_f</td>
<td>Electric resistance (ohm)</td>
</tr>
<tr>
<td>FB_READ_061_SNVT_sound_db_f</td>
<td>Sound pressure level (dBspl)</td>
</tr>
<tr>
<td>FB_READ_062_SNVT_speed_f</td>
<td>Linear velocity (meters / second)</td>
</tr>
<tr>
<td>FB_READ_063_SNVT_temp_f</td>
<td>Temperature (°C)</td>
</tr>
<tr>
<td>FB_READ_064_SNVT_time_f</td>
<td>Elapsed time (seconds)</td>
</tr>
<tr>
<td>FB_READ_065_SNVT_vol_f</td>
<td>Volume (liter)</td>
</tr>
<tr>
<td>FB_READ_066_SNVT_volt_f</td>
<td>Electrical voltage (volt)</td>
</tr>
<tr>
<td>FB_READ_067_SNVT_btu_f</td>
<td>Thermal energy (Btu)</td>
</tr>
<tr>
<td>FB_READ_068_SNVT_elec_whr_f</td>
<td>Electric energy (Watt / hour)</td>
</tr>
<tr>
<td>FB_READ_069_SNVT_config_src</td>
<td>Configuration properties</td>
</tr>
<tr>
<td>FB_READ_070_SNVT_color</td>
<td>Color according to CIE standard</td>
</tr>
<tr>
<td>FB_READ_071_SNVT_grammage</td>
<td>American measure for paper weight and density</td>
</tr>
<tr>
<td>FB_READ_072_SNVT_grammage_f</td>
<td>American measure for paper weight and density</td>
</tr>
<tr>
<td>FB_READ_073_SNVT_file_req</td>
<td>File request</td>
</tr>
<tr>
<td>FB_READ_074_SNVT_file_status</td>
<td>File information</td>
</tr>
<tr>
<td>FB_READ_075_SNVT_freq_f</td>
<td>Frequency (hertz)</td>
</tr>
<tr>
<td>FB_READ_076_SNVT_freq_hz</td>
<td>Frequency (hertz)</td>
</tr>
<tr>
<td>FB_READ_077_SNVT_freq_kilohz</td>
<td>Frequency (kilohertz)</td>
</tr>
<tr>
<td>FB_READ_078_SNVT_freq_milhz</td>
<td>Frequency (millihertz)</td>
</tr>
<tr>
<td>FB_READ_079_SNVT_lux</td>
<td>Illuminance (lux)</td>
</tr>
<tr>
<td>FB_READ_081_SNVT_lev_percent</td>
<td>Percentage value</td>
</tr>
<tr>
<td>FB_READ_082_SNVT_multiplier</td>
<td>Multiplier</td>
</tr>
<tr>
<td>FB_READ_083_SNVT_state</td>
<td>Status information</td>
</tr>
<tr>
<td>FB_READ_084_SNVT_time_stamp</td>
<td>Time stamp</td>
</tr>
<tr>
<td>FB_READ_085_SNVT_zerospan</td>
<td>Zero point and proportionality factor</td>
</tr>
<tr>
<td>FB_READ_086_SNVT_magcard</td>
<td>ISO 7811 (40 hexadecimal numbers)</td>
</tr>
<tr>
<td>FB_READ_087_SNVT_elapsed_tm</td>
<td>Elapsed time</td>
</tr>
<tr>
<td>FB_READ_088_SNVT_alarm</td>
<td>Alarm status</td>
</tr>
<tr>
<td>FB_READ_089_SNVT_currency</td>
<td>Currency</td>
</tr>
<tr>
<td>FB_READ_090_SNVT_file_pos</td>
<td>File position</td>
</tr>
<tr>
<td>FB_READ_091_SNVT_muldiv</td>
<td>Gain factor/attenuation factor</td>
</tr>
<tr>
<td>FB_READ_092_SNVT_obj_request</td>
<td>Function selection</td>
</tr>
<tr>
<td>FB_READ_093_SNVT_obj_status</td>
<td>Object status</td>
</tr>
<tr>
<td>FB_READ_094_SNVT_preset</td>
<td>Preliminary setting</td>
</tr>
<tr>
<td>FB_READ_095_SNVT_switch</td>
<td>Switch</td>
</tr>
<tr>
<td>FB_READ_096_SNVT_trans_table</td>
<td>Conversion table</td>
</tr>
<tr>
<td>Function blocks</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>FB_READ_097_SNVT_override</td>
<td>Override mode</td>
</tr>
<tr>
<td>FB_READ_098_SNVT_pwr_fact</td>
<td>Power factor</td>
</tr>
<tr>
<td>FB_READ_099_SNVT_pwr_fact_f</td>
<td>Power factor</td>
</tr>
<tr>
<td>FB_READ_100_SNVT_density</td>
<td>Density (kg/m³)</td>
</tr>
<tr>
<td>FB_READ_101_SNVT_density_f</td>
<td>Density (kg/m³)</td>
</tr>
<tr>
<td>FB_READ_102_SNVT_rpm</td>
<td>Angular velocity (revolutions/minute (RPM))</td>
</tr>
<tr>
<td>FB_READ_103_SNVT_hvac_emerg</td>
<td>HVAC emergency mode</td>
</tr>
<tr>
<td>FB_READ_104_SNVT_angle_deg</td>
<td>Angle indication</td>
</tr>
<tr>
<td>FB_READ_105_SNVT_temp_p</td>
<td>Temperature (°C)</td>
</tr>
<tr>
<td>FB_READ_106_SNVT_temp_setpt</td>
<td>Temperature (6 temperature values)</td>
</tr>
<tr>
<td>FB_READ_107_SNVT_time_sec</td>
<td>Elapsed time (second)</td>
</tr>
<tr>
<td>FB_READ_108_SNVT_hvac_mode</td>
<td>HVAC operating mode</td>
</tr>
<tr>
<td>FB_READ_109_SNVT_occupancy</td>
<td>Occupancy signal (states)</td>
</tr>
<tr>
<td>FB_READ_110_SNVT_area</td>
<td>Area (square meter)</td>
</tr>
<tr>
<td>FB_READ_111_SNVT_hvac_overid</td>
<td>HVAC override mode</td>
</tr>
<tr>
<td>FB_READ_112_SNVT_hvac_status</td>
<td>HVAC status</td>
</tr>
<tr>
<td>FB_READ_113_SNVT_press_p</td>
<td>Pressure (overpressure) (pascal)</td>
</tr>
<tr>
<td>FB_READ_114_SNVT_address</td>
<td>Neuron address</td>
</tr>
<tr>
<td>FB_READ_115_SNVT_scene</td>
<td>Scenes</td>
</tr>
<tr>
<td>FB_READ_116_SNVT_scene_cfg</td>
<td>Scene setting</td>
</tr>
<tr>
<td>FB_READ_117_SNVT_setting</td>
<td>Settings</td>
</tr>
<tr>
<td>FB_READ_118_SNVT_evap_state</td>
<td>Evaporator status</td>
</tr>
<tr>
<td>FB_READ_119_SNVT_therm_mode</td>
<td>Thermostat mode</td>
</tr>
<tr>
<td>FB_READ_120_SNVT_defr_mode</td>
<td>Defrost mode</td>
</tr>
<tr>
<td>FB_READ_121_SNVT_defr_term</td>
<td>Completion of the defrost cycle</td>
</tr>
<tr>
<td>FB_READ_122_SNVT_defr_state</td>
<td>Defrost status (enumeration)</td>
</tr>
<tr>
<td>FB_READ_123_SNVT_time_min</td>
<td>Elapsed time (minutes)</td>
</tr>
<tr>
<td>FB_READ_124_SNVT_time_hour</td>
<td>Elapsed time (hour)</td>
</tr>
<tr>
<td>FB_READ_125_SNVT_ph</td>
<td>Acidity (pH)</td>
</tr>
<tr>
<td>FB_READ_126_SNVT_ph_f</td>
<td>Acidity (pH)</td>
</tr>
<tr>
<td>FB_READ_127_SNVT_chlr_status</td>
<td>Refrigeration unit status</td>
</tr>
<tr>
<td>FB_READ_128_SNVT_tod_event</td>
<td>Presence time</td>
</tr>
<tr>
<td>FB_READ_129_SNVT_smo_obscur</td>
<td>Darkening due to smoke</td>
</tr>
<tr>
<td>FB_READ_130_SNVT_fire_test</td>
<td>Request fire alarm system test</td>
</tr>
<tr>
<td>FB_READ_131_SNVT_temp_ror</td>
<td>Value of the temperature change/increase</td>
</tr>
<tr>
<td>FB_READ_132_SNVT_fire_init</td>
<td>Fire detector type</td>
</tr>
<tr>
<td>FB_READ_133_SNVT_fire_indcte</td>
<td>Fire alarm display</td>
</tr>
<tr>
<td>FB_READ_134_SNVT_time_zone</td>
<td>Time zone description</td>
</tr>
<tr>
<td>FB_READ_135_SNVT_earth_pos</td>
<td>Position on Earth</td>
</tr>
<tr>
<td>FB_READ_136_SNVT_reg_val</td>
<td>Register value</td>
</tr>
<tr>
<td>FB_READ_137_SNVT_reg_val_ts</td>
<td>Register value</td>
</tr>
<tr>
<td>FB_READ_138_SNVT_volt_ac</td>
<td>Alternating voltage (Volt AC)</td>
</tr>
<tr>
<td>FB_READ_139_SNVT_amp_ac</td>
<td>Alternating current</td>
</tr>
<tr>
<td>Function blocks</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------</td>
</tr>
<tr>
<td>FB_READ_143_SNVT_turbidity</td>
<td>Turbidity</td>
</tr>
<tr>
<td>FB_READ_144_SNVT_turbidity_f</td>
<td>Turbidity</td>
</tr>
<tr>
<td>FB_READ_145_SNVT_hvac_type</td>
<td>HVAC plant type</td>
</tr>
<tr>
<td>FB_READ_146_SNVT_elec_kwh</td>
<td>Electric energy (kW / hour)</td>
</tr>
<tr>
<td>FB_READ_147_SNVT_temp_diff_p</td>
<td>Temperature difference (°C)</td>
</tr>
<tr>
<td>FB_READ_148_SNVT_ctrl_req</td>
<td>Control request</td>
</tr>
<tr>
<td>FB_READ_149_SNVT_ctrl_resp</td>
<td>Control response</td>
</tr>
<tr>
<td>FB_READ_150_SNVT_ptz</td>
<td>Camera PTZ (SNZ)</td>
</tr>
<tr>
<td>FB_READ_151_SNVT_privacyzone</td>
<td>Private zone</td>
</tr>
<tr>
<td>FB_READ_152_SNVT_pos_ctrl</td>
<td>Position setting for cameras in networks</td>
</tr>
<tr>
<td>FB_READ_153_SNVT_enthalpy</td>
<td>Enthalpy (kilojoules/kg)</td>
</tr>
<tr>
<td>FB_READ_154_SNVT_gfci_status</td>
<td>Residual current circuit breaker status</td>
</tr>
<tr>
<td>FB_READ_155_SNVT_motor_state</td>
<td>Motor status (motor status designation)</td>
</tr>
<tr>
<td>FB_READ_156_SNVT_pumpset_mn</td>
<td>Pump group</td>
</tr>
<tr>
<td>FB_READ_157_SNVT_ex_control</td>
<td>Exclusive control</td>
</tr>
<tr>
<td>FB_READ_158_SNVT_pumpset_sn</td>
<td>Pump group sensor</td>
</tr>
<tr>
<td>FB_READ_159_SNVT_pump_sensor</td>
<td>Pump sensor</td>
</tr>
<tr>
<td>FB_READ_160_SNVT_abs_humid</td>
<td>Absolute humidity</td>
</tr>
<tr>
<td>FB_READ_161_SNVT_flow_p</td>
<td>Volume flow (cubic meters / hour)</td>
</tr>
<tr>
<td>FB_READ_162_SNVT_dev_c_mode</td>
<td>Device operating mode</td>
</tr>
<tr>
<td>FB_READ_163_SNVT_valve_mode</td>
<td>Valve state</td>
</tr>
<tr>
<td>FB_READ_164_SNVT_alarm_2</td>
<td>Alarm status 2</td>
</tr>
<tr>
<td>FB_READ_165_SNVT_state_64</td>
<td>Status information (64 individual bit values)</td>
</tr>
<tr>
<td>FB_READ_166_SNVT_nv_type</td>
<td>Network variable type</td>
</tr>
<tr>
<td>FB_READ_168_SNVT_ent_opmode</td>
<td>Operating mode for access objects</td>
</tr>
<tr>
<td>FB_READ_169_SNVT_ent_state</td>
<td>State for access objects</td>
</tr>
<tr>
<td>FB_READ_170_SNVT_ent_status</td>
<td>Status of access objects</td>
</tr>
<tr>
<td>FB_READ_171_SNVT_flow_dir</td>
<td>Flow direction</td>
</tr>
<tr>
<td>FB_READ_172_SNVT_hvac_satsts</td>
<td>HVAC saturation status</td>
</tr>
<tr>
<td>FB_READ_173_SNVT_dev_status</td>
<td>Device status</td>
</tr>
<tr>
<td>FB_READ_174_SNVT_dev_fault</td>
<td>Error state</td>
</tr>
<tr>
<td>FB_READ_175_SNVT_dev_maint</td>
<td>Device maintenance status</td>
</tr>
<tr>
<td>FB_READ_176_SNVT_date_event</td>
<td>Event status</td>
</tr>
<tr>
<td>FB_READ_177_SNVT_sched_val</td>
<td>Calendar value</td>
</tr>
<tr>
<td>FB_READ_180_SNVT_sblnd_state</td>
<td>Blind status</td>
</tr>
<tr>
<td>FB_READ_181_SNVT_rac_ctl</td>
<td>Sound function control</td>
</tr>
<tr>
<td>FB_READ_182_SNVT_rac_req</td>
<td>Sound function request</td>
</tr>
<tr>
<td>FB_READ_183_SNVT_count_32</td>
<td>Absolute counter</td>
</tr>
<tr>
<td>FB_READ_184_SNVT_clothes_w_c</td>
<td>Washing machine / commands</td>
</tr>
<tr>
<td>FB_READ_185_SNVT_clothes_w_m</td>
<td>Washing machine / management status</td>
</tr>
<tr>
<td>FB_READ_186_SNVT_clothes_w_s</td>
<td>Washing machine / status</td>
</tr>
<tr>
<td>FB_READ_187_SNVT_clothes_w_a</td>
<td>Washing machine / alarm messages</td>
</tr>
<tr>
<td>FB_READ_188_SNVT_multiplier_s</td>
<td>Multiplier</td>
</tr>
</tbody>
</table>
### Function blocks

<table>
<thead>
<tr>
<th>Function block</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FB_READ_189_SNVT_switch_2</td>
<td>Switch for setting scenes and settings.</td>
</tr>
<tr>
<td>FB_READ_190_SNVT_color_2</td>
<td>Color</td>
</tr>
<tr>
<td>FB_READ_191_SNVT_log_status</td>
<td>Log status</td>
</tr>
<tr>
<td>FB_READ_192_SNVT_time_stamp_p</td>
<td>Precise time stamp</td>
</tr>
<tr>
<td>FB_READ_193_SNVT_log_fx_request</td>
<td>Log file transfer request</td>
</tr>
<tr>
<td>FB_READ_194_SNVT_log_fx_status</td>
<td>Log file transfer status</td>
</tr>
<tr>
<td>FB_READ_195_SNVT_log_request</td>
<td>Log status request</td>
</tr>
<tr>
<td>FB_READ_196_SNVT_enthalpy_d</td>
<td>Enthalpy difference (kJ/kg)</td>
</tr>
<tr>
<td>FB_READ_197_SNVT_amp_ac_mil</td>
<td>Electric current (milliampere)</td>
</tr>
<tr>
<td>FB_READ_198_SNVT_time_hour_p</td>
<td>Time in hours</td>
</tr>
<tr>
<td>FB_READ_199_SNVT_lamp_status</td>
<td>Lamp status</td>
</tr>
<tr>
<td>FB_READ_200_SNVT_environment</td>
<td>Environment</td>
</tr>
<tr>
<td>FB_READ_201_SNVT_geo_loc</td>
<td>Geographical location</td>
</tr>
</tbody>
</table>

### 4.1.2.1 FB_READ_001_SNVT_amp

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_amp.

**SNVT number:** 001.

**Description:** Electric current (ampere).

#### VAR_INPUT

- wNVIndex : WORD;
- bDisabled : BOOL := FALSE;

**wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled:** TRUE = deselection of the block.

#### VAR_OUTPUT

- rValue : REAL;
- bNewData : BOOL;
- bError : BOOL;
- eError : E_LON_ERROR;

**rValue:** Min: -3276.8 / Max: 3276.7.

**bNewData:** Becomes TRUE for 1 cycle when the block has received data.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

**eError:** This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.
VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() with the send/receive function (see ST_LON_Communication).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.2 FB_READ_002_SNVT_amp_mil

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_amp_mil.
SNVT number: 002.
Description: Electric current (milliampere).

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

rValue: Min: -3276.8 / Max: 3276.7.
bNewData: Becomes TRUE for 1 cycle when the block has received data.
bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.
eError: This output outputs an error code in the event of an error (see E_LON_ERROR). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() with the send/receive function (see ST_LON_Communication).
Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.3   **FB_READ_003_SNVT_angle**

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_angle.

**SNVT number:** 003.

**Description:** Angular distance (radian).

**VAR_INPUT**

| wNVIndex  : WORD; | bDisabled : BOOL := FALSE; |

- **wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

- **bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

| rValue    : REAL; | bNewData : BOOL; | bError   : BOOL; | eError   : E_LON_ERROR; |

- **rValue:** Min: 0 / Max: 65.535.

- **bNewData:** Becomes TRUE for 1 cycle when the block has received data.

- **bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable **eError**.

- **eError:** This output outputs an error code in the event of an error (see **E_LON_ERROR** [477]). bError goes TRUE at the same time.

**VAR_IN_OUT**

| stLON_Com : ST_LON_Communication; |

- **stLON_Com:** This structure is used to link **FB_LON_KL6401() [27]** with the send/receive function (see **ST_LON_Communication [587]**).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.1.2.4 FB_READ_004_SNVT_angle_vel

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_angle_vel.

**SNVT number:** 004.

**Description:** Angular velocity (radian / second).

**VAR_INPUT**

```plaintext
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

- **wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

```plaintext
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

- **rValue:** Min: -3276.8 / Max: 3276.7.
- **bNewData:** Becomes TRUE for 1 cycle when the block has received data.
- **bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable eError.
- **eError:** This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

**VAR_IN_OUT**

```plaintext
stLON_Com : ST_LON_Communication;
```

- **stLON_Com:** This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.5 FB_READ_005_SNVT_btu_kilo

This function block receives the following LON input variable (nvi):

**SNVT Name:** FB_READ_005_SNVT_btu_kilo.

**SNVT number:** 005.

**Description:** BTU (British Thermal Units) of kilowatts.

**VAR_INPUT**

```plaintext
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

- **wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

```plaintext
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

- **rValue:** Min: -3276.8 / Max: 3276.7.
- **bNewData:** Becomes TRUE for 1 cycle when the block has received data.
- **bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable eError.
- **eError:** This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.
This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_btu_kilo.

**SNVT number:** 005.

**Description:** Thermal energy (kiloBtu).

**VAR_INPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>WORD</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL := FALSE;</td>
</tr>
</tbody>
</table>

**wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>uiValue</td>
<td>UINT</td>
</tr>
<tr>
<td>bNewData</td>
<td>BOOL</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
</tr>
</tbody>
</table>

**uiValue:** Min: 0 / Max: 65535.

**bNewData:** Becomes TRUE for 1 cycle when the block has received data.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

**eError:** This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

**VAR_IN_OUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>stLON_Com</td>
<td>ST_LON_Communication;</td>
</tr>
</tbody>
</table>

**stLON_Com:** This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.2.6 FB_READ_006_SNVT_btu_mega

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_btu_mega.

**SNVT number:** 006.

**Description:** ASCII character (8-bit ASCII character).
VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

uiValue : UINT;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

uiValue: Min: 0 / Max: 65535.

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com: This structure is used to link FB_LON_KL6401() with the send/receive function (see ST LON Communication).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.7 FB_READ_007_SNVT_char_ascii

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_char_ascii.

SNVT number: 007.

Description: ASCII character (8-bit ASCII character).

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.
VAR_OUTPUT

sValue : STRING(1);
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

sValue: STRING(1).

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.8 FB_READ_008_SNVT_count

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_count.

SNVT number: 008.

Description: Counter (only positive values).

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

uiValue : UINT;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

uiValue: Min: 0 / Max: 65535.

bNewData: Becomes TRUE for 1 cycle when the block has received data.
bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

### VAR_IN_OUT

```plaintext
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.2.9 FB_READ_009_SNVT_count_inc

![FB_READ_009_SNVT_count_inc diagram]

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_count_inc.

**SNVT number:** 009.

**Description:** Counter (negative and positive values).

### VAR_INPUT

```plaintext
wNVIndex  : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled:** TRUE = deselection of the block.

### VAR_OUTPUT

```plaintext
iValue   : INT;
bNewData : BOOL;
bError   : BOOL;
eError   : E_LON_ERROR;
```

**iValue:** Min: -32768 / Max: 32767.

**bNewData:** Becomes TRUE for 1 cycle when the block has received data.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

**eError:** This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

### VAR_IN_OUT

```plaintext
stLON_Com : ST_LON_Communication;
```
stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.10  FB_READ_011_SNVT_date_day

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_date_day.

**SNVT number:** 011.

**Description:** Day of the week.

**VAR_INPUT**

| wNVIndex : WORD; |
| bDisabled : BOOL := FALSE; |

**wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

| eValue : E_LON_days_of_week_t; |
| bNewData : BOOL; |
| bError : BOOL; |
| eError : E_LON_ERROR; |

**eValue:** Enum to be received (see E_LON_days_of_week_t [513]).

**bNewData:** Becomes TRUE for 1 cycle when the block has received data.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

**eError:** This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

**VAR_IN_OUT**

| stLON_Com : ST_LON_Communication; |

**stLON_Com:** This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.1.2.11 FB_READ_013_SNVT_elec_kwh

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_elec_kwh.

**SNVT number:** 013.

**Description:** Electric energy (kW/h).

**VAR_INPUT**

wNVIndex : WORD;

bDisabled : BOOL := FALSE;

**wNVIndex**: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled**: TRUE = deselection of the block.

**VAR_OUTPUT**

uiValue : UINT;

bNewData : BOOL;

bError : BOOL;

eError : E_LON_ERROR;

**uiValue**: Min: 0 / Max: 65535.

**bNewData**: Becomes TRUE for 1 cycle when the block has received data.

**bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

**eError**: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

**VAR_IN_OUT**

stLON_Com : ST_LON_Communication;

**stLON_Com**: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.12 FB_READ_014_SNVT_elec_whr
This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_elec_whr.

**SNVT number:** 014.

**Description:** Electric energy (W/h).

### VAR_INPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>WORD</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL := FALSE</td>
</tr>
</tbody>
</table>

**wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled:** TRUE = deselection of the block.

### VAR_OUTPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>rValue</td>
<td>REAL</td>
</tr>
<tr>
<td>bNewData</td>
<td>BOOL</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
</tr>
</tbody>
</table>

**rValue:** Min: 0 / Max: 6553.5.

**bNewData:** Becomes TRUE for 1 cycle when the block has received data.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

**eError:** This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

### VAR_IN_OUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>stLON_Com</td>
<td>ST_LON_Communication</td>
</tr>
</tbody>
</table>

**stLON_Com:** This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

#### 4.1.2.13 FB_READ_015_SNVT_flow

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_flow.

**SNVT number:** 015.

**Description:** Volume flow (liters / second).
VAR_INPUT

wNVIndex : WORD;

bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

uiValue : UINT;

bNewData : BOOL;

eError : E_LON_ERROR;

uiValue: Min: 0 / Max: 65535.

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.14 FB_READ_016_SNVT_flow_mil

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_flow_mil.

**SNVT number:** 016.

**Description:** Volume flow (milliliters / second).

VAR_INPUT

wNVIndex : WORD;

bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.
**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>uiValue</td>
<td>UINT</td>
</tr>
<tr>
<td>bNewData</td>
<td>BOOL</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
</tr>
</tbody>
</table>

- **uiValue**: Min: 0 / Max: 65535.
- **bNewData**: Becomes TRUE for 1 cycle when the block has received data.
- **bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.
- **eError**: This output outputs an error code in the event of an error (see *E_LON_ERROR [477]*). *bError* goes TRUE at the same time.

**VAR_IN_OUT**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>stLON_Com</td>
<td>ST_LON_Communication;</td>
</tr>
</tbody>
</table>

- **stLON_Com**: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.15  FB_READ_017_SNVT_length

**VAR_INPUT**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>WORD</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL := FALSE;</td>
</tr>
</tbody>
</table>

- **wNVIndex**: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **bDisabled**: TRUE = deselection of the block.

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>rValue</td>
<td>REAL</td>
</tr>
<tr>
<td>bNewData</td>
<td>BOOL</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
</tr>
</tbody>
</table>

- **rValue**: Min: 0 / Max: 6553.5.
- **bNewData**: Becomes TRUE for 1 cycle when the block has received data.
bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.16 FB_READ_018_SNVT_length_kilo

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_length_kilo.

SNVT number: 018.

Description: Length (kilometer).

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

rValue: Min: 0 / Max: 6553.5.

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;
**Programming**

**stLON_Com**: This structure is used to link `FB_LON_KL6401()` with the send/receive function (see `ST_LON_Communication`).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.2.17 FB_READ_019_SNVT_length_micr

This function block receives the following LON input variable (nvi):

**SNVT Name**: SNVT_length_micr.

**SNVT number**: 019.

**Description**: Length (micrometer).

**VAR_INPUT**

| wNVIndex : WORD;    | rValue     |
| bDisabled : BOOL := FALSE; | bNewData  |

- **wNVIndex**: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **bDisabled**: TRUE = deselection of the block.

**VAR_OUTPUT**

| rValue : REAL;    | bNewData  |
| bError : BOOL;    | eError : E_LON_ERROR; |

- **rValue**: Min: 0 / Max: 6553.5.
- **bNewData**: Becomes TRUE for 1 cycle when the block has received data.
- **bError**: This becomes TRUE as soon as an error occurs. This error is described via the variable `eError`.
- **eError**: This output outputs an error code in the event of an error (see `E_LON_ERROR`). `bError` goes TRUE at the same time.

**VAR_IN_OUT**

| stLON_Com : ST_LON_Communication; |

- **stLON_Com**: This structure is used to link `FB_LON_KL6401()` with the send/receive function (see `ST_LON_Communication`).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.1.2.18  FB_READ_020_SNVT_length_mil

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_length_mil.

**SNVT number:** 020.

**Description:** Length (millimeter).

**VAR_INPUT**

- `wNVIndex` : WORD;
- `bDisabled` : BOOL := FALSE;

`wNVIndex`: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

`bDisabled`: TRUE = deselection of the block.

**VAR_OUTPUT**

- `rValue` : REAL;
- `bNewData` : BOOL;
- `bError` : BOOL;
- `eError` : E_LON_ERROR;

`rValue`: Min: 0 / Max: 6553.5.

`bNewData`: Becomes TRUE for 1 cycle when the block has received data.

`bError`: This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.

`eError`: This output outputs an error code in the event of an error (see `E_LON_ERROR[477]`). `bError` goes TRUE at the same time.

**VAR_IN_OUT**

- `stLON_Com` : ST_LON_Communication;

`stLON_Com`: This structure is used to link `FB_LON_KL6401()` with the send/receive function (see `ST_LON_Communication[587]`).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.19  FB_READ_021_SNVT_lev_cont

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_lev_cont.

**SNVT number:** 021.

**Description:** Level.

**VAR_INPUT**

- `wNVIndex` : WORD;
- `bDisabled` : BOOL := FALSE;

**VAR_OUTPUT**

- `rValue` : REAL;
- `bNewData` : BOOL;
- `bError` : BOOL;
- `eError` : E_LON_ERROR;

**VAR_IN_OUT**

- `stLON_Com` : ST_LON_Communication;
This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_lev_cont.

**SNVT number:** 021.

**Description:** Continuous change in value (% from maximum level).

---

**VAR_INPUT**

<table>
<thead>
<tr>
<th>wNVIndex</th>
<th>WORD</th>
</tr>
</thead>
<tbody>
<tr>
<td>bDisabled</td>
<td>BOOL := FALSE</td>
</tr>
</tbody>
</table>

**wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled:** TRUE = deselection of the block.

---

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>rValue</th>
<th>REAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>bNewData</td>
<td>BOOL</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
</tr>
</tbody>
</table>

**rValue:** Min: 0 / Max: 100.

**bNewData:** Becomes TRUE for 1 cycle when the block has received data.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

**eError:** This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

---

**VAR_IN_OUT**

<table>
<thead>
<tr>
<th>stLON_Com</th>
<th>ST_LON_Communication</th>
</tr>
</thead>
</table>

**stLON_Com:** This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

---

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

---

**4.1.2.20 FB_READ_023_SNVT_mass**

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_mass.

**SNVT number:** 023.

**Description:** Mass (gram).
VAR_INPUT

wNVIndex  : WORD;
bDisabled  : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

rValue   : REAL;
bNewData : BOOL;
bError   : BOOL;
eError   : E_LON_ERROR;

rValue: Min: 0 / Max: 6553.5.

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com  : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.21   FB_READ_024_SNVT_mass_kilo

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_mass_kilo.
SNVT number: 024.
Description: Mass (kilogram).

VAR_INPUT

wNVIndex  : WORD;
bDisabled  : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.
VAR_OUTPUT

rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

rValue: Min: 0 / Max: 6553.5.

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.22 FB_READ_025_SNVT_mass_mega

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_mass_mega.

SNVT number: 025.

Description: Mass (metric ton).

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

rValue: Min: 0 / Max: 6553.5.

bNewData: Becomes TRUE for 1 cycle when the block has received data.
bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

**VAR_IN_OUT**

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.23 **FB_READ_026_SNVT_mass_mil**

This function block receives the following LON input variable (nvi):

- **SNVT Name:** SNVT_mass_mil.
- **SNVT number:** 026.
- **Description:** Mass (milligram).

**VAR_INPUT**

```
  wNVIndex : WORD;
  bDisabled : BOOL := FALSE;
```

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

**VAR_OUTPUT**

```
  rValue   : REAL;
  bNewData : BOOL;
  bError   : BOOL;
  eError   : E_LON_ERROR;
```

rValue: Min: 0 / Max: 6553.5.

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

**VAR_IN_OUT**

```
stLON_Com : ST_LON_Communication;
```
**stLON_Com**: This structure is used to link [FB_LON_KL6401()][27] with the send/receive function (see [ST_LON_Communication][587]).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.2.24 **FB_READ_027_SNVT_power**

This function block receives the following LON input variable (nvi):

- **SNVT Name**: SNVT_power.
- **SNVT number**: 027.
- **Description**: Power (watt).

#### VAR_INPUT

```plaintext
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

- **wNVIndex**: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **bDisabled**: TRUE = deselection of the block.

#### VAR_OUTPUT

```plaintext
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

- **rValue**: Min: 0 / Max: 6553.5.
- **bNewData**: Becomes TRUE for 1 cycle when the block has received data.
- **bError**: This goes TRUE as soon as an error occurs. This error is described via the variable `eError`.
- **eError**: This output outputs an error code in the event of an error (see [E_LON_ERROR][477]). `bError` goes TRUE at the same time.

#### VAR_IN_OUT

```plaintext
stLON_Com : ST_LON_Communication;
```

- **stLON_Com**: This structure is used to link [FB_LON_KL6401()][27] with the send/receive function (see [ST_LON_Communication][587]).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.1.2.25  **FB_READ_028_SNVT_power_kilo**

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_power_kilo.

**SNVT number:** 028.

**Description:** Power (kilowatt).

**VAR_INPUT**

| wNVIndex : WORD; |
| bDisabled : BOOL := FALSE; |

**wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

| rValue : REAL; |
| bNewData : BOOL; |
| bError : BOOL; |
| eError : E_LON_ERROR; |

**rValue:** Min: 0 / Max: 6553.5.

**bNewData:** Becomes TRUE for 1 cycle when the block has received data.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable **eError**.

**eError:** This output outputs an error code in the event of an error (see **E_LON_ERROR**). **bError** goes TRUE at the same time.

**VAR_IN_OUT**

| stLON_Com : ST_LON_Communication; |

**stLON_Com:** This structure is used to link **FB_LON_KL6401()** with the send/receive function (see **ST_LON_Communication**).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.26  **FB_READ_029_SNVT_ppm**

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_ppm.

**SNVT number:** 028.

**Description:** Power (kilowatt).

**VAR_INPUT**

| wNVIndex : WORD; |
| bDisabled : BOOL := FALSE; |

**wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

| rValue : REAL; |
| bNewData : BOOL; |
| bError : BOOL; |
| eError : E_LON_ERROR; |

**rValue:** Min: 0 / Max: 6553.5.

**bNewData:** Becomes TRUE for 1 cycle when the block has received data.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable **eError**.

**eError:** This output outputs an error code in the event of an error (see **E_LON_ERROR**). **bError** goes TRUE at the same time.

**VAR_IN_OUT**

| stLON_Com : ST_LON_Communication; |

**stLON_Com:** This structure is used to link **FB_LON_KL6401()** with the send/receive function (see **ST_LON_Communication**).
This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_ppm.

**SNVT number:** 029.

**Description:** Concentration (ppm).

### VAR_INPUT

```plaintext
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

- **wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **bDisabled:** TRUE = deselection of the block.

### VAR_OUTPUT

```plaintext
uiValue : UINT;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

- **uiValue:** Min: 0 / Max: 65535.
- **bNewData:** Becomes TRUE for 1 cycle when the block has received data.
- **bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable eError.
- **eError:** This output outputs an error code in the event of an error (see `E_LON_ERROR` [477]). bError goes TRUE at the same time.

### VAR_IN_OUT

```plaintext
stLON_Com : ST_LON_Communication;
```

- **stLON_Com:** This structure is used to link `FB_LON_KL6401()` [27] with the send/receive function (see `ST_LON_Communication` [587]).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.2.27 FB_READ_030_SNVT_press**

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_press.

**SNVT number:** 030.

**Description:** Pressure (overpressure) (kilopascal).
VAR_INPUT
wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

rValue: Min: -3276.8 / Max: 3276.7.

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT
stLON_Com : ST_LON_Communication;

stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.28  FB_READ_031_SNVT_res

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_res.

SNVT number: 031.

Description: Electric resistance (ohm).

VAR_INPUT
wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.
VAR_OUTPUT

rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

rValue: Min: 0 / Max: 6553.5.

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.29  FB_READ_032_SNVT_res_kilo

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_res_kilo.

**SNVT number:** 032.

**Description:** Electric resistance (kiloohm).

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

rValue: Min: 0 / Max: 6553.5.

bNewData: Becomes TRUE for 1 cycle when the block has received data.
bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

**VAR_IN_OUT**

```plaintext
stLON_Com : ST_LON_Communication;
```

**stLON_Com** : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.2.30 FB_READ_033_SNVT_sound_db

This function block receives the following LON input variable (nvi):

**SNVT Name**: SNVT_sound_db.

**SNVT number**: 033.

**Description**: Sound pressure level (dB).

**VAR_INPUT**

```plaintext
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex**: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled**: TRUE = deselection of the block.

**VAR_OUTPUT**

```plaintext
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**rValue**: Min: -327.68 / Max: 327.67.

**bNewData**: Becomes TRUE for 1 cycle when the block has received data.

**bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

**eError**: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

**VAR_IN_OUT**

```plaintext
stLON_Com : ST_LON_Communication;
```
stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.31  FB_READ_034_SNVT_speed

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_speed.

**SNVT number:** 034.

**Description:** Linear velocity (meters / second).

**VAR_INPUT**

<table>
<thead>
<tr>
<th>wNIndex</th>
<th>rValue</th>
</tr>
</thead>
<tbody>
<tr>
<td>bDisabled</td>
<td>bNewData</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>stLON_Com</th>
<th>bError</th>
</tr>
</thead>
<tbody>
<tr>
<td>eError</td>
<td></td>
</tr>
</tbody>
</table>

**VAR_OUT**

| rValue | 0 / Max: 6553.5 |
| bNewData | |
| bError | |
| eError | |

**VAR_IN_OUT**

<table>
<thead>
<tr>
<th>stLON_Com</th>
<th>stLON_Communication</th>
</tr>
</thead>
</table>

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_speed.

**SNVT number:** 034.

**Description:** Linear velocity (meters / second).

**VAR_INPUT**

| wNIndex : WORD |
| bDisabled : BOOL := FALSE |

**VAR_OUTPUT**

| rValue | Min: 0 / Max: 6553.5 |
| bNewData | |
| bError | |
| eError | |

**VAR_IN_OUT**

| stLON_Com : ST_LON_Communication |

This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.1.2.32  FB_READ_035_SNVT_speed_mil

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_speed_mil.

**SNVT number:** 035.

**Description:** Linear velocity (millimeters / second).

**VAR_INPUT**

```plaintext
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

```plaintext
rValue  : REAL;
bNewData : BOOL;
bError   : BOOL;
eError   : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 65.535.

**bNewData:** Becomes TRUE for 1 cycle when the block has received data.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.

**eError:** This output outputs an error code in the event of an error (see `E_LON_ERROR`). bError goes TRUE at the same time.

**VAR_IN_OUT**

```plaintext
stLON_Com : ST_LON_Communication;
```

**stLON_Com:** This structure is used to link `FB_LON_KL6401()` with the send/receive function (see `ST_LON_Communication`).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.33  FB_READ_036_SNVT_str_asc
This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_str_asc.

**SNVT number:** 036.

**Description:** String (max. 30 characters) (ASCII string).

**VAR_INPUT**

<table>
<thead>
<tr>
<th>wNVIndex</th>
<th>WORD</th>
</tr>
</thead>
<tbody>
<tr>
<td>bDisabled</td>
<td>BOOL := FALSE</td>
</tr>
</tbody>
</table>

**wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>sValue</th>
<th>STRING(31)</th>
</tr>
</thead>
<tbody>
<tr>
<td>bNewData</td>
<td>BOOL</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
</tr>
</tbody>
</table>

**sValue:** STRING(31).

**bNewData:** Becomes TRUE for 1 cycle when the block has received data.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

**eError:** This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

**VAR_IN_OUT**

<table>
<thead>
<tr>
<th>stLON_Com</th>
<th>ST_LON_Communication</th>
</tr>
</thead>
</table>

**stLON_Com:** This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.34 **FB_READ_037_SNVT_str_int**

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_str_int.

**SNVT number:** 037.

**Description:** Wide character string with own code (max. 15 characters) (wide character string).
VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

stValue : ST_LON_SNVT_str_int;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

stValue: Structure of the received data (see ST_LON_SNVT_str_int [618]).

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.35 FB_READ_038_SNVT_telcom

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_telcom.

SNVT number: 038.

Description: Telephone status (telephone status name).

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.
VAR_OUTPUT

eValue : E_LON_telcom_states_t;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

**eValue:** Enum to be received (see E_LON_telcom_states_t [548]).

**bNewData:** Becomes TRUE for 1 cycle when the block has received data.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

**eError:** This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

**stLON_Com:** This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.36 FB_READ_039_SNVT_temp

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_temp.

**SNVT number:** 039.

**Description:** Temperature (°C).

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

**wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled:** TRUE = deselection of the block.

VAR_OUTPUT

rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

**rValue:** Min: 0 / Max: 6553.5.

**bNewData:** Becomes TRUE for 1 cycle when the block has received data.
bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.37 FB_READ_041_SNVT_vol

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_vol.
SNVT number: 041.
Description: Volume (liter).

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

rValue: Min: 0 / Max: 6553.5.
bNewData: Becomes TRUE for 1 cycle when the block has received data.
bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.
eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;
stLON_Com: This structure is used to link FB_LON_KL6401() with the send/receive function (see ST_LON_Communication).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.38 FB_READ_042_SNVT_vol_kilo

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_vol_kilo.

**SNVT number:** 042.

**Description:** Volume (kiloliter).

**VAR_INPUT**

- **wNVIndex** : WORD;
- **bDisabled** : BOOL := FALSE;

**wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

- **rValue** : REAL;
- **bNewData** : BOOL;
- **bError** : BOOL;
- **eError** : E_LON_ERROR;

**rValue:** Min: 0 / Max: 6553.5.

**bNewData:** Becomes TRUE for 1 cycle when the block has received data.

**bError:** This becomes TRUE as soon as an error occurs. This error is described via the variable **eError**.

**eError:** This output outputs an error code in the event of an error (see E_LON_ERROR). **bError** goes TRUE at the same time.

**VAR_IN_OUT**

- **stLON_Com** : ST_LON_Communication;

**stLON_Com:** This structure is used to link FB_LON_KL6401() with the send/receive function (see ST_LON_Communication).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_vol_mil.

**SNVT number:** 043.

**Description:** Volume (milliliter).

**VAR_INPUT**

```plaintext
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

```plaintext
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 6553.5.

**bNewData:** Becomes TRUE for 1 cycle when the block has received data.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.

**eError:** This output outputs an error code in the event of an error (see `E_LON_ERROR{477}`). `bError` goes TRUE at the same time.

**VAR_IN_OUT**

```plaintext
stLON_Com : ST_LON_Communication;
```

**stLON_Com:** This structure is used to link `FB_LON_KL6401()` with the send/receive function (see `ST_LON_Communication{587}`).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

---

4.1.2.40 **FB_READ_044_SNVT_volt**

```plaintext
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**VAR_INPUT**

```plaintext
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**VAR_OUTPUT**

```plaintext
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**VAR_IN_OUT**

```plaintext
stLON_Com : ST_LON_Communication;
```
This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_volt.

**SNVT number:** 044.

**Description:** Electrical voltage (volt).

**VAR_INPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>WORD</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL := FALSE;</td>
</tr>
</tbody>
</table>

**wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>rValue</td>
<td>REAL</td>
</tr>
<tr>
<td>bNewData</td>
<td>BOOL</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
</tr>
</tbody>
</table>

**rValue:** Min: -3276.8 / Max: 3276.7.

**bNewData:** Becomes TRUE for 1 cycle when the block has received data.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

**eError:** This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

**VAR_IN_OUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>stLON_Com</td>
<td>ST_LON_Communication;</td>
</tr>
</tbody>
</table>

**stLON_Com:** This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.41  **FB_READ_045_SNVT_volt_dbmv**

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_volt_dbmv.

**SNVT number:** 045.

**Description:** Electrical voltage (dB microvolt).
VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

rValue: Min: -327.68 / Max: 327.67.

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.42 FB_READ_046_SNVT_volts_kilo

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_volts_kilo.

**SNVT number:** 046.

**Description:** Electrical voltage (kilovolt).

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.
VAR_OUTPUT

rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

rValue: Min: -3276.8 / Max: 3276.7.

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.43 FB_READ_047_SNVT_volt_mil

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_volt_mil.
SNVT number: 047.
Description: Electrical voltage (millivolt).

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

rValue: Min: -3276.8 / Max: 3276.7.

bNewData: Becomes TRUE for 1 cycle when the block has received data.
bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT
stLON_Com : ST_LON_Communication;

stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.44   FB_READ_048_SNVT_amp_f

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_amp_f.

SNVT number: 048.

Description: Electric current (ampere).

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

rValue: Min: -3.40E+38 / Max: 3.40E+38.

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;
**stLON.Com**: This structure is used to link `FB_LON_KL6401()` with the send/receive function (see `ST_LON_Communication`).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.2.45 FB_READ_049_SNVT_angle_f

This function block receives the following LON input variable (nvi):

**SNVT Name**: SNVT_angle_f.

**SNVT number**: 049.

**Description**: Angular distance (radian).

**VAR_INPUT**

```plaintext
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

- **wNVIndex**: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **bDisabled**: TRUE = deselection of the block.

**VAR_OUTPUT**

```plaintext
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

- **rValue**: Min: -3.40E+38 / Max: 3.40E+38.
- **bNewData**: Becomes TRUE for 1 cycle when the block has received data.
- **bError**: This becomes TRUE as soon as an error occurs. This error is described via the variable `eError`.
- **eError**: This output outputs an error code in the event of an error (see `E_LON_ERROR`). bError goes TRUE at the same time.

**VAR_IN_OUT**

```plaintext
stLON_Com : ST_LON_Communication;
```

- **stLON_Com**: This structure is used to link `FB_LON_KL6401()` with the send/receive function (see `ST_LON_Communication`).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.1.2.46  FB_READ_050_SNVT_angle_vel_f

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_angle_vel_f.

**SNVT number:** 050.

**Description:** Angular velocity (radian / second).

**VAR_INPUT**

```plaintext
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

- **wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

```plaintext
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

- **rValue:** Min: -3.40E+38 / Max: 3.40E+38.
- **bNewData:** Becomes TRUE for 1 cycle when the block has received data.
- **bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable eError.
- **eError:** This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

**VAR_IN_OUT**

```plaintext
stLON_Com : ST_LON_Communication;
```

- **stLON_Com:** This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.47  FB_READ_051_SNVT_count_f
This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_count_f.

**SNVT number:** 051.

**Description:** Absolute number (items).

**VAR_INPUT**

- `wNVIndex`: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

- `bDisabled`: TRUE = deselection of the block.

**VAR_OUTPUT**

- `rValue`: Min: 0 / Max: 3.40E+38.

- `bNewData`: Becomes TRUE for 1 cycle when the block has received data.

- `bError`: This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.

- `eError`: This output outputs an error code in the event of an error (see `E_LON_ERROR`). `bError` goes TRUE at the same time.

**VAR_IN_OUT**

- `stLON_Com`: This structure is used to link `FB_LON_KL6401()` with the send/receive function (see `ST_LON_Communication`).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.2.48 FB_READ_052_SNVT_count_inc_f**

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_count_inc_f.

**SNVT number:** 052.

**Description:** Increment counter (items(delta)).
**VAR_INPUT**

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

**wNVIndex**: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled**: TRUE = deselection of the block.

**VAR_OUTPUT**

rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

**rValue**: Min: -3.40E+38 / Max: 3.40E+38.

**bNewData**: Becomes TRUE for 1 cycle when the block has received data.

**bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable **eError**.

**eError**: This output outputs an error code in the event of an error (see **E_LON_ERROR**). bError goes TRUE at the same time.

**VAR_IN_OUT**

stLON_Com : ST_LON_Communication;

**stLON_Com**: This structure is used to link **FB_LON_KL6401()** with the send/receive function (see **ST_LON_Communication**).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.2.49 FB_READ_053_SNVT_flow_f**

This function block receives the following LON input variable (nvi):

**SNVT Name**: SNVT_flow_f.

**SNVT number**: 053.

**Description**: Volume flow (liters / second).

**VAR_INPUT**

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

**wNVIndex**: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled**: TRUE = deselection of the block.
VAR_OUTPUT

rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

rValue: Min: -3.40E+38 / Max: 3.40E+38.

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.50 FB_READ_054_SNVT_length_f

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_length_f.

SNVT number: 054.

Description: Length (meter).

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

rValue: Min: 0 / Max: 3.40E+38.

bNewData: Becomes TRUE for 1 cycle when the block has received data.
**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.

**eError:** This output outputs an error code in the event of an error (see `E_LON_ERROR`). `bError` goes TRUE at the same time.

**VAR_IN_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON_Com** : This structure is used to link `FB_LON_KL6401()` with the send/receive function (see `ST_LON_Communication`).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.2.51 **FB_READ_055_SNVT_lev_cont_f**

This function block receives the following LON input variable (nvi):

- **SNVT Name:** `SNVT_lev_cont_f`.
- **SNVT number:** 055.
- **Description:** Continuous change in value (% from maximum level).

**VAR_INPUT**

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

- **wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

```
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

- **rValue:** Min: 0 / Max: 100.
- **bNewData:** Becomes TRUE for 1 cycle when the block has received data.
- **bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.
- **eError:** This output outputs an error code in the event of an error (see `E_LON_ERROR`). `bError` goes TRUE at the same time.

**VAR_IN_OUT**

```
stLON_Com : ST_LON_Communication;
```
stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.2.52 FB_READ_056_SNVT_mass_f

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_mass_f.

**SNVT number:** 056.

**Description:** Mass (gram).

**VAR_INPUT**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex : WORD;</td>
<td>rValue</td>
</tr>
<tr>
<td>bDisabled : BOOL := FALSE;</td>
<td></td>
</tr>
</tbody>
</table>

- **wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>rValue : REAL;</td>
<td>bNewData</td>
</tr>
<tr>
<td>bError : BOOL;</td>
<td>eError</td>
</tr>
</tbody>
</table>

- **rValue:** Min: 0 / Max: 3.40E+38.
- **bNewData:** Becomes TRUE for 1 cycle when the block has received data.
- **bError:** This becomes TRUE as soon as an error occurs. This error is described via the variable **eError**.
- **eError:** This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

**VAR_IN_OUT**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>stLON_Com : ST_LON_Communication;</td>
<td></td>
</tr>
</tbody>
</table>

**stLON_Com :** This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.1.2.53  **FB_READ_057_SNVT_power_f**

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_power_f.
**SNVT number:** 057.
**Description:** Power (watt).

**VAR_INPUT**

- `wNVIndex`: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- `bDisabled`: TRUE = deselection of the block.

**VAR_OUTPUT**

- `rValue`: Min: -3.40E+38 / Max: 3.40E+38.
- `bNewData`: Becomes TRUE for 1 cycle when the block has received data.
- `bError`: This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.
- `eError`: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). `bError` goes TRUE at the same time.

**VAR_IN_OUT**

- `stLON_Com`: This structure is used to link `FB_LON_KL6401()` with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.54  **FB_READ_058_SNVT_ppm_f**
This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_ppm_f.

**SNVT number:** 058.

**Description:** Concentration (ppm).

### VAR_INPUT

<table>
<thead>
<tr>
<th>wNVIndex</th>
<th>WORD</th>
</tr>
</thead>
<tbody>
<tr>
<td>bDisabled</td>
<td>BOOL := FALSE</td>
</tr>
</tbody>
</table>

**wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled:** TRUE = deselection of the block.

### VAR_OUTPUT

<table>
<thead>
<tr>
<th>rValue</th>
<th>REAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>bNewData</td>
<td>BOOL</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
</tr>
</tbody>
</table>

**rValue:** Min: 0 / Max: 3.40E+38.

**bNewData:** Becomes TRUE for 1 cycle when the block has received data.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

**eError:** This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

### VAR_IN_OUT

<table>
<thead>
<tr>
<th>stLON_Com</th>
<th>ST_LON_Communication</th>
</tr>
</thead>
</table>

**stLON_Com:** This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

#### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.55 **FB_READ_059_SNVT_press_f**

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_press_f.

**SNVT number:** 059.

**Description:** Pressure (overpressure) (pascal).
VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

rValue: Min: -3.40E+38 / Max: 3.40E+38.

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.56 FB_READ_060_SNVT_res_f

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.
**VAR_OUTPUT**

- **rValue**: REAL
- **bNewData**: BOOL
- **bError**: BOOL
- **eError**: E_LON_ERROR

**rValue**: Min: 0 / Max: 3.40E+38.

**bNewData**: Becomes TRUE for 1 cycle when the block has received data.

**bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable **eError**.

**eError**: This output outputs an error code in the event of an error (see **E_LON_ERROR** [477]). **bError** goes TRUE at the same time.

**VAR_IN_OUT**

- **stLON_Com**: ST_LON_Communication;

**stLON_Com**: This structure is used to link **FB_LON_KL6401()** [27] with the send/receive function (see **ST_LON_Communication** [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.2.57  ** **FB_READ_061_SNVT_sound_db_f**

This function block receives the following LON input variable (nvi):

**SNVT Name**: SNVT_sound_db_f.

**SNVT number**: 061.

**Description**: Sound pressure level (dBspl).

**VAR_INPUT**

- **wNVIndex**: WORD
- **bDisabled**: BOOL := FALSE;

**wNVIndex**: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled**: TRUE = deselection of the block.

**VAR_OUTPUT**

- **rValue**: REAL
- **bNewData**: BOOL
- **bError**: BOOL
- **eError**: E_LON_ERROR

**rValue**: Min: -3.40E+38 / Max: 3.40E+38.

**bNewData**: Becomes TRUE for 1 cycle when the block has received data.
bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT
stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.58 FB_READ_062_SNVT_speed_f

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_speed_f.

SNVT number: 062.

Description: Linear velocity (meters / second).

VAR_INPUT
wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

rValue: Min: -3.40E+38 / Max: 3.40E+38.

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT
stLON_Com : ST_LON_Communication;
**stLON_Com**: This structure is used to link FB_LON_KL6401() with the send/receive function (see ST_LON_Communication).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.2.59 FB_READ_063_SNVT_temp_f

This function block receives the following LON input variable (nvi):

**SNVT Name**: SNVT_temp_f.

**SNVT number**: 063.

**Description**: Temperature (°C).

**VAR_INPUT**

```plaintext
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

- **wNVIndex**: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **bDisabled**: TRUE = deselection of the block.

**VAR_OUTPUT**

```plaintext
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

- **rValue**: Min: -273.17 / Max: 3.40E+38.
- **bNewData**: Becomes TRUE for 1 cycle when the block has received data.
- **bError**: This goes TRUE as soon as an error occurs. This error is described via the variable `eError`.
- **eError**: This output outputs an error code in the event of an error (see `E_LON_ERROR`). `bError` goes TRUE at the same time.

**VAR_IN_OUT**

```plaintext
stLON_Com : ST_LON_Communication;
```

**stLON_Com**: This structure is used to link FB_LON_KL6401() with the send/receive function (see ST_LON_Communication).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.1.2.60  FB_READ_064_SNVT_time_f

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_time_f.

**SNVT number:** 064.

**Description:** Elapsed time (seconds).

**VAR_INPUT**

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

- **wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

```
rValue   : REAL;
bNewData  : BOOL;
bError    : BOOL;
eError    : E_LON_ERROR;
```

- **rValue:** Min: 0 / Max: 3.40E+38.
- **bNewData:** Becomes TRUE for 1 cycle when the block has received data.
- **bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable **eError**.
- **eError:** This output outputs an error code in the event of an error (see **E_LON_ERROR**[477]). bError goes TRUE at the same time.

**VAR_IN_OUT**

```
stLON_Com : ST_LON_Communication;
```

- **stLON_Com:** This structure is used to link **FB_LON_KL6401()**[27] with the send/receive function (see **ST_LON_Communication**[587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.61  FB_READ_065_SNVT_vol_f

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_vol_f.

**Description:**
This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_vol_f.

**SNVT number:** 065.

**Description:** Volume (liter).

### VAR_INPUT

- `wNVIndex` : WORD;
- `bDisabled` : BOOL := FALSE;

**wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled:** TRUE = deselection of the block.

### VAR_OUTPUT

- `rValue` : REAL;
- `bNewData` : BOOL;
- `bError` : BOOL;
- `eError` : E_LON_ERROR;

**rValue:** Min: 0 / Max: 3.40E+38.

**bNewData:** Becomes TRUE for 1 cycle when the block has received data.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.

**eError:** This output outputs an error code in the event of an error (see `E_LON_ERROR`). `bError` goes TRUE at the same time.

### VAR_IN_OUT

- `stLON_Com` : ST_LON_Communication;

**stLON_Com:** This structure is used to link `FB_LON_KL6401()` with the send/receive function (see `ST_LON_Communication`).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.2.62 FB_READ_066_SNVT_volt_f

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_volt_f.

**SNVT number:** 066.

**Description:** Electrical voltage (volt).
Programming

**VAR_INPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>WORD</td>
<td>Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL := FALSE</td>
<td>TRUE = deselection of the block.</td>
</tr>
</tbody>
</table>

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rValue</td>
<td>REAL</td>
<td>Min: -3.40E+38 / Max: 3.40E+38.</td>
</tr>
<tr>
<td>bNewData</td>
<td>BOOL</td>
<td>Becomes TRUE for 1 cycle when the block has received data.</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
<td>This output goes TRUE as soon as an error occurs. This error is described via the variable eError.</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
<td>This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.</td>
</tr>
</tbody>
</table>

**VAR_IN_OUT**

<table>
<thead>
<tr>
<th>Structure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stLON_Com</td>
<td>This structure is used to link FB_LON_KL6401 with the send/receive function (see ST_LON_Communication [587]).</td>
</tr>
</tbody>
</table>

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.2.63 FB_READ_067_SNVT_btu_f**

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_btu_f.

**SNVT number:** 067.

**Description:** Thermal energy (Btu).

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>WORD</td>
<td>Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL := FALSE</td>
<td>TRUE = deselection of the block.</td>
</tr>
</tbody>
</table>
VAR_OUTPUT

rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

rValue: Min: 0 / Max: 3.40E+38.

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.64 FB_READ_068_SNVT_elec_whr_f

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_elec_whr_f.

SNVT number: 068.

Description: Electric energy (Watt / hour).

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

rValue: Min: 0 / Max: 3.40E+38.

bNewData: Becomes TRUE for 1 cycle when the block has received data.
**bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable **eError**.

**eError**: This output outputs an error code in the event of an error (see **E_LON_ERROR** [477]). **bError** goes TRUE at the same time.

**VAR_IN_OUT**

```plaintext
stLON_Com : ST_LON_Communication;
```

**stLON_Com**: This structure is used to link **FB_LON_KL6401()** [27] with the send/receive function (see **ST_LON_Communication** [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.2.65  **FB_READ_069_SNVT_config_src**

This function block receives the following LON input variable (nvi):

**SNVT Name**: **SNVT_config_src**.

**SNVT number**: 069.

**Description**: Configuration properties (name of the configuration source (0=own, 1=external)).

**VAR_INPUT**

```plaintext
wNVIndex    : WORD;
bDisabled   : BOOL := FALSE;
```

**wNVIndex**: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled**: TRUE = deselection of the block.

**VAR_OUTPUT**

```plaintext
eValue      : E_LON_config_source_t;
bNewData    : BOOL;
bError      : BOOL;
eError      : E_LON_ERROR;
```

**eValue**: Enum to be received (see **E_LON_config_source_t** [509]).

**bNewData**: Becomes TRUE for 1 cycle when the block has received data.

**bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable **eError**.

**eError**: This output outputs an error code in the event of an error (see **E_LON_ERROR** [477]). **bError** goes TRUE at the same time.

**VAR_IN_OUT**

```plaintext
stLON_Com : ST_LON_Communication;
```
**stLON_Com**: This structure is used to link `FB_LON_KL6401()` [27] with the send/receive function (see `ST_LON_Communication` [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.2.66  FB_READ_070_SNVT_color**

This function block receives the following LON input variable (nvi):

- **SNVT Name**: SNVT_color.
- **SNVT number**: 070.
- **Description**: Color according to CIE standard (L*,a*,b).

**VAR_INPUT**

```plaintext
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

- **wNVIndex**: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **bDisabled**: TRUE = deselection of the block.

**VAR_OUTPUT**

```plaintext
stValue : ST_LON_SNVT_color;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

- **stValue**: Structure of the received data (see `ST_LON_SNVT_color` [596]).
- **bNewData**: Becomes TRUE for 1 cycle when the block has received data.
- **bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.
- **eError**: This output outputs an error code in the event of an error (see `E_LON_ERROR` [477]). `bError` goes TRUE at the same time.

**VAR_IN_OUT**

```plaintext
stLON_Com : ST_LON_Communication;
```

- **stLON_Com**: This structure is used to link `FB_LON_KL6401()` [27] with the send/receive function (see `ST_LON_Communication` [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.1.2.67 FB_READ_071_SNVT_grammage

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_grammage.

**SNVT number:** 071.

**Description:** American dimension for paper weight and density (gram/m2).

**VAR_INPUT**

| wNVIndex : WORD; |
| bDisabled : BOOL := FALSE; |

**wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

| rValue : REAL; |
| bNewData : BOOL; |
| bError : BOOL; |
| eError : E_LON_ERROR; |

**rValue:** Min: 0 / Max: 6553.5.

**bNewData:** Becomes TRUE for 1 cycle when the block has received data.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable **eError**.

**eError:** This output outputs an error code in the event of an error (see **E_LON_ERROR** [477]). **bError** goes TRUE at the same time.

**VAR_IN_OUT**

| stLON_Com : ST_LON_Communication; |

**stLON_Com:** This structure is used to link **FB_LON_KL6401()** [27] with the send/receive function (see **ST_LON_Communication** [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.68 FB_READ_072_SNVT_grammage_f

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_grammage.

**SNVT number:** 071.

**Description:** American dimension for paper weight and density (gram/m2).

**VAR_INPUT**

| wNVIndex : WORD; |
| bDisabled : BOOL := FALSE; |

**wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

| rValue : REAL; |
| bNewData : BOOL; |
| bError : BOOL; |
| eError : E_LON_ERROR; |

**rValue:** Min: 0 / Max: 6553.5.

**bNewData:** Becomes TRUE for 1 cycle when the block has received data.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable **eError**.

**eError:** This output outputs an error code in the event of an error (see **E_LON_ERROR** [477]). **bError** goes TRUE at the same time.

**VAR_IN_OUT**

| stLON_Com : ST_LON_Communication; |

**stLON_Com:** This structure is used to link **FB_LON_KL6401()** [27] with the send/receive function (see **ST_LON_Communication** [587]).
This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_grammage_f.

**SNVT number:** 072.

**Description:** American dimension for paper weight and density (gram/m2).

**VAR_INPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>WORD</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL := FALSE;</td>
</tr>
</tbody>
</table>

**wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>rValue</td>
<td>REAL</td>
</tr>
<tr>
<td>bNewData</td>
<td>BOOL</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR;</td>
</tr>
</tbody>
</table>

**rValue:** Min: 0 / Max: 3.40E+38.

**bNewData:** Becomes TRUE for 1 cycle when the block has received data.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

**eError:** This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

**VAR_IN_OUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>stLON_Com</td>
<td>ST_LON_Communication;</td>
</tr>
</tbody>
</table>

**stLON_Com:** This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.2.69  FB_READ_073_SNVT_file_req**

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_file_req.

**SNVT number:** 073.

**Description:** File request.
### VAR_INPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>wNVIndex</code></td>
<td>Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.</td>
</tr>
<tr>
<td><code>bDisabled</code></td>
<td>TRUE = deselection of the block.</td>
</tr>
</tbody>
</table>

### VAR_OUTPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>stValue</code></td>
<td>Structure of the received data (see <a href="#">ST_LON_SNVT_file_req</a>)</td>
</tr>
<tr>
<td><code>bNewData</code></td>
<td>Becomes TRUE for 1 cycle when the block has received data.</td>
</tr>
<tr>
<td><code>bError</code></td>
<td>This output goes TRUE as soon as an error occurs. This error is described via the variable <code>eError</code>.</td>
</tr>
<tr>
<td><code>eError</code></td>
<td>This output outputs an error code in the event of an error (see <a href="#">E_LON_ERROR</a>). bError goes TRUE at the same time.</td>
</tr>
</tbody>
</table>

### VAR_IN_OUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>stLON_Com</code></td>
<td>This structure is used to link <a href="#">FB_LON_KL6401()</a> with the send/receive function (see <a href="#">ST_LON_Communication</a>).</td>
</tr>
</tbody>
</table>

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.2.70 FB_READ_074_SNVT_file_status

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_file_status.

**SNVT number:** 074.

**Description:** File information (part of the LONWORKS file transfer protocol (LW-FTP)).

### VAR_INPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>wNVIndex</code></td>
<td>Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.</td>
</tr>
<tr>
<td><code>bDisabled</code></td>
<td>TRUE = deselection of the block.</td>
</tr>
</tbody>
</table>
VAR_OUTPUT

stValue : ST_LON_SNVT_file_status;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

stValue: Structure of the received data (see ST_LON_SNVT_file_status [603]). If stValue.eStatus = eLON_FS_LOOKUP_OK (1), then the structure stValue.stAddr.stDescriptor is valid, otherwise stValue.stAddr.stAddress is valid.

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.71 FB_READ_075_SNVT_freq_f

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_freq_f.

SNVT number: 075.

Description: Frequency (hertz).

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

rValue: Min: 0 / Max: 3.40E+38.
bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.72 FB_READ_076_SNVT_freq_hz

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_freq_hz.

SNVT number: 076.

Description: Frequency (hertz).

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

rValue: Min: 0 / Max: 6553.5.

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;
stLON_Com : This structure is used to link FB_LON_KL6401() with the send/receive function (see ST_LON_Communication).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.73 FB_READ_077_SNVT_freq_kilohz

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_freq_kilohz.

**SNVT number:** 077.

**Description:** Frequency (kilohertz).

**VAR_INPUT**

| wNVIndex : WORD;            | bDisabled : BOOL := FALSE; |

**wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>rValue : REAL;</th>
<th>bNewData : BOOL;</th>
</tr>
</thead>
<tbody>
<tr>
<td>bError : BOOL;</td>
<td>eError : E_LON_ERROR;</td>
</tr>
</tbody>
</table>

**rValue:** Min: 0 / Max: 6553.5.

**bNewData:** Becomes TRUE for 1 cycle when the block has received data.

**bError:** This becomes TRUE as soon as an error occurs. This error is described via the variable eError.

**eError:** This output outputs an error code in the event of an error (see E_LON_ERROR). bError goes TRUE at the same time.

**VAR_IN_OUT**

| stLON_Com : ST_LON_Communication; |

**stLON_Com:** This structure is used to link FB_LON_KL6401() with the send/receive function (see ST_LON_Communication).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.1.2.74  FB_READ_078_SNVT_freq_milhz

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_freq_milhz.

**SNVT number:** 078.

**Description:** Frequency (millihertz).

**VAR_INPUT**

```plaintext
wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.
```

**VAR_OUTPUT**

```plaintext
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

rValue: Min: 0 / Max: 6553.5.

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.
```

**VAR_IN_OUT**

```plaintext
stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).
```

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.75  FB_READ_079_SNVT_lux
This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_lux.

**SNVT number:** 079.

**Description:** Illuminance (lux) 1 lux = 1 lumen/m².

**VAR_INPUT**

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

**VAR_OUTPUT**

uiValue : UINT;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

uiValue: Min: 0 / Max: 65535.

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR[477]). bError goes TRUE at the same time.

**VAR_IN_OUT**

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401[27] with the send/receive function (see ST_LON_Communication[587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.2.76 FB_READ_081_SNVT_lev_percent**

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_lev_percent.

**SNVT number:** 081.

**Description:** Percentage value in 0.005% steps with sign.
VAR_INPUT

\[
\begin{align*}
\text{wNVIndex} &: \text{WORD}; \\
\text{bDisabled} &: \text{BOOL} := \text{FALSE};
\end{align*}
\]

**wNVIndex**: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled**: TRUE = deselection of the block.

VAR_OUTPUT

\[
\begin{align*}
\text{rValue} &: \text{REAL}; \\
\text{bNewData} &: \text{BOOL}; \\
\text{bError} &: \text{BOOL}; \\
\text{eError} &: \text{E\_LON\_ERROR};
\end{align*}
\]

**rValue**: Min: -163.84 / Max: 163.835.

**bNewData**: Becomes TRUE for 1 cycle when the block has received data.

**bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable \(\text{eError}\).

**eError**: This output outputs an error code in the event of an error (see \(\text{E\_LON\_ERROR}\)) \((\text{eError})\). bError goes TRUE at the same time.

VAR_IN_OUT

\[
\text{stLON\_Com} : \text{ST\_LON\_Communication};
\]

**stLON\_Com**: This structure is used to link \(\text{FB\_LON\_KL6401}()\) \((\text{27})\) with the send/receive function (see \(\text{ST\_LON\_Communication}\)).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.77 \hspace{1em} **FB\_READ\_082\_SNVT\_multiplier**

This function block receives the following LON input variable (nvi):

**SNVT Name**: SNVT\_multiplier.

**SNVT number**: 082.

**Description**: Multiplier in 0.0005 steps (16-bit unsigned value).

VAR_INPUT

\[
\begin{align*}
\text{wNVIndex} &: \text{WORD}; \\
\text{bDisabled} &: \text{BOOL} := \text{FALSE};
\end{align*}
\]

**wNVIndex**: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled**: TRUE = deselection of the block.
VAR_OUTPUT

rValue   : REAL;
bNewData : BOOL;
bError  : BOOL;
eError  : E_LON_ERROR;

rValue: Min: 0 / Max: 32.7675.

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com   : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.78   FB_READ_083_SNVT_state

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_state.

SNVT number: 083.

Description: Status information (16 individual bit values). Each status is a Boolean value.

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

arrValue : ARRAY [0..15] OF BOOL;
bNewData : BOOL;
bError  : BOOL;
eError  : E_LON_ERROR;

arrValue: 0-15 bit.

bNewData: Becomes TRUE for 1 cycle when the block has received data.
bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.79 FB_READ_084_SNVT_time_stamp

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_time_stamp.

SNVT number: 084.

Description: Time stamp (year, month, day, hour, minute, second).

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

stValue : TIMESTRUCT;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

stValue: Structure of the received data (see TIMESTRUCT). The structure variables wDayOfWeek and wMilliseconds are not valid here; these values are always zero.

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;
stLON_Com: This structure is used to link FB_LON_KL6401() \[27\] with the send/receive function (see ST_LON_Communication \[587\]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.80 FB_READ_085_SNVT_zerospan

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_zerospan.

SNVT number: 085.

Description: Zero point and proportionality factor. Linear transformation parameters: multiply with proportionality factor, then add origin offset.

**VAR_INPUT**

```plaintext
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

**VAR_OUTPUT**

```plaintext
stValue : ST_LON_SNVT_zerospan;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

stValue: Structure of the received data (see ST_LON_SNVT_zerospan \[621\]).

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR \[477\]). bError goes TRUE at the same time.

**VAR_IN_OUT**

```plaintext
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure is used to link FB_LON_KL6401() \[27\] with the send/receive function (see ST_LON_Communication \[587\]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.1.2.81 FB_READ_086_SNVT_magcard

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_magcard.

**SNVT number:** 086.

**Description:** ISO 7811 (40 hexadecimal numbers). Data according to ISO 7811 standard for magnetic card readers.

**VAR_INPUT**

- `wNVIndex : WORD;`
- `bDisabled : BOOL := FALSE;`

**wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

- `arrValue : ARRAY [0..40] OF BYTE;`
- `bNewData : BOOL;`
- `bError : BOOL;`
- `eError : E_LON_ERROR;`

**arrValue:** 1-40 BYTE.

**bNewData:** Becomes TRUE for 1 cycle when the block has received data.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.

**eError:** This output outputs an error code in the event of an error (see `E_LON_ERROR[477]`). bError goes TRUE at the same time.

**VAR_IN_OUT**

- `stLON_Com : ST_LON_Communication;`

**stLON_Com:** This structure is used to link `FB_LON_KL6401()` with the send/receive function (see `ST_LON_Communication[587]`).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.1.2.82 FB_READ_087_SNVT_elapsed_tm

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_elapsed_tm.

**SNVT number:** 087.

**Description:** Elapsed time (day, hour, minute, second, millisecond).

**VAR_INPUT**

- wNVIndex : WORD;
- bDisabled : BOOL := FALSE;

**wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

- stValue : ST_LON_SNVT_elapsed_tm;
- bNewData : BOOL;
- bError : BOOL;
- eError : E_LON_ERROR;

**stValue:** Structure of the received data (see ST_LON_SNVT_elapsed_tm [599]).

**bNewData:** Becomes TRUE for 1 cycle when the block has received data.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

**eError:** This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

**VAR_IN_OUT**

- stLON_Com : ST_LON_Communication;

**stLON_Com:** This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.83 FB_READ_088_SNVT_alarm
This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_alarm.

**SNVT number:** 088.

**Description:** Alarm status.

**VAR_INPUT**

<table>
<thead>
<tr>
<th>wNVIndex</th>
<th>WORD;</th>
</tr>
</thead>
<tbody>
<tr>
<td>bDisabled</td>
<td>BOOL := FALSE;</td>
</tr>
</tbody>
</table>

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>stValue</th>
<th>ST_LON_SNVT_alarm;</th>
</tr>
</thead>
<tbody>
<tr>
<td>bNewData</td>
<td>BOOL;</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL;</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR;</td>
</tr>
</tbody>
</table>

stValue: Structure of the received data (see ST_LON_SNVT_alarm [591]).

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

**VAR_IN_OUT**

<table>
<thead>
<tr>
<th>stLON_Com</th>
<th>ST_LON_Communication;</th>
</tr>
</thead>
</table>

stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.2.84 FB_READ_089_SNVT_currency**

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_currency.

**SNVT number:** 089.

**Description:** Currency (unit (euros,...), multiplier, value).
VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

stValue : ST_LON_SNVT_currency;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

stValue: Structure of the received data (see ST_LON_SNVT_currency [597]).

bNewData: Becomes TRUE for 1 cycle when the block has received data.

eError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.85   FB_READ_090_SNVT_file_pos

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_file_pos.

SNVT number: 090.

Description: File position (pointer, length).

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.
**VAR_OUTPUT**

stValue : ST_LON_SNVT_file_pos;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

**stValue**: Structure of the received data (see ST_LON_SNVT_file_pos [602]).

**bNewData**: Becomes TRUE for 1 cycle when the block has received data.

**bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable **eError**.

**eError**: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

**VAR_IN_OUT**

stLON_Com : ST_LON_Communication;

**stLON_Com**: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.86 **FB_READ_091_SNVT_muldiv**

This function block receives the following LON input variable (nvi):

**SNVT Name**: SNVT_muldiv.

**SNVT number**: 091.

**Description**: Gain factor/attenuation factor (multiplier, divisor).

**VAR_INPUT**

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

**wNVIndex**: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled**: TRUE = deselection of the block.

**VAR_OUTPUT**

stValue : ST_LON_SNVT_muldiv;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

**stValue**: Structure of the received data (see ST_LON_SNVT_muldiv [608]).

**bNewData**: Becomes TRUE for 1 cycle when the block has received data.
bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.87 FB_READ_092_SNVT_obj_request

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_obj_request.

**SNVT number:** 092.

**Description:** Function selection (ID, request).

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

stValue : ST_LON_SNVT_obj_request;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

stValue: Structure of the received data (see ST_LON_SNVT_obj_request [609]).

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;
stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LONCommunication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.2.88 FB_READ_093_SNVT_obj_status

This function block receives the following LON input variable (nvi):

**SNVT Name**: SNVT_obj_status.

**SNVT number**: 093.

**Description**: Object status (ID, status (4 byte)).

**VAR_INPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.</td>
</tr>
<tr>
<td>bDisabled</td>
<td>TRUE = deselection of the block.</td>
</tr>
</tbody>
</table>

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stValue</td>
<td>Structure of the received data (see ST_LON_SNVT_obj_status [609]).</td>
</tr>
<tr>
<td>bNewData</td>
<td>Becomes TRUE for 1 cycle when the block has received data.</td>
</tr>
<tr>
<td>bError</td>
<td>This output goes TRUE as soon as an error occurs. This error is described via the variable eError.</td>
</tr>
<tr>
<td>eError</td>
<td>This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.</td>
</tr>
</tbody>
</table>

**VAR_IN_OUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stLON_Com</td>
<td>This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LONCommunication [587]).</td>
</tr>
</tbody>
</table>

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_preset.

**SNVT number:** 094.

**Description:** Default (learn mode, value, time).

**VAR_INPUT**

```plaintext
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

- **wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

```plaintext
stValue : ST_LON_SNVT_preset;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

- **stValue:** Structure of the received data (see `ST_LON_SNVT_preset`).
- **bNewData:** Becomes TRUE for 1 cycle when the block has received data.
- **bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.
- **eError:** This output outputs an error code in the event of an error (see `E_LON_ERROR`). `bError` goes TRUE at the same time.

**VAR_IN_OUT**

```plaintext
stLON_Com : ST_LON_Communication;
```

- **stLON_Com:** This structure is used to link `FB_LON_KL6401()` with the send/receive function (see `ST_LON_Communication`).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.2.90 FB_READ_095_SNVT_switch**

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_preset.

**SNVT number:** 095.

**Description:** Switching function for SNVT_preset 094.
This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_switch.

**SNVT number:** 095.

**Description:** Switch (value, status).

### VAR_INPUT

- `wNVIndex` : WORD;
- `bDisabled` : BOOL := FALSE;

**wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled:** TRUE = deselection of the block.

### VAR_OUTPUT

- `stValue` : ST_LON_SNVT_switch;
- `bNewData` : BOOL;
- `bError` : BOOL;
- `eError` : E_LON_ERROR;

**stValue:** Structure of the received data (see `ST_LON_SNVT_switch[618]`).

**bNewData:** Becomes TRUE for 1 cycle when the block has received data.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.

**eError:** This output outputs an error code in the event of an error (see `E_LON_ERROR[477]`). `bError` goes TRUE at the same time.

### VAR_IN_OUT

- `stLON_Com` : ST_LON_Communication;

**stLON_Com:** This structure is used to link `FB_LON_KL6401() [27]` with the send/receive function (see `ST_LON_Communication[587]`).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.2.91 FB_READ_096_SNVT_trans_table

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_trans_table.

**SNVT number:** 096.

**Description:** Conversion table (number of values, interpolation).
VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

stValue : ST_LON_SNVT_trans_table;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

stValue: Structure of the received data (see ST_LON_SNVT_trans_table [621]).

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.92  FB_READ_097_SNVT_override

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_override.

SNVT number: 097.

Description: Override mode (enumeration: 0= keep current value, 1= set specific value set, 2= set default value).

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.
VAR_OUTPUT

eValue : E_LON_override_t;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

eValue: Enum to be received (see E_LON_override_t [532]).

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.93 FB_READ_098_SNVT_pwr_fact

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_pwr_fact.

SNVT number: 098.

Description: Power factor (multiplier).

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

rValue: Min: -1 / Max: 1.

bNewData: Becomes TRUE for 1 cycle when the block has received data.
**VAR_IN_OUT**

```pascal
stLON_Com : ST_LON_Communication;
```

**stLON_Com**: This structure is used to link `FB_LON_KL6401()` with the send/receive function (see `ST_LON_Communication`).

---

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

---

### 4.1.2.94 **FB_READ_099_SNVT_pwr_fact_f**

This function block receives the following LON input variable (nvi):

**SNVT Name**: SNVT_pwr_fact_f.

**SNVT number**: 099.

**Description**: Power factor (multiplier).

**VAR_INPUT**

```pascal
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex**: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled**: TRUE = deselection of the block.

**VAR_OUTPUT**

```pascal
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**rValue**: Min: -1 / Max: 1.

**bNewData**: Becomes TRUE for 1 cycle when the block has received data.

**bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.

**eError**: This output outputs an error code in the event of an error (see `E_LON_ERROR`). `bError` goes TRUE at the same time.

**VAR_IN_OUT**

```pascal
stLON_Com : ST_LON_Communication;
```
**stLON_Com**: This structure is used to link FB_LON_KL6401() with the send/receive function (see ST_LON_Communication).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.2.95 FB_READ_100_SNVT_density

This function block receives the following LON input variable (nvi):

**SNVT Name**: SNVT_density.

**SNVT number**: 100.

**Description**: Density (kg/m³).

**VAR_INPUT**

```plaintext
wNVIndex : WORD;
```

- **wNVIndex**: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

- **bDisabled**: TRUE = deselection of the block.

**VAR_OUTPUT**

```plaintext
rValue   : REAL;
```

- **rValue**: Min: 0 / Max: 32767.5.

- **bNewData**: Becomes TRUE for 1 cycle when the block has received data.

- **bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable **eError**.

- **eError**: This output outputs an error code in the event of an error (see E_LON_ERROR). **bError** goes TRUE at the same time.

**VAR_IN_OUT**

```plaintext
stLON_Com : ST_LON_Communication;
```

**stLON_Com**: This structure is used to link FB_LON_KL6401() with the send/receive function (see ST_LON_Communication).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

---

*Note: The document contains technical information about a function block in Beckhoff's TwinCAT environment, including its parameters, data types, and usage instructions.*
This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_density_f.

**SNVT number:** 101.

**Description:** Density (kg/m³).

### VAR_INPUT

- `wNVIndex` : WORD
  - `bDisabled` : BOOL := FALSE;
  - `wNVIndex`: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
  - `bDisabled`: TRUE = deselection of the block.

### VAR_OUTPUT

- `rValue` : REAL;
  - `bNewData` : BOOL;
  - `bError` : BOOL;
  - `eError` : E_LON_ERROR;
  - `rValue`: Min: 0 / Max: 3.40E+38.
  - `bNewData`: Becomes TRUE for 1 cycle when the block has received data.
  - `bError`: This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.
  - `eError`: This output outputs an error code in the event of an error (see `E_LON_ERROR` [477]). `bError` goes TRUE at the same time.

### VAR_IN_OUT

- `stLON_Com` : ST_LON_Communication;
  - `stLON_Com` : This structure is used to link `FB_LON_KL6401()` [27] with the send/receive function (see `ST_LON_Communication` [587]).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.2.97  FB_READ_102_SNVT_rpm

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_density_f.

**SNVT number:** 101.

**Description:** Density (kg/m³).
This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT\_rpm.

**SNVT number:** 102.

**Description:** Speed (revolutions/minute (RPM)).

### VAR\_INPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>WORD</td>
<td>Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL := FALSE</td>
<td>TRUE = deselection of the block.</td>
</tr>
</tbody>
</table>

### VAR\_OUTPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>uiValue</td>
<td>UINT</td>
<td>Min: 0 / Max: 65535.</td>
</tr>
<tr>
<td>bNewData</td>
<td>BOOL</td>
<td>Becomes TRUE for 1 cycle when the block has received data.</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
<td>This output goes TRUE as soon as an error occurs. This error is described via the variable eError.</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
<td>This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.</td>
</tr>
</tbody>
</table>

### VAR\_IN\_OUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stLON_Com</td>
<td>ST_LON_Communication</td>
<td>This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).</td>
</tr>
</tbody>
</table>

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.2.98 FB\_READ\_103\_SNVT\_hvac\_emerg

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT\_hvac\_emerg.

**SNVT number:** 103.

**Description:** HVAC emergency mode (operating modes).
### VAR_INPUT

| wNVIndex : WORD; |
| bDisabled : BOOL := FALSE; |

**wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled:** TRUE = deselection of the block.

### VAR_OUTPUT

| eValue : E_LON_emerg_t; |
| bNewData : BOOL; |
| bError : BOOL; |
| eError : E_LON_ERROR; |

**eValue:** Enum to be received (see [E_LON_emerg_t](#) 517).

**bNewData:** Becomes TRUE for 1 cycle when the block has received data.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable **eError**.

**eError:** This output outputs an error code in the event of an error (see [E_LON_ERROR](#) 477). **bError** goes TRUE at the same time.

### VAR_IN_OUT

| stLON_Com : ST_LON_Communication; |

**stLON_Com:** This structure is used to link **FB_LON_KL6401()** 27 with the send/receive function (see [ST_LON_Communication](#) 587).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.2.99 FB_READ_104_SNVT_angle_deg

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_angle_deg.

**SNVT number:** 104.

**Description:** Angle specification in 1/50 degree steps.

### VAR_INPUT

| wNVIndex : WORD; |
| bDisabled : BOOL := FALSE; |

**wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled:** TRUE = deselection of the block.
VAR_OUTPUT

rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

rValue: Min: -359.98 / Max: 360.

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.100   FB_READ_105_SNVT_temp_p

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_temp_p.

SNVT number: 105.

Description: Temperature (°C).

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

rValue: Min: -273.17 / Max: 327.67.

bNewData: Becomes TRUE for 1 cycle when the block has received data.
bError: This output goes TRUE as soon as an error occurs. This error is described via the variable \textit{eError}.

eError: This output outputs an error code in the event of an error (see \texttt{E\_LON\_ERROR [477]}). \textit{bError} goes TRUE at the same time.

**VAR-IN-OUT**

\begin{verbatim}
stLON_Com : ST_LON_Communication;
\end{verbatim}

\texttt{stLON\_Com}: This structure is used to link \texttt{FB\_LON\_KL6401()} with the send/receive function (see \texttt{ST\_LON\_Communication [587]}).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.101 \textbf{FB\_READ\_106\_SNVT\_temp\_setpt}

- \texttt{wNVIndex}
- \texttt{bDisabled}
- \texttt{stLON\_Com}

\begin{verbatim}
VAR_INPUT
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
\end{verbatim}

\texttt{wNVIndex}: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

\texttt{bDisabled}: TRUE = deselection of the block.

\begin{verbatim}
VAR_OUTPUT
stValue : ST\_LON\_SNVT\_temp\_setpt;
bNewData : BOOL;
bError : BOOL;
eError : E\_LON\_ERROR;
\end{verbatim}

\texttt{stValue}: Structure of the received data (see \texttt{ST\_LON\_SNVT\_temp\_setpt [619]}).

\texttt{bNewData}: Becomes TRUE for 1 cycle when the block has received data.

\texttt{bError}: This output goes TRUE as soon as an error occurs. This error is described via the variable \texttt{eError}.

\texttt{eError}: This output outputs an error code in the event of an error (see \texttt{E\_LON\_ERROR [477]}). \texttt{bError} goes TRUE at the same time.

**VAR-IN-OUT**

\begin{verbatim}
stLON_Com : ST\_LON\_Communication;
\end{verbatim}
**stLON_Com**: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.2.102 FB_READ_107_SNVT_time_sec

This function block receives the following LON input variable (nvi):

**SNVT Name**: SNVT_time_sec.

**SNVT number**: 107.

**Description**: Elapsed time (second).

**VAR_INPUT**

```plaintext
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

- **wNVIndex**: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **bDisabled**: TRUE = deselection of the block.

**VAR_OUTPUT**

```plaintext
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

- **rValue**: Min: 0 / Max: 6553.5.
- **bNewData**: Becomes TRUE for 1 cycle when the block has received data.
- **bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.
- **eError**: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

**VAR_IN_OUT**

```plaintext
stLON_Com : ST_LON_Communication;
```

**stLON_Com**: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_hvac_mode.

**SNVT number:** 108.

**Description:** HVAC operating mode (operating modes).

**VAR_INPUT**

- `wNVIndex`: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- `bDisabled`: TRUE = deseletion of the block.

**VAR_OUTPUT**

- `eValue`: Enum to be received (see `E_LON_hvac_t`).
- `bNewData`: Becomes TRUE for 1 cycle when the block has received data.
- `bError`: This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.
- `eError`: This output outputs an error code in the event of an error (see `E_LON_ERROR`). `bError` goes TRUE at the same time.

**VAR_IN_OUT**

- `stLON_Com`: This structure is used to link `FB_LON_KL6401()` with the send/receive function (see `ST_LON_Communication`).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.2.104 FB_READ_109_SNVT_occupancy**
This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_occupancy.

**SNVT number:** 109.

**Description:** Occupancy signal (states).

### VAR_INPUT

- **wNVIndex** : WORD;
- **bDisabled** : BOOL := FALSE;

**wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled:** TRUE = deselection of the block.

### VAR_OUTPUT

- **eValue** : E_LON_occup_t;
- **bNewData** : BOOL;
- **bError** : BOOL;
- **eError** : E_LON_ERROR;

**eValue:** Enum to be sent (see E_LON_occup_t[532]).

**bNewData:** Becomes TRUE for 1 cycle when the block has received data.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

**eError:** This output outputs an error code in the event of an error (see E_LON_ERROR[477]). bError goes TRUE at the same time.

### VAR_IN_OUT

- **stLON_Com** : ST_LON_Communication;

**stLON_Com:** This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication[587]).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.2.105 FB_READ_110_SNVT_area

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_area.

**SNVT number:** 110.

**Description:** Area (square meter).
VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

rValue: Min: 0 / Max: 13.107.

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.106 FB_READ_111_SNVT_hvac_overid

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_hvac_overid.

SNVT number: 111.

Description: HVAC override mode (mode, position/flow rate, min/max flow rate).

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.
VAR_OUTPUT

stValue : ST_LON_SNVT_hvac_overid;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

stValue: Structure of the received data (see ST_LON_SNVT_hvac_overid [604]).

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.107 FB_READ_112_SNVT_hvac_status

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_hvac_status.

SNVT number: 112.

Description: HVAC status (mode, 5 power ratings, alarm flag).

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

stValue : ST_LON_SNVT_hvac_status;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

stValue: Structure of the received data (see ST_LON_SNVT_hvac_status [605]).

bNewData: Becomes TRUE for 1 cycle when the block has received data.
bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.108 FB_READ_113_SNVT_press_p

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_press_p.

SNVT number: 113.

Description: Pressure (overpressure) (pascal).

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

iValue : INT;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

iValue: Min: -32768 / Max: 32767.

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;
stLON_Com : This structure is used to link FB_LON_KL6401() \([27]\) with the send/receive function (see ST_LON_Communication \([587]\)).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

#### 4.1.2.109 FB_READ_114_SNVT_address

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_address.

**SNVT number:** 114.

**Description:** Neuron address (16-bit address value).

**VAR_INPUT**

```plaintext
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

```plaintext
uiValue : UINT;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**uiValue:** Min: 16384 / Max: 64767.

**bNewData:** Becomes TRUE for 1 cycle when the block has received data.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.

**eError:** This output outputs an error code in the event of an error (see `E_LON_ERROR` \([477]\)). `bError` goes TRUE at the same time.

**VAR_IN_OUT**

```plaintext
stLON_Com : ST_LON_Communication;
```

**stLON_Com:** This structure is used to link FB_LON_KL6401() \([27]\) with the send/receive function (see ST_LON_Communication \([587]\)).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.1.2.110  FB_READ_115_SNVT_scene

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_scene.

**SNVT number:** 115.

**Description:** Scenes (function (retrieve/learn), scene number).

### VAR_INPUT

- **wNVIndex**: WORD;
- **bDisabled**: BOOL := FALSE;

**wNVIndex**: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled**: TRUE = deselection of the block.

### VAR_OUTPUT

- **stValue**: ST_LON_SNVT_scene;
- **bNewData**: BOOL;
- **bError**: BOOL;
- **eError**: E_LON_ERROR;

**stValue**: Structure of the received data (see ST_LON_SNVT_scene [617]).

**bNewData**: Becomes TRUE for 1 cycle when the block has received data.

**bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable **eError**.

**eError**: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

### VAR_IN_OUT

- **stLON_Com**: ST_LON_Communication;

**stLON_Com**: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.111  FB_READ_116_SNVT_scene_cfg

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_scene.

**SNVT number:** 116.

**Description:** Scenes (function (retrieve/learn), scene number).

### VAR_INPUT

- **wNVIndex**: WORD;
- **bDisabled**: BOOL := FALSE;

**wNVIndex**: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled**: TRUE = deselection of the block.

### VAR_OUTPUT

- **stValue**: ST_LON_SNVT_scene;
- **bNewData**: BOOL;
- **bError**: BOOL;
- **eError**: E_LON_ERROR;

**stValue**: Structure of the received data (see ST_LON_SNVT_scene [617]).

**bNewData**: Becomes TRUE for 1 cycle when the block has received data.

**bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable **eError**.

**eError**: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

### VAR_IN_OUT

- **stLON_Com**: ST_LON_Communication;

**stLON_Com**: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).
This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_scene_cfg.

**SNVT number:** 116.

**Description:** Scene setting (function, scene number, setting, transition number, transition time, delay time, priority).

**VAR_INPUT**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>WORD</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL</td>
</tr>
</tbody>
</table>

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>stValue</td>
<td>ST_LON_SNVT_scene_cfg</td>
</tr>
<tr>
<td>bNewData</td>
<td>BOOL</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
</tr>
</tbody>
</table>

stValue: Structure of the received data (see ST_LON_SNVT_scene_cfg).

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR). bError goes TRUE at the same time.

**VAR_IN_OUT**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>stLON_Com</td>
<td>ST_LON_Communication</td>
</tr>
</tbody>
</table>

stLON_Com: This structure is used to link FB_LON_KL6401() with the send/receive function (see ST_LON_Communication).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.2.112  FB_READ_117_SNVT_setting**

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_setting.

**SNVT number:** 117.

**Description:** Settings (function, value, transition number).
VAR_INPUT

\texttt{wNVIndex : WORD;
 bDisabled : BOOL := FALSE;}

\texttt{wNVIndex}: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

\texttt{bDisabled}: TRUE = deselection of the block.

VAR_OUTPUT

\texttt{stValue : ST_LON_SNVT_setting;
 bNewData : BOOL;
 bError : BOOL;
 eError : E_LON_ERROR;}

\texttt{stValue}: Structure of the received data (see \texttt{ST_LON_SNVT_setting} [618]).

\texttt{bNewData}: Becomes TRUE for 1 cycle when the block has received data.

\texttt{bError}: This output goes TRUE as soon as an error occurs. This error is described via the variable \texttt{eError}.

\texttt{eError}: This output outputs an error code in the event of an error (see \texttt{E_LON_ERROR} [477]). \texttt{bError} goes TRUE at the same time.

VAR_IN_OUT

\texttt{stLON_Com : ST_LON_Communication;}

\texttt{stLON_Com}: This structure is used to link \texttt{FB_LON_KL6401()} [27] with the send/receive function (see \texttt{ST_LON_Communication} [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.113 \hspace{1em} FB_READ_118_SNVT_evap_state

\begin{verbatim}
FB_READ_118_SNVT_evap_state
  wNVIndex \hspace{1em} eValue
  bDisabled \hspace{1em} bNewData
  stLON_Com \hspace{1em} bError
  eError
\end{verbatim}

This function block receives the following LON input variable (nvi):

**SNVT Name**: SNVT\_evap\_state.

**SNVT number**: 118.

**Description**: Evaporator status (enumeration).

VAR_INPUT

\texttt{wNVIndex : WORD;
 bDisabled : BOOL := FALSE;}

\texttt{wNVIndex}: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

\texttt{bDisabled}: TRUE = deselection of the block.
VAR_OUTPUT

eValue : E_LON_evap_t;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

eValue: Enum to be received (see E_LON_evap_t). bNewData: Becomes TRUE for 1 cycle when the block has received data. bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError. eError: This output outputs an error code in the event of an error (see E_LON_ERROR). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() with the send/receive function (see ST_LON_Communication).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.114 FB_READ_119_SNVT_therm_mode

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_therm_mode.

SNVT number: 119.

Description: Thermostat mode (enumeration (enable, modulation)).

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

eValue : E_LON_therm_mode_t;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

eValue: Enum to be received (see E_LON_therm_mode_t). bNewData: Becomes TRUE for 1 cycle when the block has received data.
bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.115  FB_READ_120_SNVT_defr_mode

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_defr_mode.

SNVT number: 120.

Description: Defrost mode (enumeration).

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

eValue : E_LON_defrost_mode_t;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

eValue: Enum to be received (see E_LON_defrost_mode_t [513]).

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;
stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.116  FB_READ_121_SNVT_defr_term

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_defr_term.

SNVT number: 121.

Description: Completion of the defrost cycle (enumeration).

VAR_INPUT

<table>
<thead>
<tr>
<th>w NVIndex</th>
<th>eValue</th>
</tr>
</thead>
<tbody>
<tr>
<td>bDisabled</td>
<td>b NewData</td>
</tr>
</tbody>
</table>

w NVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

<table>
<thead>
<tr>
<th>eValue</th>
<th>b NewData</th>
<th>b Error</th>
<th>e Error</th>
</tr>
</thead>
</table>

b Error: This output goes TRUE as soon as an error occurs. This error is described via the variable e Error.

VAR_IN_OUT

stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).
This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_defr_state.

**SNVT number:** 122.

**Description:** Defrost status (enumeration).

**VAR_INPUT**

```plaintext
VAR_INPUT
wNVIndex : WORD;
bool bDisabled := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.
```

**VAR_OUTPUT**

```plaintext
VAR_OUTPUT
eValue : E_LON_defrost_state_t;
brNewData : BOOL := FALSE;
bError : BOOL;

eValue: Enum to be received (see E_LON_defrost_state_t [514]).

brNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.
```

**VAR_IN_OUT**

```plaintext
VAR_IN_OUT
stLON_Com : ST_LON_Communication;

stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).
```

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.2.118  FB_READ_123_SNVT_time_min**
This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_time_min.

**SNVT number:** 123.

**Description:** Elapsed time (minutes).

### VAR_INPUT

```plaintext
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled:** TRUE = deselection of the block.

### VAR_OUTPUT

```plaintext
uiValue : UINT;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**uiValue:** Min: 0 / Max: 65535.

**bNewData:** Becomes TRUE for 1 cycle when the block has received data.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.

**eError:** This output outputs an error code in the event of an error (see `E_LON_ERROR` [477]). `bError` goes TRUE at the same time.

### VAR_IN_OUT

```plaintext
stLON_Com : ST_LON_Communication;
```

**stLON_Com:** This structure is used to link `FB_LON_KL6401()` [27] with the send/receive function (see `ST_LON_Communication` [587]).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

#### 4.1.2.119 **FB_READ_124_SNVT_time_hour**

![FB_READ_124_SNVT_time_hour](image)

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_time_hour.

**SNVT number:** 124.

**Description:** Elapsed time (hour).
VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

uiValue : UINT;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

uiValue: Min: 0 / Max: 65535.

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.120   FB_READ_125_SNVT_ph

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_ph.

SNVT number: 125.

Description: Acidity (pH). Ion concentration ratio.

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.
**VAR_OUTPUT**

- `rValue` : REAL;
- `bNewData` : BOOL;
- `bError` : BOOL;
- `eError` : E_LON_ERROR;

  - **rValue**: Min: -32.768 / Max: 32.767.
  - **bNewData**: Becomes TRUE for 1 cycle when the block has received data.
  - **bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.
  - **eError**: This output outputs an error code in the event of an error (see `E_LON_ERROR`). `bError` goes TRUE at the same time.

**VAR_IN_OUT**

- `stLON_Com` : ST_LON_Communication;

  - `stLON_Com` : This structure is used to link `FB_LON_KL6401()` with the send/receive function (see `ST_LON_Communication`).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.2.121 FB_READ_126_SNVT_ph_f

This function block receives the following LON input variable (nvi):

- **SNVT Name**: SNVT_ph_f.
- **SNVT number**: 126.
- **Description**: Acidity (pH). Ion concentration ratio.

**VAR_INPUT**

- `wNVIndex` : WORD;
- `bDisabled` : BOOL := FALSE;

  - **wNVIndex**: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
  - **bDisabled**: TRUE = deselection of the block.

**VAR_OUTPUT**

- `rValue` : REAL;
- `bNewData` : BOOL;
- `bError` : BOOL;
- `eError` : E_LON_ERROR;

  - **rValue**: Min: -3.40E+38 / Max: 3.40E+38.
  - **bNewData**: Becomes TRUE for 1 cycle when the block has received data.
bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.122   FB_READ_127_SNVT_chlr_status

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_chlr_status.

SNVT number: 127.

Description: Refrigeration unit status (run mode, op mode, status bits).

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

stValue : ST_LON_SNVT_chlr_status;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

stValue: Structure of the received data (see ST_LON_SNVT_chlr_status [593]).

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;
stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.123  FB_READ_128_SNVT_tod_event

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_tod_event.

**SNVT number:** 128.

**Description:** Presence time (busy status, next busy time).

**VAR_INPUT**

- `wNVIndex` : WORD;
- `bDisabled` : BOOL := FALSE;

- **wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

- **bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

- `stValue` : ST_LON_SNVT_tod_event;
- `bNewData` : BOOL;
- `bError` : BOOL;
- `eError` : E_LON_ERROR;

- **stValue:** Structure of the received data (see ST_LON_SNVT_tod_event [620]).

- **bNewData:** Becomes TRUE for 1 cycle when the block has received data.

- **bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.

- **eError:** This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

**VAR_IN_OUT**

- `stLON_Com` : ST_LON_Communication;

- **stLON_Com:** This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_smo_obscur.

**SNVT number:** 129.

**Description:** Darkening due to smoke (darkening %).

**VAR_INPUT**

```
VAR_INPUT
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

```
VAR_OUTPUT
rValue   : REAL;
bNewData : BOOL;
bError   : BOOL;
eError   : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 5.

**bNewData:** Becomes TRUE for 1 cycle when the block has received data.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

**eError:** This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

**VAR_IN_OUT**

```
VAR_IN_OUT
stLON_Com : ST_LON_Communication;
```

**stLON_Com:** This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.2.124 FB_READ_129_SNVT_smo_obscur**

**4.1.2.125 FB_READ_130_SNVT_fire_test**
This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_fire_test.

**SNVT number:** 130.

**Description:** Fire alarm system test request (fire alarm test designations).

**VAR_INPUT**

```plaintext
wNVIndex  :  WORD;
bDisabled :  BOOL := FALSE;
```

**wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

```plaintext
eValue   :  E_LON_fire_test_t;
bNewData :  BOOL;
bError   :  BOOL;
eError   :  E_LON_ERROR;
```

**eValue:** Enum to be received (see `E_LON_fire_test_t [523]`).

**bNewData:** Becomes TRUE for 1 cycle when the block has received data.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.

**eError:** This output outputs an error code in the event of an error (see `E_LON_ERROR [477]`). `bError` goes TRUE at the same time.

**VAR_IN_OUT**

```plaintext
stLON_Com :  ST_LON_Communication;
```

**stLON_Com:** This structure is used to link `FB_LON_KL6401() [27]` with the send/receive function (see `ST_LON_Communication [587]`).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.2.126 FB_READ_131_SNVT_temp_ror**

```plaintext
FB_READ_131_SNVT_temp_ror
```

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_temp_ror.

**SNVT number:** 131.

**Description:** Value of the temperature change/increase (°C/minute).
**VAR_INPUT**

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

**VAR_OUTPUT**

rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

rValue: Min: -16384 / Max: 16383.5.

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

**VAR_IN_OUT**

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.2.127 FB_READ_132_SNVT_fire_init**

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_fire_init.

**SNVT number:** 132.

**Description:** Fire detector type (fire detector type names).

**VAR_INPUT**

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.
VAR_OUTPUT

eValue : E_LON_fire_initiator_t;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

**eValue**: Enum to be received (see \texttt{E\_LON\_fire\_initiator\_t}) [522].

**bNewData**: Becomes TRUE for 1 cycle when the block has received data.

**bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable \texttt{eError}.

**eError**: This output outputs an error code in the event of an error (see \texttt{E\_LON\_ERROR}) [477]. \texttt{bError} goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST\_LON\_Communication;

**stLON\_Com**: This structure is used to link \texttt{FB\_LON\_KL6401} [27] with the send/receive function (see \texttt{ST\_LON\_Communication}) [587].

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.128 \ FB\_READ\_133\_SNVT\_fire\_indcte

This function block receives the following LON input variable (nvi):

**SNVT Name**: SNVT\_fire\_indcte.

**SNVT number**: 133.

**Description**: Fire alarm display (fire alarm display names).

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

**wNVIndex**: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled**: TRUE = deselection of the block.

VAR_OUTPUT

eValue : E_LON_fire_indicator_t;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

**eValue**: Enum to be received (see \texttt{E\_LON\_fire\_indicator\_t}) [521].

**bNewData**: Becomes TRUE for 1 cycle when the block has received data.
bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT
stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.129 FB_READ_134_SNVT_time_zone

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_time_zone.

SNVT number: 134.

Description: Time zone description (offset, type, summer time start, summer time end).

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

stValue : ST_LON_SNVT_time_zone;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

stValue: Structure of the received data (see ST_LON_SNVT_time_zone [620]).

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;
**Programming**

stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.2.130 FB_READ_135_SNVT_earth_pos

This function block receives the following LON input variable (nvi):

- **SNVT Name:** SNVT_earth_pos.
- **SNVT number:** 135.
- **Description:** Position on Earth (width & length orientation, latitude & min, longitude & min, height).

#### VAR_INPUT

- **wNVIndex**: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **bDisabled**: TRUE = deselection of the block.

#### VAR_OUTPUT

- **stValue**: Structure of the received data (see ST_LON_SNVT_earth_pos [599]).
- **bNewData**: Becomes TRUE for 1 cycle when the block has received data.
- **bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.
- **eError**: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

#### VAR_IN_OUT

- **stLON_Com**: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.1.2.131 FB_READ_136_SNVT_reg_val

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_reg_val.

**SNVT number:** 136.

**Description:** Register value (raw value, bit code, number of decimal places).

**VAR_INPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>WORD</td>
<td>Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL := FALSE</td>
<td>TRUE = deselection of the block.</td>
</tr>
</tbody>
</table>

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stValue</td>
<td>ST_LON_SNVT_reg_val</td>
<td>Structure of the received data (see ST_LON_SNVT_reg_val[615]).</td>
</tr>
<tr>
<td>bNewData</td>
<td>BOOL</td>
<td>Becomes TRUE for 1 cycle when the block has received data.</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
<td>This output goes TRUE as soon as an error occurs. This error is described via the variable eError.</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
<td>This output outputs an error code in the event of an error (see E_LON_ERROR[477]). bError goes TRUE at the same time.</td>
</tr>
</tbody>
</table>

**VAR_IN_OUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stLON_Com</td>
<td>ST_LON_Communication</td>
<td>This structure is used to link FB_LON_KL6401()[27] with the send/receive function (see ST_LON_Communication[587]).</td>
</tr>
</tbody>
</table>

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.132 FB_READ_137_SNVT_reg_val_ts

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_reg_val.

**SNVT number:** 137.

**Description:** Register value (raw value, bit code, number of decimal places).

**VAR_INPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>WORD</td>
<td>Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL := FALSE</td>
<td>TRUE = deselection of the block.</td>
</tr>
</tbody>
</table>

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stValue</td>
<td>ST_LON_SNVT_reg_val</td>
<td>Structure of the received data (see ST_LON_SNVT_reg_val[615]).</td>
</tr>
<tr>
<td>bNewData</td>
<td>BOOL</td>
<td>Becomes TRUE for 1 cycle when the block has received data.</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
<td>This output goes TRUE as soon as an error occurs. This error is described via the variable eError.</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
<td>This output outputs an error code in the event of an error (see E_LON_ERROR[477]). bError goes TRUE at the same time.</td>
</tr>
</tbody>
</table>

**VAR_IN_OUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stLON_Com</td>
<td>ST_LON_Communication</td>
<td>This structure is used to link FB_LON_KL6401()[27] with the send/receive function (see ST_LON_Communication[587]).</td>
</tr>
</tbody>
</table>

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_reg_val_ts.

**SNVT number:** 137.

**Description:** Register value (raw value, bit code, number of decimal places, status, state, timestamp).

**VAR_INPUT**

```plaintext
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

```plaintext
stValue : ST_LON_SNVT_reg_val_ts;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**stValue:** Structure of the received data (see `ST_LON_SNVT_reg_val_ts`).

**bNewData:** Becomes TRUE for 1 cycle when the block has received data.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.

**eError:** This output outputs an error code in the event of an error (see `E_LON_ERROR`). `bError` goes TRUE at the same time.

**VAR_IN_OUT**

```plaintext
stLON_Com : ST_LON_Communication;
```

**stLON_Com:** This structure is used to link `FB_LON_KL6401()` with the send/receive function (see `ST_LON_Communication`).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.2.133  FB_READ_138_SNVT_volt_ac

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_volt_ac.

**SNVT number:** 138.

**Description:** Alternating voltage (Volt AC).
VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

uiValue : UINT;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

uiValue: Min: 0 / Max: 65535.

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.134 FB_READ_139_SNVT_amp_ac

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_amp_ac.

SNVT number: 139.

Description: Alternating current (ampere AC).

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.
VAR_OUTPUT

uiValue : UINT;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

uiValue: Min: 0 / Max: 65535.

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.135   FB_READ_143_SNVT_turbidity

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_turbidity.

SNVT number: 143.

Description: Turbidity (turbidity unit).

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

rValue: Min: 0 / Max: 65.535.

bNewData: Becomes TRUE for 1 cycle when the block has received data.
bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [ 477 ]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [ 27 ] with the send/receive function (see ST_LON_Communication [ 587 ]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.136  FB_READ_144_SNVT_turbidity_f

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_turbidity_f.

SNVT number: 144.

Description: Turbidity (turbidity unit).

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

rValue: Min: 0 / Max: 3.40E+38.

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [ 477 ]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;
Programming

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.137 FB_READ_145_SNVT_hvac_type

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_hvac_type.

**SNVT number:** 145.

**Description:** HVAC plant type (HVAC plant type description).

**VAR_INPUT**

- wNVIndex : WORD;
- bDisabled : BOOL := FALSE;

- **wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

- **bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

- eValue : E_LON_hvac_hvt_t;
- bNewData : BOOL;
- bError : BOOL;
- eError : E_LON_ERROR;

- **eValue:** Enum to be received (see E_LON_hvac_hvt_t [524]).

- **bNewData:** Becomes TRUE for 1 cycle when the block has received data.

- **bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

- **eError:** This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

**VAR_IN_OUT**

- stLON_Com : ST_LON_Communication;

- **stLON_Com:** This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.1.2.138  FB_READ_146_SNVT_elec_kwh_l

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_elec_kwh_l.

**SNVT number:** 146.

**Description:** Electric energy (kW / hour).

**VAR_INPUT**

| wNVIndex    : WORD; |
| bDisabled   : BOOL := FALSE; |

- **wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

| lrValue     : LREAL; |
| bNewData    : BOOL; |
| bError      : BOOL; |
| eError      : E_LON_ERROR; |

- **lrValue:** Min: -214748364.8 / Max: 214748364.7.
- **bNewData:** Becomes TRUE for 1 cycle when the block has received data.
- **bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.
- **eError:** This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

**VAR_IN_OUT**

| stLON_Com   : ST_LON_Communication; |

- **stLON_Com:** This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.139  FB_READ_147_SNVT_temp_diff_p
This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_temp_diff_p.

**SNVT number:** 147.

**Description:** Temperature difference (°C).

**VAR_INPUT**

```plaintext
wNVIndex    : WORD;
bDisabled   : BOOL := FALSE;
```

**wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

```plaintext
rValue      : REAL;
bNewData    : BOOL;
bError      : BOOL;
eError      : E_LON_ERROR;
```

**rValue:** Min: -327.68 / Max: 327.67.

**bNewData:** Becomes TRUE for 1 cycle when the block has received data.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

**eError:** This output outputs an error code in the event of an error (see `E_LON_ERROR [477]`). bError goes TRUE at the same time.

**VAR_IN_OUT**

```plaintext
stLON_Com   : ST_LON_Communication;
```

**stLON_Com:** This structure is used to link `FB_LON_KL6401()` [27] with the send/receive function (see `ST_LON_Communication [587]`).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.140  **FB_READ_148_SNVT_ctrl_req**

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_ctrl_req.

**SNVT number:** 148.

**Description:** Control request (receiver ID, sender ID, sender priority). Request for control permission for another controller.
VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

stValue : ST_LON_SNVT_ctrl_req;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

stValue: Structure of the received data (see ST_LON_SNVT_ctrl_req[596]).

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR[477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication[587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.141  FB_READ_149_SNVT_ctrl_resp

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_ctrl_resp.

SNVT number: 149.

Description: Control response (status, transmitter, controller ID). Response to a control request.

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.
VAR_OUTPUT

stValue : ST_LON_SNVT_ctrl_resp;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

stValue: Structure of the received data (see ST_LON_SNVT_ctrl_resp [597]).

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.142  FB_READ_150_SNVT_ptz

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_ptz.

SNVT number: 150.

Description: Camera PTZ (SNZ) (swivel, swivel speed, tilt, tilt speed, zoom, zoom speed).

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

stValue : ST_LON_SNVT_ptz;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

stValue: Structure of the received data (see ST_LON_SNVT_ptz [612]).

bNewData: Becomes TRUE for 1 cycle when the block has received data.
bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

**VAR_IN_OUT**

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.2.143 FB_READ_151_SNVT_privacyzone

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_privacyzone.

**SNVT number:** 151.

**Description:** Private zone (action area, zone number, camera ID). Certain areas can be excluded from the camera.

**VAR_INPUT**

<table>
<thead>
<tr>
<th>wNVIndex : WORD;</th>
</tr>
</thead>
<tbody>
<tr>
<td>bDisabled : BOOL := FALSE;</td>
</tr>
</tbody>
</table>

**wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>stValue : ST_LON_SNVT_privacyzone;</th>
</tr>
</thead>
<tbody>
<tr>
<td>bNewData : BOOL;</td>
</tr>
<tr>
<td>bError : BOOL;</td>
</tr>
<tr>
<td>eError : E_LON_ERROR;</td>
</tr>
</tbody>
</table>

**stValue:** Structure of the received data (see ST_LON_SNVT_privacyzone [611]).

**bNewData:** Becomes TRUE for 1 cycle when the block has received data.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

**eError:** This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

**VAR_IN_OUT**

stLON_Com : ST_LON_Communication;
stLON_Com: This structure is used to link FB_LON_KL6401() with the send/receive function (see ST_LON_Communication).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.144 FB_READ_152_SNVT_pos_ctrl

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_pos_ctrl.
SNVT number: 152.
Description: Position setting for cameras in networks (receiver, controller ID, controller priority, function, action, value).

VAR_INPUT

<table>
<thead>
<tr>
<th>wNVIndex</th>
<th>bDisabled</th>
<th>stValue</th>
<th>bNewData</th>
<th>eError</th>
</tr>
</thead>
</table>

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

<table>
<thead>
<tr>
<th>stValue</th>
<th>bNewData</th>
<th>bError</th>
<th>eError</th>
</tr>
</thead>
</table>

stValue: Structure of the received data (see ST_LON_SNVT_pos_ctrl).
bNewData: Becomes TRUE for 1 cycle when the block has received data.
bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.
eError: This output outputs an error code in the event of an error (see E_LON_ERROR). bError goes TRUE at the same time.

VAR_IN_OUT

<table>
<thead>
<tr>
<th>stLON_Com</th>
</tr>
</thead>
</table>

stLON_Com: This structure is used to link FB_LON_KL6401() with the send/receive function (see ST_LON_Communication).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
FB_READ_153_SNVT_enthalpy

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_enthalpy.

**SNVT number:** 153.

**Description:** Enthalpy (kilojoules/kg).

**VAR_INPUT**

- `wNVIndex` : WORD;
- `bDisabled` : BOOL := FALSE;

- **wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

- `rValue` : REAL;
- `bNewData` : BOOL;
- `bError` : BOOL;
- `eError` : E_LON_ERROR;

- **rValue:** Min: -327.68 / Max: 327.67.
- **bNewData:** Becomes TRUE for 1 cycle when the block has received data.
- **bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.
- **eError:** This output outputs an error code in the event of an error (see `E_LON_ERROR` [477]). bError goes TRUE at the same time.

**VAR_IN_OUT**

- `stLON_Com` : ST_LON_Comunication;

- **stLON_Com:** This structure is used to link `FB_LON_KL6401()` [27] with the send/receive function (see `ST_LON_Communication` [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

FB_READ_154_SNVT_gfci_status

**VAR_INPUT**

- `wNVIndex` : WORD;
- `bDisabled` : BOOL := FALSE;

- **wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

- `eValue` ;
- `bNewData` ;
- `bError` ;
- `eError` ;

- **eValue:**
- **bNewData:**
- **bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.
- **eError:** This output outputs an error code in the event of an error (see `E_LON_ERROR` [477]). bError goes TRUE at the same time.
This function block receives the following LON input variable (nvi):

**SNVT Name**: SNVT_gfci_status.

**SNVT number**: 154.

**Description**: Residual current circuit breaker status (residual current circuit breaker status text).

**VAR_INPUT**

<table>
<thead>
<tr>
<th>wNVIndex</th>
<th>WORD</th>
</tr>
</thead>
<tbody>
<tr>
<td>bDisabled</td>
<td>BOOL := FALSE;</td>
</tr>
</tbody>
</table>

**wNVIndex**: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled**: TRUE = deselection of the block.

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>eValue</th>
<th>E_LON_gfci_status_t</th>
</tr>
</thead>
<tbody>
<tr>
<td>bNewData</td>
<td>BOOL</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
</tr>
</tbody>
</table>

**eValue**: Enum to be received (see E_LON_gfci_status_t [\[524\]])

**bNewData**: Becomes TRUE for 1 cycle when the block has received data.

**bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

**eError**: This output outputs an error code in the event of an error (see E_LON_ERROR [\[477\]<br>]. bError goes TRUE at the same time.

**VAR_IN_OUT**

| stLON_Com | ST_LON_Communication; |

**stLON_Com**: This structure is used to link FB_LON_KL6401() [\[27\]] with the send/receive function (see ST_LON_Communication [\[587\]<br>]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.147 **FB_READ_155_SNVT_motor_state**

This function block receives the following LON input variable (nvi):

**SNVT Name**: SNVT_motor_state.

**SNVT number**: 155.

**Description**: Motor status (motor status designation).
VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

**wNVIndex**: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled**: TRUE = deselection of the block.

VAR_OUTPUT

eValue : E_LON_motor_state_t;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

eValue**: Enum to be received (see \[E_LON_motor_state_t\] [529]).

**bNewData**: Becomes TRUE for 1 cycle when the block has received data.

**bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable **eError**.

**eError**: This output outputs an error code in the event of an error (see \[E_LON_ERROR\] [477]). **bError** goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

**stLON_Com**: This structure is used to link \[FB_LON_KL6401()\] [27] with the send/receive function (see \[ST_LON_Communication\] [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.148 **FB_READ_156_SNVT_pumpset_mn**

This function block receives the following LON input variable (nvi):

**SNVT Name**: SNVT_pumpset_mn.

**SNVT number**: 156.

**Description**: Pump group (main pump, auxiliary pump, priority, ready, emergency off, main pump activated, auxiliary pump activated, maintenance request). Synchronized vacuum pumps.

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

**wNVIndex**: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled**: TRUE = deselection of the block.
VAR_OUTPUT

stValue : ST_LON_SNVT_pumpset_mn;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

stValue: Structure of the received data (see ST_LON_SNVT_pumpset_mn [613]).
bNewData: Becomes TRUE for 1 cycle when the block has received data.
bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.
eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.149 FB_READ_157_SNVT_ex_control

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_ex_control.

SNVT number: 157.

Description: Exclusive control (status, address). A device has exclusive control over another device.

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

stValue : ST_LON_SNVT_ex_control;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

stValue: Structure of the received data (see ST_LON_SNVT_ex_control [602]).
bNewData: Becomes TRUE for 1 cycle when the block has received data.
bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT
stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.150  FB_READ_158_SNVT_pumpset_sn

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_pumpset_sn.
SNVT number: 158.
Description: Pump group sensor (thinner, output, pressure, vacuum, ...).

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

stValue : ST_LON_SNVT_pumpset_sn;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

stValue: Structure of the received data (see ST_LON_SNVT_pumpset_sn [614]).
bNewData: Becomes TRUE for 1 cycle when the block has received data.
bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.
eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT
stLON_Com : ST_LON_Communication;
**Programming**

**stLON_Com**: This structure is used to link `FB_LON_KL6401()` with the send/receive function (see `ST_LON_Communication[587]`).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.151 **FB_READ_159_SNVT_pump_sensor**

This function block receives the following LON input variable (nvi):

**SNVT Name**: SNVT_pump_sensor.

**SNVT number**: 159.

**Description**: Pump sensor (speed, temperature, status).

**VAR_INPUT**

- `wNVIndex` : WORD;
- `bDisabled` : BOOL := FALSE;

- `wNVIndex`: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

- `bDisabled`: TRUE = deselection of the block.

**VAR_OUTPUT**

- `stValue` : `ST_LON_SNVT_pump_sensor`;
- `bNewData` : BOOL;
- `bError` : BOOL;
- `eError` : `E_LON_ERROR`;

- `stValue`: Structure of the received data (see `ST_LON_SNVT_pump_sensor[612]`).

- `bNewData`: Becomes TRUE for 1 cycle when the block has received data.

- `bError`: This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.

- `eError`: This output outputs an error code in the event of an error (see `E_LON_ERROR[477]`). `bError` goes TRUE at the same time.

**VAR_IN_OUT**

- `stLON_Com` : `ST_LON_Communication`;

- `stLON_Com`: This structure is used to link `FB_LON_KL6401()` with the send/receive function (see `ST_LON_Communication[587]`).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
This function block receives the following LON input variable (nvi):

**SNVT Name**: SNVT_abs_humid.

**SNVT number**: 160.

**Description**: Absolute humidity (gram/kg).

**VAR_INPUT**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.</td>
</tr>
<tr>
<td>bDisabled</td>
<td>TRUE = deselection of the block.</td>
</tr>
</tbody>
</table>

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rValue</td>
<td>Min: 0 / Max: 655.35.</td>
</tr>
<tr>
<td>bNewData</td>
<td>Becomes TRUE for 1 cycle when the block has received data.</td>
</tr>
<tr>
<td>bError</td>
<td>This output goes TRUE as soon as an error occurs. This error is described via the variable <code>eError</code>.</td>
</tr>
<tr>
<td>eError</td>
<td>This output outputs an error code in the event of an error (see <code>E_LON_ERROR</code>). bError goes TRUE at the same time.</td>
</tr>
</tbody>
</table>

**VAR_IN_OUT**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stLON_Com</td>
<td>This structure is used to link FB_LON_KL6401() with the send/receive function (see ST_LON_Communication).</td>
</tr>
</tbody>
</table>

**Requirements**

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Development environment</td>
<td>TwinCAT from v3.1.4020.14</td>
</tr>
<tr>
<td>required TC3 PLC library</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

---

### 4.1.2.153 FB_READ_161_SNVT_flow_p

This function block receives the following LON input variable (nvi):

**SNVT Name**: SNVT_flow_p.

**SNVT number**: 161.

**Description**: Flow (m³/h).

**VAR_INPUT**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.</td>
</tr>
<tr>
<td>bDisabled</td>
<td>TRUE = deselection of the block.</td>
</tr>
</tbody>
</table>

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rValue</td>
<td>Min: 0 / Max: 655.35.</td>
</tr>
<tr>
<td>bNewData</td>
<td>Becomes TRUE for 1 cycle when the block has received data.</td>
</tr>
<tr>
<td>bError</td>
<td>This output goes TRUE as soon as an error occurs. This error is described via the variable <code>eError</code>.</td>
</tr>
<tr>
<td>eError</td>
<td>This output outputs an error code in the event of an error (see <code>E_LON_ERROR</code>). bError goes TRUE at the same time.</td>
</tr>
</tbody>
</table>
This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_flow_p.

**SNVT number:** 161.

**Description:** Volume flow (cubic meters / hour).

**VAR_INPUT**

| wNVIndex : WORD; |
| bDisabled : BOOL := FALSE; |

**wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

| rValue : REAL; |
| bNewData : BOOL; |
| bError : BOOL; |
| eError : E_LON_ERROR; |

**rValue:** Min: 0 / Max: 655.35.

**bNewData:** Becomes TRUE for 1 cycle when the block has received data.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

**eError:** This output outputs an error code in the event of an error (see [E_LON_ERROR](#)). bError goes TRUE at the same time.

**VAR_IN_OUT**

| stLON_Com : ST_LON_Communication; |

**stLON_Com:** This structure is used to link [FB_LON_KL6401](#) with the send/receive function (see [ST_LON_Communication](#)).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.2.154 ** **FB_READ_162_SNVT_dev_c_mode**

- wNVIndex
- eValue
- bDisabled
- bNewData
- stLON_Com
- bError
- eError

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_dev_c_mode.

**SNVT number:** 162.

**Description:** Device operating mode (device operating mode states).
VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

eValue : E_LON_device_c_mode_t;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

eValue: Enum to be received (see E_LON_device_c_mode_t [515]).

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.155 FB_READ_163_SNVT_valve_mode

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_valve_mode.

SNVT number: 163.

Description: Valve state.

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.
**VAR_OUTPUT**

**eValue:** Enum to be received (see `E_LON_valve_mode_t` [550]).

**bNewData:** Becomes TRUE for 1 cycle when the block has received data.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.

**eError:** This output outputs an error code in the event of an error (see `E_LON_ERROR` [477]). `bError` goes TRUE at the same time.

**VAR_IN_OUT**

**stLON_Com:** This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see `ST_LON_Communication` [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.2.156 FB_READ_164_SNVT_alarm_2**

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_alarm_2.

**SNVT number:** 164.

**Description:** Alarm status 2. Signals the alarm status of a function block or device. Replaces SNVT_alarm.

**VAR_INPUT**

**wNVIndex** : WORD;
**bDisabled** : BOOL := FALSE;

**wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

**stValue** : `ST_LON_SNVT_alarm_2`;
**bNewData** : BOOL;
**bError** : BOOL;
**eError** : `E_LON_ERROR`;

**stValue:** Structure of the received data (see `ST_LON_SNVT_alarm_2` [592]).

**bNewData:** Becomes TRUE for 1 cycle when the block has received data.
**bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.

**eError**: This output outputs an error code in the event of an error (see `E_LON_ERROR`). `bError` goes TRUE at the same time.

**VAR_IN_OUT**

```delphi
stLON_Com : ST_LON_Communication;
```

`stLON_Com`: This structure is used to link `FB_LON_KL6401()` with the send/receive function (see `ST_LON_Communication`).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.2.157 FB_READ_165_SNVT_state_64**

This function block receives the following LON input variable (`nvi`):

- **SNVT Name**: SNVT_state_64.
- **SNVT number**: 165.
- **Description**: Status information (64 individual bit values). Each status is a Boolean value.

**VAR_INPUT**

```delphi
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

- **wNVIndex**: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **bDisabled**: TRUE = deselection of the block.

**VAR_OUTPUT**

```delphi
arrValue : ARRAY [0..63] OF BOOL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

- **arrValue**: 0-63 bit.
- **bNewData**: Becomes TRUE for 1 cycle when the block has received data.
- **bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.
- **eError**: This output outputs an error code in the event of an error (see `E_LON_ERROR`). `bError` goes TRUE at the same time.

**VAR_IN_OUT**

```delphi
stLON_Com : ST_LON_Communication;
```
stLON_Com: This structure is used to link FB_LON_KL6401() with the send/receive function (see ST_LON_Communication).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.158 FB_READ_166_SNVT_nv_type

This function block receives the following LON input variable (nvi):

- **SNVT Name**: SNVT_nv_type.
- **SNVT number**: 166.
- **Description**: Network variable type. Type description for network variables.

### VAR_INPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>WORD</td>
<td>Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL</td>
<td>TRUE = deselection of the block.</td>
</tr>
</tbody>
</table>

### VAR_OUTPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stValue</td>
<td>ST_LON_SNVT_nv_type</td>
<td>Structure of the received data (see ST_LON_SNVT_nv_type).</td>
</tr>
<tr>
<td>bNewData</td>
<td>BOOL</td>
<td>Becomes TRUE for 1 cycle when the block has received data.</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
<td>This output goes TRUE as soon as an error occurs. This error is described via the variable eError.</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
<td>This output outputs an error code in the event of an error (see E_LON_ERROR). bError goes TRUE at the same time.</td>
</tr>
</tbody>
</table>

### VAR_IN_OUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>stLON_Com</td>
<td>ST_LON_Communication</td>
</tr>
</tbody>
</table>

stLON_Com: This structure is used to link FB_LON_KL6401() with the send/receive function (see ST_LON_Communication).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
## 4.1.2.159 FB_READ_168_SNVT_ent_opmode

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_ent_opmode.

**SNVT number:** 168.

**Description:** Operating mode of access objects (doors, locks or objects permitting or prohibiting access).

### VAR_INPUT

```plaintext
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

- **wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **bDisabled:** TRUE = deselection of the block.

### VAR_OUTPUT

```plaintext
eValue : E_LON_ent_opmode_cmd_t;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

- **eValue:** Enum to be received (see `E_LON_ent_opmode_cmd_t`[518]).
- **bNewData:** Becomes TRUE for 1 cycle when the block has received data.
- **bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.
- **eError:** This output outputs an error code in the event of an error (see `E_LON_ERROR`[477]). `bError` goes TRUE at the same time.

### VAR_IN_OUT

```plaintext
stLON_Com : ST_LON_Communication;
```

- **stLON_Com:** This structure is used to link `FB_LON_KL6401()`[27] with the send/receive function (see `ST_LON_Communication`[587]).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

## 4.1.2.160 FB_READ_169_SNVT_ent_state

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_ent_opmode.

**SNVT number:** 169.

**Description:** Operating mode of access objects (doors, locks or objects permitting or prohibiting access).

### VAR_INPUT

```plaintext
wNVIndex : E_LON_ent_opmode_cmd_t;
```

- **wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

### VAR_OUTPUT

```plaintext
eValue : E_LON_ent_opmode_cmd_t;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

- **eValue:** Enum to be received (see `E_LON_ent_opmode_cmd_t`[518]).
- **bNewData:** Becomes TRUE for 1 cycle when the block has received data.
- **bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.
- **eError:** This output outputs an error code in the event of an error (see `E_LON_ERROR`[477]). `bError` goes TRUE at the same time.
This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_ent_state.

**SNVT number:** 169.

**Description:** State of access objects (doors, locks or objects permitting or prohibiting access).

### VAR_INPUT

```plaintext
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

- **wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **bDisabled:** TRUE = deselection of the block.

### VAR_OUTPUT

```plaintext
eValue : E_LON_ent_cmd_t;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

- **eValue:** Enum to be received (see `E_LON_ent_cmd_t` [517]).
- **bNewData:** Becomes TRUE for 1 cycle when the block has received data.
- **bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.
- **eError:** This output outputs an error code in the event of an error (see `E_LON_ERROR` [477]). `bError` goes TRUE at the same time.

### VAR_IN_OUT

```plaintext
stLON_Com : ST_LON_Communication;
```

- **stLON_Com:** This structure is used to link `FB_LON_KL6401()` [27] with the send/receive function (see `ST_LON_Communication` [587]).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.2.161 FB_READ_170_SNVT_ent_status

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_ent_status.

**SNVT number:** 170.

**Description:** Status of access objects (doors, locks or objects permitting or prohibiting access).
**VAR_INPUT**

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

**wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

stValue : ST_LON_SNVT_ent_status;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

**stValue:** Structure of the received data (see ST_LON_SNVT_ent_status[600]).

**bNewData:** Becomes TRUE for 1 cycle when the block has received data.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable **eError**.

**eError:** This output outputs an error code in the event of an error (see E_LON_ERROR[477]). **bError** goes TRUE at the same time.

**VAR_IN_OUT**

stLON_Com : ST_LON_Communication;

**stLON_Com :** This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.2.162   FB_READ_171_SNVT_flow_dir**

<table>
<thead>
<tr>
<th>FB_READ_171_SNVT_flow_dir</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
</tr>
<tr>
<td>bDisabled</td>
</tr>
<tr>
<td>stLON_Com</td>
</tr>
<tr>
<td>eValue</td>
</tr>
<tr>
<td>bNewData</td>
</tr>
<tr>
<td>bError</td>
</tr>
<tr>
<td>eError</td>
</tr>
</tbody>
</table>

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_flow_dir.

**SNVT number:** 171.

**Description:** Flow direction. Direction of the flow to be permitted or direction of the current flow.

**VAR_INPUT**

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

**wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled:** TRUE = deselection of the block.
**VAR_OUTPUT**

- `eValue` : Enum to be received (see `E_LON_flow_direction_t` [523]).
- `bNewData` : Becomes TRUE for 1 cycle when the block has received data.
- `bError` : This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.
- `eError` : This output outputs an error code in the event of an error (see `E_LON_ERROR` [477]). `bError` goes TRUE at the same time.

**VAR_IN_OUT**

- `stLON_Com` : This structure is used to link `FB_LON_KL6401()` with the send/receive function (see `ST_LON_Communication` [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.2.163 FB_READ_172_SNVT_hvac_satsts**

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_hvac_satsts.

**SNVT number:** 172.

**Description:** HVAC saturation status.

0 in a field means that plant (device) linked to the field is not saturated or does not reach the limit stop before the required set value is reached.

1 in a field means that plant (device) linked to the field is saturated or reaches the limit stop before the required set value is not reached.

**VAR_INPUT**

- `wNVIndex` : WORD;
- `bDisabled` : BOOL := FALSE;

**wNVIndex** : Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled** : TRUE = deselection of the block.
VAR_OUTPUT

stValue : ST_LON_SNVT_hvac_satsts;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

stValue: Structure of the received data (see ST_LON_SNVT_hvac_satsts [604]).
bNewData: Becomes TRUE for 1 cycle when the block has received data.
bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.
eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.164  FB_READ_173_SNVT_dev_status

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_dev_status.
SNVT number: 173.
Description: Device status.

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

stValue : ST_LON_SNVT_dev_status;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

stValue: Structure of the received data (see ST_LON_SNVT_dev_status [599]).
bNewData: Becomes TRUE for 1 cycle when the block has received data.
bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.165 FB_READ_174_SNVT_dev_fault

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_dev_fault.

SNVT number: 174.

Description: Error state. Error information for a device.

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

stValue : ST_LON_SNVT_dev_fault;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

stValue: Structure of the received data (see ST_LON_SNVT_dev_fault [598]).

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;
stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.166 FB_READ_175_SNVT_dev_maint

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_dev_maint.

SNVT number: 175.

Description: Device maintenance station.

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

stValue : ST_LON_SNVT_dev_maint;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

stValue: Structure of the received data (see ST_LON_SNVT_dev_maint [598]).

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_date_event.

**SNVT number:** 176.

**Description:** Event status.

### VAR_INPUT

```plaintext
VAR_INPUT
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled:** TRUE = deselection of the block.

### VAR_OUTPUT

```plaintext
VAR_OUTPUT
stValue : ST_LON_SNVT_date_event;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**stValue:** Structure of the received data (see ST_LON_SNVT_date_event [p. 598]).

**bNewData:** Becomes TRUE for 1 cycle when the block has received data.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.

**eError:** This output outputs an error code in the event of an error (see E_LON_ERROR [p. 477]). bError goes TRUE at the same time.

### VAR_IN_OUT

```plaintext
VAR_IN_OUT
stLON_Com : ST_LON_Communication;
```

**stLON_Com:** This structure is used to link FB_LON_KL6401() [p. 27] with the send/receive function (see ST_LON_Communication [p. 587]).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.2.168 FB_READ_177_SNVT_sched_val
This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_sched_val.

**SNVT number:** 177.

**Description:** Calendar value. Index of a calendar value that selects an entry in an SCPT value definition field array or is a direct value output.

**VAR_INPUT**

| wNVIndex : WORD; |
| bDisabled : BOOL := FALSE; |

**wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

| byValue : BYTE; |
| bNewData : BOOL; |
| bError : BOOL; |
| eError : E_LON_ERROR; |

**byValue:** Min: 0 / Max: 255.

**bNewData:** Becomes TRUE for 1 cycle when the block has received data.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

**eError:** This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

**VAR_IN_OUT**

| stLON_Com : ST_LON_Communication; |

**stLON_Com:** This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.2.169 FB_READ_180_SNVT_sblnd_state**

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_sblnd_state.

**SNVT number:** 180.

**Description:** Blind status.
VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

VAR OUTPUT

stValue : ST_LON_SNVT_sblnd_state;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

stValue: Structure of the received data (see ST_LON_SNVT_sblnd_state [616]).

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR[477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.170 FB_READ_181_SNVT_rac_ctrl

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_rac_ctrl.

SNVT number: 181.

Description: Sound function control. Activates the sound function for a particular source.

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.
VAR_OUTPUT

stValue : ST_LON_SNVT_rac_ctrl;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

stValue: Structure of the received data (see ST_LON_SNVT_rac_ctrl [614]).
bNewData: Becomes TRUE for 1 cycle when the block has received data.
bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.
eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.171    FB_READ_182_SNVT_rac_req

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_rac_req.

SNVT number: 182.

Description: Sound function request. Requests the sound function for a particular source.

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

stValue : ST_LON_SNVT_rac_req;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

stValue: Structure of the received data (see ST_LON_SNVT_rac_req [615]).
bNewData: Becomes TRUE for 1 cycle when the block has received data.
**bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.

**eError**: This output outputs an error code in the event of an error (see `E_LON_ERROR [477]`). `bError` goes TRUE at the same time.

### VAR_IN_OUT

```pascal
stLON_Com : ST_LON_Communication;
```

**stLON_Com**: This structure is used to link `FB_LON_KL6401()` with the send/receive function (see `ST_LON_Communication [587]`).

#### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.2.172  **FB_READ_183_SNVT_count_32**

This function block receives the following LON input variable (nvi):

**SNVT Name**: SNVT_count_32.

**SNVT number**: 183.

**Description**: Absolute counter. 32 bit counter.

#### VAR_INPUT

```pascal
wNVIndex  : WORD;
bDisabled : BOOL := FALSE;
```

- **wNVIndex**: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **bDisabled**: TRUE = deselection of the block.

#### VAR_OUTPUT

```pascal
udiValue  : UDINT;
bNewData  : BOOL;
bError    : BOOL;
eError    : E_LON_ERROR;
```

- **udiValue**: Min: 0 / Max: 4294967294.
- **bNewData**: Becomes TRUE for 1 cycle when the block has received data.
- **bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.
- **eError**: This output outputs an error code in the event of an error (see `E_LON_ERROR [477]`). `bError` goes TRUE at the same time.

### VAR_IN_OUT

```pascal
stLON_Com : ST_LON_Communication;
```
stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.173  FB_READ_184_SNVT_clothes_w_c

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_clothes_w_c.

**SNVT number:** 184.

**Description:** Washing machine / commands. For programming and starting a washing machine.

**VAR_INPUT**

```plaintext
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

- **wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

```plaintext
stValue : ST_LON_SNVT_clothes_w_c;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

- **stValue:** Structure of the received data (see ST_LON_SNVT_clothes_w_c [595]).
- **bNewData:** Becomes TRUE for 1 cycle when the block has received data.
- **bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable **eError**.
- **eError:** This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

**VAR_IN_OUT**

```plaintext
stLON_Com : ST_LON_Communication;
```

- **stLON_Com:** This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_clothes_w_m.

**SNVT number:** 185.

**Description:** Washing machine /management status Current status of door / lid and drain.

### VAR_INPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>WORD</td>
<td>Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL</td>
<td>TRUE = deselection of the block.</td>
</tr>
</tbody>
</table>

### VAR_OUTPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stValue</td>
<td>ST_LON_SNVT_clothes_w_m</td>
<td>Structure of the received data (see ST_LON_SNVT_clothes_w_m).</td>
</tr>
<tr>
<td>bNewData</td>
<td>BOOL</td>
<td>Becomes TRUE for 1 cycle when the block has received data.</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
<td>This output goes TRUE as soon as an error occurs. This error is described via the variable eError.</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
<td>This output outputs an error code in the event of an error (see E_LON_ERROR). bError goes TRUE at the same time.</td>
</tr>
</tbody>
</table>

### VAR_IN_OUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stLON_Com</td>
<td>ST_LON_Communication</td>
<td>This structure is used to link FB_LON_KL6401() with the send/receive function (see ST_LON_Communication).</td>
</tr>
</tbody>
</table>

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.2.175  FB_READ_186_SNVT_clothes_w_s

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_clothes_w_s.

**SNVT number:** 186.

**Description:** Washing machine /management status Current status of door / lid and drain.

### VAR_INPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>WORD</td>
<td></td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL</td>
<td></td>
</tr>
<tr>
<td>stLON_Com</td>
<td>ST_LON_Communication</td>
<td></td>
</tr>
</tbody>
</table>
This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_clothes_w_s.

**SNVT number:** 186.

**Description:** Washing machine / status. Current state of a washing machine, including command and alarm information.

**VAR_INPUT**

```plaintext
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

```plaintext
stValue : ST_LON_SNVT_clothes_w_s;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**stValue:** Structure of the received data (see `ST_LON_SNVT_clothes_w_s`).

**bNewData:** Becomes TRUE for 1 cycle when the block has received data.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.

**eError:** This output outputs an error code in the event of an error (see `E_LON_ERROR`). `bError` goes TRUE at the same time.

**VAR_IN_OUT**

```plaintext
stLON_Com : ST_LON_Communication;
```

**stLON_Com:** This structure is used to link `FB_LON_KL6401()` with the send/receive function (see `ST_LON_Communication`).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.2.176 FB_READ_187_SNVT_clothes_w_a

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_clothes_w_a.

**SNVT number:** 187.

**Description:** Washing machine / alarm messages.
VAR_INPUT

<table>
<thead>
<tr>
<th>wNVIndex</th>
<th>WORD</th>
</tr>
</thead>
<tbody>
<tr>
<td>bDisabled</td>
<td>BOOLEAN</td>
</tr>
</tbody>
</table>

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

<table>
<thead>
<tr>
<th>stValue</th>
<th>ST_LON_SNVT_clothes_w_a</th>
</tr>
</thead>
<tbody>
<tr>
<td>bNewData</td>
<td>BOOLEAN</td>
</tr>
<tr>
<td>bError</td>
<td>BOOLEAN</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
</tr>
</tbody>
</table>

stValue: Structure of the received data (see ST_LON_SNVT_clothes_w_a [593]).

bNewData: Becomes TRUE for 1 cycle when the block has received data.

eError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

VAR_IN_OUT

| stLON_Com | ST_LON_Communication |

stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.177 FB_READ_188_SNVT_multiplier_s

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_multiplier_s.

SNVT number: 188.

Description: Multiplier.

VAR_INPUT

<table>
<thead>
<tr>
<th>wNVIndex</th>
<th>WORD</th>
</tr>
</thead>
<tbody>
<tr>
<td>bDisabled</td>
<td>BOOLEAN</td>
</tr>
</tbody>
</table>

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.
### VAR_OUTPUT

- **rValue**: REAL;  
  - Min: 0 / Max: 2.54.
- **bNewData**: BOO;  
  - Becomes TRUE for 1 cycle when the block has received data.
- **bError**: BOO;  
  - This output goes TRUE as soon as an error occurs. This error is described via the variable eError.
- **eError**: E_LON_ERROR;  
  - This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

### VAR_IN_OUT

- **stLON_Com**: ST_LON_Communication;  
  - This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.2.178 FB_READ_189_SNVT_switch_2

#### Function Block Description

- **wNVIndex**: WORD;  
  - Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **bDisabled**: BOOL := FALSE;  
  - TRUE = deselection of the block.

#### VAR_INPUT

- **stValue**: ST_LON_SNVT_switch_2;  
  - Structure of the received data (see ST_LON_SNVT_switch_2 [619]).
- **bNewData**: BOOL;  
  - Becomes TRUE for 1 cycle when the block has received data.

### FB_READ_189_SNVT_switch_2

#### Parameters

- **wNVIndex**
- **bDisabled**
- **stLON_Com**
- **sValue**
- **bNewData**
- **bError**
- **eError**

This function block receives the following LON input variable (nvi):

- **SNVT Name**: SNVT_switch_2.
- **SNVT number**: 189.
- **Description**: Switch for setting scenes and settings. Extended version of the SNVT_switch for setting of scenes and settings similar to the SNVT_scene and SNVT_setting.
bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.179 FB_READ_190_SNVT_color_2

This function block receives the following LON input variable (nvi):

SNVT Name: SNVT_color_2.

SNVT number: 190.

Description: Color.

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

stValue : ST_LON_SNVT_color_2;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

stValue: Structure of the received data (see ST_LON_SNVT_color_2 [596]).

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;
stLON_Com: This structure is used to link FB_LON_KL6401() (27) with the send/receive function (see ST_LON_Communication (587)).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.2.180 FB_READ_191_SNVT_log_status

```plaintext
VAR_INPUT

- wNVIndex : WORD;
- bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

- stValue : ST_LON_SNVT_log_status;
- bNewData : BOOL;
- bError : BOOL;
- eError : E_LON_ERROR;

stValue: Structure of the received data (see ST_LON_SNVT_log_status (607)).

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR (477)). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com: This structure is used to link FB_LON_KL6401() (27) with the send/receive function (see ST_LON_Communication (587)).
```

SNVT Name: SNVT_log_status.

SNVT number: 191

Description: Log status (hundredth of a second). Represents the current status of a data logging. Update based on the cpLogNotificationThreshold value. Displays only the status. Alarms are output via the node object nvoAlarm2. Is required if the node object contains no nvoLogStat output.
**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.2.181 FB_READ_192_SNVT_time_stamp_p

This function block receives the following LON input variable (nvi):

**SNVT Name**: SNVT_time_stamp_p.

**SNVT number**: 192.

**Description**: Precise timestamp (seconds). Timestamp with a resolution of a hundredth second.

**VAR_INPUT**

<table>
<thead>
<tr>
<th>wNVIndex</th>
<th>: WORD;</th>
</tr>
</thead>
<tbody>
<tr>
<td>bDisabled</td>
<td>: BOOL := FALSE;</td>
</tr>
</tbody>
</table>

**wNVIndex**: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled**: TRUE = deselection of the block.

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>stValue</th>
<th>: TIMESTRUCT;</th>
</tr>
</thead>
<tbody>
<tr>
<td>bNewData</td>
<td>: BOOL;</td>
</tr>
<tr>
<td>bError</td>
<td>: BOOL;</td>
</tr>
<tr>
<td>eError</td>
<td>: E_LON_ERROR;</td>
</tr>
</tbody>
</table>

**stValue**: Structure of the received data (see TIMESTRUCT).

**bNewData**: Becomes TRUE for 1 cycle when the block has received data.

**bError**:

This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.

**eError**:

This output outputs an error code in the event of an error (see `E_LON_ERROR [477]`). bError goes TRUE at the same time.

**VAR_IN_OUT**

<table>
<thead>
<tr>
<th>stLON_Com</th>
<th>: ST_LON_Communication;</th>
</tr>
</thead>
</table>

**stLON_Com**: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
FB_READ_193_SNVT_log_fx_request

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_log_fx_request.

**SNVT number:** 193.

**Description:** Log file transfer request. Requests a data log via FTP transfer. Must follow a standard ftp request to obtain a data log file. Required on devices that use the data logger function profile, which enables data log transfer via FTP.

**VAR_INPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.</td>
</tr>
<tr>
<td>bDisabled</td>
<td>TRUE = deselection of the block.</td>
</tr>
</tbody>
</table>

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stValue</td>
<td>Structure of the received data (see ST_LON_SNVT_log_fx_request [606]).</td>
</tr>
<tr>
<td>bNewData</td>
<td>Becomes TRUE for 1 cycle when the block has received data.</td>
</tr>
<tr>
<td>bError</td>
<td>This output goes TRUE as soon as an error occurs. This error is described via the variable eError.</td>
</tr>
<tr>
<td>eError</td>
<td>This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.</td>
</tr>
</tbody>
</table>

**VAR_IN_OUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stLON_Com</td>
<td>This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).</td>
</tr>
</tbody>
</table>

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.1.2.183  **FB_READ_194_SNVT_log_fx_status**

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_log_fx_status.

**SNVT number:** 194.

**Description:** Log file transfer status. Indicates the status of a data log via FTP transfer. Required on devices that use the data logger function profile, which enables data log transfer via FTP.

**VAR_INPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>WORD</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL := FALSE</td>
</tr>
</tbody>
</table>

**wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>stValue</td>
<td>ST_LON_SNVT_log_fx_status</td>
</tr>
<tr>
<td>bNewData</td>
<td>BOOL</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
</tr>
</tbody>
</table>

**stValue:** Structure of the received data (see ST_LON_SNVT_log_fx_status [607]).

**bNewData:** Becomes TRUE for 1 cycle when the block has received data.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

**eError:** This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

**VAR_IN_OUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>stLON_Com</td>
<td>ST_LON_Communication</td>
</tr>
</tbody>
</table>

**stLON_Com:** This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
FB_READ_195_SNVT_log_request

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_log_request.

**SNVT number:** 195.

**Description:** Log status request. Requests the current status of a data log. The status is output via the SNVT_log_status output.

**VAR_INPUT**

- wNVIndex : WORD;
- bDisabled : BOOL := FALSE;

**wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

- uiValue : UINT;
- bNewData : BOOL;
- bError : BOOL;
- eError : E_LON_ERROR;

**uiValue:** Min: 0 / Max: 65535.

**bNewData:** Becomes TRUE for 1 cycle when the block has received data.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

**eError:** This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

**VAR_IN_OUT**

- stLON_Com : ST_LON_Communication;

**stLON_Com:** This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.1.2.185  FB_READ_196_SNVT_enthalpy_d

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_enthalpy_d.

**SNVT number:** 196.

**Description:** Enthalpy difference (kJ/kg).

**VAR_INPUT**

<table>
<thead>
<tr>
<th>wNVIndex</th>
<th>WORD</th>
</tr>
</thead>
<tbody>
<tr>
<td>bDisabled</td>
<td>BOOL := FALSE</td>
</tr>
</tbody>
</table>

**wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>rValue</th>
<th>REAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>bNewData</td>
<td>BOOL</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
</tr>
</tbody>
</table>

**rValue:** Min: -327.68 / Max: 327.66.

**bNewData:** Becomes TRUE for 1 cycle when the block has received data.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable **eError**.

**eError:** This output outputs an error code in the event of an error (see **E_LON_ERROR** [477]). bError goes TRUE at the same time.

**VAR_IN_OUT**

| stLON_Com         | ST_LON_Communication |

**stLON_Com:** This structure is used to link **FB_LON_KL6401()** [27] with the send/receive function (see **ST_LON_Communication** [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

Application

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_enthalpy_d.

**SNVT number:** 196.

**Description:** Enthalpy difference (kJ/kg).

**VAR_INPUT**

<table>
<thead>
<tr>
<th>wNVIndex</th>
<th>WORD</th>
</tr>
</thead>
<tbody>
<tr>
<td>bDisabled</td>
<td>BOOL := FALSE</td>
</tr>
</tbody>
</table>

**wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>rValue</th>
<th>REAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>bNewData</td>
<td>BOOL</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
</tr>
</tbody>
</table>

**rValue:** Min: -327.68 / Max: 327.66.

**bNewData:** Becomes TRUE for 1 cycle when the block has received data.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable **eError**.

**eError:** This output outputs an error code in the event of an error (see **E_LON_ERROR** [477]). bError goes TRUE at the same time.

**VAR_IN_OUT**

| stLON_Com         | ST_LON_Communication |

**stLON_Com:** This structure is used to link **FB_LON_KL6401()** [27] with the send/receive function (see **ST_LON_Communication** [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.1.2.186   FB_READ_197_SNVT_amp_ac_mil

Application
This function block receives the following LON input variable (nvi):

- **SNVT Name**: SNVT_amp_ac_mil.
- **SNVT number**: 197.
- **Description**: Electric current (milliampere).

**VAR_INPUT**

- **wNVIndex**: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **bDisabled**: TRUE = deselection of the block.

**VAR_OUTPUT**

- **uiValue**: Min: 0 / Max: 65535.
- **bNewData**: Becomes TRUE for 1 cycle when the block has received data.
- **bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable **eError**.
- **eError**: This output outputs an error code in the event of an error (see **E_LON_ERROR** [477]). bError goes TRUE at the same time.

**VAR_IN_OUT**

- **stLON_Com**: ST_LON_Communication;

- **stLON_Com**: This structure is used to link **FB_LON_KL6401()** [27] with the send/receive function (see **ST_LON_Communication** [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_time_hour_p.

**SNVT number:** 198.

**Description:** Time in hours.

### VAR_INPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>WORD</td>
<td>Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL</td>
<td>TRUE = deselection of the block.</td>
</tr>
</tbody>
</table>

### VAR_OUTPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>udiValue</td>
<td>UDINT</td>
<td>Min: 0 / Max: 4294967294.</td>
</tr>
<tr>
<td>bNewData</td>
<td>BOOL</td>
<td>Becomes TRUE for 1 cycle when the block has received data.</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
<td>This output goes TRUE as soon as an error occurs. This error is described via the variable eError.</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
<td>This output outputs an error code in the event of an error (see E_LON_ERROR[477]). bError goes TRUE at the same time.</td>
</tr>
</tbody>
</table>

### VAR_IN_OUT

<table>
<thead>
<tr>
<th>Structure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stLON_Com</td>
<td>This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).</td>
</tr>
</tbody>
</table>

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.2.188  FB_READ_199_SNVT_lamp_status

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_lamp_status.

**SNVT number:** 4.

**Description:** Lamp status.

### VAR_INPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>WORD</td>
<td></td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL</td>
<td></td>
</tr>
</tbody>
</table>

### VAR_OUTPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sValue</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bNewData</td>
<td>BOOL</td>
<td></td>
</tr>
</tbody>
</table>

### VAR_IN_OUT

<table>
<thead>
<tr>
<th>Structure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stLON_Com</td>
<td></td>
</tr>
</tbody>
</table>
This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_lamp_status.

**SNVT number:** 199.

**Description:** Lamp status.

**VAR_INPUT**

```plaintext
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

```plaintext
stValue : ST_LON_SNVT_lamp_status;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**stValue:** Structure of the received data (see ST_LON_SNVT_lamp_status [606]).

**bNewData:** Becomes TRUE for 1 cycle when the block has received data.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

**eError:** This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

**VAR_IN_OUT**

```plaintext
stLON_Com : ST_LON_Communication;
```

**stLON_Com** : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.2.189 FB_READ_200_SNVT_environment**

This function block receives the following LON input variable (nvi):

**SNVT Name:** SNVT_environment.

**SNVT number:** 200.

**Description:** Environment.
VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

**wNVIndex**: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled**: TRUE = deselection of the block.

VAR_OUTPUT

stValue : ST_LON_SNVT_environment;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

**stValue**: Structure of the received data (see `ST_LON_SNVT_environment`).

**bNewData**: Becomes TRUE for 1 cycle when the block has received data.

**bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.

**eError**: This output outputs an error code in the event of an error (see `E_LON_ERROR`). `bError` goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

**stLON_Com**: This structure is used to link `FB_LON_KL6401` with the send/receive function (see `ST_LON_Communication`).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.2.190  **FB_READ_201_SNVT_geo_loc**

This function block receives the following LON input variable (nvi):

**SNVT Name**: SNVT_geo_loc.

**SNVT number**: 201.

**Description**: Geographical location.

VAR_INPUT

wNVIndex : WORD;
bDisabled : BOOL := FALSE;

**wNVIndex**: Unique index. This is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**bDisabled**: TRUE = deselection of the block.
VAR_OUTPUT

stValue : ST_LON_SNVT_geo_loc;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;

stValue: Structure of the received data (see ST_LON_SNVT_geo_loc [604]).

bNewData: Becomes TRUE for 1 cycle when the block has received data.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3 Send

<table>
<thead>
<tr>
<th>POUs</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FB_SEND_001_SNVT_amp [202]</td>
<td>Electric current (Amperes)</td>
</tr>
<tr>
<td>FB_SEND_002_SNVT_amp_mil [203]</td>
<td>Electric current (milliAmperes)</td>
</tr>
<tr>
<td>FB_SEND_003_SNVT_angle [204]</td>
<td>Angular distance (radians)</td>
</tr>
<tr>
<td>FB_SEND_004_SNVT_angle_vel [206]</td>
<td>Angular velocity (radians/second)</td>
</tr>
<tr>
<td>FB_SEND_005_SNVT_btu_kilo [207]</td>
<td>Thermal energy (kilo-Btus)</td>
</tr>
<tr>
<td>FB_SEND_006_SNVT_btu_mega [209]</td>
<td>Thermal energy (mega-Btus)</td>
</tr>
<tr>
<td>FB_SEND_007_SNVT_char_ascii [210]</td>
<td>ASCII character (8-bit ASCII character)</td>
</tr>
<tr>
<td>FB_SEND_008_SNVT_count [211]</td>
<td>Absolute count (units)</td>
</tr>
<tr>
<td>FB_SEND_009_SNVT_count_inc [213]</td>
<td>Increment count (units (delta))</td>
</tr>
<tr>
<td>FB_SEND_011_SNVT_date_day [214]</td>
<td>Day of week (day names)</td>
</tr>
<tr>
<td>FB_SEND_013_SNVT_elec_kwh [215]</td>
<td>Electric energy (kiloWatt-hours)</td>
</tr>
<tr>
<td>FB_SEND_014_SNVT_elec_whr [217]</td>
<td>Electric energy (Watt-hours)</td>
</tr>
<tr>
<td>FB_SEND_015_SNVT_flow [218]</td>
<td>Flow volume (liters/second)</td>
</tr>
<tr>
<td>FB_SEND_017_SNVT_length [221]</td>
<td>Length (meters)</td>
</tr>
<tr>
<td>FB_SEND_018_SNVT_length_kilo [222]</td>
<td>Length (kilometers)</td>
</tr>
<tr>
<td>FB_SEND_019_SNVT_length_micr [224]</td>
<td>Length (micrometers (microns))</td>
</tr>
<tr>
<td>FB_SEND_020_SNVT_length_mil [225]</td>
<td>Length (millimeters)</td>
</tr>
<tr>
<td>FB_SEND_021_SNVT_lev_cont [226]</td>
<td>Continuous level (% of full level)</td>
</tr>
<tr>
<td>FB_SEND_023_SNVT_mass [228]</td>
<td>Mass (grams)</td>
</tr>
<tr>
<td>FB_SEND_024_SNVT_mass_kilo [229]</td>
<td>Mass (kilograms)</td>
</tr>
<tr>
<td>FB_SEND_025_SNVT_mass_mega [231]</td>
<td>Mass (metric tons)</td>
</tr>
<tr>
<td>POUs</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>FB_SEND_026_SNVT_mass_mil</td>
<td>Mass (milligrams)</td>
</tr>
<tr>
<td>FB_SEND_027_SNVT_power</td>
<td>Power (Watts)</td>
</tr>
<tr>
<td>FB_SEND_028_SNVT_power_kilo</td>
<td>Power (kiloWatts)</td>
</tr>
<tr>
<td>FB_SEND_029_SNVT_ppm</td>
<td>Concentration (ppm)</td>
</tr>
<tr>
<td>FB_SEND_030_SNVT_press</td>
<td>Pressure (gauge) (kiloPascals)</td>
</tr>
<tr>
<td>FB_SEND_031_SNVT_res</td>
<td>Electric resistance (Ohms)</td>
</tr>
<tr>
<td>FB_SEND_032_SNVT_res_kilo</td>
<td>Electric resistance (kiloOhms)</td>
</tr>
<tr>
<td>FB_SEND_033_SNVT_sound_db</td>
<td>Sound level (dB)</td>
</tr>
<tr>
<td>FB_SEND_034_SNVT_speed</td>
<td>Linear velocity (meters/second)</td>
</tr>
<tr>
<td>FB_SEND_035_SNVT_speed_mil</td>
<td>Linear velocity (meters/second)</td>
</tr>
<tr>
<td>FB_SEND_036_SNVT_str_asc</td>
<td>Character string (30 characters max)</td>
</tr>
<tr>
<td>FB_SEND_037_SNVT_str_int</td>
<td>Wide character string</td>
</tr>
<tr>
<td>FB_SEND_038_SNVT_telecom</td>
<td>Telecomm states (telecomm state names)</td>
</tr>
<tr>
<td>FB_SEND_039_SNVT_temp</td>
<td>Temperature (degrees Celsius)</td>
</tr>
<tr>
<td>FB_SEND_041_SNVT_vol</td>
<td>Volume (liters)</td>
</tr>
<tr>
<td>FB_SEND_042_SNVT_vol_kilo</td>
<td>Volume (kiloliters)</td>
</tr>
<tr>
<td>FB_SEND_043_SNVT_vol_mil</td>
<td>Volume (milliliters)</td>
</tr>
<tr>
<td>FB_SEND_044_SNVT_volt</td>
<td>Electric voltage (Volts)</td>
</tr>
<tr>
<td>FB_SEND_045_SNVT_volt_dbmv</td>
<td>Electric voltage (dB microVolts)</td>
</tr>
<tr>
<td>FB_SEND_046_SNVT_volt_kilo</td>
<td>Electric voltage (kiloVolts)</td>
</tr>
<tr>
<td>FB_SEND_047_SNVT_volt_mil</td>
<td>Electric voltage (milliVolts)</td>
</tr>
<tr>
<td>FB_SEND_048_SNVT_amp_f</td>
<td>Electric current (Amperes)</td>
</tr>
<tr>
<td>FB_SEND_049_SNVT_angle_f</td>
<td>Angular distance (radians)</td>
</tr>
<tr>
<td>FB_SEND_050_SNVT_angle_vel_f</td>
<td>Angular velocity (radians/second)</td>
</tr>
<tr>
<td>FB_SEND_051_SNVT_count_f</td>
<td>Absolute count (units)</td>
</tr>
<tr>
<td>FB_SEND_052_SNVT_count_inc_f</td>
<td>Increment count (units (delta))</td>
</tr>
<tr>
<td>FB_SEND_053_SNVT_flow_f</td>
<td>Flow volume (liters/second)</td>
</tr>
<tr>
<td>FB_SEND_054_SNVT_length_f</td>
<td>Length (meters)</td>
</tr>
<tr>
<td>FB_SEND_055_SNVT_lev_cont_f</td>
<td>Continuous level (% of full scale)</td>
</tr>
<tr>
<td>FB_SEND_056_SNVT_mass_f</td>
<td>Mass (grams)</td>
</tr>
<tr>
<td>FB_SEND_057_SNVT_power_f</td>
<td>Power (Watts)</td>
</tr>
<tr>
<td>FB_SEND_058_SNVT_ppm_f</td>
<td>Concentration (ppm)</td>
</tr>
<tr>
<td>FB_SEND_059_SNVT_press_f</td>
<td>Pressure (gauge) (Pascals)</td>
</tr>
<tr>
<td>FB_SEND_060_SNVT_res_f</td>
<td>Electric resistance (Ohms)</td>
</tr>
<tr>
<td>FB_SEND_061_SNVT_sound_db_f</td>
<td>Sound level (dBSpl)</td>
</tr>
<tr>
<td>FB_SEND_062_SNVT_speed_f</td>
<td>Linear velocity (meters/second)</td>
</tr>
<tr>
<td>FB_SEND_063_SNVT_temp_f</td>
<td>Temperature (degrees Celsius)</td>
</tr>
<tr>
<td>FB_SEND_064_SNVT_time_f</td>
<td>Elapsed time (seconds)</td>
</tr>
<tr>
<td>FB_SEND_065_SNVT_vol_f</td>
<td>Volume (liters)</td>
</tr>
<tr>
<td>FB_SEND_066_SNVT_volt_f</td>
<td>Electric voltage (Volts)</td>
</tr>
<tr>
<td>FB_SEND_067_SNVT_btu_f</td>
<td>Thermal energy (Btus)</td>
</tr>
<tr>
<td>FB_SEND_068_SNVT_elec_whr_f</td>
<td>Electric energy (Watt-hours)</td>
</tr>
<tr>
<td>FB_SEND_069_SNVT_config_src</td>
<td>Configuration source (configuration source names)</td>
</tr>
<tr>
<td>POUs</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>FB_SEND_070_SNVT_color [292]</td>
<td>CIELAB color (L*, a*, b)</td>
</tr>
<tr>
<td>FB_SEND_071_SNVT_grammage [293]</td>
<td>Grammage (grams/sq meter)</td>
</tr>
<tr>
<td>FB_SEND_072_SNVT_grammage_f [295]</td>
<td>Grammage (grams/sq meter)</td>
</tr>
<tr>
<td>FB_SEND_073_SNVT_file_req [296]</td>
<td>File request</td>
</tr>
<tr>
<td>FB_SEND_074_SNVT_file_status [297]</td>
<td>File status</td>
</tr>
<tr>
<td>FB.Send_075_SNVT_freq_f [299]</td>
<td>Frequency (Hertz)</td>
</tr>
<tr>
<td>FB.Send_076_SNVT_freq_hz [300]</td>
<td>Frequency (Hertz)</td>
</tr>
<tr>
<td>FB.Send_077_SNVT_freq_kilohz [301]</td>
<td>Frequency (kiloHertz)</td>
</tr>
<tr>
<td>FB.Send_078_SNVT_freq_milhz [303]</td>
<td>Frequency (Hertz)</td>
</tr>
<tr>
<td>FB_SEND_079_SNVT_lux [304]</td>
<td>Illumination (lux)</td>
</tr>
<tr>
<td>FB.Send_081_SNVT_lev_percent [306]</td>
<td>Percentage level (% of full level)</td>
</tr>
<tr>
<td>FB.Send_082_SNVT_multiplier [307]</td>
<td>Multiplier (16-bit unsigned value)</td>
</tr>
<tr>
<td>FB.Send_083_SNVT_state [308]</td>
<td>State vector (16 individual bit values)</td>
</tr>
<tr>
<td>FB.Send_084_SNVT_time_stamp [310]</td>
<td>Time stamp (year, month, day, hour, minute, second)</td>
</tr>
<tr>
<td>FB.Send_085_SNVT_zerospan [311]</td>
<td>Zero and span (Zero, span)</td>
</tr>
<tr>
<td>FB.Send_086_SNVT_maggspan [312]</td>
<td>ISO 7811 (40 hexadecimal digits)</td>
</tr>
<tr>
<td>FB.Send_087_SNVT_elapsed_tm [314]</td>
<td>Elapsed time (day, hour, minute, second, millisecond)</td>
</tr>
<tr>
<td>FB.Send_088_SNVT_alarm [315]</td>
<td>Alarm status</td>
</tr>
<tr>
<td>FB.Send_089_SNVT_currency [316]</td>
<td>Currency (unit, magnitude, value)</td>
</tr>
<tr>
<td>FB.Send_090_SNVT_file_pos [318]</td>
<td>File position (pointer, length)</td>
</tr>
<tr>
<td>FB.Send_091_SNVT_muldiv [319]</td>
<td>Multiply/Divide (multiplier, divisor)</td>
</tr>
<tr>
<td>FB.Send_092_SNVT_obj_request [320]</td>
<td>Object request (ID, request)</td>
</tr>
<tr>
<td>FB.Send_093_SNVT_obj_status [322]</td>
<td>Object status (ID, status flags)</td>
</tr>
<tr>
<td>FB.Send_094_SNVT_preset [324]</td>
<td>Preset (mode, data, time)</td>
</tr>
<tr>
<td>FB.Send_095_SNVT_switch [325]</td>
<td>Switch (value, state)</td>
</tr>
<tr>
<td>FB.Send_096_SNVT_translation_table [327]</td>
<td>Translation table (points, interpolation)</td>
</tr>
<tr>
<td>FB.Send_097_SNVT_override [328]</td>
<td>Override code (override code names)</td>
</tr>
<tr>
<td>FB.Send_098_SNVT_pwr_fact [329]</td>
<td>Power factor (multiplier)</td>
</tr>
<tr>
<td>FB.Send_099_SNVT_pwr_fact_f [331]</td>
<td>Power factor (multiplier)</td>
</tr>
<tr>
<td>FB.Send_100_SNVT_density [332]</td>
<td>Density (kilograms/cubic meter)</td>
</tr>
<tr>
<td>FB.Send_101_SNVT_density_f [333]</td>
<td>Density (kilograms/cubic meter)</td>
</tr>
<tr>
<td>FB.Send_102_SNVT_rpm [335]</td>
<td>Angular velocity (revolutions/minute (RPM))</td>
</tr>
<tr>
<td>FB.Send_103_SNVT_hvac_emerg [336]</td>
<td>HVAC emergency mode (emergency mode names)</td>
</tr>
<tr>
<td>FB.Send_104_SNVT_angle_deg [337]</td>
<td>Angular distance (degrees)</td>
</tr>
<tr>
<td>FB.Send_105_SNVT_temp_p [339]</td>
<td>Temperature (degrees Celsius)</td>
</tr>
<tr>
<td>FB.Send_106_SNVT_temp_setpt [340]</td>
<td>Temperature (6 temperature values)</td>
</tr>
<tr>
<td>FB.Send_107_SNVT_time_sec [341]</td>
<td>Elapsed time (seconds)</td>
</tr>
<tr>
<td>FB.Send_108_SNVT_hvac_mode [343]</td>
<td>HVAC mode (HVAC mode names)</td>
</tr>
<tr>
<td>FB.Send_109_SNVT_hvac_occupancy [344]</td>
<td>Occupancy (occupancy code names)</td>
</tr>
<tr>
<td>FB.Send_110_SNVT_area [345]</td>
<td>Area (square meters)</td>
</tr>
<tr>
<td>FB.Send_111_SNVT_hvac_overid [347]</td>
<td>HVAC override (state, pct, flow)</td>
</tr>
<tr>
<td>FB.Send_112_SNVT_hvac_status [348]</td>
<td>HVAC status (mode, 5 percents, flag)</td>
</tr>
<tr>
<td>FB.Send_113_SNVT_pres_p [349]</td>
<td>Pressure (gauge) (Pascals)</td>
</tr>
<tr>
<td>POUs</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>FB_SEND_114_SNVT_address [P.351]</td>
<td>Neuron address (16-bit address value)</td>
</tr>
<tr>
<td>FB_SEND_115_SNVT_scene [P.352]</td>
<td>Scene control (function, scene number)</td>
</tr>
<tr>
<td>FB_SEND_116_SNVT_scene_cfg [P.353]</td>
<td>Scene configuration</td>
</tr>
<tr>
<td>FB_SEND_117_SNVT_setting [P.355]</td>
<td>Setting control (function, setting, rotation)</td>
</tr>
<tr>
<td>FB_SEND_118_SNVT_evap_state [P.356]</td>
<td>Evaporator state (evaporator state names)</td>
</tr>
<tr>
<td>FB_SEND_119_SNVT_therm_mode [P.357]</td>
<td>Thermostat mode (thermostat mode names)</td>
</tr>
<tr>
<td>FB_SEND_120_SNVT_defr_mode [P.359]</td>
<td>Defrost mode (defrost mode names)</td>
</tr>
<tr>
<td>FB_SEND_121_SNVT_defr_term [P.360]</td>
<td>Defrost termination (defrost termination names)</td>
</tr>
<tr>
<td>FB_SEND_122_SNVT_defr_state [P.361]</td>
<td>Defrost state (defrost state names)</td>
</tr>
<tr>
<td>FB_SEND_123_SNVT_time_min [P.363]</td>
<td>Elapsed time (minutes)</td>
</tr>
<tr>
<td>FB_SEND_124_SNVT_time_hour [P.364]</td>
<td>Elapsed time (hours)</td>
</tr>
<tr>
<td>FB_SEND_125_SNVT_ph [P.365]</td>
<td>Acidity (pH) . Ratio of concentration of ions</td>
</tr>
<tr>
<td>FB_SEND_126_SNVT_ph_f [P.367]</td>
<td>Acidity (pH) . Ratio of concentration of ions</td>
</tr>
<tr>
<td>FB_SEND_127_SNVT_chlr_status [P.368]</td>
<td>Chiller status (run mode, op mode, state bits)</td>
</tr>
<tr>
<td>FB_SEND_128_SNVT_tod_event [P.369]</td>
<td>Time of day event (current, next, time)</td>
</tr>
<tr>
<td>FB_SEND_129_SNVT smo_ obscur [P.371]</td>
<td>Smoke obscuration (percent obscuration)</td>
</tr>
<tr>
<td>FB_SEND_130_SNVT_fire_test [P.372]</td>
<td>Fire test request (fire test names)</td>
</tr>
<tr>
<td>FB_SEND_131_SNVT_temp_ror [P.373]</td>
<td>Temperature rate of change/rise (degrees Celsius/minute)</td>
</tr>
<tr>
<td>FB_SEND_132_SNVT_fire_init [P.375]</td>
<td>Fire initiator type (fire initiator type names)</td>
</tr>
<tr>
<td>FB_SEND_133_SNVT_fire_indcte [P.376]</td>
<td>Fire indicator type (fire indicator type names)</td>
</tr>
<tr>
<td>FB_SEND_134_SNVT_time_zone [P.377]</td>
<td>Time zone descriptor (offset, type, startDST, endDST)</td>
</tr>
<tr>
<td>FB_SEND_135_SNVT_earth_pos [P.379]</td>
<td>Earth position</td>
</tr>
<tr>
<td>FB_SEND_136_SNVT_reg_val [P.380]</td>
<td>Register value</td>
</tr>
<tr>
<td>FB_SEND_137_SNVT_reg_val ts [P.381]</td>
<td>Register value</td>
</tr>
<tr>
<td>FB_SEND_138_SNVT_volt_ac [P.383]</td>
<td>Voltage in alternating current (volts AC)</td>
</tr>
<tr>
<td>FB_SEND_139_SNVT_amp_ac [P.384]</td>
<td>Amperage in alternating current (amperes AC)</td>
</tr>
<tr>
<td>FB_SEND_143_SNVT_turbidity [P.385]</td>
<td>Turbidity (nephelometric turbidity units)</td>
</tr>
<tr>
<td>FB_SEND_144_SNVT_turbidity_f [P.387]</td>
<td>Turbidity (nephelometric turbidity units)</td>
</tr>
<tr>
<td>FB_SEND_145_SNVT_hvac_type [P.388]</td>
<td>HVAC unit type (HVAC unit type names)</td>
</tr>
<tr>
<td>FB_SEND_146_SNVT_elec_kwh_l [P.389]</td>
<td>Electric energy (kiloWatt-hours) .</td>
</tr>
<tr>
<td>FB_SEND_147_SNVT_temp_diff_p [P.391]</td>
<td>Temp difference (degrees Celsius) .</td>
</tr>
<tr>
<td>FB_SEND_148_SNVT_ctrl_req [P.392]</td>
<td>Control request (receiver ID, sender ID, sender priority)</td>
</tr>
<tr>
<td>FB_SEND_149_SNVT_ctrl_resp [P.393]</td>
<td>Control response (status, sender, controller ID)</td>
</tr>
<tr>
<td>FB_SEND_150_SNVT_ptz [P.395]</td>
<td>Camera PTZ (pan, pan speed, tilt, tilt speed, zoom, zoom speed)</td>
</tr>
<tr>
<td>FB_SEND_151_SNVT_privacyzone [P.396]</td>
<td>Privacy zone (action, zone number, camera ID)</td>
</tr>
<tr>
<td>FB_SEND_152_SNVT_pos ctrl [P.397]</td>
<td>Position control</td>
</tr>
<tr>
<td>FB_SEND_153_SNVT_enthalpy [P.399]</td>
<td>Enthalpy (kiloJoules/kg)</td>
</tr>
<tr>
<td>FB_SEND_154_SNVT_gfci_status [P.400]</td>
<td>GFCI status type</td>
</tr>
<tr>
<td>FB_SEND_155_SNVT_motor_state [P.401]</td>
<td>Motor state (motor state names)</td>
</tr>
<tr>
<td>FB_SEND_156_SNVT_pumpset_mn [P.403]</td>
<td>Pumpset</td>
</tr>
<tr>
<td>FB_SEND_157_SNVT_ex_control [P.404]</td>
<td>Exclusive control (status, address)</td>
</tr>
<tr>
<td>FB_SEND_158_SNVT_pumpset_sn [P.405]</td>
<td>Pumpset sensor</td>
</tr>
<tr>
<td>POUs</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>FB_SEND_159_SNVT_pump_sensor [407]</td>
<td>Pump sensor (speed, temperature, status)</td>
</tr>
<tr>
<td>FB_SEND_160_SNVT_abs_humid [408]</td>
<td>Absolute humidity (gram/kilogram)</td>
</tr>
<tr>
<td>FB_SEND_161_SNVT_flow_p [409]</td>
<td>Flow volume (cubic meters/hour)</td>
</tr>
<tr>
<td>FB_SEND_162_SNVT_dev_c_mode [411]</td>
<td>Device control mode (device control mode names)</td>
</tr>
<tr>
<td>FB_SEND_163_SNVT_valve_mode [412]</td>
<td>Valve mode (valve mode names)</td>
</tr>
<tr>
<td>FB_SEND_164_SNVT_alarm_2 [413]</td>
<td>Alarm status 2</td>
</tr>
<tr>
<td>FB_SEND_165_SNVT_state_64 [415]</td>
<td>State vector (64 individual bit values)</td>
</tr>
<tr>
<td>FB_SEND_166_SNVT_nv_type [416]</td>
<td>Network variable type</td>
</tr>
<tr>
<td>FB_SEND_168_SNVT_ent_opmode [417]</td>
<td>Entry operation mode</td>
</tr>
<tr>
<td>FB_SEND_169_SNVT_ent_state [419]</td>
<td>Entry state</td>
</tr>
<tr>
<td>FB_SEND_170_SNVT_ent_status [420]</td>
<td>Entry status</td>
</tr>
<tr>
<td>FB_SEND_171_SNVT_flow_dir [421]</td>
<td>Flow direction (flow direction names)</td>
</tr>
<tr>
<td>FB_SEND_172_SNVT_hvac_satsts [423]</td>
<td>HVAC saturation status</td>
</tr>
<tr>
<td>FB_SEND_173_SNVT_dev_status [424]</td>
<td>Device status</td>
</tr>
<tr>
<td>FB_SEND_174_SNVT_dev_fault [425]</td>
<td>Device fault states</td>
</tr>
<tr>
<td>FB_SEND_175_SNVT_dev_maint [427]</td>
<td>Device maintenance</td>
</tr>
<tr>
<td>FB_SEND_176_SNVT_date_event [428]</td>
<td>Date event</td>
</tr>
<tr>
<td>FB_SEND_177_SNVT_sched_val [429]</td>
<td>Scheduler value</td>
</tr>
<tr>
<td>FB_SEND_180_SNVT_sblnd_state [431]</td>
<td>Sunblind State</td>
</tr>
<tr>
<td>FB_SEND_181_SNVT_rac_ctrl [432]</td>
<td>Rail-Audio Controller Control</td>
</tr>
<tr>
<td>FB_SEND_182_SNVT_rac_req [433]</td>
<td>Rail-Audio Controller Request</td>
</tr>
<tr>
<td>FB_SEND_184_SNVT_clothes_w_c [436]</td>
<td>Clothes Washer Command</td>
</tr>
<tr>
<td>FB_SEND_185_SNVT_clothes_w_m [437]</td>
<td>Clothes Washer-Management Status</td>
</tr>
<tr>
<td>FB_SEND_186_SNVT_clothes_w_s [439]</td>
<td>Clothes Washer Status</td>
</tr>
<tr>
<td>FB_SEND_188_SNVT_clothes_w_a [441]</td>
<td>Clothes Washer Alarm</td>
</tr>
<tr>
<td>FB_SEND_189_SNVT_switch_2 [443]</td>
<td>Switch with scene and setting control</td>
</tr>
<tr>
<td>FB_SEND_190_SNVT_color_2 [444]</td>
<td>Color.</td>
</tr>
<tr>
<td>FB_SEND_191_SNVT_log_status [445]</td>
<td>Log status (hundredths of second)</td>
</tr>
<tr>
<td>FB_SEND_192_SNVT_time_stamp_p [447]</td>
<td>Precision timestamp. (seconds)</td>
</tr>
<tr>
<td>FB_SEND_193_SNVT_log_fx_request [448]</td>
<td>Log file transfer request.</td>
</tr>
<tr>
<td>FB_SEND_194_SNVT_log_fx_status [449]</td>
<td>Log file transfer status.</td>
</tr>
<tr>
<td>FB_SEND_195_SNVT_log_request [451]</td>
<td>Log status request.</td>
</tr>
<tr>
<td>FB_SEND_196_SNVT_enthalpy_d [452]</td>
<td>Enthalpy difference (kJ/kg)</td>
</tr>
<tr>
<td>FB_SEND_197_SNVT_ambient_ac_mil [454]</td>
<td>Electrical current (milliAmperes)</td>
</tr>
<tr>
<td>FB_SEND_198_SNVT_time_hour_p [455]</td>
<td>Time hour (hours)</td>
</tr>
<tr>
<td>FB_SEND_199_SNVT_lamp_status [456]</td>
<td>Lamp Status</td>
</tr>
<tr>
<td>FB_SEND_200_SNVT_environment [458]</td>
<td>Environment</td>
</tr>
<tr>
<td>FB_SEND_201_SNVT_geo_loc [459]</td>
<td>Geographic Location</td>
</tr>
</tbody>
</table>
This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_amp.

**SNVT number:** 001.

**Description:** Electric current (ampere).

### VAR_INPUT

- `wNVIndex` : WORD;
- `rValue` : REAL;
- `bStart` : BOOL;
- `bSendInit` : BOOL := bSendInitDefault;
- `bAuto` : BOOL := bAutoDefault;
- `rValueLimit` : REAL := 1;
- `tMinSendTime` : TIME := tMinSendTimeDefault;
- `tMaxSendTime` : TIME := tMaxSendTimeDefault;
- `bDisabled` : BOOL := FALSE;

**wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**rValue:** Min: -3276.8 / Max: 3276.7.

**bStart:** A positive edge starts the send process (irrespective of `bAuto`).

**bSendInit:** Automatically sends the value once when the PLC restarts.

**bAuto:** Automatic sending [630] is selected when a value changes or the time `tMaxSendTime` has elapsed (polling).

**rValueLimit:** Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

**tMinSendTime:** Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

**tMaxSendTime:** Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (`rValueLimit`) was not reached (polling). The value 0 disables this function.

**bDisabled:** TRUE = deselection of the block.

### VAR_OUTPUT

- `bBusy` : BOOL;
- `bError` : BOOL;
- `eError` : E_LON_ERROR;
- `dwErrorKL` : DWORD;
bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

VAR_INPUT

wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;

This function block sends the following LON output variable (nvo):

SNVT Name: SNVT_amp_mil.

SNVT number: 002.

Description: Electric current (milliampere).

wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

rValue: Min: -3276.8 / Max: 3276.7.

bStart: A positive edge starts the send process (irrespective of bAuto).
bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

rValueLimit: Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.3 FB_SEND_003_SNVT_angle
This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_angle.

**SNVT number:** 003.

**Description:** Angular distance (radian).

### VAR_INPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>WORD</td>
</tr>
<tr>
<td>rValue</td>
<td>REAL</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL</td>
</tr>
<tr>
<td>bSendInit</td>
<td>BOOL</td>
</tr>
<tr>
<td>bAuto</td>
<td>BOOL</td>
</tr>
<tr>
<td>rValueLimit</td>
<td>REAL</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>TIME</td>
</tr>
<tr>
<td>tMaxSendTime</td>
<td>TIME</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL</td>
</tr>
</tbody>
</table>

- **wNVIndex**: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **rValue**: Min: 0 / Max: 65.535.
- **bStart**: A positive edge starts the send process (irrespective of bAuto).
- **bSendInit**: Automatically sends the value once when the PLC restarts.
- **bAuto**: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).
- **rValueLimit**: Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.
- **tMinSendTime**: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.
- **tMaxSendTime**: Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.
- **bDisabled**: TRUE = deselection of the block.

### VAR_OUTPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>BOOL</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>DWORD</td>
</tr>
</tbody>
</table>

- **bBusy**: The bBusy output is TRUE as long as values are sent.
- **bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.
- **eError**: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.
- **dwErrorKL**: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

### VAR_IN_OUT

<table>
<thead>
<tr>
<th>Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>stLON_Com</td>
</tr>
</tbody>
</table>

- **stLON_Com**: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).
### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

#### 4.1.3.4  FB_SEND_004_SNVT_angle_vel

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_angle_vel.

**SNVT number:** 004.

**Description:** Angular velocity (radian / second).

**VAR_INPUT**

<table>
<thead>
<tr>
<th>wNVIndex</th>
<th>bBusy</th>
</tr>
</thead>
<tbody>
<tr>
<td>rValue</td>
<td>bError</td>
</tr>
<tr>
<td>bStart</td>
<td>eError</td>
</tr>
<tr>
<td>bSendInit</td>
<td>dwErrorKL</td>
</tr>
<tr>
<td>bAuto</td>
<td></td>
</tr>
<tr>
<td>rValueLimit</td>
<td></td>
</tr>
<tr>
<td>tMinSendTime</td>
<td></td>
</tr>
<tr>
<td>tMaxSendTime</td>
<td></td>
</tr>
<tr>
<td>tDisabled</td>
<td></td>
</tr>
<tr>
<td>stLON_Corn</td>
<td></td>
</tr>
</tbody>
</table>

- **wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **rValue:** Min: -3276.8 / Max: 3276.7.
- **bStart:** A positive edge starts the send process (irrespective of bAuto).
- **bSendInit:** Automatically sends the value once when the PLC restarts.
- **bAuto:** Automatic sending \([630]\) is selected when a value changes or the time \(tMaxSendTime\) has elapsed (polling).
- **rValueLimit:** Parameter for automatic sending \([630]\). The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.
- **tMinSendTime:** Parameter for automatic sending \([630]\). A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.
- **tMaxSendTime:** Parameter for automatic sending \([630]\). The value is sent when this time has elapsed at the latest, even if the minimum change in value \((rValueLimit)\) was not reached (polling). The value 0 disables this function.
- **bDisabled:** TRUE = deselection of the block.
VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.5 FB_SEND_005_SNVT_btu_kilo

This function block sends the following LON output variable (nvo):

SNVT Name: SNVT_btu_kilo.
SNVT number: 005.
Description: Thermal energy (kiloBtu).

VAR_INPUT

wNVIndex : WORD;
uValue : UINT;
bStart : BOOL;
bSendInit := bSendInitDefault;
bAuto := bAutoDefault;
uValueLimit := 1;
tMinSendTime := tMinSendTimeDefault;
tMaxSendTime := tMaxSendTimeDefault;
bDisabled := FALSE;
wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

uiValue: Min: 0 / Max: 65535.

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

uiValueLimit: Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (uiValueLimit) was not reached (polling). The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

<table>
<thead>
<tr>
<th>bBusy</th>
<th>BOOL;</th>
</tr>
</thead>
<tbody>
<tr>
<td>bError</td>
<td>BOOL;</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR;</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>DWORD;</td>
</tr>
</tbody>
</table>

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

<table>
<thead>
<tr>
<th>stLON_Com</th>
<th>ST_LON_Communication;</th>
</tr>
</thead>
</table>

stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_btu_mega.

**SNVT number:** 006.

**Description:** Thermal energy (megaBtu).

### VAR_INPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.</td>
</tr>
<tr>
<td>uiValue</td>
<td>Min: 0 / Max: 65535.</td>
</tr>
<tr>
<td>bStart</td>
<td>A positive edge starts the send process (irrespective of bAuto).</td>
</tr>
<tr>
<td>bSendInit</td>
<td>Automatically sends the value once when the PLC restarts.</td>
</tr>
<tr>
<td>bAuto</td>
<td>Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).</td>
</tr>
<tr>
<td>uiValueLimit</td>
<td>Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.</td>
</tr>
<tr>
<td>tMaxSendTime</td>
<td>Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (uiValueLimit) was not reached (polling). The value 0 disables this function.</td>
</tr>
<tr>
<td>bDisabled</td>
<td>TRUE = deselection of the block.</td>
</tr>
</tbody>
</table>

### VAR_OUTPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td></td>
</tr>
<tr>
<td>bError</td>
<td></td>
</tr>
<tr>
<td>eError</td>
<td></td>
</tr>
<tr>
<td>dwErrorKL</td>
<td></td>
</tr>
</tbody>
</table>
bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.7 FB_SEND_007_SNVT_char_ascii

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_char_ascii.

**SNVT number:** 007.

**Description:** ASCII character (8-bit ASCII character).

VAR_INPUT

wNVIndex : WORD;
sValue : STRING(1);
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

sValue: STRING(1).

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.
**bAuto**: Automatic sending [630] is selected when a value changes or the time \( t_{\text{MaxSendTime}} \) has elapsed (polling).

**tMinSendTime**: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

**tMaxSendTime**: Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

**bDisabled**: TRUE = deselection of the block.

### VAR_OUTPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>The bBusy output is TRUE as long as values are sent.</td>
</tr>
<tr>
<td>bError</td>
<td>This output goes TRUE as soon as an error occurs. This error is described via the variable eError.</td>
</tr>
<tr>
<td>eError</td>
<td>This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.</td>
</tr>
</tbody>
</table>

### VAR_IN_OUT

<table>
<thead>
<tr>
<th>Structure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stLON_Com</td>
<td>This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).</td>
</tr>
</tbody>
</table>

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.3.8 FB_SEND_008_SNVT_count

This function block sends the following LON output variable (nvo):

**SNVT Name**: SNVT_count.

**SNVT number**: 008.

**Description**: Counter (only positive values).
VAR_INPUT

wNVIndex : WORD;
uiValue : UINT;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
uiValueLimit : UINT := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

uiValue: Min: 0 / Max: 65535.

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

uiValueLimit: Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (uiValueLimit) was not reached (polling). The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
This function block sends the following LON output variable (nvo):

**SNVT Name**: SNVT_count_inc.

**SNVT number**: 009.

**Description**: Counter (negative and positive values).

### VAR_INPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>WORD</td>
</tr>
<tr>
<td>iValue</td>
<td>INT</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL</td>
</tr>
<tr>
<td>bSendInit</td>
<td>BOOL</td>
</tr>
<tr>
<td>bAuto</td>
<td>BOOL</td>
</tr>
<tr>
<td>iValueLimit</td>
<td>UINT</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>TIME</td>
</tr>
<tr>
<td>tMaxSendTime</td>
<td>TIME</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL</td>
</tr>
</tbody>
</table>

- **wNVIndex**: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **iValue**: Min: -32768 / Max: 32767.
- **bStart**: A positive edge starts the send process (irrespective of bAuto).
- **bSendInit**: Automatically sends the value once when the PLC restarts.
- **bAuto**: **Automatic sending** [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).
- **iValueLimit**: Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.
- **tMinSendTime**: Parameter for automatic sending [630]. A new value is sent after this time has elapsed at the earliest. This prevents continuous sending.
- **tMaxSendTime**: Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (iValueLimit) was not reached (polling). The value 0 disables this function.
- **bDisabled**: TRUE = deselection of the block.

### VAR_OUTPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>BOOL</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>DWORD</td>
</tr>
</tbody>
</table>

**bBusy**: TRUE = block is busy sending.

**bError**: TRUE = an error occurred.

**eError**: Error code.

**dwErrorKL**: Error code.
bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see _E_LON_ERROR_ [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [271] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com: This structure is used to link FB_LON_KL6401 [271] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.10 FB_SEND_011_SNVT_date_day

```
FB_SEND_011_SNVT_date_day

wNVIndex   bBusy
eValue     bError
bStart     eError
bSendInit  dwErrorKL
bAuto
tMinSendTime
bAuto
tMaxSendTime
bAuto
tbDisabled
stLON_Com
```

This function block sends the following LON output variable (nvo):

SNVT Name: SNVT_date_day.

SNVT number: 011.

Description: Day of the week.

VAR_INPUT

```
wNVIndex : WORD;
eValue : _E_LON_days_of_week_t;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

eValue: Enum to be sent (see _E_LON_days_of_week_t_ [513]).

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.
bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>BOOL</td>
<td>The bBusy output is TRUE as long as values are sent.</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
<td>This output goes TRUE as soon as an error occurs. This error is described via the variable eError.</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
<td>This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>DWORD</td>
<td>Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.</td>
</tr>
</tbody>
</table>

VAR_IN_OUT

<table>
<thead>
<tr>
<th>Structure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stLON_Com</td>
<td>This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).</td>
</tr>
</tbody>
</table>

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.11 FB_SEND_013_SNVT_elec_kwh

This function block sends the following LON output variable (nvo):

SNVT Name: SNVT_elec_kwh.

SNVT number: 013.

Description: Electric energy (kW/h).
**VAR_INPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>WORD</td>
<td>Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.</td>
</tr>
<tr>
<td>uiValue</td>
<td>UINT</td>
<td>Min: 0 / Max: 65535.</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL</td>
<td>A positive edge starts the send process (irrespective of bAuto).</td>
</tr>
<tr>
<td>bSendInit</td>
<td>BOOL</td>
<td>Automatically sends the value once when the PLC restarts.</td>
</tr>
<tr>
<td>bAuto</td>
<td>BOOL</td>
<td>Automatic sending is selected when a value changes or the time tMaxSendTime has elapsed (polling).</td>
</tr>
<tr>
<td>uiValueLimit</td>
<td>UINT</td>
<td>Parameter for automatic sending. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>TIME</td>
<td>Parameter for automatic sending. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.</td>
</tr>
<tr>
<td>tMaxSendTime</td>
<td>TIME</td>
<td>Parameter for automatic sending. The value is sent when this time has elapsed at the latest, even if the minimum change in value (uiValueLimit) was not reached (polling). The value 0 disables this function.</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL</td>
<td>TRUE = deselection of the block.</td>
</tr>
</tbody>
</table>

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>BOOL</td>
<td>The bBusy output is TRUE as long as values are sent.</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
<td>This output goes TRUE as soon as an error occurs. This error is described via the variable eError.</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
<td>This output outputs an error code in the event of an error (see E_LON_ERROR). bError goes TRUE at the same time.</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>DWORD</td>
<td>Error ID of the function block FB_LON_KL6401() (see dwErrorKL). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.</td>
</tr>
</tbody>
</table>

**VAR_IN_OUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stLON_Com</td>
<td>ST_LON_Communication</td>
<td>This structure is used to link FB_LON_KL6401() with the send/receive function (see ST_LON_Communication).</td>
</tr>
</tbody>
</table>

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_elec_whr.

**SNVT number:** 014.

**Description:** Electric energy (W/h).

**VAR_INPUT**

- `wNVIndex`: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- `rValue`: Min: 0 / Max: 6553.5.
- `bStart`: A positive edge starts the send process (irrespective of `bAuto`).
- `bSendInit`: Automatically sends the value once when the PLC restarts.
- `bAuto`: Automatic sending [630] is selected when a value changes or the time `tMaxSendTime` has elapsed (polling).
- `rValueLimit`: Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.
- `tMinSendTime`: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.
- `tMaxSendTime`: Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (`rValueLimit`) was not reached (polling). The value 0 disables this function.
- `bDisabled`: TRUE = deselection of the block.

**VAR_OUTPUT**

- `bBusy`: BOOL;
- `bError`: BOOL;
- `eError`: E_LON_ERROR;
- `dwErrorKL`: DWORD;
bBusy: The `bBusy` output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.

eError: This output outputs an error code in the event of an error (see `F_LON_ERROR [477]`). `bError` goes TRUE at the same time.

dwErrorKL: Error ID of the function block `FB_LON_KL6401 [27]` (see `dwErrorKL [631]`). In this case the variable `eError` has the value `eKL6401_Error`. `bError` goes TRUE at the same time.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

This structure is used to link `FB_LON_KL6401() [27]` with the send/receive function (see `ST_LON_Communication [587]`).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.13 FB_SEND_015_SNVT_flow

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_flow.

**SNVT number:** 015.

**Description:** Volume flow (liters / second).

VAR_INPUT

```
wNVIndex    : WORD;
uValue      : UINT;
bStart       : BOOL;
bSendInit    : BOOL := bSendInitDefault;
bAuto        : BOOL := bAutoDefault;
uValueLimit  : UINT := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled    : BOOL := FALSE;
```

- `wNVIndex`: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- `uiValue`: Min: 0 / Max: 65535.
- `bStart`: A positive edge starts the send process (irrespective of `bAuto`).
bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending ([630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

uiValueLimit: Parameter for automatic sending ([630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

tMinSendTime: Parameter for automatic sending ([630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending ([630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (uiValueLimit) was not reached (polling). The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR ([477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 ([27]) (see dwErrorKL ([631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com: This structure is used to link FB_LON_KL6401() ([27]) with the send/receive function (see ST_LON_Communication ([587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.14   FB_SEND_016_SNVT_flow_mil

```
FB_SEND_016_SNVT_flow_mil
  wNVIndex
  uiValue
  bStart
  bSendInit
  bAuto
  uiValueLimit
  tMinSendTime
  tMaxSendTime
  bDisabled
  stLON_Com
```
This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_flow_mil.

**SNVT number:** 016.

**Description:** Volume flow (milliliters / second).

### VAR_INPUT

- `wNVIndex`: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- `uiValue`: Min: 0 / Max: 65535.
- `bStart`: A positive edge starts the send process (irrespective of `bAuto`).
- `bSendInit`: Automatically sends the value once when the PLC restarts.
- `bAuto`: Automatic sending [630] is selected when a value changes or the time `tMaxSendTime` has elapsed (polling).
- `uiValueLimit`: Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.
- `tMinSendTime`: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.
- `tMaxSendTime`: Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (`uiValueLimit`) was not reached (polling). The value 0 disables this function.
- `bDisabled`: TRUE = deselection of the block.

### VAR_OUTPUT

- `bBusy`: The `bBusy` output is TRUE as long as values are sent.
- `bError`: This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.
- `eError`: This output outputs an error code in the event of an error (see `E_LON_ERROR` [477]). `bError` goes TRUE at the same time.
- `dwErrorKL`: Error ID of the function block `FB_LON_KL6401` [27] (see `dwErrorKL` [631]). In this case the variable `eError` has the value eKL6401_Error. `bError` goes TRUE at the same time.

### VAR_IN_OUT

- `stLON_Com`: This structure is used to link `FB_LON_KL6401` [27] with the send/receive function (see `ST_LON_Communication` [587]).
Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.15 FB_SEND_017_SNVT_length

This function block sends the following LON output variable (nvo):

**SNVT Name**: SNVT_length.

**SNVT number**: 017.

**Description**: Length (meter).

**VAR_INPUT**

<table>
<thead>
<tr>
<th>wNVIndex</th>
<th>: WORD;</th>
</tr>
</thead>
<tbody>
<tr>
<td>rValue</td>
<td>: REAL;</td>
</tr>
<tr>
<td>bStart</td>
<td>: BOOL;</td>
</tr>
<tr>
<td>bSendInit</td>
<td>: BOOL := bSendInitDefault;</td>
</tr>
<tr>
<td>bAuto</td>
<td>: BOOL := bAutoDefault;</td>
</tr>
<tr>
<td>rValueLimit</td>
<td>: REAL := 1;</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>: TIME := tMinSendTimeDefault;</td>
</tr>
<tr>
<td>tMaxSendTime</td>
<td>: TIME := tMaxSendTimeDefault;</td>
</tr>
<tr>
<td>bDisabled</td>
<td>: BOOL := FALSE;</td>
</tr>
</tbody>
</table>

**wNVIndex**: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**rValue**: Min: 0 / Max: 6553.5.

**bStart**: A positive edge starts the send process (irrespective of bAuto).

**bSendInit**: Automatically sends the value once when the PLC restarts.

**bAuto**: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

**rValueLimit**: Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

**tMinSendTime**: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

**tMaxSendTime**: Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

**bDisabled**: TRUE = deselection of the block.
VAR_OUTPUT

<table>
<thead>
<tr>
<th>bBusy</th>
<th>BOOL</th>
</tr>
</thead>
<tbody>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>DWORD</td>
</tr>
</tbody>
</table>

**bBusy**: The *bBusy* output is TRUE as long as values are sent.

**bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

**eError**: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). *bError* goes TRUE at the same time.

**dwErrorKL**: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable *eError* has the value eKL6401_Error. *bError* goes TRUE at the same time.

VAR_IN_OUT

<table>
<thead>
<tr>
<th>stLON_Com</th>
<th>ST_LON_Communication</th>
</tr>
</thead>
</table>

**stLON_Com**: This structure is used to link FB_LON_KL6401 [27] with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.3.16 FB_SEND_018_SNVT_length_kilo**

This function block sends the following LON output variable (nvo):

**SNVT Name**: SNVT_length_kilo.

**SNVT number**: 018.

**Description**: Length (kilometer).

VAR_INPUT

<table>
<thead>
<tr>
<th>wNVIndex</th>
<th>WORD</th>
</tr>
</thead>
<tbody>
<tr>
<td>rValue</td>
<td>REAL</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL</td>
</tr>
<tr>
<td>bSendInit</td>
<td>BOOL := bSendInitDefault;</td>
</tr>
<tr>
<td>bAuto</td>
<td>BOOL := bAutoDefault;</td>
</tr>
<tr>
<td>rValueLimit</td>
<td>REAL := 1;</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>TIME := tMinSendTimeDefault;</td>
</tr>
<tr>
<td>tMaxSendTime</td>
<td>TIME := tMaxSendTimeDefault;</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL := FALSE;</td>
</tr>
</tbody>
</table>

**rValue**: Min: 0 / Max: 6553.5.
bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

rValueLimit: Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

| bBusy       : BOOL; |
| bError      : BOOL; |
| eError      : E_LON_ERROR; |
| dwErrorKL   : DWORD; |

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27]() (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

| stLON_Com : ST_LON_Communication; |

stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_length_micr.

**SNVT number:** 019.

**Description:** Length (micrometer).

**VAR_INPUT**

```plaintext
wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

- **wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **rValue:** Min: 0 / Max: 6553.5.
- **bStart:** A positive edge starts the send process (irrespective of **bAuto**).
- **bSendInit:** Automatically sends the value once when the PLC restarts.
- **bAuto:** Automatic sending [630] is selected when a value changes or the time **tMaxSendTime** has elapsed (polling).
- **rValueLimit:** Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.
- **tMinSendTime:** Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.
- **tMaxSendTime:** Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (**rValueLimit**) was not reached (polling). The value 0 disables this function.

**VAR_OUTPUT**

```plaintext
bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;
```

- **bBusy:**
- **bError:**
- **eError:**
- **dwErrorKL:**
bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT
stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

Development environment | required TC3 PLC library
------------------------|--------------------------
TwinCAT from v3.1.4020.14 | Tc2_LON from 3.3.4.0

4.1.3.18 FB_SEND_020_SNVT_length_mil

This function block sends the following LON output variable (nvo):

SNVT Name: SNVT_length_mil.

SNVT number: 020.

Description: Length (millimeter).

VAR_INPUT
wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

rValue: Min: 0 / Max: 6553.5.

bStart: A positive edge starts the send process (irrespective of bAuto).
bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

rValueLimit: Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT
bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT
stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.19 FB_SEND_021_SNVT_lev_cont
This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_lev_cont.

**SNVT number:** 021.

**Description:** Continuous change in value (% from maximum level).

### VAR_INPUT

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>WORD</td>
</tr>
<tr>
<td>rValue</td>
<td>REAL</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL</td>
</tr>
<tr>
<td>bSendInit</td>
<td>BOOL</td>
</tr>
<tr>
<td>bAuto</td>
<td>BOOL</td>
</tr>
<tr>
<td>rValueLimit</td>
<td>REAL</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>TIME</td>
</tr>
<tr>
<td>tMaxSendTime</td>
<td>TIME</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL</td>
</tr>
</tbody>
</table>

- **wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

- **rValue:** Min: 0 / Max: 100.

- **bStart:** A positive edge starts the send process (irrespective of bAuto).

- **bSendInit:** Automatically sends the value once when the PLC restarts.

- **bAuto:** Automatic sending is selected when a value changes or the time tMaxSendTime has elapsed (polling).

- **rValueLimit:** Parameter for automatic sending. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

- **tMinSendTime:** Parameter for automatic sending. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

- **tMaxSendTime:** Parameter for automatic sending. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

- **bDisabled:** TRUE = deselection of the block.

### VAR_OUTPUT

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>BOOL</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>DWORD</td>
</tr>
</tbody>
</table>

- **bBusy:** The bBusy output is TRUE as long as values are sent.

- **bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

- **eError:** This output outputs an error code in the event of an error (see E_LON_ERROR). bError goes TRUE at the same time.

- **dwErrorKL:** Error ID of the function block FB_LON_KL6401() (see dwErrorKL). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

### VAR_IN_OUT

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>stLON_Com</td>
<td>ST_LON_Communication</td>
</tr>
</tbody>
</table>

- **stLON_Com:** This structure is used to link FB_LON_KL6401() with the send/receive function (see ST_LON_Communication).
Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.20 **FB_SEND_023_SNVT_mass**

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_mass.

**SNVT number:** 023.

**Description:** Mass (gram).

**VAR_INPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>WORD</td>
</tr>
<tr>
<td>rValue</td>
<td>REAL</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL</td>
</tr>
<tr>
<td>bSendInit</td>
<td>BOOL := bSendInitDefault;</td>
</tr>
<tr>
<td>bAuto</td>
<td>BOOL := bAutoDefault;</td>
</tr>
<tr>
<td>rValueLimit</td>
<td>REAL := 1;</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>TIME := tMinSendTimeDefault;</td>
</tr>
<tr>
<td>tMaxSendTime</td>
<td>TIME := tMaxSendTimeDefault;</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL := FALSE;</td>
</tr>
</tbody>
</table>

**wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**rValue:** Min: 0 / Max: 6553.5.

**bStart:** A positive edge starts the send process (irrespective of bAuto).

**bSendInit:** Automatically sends the value once when the PLC restarts.

**bAuto:** Automatic sending [*630*] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

**rValueLimit:** Parameter for automatic sending [*630*]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

**tMinSendTime:** Parameter for automatic sending [*630*]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

**tMaxSendTime:** Parameter for automatic sending [*630*]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

**bDisabled:** TRUE = deselection of the block.
VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.21 FB_SEND_024_SNVT_mass_kilo

This function block sends the following LON output variable (nvo):

SNVT Name: SNVT_mass_kilo.

SNVT number: 024.

Description: Mass (kilogram).

VAR_INPUT

wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit := bSendInitDefault;
bAuto := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime := tMinSendTimeDefault;
tMaxSendTime := tMaxSendTimeDefault;
bDisabled := FALSE;
wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

rValue: Min: 0 / Max: 6553.5.

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

rValueLimit: Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

<table>
<thead>
<tr>
<th>bBusy</th>
<th>BOOL</th>
</tr>
</thead>
<tbody>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>DWORD</td>
</tr>
</tbody>
</table>

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

<table>
<thead>
<tr>
<th>stLON_Com</th>
<th>ST_LON_Communication</th>
</tr>
</thead>
</table>

stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_mass_mega.

**SNVT number:** 025.

**Description:** Mass (metric ton).

### VAR_INPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>WORD</td>
</tr>
<tr>
<td>rValue</td>
<td>REAL</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL</td>
</tr>
<tr>
<td>bSendInit</td>
<td>BOOL</td>
</tr>
<tr>
<td>bAuto</td>
<td>BOOL</td>
</tr>
<tr>
<td>rValueLimit</td>
<td>REAL</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>TIME</td>
</tr>
<tr>
<td>tMaxSendTime</td>
<td>TIME</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL</td>
</tr>
</tbody>
</table>

**wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**rValue:** Min: 0 / Max: 6553.5.

**bStart:** A positive edge starts the send process (irrespective of bAuto).

**bSendInit:** Automatically sends the value once when the PLC restarts.

**bAuto:** Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

**rValueLimit:** Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

**tMinSendTime:** Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

**tMaxSendTime:** Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

**bDisabled:** TRUE = deselection of the block.

### VAR_OUTPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>BOOL</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>DWORD</td>
</tr>
</tbody>
</table>
bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see _E_LON_ERROR_ [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see _dwErrorKL_ [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

**VAR_IN_OUT**

```plaintext
stLON_Com : ST_LON_Communication;
```

**VAR_INPUT**

```plaintext
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
stLON_Com
```

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_mass_mil.

**SNVT number:** 026.

**Description:** Mass (milligram).

**VAR_INPUT**

```plaintext
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
stLON_Com
```

**wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**rValue:** Min: 0 / Max: 6553.5.

**bStart:** A positive edge starts the send process (irrespective of bAuto).
bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

rValueLimit: Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

| bBusy    : BOOL; |
| bError   : BOOL; |
| eError   : E_LON_ERROR; |
| dwErrorKL : DWORD; |

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401 [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.24 FB_SEND_027_SNVT_power

```plaintext
FB_SEND_027_SNVT_power
  wNIndex
  rValue
  bStart
  bSendInit
  bAuto
  rValueLimit
  tMinSendTime
  tMaxSendTime
  bDisabled
  stLON_Com
```
This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_power.

**SNVT number:** 027.

**Description:** Power (watt).

### VAR_INPUT

- **wNVIndex** : WORD;
- **rValue** : REAL;
- **bStart** : BOOL;
- **bSendInit** : BOOL := bSendInitDefault;
- **bAuto** : BOOL := bAutoDefault;
- **rValueLimit** : REAL := 1;
- **tMinSendTime** : TIME := tMinSendTimeDefault;
- **tMaxSendTime** : TIME := tMaxSendTimeDefault;
- **bDisabled** : BOOL := FALSE;

**wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**rValue:** Min: 0 / Max: 6553.5.

**bStart:** A positive edge starts the send process (irrespective of **bAuto**).

**bSendInit:** Automatically sends the value once when the PLC restarts.

**bAuto:** Automatic sending is selected when a value changes or the time **tMaxSendTime** has elapsed (polling).

**rValueLimit:** Parameter for automatic sending. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

**tMinSendTime:** Parameter for automatic sending. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

**tMaxSendTime:** Parameter for automatic sending. The value is sent when this time has elapsed at the latest, even if the minimum change in value (**rValueLimit**) was not reached (polling). The value 0 disables this function.

**bDisabled:** TRUE = deselection of the block.

### VAR_OUTPUT

- **bBusy** : BOOL;
- **bError** : BOOL;
- **eError** : E_LON_ERROR;
- **dwErrorKL** : DWORD;

**bBusy:** The **bBusy** output is TRUE as long as values are sent.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable **eError**.

**eError:** This output outputs an error code in the event of an error (see E_LON_ERROR). **bError** goes TRUE at the same time.

**dwErrorKL:** Error ID of the function block **FB_LON_KL6401** (see dwErrorKL). In this case the variable **eError** has the value eKL6401_Error. **bError** goes TRUE at the same time.

### VAR_IN_OUT

**stLON_Com** : ST_LON_Communication;

**stLON_Com:** This structure is used to link **FB_LON_KL6401** with the send/receive function (see ST_LON_Communication).
Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.25 **FB_SEND_028_SNVT_power_kilo**

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_power_kilo.

**SNVT number:** 028.

**Description:** Power (kilowatt).

**VAR_INPUT**

```plaintext
wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit := bSendInitDefault;
bAuto := bAutoDefault;
rValueLimit := 1;
tMinSendTime := tMinSendTimeDefault;
tMaxSendTime := tMaxSendTimeDefault;
bDisabled := FALSE;
```

- **wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

- **rValue:** Min: 0 / Max: 6553.5.

- **bStart:** A positive edge starts the send process (irrespective of bAuto).

- **bSendInit:** Automatically sends the value once when the PLC restarts.

- **bAuto:** Automatic sending is selected when a value changes or the time tMaxSendTime has elapsed (polling).

- **rValueLimit:** Parameter for automatic sending. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

- **tMinSendTime:** Parameter for automatic sending. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

- **tMaxSendTime:** Parameter for automatic sending. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

- **bDisabled:** TRUE = deselection of the block.
VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.26  FB_SEND_029_SNVT_ppm

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_ppm.

**SNVT number:** 029.

**Description:** Concentration (ppm).

VAR_INPUT

wNVIndex : WORD;
uValue : UINT;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
uValueLimit : UINT := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

uiValue: Min: 0 / Max: 65535.

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

uiValueLimit: Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (uiValueLimit) was not reached (polling). The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401[] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_press.

**SNVT number:** 030.

**Description:** Pressure (overpressure) (kilopascal).

**VAR_INPUT**

- wNVIndex : WORD;
- rValue : REAL;
- bStart : BOOL;
- bSendInit : BOOL := bSendInitDefault;
- bAuto : BOOL := bAutoDefault;
- rValueLimit : REAL := 1;
- tMinSendTime : TIME := tMinSendTimeDefault;
- tMaxSendTime : TIME := tMaxSendTimeDefault;
- bDisabled : BOOL := FALSE;

**wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**rValue:** Min: -3276.8 / Max: 3276.7.

**bStart:** A positive edge starts the send process (irrespective of bAuto).

**bSendInit:** Automatically sends the value once when the PLC restarts.

**bAuto:** Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

**rValueLimit:** Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

**tMinSendTime:** Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

**tMaxSendTime:** Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

**bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

- bBusy : BOOL;
- bError : BOOL;
- eError : E_LON_ERROR;
- dwErrorKL : DWORD;
bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401 [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.28 FB_SEND_031_SNVT_res

This function block sends the following LON output variable (nvo):

SNVT Name: SNVT_res.

SNVT number: 031.

Description: Electric resistance (ohm).

VAR_INPUT

wNVIndex : WORD;

rValue : REAL;

bStart : BOOL;

bSendInit : BOOL := bSendInitDefault;

bAuto : BOOL := bAutoDefault;

rValueLimit : REAL := 1;

tMinSendTime : TIME := tMinSendTimeDefault;

tMaxSendTime : TIME := tMaxSendTimeDefault;

bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

rValue: Min: 0 / Max: 6553.5.

bStart: A positive edge starts the send process (irrespective of bAuto).
Programming

bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time $t_{MaxSendTime}$ has elapsed (polling).

rValueLimit: Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value ($r_{ValueLimit}$) was not reached (polling). The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.29 FB_SEND_032_SNVT_res_kilo

```plaintext
FB_SEND_032_SNVT_res_kilo

wNIndex
rValue
bStart
bSendInit
bAuto
rValueLimit
tMinSendTime
tMaxSendTime
bDisabled
stLON_Com
```

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_res_kilo.

**SNVT number:** 032.

**Description:** Electric resistance (kiloohm).

### VAR_INPUT

- **wNVIndex**: WORD;
- **rValue**: REAL;
- **bStart**: BOOL;
- **bSendInit**: BOOL := bSendInitDefault;
- **bAuto**: BOOL := bAutoDefault;
- **rValueLimit**: REAL := 1;
- **tMinSendTime**: TIME := tMinSendTimeDefault;
- **tMaxSendTime**: TIME := tMaxSendTimeDefault;
- **bDisabled**: BOOL := FALSE;

**wNVIndex**: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**rValue**: Min: 0 / Max: 6553.5.

**bStart**: A positive edge starts the send process (irrespective of bAuto).

**bSendInit**: Automatically sends the value once when the PLC restarts.

**bAuto**: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

**rValueLimit**: Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

**tMinSendTime**: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

**tMaxSendTime**: Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

**bDisabled**: TRUE = deselection of the block.

### VAR_OUTPUT

- **bBusy**: BOOL;
- **bError**: BOOL;
- **eError**: E_LON_ERROR;
- **dwErrorKL**: DWORD;

**bBusy**: The bBusy output is TRUE as long as values are sent.

**bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

**eError**: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

**dwErrorKL**: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

### VAR_IN_OUT

- **stLON_Com**: ST_LON_Communication;

**stLON_Com**: This structure is used to link FB_LON_KL6401 [27] with the send/receive function (see ST_LON_Communication [587]).
### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.30 **FB_SEND_033_SNVT_sound_db**

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_sound_db.

**SNVT number:** 033.

**Description:** Sound pressure level (dB).

**VAR_INPUT**

```plaintext
wNVIndex : WORD;
rValue : REAL;
bStart : BOOL := bStartDefault;
bAuto : BOOL := bAutoDefault;

bSendInit : BOOL := bSendInitDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**rValue:** Min: -327.68 / Max: 327.67.

**bStart:** A positive edge starts the send process (irrespective of bAuto).

**bSendInit:** Automatically sends the value once when the PLC restarts.

**bAuto:** **Automatic sending** [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

**rValueLimit:** Parameter for **automatic sending** [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

**tMinSendTime:** Parameter for **automatic sending** [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

**tMaxSendTime:** Parameter for **automatic sending** [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

**bDisabled:** TRUE = deselection of the block.
VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.31 FB_SEND_034_SNVT_speed

This function block sends the following LON output variable (nvo):

**SNVT Name**: SNVT_speed.

**SNVT number**: 034.

**Description**: Linear velocity (meters / second).

VAR_INPUT

wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit := bSendInitDefault;
bAuto := bAutoDefault;
rValueLimit := REAL := 1;
tMinSendTime := tMinSendTimeDefault;
tMaxSendTime := tMaxSendTimeDefault;
bDisabled := BOOL := FALSE;
**wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**rValue:** Min: 0 / Max: 6553.5.

**bStart:** A positive edge starts the send process (irrespective of bAuto).

**bSendInit:** Automatically sends the value once when the PLC restarts.

**bAuto:** Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

**rValueLimit:** Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

**tMinSendTime:** Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

**tMaxSendTime:** Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

**bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>BOOL</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>DWORD</td>
</tr>
</tbody>
</table>

**bBusy:** The bBusy output is TRUE as long as values are sent.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

**eError:** This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

**dwErrorKL:** Error ID of the function block FB_LON_KL6401 [27]() (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

**VAR_IN_OUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>stLON_Com</td>
<td>ST_LON_Communication;</td>
</tr>
</tbody>
</table>

**stLON_Com:** This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
This function block sends the following LON output variable (nvo):

**SNVT Name**: SNVT_speed_mil.

**SNVT number**: 035.

**Description**: Linear velocity (millimeters / second).

**VAR_INPUT**

- **wNVIndex**: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **rValue**: Min: 0 / Max: 65.535.
- **bStart**: A positive edge starts the send process (irrespective of **bAuto**).
- **bSendInit**: Automatically sends the value once when the PLC restarts.
- **bAuto**: Automatic sending [630] is selected when a value changes or the time **tMaxSendTime** has elapsed (polling).
- **rValueLimit**: Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.
- **tMinSendTime**: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.
- **tMaxSendTime**: Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (**rValueLimit**) was not reached (polling). The value 0 disables this function.
- **bDisabled**: TRUE = deselection of the block.

**VAR_OUTPUT**

- **bBusy**: BOOL;
- **bError**: BOOL;
- **eError**: E_LON_ERROR;
- **dwErrorKL**: DWORD;
bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E López ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

<table>
<thead>
<tr>
<th>VAR_INPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex : WORD;</td>
</tr>
<tr>
<td>sValue : STRING(31);</td>
</tr>
<tr>
<td>bStart : BOOL;</td>
</tr>
<tr>
<td>bSendInit : BOOL := bSendInitDefault;</td>
</tr>
<tr>
<td>bAuto : BOOL := bAutoDefault;</td>
</tr>
<tr>
<td>tMinSendTime : TIME := tMinSendTimeDefault;</td>
</tr>
<tr>
<td>tMaxSendTime : TIME := tMaxSendTimeDefault;</td>
</tr>
<tr>
<td>bDisabled : BOOL := FALSE;</td>
</tr>
</tbody>
</table>

This function block sends the following LON output variable (nvo):

SNVT Name: SNVT_str_asc.

SNVT number: 036.

Description: String (max. 30 characters) (ASCII string).

VAR_INPUT

wNVIndex : WORD;
sValue : STRING(31);
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

sValue: STRING(31).

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.
bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>bBusy</th>
<th>: BOOL;</th>
</tr>
</thead>
<tbody>
<tr>
<td>bError</td>
<td>: BOOL;</td>
</tr>
<tr>
<td>eError</td>
<td>: E_LON_ERROR;</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>: DWORD;</td>
</tr>
</tbody>
</table>

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

**VAR_IN_OUT**

stLON_Com : ST_LON_communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.34 **FB_SEND_037_SNVT_str_int**

This function block sends the following LON output variable (nvo):

**SNVT Name**: SNVT_str_int.

**SNVT number**: 037.

**Description**: Wide character string with own code (max. 15 characters) (wide character string).
VAR_INPUT

wNVIndex : WORD;
stValue : ST_LON_SNVT_str_int;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

stValue: Structure of the data to be sent (see ST_LON_SNVT_str_int [618]).

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27](). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

248

Version: 1.5

T1000
4.1.3.35   **FB_SEND_038_SNVT_telcom**

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_telcom.

**SNVT number:** 038.

**Description:** Telephone status (telephone status name).

**VAR_INPUT**

```
VAR_INPUT
wNVIndex     : WORD;
eValue       : E_LON_telcom_states_t;
bStart       : BOOL;
bSendInit    : BOOL := bSendInitDefault;
bAuto        : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled    : BOOL := FALSE;
```

**wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**eValue:** Enum to be sent (see `E_LON_telcom_states_t`

**bStart:** A positive edge starts the send process (irrespective of `bAuto`).

**bSendInit:** Automatically sends the value once when the PLC restarts.

**bAuto:** Automatic sending [630] is selected when a value changes or the time `tMaxSendTime` has elapsed (polling).

**tMinSendTime:** Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

**tMaxSendTime:** Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

**bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

```
VAR_OUTPUT
bBusy     : BOOL;
bError    : BOOL;
eError    : E_LON_ERROR;
dwErrorKL : DWORD;
```

**bBusy:** The `bBusy` output is TRUE as long as values are sent.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.

**eError:** This output outputs an error code in the event of an error (see `E_LON_ERROR` [477]). `bError` goes TRUE at the same time.
**Programming**

**dwErrorKL**: Error ID of the function block FB_LON_KL6401(27) (see dwErrorKL 631). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

**VAR_IN_OUT**

```plaintext
stLON_Com : ST_LON_Communication;
```

**stLON_Com**: This structure is used to link FB_LON_KL6401(27) with the send/receive function (see ST_LON_Communication 587).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.3.36 **FB_SEND_039_SNVT_temp**

This function block sends the following LON output variable (nvo):

**SNVT Name**: SNVT_temp.

**SNVT number**: 039.

**Description**: Temperature (°C).

**VAR_INPUT**

```plaintext
wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex**: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**rValue**: Min: -274 / Max: 6279.5.

**bStart**: A positive edge starts the send process (irrespective of bAuto).

**bSendInit**: Automatically sends the value once when the PLC restarts.

**bAuto**: Automatic sending 630 is selected when a value changes or the time tMaxSendTime has elapsed (polling).
**rValueLimit**: Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

**tMinSendTime**: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

**tMaxSendTime**: Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

**bDisabled**: TRUE = deselection of the block.

**VAR_OUTPUT**

```plaintext
bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;
```

**bBusy**: The bBusy output is TRUE as long as values are sent.

**bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

**eError**: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

**dwErrorKL**: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

**VAR_IN_OUT**

```plaintext
stLON_Com : ST_LON_Communication;
```

**stLON_Com**: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.3.37 FB_SEND_041_SNVT_vol**

This function block sends the following LON output variable (nvo):

**SNVT Name**: SNVT_vol.

**SNVT number**: 041.
**Description:** Volume (liter).

**VAR_INPUT**

```pascal
wNVIndex : WORD;
rValue : REAL;
bStart : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**rValue:** Min: 0 / Max: 6553.5.

**bStart:** A positive edge starts the send process (irrespective of bAuto).

**bSendInit:** Automatically sends the value once when the PLC restarts.

**bAuto:** Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

**rValueLimit:** Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

**tMinSendTime:** Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

**tMaxSendTime:** Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

**bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

```pascal
bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;
```

**bBusy:** The bBusy output is TRUE as long as values are sent.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

**eError:** This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

**dwErrorKL:** Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

**VAR_IN_OUT**

```pascal
stLON_Com : ST_LON_Communication;
```

**stLON_Com:** This structure is used to link FB_LON_KL6401 [27] with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_vol_kilo.

**SNVT number:** 042.

**Description:** Volume (kiloliter).

### VAR_INPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>WORD</td>
</tr>
<tr>
<td>rValue</td>
<td>REAL</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL</td>
</tr>
<tr>
<td>bSendInit</td>
<td>BOOL := bSendInitDefault;</td>
</tr>
<tr>
<td>bAuto</td>
<td>BOOL := bAutoDefault;</td>
</tr>
<tr>
<td>rValueLimit</td>
<td>REAL := 1;</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>TIME := tMinSendTimeDefault;</td>
</tr>
<tr>
<td>tMaxSendTime</td>
<td>TIME := tMaxSendTimeDefault;</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL := FALSE;</td>
</tr>
</tbody>
</table>

**wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**rValue:** Min: 0 / Max: 6553.5.

**bStart:** A positive edge starts the send process (irrespective of bAuto).

**bSendInit:** Automatically sends the value once when the PLC restarts.

**bAuto:** Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

**rValueLimit:** Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

**tMinSendTime:** Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

**tMaxSendTime:** Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

**bDisabled:** TRUE = deselection of the block.

### VAR_OUTPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>BOOL</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR;</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>DWORD;</td>
</tr>
</tbody>
</table>
bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.39 FB_SEND_043_SNVT_vol_mil

This function block sends the following LON output variable (nvo):

SNVT Name: SNVT_vol_mil.

SNVT number: 043.

Description: Volume (milliliter).

VAR_INPUT

wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
 bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

rValue: Min: 0 / Max: 6553.5.

bStart: A positive edge starts the send process (irrespective of bAuto).
bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

rValueLimit: Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT
bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT
stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements
<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.40 FB_SEND_044_SNVT_volt
This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_volt.

**SNVT number:** 044.

**Description:** Electrical voltage (volt).

### VAR_INPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>WORD</td>
</tr>
<tr>
<td>rValue</td>
<td>REAL</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL</td>
</tr>
<tr>
<td>bSendInit</td>
<td>BOOL := bSendInitDefault;</td>
</tr>
<tr>
<td>bAuto</td>
<td>BOOL := bAutoDefault;</td>
</tr>
<tr>
<td>rValueLimit</td>
<td>REAL := 1;</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>TIME := tMinSendTimeDefault;</td>
</tr>
<tr>
<td>tMaxSendTime</td>
<td>TIME := tMaxSendTimeDefault;</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL := FALSE;</td>
</tr>
</tbody>
</table>

- **wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **rValue:** Min: -3276.8 / Max: 3276.7.
- **bStart:** A positive edge starts the send process (irrespective of `bAuto`).
- **bSendInit:** Automatically sends the value once when the PLC restarts.
- **bAuto:** Automatic sending is selected when a value changes or the time `tMaxSendTime` has elapsed (polling).
- **rValueLimit:** Parameter for automatic sending. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.
- **tMinSendTime:** Parameter for automatic sending. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.
- **tMaxSendTime:** Parameter for automatic sending. The value is sent when this time has elapsed at the latest, even if the minimum change in value (`rValueLimit`) was not reached (polling). The value 0 disables this function.
- **bDisabled:** TRUE = deselection of the block.

### VAR_OUTPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>BOOL</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR;</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>DWORD</td>
</tr>
</tbody>
</table>

- **bBusy:** The `bBusy` output is TRUE as long as values are sent.
- **bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.
- **eError:** This output outputs an error code in the event of an error (see `E_LON_ERROR`). `bError` goes TRUE at the same time.
- **dwErrorKL:** Error ID of the function block `FB_LON_KL6401` (see `dwErrorKL`). In this case the variable `eError` has the value eKL6401_Error. `bError` goes TRUE at the same time.

### VAR_IN_OUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>stLON_Com</td>
<td>ST_LON_Communication;</td>
</tr>
</tbody>
</table>

- **stLON_Com:** This structure is used to link `FB_LON_KL6401` with the send/receive function (see `ST_LON_Communication`).
Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.41  **FB_SEND_045_SNVT_volt_dbmv**

```c
FB_SEND_045_SNVT_volt_dbmv
```

This function block sends the following LON output variable (nvo):

**SNVT Name**: SNVT_volt_dbmv.

**SNVT number**: 045.

**Description**: Electrical voltage (dB microvolt).

**VAR_INPUT**

```c
wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit := bSendInitDefault;
bAuto := bAutoDefault;
rValueLimit := REAL := 1;
tMinSendTime := TIME := tMinSendTimeDefault;
tMaxSendTime := TIME := tMaxSendTimeDefault;
bDisabled := BOOL := FALSE;
```

- **wNVIndex**: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

- **rValue**: Min: -327.68 / Max: 327.67.

- **bStart**: A positive edge starts the send process (irrespective of bAuto).

- **bSendInit**: Automatically sends the value once when the PLC restarts.

- **bAuto**: Automatic sending is selected when a value changes or the time tMaxSendTime has elapsed (polling).

- **rValueLimit**: Parameter for automatic sending. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

- **tMinSendTime**: Parameter for automatic sending. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

- **tMaxSendTime**: Parameter for automatic sending. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

- **bDisabled**: TRUE = deselection of the block.
**VAR_OUTPUT**

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

**bBusy:** The *bBusy* output is TRUE as long as values are sent.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

**eError:** This output outputs an error code in the event of an error (see *E_LON_ERROR*). *bError* goes TRUE at the same time.

**dwErrorKL:** Error ID of the function block FB_LON_KL6401 (27) (see *dwErrorKL*). In this case the variable *eError* has the value eKL6401_Error. *bError* goes TRUE at the same time.

**VAR_IN_OUT**

stLON_Com : ST_LON_Communication;

**stLON_Com:** This structure is used to link FB_LON_KL6401 with the send/receive function (see ST_LON_Communication).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.3.42 FB_SEND_046_SNVT_volt_kilo**

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_volt_kilo.

**SNVT number:** 046.

**Description:** Electrical voltage (kilovolt).

**VAR_INPUT**

wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

rValue: Min: -3276.8 / Max: 3276.7.

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

rValueLimit: Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>bBusy</th>
<th>BOOL</th>
</tr>
</thead>
<tbody>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>DWORD</td>
</tr>
</tbody>
</table>

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

**VAR_IN_OUT**

<table>
<thead>
<tr>
<th>stLON_Com</th>
<th>ST_LON_Communication</th>
</tr>
</thead>
</table>

stLON_Com: This structure is used to link FB_LON_KL6401 [27] with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.1.3.43 FB_SEND_047_SNVT_volt_mil

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_volt_mil.

**SNVT number:** 047.

**Description:** Electrical voltage (millivolt).

**VAR_INPUT**

<table>
<thead>
<tr>
<th>wNVIndex</th>
<th>bBusy</th>
</tr>
</thead>
<tbody>
<tr>
<td>rValue</td>
<td>bError</td>
</tr>
<tr>
<td>bStart</td>
<td>eError</td>
</tr>
<tr>
<td>bSendInit</td>
<td>dwErrorKL</td>
</tr>
<tr>
<td>bAuto</td>
<td></td>
</tr>
<tr>
<td>rValueLimit</td>
<td></td>
</tr>
<tr>
<td>tMinSendTime</td>
<td></td>
</tr>
<tr>
<td>tMaxSendTime</td>
<td></td>
</tr>
<tr>
<td>bDisabled</td>
<td></td>
</tr>
</tbody>
</table>

**wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**rValue:** Min: -3276.8 / Max: 3276.7.

**bStart:** A positive edge starts the send process (irrespective of **bAuto**).

**bSendInit:** Automatically sends the value once when the PLC restarts.

**bAuto:** Automatic sending [630] is selected when a value changes or the time **tMaxSendTime** has elapsed (polling).

**rValueLimit:** Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

**tMinSendTime:** Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

**tMaxSendTime:** Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (**rValueLimit**) was not reached (polling). The value 0 disables this function.

**bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>bBusy</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>bError</td>
<td></td>
</tr>
<tr>
<td>eError</td>
<td></td>
</tr>
<tr>
<td>dwErrorKL</td>
<td></td>
</tr>
</tbody>
</table>
**bBusy:** The `bBusy` output is TRUE as long as values are sent.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.

**eError:** This output outputs an error code in the event of an error (see [E_LON_ERROR](#477)). `bError` goes TRUE at the same time.

dwErrorKL: Error ID of the function block `FB_LON_KL6401` (see [dwErrorKL](#631)). In this case the variable `eError` has the value `eKL6401_Error`. `bError` goes TRUE at the same time.

**VAR_IN_OUT**

```plaintext
stLON_Com : ST_LON_Communication;
```

**VAR_IN_OUT**

```plaintext
stLON_Com : This structure is used to link `FB_LON_KL6401` with the send/receive function (see [ST_LON_Communication](#587)).
```

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.3.44 FB_SEND_048_SNVT_amp_f**

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_amp_f.

**SNVT number:** 048.

**Description:** Electric current (ampere).

**VAR_INPUT**

```plaintext
wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**rValue:** Min: -3.40E+38 / Max: 3.40E+38.

**bStart:** A positive edge starts the send process (irrespective of `bAuto`).
bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

rValueLimit: Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.45 FB_SEND_049_SNVT_angle_f
This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_angle_f.

**SNVT number:** 049.

**Description:** Angular distance (radian).

**VAR_INPUT**

```plaintext
wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

- **wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **rValue:** Min: -3.40E+38 / Max: 3.40E+38.
- **bStart:** A positive edge starts the send process (irrespective of *bAuto*).
- **bSendInit:** Automatically sends the value once when the PLC restarts.
- **bAuto:** Automatic sending [630] is selected when a value changes or the time *tMaxSendTime* has elapsed (polling).
- **rValueLimit:** Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.
- **tMinSendTime:** Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.
- **tMaxSendTime:** Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (*rValueLimit*) was not reached (polling). The value 0 disables this function.
- **bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

```plaintext
bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;
```

- **bBusy:** The *bBusy* output is TRUE as long as values are sent.
- **bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.
- **eError:** This output outputs an error code in the event of an error (see E_LON_ERROR [477]). *bError* goes TRUE at the same time.
- **dwErrorKL:** Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable *eError* has the value eKL6401_Error. *bError* goes TRUE at the same time.

**VAR_IN_OUT**

```plaintext
stLON_Com : ST_LON_Communication;
```

- **stLON_Com:** This structure is used to link FB_LON_KL6401 [27] with the send/receive function (see ST_LON_Communication [587]).
### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

#### 4.1.3.46  FB_SEND_050_SNVT_angle_vel_f

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_angle_vel_f.

**SNVT number:** 050.

**Description:** Angular velocity (radian / second).

**VAR_INPUT**

- **wNVIndex**: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

- **rValue**: Min: -3.40E+38 / Max: 3.40E+38.

- **bStart**: A positive edge starts the send process (irrespective of bAuto).

- **bSendInit**: Automatically sends the value once when the PLC restarts.

- **bAuto**: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

- **rValueLimit**: Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

- **tMinSendTime**: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

- **tMaxSendTime**: Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

- **bDisabled**: TRUE = deselection of the block.
VAR_OUTPUT
bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

**bBusy**: The `bBusy` output is TRUE as long as values are sent.

**bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.

**eError**: This output outputs an error code in the event of an error (see `E_LON_ERROR`). `bError` goes TRUE at the same time.

**dwErrorKL**: Error ID of the function block `FB_LON_KL6401` (see `dwErrorKL`). In this case the variable `eError` has the value eKL6401_Error. `bError` goes TRUE at the same time.

VAR_IN_OUT
stLON_Com : ST_LON_Communication;

**stLON_Com**: This structure is used to link `FB_LON_KL6401` with the send/receive function (see `ST_LON_Communication`).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.47 **FB_SEND_051_SNVT_count_f**

This function block sends the following LON output variable (nvo):

**SNVT Name**: SNVT_count_f.

**SNVT number**: 051.

**Description**: Absolute number (items).

VAR_INPUT

| wNVIndex : WORD; |
| rValue : REAL; |
| bStart : BOOL; |
| bSendInit : BOOL := bSendInitDefault; |
| bAuto : BOOL := bAutoDefault; |
| rValueLimit : REAL := 1; |
| tMinSendTime : TIME := tMinSendTimeDefault; |
| tMaxSendTime : TIME := tMaxSendTimeDefault; |
| bDisabled : BOOL := FALSE; |
wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

rValue: Min: 0 / Max: 3.40E+38.

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

rValueLimit: Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27](). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_count_inc_f.

**SNVT number:** 052.

**Description:** Increment counter (items(delta)).

**VAR_INPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.</td>
</tr>
<tr>
<td>rValue</td>
<td>Min: -3.40E+38 / Max: 3.40E+38.</td>
</tr>
<tr>
<td>bStart</td>
<td>A positive edge starts the send process (irrespective of bAuto).</td>
</tr>
<tr>
<td>bSendInit</td>
<td>Automatically sends the value once when the PLC restarts.</td>
</tr>
<tr>
<td>bAuto</td>
<td>Automatic sending is selected when a value changes or the time tMaxSendTime has elapsed (polling).</td>
</tr>
<tr>
<td>rValueLimit</td>
<td>Parameter for automatic sending. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>Parameter for automatic sending. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.</td>
</tr>
<tr>
<td>tMaxSendTime</td>
<td>Parameter for automatic sending. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.</td>
</tr>
<tr>
<td>bDisabled</td>
<td>TRUE = deselection of the block.</td>
</tr>
</tbody>
</table>

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td></td>
</tr>
<tr>
<td>bError</td>
<td></td>
</tr>
<tr>
<td>eError</td>
<td></td>
</tr>
<tr>
<td>dwErrorKL</td>
<td></td>
</tr>
</tbody>
</table>
bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.49 FB_SEND_053_SNVT_flow_f

This function block sends the following LON output variable (nvo):

SNVT Name: SNVT_flow_f.

SNVT number: 053.

Description: Volume flow (liters / second).

VAR_INPUT

wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

rValue: Min: -3.40E+38 / Max: 3.40E+38.

bStart: A positive edge starts the send process (irrespective of bAuto).
bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

rValueLimit: Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401 [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.50 FB_SEND_054_SNVT_length_f
This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT\_length\_f.

**SNVT number:** 054.

**Description:** Length (meter).

### VAR\_INPUT

| wNVIndex | : WORD; |
| rValue | : REAL; |
| bStart | : BOOL; |
| bSendInit | : BOOL := bSendInitDefault; |
| bAuto | : BOOL := bAutoDefault; |
| rValueLimit | : REAL := 1; |
| tMinSendTime | : TIME := tMinSendTimeDefault; |
| tMaxSendTime | : TIME := tMaxSendTimeDefault; |
| bDisabled | : BOOL := FALSE; |

- **wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

- **rValue:** Min: 0 / Max: 3.40E+38.

- **bStart:** A positive edge starts the send process (irrespective of bAuto).

- **bSendInit:** Automatically sends the value once when the PLC restarts.

- **bAuto:** Automatic sending ([630]) is selected when a value changes or the time tMaxSendTime has elapsed (polling).

- **rValueLimit:** Parameter for automatic sending ([630]). The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

- **tMinSendTime:** Parameter for automatic sending ([630]). A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

- **tMaxSendTime:** Parameter for automatic sending ([630]). The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

- **bDisabled:** TRUE = deselection of the block.

### VAR\_OUTPUT

| bBusy | : BOOL; |
| bError | : BOOL; |
| eError | : E\_LON\_ERROR; |
| dwErrorKL | : DWORD; |

- **bBusy:** The bBusy output is TRUE as long as values are sent.

- **bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

- **eError:** This output outputs an error code in the event of an error (see E\_LON\_ERROR [477]). bError goes TRUE at the same time.

- **dwErrorKL:** Error ID of the function block FB\_LON\_KL6401 ([27]) (see dwErrorKL [631]). In this case the variable eError has the value eKL6401\_Error. bError goes TRUE at the same time.

### VAR\_IN\_OUT

| stLON\_Com | : ST\_LON\_Communication; |

- **stLON\_Com:** This structure is used to link FB\_LON\_KL6401() ([27]) with the send/receive function (see ST\_LON\_Communication [587]).
### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

#### 4.1.3.51 FB_SEND_055_SNVT_lev_cont_f

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_lev_cont_f.

**SNVT number:** 055.

**Description:** Continuous change in value (% from maximum level).

**VAR_INPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.</td>
</tr>
<tr>
<td>rValue</td>
<td>Min: 0 / Max: 100.</td>
</tr>
<tr>
<td>bStart</td>
<td>A positive edge starts the send process (irrespective of bAuto).</td>
</tr>
<tr>
<td>bSendInit</td>
<td>Automatically sends the value once when the PLC restarts.</td>
</tr>
<tr>
<td>bAuto</td>
<td>Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).</td>
</tr>
<tr>
<td>rValueLimit</td>
<td>Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.</td>
</tr>
<tr>
<td>tMaxSendTime</td>
<td>Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.</td>
</tr>
<tr>
<td>bDisabled</td>
<td>TRUE = deselection of the block.</td>
</tr>
</tbody>
</table>
VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.
bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.
eError: This output outputs an error code in the event of an error (see E_LON_ERROR[477]). bError goes TRUE at the same time.
dwErrorKL: Error ID of the function block FB_LON_KL6401[27] (see dwErrorKL[631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401[27] with the send/receive function (see ST_LON_Communication[587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.52 FB_SEND_056_SNVT_mass_f

This function block sends the following LON output variable (nvo):

SNVT Name: SNVT_mass_f.

SNVT number: 056.

Description: Mass (gram).

VAR_INPUT

wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

rValue: Min: 0 / Max: 3.40E+38.

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time $tMaxSendTime$ has elapsed (polling).

rValueLimit: Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

$tMinSendTime$: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

$tMaxSendTime$: Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value ($rValueLimit$) was not reached (polling). The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

<table>
<thead>
<tr>
<th>bBusy</th>
<th>BOOL</th>
</tr>
</thead>
<tbody>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>DWORD</td>
</tr>
</tbody>
</table>

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

| stLON_Com : ST_LON_Communication |

stLON_Com: This structure is used to link FB_LON_KL6401 [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_power_f.

**SNVT number:** 057.

**Description:** Power (watt).

### VAR_INPUT

- wNVIndex : WORD;
- rValue : REAL;
- bStart : BOOL;
- bSendInit : BOOL := bSendInitDefault;
- bAuto : BOOL := bAutoDefault;
- rValueLimit : REAL := 1;
- tMinSendTime : TIME := tMinSendTimeDefault;
- tMaxSendTime : TIME := tMaxSendTimeDefault;
- bDisabled : BOOL := FALSE;

**wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**rValue:** Min: -3.40E+38 / Max: 3.40E+38.

**bStart:** A positive edge starts the send process (irrespective of \( bAuto \)).

**bSendInit:** Automatically sends the value once when the PLC restarts.

**bAuto:** Automatic sending \([630]\) is selected when a value changes or the time \( tMaxSendTime \) has elapsed (polling).

**rValueLimit:** Parameter for automatic sending \([630]\). The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

**tMinSendTime:** Parameter for automatic sending \([630]\). A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

**tMaxSendTime:** Parameter for automatic sending \([630]\). The value is sent when this time has elapsed at the latest, even if the minimum change in value (\( rValueLimit \)) was not reached (polling). The value 0 disables this function.

**bDisabled:** TRUE = deselection of the block.

### VAR_OUTPUT

- bBusy : BOOL;
- bError : BOOL;
- eError : E_LON_ERROR;
- dwErrorKL : DWORD;

The function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_power_f.

**SNVT number:** 057.

**Description:** Power (watt).
bBusy: The `bBusy` output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.

eError: This output outputs an error code in the event of an error (see `E_LON_ERROR`). `bError` goes TRUE at the same time.

dwErrorKL: Error ID of the function block `FB_LON_KL6401` (see `dwErrorKL`). In this case the variable `eError` has the value `eKL6401_Error`. `bError` goes TRUE at the same time.

VAR_IN_OUT

```plaintext
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure is used to link `FB_LON_KL6401` with the send/receive function (see `ST_LON_Communication`).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.54 FB_SEND_058_SNVT_ppm_f

```plaintext
FB_SEND_058_SNVT_ppm_f

- wNVIndex  bBusy
- rValue    bError
- bStart    eError
- bSendInit dwErrorKL
- bAuto
- rValueLimit
- tMinSendTime
- tMaxSendTime
- bDisabled
  stLON_Com
```

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_ppm_f.

**SNVT number:** 058.

**Description:** Concentration (ppm).

**VAR_INPUT**

```plaintext
wNVIndex : WORD;
rValue  : REAL;
bStart  : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto   : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

rValue: Min: 0 / Max: 3.40E+38.

bStart: A positive edge starts the send process (irrespective of `bAuto`).
bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

rValueLimit: Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enum types and structures, where sending takes place after each change in value.

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT
bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27](see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT
stLON_Com : ST_LON_Communication;

stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.55 FB_SEND_059_SNVT_press_f
Application
This function block sends the following LON output variable (nvo):

SNVT Name: SNVT_press_f.
SNVT number: 059.
Description: Pressure (overpressure) (pascal).

VAR_INPUT

wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

rValue: Min: -3.40E+38 / Max: 3.40E+38.

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

rValueLimit: Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401 [27] with the send/receive function (see ST_LON_Communication [587]).
### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

#### 4.1.3.56 FB_SEND_060_SNVT_res_f

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_res_f  
**SNVT number:** 060.  
**Description:** Electric resistance (ohm).

#### VAR_INPUT

<table>
<thead>
<tr>
<th>wNVIndex</th>
<th>rValue</th>
<th>bStart</th>
<th>bSendInit</th>
<th>bAuto</th>
<th>rValueLimit</th>
<th>tMinSendTime</th>
<th>tMaxSendTime</th>
<th>bDisabled</th>
</tr>
</thead>
</table>

- **wNVIndex**: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **rValue**: Min: 0 / Max: 3.40E+38.
- **bStart**: A positive edge starts the send process (irrespective of bAuto).
- **bSendInit**: Automatically sends the value once when the PLC restarts.
- **bAuto**: `Automatic sending` is selected when a value changes or the time `tMaxSendTime` has elapsed (polling).
- **rValueLimit**: Parameter for `automatic sending`. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.
- **tMinSendTime**: Parameter for `automatic sending`. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.
- **tMaxSendTime**: Parameter for `automatic sending`. The value is sent when this time has elapsed at the latest, even if the minimum change in value (`rValueLimit`) was not reached (polling). The value 0 disables this function.
- **bDisabled**: TRUE = deselection of the block.
**VAR_OUTPUT**

- **bBusy**: The *bBusy* output is TRUE as long as values are sent.
- **bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.
- **eError**: This output outputs an error code in the event of an error (see `E_LON_ERROR` [477]). *bError* goes TRUE at the same time.
- **dwErrorKL**: Error ID of the function block `FB_LON_KL6401` [27] (see `dwErrorKL` [631]). In this case the variable *eError* has the value eKL6401_Error. *bError* goes TRUE at the same time.

**VAR_IN_OUT**

- **stLON_Com**: This structure is used to link `FB_LON_KL6401` [27] with the send/receive function (see `ST_LON_Communication` [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.3.57 FB_SEND_061_SNVT_sound_db_f**

This function block sends the following LON output variable (nvo):

**SNVT Name**: SNVT_sound_db_f.

**SNVT number**: 061.

**Description**: Sound pressure level (dBspl).

**VAR_INPUT**

- **wNVIndex**: WORD;
- **rValue**: REAL;
- **bStart**: BOOL;
- **bSendInit**: BOOL := bSendInitDefault;
- **bAuto**: BOOL := bAutoDefault;
- **rValueLimit**: REAL := 1;
- **tMinSendTime**: TIME := tMinSendTimeDefault;
- **tMaxSendTime**: TIME := tMaxSendTimeDefault;
- **bDisabled**: BOOL := FALSE;
wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

rValue: Min: -3.40E+38 / Max: 3.40E+38.

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

rValueLimit: Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_speed_f.

**SNVT number:** 062.

**Description:** Linear velocity (meters / second).

**VAR_INPUT**

- wNVIndex : WORD;
- rValue : REAL;
- bStart : BOOL;
- bSendInit : BOOL := bSendInitDefault;
- bAuto : BOOL := bAutoDefault;
- rValueLimit : REAL := 1;
- tMinSendTime : TIME := tMinSendTimeDefault;
- tMaxSendTime : TIME := tMaxSendTimeDefault;
- bDisabled : BOOL := FALSE;

**wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**rValue:** Min: -3.40E+38 / Max: 3.40E+38.

**bStart:** A positive edge starts the send process (irrespective of bAuto).

**bSendInit:** Automatically sends the value once when the PLC restarts.

**bAuto:** Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

**rValueLimit:** Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

**tMinSendTime:** Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

**tMaxSendTime:** Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

**bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

- bBusy : BOOL;
- bError : BOOL;
- eError : E_LON_ERROR;
- dwErrorKL : DWORD;

FB_SEND_062_SNVT_speed_f
bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see _E_LON_ERROR_ [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401 [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.59 FB_SEND_063_SNVT_temp_f

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_temp_f.

**SNVT number:** 063.

**Description:** Temperature (°C).

VAR_INPUT

wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

rValue: Min: -273.17 / Max: 3.40E+38.

bStart: A positive edge starts the send process (irrespective of bAuto).
bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

rValueLimit: Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT
bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] () (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT
stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.60 FB_SEND_064_SNVT_time_f

```plaintext
FB_SEND_064_SNVT_time_f
- wNVIndex
- iValue
- bStart
- bSendInit
- bAuto
- rValueLimit
- tMinSendTime
- tMaxSendTime
- bDisabled
  - stLON_Com
```
This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_time_f.

**SNVT number:** 064.

**Description:** Elapsed time (seconds).

### VAR_INPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>WORD</td>
<td>Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.</td>
</tr>
<tr>
<td>rValue</td>
<td>REAL</td>
<td>Min: 0 / Max: 3.40E+38.</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL</td>
<td>A positive edge starts the send process (irrespective of bAuto).</td>
</tr>
<tr>
<td>bSendInit</td>
<td>BOOL</td>
<td>Automatically sends the value once when the PLC restarts.</td>
</tr>
<tr>
<td>bAuto</td>
<td>BOOL</td>
<td>Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).</td>
</tr>
<tr>
<td>rValueLimit</td>
<td>REAL</td>
<td>Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>TIME</td>
<td>Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.</td>
</tr>
<tr>
<td>tMaxSendTime</td>
<td>TIME</td>
<td>Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL</td>
<td>TRUE = deselection of the block.</td>
</tr>
</tbody>
</table>

### VAR_OUTPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>BOOL</td>
<td>The bBusy output is TRUE as long as values are sent.</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
<td>This output goes TRUE as soon as an error occurs. This error is described via the variable eError.</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
<td>This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>DWORD</td>
<td>Error ID of the function block FB_LON_KL6401 [271]) (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.</td>
</tr>
</tbody>
</table>

### VAR_IN_OUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stLON_Com</td>
<td>ST_LON_Communication</td>
<td>This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).</td>
</tr>
</tbody>
</table>
Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.61   **FB_SEND_065_SNVT_vol_f**

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_vol_f.

**SNVT number:** 065.

**Description:** Volume (liter).

**VAR_INPUT**

<table>
<thead>
<tr>
<th>wNVIndex</th>
<th>bBusy</th>
</tr>
</thead>
<tbody>
<tr>
<td>rValue</td>
<td>bError</td>
</tr>
<tr>
<td>bStart</td>
<td>aError</td>
</tr>
<tr>
<td>bSendInit</td>
<td>dwErrorKL</td>
</tr>
<tr>
<td>bAuto</td>
<td></td>
</tr>
<tr>
<td>rValueLimit</td>
<td></td>
</tr>
<tr>
<td>tMinSendTime</td>
<td></td>
</tr>
<tr>
<td>tMaxSendTime</td>
<td></td>
</tr>
<tr>
<td>bDisabled</td>
<td>tLON_Corn</td>
</tr>
</tbody>
</table>

wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

rValue: Min: 0 / Max: 3.40E+38.

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: **Automatic sending** [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

rValueLimit: Parameter for **automatic sending** [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

TMinSendTime: Parameter for **automatic sending** [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

TMaxSendTime: Parameter for **automatic sending** [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

bDisabled: TRUE = deselection of the block.
VAR OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 () (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.62 FB_SEND_066_SNVT_volt_f

This function block sends the following LON output variable (nvo):

SNVT Name: SNVT_volt_f.

SNVT number: 066.

Description: Electrical voltage (volt).

VAR_INPUT

wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit := bSendInitDefault;
bAuto := bAutoDefault;
rValueLimit := REAL := 1;
tMinSendTime := tMinSendTimeDefault;
tMaxSendTime := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
**Programming**

**wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**rValue:** Min: -3.40E+38 / Max: 3.40E+38.

**bStart:** A positive edge starts the send process (irrespective of bAuto).

**bSendInit:** Automatically sends the value once when the PLC restarts.

**bAuto:** Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

**rValueLimit:** Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

**tMinSendTime:** Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

**tMaxSendTime:** Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

**bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

```plaintext
bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;
```

**bBusy:** The bBusy output is TRUE as long as values are sent.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

**eError:** This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

**dwErrorKL:** Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

**VAR_IN_OUT**

```plaintext
stLON_Com : ST_LON_Communication;
```

**stLON_Com:** This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT\textunderscore btu\textunderscore f.

**SNVT number:** 067.

**Description:** Thermal energy (Btu).

**VAR\textunderscore INPUT**

- **wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **rValue:** Min: 0 / Max: 3.40E+38.
- **bStart:** A positive edge starts the send process (irrespective of \textit{bAuto}).
- **bSendInit:** Automatically sends the value once when the PLC restarts.
- **bAuto:** Automatic sending \([\item 630]\) is selected when a value changes or the time \(tMaxSendTime\) has elapsed (polling).
- **rValueLimit:** Parameter for automatic sending \([\item 630]\). The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.
- **tMinSendTime:** Parameter for automatic sending \([\item 630]\). A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.
- **tMaxSendTime:** Parameter for automatic sending \([\item 630]\). The value is sent when this time has elapsed at the latest, even if the minimum change in value (\(rValueLimit\)) was not reached (polling). The value 0 disables this function.
- **bDisabled:** TRUE = deselection of the block.

**VAR\textunderscore OUTPUT**

- **bBusy:** BOOL;
- **bError:** BOOL;
- **eError:** E\_LON\_ERROR;
- **dwErrorKL:** DWORD;
**bBusy:** The *bBusy* output is TRUE as long as values are sent.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

**eError:** This output outputs an error code in the event of an error (see `E_LON_ERROR[477]`). *bError* goes TRUE at the same time.

**dwErrorKL:** Error ID of the function block `FB_LON_KL6401[27]()` (see `dwErrorKL[631]`). In this case the variable *eError* has the value `eKL6401_Error`. *bError* goes TRUE at the same time.

**VAR_IN_OUT**

```plaintext
stLON_Com : ST_LON_Communication;
```

*stLON_Com:* This structure is used to link `FB_LON_KL6401[27]()` with the send/receive function (see `ST_LON_Communication[587]`).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.3.64**  
**FB_SEND_068_SNVT_elec_whr_f**

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_elec_whr_f.

**SNVT number:** 068.

**Description:** Electric energy (Watt / hour).

**VAR_INPUT**

```plaintext
wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**rValue:** Min: 0 / Max: 3.40E+38.

**bStart:** A positive edge starts the send process (irrespective of *bAuto*).
bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending \[630\] is selected when a value changes or the time \(tMaxSendTime\) has elapsed (polling).

rValueLimit: Parameter for automatic sending \[630\]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

\(tMinSendTime\): Parameter for automatic sending \[630\]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

\(tMaxSendTime\): Parameter for automatic sending \[630\]. The value is sent when this time has elapsed at the latest, even if the minimum change in value \(rValueLimit\) was not reached (polling). The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

| bBusy     : BOOL;   | bBusy: The \textit{bBusy} output is TRUE as long as values are sent. |
| bError    : BOOL;   | bError: This output goes TRUE as soon as an error occurs. This error is described via the variable \textit{eError}. |
| eError    : E_LON_ERROR; | eError: This output outputs an error code in the event of an error (see \textit{E_LON_ERROR} \[477\]). bError goes TRUE at the same time. |
| dwErrorKL : DWORD;  | dwErrorKL: Error ID of the function block \textit{FB_LON_KL6401} \[27\] () (see \textit{dwErrorKL} \[631\]). In this case the variable \textit{eError} has the value eKL6401_Error. bError goes TRUE at the same time. |

VAR_IN_OUT

| stLON_Com : ST_LON_Communication; |
| stLON_Com: This structure is used to link \textit{FB_LON_KL6401} \[27\] with the send/receive function (see \textit{ST_LON_Communication} \[587\]). |

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.65 \textbf{FB_SEND_069_SNVT_config_src}

This function block sends the following LON output variable (nvo):
SNVT Name: SNVT_config_src.

SNVT number: 069.

Description: Configuration properties (name of the configuration source (0=own, 1=external)).

**VAR_INPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.</td>
</tr>
<tr>
<td>eValue</td>
<td>Enum to be sent (see E_LON_config_source_t [509]).</td>
</tr>
<tr>
<td>bStart</td>
<td>A positive edge starts the send process (irrespective of bAuto).</td>
</tr>
<tr>
<td>bSendInit</td>
<td>Automatically sends the value once when the PLC restarts.</td>
</tr>
<tr>
<td>bAuto</td>
<td>Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.</td>
</tr>
<tr>
<td>tMaxSendTime</td>
<td>Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.</td>
</tr>
<tr>
<td>bDisabled</td>
<td>TRUE = deselection of the block.</td>
</tr>
</tbody>
</table>

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>The bBusy output is TRUE as long as values are sent.</td>
</tr>
<tr>
<td>bError</td>
<td>This output goes TRUE as soon as an error occurs. This error is described via the variable eError.</td>
</tr>
<tr>
<td>eError</td>
<td>This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.</td>
</tr>
</tbody>
</table>

**VAR_IN_OUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stLON_Com</td>
<td>This structure is used to link FB_LON_KL6401 [27] with the send/receive function (see ST_LON_Communication [587]).</td>
</tr>
</tbody>
</table>

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_color.

**SNVT number:** 070.

**Description:** Color according to CIE standard (L*, a*, b).

**VAR_INPUT**

- **wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per terminal (KL6401). Values between 0 and 61 are possible.
- **stValue:** Structure of the data to be sent.
  - *rL_star:* Min: 0 / Max: 100.0 / L*
  - *rA_star:* Min: -200.0 / Max: 200.0 / a*
  - *rB_star:* Min: -200.0 / Max: 200.0 / b*
- **bStart:** The function block starts sending when these variables have a positive edge (irrespective of "bAuto").
- **bSendInit:** The function block automatically sends its values once after a PLC restart.
- **bAuto:** Automatic sending is selected when a value changes or the time "tMaxSendTime" has elapsed (polling).
- **tMinSendTime:** Parameter for automatic sending. A new value is only sent once a minimum time has elapsed. This prevents continuous sending.
- **tMaxSendTime:** Parameter for automatic sending. The value is sent when this time has elapsed, even if the minimum change in value (ValueLimit) was not reached (polling). The value 0 disables this function.
- **bDisabled:** TRUE = deselection of the block
  - **wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
  - **stValue:** Structure of the data to be sent (see **ST_LON_SNVT_color**).
  - **bStart:** A positive edge starts the send process (irrespective of bAuto).
bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

| bBusy   | BOOL |
| bError  | BOOL |
| eError  | E_LON_ERROR |
| dwErrorKL | DWORD |

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

| stLON_Com : ST_LON_Communication; |

stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.67 FB_SEND_071_SNVT_grammage

This function block sends the following LON output variable (nvo):

SNVT Name: SNVT_grammage.

SNVT number: 071.
Description: American dimension for paper weight and density (gram/m2).

VAR_INPUT

wNVIndex : WORD;
rValue : REAL;
bStart : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

rValue: Min: 0 / Max: 6553.5.

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

rValueLimit: Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com: This structure is used to link FB_LON_KL6401 [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_grammage_f.

**SNVT number:** 072.

**Description:** American dimension for paper weight and density (gram/m2).

### VAR_INPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>WORD</td>
<td>Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.</td>
</tr>
<tr>
<td>rValue</td>
<td>REAL</td>
<td>Min: 0 / Max: 3.40E+38.</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL</td>
<td>A positive edge starts the send process (irrespective of bAuto).</td>
</tr>
<tr>
<td>bSendInit</td>
<td>BOOL</td>
<td>Automatically sends the value once when the PLC restarts.</td>
</tr>
<tr>
<td>bAuto</td>
<td>BOOL</td>
<td>Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).</td>
</tr>
<tr>
<td>rValueLimit</td>
<td>REAL</td>
<td>Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>TIME</td>
<td>Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.</td>
</tr>
<tr>
<td>tMaxSendTime</td>
<td>TIME</td>
<td>Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL</td>
<td>TRUE = deselection of the block.</td>
</tr>
</tbody>
</table>

### VAR_OUTPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>BOOL</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>DWORD</td>
</tr>
</tbody>
</table>

---

**FB_SEND_072_SNVT_grammage_f**
bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401 [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.69 FB_SEND_073_SNVT_file_req

This function block sends the following LON output variable (nvo):

SNVT Name: SNVT_file_req.

SNVT number: 073.

Description: File request.

VAR_INPUT

wNVIndex : WORD;
stValue : ST_LON_SNVT_file_req;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

stValue: Structure of the data to be sent (see ST_LON_SNVT_file_req [603]).

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.
bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

**VAR_OUTPUT**

| bBusy     : BOOL; |
| bError    : BOOL; |
| eError    : E_LON_ERROR; |
| dwErrorKL : DWORD; |

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

**VAR_IN_OUT**

| stLON_Com : ST_LON_Communication; |

stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.3.70 FB_SEND_074_SNVT_file_status**

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_file_status.

**SNVT number:** 074.

**Description:** File information (part of the LONWORKS file transfer protocol (LW-FTP)).
VAR_INPUT

wNVIndex : WORD;
stValue : ST_LON_SNVT_file_status;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

stValue: Structure of the data to be sent (see ST_LON_SNVT_file_req). If stValue.eStatus = eLON_FS_LOOKUP_OK (1), then the structure stValue.stAddr.stDescriptor is sent. Otherwise stValue.stAddr.stAddress is sent.

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending is selected when a value changes or the time tMaxSendTime has elapsed (polling).

tMinSendTime: Parameter for automatic sending. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401(). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com: This structure is used to link FB_LON_KL6401() with the send/receive function (see ST_LON_Communication).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v.3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_freq_f.

**SNVT number:** 075.

**Description:** Frequency (hertz).

### VAR_INPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>WORD</td>
<td>Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.</td>
</tr>
<tr>
<td>rValue</td>
<td>REAL</td>
<td>Min: 0 / Max: 3.40E+38.</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL</td>
<td>A positive edge starts the send process (irrespective of bAuto).</td>
</tr>
<tr>
<td>bSendInit</td>
<td>BOOL</td>
<td>Automatically sends the value once when the PLC restarts.</td>
</tr>
<tr>
<td>bAuto</td>
<td>BOOL</td>
<td>Automatic sending is selected when a value changes or the time tMaxSendTime has elapsed (polling).</td>
</tr>
<tr>
<td>rValueLimit</td>
<td>REAL</td>
<td>Parameter for automatic sending. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>TIME</td>
<td>Parameter for automatic sending. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.</td>
</tr>
<tr>
<td>tMaxSendTime</td>
<td>TIME</td>
<td>Parameter for automatic sending. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL</td>
<td>TRUE = deselection of the block.</td>
</tr>
</tbody>
</table>

### VAR_OUTPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>BOOL</td>
<td></td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
<td></td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
<td></td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>DWORD</td>
<td></td>
</tr>
</tbody>
</table>
**bBusy**: The *bBusy* output is TRUE as long as values are sent.

**bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

**eError**: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). *bError* goes TRUE at the same time.

**dwErrorKL**: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable *eError* has the value eKL6401_Error. *bError* goes TRUE at the same time.

**VAR_IN_OUT**

```plaintext
stLON_Com : ST_LON_Communication;
```

**stLON_Com**: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.72 **FB_SEND_076_SNVT_freq_hz**

This function block sends the following LON output variable (nvo):

**SNVT Name**: SNVT_freq_hz.

**SNVT number**: 076.

**Description**: Frequency (hertz).

**VAR_INPUT**

```plaintext
wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex**: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**rValue**: Min: 0 / Max: 6553.5.

**bStart**: A positive edge starts the send process (irrespective of *bAuto*).
bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

rValueLimit: Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

<table>
<thead>
<tr>
<th>bBusy</th>
<th>BOOL</th>
</tr>
</thead>
<tbody>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>DWORD</td>
</tr>
</tbody>
</table>

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from 3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.73 FB_SEND_077_SNVT_freq_kilohz

```
FB_SEND_077_SNVT_freq_kilohz
  wnVIndex   bBusy
  rValue     bError
  bStart     eError
  bSendInit  dwErrorKL
  bAuto      
  rValueLimit
  tMinSendTime
  tMaxSendTime
  bDisabled
  stLON_Com
```
This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_freq_kilohz.

**SNVT number:** 077.

**Description:** Frequency (kilohertz).

**VAR_INPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.</td>
</tr>
<tr>
<td>rValue</td>
<td>Min: 0 / Max: 6553.5.</td>
</tr>
<tr>
<td>bStart</td>
<td>A positive edge starts the send process (irrespective of bAuto).</td>
</tr>
<tr>
<td>bSendInit</td>
<td>Automatically sends the value once when the PLC restarts.</td>
</tr>
<tr>
<td>bAuto</td>
<td>Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).</td>
</tr>
<tr>
<td>rValueLimit</td>
<td>Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.</td>
</tr>
<tr>
<td>tMaxSendTime</td>
<td>Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.</td>
</tr>
<tr>
<td>bDisabled</td>
<td>TRUE = deselection of the block.</td>
</tr>
</tbody>
</table>

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>The bBusy output is TRUE as long as values are sent.</td>
</tr>
<tr>
<td>bError</td>
<td>This output goes TRUE as soon as an error occurs. This error is described via the variable eError.</td>
</tr>
<tr>
<td>eError</td>
<td>This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>Error ID of the function block FB_LON_KL6401 [271] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.</td>
</tr>
</tbody>
</table>

**VAR_IN_OUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stLON_Com</td>
<td>This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).</td>
</tr>
</tbody>
</table>
Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.74 FB_SEND_078_SNVT_freq_milhz

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_freq_milhz.

**SNVT number:** 078.

**Description:** Frequency (millihertz).

### VAR_INPUT

- `wNVIndex`: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- `rValue`: Min: 0 / Max: 6553.5.
- `bStart`: A positive edge starts the send process (irrespective of `bAuto`).
- `bSendInit`: Automatically sends the value once when the PLC restarts.
- `bAuto`: Automatic sending [630] is selected when a value changes or the time `tMaxSendTime` has elapsed (polling).
- `rValueLimit`: Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.
- `tMinSendTime`: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.
- `tMaxSendTime`: Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (`rValueLimit`) was not reached (polling). The value 0 disables this function.
- `bDisabled`: TRUE = deselection of the block.
VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

**bBusy:** The *bBusy* output is TRUE as long as values are sent.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable *eError*. 

**eError:** This output outputs an error code in the event of an error (see `E_LON_ERROR [477]`). *bError* goes TRUE at the same time.

**dwErrorKL:** Error ID of the function block `FB_LON_KL6401 [27]` (see `dwErrorKL [631]`). In this case the variable *eError* has the value eKL6401_Error. *bError* goes TRUE at the same time.

VAR_IN_OUT

stLON_Comm : ST_LON_Communication;

**stLON_Comm:** This structure is used to link `FB_LON_KL6401 [27]` with the send/receive function (see `ST_LON_Communication [587]`).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.75 **FB_SEND_079_SNVT_lux**

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_lux.

**SNVT number:** 079.

**Description:** Illuminance (lux) 1 lux = 1 lumen/m².

VAR_INPUT

wNVIndex : WORD;
uiValue : UINT;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
uiValueLimit : UINT := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

uiValue: Min: 0 / Max: 65535.

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

uiValueLimit: Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (uiValueLimit) was not reached (polling). The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_lev_percent.

**SNVT number:** 081.

**Description:** Percentage value in 0.005% steps with sign.

### VAR_INPUT

- **wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **rValue:** Min: -163.84 / Max: 163.835.
- **bStart:** A positive edge starts the send process (irrespective of **bAuto**).
- **bSendInit:** Automatically sends the value once when the PLC restarts.
- **bAuto:** Automatic sending is selected when a value changes or the time **tMaxSendTime** has elapsed (polling).
- **rValueLimit:** Parameter for automatic sending. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.
- **tMinSendTime:** Parameter for automatic sending. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.
- **tMaxSendTime:** Parameter for automatic sending. The value is sent when this time has elapsed at the latest, even if the minimum change in value (**rValueLimit**) was not reached (polling). The value 0 disables this function.
- **bDisabled:** TRUE = deselection of the block.

### VAR_OUTPUT

- **bBusy:**
- **bError:**
- **eError:**
- **dwErrorKL:**
bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.77 FB_SEND_082_SNVT_multiplier

This function block sends the following LON output variable (nvo):

SNVT Name: SNVT_multiplier.

SNVT number: 082.

Description: Multiplier in 0.0005 steps (16-bit unsigned value).

VAR_INPUT

wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

rValue: Min: 0 / Max: 32.7675.

bStart: A positive edge starts the send process (irrespective of bAuto).
**bSendInit:** Automatically sends the value once when the PLC restarts.

**bAuto:** Automatic sending. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

**rValueLimit:** Parameter for automatic sending. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

**tMinSendTime:** Parameter for automatic sending. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

**tMaxSendTime:** Parameter for automatic sending. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

**bDisabled:** TRUE = deselection of the block.

### VAR_OUTPUT

- **bBusy**: The *bBusy* output is TRUE as long as values are sent.
- **bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.
- **eError**: This output outputs an error code in the event of an error (see *E_LON_ERROR*). bError goes TRUE at the same time.
- **dwErrorKL**: Error ID of the function block FB_LON_KL6401. In this case the variable *eError* has the value eKL6401_Error. bError goes TRUE at the same time.

### VAR_IN_OUT

- **stLON_Com**: This structure is used to link FB_LON_KL6401 with the send/receive function (see ST_LON_Communication).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.3.78 FB_SEND_083_SNVT_state

This function block sends the following LON output variable (nvo):
SNVT Name: SNVT_state.

SNVT number: 083.

Description: Status information (16 individual bit values). Each status is a Boolean value.

**VAR_INPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>WORD</td>
</tr>
<tr>
<td>arrValue</td>
<td>ARRAY [0..15] OF BOOL</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL</td>
</tr>
<tr>
<td>bSendInit</td>
<td>BOOL := bSendInitDefault;</td>
</tr>
<tr>
<td>bAuto</td>
<td>BOOL := bAutoDefault;</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>TIME := tMinSendTimeDefault;</td>
</tr>
<tr>
<td>tMaxSendTime</td>
<td>TIME := tMaxSendTimeDefault;</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL := FALSE;</td>
</tr>
</tbody>
</table>

**wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**arrValue:** 0-15 bit.

**bStart:** A positive edge starts the send process (irrespective of bAuto).

**bSendInit:** Automatically sends the value once when the PLC restarts.

**bAuto:** Automatic sending is selected when a value changes or the time tMaxSendTime has elapsed (polling).

**tMinSendTime:** Parameter for automatic sending. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

**tMaxSendTime:** Parameter for automatic sending. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

**bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>BOOL</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>DWORD</td>
</tr>
</tbody>
</table>

**bBusy:** The bBusy output is TRUE as long as values are sent.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

**eError:** This output outputs an error code in the event of an error (see E_LON_ERROR). bError goes TRUE at the same time.

**dwErrorKL:** Error ID of the function block FB_LON_KL6401() (see dwErrorKL). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

**VAR_IN_OUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>stLON_Com</td>
<td>ST_LON_Communication;</td>
</tr>
</tbody>
</table>

**stLON_Com:** This structure is used to link FB_LON_KL6401() with the send/receive function (see ST_LON_Communication).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_time_stamp.

**SNVT number:** 084.

**Description:** Time stamp (year, month, day, hour, minute, second).

### VAR_INPUT

- `wNVIndex`: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- `stValue`: Structure of the data to be sent (see TIMESTRUCT). The structure variables `wDayOfWeek` and `wMilliseconds` are not valid here. These values are not transferred.
- `bStart`: A positive edge starts the send process (irrespective of `bAuto`).
- `bSendInit`: Automatically sends the value once when the PLC restarts.
- `bAuto`: **Automatic sending** is selected when a value changes or the time `tMaxSendTime` has elapsed (polling).
- `tMinSendTime`: Parameter for **automatic sending**. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.
- `tMaxSendTime`: Parameter for **automatic sending**. The value is sent after this time has elapsed at the latest. The value 0 disables this function.
- `bDisabled`: TRUE = deselection of the block.

### VAR_OUTPUT

- `bBusy`: The `bBusy` output is TRUE as long as values are sent.
- `bError`: This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.
- `eError`: This output outputs an error code in the event of an error (see **E_LON_ERROR**). `bError` goes TRUE at the same time.
**dwErrorKL:** Error ID of the function block `FB_LON_KL6401()` (see `dwErrorKL [27]`). In this case the variable `eError` has the value `eKL6401_Error`. `bError` goes TRUE at the same time.

**VAR_IN_OUT**

```
stLON_Com : ST_LON_Communication;
```

`stLON_Com`: This structure is used to link `FB_LON_KL6401()` with the send/receive function (see `ST_LON_Communication [587]`).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.3.80 **FB_SEND_085_SNVT_zerospan**

```
VAR_INPUT

wNVIndex : WORD;
stValue : ST_LON_SNVT_zerospan;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

- **wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **stValue:** Structure of the data to be sent (see `ST_LON_SNVT_zerospan [621]`).
- **bStart:** A positive edge starts the send process (irrespective of `bAuto`).
- **bSendInit:** Automatically sends the value once when the PLC restarts.
- **bAuto:** Automatic sending [630] is selected when a value changes or the time `tMaxSendTime` has elapsed (polling).
- **tMinSendTime:** Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_zerospan

**SNVT number:** 085

**Description:** Zero point and proportionality factor. Linear transformation parameters: multiply with proportionality factor, then add origin offset.
**tMaxSendTime**: Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

**bDisabled**: TRUE = deselection of the block.

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>The bBusy output is TRUE as long as values are sent.</td>
</tr>
<tr>
<td>bError</td>
<td>This output goes TRUE as soon as an error occurs. This error is described via the variable eError.</td>
</tr>
<tr>
<td>eError</td>
<td>This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.</td>
</tr>
</tbody>
</table>

**VAR_IN_OUT**

<table>
<thead>
<tr>
<th>Structure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stLON_Com</td>
<td>This structure is used to link FB_LON_KL6401 [27] with the send/receive function (see ST_LON_Communication [587]).</td>
</tr>
</tbody>
</table>

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.3.81 FB_SEND_086_SNVT_magcard**

This function block sends the following LON output variable (nvo):

**SNVT Name**: SNVT_magcard.

**SNVT number**: 086.

**Description**: ISO 7811 (40 hexadecimal numbers). Data according to ISO 7811 standard for magnetic card readers.

**VAR_INPUT**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>WORD;</td>
</tr>
<tr>
<td>arrValue</td>
<td>ARRAY [0..40] OF BYTE;</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL;</td>
</tr>
<tr>
<td>bSendInit</td>
<td>BOOL := bSendInitDefault;</td>
</tr>
<tr>
<td>bAuto</td>
<td>BOOL := bAutoDefault;</td>
</tr>
</tbody>
</table>
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

arrValue: 1 - 40 bytes.

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27]() (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT\_elapsed\_tm.

**SNVT number:** 087.

**Description:** Elapsed time (day, hour, minute, second, millisecond).

### VAR\_INPUT

- `wNVIndex` : WORD;
- `stValue` : ST\_LON\_SNVT\_elapsed\_tm;
- `bStart` : BOOL;
- `bSendInit` : BOOL := bSendInitDefault;
- `bAuto` : BOOL := bAutoDefault;
- `tMinSendTime` : TIME := tMinSendTimeDefault;
- `tMaxSendTime` : TIME := tMaxSendTimeDefault;
- `bDisabled` : BOOL := FALSE;

- `wNVIndex`: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

- `stValue`: Structure of the data to be sent (see ST\_LON\_SNVT\_elapsed\_tm[599]).

- `bStart`: A positive edge starts the send process (irrespective of `bAuto`).

- `bSendInit`: Automatically sends the value once when the PLC restarts.

- `bAuto`: **Automatic sending[630]** is selected when a value changes or the time `tMaxSendTime` has elapsed (polling).

- `tMinSendTime`: Parameter for **automatic sending[630]**. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

- `tMaxSendTime`: Parameter for **automatic sending[630]**. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

- `bDisabled`: TRUE = deselection of the block.

### VAR\_OUTPUT

- `bBusy` : BOOL;
- `bError` : BOOL;
- `eError` : E\_LON\_ERROR;
- `dwErrorKL` : DWORD;

- `bBusy`: The `bBusy` output is TRUE as long as values are sent.

- `bError`: This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.

- `eError`: This output outputs an error code in the event of an error (see E\_LON\_ERROR[477]). `bError` goes TRUE at the same time.
**dwErrorKL**: Error ID of the function block **FB_LON_KL6401** (see **dwErrorKL**). In this case the variable **eError** has the value eKL6401_Error. **bError** goes TRUE at the same time.

**VAR_IN_OUT**

```plaintext
stLON_Com : ST_LON_Communication;
```

**stLON_Com**: This structure is used to link **FB_LON_KL6401** with the send/receive function (see **ST_LON_Communication**).  

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.3.83 FB_SEND_088_SNVT_alarm

This function block sends the following LON output variable (nvo):

**SNVT Name**: SNVT_alarm.  
**SNVT number**: 088.  
**Description**: Alarm status.

**VAR_INPUT**

```plaintext
wNVIndex : WORD;
stValue : ST_LON_SNVT_alarm;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex**: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.  
**stValue**: Structure of the data to be sent (see **ST_LON_SNVT_alarm**).  
**bStart**: A positive edge starts the send process (irrespective of **bAuto**).  
**bSendInit**: Automatically sends the value once when the PLC restarts.  
**bAuto**: Automatic sending is selected when a value changes or the time **tMaxSendTime** has elapsed (polling).  
**tMinSendTime**: Parameter for automatic sending. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.
**tMaxSendTime**: Parameter for **automatic sending** [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

**bDisabled**: TRUE = deselection of the block.

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>BOOL; The <strong>bBusy</strong> output is TRUE as long as values are sent.</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL; This output goes TRUE as soon as an error occurs. This error is described via the variable <strong>eError</strong>.</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR;</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>DWORD;</td>
</tr>
</tbody>
</table>

**VAR_IN_OUT**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stLON_Com</td>
<td>ST_LON_Communication;</td>
</tr>
</tbody>
</table>

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**FB_SEND_089_SNVT_currency**

This function block sends the following LON output variable (nvo):

**SNVT Name**: SNVT_currency.

**SNVT number**: 089.

**Description**: Currency (unit (euros,...), multiplier, value).

**VAR_INPUT**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>WORD;</td>
</tr>
<tr>
<td>stValue</td>
<td>ST_LON_SNVT_currency;</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL;</td>
</tr>
<tr>
<td>bSendInit</td>
<td>BOOL := bSendInitDefault;</td>
</tr>
<tr>
<td>bAuto</td>
<td>BOOL := bAutoDefault;</td>
</tr>
</tbody>
</table>
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;

**wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**stValue:** Structure of the data to be sent (see ST_LON_SNVT_currency [5.97]).

**bStart:** A positive edge starts the send process (irrespective of bAuto).

**bSendInit:** Automatically sends the value once when the PLC restarts.

**bAuto:** Automatic sending [6.30] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

**tMinSendTime:** Parameter for automatic sending [6.30]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

**tMaxSendTime:** Parameter for automatic sending [6.30]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

**bDisabled:** TRUE = deselection of the block.

### VAR_OUTPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy    : BOOL;</td>
<td></td>
</tr>
<tr>
<td>bError   : BOOL;</td>
<td></td>
</tr>
<tr>
<td>eError   : E_LON_ERROR;</td>
<td></td>
</tr>
<tr>
<td>dwErrorKL : DWORD;</td>
<td></td>
</tr>
</tbody>
</table>

**bBusy:** The bBusy output is TRUE as long as values are sent.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

**eError:** This output outputs an error code in the event of an error (see E_LON_ERROR [4.77]). bError goes TRUE at the same time.

**dwErrorKL:** Error ID of the function block FB_LON_KL6401 [27]() (see dwErrorKL [6.31]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

### VAR_IN_OUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>stLON_Com : ST_LON_Communication;</td>
<td></td>
</tr>
</tbody>
</table>

**stLON_Com** : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [5.87]).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_file_pos.

**SNVT number:** 090.

**Description:** File position (pointer, length).

### VAR_INPUT

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td></td>
<td>Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.</td>
</tr>
<tr>
<td>stValue</td>
<td></td>
<td>Structure of the data to be sent (see ST_LON_SNVT_file_pos [602]).</td>
</tr>
<tr>
<td>bStart</td>
<td></td>
<td>A positive edge starts the send process (irrespective of bAuto).</td>
</tr>
<tr>
<td>bSendInit</td>
<td></td>
<td>Automatically sends the value once when the PLC restarts.</td>
</tr>
<tr>
<td>bAuto</td>
<td></td>
<td>Automatic sending [630] is selected when a value changes or the time ( t_{\text{MaxSendTime}} ) has elapsed (polling).</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td></td>
<td>Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.</td>
</tr>
<tr>
<td>tMaxSendTime</td>
<td></td>
<td>Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.</td>
</tr>
<tr>
<td>bDisabled</td>
<td></td>
<td>TRUE = deselection of the block.</td>
</tr>
</tbody>
</table>

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td></td>
<td>The ( b_{\text{Busy}} ) output is TRUE as long as values are sent.</td>
</tr>
<tr>
<td>bError</td>
<td></td>
<td>This output goes TRUE as soon as an error occurs. This error is described via the variable ( e_{\text{Error}} ).</td>
</tr>
<tr>
<td>eError</td>
<td></td>
<td>This output outputs an error code in the event of an error (see E_LON_ERROR [477]). ( b_{\text{Error}} ) goes TRUE at the same time.</td>
</tr>
</tbody>
</table>
**Programming**

**T1000**

**Version: 1.5**

---

**dwErrorKL**: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

**VAR_IN_OUT**

```plaintext
stLON_Com : ST_LON_Communication;
```

**stLON_Com**: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

---

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

---

### 4.1.3.86 **FB_SEND_091_SNVT_muldiv**

```plaintext
VAR_INPUT

wNVIndex : WORD;
stValue : ST_LON_SNVT_muldiv;
bStart : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

This function block sends the following LON output variable (nvo):

**SNVT Name**: SNVT_muldiv.

**SNVT number**: 091.

**Description**: Gain factor/attenuation factor (multiplier, divisor).

---

### VAR_INPUT

- **wNVIndex**: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

- **stValue**: Structure of the data to be sent (see ST_LON_SNVT_muldiv [608]).

- **bStart**: A positive edge starts the send process (irrespective of bAuto).

- **bSendInit**: Automatically sends the value once when the PLC restarts.

- **bAuto**: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

- **tMinSendTime**: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.
Programming

**tMaxSendTime**: Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

**bDisabled**: TRUE = deselection of the block.

### VAR_OUTPUT

<table>
<thead>
<tr>
<th>bBusy</th>
<th>BOOL;</th>
</tr>
</thead>
<tbody>
<tr>
<td>bError</td>
<td>BOOL;</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR;</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>DWORD;</td>
</tr>
</tbody>
</table>

**bBusy**: The *bBusy* output is TRUE as long as values are sent.

**bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

**eError**: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

**dwErrorKL**: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable *eError* has the value eKL6401_Error. bError goes TRUE at the same time.

### VAR_IN_OUT

| stLON_Com    | ST_LON_Communication; |

**stLON_Com**: This structure is used to link FB_LON_KL6401 [27] with the send/receive function (see ST_LON_Communication [587]).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.3.87 FB_SEND_092_SNVT_obj_request

This function block sends the following LON output variable (nvo):

**SNVT Name**: SNVT_obj_request.

**SNVT number**: 092.

**Description**: Function selection (ID, request).

### VAR_INPUT

<table>
<thead>
<tr>
<th>wNVIndex</th>
<th>WORD;</th>
</tr>
</thead>
<tbody>
<tr>
<td>stValue</td>
<td>ST_LON_SNVT_obj_request;</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL;</td>
</tr>
<tr>
<td>bSendInit</td>
<td>BOOL := bSendInitDefault;</td>
</tr>
<tr>
<td>bAuto</td>
<td>BOOL := bAutoDefault;</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td></td>
</tr>
<tr>
<td>tMaxSendTime</td>
<td></td>
</tr>
<tr>
<td>bDisabled</td>
<td></td>
</tr>
<tr>
<td><strong>stLON_Com</strong></td>
<td></td>
</tr>
</tbody>
</table>

This function block sends the following LON output variable (nvo):
### tMinSendTime

`: TIME := tMinSendTimeDefault;`

### tMaxSendTime

`: TIME := tMaxSendTimeDefault;`

### bDisabled

`: BOOL := FALSE;`

### wNVIndex

Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

### stValue

Structure of the data to be sent (see ST_LON_SNVT_obj_request [609]).

### bStart

A positive edge starts the send process (irrespective of bAuto).

### bSendInit

Automatically sends the value once when the PLC restarts.

### bAuto

Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

### tMinSendTime

Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

### tMaxSendTime

Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

### bDisabled

TRUE = deselection of the block.

### VAR_OUTPUT

- **bBusy**: The bBusy output is TRUE as long as values are sent.
- **bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.
- **eError**: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.
- **dwErrorKL**: Error ID of the function block FB_LON_KL6401 [27]() (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

### VAR_IN_OUT

- **stLON_Com**: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_obj_status.

**SNVT number:** 093.

**Description:** Object status (ID, status (4 byte)).

**VAR_INPUT**

- `wNVIndex`: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per terminal (KL6401). Values between 0 and 61 are possible.
- `stValue.uiObject_id`: Min: 0 / Max: 65535 / Object ID (object index).
- `stValue.bInvalid_id`: Invalid-ID flag (boolean).
- `stValue.bInvalid_request`: Invalid-request flag (boolean).
- `stValue.bDisabled`: Disabled flag (boolean).
- `stValue.bOpen_circuit`: Open-circuit flag (boolean).
- `stValue.bMechanical_fault`: Mechanical-fault flag (boolean).
- `stValue.bFeedback_failure`: Feedback-failure flag (boolean).
- `stValue.bOver_range`: Over-range flag (boolean).
- `stValue.bUnder_range`: Under-range flag (boolean).
- `stValue.bElectrical_fault`: Electrical-fault flag (boolean).
- `stValue.bUnable_to_measure`: Unable-to-measure flag (boolean).
- `stValue.bComm_failure`: Communications-failure flag (boolean).
- `stValue.bFail_self_test`: Failed-self-test flag (boolean).
- `stValue.bSelf_test_in_progress`: Self-test-in-progress flag (boolean).
stValue. bLocked_out: Locked-out flag (boolean).
stValue. bln_alarm: Input-alarm flag (boolean).
stValue. bln_override: Input-override flag (boolean).
stValue. bProgramming_mode: Programming-mode flag (boolean).
stValue. bProgramming_fail: Programming-fail flag (boolean).
stValue. bAlarm_notify_disabled: Alarm-notify-disabled flag (boolean).
stValue. bReset_complete: Reset (boolean).
stValue. byReserved2: This field is reserved.
bStart: The function block starts sending when these variables have a positive edge (irrespective of "bAuto").
bSendInit: The function block automatically sends its values once after a PLC restart.
bAuto: Automatic sending \((\text{630})\) is selected when a value changes or the time "tMaxSendTime" has elapsed (polling).
tMinSendTime: Parameter for automatic sending \((\text{630})\). A new value is only sent once a minimum time has elapsed. This prevents continuous sending.
tMaxSendTime: Parameter for automatic sending \((\text{630})\). The value is sent when this time has elapsed, even if the minimum change in value (ValueLimit) was not reached (polling). The value 0 disables this function.
bDisabled: TRUE = deselection of the block.
wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
stValue: Structure of the data to be sent (see **ST_LON_SNVT_obj_status** \((\text{609})\)).
bStart: A positive edge starts the send process (irrespective of bAuto).
bSendInit: Automatically sends the value once when the PLC restarts.
bAuto: Automatic sending \((\text{630})\) is selected when a value changes or the time \(tMaxSendTime\) has elapsed (polling).
tMinSendTime: Parameter for automatic sending \((\text{630})\). A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.
tMaxSendTime: Parameter for automatic sending \((\text{630})\). The value is sent after this time has elapsed at the latest. The value 0 disables this function.
bDisabled: TRUE = deselection of the block.

**VAR_OUTPUT**

```plaintext
bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;
```

**bBusy**: The **bBusy** output is TRUE as long as values are sent.

**bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable **eError**.
**eError**: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). `bError` goes TRUE at the same time.

**dwErrorKL**: Error ID of the function block `FB_LON_KL6401` (27) (see `dwErrorKL` [631]). In this case the variable `eError` has the value `eKL6401_Error`. `bError` goes TRUE at the same time.

### VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

`stLON_Com`: This structure is used to link `FB_LON_KL6401() [27]` with the send/receive function (see `ST_LON_Communication` [587]).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

#### 4.1.3.89  **FB_SEND_094_SNVT_preset**

This function block sends the following LON output variable (nvo):

**SNVT Name**: SNVT_preset.

**SNVT number**: 094.

**Description**: Default (learn mode, value, time).

### VAR_INPUT

```
  wNVIndex : WORD;
  stValue : ST_LON_SNVT_preset;
  bStart : BOOL;
  bSendInit := bSendInitDefault;
  bAuto := bAutoDefault;
  tMinSendTime := tMinSendTimeDefault;
  tMaxSendTime := tMaxSendTimeDefault;
  bDisabled := FALSE;
```

- **wNVIndex**: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **stValue**: Structure of the data to be sent (see `ST_LON_SNVT_preset` [611]).
- **bStart**: A positive edge starts the send process (irrespective of `bAuto`).
- **bSendInit**: Automatically sends the value once when the PLC restarts.
- **bAuto**: Automatic sending [630] is selected when a value changes or the time `tMaxSendTime` has elapsed (polling).
tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>bBusy</th>
<th>BOOL;</th>
</tr>
</thead>
<tbody>
<tr>
<td>bError</td>
<td>BOOL;</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR;</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>DWORD;</td>
</tr>
</tbody>
</table>

**bBusy:** The *bBusy* output is TRUE as long as values are sent.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

**eError:** This output outputs an error code in the event of an error (see E_LON_ERROR [477]). *bError* goes TRUE at the same time.

**dwErrorKL:** Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable *eError* has the value eKL6401_Error. *bError* goes TRUE at the same time.

**VAR_IN_OUT**

| stLON_Com | ST_LON_Communication; |

**stLON_Com:** This structure is used to link FB_LON_KL6401 [27] with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.3.90 FB_SEND_095_SNVT_switch**

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_switch.

**SNVT number:** 095.

**Description:** Switch (value, status).
VAR_INPUT

wNVIndex  : WORD;
stValue   : ST_LON_SNVT_switch;
bStart    : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto     : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

stValue: Structure of the data to be sent (see ST_LON_SNVT_switch [618]).

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

VAR_OUTPUT

bBusy     : BOOL;
bError    : BOOL;
eError    : E_LON_ERROR;
dwErrorKL : DWORD;

eBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
This function block sends the following LON output variable (nvo):

**SNVT Name**: SNVT_trans_table.

**SNVT number**: 096.

**Description**: Conversion table (number of values, interpolation).

### VAR_INPUT

- **wNVIndex**: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **stValue**: Structure of the data to be sent (see `ST_LON_SNVT_trans_table`).
- **bStart**: A positive edge starts the send process (irrespective of `bAuto`).
- **bSendInit**: Automatically sends the value once when the PLC restarts.
- **bAuto**: Automatic sending is selected when a value changes or the time `tMaxSendTime` has elapsed (polling).
- **tMinSendTime**: Parameter for automatic sending. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.
- **tMaxSendTime**: Parameter for automatic sending. The value is sent after this time has elapsed at the latest. The value 0 disables this function.
- **bDisabled**: TRUE = deselection of the block.

### VAR_OUTPUT

- **bBusy**: The `bBusy` output is TRUE as long as values are sent.
- **bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.
- **eError**: This output outputs an error code in the event of an error (see `E_LON_ERROR`). `bError` goes TRUE at the same time.
**Programming**

**dwErrorKL:** Error ID of the function block FB_LON_KL6401(27) (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

**VAR_IN_OUT**

stLON_Com : ST_LON_Communication;

**stLON_Com:** This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.3.92 FB_SEND_097_SNVT_override**

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_override.

**SNVT number:** 097.

**Description:** Override mode (enumeration: 0= keep current value, 1= set specific value set, 2= set default value).

**VAR_INPUT**

<table>
<thead>
<tr>
<th>wNVIndex</th>
<th>eValue</th>
<th>bStart</th>
<th>bSendInit</th>
<th>bAuto</th>
<th>tMinSendTime</th>
<th>tMaxSendTime</th>
<th>bDisabled</th>
<th>stLON_Com</th>
</tr>
</thead>
<tbody>
<tr>
<td>WORD</td>
<td>E_LON_override_t</td>
<td>BOOL</td>
<td>BOOL := bSendInitDefault;</td>
<td>BOOL := bAutoDefault;</td>
<td>TIME := tMinSendTimeDefault;</td>
<td>TIME := tMaxSendTimeDefault;</td>
<td>BOOL := FALSE;</td>
<td></td>
</tr>
</tbody>
</table>

**wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**eValue:** Enum to be sent (see E_LON_override_t [532]).

**bStart:** A positive edge starts the send process (irrespective of bAuto).

**bSendInit:** Automatically sends the value once when the PLC restarts.

**bAuto:** Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

**tMinSendTime:** Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.
tMaxSendTime: Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

| bBusy      | BOOL |
| bError     | BOOL |
| eError     | E_LON_ERROR |
| dwErrorKL  | DWORD |

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

| stLON_Com | ST_LON_Communication |

stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.93 FB_SEND_098_SNVT_pwr_fact

This function block sends the following LON output variable (nvo):

SNVT Name: SNVT_pwr_fact.

SNVT number: 098.

Description: Power factor (multiplier).

VAR_INPUT

| wNVIndex      | WORD |
| rValue        | REAL |
| bStart        | BOOL |
| bSendInit     | BOOL := bSendInitDefault; |

| tMinSendTime  |
| tMaxSendTime  |
| bDisabled     |

| stLON_Com     |
wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

rValue: Min: -1 / Max: 1.

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

rValueLimit: Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401 [27] with the send/receive function (see ST LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
FB_SEND_099_SNVT_pwr_fact_f

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_pwr_fact_f.

**SNVT number:** 099.

**Description:** Power factor (multiplier).

### VAR_INPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>WORD</td>
<td>Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.</td>
</tr>
<tr>
<td>rValue</td>
<td>REAL</td>
<td>Min: -1 / Max: 1.</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL</td>
<td>A positive edge starts the send process (irrespective of bAuto).</td>
</tr>
<tr>
<td>bSendInit</td>
<td>BOOL := bSendInitDefault;</td>
<td>Automatically sends the value once when the PLC restarts.</td>
</tr>
<tr>
<td>bAuto</td>
<td>BOOL := bAutoDefault;</td>
<td>Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).</td>
</tr>
<tr>
<td>rValueLimit</td>
<td>REAL := 0.1;</td>
<td>Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>TIME := tMinSendTimeDefault;</td>
<td>Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.</td>
</tr>
<tr>
<td>tMaxSendTime</td>
<td>TIME := tMaxSendTimeDefault;</td>
<td>Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL := FALSE;</td>
<td>TRUE = deselection of the block.</td>
</tr>
</tbody>
</table>

### VAR_OUTPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>BOOL</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>DWORD</td>
</tr>
</tbody>
</table>
**bBusy**: The `bBusy` output is TRUE as long as values are sent.

**bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.

**eError**: This output outputs an error code in the event of an error (see `E_LON_ERROR` [477]). `bError` goes TRUE at the same time.

**dwErrorKL**: Error ID of the function block `FB_LON_KL6401` [27] (see `dwErrorKL` [631]). In this case the variable `eError` has the value `eKL6401_Error`. `bError` goes TRUE at the same time.

### VAR_IN_OUT

```plaintext
stLON_Com : ST_LON_Communication;
```

**stLON_Com**: This structure is used to link `FB_LON_KL6401` [27] with the send/receive function (see `ST_LON_Communication` [587]).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.3.95 FB_SEND_100_SNVT_density

This function block sends the following LON output variable (nvo):

**SNVT Name**: SNVT_density.

**SNVT number**: 100.

**Description**: Density (kg/m³).

### VAR_INPUT

```plaintext
wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex**: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**rValue**: Min: 0 / Max: 32767.5.

**bStart**: A positive edge starts the send process (irrespective of `bAuto`).
bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending \[630\] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

rValueLimit: Parameter for automatic sending \[630\]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

tMinSendTime: Parameter for automatic sending \[630\]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending \[630\]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR \[477\]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 \[27\](). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() \[27\] with the send/receive function (see ST_LON_Communication \[587\]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.96 FB_SEND_101_SNVT_density_f
This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_density_f.

**SNVT number:** 101.

**Description:** Density (kg/m³).

### VAR_INPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>WORD</td>
<td>Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.</td>
</tr>
<tr>
<td>rValue</td>
<td>REAL</td>
<td>Min: 0 / Max: 3.40E+38.</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL</td>
<td>A positive edge starts the send process (irrespective of bAuto).</td>
</tr>
<tr>
<td>bSendInit</td>
<td>BOOL</td>
<td>Automatically sends the value once when the PLC restarts.</td>
</tr>
<tr>
<td>bAuto</td>
<td>BOOL</td>
<td>Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).</td>
</tr>
<tr>
<td>rValueLimit</td>
<td>REAL</td>
<td>Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>TIME</td>
<td>Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.</td>
</tr>
<tr>
<td>tMaxSendTime</td>
<td>TIME</td>
<td>Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL</td>
<td>TRUE = deselection of the block.</td>
</tr>
</tbody>
</table>

### VAR_OUTPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>BOOL</td>
<td>The bBusy output is TRUE as long as values are sent.</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
<td>This output goes TRUE as soon as an error occurs. This error is described via the variable eError.</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
<td>This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>DWORD</td>
<td>Error ID of the function block FB_LON_KL6401 [271] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.</td>
</tr>
</tbody>
</table>

### VAR_IN_OUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stLON_Com</td>
<td>ST_LON_Communication</td>
<td>This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).</td>
</tr>
</tbody>
</table>
Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.97 **FB_SEND_102_SNVT_rpm**

```
FB_SEND_102_SNVT_rpm
  wNVIndex : WORD;
  uiValue : UINT;
  bStart : BOOL;
  bSendInit : BOOL := bSendInitDefault;
  bAuto : BOOL := bAutoDefault;
  uiValueLimit : UINT := 1;
  tMinSendTime : TIME := tMinSendTimeDefault;
  tMaxSendTime : TIME := tMaxSendTimeDefault;
  bDisabled : BOOL := FALSE;
```

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_rpm.

**SNVT number:** 102.

**Description:** Speed (revolutions/minute (RPM)).

**VAR_INPUT**

- **wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

- **uiValue:** Min: 0 / Max: 65535.

- **bStart:** A positive edge starts the send process (irrespective of bAuto).

- **bSendInit:** Automatically sends the value once when the PLC restarts.

- **bAuto:** Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

- **uiValueLimit:** Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

- **tMinSendTime:** Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

- **tMaxSendTime:** Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (uiValueLimit) was not reached (polling). The value 0 disables this function.

- **bDisabled:** TRUE = deselection of the block.
**VAR_OUTPUT**

- **bBusy**: The `bBusy` output is TRUE as long as values are sent.
- **bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.
- **eError**: This output outputs an error code in the event of an error (see `E_LON_ERROR`). `bError` goes TRUE at the same time.
- **dwErrorKL**: Error ID of the function block `FB_LON_KL6401` (see `dwErrorKL`). In this case the variable `eError` has the value `eKL6401_Error`. `bError` goes TRUE at the same time.

**VAR_IN_OUT**

- **stLON_Com**: This structure is used to link `FB_LON_KL6401` with the send/receive function (see `ST_LON_Communication`).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

#### 4.1.3.98  FB_SEND_103_SNVT_hvac_emerg

This function block sends the following LON output variable (nvo):

- **SNVT Name**: `SNVT_hvac_emerg`.
- **SNVT number**: 103.
- **Description**: HVAC emergency mode (operating modes).

**VAR_INPUT**

- **wNVIndex**: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
eValue: Enum to be sent (see E_LON_emerg_t [517]).

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.99 FB_SEND_104_SNVT_angle_deg

This function block sends the following LON output variable (nvo):
SNVT Name: SNVT_angle_deg.
SNVT number: 104.
Description: Angle specification in 1/50 degree steps.

VAR_INPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>WORD</td>
</tr>
<tr>
<td>rValue</td>
<td>REAL</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL</td>
</tr>
<tr>
<td>bSendInit</td>
<td>BOOL</td>
</tr>
<tr>
<td>bAuto</td>
<td>BOOL</td>
</tr>
<tr>
<td>rValueLimit</td>
<td>REAL</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>TIME</td>
</tr>
<tr>
<td>tMaxSendTime</td>
<td>TIME</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL</td>
</tr>
</tbody>
</table>

wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

rValue: Min: -359.98 / Max: 360.

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending is selected when a value changes or the time tMaxSendTime has elapsed (polling).

rValueLimit: Parameter for automatic sending. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

tMinSendTime: Parameter for automatic sending. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>BOOL</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>DWORDKL</td>
</tr>
</tbody>
</table>

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401() (see dwErrorKL). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>stLON_Com</td>
<td>ST_LON_Communication</td>
</tr>
</tbody>
</table>

stLON_Com: This structure is used to link FB_LON_KL6401() with the send/receive function (see ST_LON_Communication).
Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.3.100 FB_SEND_105_SNVT_temp_p

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_temp_p.

**SNVT number:** 105.

**Description:** Temperature (°C).

**VAR_INPUT**

```plaintext
wNVIndex : WORD;
rValue   : REAL;
bStart   : BOOL := bSendInitDefault;
bAuto    : BOOL := bAutoDefault;
```

- **wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **rValue:** Min: -273.17 / Max: 327.67.
- **bStart:** A positive edge starts the send process (irrespective of bAuto).
- **bSendInit:** Automatically sends the value once when the PLC restarts.
- **bAuto:** Automatic sending is selected when a value changes or the time tMaxSendTime has elapsed (polling).
- **rValueLimit:** Parameter for automatic sending. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.
- **tMinSendTime:** Parameter for automatic sending. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.
- **tMaxSendTime:** Parameter for automatic sending. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.
- **bDisabled:** TRUE = deselection of the block.
VAR_OUTPUT

**bBusy**: The *bBusy* output is TRUE as long as values are sent.

**bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

**eError**: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). *bError* goes TRUE at the same time.

**dwErrorKL**: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable *eError* has the value eKL6401_Error. *bError* goes TRUE at the same time.

VAR_IN_OUT

**stLON_Com**: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.101  **FB_SEND_106_SNVT_temp_setpt**

This function block sends the following LON output variable (nvo):

**SNVT Name**: SNVT_temp_setpt.

**SNVT number**: 106.

**Description**: Temperature (6 temperature values).

VAR_INPUT

**wNVIndex** : WORD;

**stValue** : ST_LON_SNVT_temp_setpt;

**bStart** : BOOL;

**bSendInit** : BOOL := bSendInitDefault;

**bAuto** : BOOL := bAutoDefault;

**tMinSendTime** : TIME := tMinSendTimeDefault;

**tMaxSendTime** : TIME := tMaxSendTimeDefault;

**bDisabled** : BOOL := FALSE;

**wNVIndex**: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
**stValue**: Structure of the data to be sent (see ST_LON_SNVT_temp_setpt [619]).

**bStart**: A positive edge starts the send process (irrespective of bAuto).

**bSendInit**: Automatically sends the value once when the PLC restarts.

**bAuto**: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

**tMinSendTime**: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

**tMaxSendTime**: Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

**bDisabled**: TRUE = deselection of the block.

**VAR_OUTPUT**

| bBusy   : BOOL;   |
| bError  : BOOL;   |
| eError  : E_LON_ERROR; |
| dwErrorKL : DWORD; |

**bBusy**: The bBusy output is TRUE as long as values are sent.

**bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

**eError**: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

**dwErrorKL**: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

**VAR_IN_OUT**

| stLON_Com  : ST_LON_Communication; |

**stLON_Com**: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.3.102 FB_SEND_107_SNVT_time_sec**

This function block sends the following LON output variable (nvo):
**SNVT Name:** SNVT_time_sec.

**SNVT number:** 107.

**Description:** Elapsed time (second).

### VAR_INPUT

| wNVIndex    | WORD;  
| rValue      | REAL;  
| bStart      | BOOL;  
| bSendInit   | BOOL := bSendInitDefault;  
| bAuto       | BOOL := bAutoDefault;  
| rValueLimit | REAL := 1;  
| tMinSendTime| TIME := tMinSendTimeDefault;  
| tMaxSendTime| TIME := tMaxSendTimeDefault;  
| bDisabled   | BOOL := FALSE;  

**wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**rValue:** Min: 0 / Max: 6553.5.

**bStart:** A positive edge starts the send process (irrespective of `bAuto`).

**bSendInit:** Automatically sends the value once when the PLC restarts.

**bAuto:** Automatic sending is selected when a value changes or the time `tMaxSendTime` has elapsed (polling).

**rValueLimit:** Parameter for automatic sending. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

**tMinSendTime:** Parameter for automatic sending. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

**tMaxSendTime:** Parameter for automatic sending. The value is sent when this time has elapsed at the latest, even if the minimum change in value (`rValueLimit`) was not reached (polling). The value 0 disables this function.

**bDisabled:** TRUE = deselection of the block.

### VAR_OUTPUT

| bBusy        | BOOL;  
| bError       | BOOL;  
| eError       | E_LON_ERROR;  
| dwErrorKL    | DWORD;  

**bBusy:** The `bBusy` output is TRUE as long as values are sent.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.

**eError:** This output outputs an error code in the event of an error (see `E_LON_ERROR`). `bError` goes TRUE at the same time.

**dwErrorKL:** Error ID of the function block `FB_LON_KL6401` (see `dwErrorKL`). In this case the variable `eError` has the value `eKL6401_Error`. `bError` goes TRUE at the same time.

### VAR_IN_OUT

| stLON_Com    | ST_LON_Communication;  

**stLON_Com:** This structure is used to link `FB_LON_KL6401` with the send/receive function (see `ST_LON_Communication`).
Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.103 FB_SEND_108_SNVT_hvac_mode

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_hvac_mode.

**SNVT number:** 108.

**Description:** HVAC operating mode (operating modes).

**VAR_INPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.</td>
</tr>
<tr>
<td>eValue</td>
<td>Enum to be sent (see E_LON_hvac_t[527]).</td>
</tr>
<tr>
<td>bStart</td>
<td>A positive edge starts the send process (irrespective of bAuto).</td>
</tr>
<tr>
<td>bSendInit</td>
<td>Automatically sends the value once when the PLC restarts.</td>
</tr>
<tr>
<td>bAuto</td>
<td>Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.</td>
</tr>
<tr>
<td>tMaxSendTime</td>
<td>Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.</td>
</tr>
<tr>
<td>bDisabled</td>
<td>TRUE = deselection of the block.</td>
</tr>
</tbody>
</table>

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>BOOL;</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL;</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR;</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>DWORD;</td>
</tr>
</tbody>
</table>
bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.104  FB_SEND_109_SNVT_occupancy

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_occupancy.

**SNVT number:** 109.

**Description:** Occupancy signal (states).

VAR_INPUT

wNVIndex : WORD;
eValue : E_LON_occup_t;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

eValue: Enum to be sent (see E_LON_occup_t [532]).

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.
bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.105 FB_SEND_110_SNVT_area

This function block sends the following LON output variable (nvo):

SNVT Name: SNVT_area.

SNVT number: 110.

Description: Area (square meter).
VAR_INPUT

wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

rValue: Min: 0 / Max: 13.107.

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

rValueLimit: Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_hvac_overid.

**SNVT number:** 111.

**Description:** HVAC override mode (mode, position/flow rate, min/max flow rate).

**VAR_INPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.</td>
</tr>
<tr>
<td>stValue</td>
<td>Structure of the data to be sent (see ST_LON_SNVT_hvac_overid[604]).</td>
</tr>
<tr>
<td>bStart</td>
<td>A positive edge starts the send process (irrespective of bAuto).</td>
</tr>
<tr>
<td>bSendInit</td>
<td>Automatically sends the value once when the PLC restarts.</td>
</tr>
<tr>
<td>bAuto</td>
<td>Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.</td>
</tr>
<tr>
<td>tMaxSendTime</td>
<td>Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.</td>
</tr>
<tr>
<td>bDisabled</td>
<td>TRUE = deselection of the block.</td>
</tr>
</tbody>
</table>

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>The bBusy output is TRUE as long as values are sent.</td>
</tr>
<tr>
<td>bError</td>
<td>This output goes TRUE as soon as an error occurs. This error is described via the variable eError.</td>
</tr>
<tr>
<td>eError</td>
<td>This output outputs an error code in the event of an error (see E_LON_ERROR[477]). bError goes TRUE at the same time.</td>
</tr>
</tbody>
</table>
**Programming**

dwErrorKL: Error ID of the function block FB_LON_KL6401() (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

**VAR_IN_OUT**

stLon_Com : ST_LON_Communication;

stLon_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.3.107  FB_SEND_112_SNVT_hvac_status**

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_hvac_status.

**SNVT number:** 112.

**Description:** HVAC status (mode, 5 power ratings, alarm flag).

**VAR_INPUT**

wNVIndex : WORD;
stValue : ST_LON_SNVT_hvac_status;
bStart : BOOL;
bSendInit := BOOL := bSendInitDefault;
bAuto := BOOL := bAutoDefault;
tMinSendTime := TIME := tMinSendTimeDefault;
tMaxSendTime := TIME := tMaxSendTimeDefault;
bDisabled := BOOL := FALSE;

wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

stValue: Structure of the data to be sent (see ST_LON_SNVT_hvac_status [605]).

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.
tMaxSendTime: Parameter for automatic sending. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 (27) (see dwErrorKL 631). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com: This structure is used to link FB_LON_KL6401() with the send/receive function (see ST_LON_Communication).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.108 FB_SEND_113_SNVT_press_p

This function block sends the following LON output variable (nvo):

SNVT Name: SNVT_press_p.

SNVT number: 113.

Description: Pressure (overpressure) (pascal).

VAR_INPUT

wNVIndex : WORD;
iValue : INT;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
iValueLimit : UINT := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

iValue: Min: -32768 / Max: 32767.

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

iValueLimit: Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

tMinSendTime: Parameter for automatic sending [630]. A new value is sent after this time has elapsed at the earliest. This prevents continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (iValueLimit) was not reached (polling). The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com: This structure is used to link FB_LON_KL6401 [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
This function block sends the following LON output variable (nvo):

**SNVT Name**: SNVT_address.

**SNVT number**: 114.

**Description**: Neuron address (16-bit address value).

**VAR_INPUT**

- **wNVIndex**: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **uiValue**: Min: 16384 / Max: 64767.
- **bStart**: A positive edge starts the send process (irrespective of bAuto).
- **bSendInit**: Automatically sends the value once when the PLC restarts.
- **bAuto**: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).
- **uiValueLimit**: Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.
- **tMinSendTime**: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.
- **tMaxSendTime**: Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (uiValueLimit) was not reached (polling). The value 0 disables this function.
- **bDisabled**: TRUE = deselection of the block.

**VAR_OUTPUT**

- **bBusy**: BOOL;
- **bError**: BOOL;
- **eError**: E_LON_ERROR;
- **dwErrorKL**: DWORD;
bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

VAR_INPUT

VAR_OUTPUT

Requirements

Development environment | required TC3 PLC library
TwinCAT from v3.1.4020.14 | Tc2_LON from 3.3.4.0

4.1.3.110 FB_SEND_115_SNVT_scene

This function block sends the following LON output variable (nvo):

SNVT Name: SNVT_scene.

SNVT number: 115.

Description: Scenes (function (retrieve/learn), scene number).

VAR_INPUT

wNVIndex : WORD;
stValue : ST_LON_SNVT_scene;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

stValue: Structure of the data to be sent (see ST_LON_SNVT_scene [617]).

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.
bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

bAuto: Automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>BOOL</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>DWORD</td>
</tr>
</tbody>
</table>

**bBusy**: The bBusy output is TRUE as long as values are sent.

**bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

**eError**: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

**dwErrorKL**: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

**VAR_IN_OUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>stLON_Com</td>
<td>ST_LON_Communication</td>
</tr>
</tbody>
</table>

**stLON_Com**: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.3.111  FB_SEND_116_SNVT_scene_cfg

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_scene_cfg.

**SNVT number:** 116.

**Description:** Scene setting (function, scene number, setting, transition number, transition time, delay time, priority).
**VAR_INPUT**

- **wNVIndex**: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **stValue**: Structure of the data to be sent (see `ST_LON_SNVT_scene_cfg`).
- **bStart**: A positive edge starts the send process (irrespective of `bAuto`).
- **bSendInit**: Automatically sends the value once when the PLC restarts.
- **bAuto**: Automatic sending is selected when a value changes or the time `tMaxSendTime` has elapsed (polling).
- **tMinSendTime**: Parameter for automatic sending. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.
- **tMaxSendTime**: Parameter for automatic sending. The value is sent after this time has elapsed at the latest. The value 0 disables this function.
- **bDisabled**: TRUE = deselection of the block.

**VAR_OUTPUT**

- **bBusy**: The `bBusy` output is TRUE as long as values are sent.
- **bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.
- **eError**: This output outputs an error code in the event of an error (see `E_LON_ERROR`). `bError` goes TRUE at the same time.
- **dwErrorKL**: Error ID of the function block `FB_LON_KL6401()`. In this case the variable `eError` has the value `eKL6401_Error`. `bError` goes TRUE at the same time.

**VAR_IN_OUT**

- **stLON_Com**: This structure is used to link `FB_LON_KL6401()` with the send/receive function (see `ST_LON_Communication`).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_setting.

**SNVT number:** 117.

**Description:** Settings (function, value, transition number).

### VAR_INPUT

- `wNVIndex` : WORD;
- `stValue` : ST_LON_SNVT_setting;
- `bStart` : BOOL;
- `bSendInit` : BOOL := bSendInitDefault;
- `bAuto` : BOOL := bAutoDefault;
- `tMinSendTime` : TIME := tMinSendTimeDefault;
- `tMaxSendTime` : TIME := tMaxSendTimeDefault;
- `bDisabled` : BOOL := FALSE;

`wNVIndex` : Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

`stValue` : Structure of the data to be sent (see `ST_LON_SNVT_setting`).

`bStart` : A positive edge starts the send process (irrespective of `bAuto`).

`bSendInit` : Automatically sends the value once when the PLC restarts.

`bAuto` : Automatic sending is selected when a value changes or the time `tMaxSendTime` has elapsed (polling).

`tMinSendTime` : Parameter for automatic sending. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

`tMaxSendTime` : Parameter for automatic sending. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

`bDisabled` : TRUE = deselection of the block.

### VAR_OUTPUT

- `bBusy` : BOOL;
- `bError` : BOOL;
- `eError` : E_LON_ERROR;
- `dwErrorKL` : DWORD;

`bBusy` : The `bBusy` output is TRUE as long as values are sent.

`bError` : This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.

`eError` : This output outputs an error code in the event of an error (see `E_LON_ERROR`). `bError` goes TRUE at the same time.
Programming

**dwErrorKL**: Error ID of the function block `FB_LON_KL6401()` (see `dwErrorKL` [631]). In this case the variable `eError` has the value `eKL6401_Error`. `bError` goes TRUE at the same time.

**VAR_IN_OUT**

```plaintext
stLON_Com : ST_LON_Communication;
```

- `stLON_Com`: This structure is used to link `FB_LON_KL6401()` with the send/receive function (see `ST_LON_Communication` [587]).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.3.113  FB_SEND_118_SNVT_evap_state**

```plaintext
VAR_INPUT
wNVIndex : WORD;
eValue : E_LON_evap_t;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

- `wNVIndex`: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTS are permitted per LON terminal. Values between 0 and 61 are possible.
- `eValue`: Enum to be sent (see `E_LON_evap_t` [519]).
- `bStart`: A positive edge starts the send process (irrespective of `bAuto`).
- `bSendInit`: Automatically sends the value once when the PLC restarts.
- `bAuto`: Automatic sending [630] is selected when a value changes or the time `tMaxSendTime` has elapsed (polling).
- `tMinSendTime`: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

This function block sends the following LON output variable (nvo):

<table>
<thead>
<tr>
<th><strong>SNVT Name</strong></th>
<th><strong>SNVT number</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>SNVT_evap_state</td>
<td>118.</td>
<td>Evaporator status (enumeration).</td>
</tr>
</tbody>
</table>

**4.1.3.113  FB_SEND_118_SNVT_evap_state**

```plaintext
VAR_INPUT
wNVIndex : WORD;
eValue : E_LON_evap_t;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

- `wNVIndex`: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTS are permitted per LON terminal. Values between 0 and 61 are possible.
- `eValue`: Enum to be sent (see `E_LON_evap_t` [519]).
- `bStart`: A positive edge starts the send process (irrespective of `bAuto`).
- `bSendInit`: Automatically sends the value once when the PLC restarts.
- `bAuto`: Automatic sending [630] is selected when a value changes or the time `tMaxSendTime` has elapsed (polling).
- `tMinSendTime`: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.
**tMaxSendTime**: Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

**bDisabled**: TRUE = deselection of the block.

### VAR_OUTPUT

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>BOOL</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>DWORD</td>
</tr>
</tbody>
</table>

**bBusy**: The bBusy output is TRUE as long as values are sent.

**bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

**eError**: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

**dwErrorKL**: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

### VAR_IN_OUT

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>stLON_Com</td>
<td>ST_LON_Communication</td>
</tr>
</tbody>
</table>

**stLON_Com**: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.3.114 FB_SEND_119_SNVT_therm_mode

This function block sends the following LON output variable (nvo):

**SNVT Name**: SNVT_therm_mode.

**SNVT number**: 119.

**Description**: Thermostat mode (enumeration (enable, modulation)).

### VAR_INPUT

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>WORD</td>
</tr>
<tr>
<td>eValue</td>
<td>E_LON_therm_mode_t</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL</td>
</tr>
<tr>
<td>bSendInit</td>
<td>BOOL := bSendInitDefault;</td>
</tr>
<tr>
<td>bAuto</td>
<td>BOOL := bAutoDefault;</td>
</tr>
</tbody>
</table>

This function block sends the following LON output variable (nvo):

**SNVT Name**: SNVT_therm_mode.

**SNVT number**: 119.

**Description**: Thermostat mode (enumeration (enable, modulation)).
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;

\textbf{wNVIndex:} Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

\textbf{eValue:} Enum to be sent (see \texttt{E\_LON\_therm\_mode\_t [549]}).

\textbf{bStart:} A positive edge starts the send process (irrespective of \texttt{bAuto}).

\textbf{bSendInit:} Automatically sends the value once when the PLC restarts.

\textbf{bAuto:} \texttt{Automatic sending [630]} is selected when a value changes or the time \texttt{tMaxSendTime} has elapsed (polling).

\textbf{tMinSendTime:} Parameter for \texttt{automatic sending [630]}. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

\textbf{tMaxSendTime:} Parameter for \texttt{automatic sending [630]}. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

\textbf{bDisabled:} TRUE = deselection of the block.

\textbf{VAR OUTPUT}

<table>
<thead>
<tr>
<th>bBusy</th>
<th>BOOL</th>
</tr>
</thead>
<tbody>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>DWORD</td>
</tr>
</tbody>
</table>

\texttt{bBusy: The bBusy output is TRUE as long as values are sent.}

\texttt{bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.}

\texttt{eError: This output outputs an error code in the event of an error (see \texttt{E\_LON\_ERROR [477]}). bError goes TRUE at the same time.}

\texttt{dwErrorKL: Error ID of the function block \texttt{FB\_LON\_KL6401 [27]}() (see dwErrorKL [631]). In this case the variable eError has the value eKL6401\_Error. bError goes TRUE at the same time.}

\textbf{VAR_IN_OUT}

<table>
<thead>
<tr>
<th>stLON_Com</th>
<th>ST_LON_Communication</th>
</tr>
</thead>
</table>

\texttt{stLON\_Com : This structure is used to link \texttt{FB\_LON\_KL6401 [27]} with the send/receive function (see \texttt{ST\_LON\_Communication [587]})}.

\textbf{Requirements}

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
This function block sends the following LON output variable (nvo):

SNVT Name: SNVT_defr_mode.
SNVT number: 120.
Description: Defrost mode (enumeration).

VAR_INPUT

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.</td>
</tr>
<tr>
<td>eValue</td>
<td>Enum to be sent (see E_LON_defrost_mode_t [513]).</td>
</tr>
<tr>
<td>bStart</td>
<td>A positive edge starts the send process (irrespective of bAuto).</td>
</tr>
<tr>
<td>bSendInit</td>
<td>Automatically sends the value once when the PLC restarts.</td>
</tr>
<tr>
<td>bAuto</td>
<td>Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.</td>
</tr>
<tr>
<td>tMaxSendTime</td>
<td>Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.</td>
</tr>
<tr>
<td>bDisabled</td>
<td>TRUE = deselection of the block.</td>
</tr>
</tbody>
</table>

VAR_OUTPUT

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>The bBusy output is TRUE as long as values are sent.</td>
</tr>
<tr>
<td>bError</td>
<td>This output goes TRUE as soon as an error occurs. This error is described via the variable eError.</td>
</tr>
<tr>
<td>eError</td>
<td>This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.</td>
</tr>
</tbody>
</table>
**Programming**

**VAR_IN_OUT**

```
stLON_Com : ST_LON_Communication;
```

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.3.116 FB_SEND_121_SNVT_defr_term

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_defr_term.

**SNVT number:** 121.

**Description:** Completion of the defrost cycle (enumeration).

**VAR_INPUT**

```
wNVIndex : WORD;
eValue : E_LON_defrost_term_t;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**eValue:** Enum to be sent (see `E_LON_defrost_term_t`).

**bStart:** A positive edge starts the send process (irrespective of `bAuto`).

**bSendInit:** Automatically sends the value once when the PLC restarts.

**bAuto:** Automatic sending (`630`) is selected when a value changes or the time `tMaxSendTime` has elapsed (polling).

**tMinSendTime:** Parameter for automatic sending (`630`). A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.
tMaxSendTime: Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>bBusy</th>
<th>BOOL</th>
</tr>
</thead>
<tbody>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>DWORD</td>
</tr>
</tbody>
</table>

**bBusy:** The *bBusy* output is TRUE as long as values are sent.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

**eError:** This output goes TRUE as soon as an error occurs (see E_LON_ERROR [477]). *bError* goes TRUE at the same time.

**dwErrorKL:** Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable *eError* has the value eKL6401_Error. *bError* goes TRUE at the same time.

**VAR_IN_OUT**

| stLON_Com | ST_LON_Communication; |

**stLON_Com:** This structure is used to link FB_LON_KL6401 [27] with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.3.117 FB_SEND_122_SNVT_defr_state**

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_defr_state.

**SNVT number:** 122.

**Description:** Defrost status (enumeration).

**VAR_INPUT**

<table>
<thead>
<tr>
<th>wNVIndex</th>
<th>WORD</th>
</tr>
</thead>
<tbody>
<tr>
<td>eValue</td>
<td>E_LON_defrost_state_t</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL</td>
</tr>
<tr>
<td>bSendInit</td>
<td>BOOL := bSendInitDefault;</td>
</tr>
<tr>
<td>bAuto</td>
<td>BOOL := bAutoDefault;</td>
</tr>
</tbody>
</table>
**wNVIndex**: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**eValue**: Enum to be sent (see `E_LON_defrost_state_t [514]`).

**bStart**: A positive edge starts the send process (irrespective of `bAuto`).

**bSendInit**: Automatically sends the value once when the PLC restarts.

**bAuto**: Automatic sending [630] is selected when a value changes or the time `tMaxSendTime` has elapsed (polling).

**tMinSendTime**: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

**tMaxSendTime**: Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

**bDisabled**: TRUE = deselection of the block.

### VAR_OUTPUT

**bBusy** : BOOL;

**bError** : BOOL;

**eError** : E_LON_ERROR;

**dwErrorKL** : DWORD;

**bBusy**: The `bBusy` output is TRUE as long as values are sent.

**bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.

**eError**: This output outputs an error code in the event of an error (see `E_LON_ERROR [477]`). `bError` goes TRUE at the same time.

**dwErrorKL**: Error ID of the function block `FB_LON_KL6401 [27]()` (see `dwErrorKL [631]`). In this case the variable `eError` has the value `eKL6401_Error`. `bError` goes TRUE at the same time.

### VAR_IN_OUT

**stLON_Com** : ST_LON_Communication;

**stLON_Com**: This structure is used to link `FB_LON_KL6401()` [27] with the send/receive function (see `ST_LON_Communication [587]`).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.1.3.118  FB_SEND_123_SNVT_time_min

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_time_min.

**SNVT number:** 123.

**Description:** Elapsed time (minutes).

**VAR_INPUT**

- wNVIndex : WORD;
- uiValue : UINT;
- bStart : BOOL;
- bSendInit : BOOL := bSendInitDefault;
- bAuto : BOOL := bAutoDefault;
- uiValueLimit : UINT := 1;
- tMinSendTime : TIME := tMinSendTimeDefault;
- tMaxSendTime : TIME := tMaxSendTimeDefault;
- bDisabled : BOOL := FALSE;
- stLON_Com

**wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**uiValue:** Min: 0 / Max: 65535.

**bStart:** A positive edge starts the send process (irrespective of bAuto).

**bSendInit:** Automatically sends the value once when the PLC restarts.

**bAuto:** Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

**uiValueLimit:** Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

**tMinSendTime:** Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

**tMaxSendTime:** Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (uiValueLimit) was not reached (polling). The value 0 disables this function.

**bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

- bBusy : BOOL;
- bError : BOOL;
- eError : E_LON_ERROR;
- dwErrorKL : DWORD;
bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.119 FB_SEND_124_SNVT_time_hour

This function block sends the following LON output variable (nvo):

SNVT Name: SNVT_time_hour.

SNVT number: 124.

Description: Elapsed time (hour).

VAR_INPUT

wNVIndex : WORD;
uiValue : UINT;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
uiValueLimit : UINT := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

uiValue: Min: 0 / Max: 65535.

bStart: A positive edge starts the send process (irrespective of bAuto).
**bSendInit:** Automatically sends the value once when the PLC restarts.

**bAuto:** Automatic sending [630] is selected when a value changes or the time $tMaxSendTime$ has elapsed (polling).

**uiValueLimit:** Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

**tMinSendTime:** Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

**tMaxSendTime:** Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value ($uiValueLimit$) was not reached (polling). The value 0 disables this function.

**bDisabled:** TRUE = deselection of the block.

### VAR_OUTPUT

- **bBusy** : BOOL;
- **bError** : BOOL;
- **eError** : E_LON_ERROR;
- **dwErrorKL** : DWORD;

**bBusy:** The $bBusy$ output is TRUE as long as values are sent.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable $eError$.

**eError:** This output outputs an error code in the event of an error (see E_LON_ERROR [477]). $bError$ goes TRUE at the same time.

**dwErrorKL:** Error ID of the function block FB_LON_KL6401 [27] with the send/receive function (see ST_LON_Communication [587]). In this case the variable $eError$ has the value eKL6401_Error. $bError$ goes TRUE at the same time.

### VAR_IN_OUT

- **stLON_Com** : ST_LON_Communication;

**stLON_Com:** This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.3.120 FB_SEND_125_SNVT_ph

```
 FB_SEND_125_SNVT_ph
 | wNVIndex | bBusy |
 | rValue   | bError |
 | bStart   | eError |
 | bSendInit| dwErrorKL |
 | bAuto    |
 | rValueLimit |
 | tMinSendTime |
 | tMaxSendTime |
 | bDisabled  |
 | stLON_Com  |
```
This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_ph.

**SNVT number:** 125.

**Description:** Acidity (pH). Ion concentration ratio.

### VAR_INPUT

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.</td>
</tr>
<tr>
<td>rValue</td>
<td>Min: -32.768 / Max: 32.767.</td>
</tr>
<tr>
<td>bStart</td>
<td>A positive edge starts the send process (irrespective of bAuto).</td>
</tr>
<tr>
<td>bSendInit</td>
<td>Automatically sends the value once when the PLC restarts.</td>
</tr>
<tr>
<td>bAuto</td>
<td>Automatic sending is selected when a value changes or the time tMaxSendTime has elapsed (polling).</td>
</tr>
<tr>
<td>rValueLimit</td>
<td>Parameter for automatic sending. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>Parameter for automatic sending. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.</td>
</tr>
<tr>
<td>tMaxSendTime</td>
<td>Parameter for automatic sending. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.</td>
</tr>
<tr>
<td>bDisabled</td>
<td>TRUE = deselection of the block.</td>
</tr>
</tbody>
</table>

### VAR_OUTPUT

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>The bBusy output is TRUE as long as values are sent.</td>
</tr>
<tr>
<td>bError</td>
<td>This output goes TRUE as soon as an error occurs. This error is described via the variable eError.</td>
</tr>
<tr>
<td>eError</td>
<td>This output outputs an error code in the event of an error (see E_LON_ERROR). bError goes TRUE at the same time.</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>Error ID of the function block FB_LON_KL6401() (see dwErrorKL) (see E_LON_ERROR). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.</td>
</tr>
</tbody>
</table>

### VAR_IN_OUT

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stLON_Com</td>
<td>This structure is used to link FB_LON_KL6401() with the send/receive function (see ST_LON_Communication).</td>
</tr>
</tbody>
</table>
Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.121  **FB_SEND_126_SNVT_ph_f**

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_ph_f.

**SNVT number:** 126.

**Description:** Acidity (pH). Ion concentration ratio.

### VAR_INPUT

| wNVIndex | : WORD; |
| rValue | : REAL; |
| bStart | : BOOL; |
| bSendInit | : BOOL := bSendInitDefault; |
| bAuto | : BOOL := bAutoDefault; |
| rValueLimit | : REAL := 1; |
| tMinSendTime | : TIME := tMinSendTimeDefault; |
| tMaxSendTime | : TIME := tMaxSendTimeDefault; |
| bDisabled | : BOOL := FALSE; |

**wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**rValue:** Min: -3.40E+38 / Max: 3.40E+38.

**bStart:** A positive edge starts the send process (irrespective of bAuto).

**bSendInit:** Automatically sends the value once when the PLC restarts.

**bAuto:** Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

**rValueLimit:** Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

**tMinSendTime:** Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

**tMaxSendTime:** Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

**bDisabled:** TRUE = deselection of the block.
VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.122  FB_SEND_127_SNVT_chlr_status

This function block sends the following LON output variable (nvo):

SNVT Name: SNVT_chlr_status.

SNVT number: 127.

Description: Refrigeration unit status (run mode, op mode, status bits).

VAR_INPUT

wNVIndex : WORD;
stValue : ST_LON_SNVT_chlr_status;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
**stValue:** Structure of the data to be sent (see `ST_LON_SNVT_chlr_status` [593]).

**bStart:** A positive edge starts the send process (irrespective of `bAuto`).

**bSendInit:** Automatically sends the value once when the PLC restarts.

**bAuto:** Automatic sending (630) is selected when a value changes or the time `tMaxSendTime` has elapsed (polling).

**tMinSendTime:** Parameter for automatic sending (630). A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

**tMaxSendTime:** Parameter for automatic sending (630). The value is sent after this time has elapsed at the latest. The value 0 disables this function.

**bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

```plaintext
bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;
```

**bBusy:** The `bBusy` output is TRUE as long as values are sent.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.

**eError:** This output outputs an error code in the event of an error (see `E_LON_ERROR` [477]). `bError` goes TRUE at the same time.

**dwErrorKL:** Error ID of the function block `FB_LON_KL6401` (27) (see `dwErrorKL` [631]). In this case the variable `eError` has the value `eKL6401_Error`. `bError` goes TRUE at the same time.

**VAR_IN_OUT**

```plaintext
stLON_Com : ST_LON_Communication;
```

**stLON_Com:** This structure is used to link `FB_LON_KL6401` (27) with the send/receive function (see `ST_LON_Communication` [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.3.123 FB_SEND_128_SNVT_tod_event**

This function block sends the following LON output variable (nvo):

**SNVT Name:** `SNVT_tod_event`. 
**SNVT number:** 128.

**Description:** Presence time (busy status, next busy time).

**VAR_INPUT**

<table>
<thead>
<tr>
<th>Declaration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex : WORD;</td>
<td>Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.</td>
</tr>
<tr>
<td>stValue : ST_LON_SNVT_tod_event;</td>
<td>Structure of the data to be sent (see ST_LON_SNVT_tod_event [620]).</td>
</tr>
<tr>
<td>bStart : BOOL;</td>
<td>A positive edge starts the send process (irrespective of bAuto).</td>
</tr>
<tr>
<td>bSendInit : BOOL := bSendInitDefault;</td>
<td>Automatically sends the value once when the PLC restarts.</td>
</tr>
<tr>
<td>bAuto : BOOL := bAutoDefault;</td>
<td>Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).</td>
</tr>
<tr>
<td>tMinSendTime : TIME := tMinSendTimeDefault;</td>
<td>Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.</td>
</tr>
<tr>
<td>tMaxSendTime : TIME := tMaxSendTimeDefault;</td>
<td>Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.</td>
</tr>
<tr>
<td>bDisabled : BOOL := FALSE;</td>
<td>TRUE = deselection of the block.</td>
</tr>
</tbody>
</table>

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>Declaration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy : BOOL;</td>
<td>The bBusy output is TRUE as long as values are sent.</td>
</tr>
<tr>
<td>bError : BOOL;</td>
<td>This output goes TRUE as soon as an error occurs. This error is described via the variable eError.</td>
</tr>
<tr>
<td>eError : E_LON_ERROR;</td>
<td>This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.</td>
</tr>
<tr>
<td>dwErrorKL : DWORD;</td>
<td>Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.</td>
</tr>
</tbody>
</table>

**VAR_IN_OUT**

<table>
<thead>
<tr>
<th>Declaration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stLON_Com : ST_LON_Communication;</td>
<td>This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).</td>
</tr>
</tbody>
</table>

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.1.3.124  **FB_SEND_129_SNVT_smo_obscur**

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_smo_obscur.

**SNVT number:** 129.

**Description:** Darkening due to smoke (darkening %).

### VAR_INPUT

- **wNVIndex** : WORD;
- **rValue** : REAL;
- **bStart** : BOOL;
- **bSendInit** : BOOL := bSendInitDefault;
- **bAuto** : BOOL := bAutoDefault;
- **rValueLimit** : REAL := 1;
- **tMinSendTime** : TIME := tMinSendTimeDefault;
- **tMaxSendTime** : TIME := tMaxSendTimeDefault;
- **bDisabled** : BOOL := FALSE;

**wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**rValue:** Min: 0 / Max: 5.

**bStart:** A positive edge starts the send process (irrespective of **bAuto**).

**bSendInit:** Automatically sends the value once when the PLC restarts.

**bAuto:** **Automatic sending** [630] is selected when a value changes or the time **tMaxSendTime** has elapsed (polling).

**rValueLimit:** Parameter for **automatic sending** [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

**tMinSendTime:** Parameter for **automatic sending** [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

**tMaxSendTime:** Parameter for **automatic sending** [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

**bDisabled:** TRUE = deselection of the block.

### VAR_OUTPUT

- **bBusy** : BOOL;
- **bError** : BOOL;
- **eError** : E_LON_ERROR;
- **dwErrorKL** : DWORD;
bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] () (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

VAR_INPUT

wNVIndex : WORD;
eValue : E_LON_fire_test_t;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;

This function block sends the following LON output variable (nvo):

SNVT Name: SNVT_fire_test.

SNVT number: 130.

Description: Fire alarm system test request (fire alarm test designations).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>TC2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.125  FB_SEND_130_SNVT_fire_test

This function block sends the following LON output variable (nvo):

SNVT Name: SNVT_fire_test.

SNVT number: 130.

Description: Fire alarm system test request (fire alarm test designations).

VAR_INPUT

wNVIndex : WORD;
eValue : E_LON_fire_test_t;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

eValue: Enum to be sent (see E_LON_fire_test_t [523]).

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.
**bAuto:** Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

**tMinSendTime:** Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

**tMaxSendTime:** Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

**bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

```plaintext
bBusy        : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
dwErrorKL    : DWORD;
```

**bBusy:** The bBusy output is TRUE as long as values are sent.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

**eError:** This output goes TRUE in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

**dwErrorKL:** Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

**VAR_IN_OUT**

```plaintext
stLON_Com    : ST_LON_Communication;
```

**stLON_Com:** This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.3.126 FB_SEND_131_SNVT_temp_ror

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_temp_ror.

**SNVT number:** 131.

**Description:** Value of the temperature change/increase (°C/minute).
VAR_INPUT

wNVIndex : WORD;
iVar : REAL;
bSt : BOOL;
bSE : BOOL := bSEDefault;
bSt : BOOL := bStDefault;
iVar : REAL := 1;
tMinSt : TIME := tMinStDefault;
tMaxSt : TIME := tMaxStDefault;
bD : BOOL := FALSE;

wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

iVar: Min: -16384 / Max: 16383.5.

bSt: A positive edge starts the send process (irrespective of bSt).

bSE: Automatically sends the value once when the PLC restarts.

bSt: Automatic sending [630] is selected when a value change or the time tMaxSt has elapsed (polling).

iVar: Parameter for automatic sending [630]. The value is only sent if the change of the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

tMinSt: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSt: Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (iVar) was not reached (polling). The value 0 disables this function.

bD: TRUE = deselection of the block.

VAR_OUTPUT

bSt : BOOL;
bE : BOOL;
eE : E_LON_ERROR;
dwKL : DWORD;

bSt: The bSt output is TRUE as long as values are sent.

bE: This output goes TRUE as soon as an error occurs. This error is described via the variable eE.

eE: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bE goes TRUE at the same time.

dwKL: Error ID of the function block FB_LON_KL6401 [27] (see dwKL [631]). In this case the variable eE has the value eKL6401_Error. bE goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com: This structure is used to link FB_LON_KL6401 [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT\_fire\_init.

**SNVT number:** 132.

**Description:** Fire detector type (fire detector type names).

### VAR\_INPUT

- `wNVIndex` : WORD;
- `eValue` : E\_LON\_fire\_initiator\_t;
- `bStart` : BOOL;
- `bSendInit` : BOOL := bSendInitDefault;
- `bAuto` : BOOL := bAutoDefault;
- `tMinSendTime` : TIME := tMinSendTimeDefault;
- `tMaxSendTime` : TIME := tMaxSendTimeDefault;
- `bDisabled` : BOOL := FALSE;

**wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**eValue:** Enum to be sent (see [E\_LON\_fire\_initiator\_t](#) [522]).

**bStart:** A positive edge starts the send process (irrespective of `bAuto`).

**bSendInit:** Automatically sends the value once when the PLC restarts.

**bAuto:** Automatic sending ([630]) is selected when a value changes or the time `tMaxSendTime` has elapsed (polling).

**tMinSendTime:** Parameter for automatic sending ([630]). A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

**tMaxSendTime:** Parameter for automatic sending ([630]). The value is sent after this time has elapsed at the latest. The value 0 disables this function.

**bDisabled:** TRUE = deselection of the block.

### VAR\_OUTPUT

- `bBusy` : BOOL;
- `bError` : BOOL;
- `eError` : E\_LON\_ERROR;
- `dwErrorKL` : DWORD;

**bBusy:** The `bBusy` output is TRUE as long as values are sent.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.

**eError:** This output outputs an error code in the event of an error (see [E\_LON\_ERROR](#) [477]). `bError` goes TRUE at the same time.
Programming

**VAR_IN_OUT**

```plaintext
stLON_Com : ST_LON_Communication;
```

```
VAR_INPUT
wNVIndex : WORD;
eValue : E_LON_fire_indicator_t;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.3.128 FB_SEND_133_SNVT_fire_indcte

This function block sends the following LON output variable (nvo):

- **SNVT Name:** SNVT_fire_indcte.
- **SNVT number:** 133.
- **Description:** Fire alarm display (fire alarm display names).

**VAR_INPUT**

- `wNVIndex`: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- `eValue`: Enum to be sent (see `E_LON_fire_indicator_t` [621]).
- `bStart`: A positive edge starts the send process (irrespective of `bAuto`).
- `bSendInit`: Automatically sends the value once when the PLC restarts.
- `bAuto`: Automatic sending [630] is selected when a value changes or the time `tMaxSendTime` has elapsed (polling).
- `tMinSendTime`: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.
tMaxSendTime: Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

| bBusy    | BOOL;               |
| bError   | BOOL;               |
| eError   | E_LON_ERROR;       |
| dwErrorKL| DWORD;              |

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

| stLON_Com | ST_LON_Communication; |

stLON_Com: This structure is used to link FB_LON_KL6401 [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.129 FB_SEND_134_SNVT_time_zone

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_time_zone.

**SNVT number:** 134.

**Description:** Time zone description (offset, type, summer time start, summer time end).

VAR_INPUT

| wNVIndex    | WORD;               |
| stValue     | ST_LON_SNVT_time_zone; |
| bStart      | BOOL;               |
| bSendInit   | BOOL := bSendInitDefault; |
| bAuto       | BOOL := bAutoDefault; |
wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

stValue: Structure of the data to be sent (see ST_LON_SNVT_time_zone [620]).

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27]() (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401[27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_earth_pos.

**SNVT number:** 135.

**Description:** Position on Earth (width & length orientation, latitude & min, longitude & min, height).

### VAR_INPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>WORD</td>
<td>Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.</td>
</tr>
<tr>
<td>stValue</td>
<td>ST_LON_SNVT_earth_pos</td>
<td>Structure of the data to be sent (see ST_LON_SNVT_earth_pos [599]).</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL</td>
<td>A positive edge starts the send process (irrespective of bAuto).</td>
</tr>
<tr>
<td>bSendInit</td>
<td>BOOL := bSendInitDefault;</td>
<td>Automatically sends the value once when the PLC restarts.</td>
</tr>
<tr>
<td>bAuto</td>
<td>BOOL := bAutoDefault;</td>
<td>Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>TIME := tMinSendTimeDefault;</td>
<td>Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.</td>
</tr>
<tr>
<td>tMaxSendTime</td>
<td>TIME := tMaxSendTimeDefault;</td>
<td>Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL := FALSE;</td>
<td>TRUE = deselection of the block.</td>
</tr>
</tbody>
</table>

### VAR_OUTPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>BOOL</td>
<td>The bBusy output is TRUE as long as values are sent.</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
<td>This output goes TRUE as soon as an error occurs. This error is described via the variable eError.</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR;</td>
<td>This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>DWORD</td>
<td></td>
</tr>
</tbody>
</table>
Programming

**dwErrorKL:** Error ID of the function block FB_LON_KL6401() (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

**VAR_IN_OUT**

```plaintext
stLON_Com : ST_LON_Communication;
```

**stLON_Com:** This structure is used to link FB_LON_KL6401() with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.3.131 FB_SEND_136_SNVT_reg_val**

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_reg_val.

**SNVT number:** 136.

**Description:** Register value (raw value, bit code, number of decimal places).

**VAR_INPUT**

```plaintext
wNVIndex : WORD;
stValue : ST_LON_SNVT_reg_val;
bStart : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**stValue:** Structure of the data to be sent (see ST_LON_SNVT_reg_val [615]).

**bStart:** A positive edge starts the send process (irrespective of bAuto).

**bSendInit:** Automatically sends the value once when the PLC restarts.

**bAuto:** Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

**tMinSendTime:** Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.
tMaxSendTime: Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

| bBusy   | BOOL; |
| bError  | BOOL; |
| eError  | E_LON_ERROR; |
| dwErrorKL | DWORD; |

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output puts an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

| stLON_Com | ST_LON_Communication; |

stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.132 FB_SEND_137_SNVT_reg_val_ts

This function block sends the following LON output variable (nvo):

SNVT Name: SNVT_reg_val_ts.

SNVT number: 137.

Description: Register value (raw value, bit code, number of decimal places, status, state, timestamp).

VAR_INPUT

| wNVIndex    | WORD; |
| stValue     | ST_LON_SNVT_reg_val_ts; |
| bStart      | BOOL; |
| bSendInit   | BOOL := bSendInitDefault; |
| bAuto       | BOOL := bAutoDefault; |
**Programming**

```plaintext
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled    : BOOL := FALSE;
```

**wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**stValue:** Structure of the data to be sent (see [ST_LON_SNVT_reg_val_ts](#) [616]).

**bStart:** A positive edge starts the send process (irrespective of bAuto).

**bSendInit:** Automatically sends the value once when the PLC restarts.

**bAuto:** Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

**tMinSendTime:** Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

**tMaxSendTime:** Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

**bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

```plaintext
bBusy     : BOOL;
bError    : BOOL;
eError    : E_LON_ERROR;
dwErrorKL : DWORD;
```

**bBusy:** The bBusy output is TRUE as long as values are sent.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

**eError:** This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

**dwErrorKL:** Error ID of the function block FB_LON_KL6401 [27](). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

**VAR_IN_OUT**

```plaintext
stLON_Com : ST_LON_Communication;
```

**stLON_Com:** This structure is used to link FB_LON_KL6401 [27] with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_volt_ac.

**SNVT number:** 138.

**Description:** Alternating voltage (Volt AC).

### VAR_INPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>WORD</td>
<td>Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.</td>
</tr>
<tr>
<td>uiValue</td>
<td>UINT</td>
<td>Min: 0 / Max: 65535.</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL</td>
<td>A positive edge starts the send process (irrespective of bAuto).</td>
</tr>
<tr>
<td>bSendInit</td>
<td>BOOL</td>
<td>Automatically sends the value once when the PLC restarts.</td>
</tr>
<tr>
<td>bAuto</td>
<td>BOOL</td>
<td>Automatic sending is selected when a value changes or the time tMaxSendTime has elapsed (polling).</td>
</tr>
<tr>
<td>uiValueLimit</td>
<td>UINT</td>
<td>Parameter for automatic sending. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>TIME</td>
<td>Parameter for automatic sending. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.</td>
</tr>
<tr>
<td>tMaxSendTime</td>
<td>TIME</td>
<td>Parameter for automatic sending. The value is sent when this time has elapsed at the latest, even if the minimum change in value (uiValueLimit) was not reached (polling). The value 0 disables this function.</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL</td>
<td>TRUE = deselection of the block.</td>
</tr>
</tbody>
</table>

### VAR_OUTPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>BOOL</td>
<td></td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
<td></td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
<td></td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>DWORD</td>
<td></td>
</tr>
</tbody>
</table>
bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.134  FB_SEND_139_SNVT_amp_ac

This function block sends the following LON output variable (nvo):

SNVT Name: SNVT_amp_ac.

SNVT number: 139.

Description: Alternating current (ampere AC).

VAR_INPUT

wNVIndex : WORD;
uiValue : UINT;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
uiValueLimit : UINT := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

uiValue: Min: 0 / Max: 65535.

bStart: A positive edge starts the send process (irrespective of bAuto).
bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

uiValueLimit: Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (uiValueLimit) was not reached (polling). The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.135 FB_SEND_143_SNVT_turbidity

```
<table>
<thead>
<tr>
<th>bSendInit</th>
<th>uiValueLimit</th>
<th>tMinSendTime</th>
<th>tMaxSendTime</th>
<th>bDisabled</th>
<th>bAuto</th>
<th>rValueLimit</th>
</tr>
</thead>
<tbody>
<tr>
<td>dwErrorKL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

stLON_Com
This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_turbidity.

**SNVT number:** 143.

**Description:** Turbidity (turbidity unit).

### VAR_INPUT

```plaintext
wNVIndex    : WORD;
rValue      : REAL;
bStart      : BOOL;
bSendInit   : BOOL := bSendInitDefault;
bAuto       : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime: TIME := tMinSendTimeDefault;
tMaxSendTime: TIME := tMaxSendTimeDefault;
bDisabled   : BOOL := FALSE;
```

- **wNVIndex**: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

- **rValue**: Min: 0 / Max: 65.535.

- **bStart**: A positive edge starts the send process (irrespective of bAuto).

- **bSendInit**: Automatically sends the value once when the PLC restarts.

- **bAuto**: Automatic sending is selected when a value changes or the time tMaxSendTime has elapsed (polling).

- **rValueLimit**: Parameter for automatic sending. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

- **tMinSendTime**: Parameter for automatic sending. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

- **tMaxSendTime**: Parameter for automatic sending. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

- **bDisabled**: TRUE = deselection of the block.

### VAR_OUTPUT

```plaintext
bBusy       : BOOL;
bError      : BOOL;
eError      : E_LON_ERROR;
dwErrorKL   : DWORD;
```

- **bBusy**: The bBusy output is TRUE as long as values are sent.

- **bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

- **eError**: This output outputs an error code in the event of an error (see E_LON_ERROR). bError goes TRUE at the same time.

- **dwErrorKL**: Error ID of the function block FB_LON_KL6401 (see dwErrorKL). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

### VAR_IN_OUT

```plaintext
stLON_Com   : ST_LON_Communication;
```

- **stLON_Com**: This structure is used to link FB_LON_KL6401 with the send/receive function (see ST_LON_Communication).
Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.136  **FB_SEND_144_SNVT_turbidity_f**

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_turbidity_f.

**SNVT number:** 144.

**Description:** Turbidity (turbidity unit).

**VAR_INPUT**

<table>
<thead>
<tr>
<th>wNVIndex</th>
<th>bEndy</th>
</tr>
</thead>
<tbody>
<tr>
<td>rValue</td>
<td>bError</td>
</tr>
<tr>
<td>bStart</td>
<td>eError</td>
</tr>
<tr>
<td>bSendInit</td>
<td>dwErrorKL</td>
</tr>
<tr>
<td>bAuto</td>
<td></td>
</tr>
<tr>
<td>rValueLimit</td>
<td></td>
</tr>
<tr>
<td>tMinSendTime</td>
<td></td>
</tr>
<tr>
<td>tMaxSendTime</td>
<td></td>
</tr>
<tr>
<td>bDisabled</td>
<td></td>
</tr>
</tbody>
</table>

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_turbidity_f.

**SNVT number:** 144.

**Description:** Turbidity (turbidity unit).

**VAR_INPUT**

<table>
<thead>
<tr>
<th>wNVIndex</th>
<th>WORD;</th>
</tr>
</thead>
<tbody>
<tr>
<td>rValue</td>
<td>REAL;</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL;</td>
</tr>
<tr>
<td>bSendInit</td>
<td>BOOL := bSendInitDefault;</td>
</tr>
<tr>
<td>bAuto</td>
<td>BOOL := bAutoDefault;</td>
</tr>
<tr>
<td>rValueLimit</td>
<td>REAL := 1;</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>TIME := tMinSendTimeDefault;</td>
</tr>
<tr>
<td>tMaxSendTime</td>
<td>TIME := tMaxSendTimeDefault;</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL := FALSE;</td>
</tr>
</tbody>
</table>

**wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**rValue:** Min: 0 / Max: 3.40E+38.

**bStart:** A positive edge starts the send process (irrespective of bAuto).

**bSendInit:** Automatically sends the value once when the PLC restarts.

**bAuto:** Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

**rValueLimit:** Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

**tMinSendTime:** Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

**tMaxSendTime:** Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

**bDisabled:** TRUE = deselection of the block.
VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com: This structure is used to link FB_LON_KL6401 [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.137 FB_SEND_145_SNVT_hvac_type

This function block sends the following LON output variable (nvo):

SNVT Name: SNVT_hvac_type.

SNVT number: 145.

Description: HVAC plant type (HVAC plant type description).

VAR_INPUT

wNVIndex : WORD;
eValue : E_LON_hvac_hvt_t;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
eValue: Enum to be sent (see E_LON_hvac_hvt_t [524]).

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

<table>
<thead>
<tr>
<th>bBusy</th>
<th>BOOL</th>
</tr>
</thead>
<tbody>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>DWORD</td>
</tr>
</tbody>
</table>

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

| stLON_Com | ST_LON_Communication |

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.138 FB_SEND_146_SNVT_elec_kwh_l

This function block sends the following LON output variable (nvo):
SNVT Name: SNVT_elec_kwh_l.
SNVT number: 146.
Description: Electric energy (kW / hour).

VAR_INPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>WORD</td>
</tr>
<tr>
<td>lrValue</td>
<td>LREAL</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL</td>
</tr>
<tr>
<td>bSendInit</td>
<td>BOOL</td>
</tr>
<tr>
<td>bAuto</td>
<td>BOOL</td>
</tr>
<tr>
<td>lrValueLimit</td>
<td>LREAL</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>TIME</td>
</tr>
<tr>
<td>tMaxSendTime</td>
<td>TIME</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL</td>
</tr>
</tbody>
</table>

wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

lrValue: Min: -214748364.8 / Max: 214748364.7.

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending is selected when a value changes or the time tMaxSendTime has elapsed (polling).

lrValueLimit: Parameter for automatic sending. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

tMinSendTime: Parameter for automatic sending. A new value is sent after this time has elapsed at the earliest. This prevents continuous sending.

tMaxSendTime: Parameter for automatic sending. The value is sent when this time has elapsed at the latest, even if the minimum change in value (lrValueLimit) was not reached (polling). The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>BOOL</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>DWORD</td>
</tr>
</tbody>
</table>

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 (see dwErrorKL). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stLON_Com</td>
<td>This structure is used to link FB_LON_KL6401 with the send/receive function (see ST_LON_Communication).</td>
</tr>
</tbody>
</table>
4.1.3.139 **FB_SEND_147_SNVT_temp_diff_p**

This function block sends the following LON output variable (nvo):

**SNVT Name**: SNVT_temp_diff_p.

**SNVT number**: 147.

**Description**: Temperature difference (°C).

**VAR_INPUT**

```
  varIndex : WORD;
  rValue : REAL;
  bStart : BOOL;
  bSendInit := bSendInitDefault;
  bAuto := bAutoDefault;
  rValueLimit := REAL := 1;
  tMinSendTime := TIME := tMinSendTimeDefault;
  tMaxSendTime := TIME := tMaxSendTimeDefault;
  bDisabled := BOOL := FALSE;
```

- **wNVIndex**: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **rValue**: Min: -327.68 / Max: 327.67.
- **bStart**: A positive edge starts the send process (irrespective of bAuto).
- **bSendInit**: Automatically sends the value once when the PLC restarts.
- **bAuto**: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).
- **rValueLimit**: Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.
- **tMinSendTime**: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.
- **tMaxSendTime**: Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.
- **bDisabled**: TRUE = deselection of the block.
VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.140 FB_SEND_148_SNVT_ctrl_req

This function block sends the following LON output variable (nvo):

SNVT Name: SNVT_ctrl_req.

SNVT number: 148.

Description: Control request (receiver ID, sender ID, sender priority). Request for control permission for another controller.

VAR_INPUT

wNVIndex : WORD;
stValue : ST_LON_SNVT_ctrl_req;
bStart : BOOL;
bSendInit := bSendInitDefault;
bAuto := bAutoDefault;
tMinSendTime := tMinSendTimeDefault;
tMaxSendTime := tMaxSendTimeDefault;
bDisabled := FALSE;

wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
stValue: Structure of the data to be sent (see ST_LON_SNVT_ctrl_req [596]).

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

**bBusy**: The bBusy output is TRUE as long as values are sent.

**bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

**eError**: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

**dwErrorKL**: Error ID of the function block FB_LON_KL6401 [27]() (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

**stLON_Com**: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.141 FB_SEND_149_SNVT_ctrl_resp

This function block sends the following LON output variable (nvo):

**SNVT Name**: SNVT_ctrl_resp.
**SNVT number:** 149.

**Description:** Control response (status, transmitter, controller ID). Response to a control request.

**VAR_INPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>WORD</td>
<td>Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.</td>
</tr>
<tr>
<td>stValue</td>
<td>ST_LON_SNVT_ctrl_resp</td>
<td>Structure of the data to be sent (see ST_LON_SNVT_ctrl_resp).</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL</td>
<td>A positive edge starts the send process (irrespective of bAuto).</td>
</tr>
<tr>
<td>bSendInit</td>
<td>BOOL</td>
<td>Automatically sends the value once when the PLC restarts.</td>
</tr>
<tr>
<td>bAuto</td>
<td>BOOL</td>
<td>Automatic sending is selected when a value changes or the time tMaxSendTime has elapsed (polling).</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>TIME</td>
<td>Parameter for automatic sending. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.</td>
</tr>
<tr>
<td>tMaxSendTime</td>
<td>TIME</td>
<td>Parameter for automatic sending. The value is sent after this time has elapsed at the latest. The value 0 disables this function.</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL</td>
<td>TRUE = deselection of the block.</td>
</tr>
</tbody>
</table>

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>BOOL</td>
<td>The bBusy output is TRUE as long as values are sent.</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
<td>This output goes TRUE as soon as an error occurs. This error is described via the variable eError.</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
<td>This output outputs an error code in the event of an error (see E_LON_ERROR). bError goes TRUE at the same time.</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>DWORD</td>
<td>Error ID of the function block FB_LON_KL6401() (see dwErrorKL). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.</td>
</tr>
</tbody>
</table>

**VAR_IN_OUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stLON_Com</td>
<td>ST_LON_Communication</td>
<td>This structure is used to link FB_LON_KL6401() with the send/receive function (see ST_LON_Communication).</td>
</tr>
</tbody>
</table>

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
This function block sends the following LON output variable (nvo):

**SNVT Name**: SNVT_ptz.

**SNVT number**: 150.

**Description**: Camera PTZ (SNZ) (swivel, swivel speed, tilt, tilt speed, zoom, zoom speed).

### VAR_INPUT

- `wNVIndex` : `WORD`;
- `stValue` : `ST_LON_SNVT_ptz`;
- `bStart` : `BOOL`;
- `bSendInit` : `BOOL` := `bSendInitDefault`;
- `bAuto` : `BOOL` := `bAutoDefault`;
- `tMinSendTime` : `TIME` := `tMinSendTimeDefault`;
- `tMaxSendTime` : `TIME` := `tMaxSendTimeDefault`;
- `bDisabled` : `BOOL` := `FALSE`;

- `stLON_Com`

**wNVIndex**: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**stValue**: Structure of the data to be sent (see `ST_LON_SNVT_ptz`).

**bStart**: A positive edge starts the send process (irrespective of `bAuto`).

**bSendInit**: Automatically sends the value once when the PLC restarts.

**bAuto**: Automatic sending is selected when a value changes or the time `tMaxSendTime` has elapsed (polling).

**tMinSendTime**: Parameter for automatic sending. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

**tMaxSendTime**: Parameter for automatic sending. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

**bDisabled**: TRUE = deselection of the block.

### VAR_OUTPUT

- `bBusy` : `BOOL`;
- `bError` : `BOOL`;
- `eError` : `E_LON_ERROR`;
- `dwErrorKL` : `DWORD`;

- `stLON_Com`

**bBusy**: The `bBusy` output is TRUE as long as values are sent.

**bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.

**eError**: This output outputs an error code in the event of an error (see `E_LON_ERROR`). `bError` goes TRUE at the same time.
**Programming**

**VAR_IN_OUT**

stLON_Com : ST_LON_Communication;

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.3.143 FB_SEND_151_SNVT_privacyzone

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_privacyzone

**SNVT number:** 151

**Description:** Private zone (action area, zone number, camera ID). Certain areas can be excluded from the camera.

**VAR_INPUT**

- wNVIndex : WORD;
- stValue : ST_LON_SNVT_privacyzone;
- bStart : BOOL;
- bSendInit : BOOL := bSendInitDefault;
- bAuto : BOOL := bAutoDefault;
- tMinSendTime : TIME := tMinSendTimeDefault;
- tMaxSendTime : TIME := tMaxSendTimeDefault;
- bDisabled : BOOL := FALSE;

**wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**stValue:** Structure of the data to be sent (see ST_LON_SNVT_privacyzone [p. 611]).

**bStart:** A positive edge starts the send process (irrespective of bAuto).

**bSendInit:** Automatically sends the value once when the PLC restarts.

**bAuto:** Automatic sending [p. 630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

**tMinSendTime:** Parameter for automatic sending [p. 630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.
**tMaxSendTime**: Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

**bDisabled**: TRUE = deselection of the block.

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>BOOL</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>DWORD</td>
</tr>
</tbody>
</table>

**bBusy**: The bBusy output is TRUE as long as values are sent.

**bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

**eError**: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

**dwErrorKL**: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

**VAR_IN_OUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>stLON_Com</td>
<td>ST_LON_Communication</td>
</tr>
</tbody>
</table>

**stLON_Com**: This structure is used to link FB_LON_KL6401 [27] with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.3.144 FB_SEND_152_SNVT_pos_ctrl

This function block sends the following LON output variable (nvo):

**SNVT Name**: SNVT_pos_ctrl.

**SNVT number**: 152.

**Description**: Position setting for cameras in networks (receiver, controller ID, controller priority, function, action, value).

**VAR_INPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>WORD</td>
</tr>
<tr>
<td>stValue</td>
<td>ST_LON_SNVT_pos_ctrl</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL</td>
</tr>
<tr>
<td>bSendInit</td>
<td>BOOL := bSendInitDefault;</td>
</tr>
<tr>
<td>bAuto</td>
<td>BOOL := bAutoDefault;</td>
</tr>
</tbody>
</table>

This function block sends the following LON output variable (nvo):

**SNVT Name**: SNVT_pos_ctrl.

**SNVT number**: 152.

**Description**: Position setting for cameras in networks (receiver, controller ID, controller priority, function, action, value).
WNIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

stValue: Structure of the data to be sent (see ST_LON_SNVT_pos_ctrl [610]).

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disabling this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27]() (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Comm : ST_LON_Communication;

stLON_Comm : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_enthalpy.

**SNVT number:** 153.

**Description:** Enthalpy (kilojoules/kg).

### VAR_INPUT

- `wNVIndex` : WORD;
- `rValue` : REAL;
- `bStart` : BOOL;
- `bSendInit` : BOOL := bSendInitDefault;
- `bAuto` : BOOL := bAutoDefault;
- `rValueLimit` : REAL := 1;
- `tMinSendTime` : TIME := tMinSendTimeDefault;
- `tMaxSendTime` : TIME := tMaxSendTimeDefault;
- `bDisabled` : BOOL := FALSE;

- `wNVIndex`: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- `rValue`: Min: -327.68 / Max: 327.67.
- `bStart`: A positive edge starts the send process (irrespective of `bAuto`).
- `bSendInit`: Automatically sends the value once when the PLC restarts.
- `bAuto`: **Automatic sending** [630] is selected when a value changes or the time `tMaxSendTime` has elapsed (polling).
- `rValueLimit`: Parameter for **automatic sending** [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.
- `tMinSendTime`: Parameter for **automatic sending** [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.
- `tMaxSendTime`: Parameter for **automatic sending** [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (`rValueLimit`) was not reached (polling). The value 0 disables this function.
- `bDisabled`: TRUE = deselection of the block.

### VAR_OUTPUT

- `bBusy` : BOOL;
- `bError` : BOOL;
- `eError` : E_LON_ERROR;
- `dwErrorKL` : DWORD;
bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

VAR_INPUT

wNVIndex : WORD;
eValue : E_LON_gfci_status_t;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;

This function block sends the following LON output variable (nvo):

SNVT Name: SNVT_gfci_status.

SNVT number: 154.

Description: Residual current circuit breaker status (residual current circuit breaker status text).

VAR_INPUT

wNVIndex : WORD;
eValue : E_LON_gfci_status_t;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

eValue: Enum to be sent (see E_LON_gfci_status_t [524]).

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.
bAuto: Automatic sending [630] is selected when a value changes or the time \( t_{MaxSendTime} \) has elapsed (polling).

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

**VAR_OUTPUT**

- bBusy : BOOL;
- bError : BOOL;
- eError : E_LON_ERROR;
- dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

**VAR_IN_OUT**

- stLON_Com : ST_LON_Communication;

stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.147 **FB_SEND_155_SNVT_motor_state**

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_motor_state.

**SNVT number:** 155.

**Description:** Motor status (motor status designation).
VAR_INPUT

wNVIndex : WORD;
eValue : E_LON_motor_state_t;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

eValue: Enum to be sent (see E_LON_motor_state_t [529]).

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 () (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com: This structure is used to link FB_LON_KL6401 () with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_pumpset_mn.

**SNVT number:** 156.

**Description:** Pump group (main pump, auxiliary pump, priority, ready, emergency off, main pump activated, auxiliary pump activated, maintenance request). Synchronized vacuum pumps.

### VAR_INPUT

- **wNVIndex**: WORD;
- **stValue**: ST_LON_SNVT_pumpset_mn;
- **bStart**: BOOL;
- **bSendInit**: BOOL := bSendInitDefault;
- **bAuto**: BOOL := bAutoDefault;
- **tMinSendTime**: TIME := tMinSendTimeDefault;
- **tMaxSendTime**: TIME := tMaxSendTimeDefault;
- **bDisabled**: BOOL := FALSE;

**wNVIndex**: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**stValue**: Structure of the data to be sent (see [ST_LON_SNVT_pumpset_mn](#613)).

**bStart**: A positive edge starts the send process (irrespective of **bAuto**).

**bSendInit**: Automatically sends the value once when the PLC restarts.

**bAuto**: Automatic sending ([630](#630)) is selected when a value changes or the time **tMaxSendTime** has elapsed (polling).

**tMinSendTime**: Parameter for automatic sending ([630](#630)). A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

**tMaxSendTime**: Parameter for automatic sending ([630](#630)). The value is sent after this time has elapsed at the latest. The value 0 disables this function.

**bDisabled**: TRUE = deselection of the block.

### VAR_OUTPUT

- **bBusy**: BOOL;
- **bError**: BOOL;
- **eError**: E_LON_ERROR;
- **dwErrorKL**: DWORD;

**bBusy**: The **bBusy** output is TRUE as long as values are sent.

**bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable **eError**.

**eError**: This output outputs an error code in the event of an error (see [E_LON_ERROR](#477)). **bError** goes TRUE at the same time.
Programming

**dwErrorKL**: Error ID of the function block **FB_LON_KL6401** (see **dwErrorKL**). In this case the variable **eError** has the value **eKL6401_Error**. **bError** goes TRUE at the same time.

**VAR_IN_OUT**

```plaintext
stLON_Com : ST_LON_Communication;
```

**stLON_Com**: This structure is used to link **FB_LON_KL6401()** with the send/receive function (see **ST_LON_Communication**).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.3.149** **FB_SEND_157_SNVT_ex_control**

This function block sends the following LON output variable (nvo):

**SNVT Name**: SNVT_ex_control.

**SNVT number**: 157.

**Description**: Exclusive control (status, address). A device has exclusive control over another device.

**VAR_INPUT**

```plaintext
wNVIndex : WORD;
stValue : ST_LON_SNVT_ex_control;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex**: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**stValue**: Structure of the data to be sent (see **ST_LON_SNVT_ex_control**).

**bStart**: A positive edge starts the send process (irrespective of **bAuto**).

**bSendInit**: Automatically sends the value once when the PLC restarts.

**bAuto**: **Automatic sending** is selected when a value changes or the time **tMaxSendTime** has elapsed (polling).

**tMinSendTime**: Parameter for **automatic sending**. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.
tMaxSendTime: Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

| bBusy      : BOOL; |
| bError     : BOOL; |
| eError     : E_LON_ERROR; |
| dwErrorKL  : DWORD; |

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output puts an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.150   FB_SEND_158_SNVT_pumpset_sn

This function block sends the following LON output variable (nvo):

SNVT Name: SNVT_pumpset_sn.

SNVT number: 158.

Description: Pump group sensor (thinner, output, pressure, vacuum, ...).

VAR_INPUT

| wNVIndex : WORD; |
| stValue  : ST_LON_SNVT_pumpset_sn; |
| bStart   : BOOL; |
| bSendInit : BOOL := bSendInitDefault; |
| bAuto    : BOOL := bAutoDefault; |
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

stValue: Structure of the data to be sent (see ST_LON_SNVT_pumpset_sn [614]).

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27](). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.1.3.151   FB_SEND_159_SNVT_pump_sensor

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_pump_sensor.

**SNVT number:** 159.

**Description:** Pump sensor (speed, temperature, status).

**VAR_INPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>WORD</td>
</tr>
<tr>
<td>stValue</td>
<td>ST_LON_SNVT_pump_sensor</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL</td>
</tr>
<tr>
<td>bSendInit</td>
<td>BOOL := bSendInitDefault;</td>
</tr>
<tr>
<td>bAuto</td>
<td>BOOL := bAutoDefault;</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>TIME := tMinSendTimeDefault;</td>
</tr>
<tr>
<td>tMaxSendTime</td>
<td>TIME := tMaxSendTimeDefault;</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL := FALSE;</td>
</tr>
</tbody>
</table>

**wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**stValue:** Structure of the data to be sent (see `ST_LON_SNVT_pump_sensor`).

**bStart:** A positive edge starts the send process (irrespective of `bAuto`).

**bSendInit:** Automatically sends the value once when the PLC restarts.

**bAuto:** Automatic sending is selected when a value changes or the time `tMaxSendTime` has elapsed (polling).

**tMinSendTime:** Parameter for automatic sending. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

**tMaxSendTime:** Parameter for automatic sending. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

**bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>BOOL</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>DWORD</td>
</tr>
</tbody>
</table>

**bBusy:** The `bBusy` output is TRUE as long as values are sent.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.

**eError:** This output outputs an error code in the event of an error (see `E_LON_ERROR`). `bError` goes TRUE at the same time.
**VAR_IN_OUT**

```languagemarkup
stLON_Com : ST_LON_Communication;
```

**stLON_Com**: This structure is used to link FB_LON_KL6401() with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.3.152 FB_SEND_160_SNVT_abs_humid

This function block sends the following LON output variable (nvo):

**SNVT Name**: SNVT_abs_humid.

**SNVT number**: 160.

**Description**: Absolute humidity (gram/kg).

**VAR_INPUT**

```languagemarkup
wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex**: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**rValue**: Min: 0 / Max: 655.35.

**bStart**: A positive edge starts the send process (irrespective of bAuto).

**bSendInit**: Automatically sends the value once when the PLC restarts.

**bAuto**: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).
**tMinSendTime**: Parameter for automatic sending [p 630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

**tMaxSendTime**: Parameter for automatic sending [p 630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

**bDisabled**: TRUE = deselection of the block.

### VAR_OUTPUT

- **bBusy**: The *bBusy* output is TRUE as long as values are sent.
- **bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.
- **eError**: This output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.
- **dwErrorKL**: Error ID of the function block FB_LON_KL6401 [p 271] (see **dwErrorKL** [p 631]). In this case the variable *eError* has the value eKL6401_Error. *bError* goes TRUE at the same time.

### VAR_IN_OUT

- **stLON_Com**: This structure is used to link FB_LON_KL6401() [p 271] with the send/receive function (see **stLON_Com** [p 587]).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.3.153 FB_SEND_161_SNVT_flow_p

This function block sends the following LON output variable (nvo):

**SNVT Name**: SNVT_flow_p.

**SNVT number**: 161.

**Description**: Volume flow (cubic meters / hour).
VAR_INPUT

wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

rValue: Min: 0 / Max: 655.35.

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

rValueLimit: Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27]() (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_dev_c_mode.

**SNVT number:** 162.

**Description:** Device operating mode (device operating mode states).

### VAR_INPUT

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>WORD</td>
</tr>
<tr>
<td>eValue</td>
<td>E_LON_device_c_mode_t</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL</td>
</tr>
<tr>
<td>bSendInit</td>
<td>BOOL := bSendInitDefault;</td>
</tr>
<tr>
<td>bAuto</td>
<td>BOOL := bAutoDefault;</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>TIME := tMinSendTimeDefault;</td>
</tr>
<tr>
<td>tMaxSendTime</td>
<td>TIME := tMaxSendTimeDefault;</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL := FALSE;</td>
</tr>
</tbody>
</table>

- **wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **eValue:** Enum to be sent (see E_LON_device_c_mode_t [515]).
- **bStart:** A positive edge starts the send process (irrespective of bAuto).
- **bSendInit:** Automatically sends the value once when the PLC restarts.
- **bAuto:** Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).
- **tMinSendTime:** Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.
- **tMaxSendTime:** Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.
- **bDisabled:** TRUE = deselection of the block.

### VAR_OUTPUT

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>BOOL</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR;</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>DWORD</td>
</tr>
</tbody>
</table>

- **bBusy:** The bBusy output is TRUE as long as values are sent.
- **bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable eError.
- **eError:** This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.
**Programming**

**T1000**

**Version: 1.5**

**T1000**

**ewErrorKL**: Error ID of the function block FB_LON_KL6401() (see dwErrorKL [631]). In this case the variable `eError` has the value `eKL6401_Error`. `bError` goes TRUE at the same time.

**VAR_IN_OUT**

```plaintext
stLON_Com : ST_LON_Communication;
```

**stLON_Com**: This structure is used to link FB_LON_KL6401() with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.3.155 FB_SEND_163_SNVT_valve_mode**

```plaintext
VAR_INPUT
wNVIndex : WORD;
eValue : E_LON_valve_mode_t;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**VAR_IN_OUT**

```plaintext
VAR_IN_OUT
wNVIndex : WORD;
eValue : E_LON_valve_mode_t;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

`wNVIndex`: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**SNVT Name**: SNVT_valve_mode.

**SNVT number**: 163.

**Description**: Valve state.

**VAR_INPUT**

- **wNVIndex**: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **eValue**: Enum to be sent (see E_LON_valve_mode_t [550]).
- **bStart**: A positive edge starts the send process (irrespective of `bAuto`).
- **bSendInit**: Automatically sends the value once when the PLC restarts.
- **bAuto**: Automatic sending [630] is selected when a value changes or the time `tMaxSendTime` has elapsed (polling).
- **tMinSendTime**: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

This function block sends the following LON output variable (nvo):

**SNVT Name**: SNVT_valve_mode.

**SNVT number**: 163.

**Description**: Valve state.
**tMaxSendTime:** Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

**bDisabled:** TRUE = deselection of the block.

### VAR_OUTPUT

<table>
<thead>
<tr>
<th>bBusy</th>
<th>bool</th>
</tr>
</thead>
<tbody>
<tr>
<td>bError</td>
<td>bool</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>DWORD</td>
</tr>
</tbody>
</table>

**bBusy:** The **bBusy** output is TRUE as long as values are sent.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable **eError**.

**eError:** This output outputs an error code in the event of an error (see **E_LON_ERROR** [477]). **bError** goes TRUE at the same time.

**dwErrorKL:** Error ID of the function block **FB_LON_KL6401** [27] (see **dwErrorKL** [631]). In this case the variable **eError** has the value eKL6401_Error. **bError** goes TRUE at the same time.

### VAR_IN_OUT

<table>
<thead>
<tr>
<th>stLON_Com</th>
<th>ST_LON_Communication</th>
</tr>
</thead>
</table>

**stLON_Com:** This structure is used to link **FB_LON_KL6401** [27] with the send/receive function (see **ST_LON_Communication** [587]).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.3.156 **FB_SEND_164_SNVT_alarm_2**

```
FB_SEND_164_SNVT_alarm_2

wNVIndex bBusy
stValue bError
bStart eError
bSendInit dwErrorKL
bAuto

wMINSendTime
wMaxSendTime
bDisabled

stLON_Com
```

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_alarm_2.

**SNVT number:** 164.

**Description:** Alarm status 2. Signals the alarm status of a function block or device. Replaces SNVT_alarm.

### VAR_INPUT

<table>
<thead>
<tr>
<th>wNVIndex</th>
<th>WORD</th>
</tr>
</thead>
<tbody>
<tr>
<td>stValue</td>
<td>ST_LON_SNVT_alarm_2</td>
</tr>
<tr>
<td>bStart</td>
<td>bool</td>
</tr>
<tr>
<td>bSendInit</td>
<td>bool</td>
</tr>
<tr>
<td>bAuto</td>
<td>bool</td>
</tr>
</tbody>
</table>

**wNVIndex:**

**stValue:**

**bStart:**

**bSendInit:**

**bAuto:**
### tMinSendTime
TIME := tMinSendTimeDefault;

### tMaxSendTime
TIME := tMaxSendTimeDefault;

### bDisabled
BOOL := FALSE;

### wNVIndex
Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

### stValue
Structure of the data to be sent (see `ST_LON_SNVT_alarm_2 [592]`).

### bStart
A positive edge starts the send process (irrespective of `bAuto`).

### bSendInit
Automatically sends the value once when the PLC restarts.

### bAuto
Automatic sending [630] is selected when a value changes or the time `tMaxSendTime` has elapsed (polling).

### tMinSendTime
Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

### tMaxSendTime
Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

### bDisabled
TRUE = deselection of the block.

### VAR_OUTPUT
- **bBusy**
  The `bBusy` output is TRUE as long as values are sent.

- **bError**
  This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.

- **eError**
  This output outputs an error code in the event of an error (see `E_LON_ERROR [477]`). `bError` goes TRUE at the same time.

- **dwErrorKL**
  Error ID of the function block `FB_LON_KL6401 [27]()` (see `dwErrorKL [631]`). In this case the variable `eError` has the value `eKL6401_Error`. `bError` goes TRUE at the same time.

### VAR_IN_OUT
- **stLON_Com**
  This structure is used to link `FB_LON_KL6401 [27]` with the send/receive function (see `ST_LON_Communication [587]`).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.1.3.157 FB_SEND_165_SNVT_state_64

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_state_64.

**SNVT number:** 165.

**Description:** Status information (64 individual bit values). Each status is a Boolean value.

**VAR_INPUT**

- **wNVIndex** : WORD;
- **arrValue** : ARRAY [0..63] OF BOOL;
- **bStart** : BOOL;
- **bSendInit** : BOOL := bSendInitDefault;
- **bAuto** : BOOL := bAutoDefault;
- **tMinSendTime** : TIME := tMinSendTimeDefault;
- **tMaxSendTime** : TIME := tMaxSendTimeDefault;
- **bDisabled** := FALSE;

  - **wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
  - **arrValue:** 0-63 bit.
  - **bStart:** A positive edge starts the send process (irrespective of **bAuto**).
  - **bSendInit:** Automatically sends the value once when the PLC restarts.
  - **bAuto:** **Automatic sending [630]** is selected when a value changes or the time **tMaxSendTime** has elapsed (polling).
  - **tMinSendTime:** Parameter for **automatic sending [630]**. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.
  - **tMaxSendTime:** Parameter for **automatic sending [630]**. The value is sent after this time has elapsed at the latest. The value 0 disables this function.
  - **bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

- **bBusy** : BOOL;
- **bError** : BOOL;
- **eError** : E_LON_ERROR;
- **dwErrorKL** : DWORD;

  - **bBusy:** The **bBusy** output is TRUE as long as values are sent.
  - **bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable **eError**.
  - **eError:** This output outputs an error code in the event of an error (see **E_LON_ERROR [477]**). **bError** goes TRUE at the same time.
**Programming**

**dwErrorKL**: Error ID of the function block **FB_LON_KL6401 [27]** (see **dwErrorKL [631]**). In this case the variable **eError** has the value eKL6401_Error. **bError** goes TRUE at the same time.

**VAR_IN_OUT**

```plaintext
stLON_Com : ST_LON_Communication;
```

**stLON_Com**: This structure is used to link **FB_LON_KL6401() [27]** with the send/receive function (see **ST_LON_Communication [587]**).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.3.158  **FB_SEND_166_SNVT_nv_type**

This function block sends the following LON output variable (nvo):

**SNVT Name**: SNVT_nv_type.

**SNVT number**: 166.

**Description**: Network variable type. Type description for network variables.

**VAR_INPUT**

```plaintext
wNVIndex : WORD;
stValue : ST_LON_SNVT_nv_type;
bStart : BOOL := bStartDefault;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex**: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**stValue**: Structure of the data to be sent (see **ST_LON_SNVT_nv_type [608]**).

**bStart**: A positive edge starts the send process (irrespective of **bAuto**).

**bSendInit**: Automatically sends the value once when the PLC restarts.

**bAuto**: Automatic sending [630] is selected when a value changes or the time **tMaxSendTime** has elapsed (polling).
**rValueLimit:** Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

**tMinSendTime:** Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

**tMaxSendTime:** Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

**bDisabled:** TRUE = deselection of the block.

### VAR_OUTPUT

- **bBusy** : BOOL;
- **bError** : BOOL;
- **eError** : E_LON_ERROR;
- **dwErrorKL** : DWORD;

**bBusy:** The bBusy output is TRUE as long as values are sent.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

**eError:** This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

**dwErrorKL:** Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

### VAR_IN_OUT

- **stLON_Com** : ST_LON_Communication;

**stLON_Com:** This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.3.159 FB_SEND_168_SNVT_ent_opmode

```c
FD_SEND_168_SNVT_ent_opmode
 wNVindex bBusy
 eValue bError
 bStart eError
 bSendInit dwErrorKL
 bAuto tMinSendTime
 tMaxSendTime
 bDisabled stLON_Com
```

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_ent_opmode.

**SNVT number:** 168.
**Description:** Operating mode of access objects (doors, locks or objects permitting or prohibiting access).

**VAR_INPUT**

- **wNVIndex**: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

- **eValue**: Enum to be sent (see `E_LON_ent_opmode_cmd_t`).

- **bStart**: A positive edge starts the send process (irrespective of `bAuto`).

- **bSendInit**: Automatically sends the value once when the PLC restarts.

- **bAuto**: Automatic sending is selected when a value changes or the time `tMaxSendTime` has elapsed (polling).

- **tMinSendTime**: Parameter for automatic sending. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

- **tMaxSendTime**: Parameter for automatic sending. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

- **bDisabled**: TRUE = deselection of the block.

**VAR_OUTPUT**

- **bBusy**: The `bBusy` output is TRUE as long as values are sent.

- **bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.

- **eError**: This output outputs an error code in the event of an error (see `E_LON_ERROR`). `bError` goes TRUE at the same time.

- **dwErrorKL**: Error ID of the function block `FB_LON_KL6401` (see `dwErrorKL`). In this case the variable `eError` has the value eKL6401_Error. `bError` goes TRUE at the same time.

**VAR_IN_OUT**

- **stLON_Com**: This structure is used to link `FB_LON_KL6401` with the send/receive function (see `ST_LON_Communication`).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.1.3.160  FB_SEND_169_SNVT_ent_state

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_ent_state.

**SNVT number:** 169.

**Description:** State of access objects (doors, locks or objects permitting or prohibiting access).

### VAR_INPUT

- **wNVIndex**: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **eValue**: Enum to be sent (see `E_LON_ent_cmd_t` [517]).
- **bStart**: A positive edge starts the send process (irrespective of `bAuto`).
- **bSendInit**: Automatically sends the value once when the PLC restarts.
- **bAuto**: Automatic sending [630] is selected when a value changes or the time `tMaxSendTime` has elapsed (polling).
- **tMinSendTime**: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.
- **tMaxSendTime**: Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.
- **bDisabled**: TRUE = deselection of the block.

### VAR_OUTPUT

- **bBusy**: The `bBusy` output is TRUE as long as values are sent.
- **bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.
- **eError**: This output outputs an error code in the event of an error (see `E_LON_ERROR` [477]). `bError` goes TRUE at the same time.
**Programming**

**VAR_IN_OUT**

`stLON_Com : ST_LON_Communication;`

`stLON_Com`: This structure is used to link `FB_LON_KL6401()` with the send/receive function (see `ST_LON_Communication` [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.3.161 FB_SEND_170_SNVT_ent_status**

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_ent_status.

**SNVT number:** 170.

**Description:** Status of access objects (doors, locks or objects permitting or prohibiting access).

**VAR_INPUT**

```plaintext
wNVIndex : WORD;
stValue : ST_LON_SNVT_ent_status;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**stValue:** Structure of the data to be sent (see `ST_LON_SNVT_ent_status` [600]).

**bStart:** A positive edge starts the send process (irrespective of `bAuto`).

**bSendInit:** Automatically sends the value once when the PLC restarts.

**bAuto:** Automatic sending [630] is selected when a value changes or the time `tMaxSendTime` has elapsed (polling).

**tMinSendTime:** Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.
tMaxSendTime: Parameter for automatic sending \[630\]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>bBusy</th>
<th>BOOL;</th>
</tr>
</thead>
<tbody>
<tr>
<td>bError</td>
<td>BOOL;</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR;</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>DWORD;</td>
</tr>
</tbody>
</table>

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR \[477\]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 \[27\] (see dwErrorKL \[631\]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

**VAR_IN_OUT**

| stLON_Com | ST_LON_Communication; |

stLON_Com: This structure is used to link FB_LON_KL6401 \[27\] with the send/receive function (see ST_LON_Communication \[587\]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.3.162 FB_SEND_171_SNVT_flow_dir**

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_flow_dir.

**SNVT number:** 171.

**Description:** Flow direction. Direction of the flow to be permitted or direction of the current flow.

**VAR_INPUT**

<table>
<thead>
<tr>
<th>wNVIndex</th>
<th>WORD;</th>
</tr>
</thead>
<tbody>
<tr>
<td>eValue</td>
<td>E_LON_flow_direction_t;</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL;</td>
</tr>
<tr>
<td>bSendInit</td>
<td>BOOL := bSendInitDefault;</td>
</tr>
<tr>
<td>bAuto</td>
<td>BOOL := bAutoDefault;</td>
</tr>
</tbody>
</table>
wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

eValue: Enum to be sent (see E_LON_flow_direction_t [523]).

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27]() (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.1.3.163  FB_SEND_172_SNVT_hvac_satsts

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_hvac_satsts.

**SNVT number:** 172.

**Description:** HVAC saturation status.

0 in a field means that plant (device) linked to the field is not saturated or does not reach the limit stop before the required set value is reached.

1 in a field means that plant (device) linked to the field is saturated or reaches the limit stop before the required set value is not reached.

**VAR_INPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>WORD</td>
<td>Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.</td>
</tr>
<tr>
<td>stValue</td>
<td>ST_LON_SNVT_hvac_satsts</td>
<td>Structure of the data to be sent (see ST_LON_SNVT_hvac_satsts [604]).</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL</td>
<td>A positive edge starts the send process (irrespective of bAuto).</td>
</tr>
<tr>
<td>bSendInit</td>
<td>BOOL := bSendInitDefault</td>
<td>Automatically sends the value once when the PLC restarts.</td>
</tr>
<tr>
<td>bAuto</td>
<td>BOOL := bAutoDefault</td>
<td>Automatic sending is selected when a value changes or the time tMaxSendTime has elapsed (polling).</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>TIME := tMinSendTimeDefault</td>
<td>Parameter for automatic sending. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.</td>
</tr>
<tr>
<td>tMaxSendTime</td>
<td>TIME := tMaxSendTimeDefault</td>
<td>Parameter for automatic sending. The value is sent after this time has elapsed at the latest. The value 0 disables this function.</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL := FALSE</td>
<td>TRUE = deselection of the block.</td>
</tr>
</tbody>
</table>

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>BOOL</td>
<td>The bBusy output is TRUE as long as values are sent.</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
<td></td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
<td></td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>DWORD</td>
<td></td>
</tr>
</tbody>
</table>

VAR_INPUT

wNVIndex : WORD;
stValue : ST_LON_SNVT_hvac_satsts;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;

VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;
**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable **eError.**

**eError:** This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

**dwErrorKL:** Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable **eError** has the value eKL6401_Error. bError goes TRUE at the same time.

### VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

**stLON_Com:** This structure is used to link FB_LON_KL6401 [27] with the send/receive function (see ST_LON_Communication [587]).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.3.164 FB_SEND_173_SNVT_dev_status

```plaintext
FB_SEND_173_SNVT_dev_status
- wNVIndex
- stValue
- bStart
- bSendInit
- bAuto
- tMinSendTime
- tMaxSendTime
- bDisabled
- stLON_Com
```

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_dev_status.

**SNVT number:** 173.

**Description:** Device status.

### VAR_INPUT

```plaintext
wNVIndex : WORD;
stValue : ST_LON_SNVT_dev_status;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**stValue:** Structure of the data to be sent (see ST_LON_SNVT_dev_status [599]).

**bStart:** A positive edge starts the send process (irrespective of **bAuto**).

**bSendInit:** Automatically sends the value once when the PLC restarts.
bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

<table>
<thead>
<tr>
<th>bBusy</th>
<th>BOOL</th>
</tr>
</thead>
<tbody>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>DWORD</td>
</tr>
</tbody>
</table>

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

<table>
<thead>
<tr>
<th>stLON_Com</th>
<th>ST_LON_Communication</th>
</tr>
</thead>
</table>

stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.165  FB_SEND_174_SNVT_dev_fault

This function block sends the following LON output variable (nvo):

SNVT Name: SNVT_dev_fault.

SNVT number: 174.

Description: Error state. Error information for a device.
VAR_INPUT

wNVIndex : WORD;
stValue : ST_LON_SNVT_dev_fault;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

stValue: Structure of the data to be sent (see ST_LON_SNVT_dev_fault [598]).

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27](). (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
FB_SEND_175_SNVT_dev_maint

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_dev_maint.

**SNVT number:** 175.

**Description:** Device maintenance station.

**VAR_INPUT**

- `wNVIndex`: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

- `stValue`: Structure of the data to be sent (see `ST_LON_SNVT_dev_maint`).

- `bStart`: A positive edge starts the send process (irrespective of `bAuto`).

- `bSendInit`: Automatically sends the value once when the PLC restarts.

- `bAuto`: Automatic sending is selected when a value changes or the time `tMaxSendTime` has elapsed (polling).

- `tMinSendTime`: Parameter for automatic sending. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

- `tMaxSendTime`: Parameter for automatic sending. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

- `bDisabled`: TRUE = deselection of the block.

**VAR_OUTPUT**

- `bBusy`: The `bBusy` output is TRUE as long as values are sent.

- `bError`: This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.

- `eError`: This output outputs an error code in the event of an error (see `E_LON_ERROR`). `bError` goes TRUE at the same time.
**dwErrorKL**: Error ID of the function block `FB_LON_KL6401()` (see `dwErrorKL`). In this case the variable `eError` has the value `eKL6401_Error`. `bError` goes TRUE at the same time.

**VAR_IN_OUT**

```plaintext
stLON_Com : ST_LON_Communication;
```

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.3.167 FB_SEND_176_SNVT_date_event

This function block sends the following LON output variable (nvo):

**SNVT Name**: SNVT\_date\_event.

**SNVT number**: 176.

**Description**: Event status.

**VAR_INPUT**

```plaintext
wNVIndex : WORD;
stValue : ST_LON_SNVT_date_event;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex**: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**stValue**: Structure of the data to be sent (see `ST_LON_SNVT_date_event`).

**bStart**: A positive edge starts the send process (irrespective of `bAuto`).

**bSendInit**: Automatically sends the value once when the PLC restarts.

**bAuto**: Automatic sending is selected when a value changes or the time `tMaxSendTime` has elapsed (polling).

**tMinSendTime**: Parameter for automatic sending. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.
**tMaxSendTime**: Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

**bDisabled**: TRUE = deselection of the block.

**VAR_OUTPUT**

```plaintext
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy**: The bBusy output is TRUE as long as values are sent.

**bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

**eError**: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

**dwErrorKL**: Error ID of the function block FB_LON_KL6401 [27]) (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

**VAR_IN_OUT**

```plaintext
stLON_Com : ST_LON_Communication;
```

**stLON_Com**: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.3.168 FB_SEND_177_SNVT_sched_val**

This function block sends the following LON output variable (nvo):

**SNVT Name**: SNVT_sched_val.

**SNVT number**: 177.

**Description**: Calendar value. Index of a calendar value that selects an entry in an SCPT value definition field array or is a direct value output.

**VAR_INPUT**

```plaintext
wNVIndex    : WORD;
byValue     : BYTE;
bStart       : BOOL;
```

This function block sends the following LON output variable (nvo):

**SNVT Name**: SNVT_sched_val.

**SNVT number**: 177.

**Description**: Calendar value. Index of a calendar value that selects an entry in an SCPT value definition field array or is a direct value output.
wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

byValue: Min: 0 / Max: 255.

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

byValueLimit: Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

tMinSendTime: Parameter for automatic sending [630]. A new value is sent after this time has elapsed at the earliest. This prevents continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (byValueLimit) was not reached (polling). The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401 [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_sblnd_state.

**SNVT number:** 180.

**Description:** Blind status.

### VAR_INPUT

- `wNVIndex`: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- `stValue`: Structure of the data to be sent (see `ST_LON_SNVT_sblnd_state`).
- `bStart`: A positive edge starts the send process (irrespective of `bAuto`).
- `bSendInit`: Automatically sends the value once when the PLC restarts.
- `bAuto`: Automatic sending is selected when a value changes or the time `tMaxSendTime` has elapsed (polling).
- `tMinSendTime`: Parameter for automatic sending. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.
- `tMaxSendTime`: Parameter for automatic sending. The value is sent after this time has elapsed at the latest. The value 0 disables this function.
- `bDisabled`: TRUE = deselection of the block.

### VAR_OUTPUT

- `bBusy`: The `bBusy` output is TRUE as long as values are sent.
- `bError`: This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.
- `eError`: This output outputs an error code in the event of an error (see `E_LON_ERROR`). `bError` goes TRUE at the same time.
**Programming**

*Programming*

**VAR_IN_OUT**

```pascal
stLON_Com : ST_LON_Communication;
```

**VAR_IN_OUT**

This structure is used to link `FB_LON_KL6401()` with the send/receive function (see `ST_LON_Communication`).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.3.170 FB_SEND_181_SNVT_rac_ctrl

This function block sends the following LON output variable (nvo):

- **SNVT Name:** SNVT_rac_ctrl.
- **SNVT number:** 181.
- **Description:** Sound function control. Activates the sound function for a particular source.

**VAR_INPUT**

```pascal
wNVIndex : WORD;
stValue : ST_LON_SNVT_rac_ctrl;
bStart : BOOL;
bSendInit := bSendInitDefault;
bAuto := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled := FALSE;
```

- **wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **stValue:** Structure of the data to be sent (see `ST_LON_SNVT_rac_ctrl`).
- **bStart:** A positive edge starts the send process (irrespective of `bAuto`).
- **bSendInit:** Automatically sends the value once when the PLC restarts.
- **bAuto:**́ Automatic sending (is selected when a value changes or the time `tMaxSendTime` has elapsed (polling).
- **tMinSendTime:** Parameter for automatic sending (is selected when a value changes or the time `tMaxSendTime` has elapsed (polling).
- **tMaxSendTime:** Parameter for automatic sending (is selected when a value changes or the time `tMaxSendTime` has elapsed (polling).
- **bDisabled:** False;

---

**dwErrorKL:** Error ID of the function block `FB_LON_KL6401()` (see `dwErrorKL [631]`). In this case the variable `eError` has the value eKL6401_Error. `bError` goes TRUE at the same time.

**VAR_IN_OUT**

```pascal
stLON_Com : ST_LON_Communication;
```

**VAR_IN_OUT**

This structure is used to link `FB_LON_KL6401()` with the send/receive function (see `ST_LON_Communication`).
**tMaxSendTime**: Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

**bDisabled**: TRUE = deselection of the block.

### VAR_OUTPUT

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>The <strong>bBusy</strong> output is TRUE as long as values are sent.</td>
</tr>
<tr>
<td>bError</td>
<td>This output goes TRUE as soon as an error occurs. This error is described via the variable <strong>eError</strong>.</td>
</tr>
<tr>
<td>eError</td>
<td>This output goes TRUE as soon as an error occurs. This error is described via the variable <strong>eError</strong>.</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>Error ID of the function block <strong>FB_LON_KL6401</strong> (see <strong>dwErrorKL</strong>). In this case the variable <strong>eError</strong> has the value eKL6401_Error. <strong>bError</strong> goes TRUE at the same time.</td>
</tr>
</tbody>
</table>

### VAR_IN_OUT

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stLON_Com</td>
<td>This structure is used to link <strong>FB_LON_KL6401</strong> with the send/receive function (see <strong>ST_LON_Communication</strong>).</td>
</tr>
</tbody>
</table>

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.3.171 **FB_SEND_182_SNVT_rac_req**

This function block sends the following LON output variable (nvo):

**SNVT Name**: SNVT_rac_req.

**SNVT number**: 182.

**Description**: Sound function request. Requests the sound function for a particular source.

### VAR_INPUT

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>WORD;</td>
</tr>
<tr>
<td>stValue</td>
<td>ST_LON_SNVT_rac_req;</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL;</td>
</tr>
<tr>
<td>bSendInit</td>
<td>BOOL := bSendInitDefault;</td>
</tr>
<tr>
<td>bAuto</td>
<td>BOOL := bAutoDefault;</td>
</tr>
</tbody>
</table>
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

stValue: Structure of the data to be sent (see ST_LON_SNVT_rac_req [615]).

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
### FB_SEND_183_SNVT_count_32

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_count_32.

**SNVT number:** 183.

**Description:** Absolute counter. 32 bit counter.

#### VAR_INPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.</td>
</tr>
<tr>
<td>udiValue</td>
<td>Min: 0 / Max: 4294967294.</td>
</tr>
<tr>
<td>bStart</td>
<td>A positive edge starts the send process (irrespective of bAuto).</td>
</tr>
<tr>
<td>bSendInit</td>
<td>Automatically sends the value once when the PLC restarts.</td>
</tr>
<tr>
<td>bAuto</td>
<td>Automatic sending [\text{630}] is selected when a value changes or the time tMaxSendTime has elapsed (polling).</td>
</tr>
<tr>
<td>udiValueLimit</td>
<td>Parameter for automatic sending [\text{630}]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>Parameter for automatic sending [\text{630}]. A new value is sent after this time has elapsed at the earliest. This prevents continuous sending.</td>
</tr>
<tr>
<td>tMaxSendTime</td>
<td>Parameter for automatic sending [\text{630}]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (udiValueLimit) was not reached (polling). The value 0 disables this function.</td>
</tr>
<tr>
<td>bDisabled</td>
<td>TRUE = deselection of the block.</td>
</tr>
</tbody>
</table>

#### VAR_OUTPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td></td>
</tr>
<tr>
<td>bError</td>
<td></td>
</tr>
<tr>
<td>eError</td>
<td></td>
</tr>
<tr>
<td>dwErrorKL</td>
<td></td>
</tr>
</tbody>
</table>

---

**T1000** Version: 1.5

435
bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.173 FB_SEND_184_SNVT_clothes_w_c

This function block sends the following LON output variable (nvo):

SNVT Name: SNVT_clothes_w_c

SNVT number: 184

Description: Washing machine / commands. For programming and starting a washing machine

VAR_INPUT

wNVIndex : WORD;
stValue : ST_LON_SNVT_clothes_w_c;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

stValue: Structure of the data to be sent (see ST_LON_SNVT_clothes_w_c [595]).

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.
bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.174 FB_SEND_185_SNVT_clothes_w_m

This function block sends the following LON output variable (nvo):

SNVT Name: SNVT_clothes_w_m

SNVT number: 185

Description: Washing machine /management status Current status of door / lid and drain.
VAR_INPUT

wNVIndex : WORD;
stValue : ST_LON_SNVT_clothes_w_m;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

stValue: Structure of the data to be sent (see ST_LON_SNVT_clothes_w_m [595]).

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_clothes_w_s.

**SNVT number:** 186.

**Description:** Washing machine / status. Current state of a washing machine, including command and alarm information.

**VAR_INPUT**

- wNVIndex : WORD;
- stValue : ST_LON_SNVT_clothes_w_s;
- bStart : BOOL;
- bSendInit : BOOL := bSendInitDefault;
- bAuto : BOOL := bAutoDefault;
- tMinSendTime : TIME := tMinSendTimeDefault;
- tMaxSendTime : TIME := tMaxSendTimeDefault;
- bDisabled : BOOL := FALSE;

**VAR_OUTPUT**

- bBusy : BOOL;
- bError : BOOL;
- eError : E_LON_ERROR;
- dwErrorKL : DWORD;

**wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**stValue:** Structure of the data to be sent (see ST_LON_SNVT_clothes_w_s [595]).

**bStart:** A positive edge starts the send process (irrespective of bAuto).

**bSendInit:** Automatically sends the value once when the PLC restarts.

**bAuto:** Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

**tMinSendTime:** Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

**tMaxSendTime:** Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

**bDisabled:** TRUE = deselection of the block.

**bBusy:** The bBusy output is TRUE as long as values are sent.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

**eError:** This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.
**dwErrorKL**: Error ID of the function block FB_LON_KL6401 \((\text{\[27\]})\) (see dwErrorKL \((\text{\[631\]})\)). In this case the variable \(eError\) has the value eKL6401_Error. \(bError\) goes TRUE at the same time.

**VAR_IN_OUT**

\[
\text{stLON\_Com : ST\_LON\_Communication;}
\]

**stLON\_Com**: This structure is used to link FB_LON_KL6401 \((\text{\[27\]})\) with the send/receive function (see ST_LON_Communication \((\text{\[587\]})\)).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.3.176 FB_SEND_187_SNVT_clothes_w_a**

This function block sends the following LON output variable (nvo):

**SNVT Name**: SNVT_clothes_w_a.

**SNVT number**: 187.

**Description**: Washing machine / alarm messages.

**VAR_INPUT**

\[
\begin{align*}
\text{wNVIndex} & : \text{WORD;} \\
\text{stValue} & : \text{ST\_LON\_SNVT\_clothes\_w\_a;} \\
\text{bStart} & : \text{BOOL;} \\
\text{bSendInit} & : \text{BOOL} := \text{bSendInitDefault;} \\
\text{bAuto} & : \text{BOOL} := \text{bAutoDefault;} \\
\text{tMinSendTime} & : \text{TIME} := \text{tMinSendTimeDefault;} \\
\text{tMaxSendTime} & : \text{TIME} := \text{tMaxSendTimeDefault;} \\
\text{bDisabled} & : \text{BOOL} := \text{FALSE;} \\
\end{align*}
\]

**wNVIndex**: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**stValue**: Structure of the data to be sent (see ST_LON_SNVT_clothes_w_a \((\text{\[593\]})\)).

**bStart**: A positive edge starts the send process (irrespective of \(bAuto\)).

**bSendInit**: Automatically sends the value once when the PLC restarts.

**bAuto**: Automatic sending \((\text{\[630\]})\) is selected when a value changes or the time \(tMaxSendTime\) has elapsed (polling).

**tMinSendTime**: Parameter for automatic sending \((\text{\[630\]})\). A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.
tMaxSendTime: Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

**VAR_OUTPUT**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>BOOL</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>DWORD</td>
</tr>
</tbody>
</table>

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

**VAR_IN_OUT**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>stLON_Com : ST_LON_Communication</td>
<td></td>
</tr>
</tbody>
</table>

stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.3.177 FB_SEND_188_SNVT_multiplier_s**

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_multiplier_s.

**SNVT number:** 188.

**Description:** Multiplier.

**VAR_INPUT**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>WORD</td>
</tr>
<tr>
<td>rValue</td>
<td>REAL</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL</td>
</tr>
<tr>
<td>bSendInit</td>
<td>BOOL := bSendInitDefault;</td>
</tr>
</tbody>
</table>
wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

rValue: Min: 0 / Max: 2.54.

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

rValueLimit: Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401 [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_switch_2

**SNVT number:** 189

**Description:** Switch for setting scenes and settings. Extended version of the SNVT_switch for setting of scenes and settings similar to the SNVT_scene and SNVT_setting.

### VAR_INPUT

- `wNVIndex`: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- `stValue`: Structure of the data to be sent (see `ST_LON_SNVT_switch_2`).
- `bStart`: A positive edge starts the send process (irrespective of `bAuto`).
- `bSendInit`: Automatically sends the value once when the PLC restarts.
- `bAuto`: Automatic sending is selected when a value changes or the time `tMaxSendTime` has elapsed (polling).
- `tMinSendTime`: Parameter for automatic sending. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.
- `tMaxSendTime`: Parameter for automatic sending. The value is sent after this time has elapsed at the latest. The value 0 disables this function.
- `bDisabled`: TRUE = deselection of the block.

### VAR_OUTPUT

- `bBusy`: The `bBusy` output is TRUE as long as values are sent.
- `bError`: This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.
- `eError`: This output outputs an error code in the event of an error (see `E_LON_ERROR`). `bError` goes TRUE at the same time.
**dwErrorKL**: Error ID of the function block `FB_LON_KL6401()` (see `dwErrorKL` [631]). In this case the variable `eError` has the value `eKL6401_Error`. `bError` goes TRUE at the same time.

**VAR_IN_OUT**

```plaintext
stLON_Com : ST_LON_Communication;
```

`stLON_Com`: This structure is used to link `FB_LON_KL6401()` with the send/receive function (see `ST_LON_Communication` [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.3.179  **FB_SEND_190_SNVT_color_2**

This function block sends the following LON output variable (nvo):

- **SNVT Name**: `SNVT_color_2`.
- **SNVT number**: 190.
- **Description**: Color.

**VAR_INPUT**

```plaintext
wNVIndex        : WORD;
stValue         : ST_LON_SNVT_color_2;
bStart          : BOOL := bSendInitDefault;
bAuto           : BOOL := bAutoDefault;
tMinSendTime    : TIME := tMinSendTimeDefault;
tMaxSendTime    : TIME := tMaxSendTimeDefault;
bDisabled       : BOOL := FALSE;
```

- **wNVIndex**: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **stValue**: Structure of the data to be sent (see `ST_LON_SNVT_color_2` [596]).
- **bStart**: A positive edge starts the send process (irrespective of `bAuto`).
- **bSendInit**: Automatically sends the value once when the PLC restarts.
- **bAuto**: Automatic sending [630] is selected when a value changes or the time `tMaxSendTime` has elapsed (polling).
- **tMinSendTime**: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.
**tMaxSendTime**: Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

**bDisabled**: TRUE = deselection of the block.

### VAR_OUTPUT

<table>
<thead>
<tr>
<th>bBusy</th>
<th>BOOL</th>
</tr>
</thead>
<tbody>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>DWORD</td>
</tr>
</tbody>
</table>

**bBusy**: The **bBusy** output is TRUE as long as values are sent.

**bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable **eError**.

**eError**: This output outputs an error code in the event of an error (see **E_LON_ERROR** [477]). **bError** goes TRUE at the same time.

**dwErrorKL**: Error ID of the function block **FB_LON_KL6401** [27] (see **dwErrorKL** [631]). In this case the variable **eError** has the value **eKL6401_Error**. **bError** goes TRUE at the same time.

### VAR_IN_OUT

<table>
<thead>
<tr>
<th>stLON_Com</th>
<th>ST_LON_Communication</th>
</tr>
</thead>
</table>

**stLON_Com**: This structure is used to link **FB_LON_KL6401** [27] with the send/receive function (see **ST_LON_Communication** [587]).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.3.180  
**FB_SEND_191_SNVT_log_status**

This function block sends the following LON output variable (nvo):

**SNVT Name**: SNVT_log_status.

**SNVT number**: 191.

**Description**: Log status (hundredth of a second). Represents the current status of a data logging. Update based on the cpLogNotificationThreshold value. Displays only the status. Alarms are output via the node object **nvoAlarm2**. Is required if the node object contains no **nvoLogStat** output.

### VAR_INPUT

<table>
<thead>
<tr>
<th>wNVIndex</th>
<th>WORD</th>
</tr>
</thead>
<tbody>
<tr>
<td>stValue</td>
<td>ST_LON_SNVT_log_status</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL</td>
</tr>
</tbody>
</table>

**wNVIndex**: 

**stValue**: 

**bStart**: 

bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

stValue: Structure of the data to be sent (see ST_LON_SNVT_log_status [607]).

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT
bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27]() (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT
stLON_Com : ST_LON_Communication;

stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_time_stamp_p.

**SNVT number:** 192.

**Description:** Precise timestamp (seconds). Timestamp with a resolution of a hundredth second.

### VAR_INPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>wNVIndex</td>
<td>WORD</td>
</tr>
<tr>
<td>stValue</td>
<td>TIMESTRUCT</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL</td>
</tr>
<tr>
<td>bSendInit</td>
<td>BOOL := bSendInitDefault;</td>
</tr>
<tr>
<td>bAuto</td>
<td>BOOL := bAutoDefault;</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>TIME := tMinSendTimeDefault;</td>
</tr>
<tr>
<td>tMaxSendTime</td>
<td>TIME := tMaxSendTimeDefault;</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL := FALSE;</td>
</tr>
</tbody>
</table>

**wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**stValue:** Structure of the data to be sent (see TIMESTRUCT). The Element wDayOfWeek is not valid here. This value is not transferred.

**bStart:** A positive edge starts the send process (irrespective of bAuto).

**bSendInit:** Automatically sends the value once when the PLC restarts.

**bAuto:** Automatic sending is selected when a value changes or the time tMaxSendTime has elapsed (polling).

**tMinSendTime:** Parameter for automatic sending. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

**tMaxSendTime:** Parameter for automatic sending. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

**bDisabled:** TRUE = deselection of the block.

### VAR_OUTPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>BOOL</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>DWORD</td>
</tr>
</tbody>
</table>

**bBusy:** The bBusy output is TRUE as long as values are sent.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

**eError:** This output outputs an error code in the event of an error (see E_LON_ERROR). bError goes TRUE at the same time.
**Programming**

**dwErrorKL:** Error ID of the function block FB_LON_KL6401 (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

**VAR_IN_OUT**

<table>
<thead>
<tr>
<th>stLON_Com : ST_LON_Communication;</th>
</tr>
</thead>
</table>

**stLON_Com :** This structure is used to link FB_LON_KL6401 with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.3.182 FB_SEND_193_SNVT_log_fx_request

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_log_fx_request

**SNVT number:** 193

**Description:** Log file transfer request. Requests a data log via FTP transfer. Must follow a standard ftp request to obtain a data log file. Required on devices that use the data logger function profile, which enables data log transfer via FTP.

**VAR_INPUT**

| wNVIndex : WORD; |
| stValue : ST_LON_SNVT_log_fx_request; |
| bStart : BOOL; |
| bSendInit : BOOL := bSendInitDefault; |
| bAuto : BOOL := bAutoDefault; |
| tMinSendTime : TIME := tMinSendTimeDefault; |
| tMaxSendTime : TIME := tMaxSendTimeDefault; |
| bDisabled : BOOL := FALSE; |

**wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**stValue:** Structure of the data to be sent (see ST_LON_SNVT_log_fx_request [606]).

**bStart:** A positive edge starts the send process (irrespective of bAuto).

**bSendInit:** Automatically sends the value once when the PLC restarts.

**bAuto:** Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).
tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>bBusy</th>
<th>BOOL</th>
</tr>
</thead>
<tbody>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>DWORD</td>
</tr>
</tbody>
</table>

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27](). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

**VAR_IN_OUT**

| stLON_Com : ST_LON_Communication; |

stLON_Com : This structure is used to link FB_LON_KL6401() with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.3.183 FB_SEND_194_SNVT_log_fx_status**

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_log_fx_status.

**SNVT number:** 194.

**Description:** Log file transfer status. Indicates the status of a data log via FTP transfer. Required on devices that use the data logger function profile, which enables data log transfer via FTP.
**VAR_INPUT**

```plaintext
VAR_INPUT

wNVIndex : WORD;
stValue : ST_LON_SNVT_log_fx_status;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

**stValue:** Structure of the data to be sent (see ST_LON_SNVT_log_fx_request [607]).

**bStart:** A positive edge starts the send process (irrespective of bAuto).

**bSendInit:** Automatically sends the value once when the PLC restarts.

**bAuto:** Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

**tMinSendTime:** Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

**tMaxSendTime:** Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

**bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

```plaintext
VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;
```

**bBusy:** The bBusy output is TRUE as long as values are sent.

**bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

**eError:** This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

**dwErrorKL:** Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

**VAR_IN_OUT**

```plaintext
VAR_IN_OUT

stLON_Com : ST_LON_Communication;
```

**stLON_Com:** This structure is used to link FB_LON_KL6401 [27] with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_log_request.

**SNVT number:** 195.

**Description:** Log status request. Requests the current status of a data log. The status is output via the SNVT_log_status output.

### VAR_INPUT

<table>
<thead>
<tr>
<th>wNVIndex</th>
<th>WORD</th>
</tr>
</thead>
<tbody>
<tr>
<td>uiValue</td>
<td>UINT</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL</td>
</tr>
<tr>
<td>bSendInit</td>
<td>BOOL</td>
</tr>
<tr>
<td>bAuto</td>
<td>BOOL</td>
</tr>
<tr>
<td>uiValueLimit</td>
<td>UINT</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>TIME</td>
</tr>
<tr>
<td>tMaxSendTime</td>
<td>TIME</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL</td>
</tr>
</tbody>
</table>

- **wNVIndex:** Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- **uiValue:** Min: 1 / Max: 65535.
- **bStart:** A positive edge starts the send process (irrespective of bAuto).
- **bSendInit:** Automatically sends the value once when the PLC restarts.
- **bAuto:** Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).
- **uiValueLimit:** Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.
- **tMinSendTime:** Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.
- **tMaxSendTime:** Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (uiValueLimit) was not reached (polling). The value 0 disables this function.
- **bDisabled:** TRUE = deselection of the block.
**VAR_OUTPUT**

`bBusy` : BOOL;
`bError` : BOOL;
`eError` : E_LON_ERROR;
`dwErrorKL `: DWORD;

*bBusy*: The *bBusy* output is TRUE as long as values are sent.

*bError*: This output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

*eError*: This output outputs an error code in the event of an error (see [E LON ERROR](#)). *bError* goes TRUE at the same time.

*dwErrorKL*: Error ID of the function block `FB_LON_KL6401` (see [dwErrorKL](#)). In this case the variable *eError* has the value *eKL6401_Error*. *bError* goes TRUE at the same time.

**VAR_IN_OUT**

`stLON_Com` : ST_LON_Communication;

*stLON_Com*: This structure is used to link `FB_LON_KL6401()` with the send/receive function (see [ST LON Communication](#)).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.3.185 FB_SEND_196_SNVT_enthalpy_d**

This function block sends the following LON output variable (nvo):

**SNVT Name**: SNVT_enthalpy_d.

**SNVT number**: 196.

**Description**: Enthalpy difference (kJ/kg).

**VAR_INPUT**

`wNVIndex` : WORD;
`rValue` : REAL;
`bStart` : BOOL;
`bSendInit` : BOOL := bSendInitDefault;
`bAuto` : BOOL := bAutoDefault;
`rValueLimit` : REAL := 1;
`tMinSendTime` : TIME := tMinSendTimeDefault;
`tMaxSendTime` : TIME := tMaxSendTimeDefault;
`bDisabled` : BOOL := FALSE;
wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

rValue: Min: -327.68 / Max: 327.67.

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

rValueLimit: Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (rValueLimit) was not reached (polling). The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_amp_ac_mil.

**SNVT number:** 197.

**Description:** Electric current (milliampere).

### VAR_INPUT

- `wNVIndex`: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.
- `uiValue`: Min: 0 / Max: 65534.
- `bStart`: A positive edge starts the send process (irrespective of `bAuto`).
- `bSendInit`: Automatically sends the value once when the PLC restarts.
- `bAuto`: Automatic sending [630] is selected when a value changes or the time `tMaxSendTime` has elapsed (polling).
- `uiValueLimit`: Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.
- `tMinSendTime`: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.
- `tMaxSendTime`: Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (uiValueLimit) was not reached (polling). The value 0 disables this function.
- `bDisabled`: TRUE = deselection of the block.

### VAR_OUTPUT

- `bBusy`: BOOL;
- `bError`: BOOL;
- `eError`: E_LON_ERROR;
- `dwErrorKL`: DWORD;
bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

VAR_INPUT

wNVIndex : WORD;
udiValue : UDINT;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
udiValueLimit : UDINT := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;

This function block sends the following LON output variable (nvo):

SNVT Name: SNVT_time_hour_p.

SNVT number: 198.

Description: Time in hours.

VAR_INPUT

wNVIndex : WORD;
udiValue : UDINT;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
udiValueLimit : UDINT := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

udiValue: Min: 0 / Max: 4294967294.

bStart: A positive edge starts the send process (irrespective of bAuto).
bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

udiValueLimit: Parameter for automatic sending [630]. The value is only sent if the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value. This value is not available for enums and structures, where sending takes place after each change in value.

tMinSendTime: Parameter for automatic sending [630]. A new value is sent after this time has elapsed at the earliest. This prevents continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent when this time has elapsed at the latest, even if the minimum change in value (udiValueLimit) was not reached (polling). The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.188 FB_SEND_199_SNVT_lamp_status

This function block sends the following LON output variable (nvo):
SNVT Name: SNVT_lamp_status.
SNVT number: 199.
Description: Lamp status.

VAR_INPUT

wNVIndex : WORD;
stValue : ST_LON_SNVT_lamp_status;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

stValue: Structure of the data to be sent (see ST_LON_SNVT_lamp_status [606]).

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

tMaxSendTime: Parameter for automatic sending [630]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output outputs an error code in the event of an error (see E_LON_ERROR [477]). bError goes TRUE at the same time.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

stLON_Com : ST_LON_Communication;

stLON_Com : This structure is used to link FB_LON_KL6401 [27] with the send/receive function (see ST_LON_Communication [587]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.1.3.189   **FB_SEND_200_SNVT_environment**

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_environment.

**SNVT number:** 200.

**Description:** Environment.

**VAR_INPUT**

- `wNVIndex`: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

- `stValue`: Structure of the data to be sent (see `ST_LON_SNVT_environment`).

- `bStart`: A positive edge starts the send process (irrespective of `bAuto`).

- `bSendInit`: Automatically sends the value once when the PLC restarts.

- `bAuto`: Automatic sending is selected when a value changes or the time `tMaxSendTime` has elapsed (polling).

- `tMinSendTime`: Parameter for automatic sending. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.

- `tMaxSendTime`: Parameter for automatic sending. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

- `bDisabled`: TRUE = deselection of the block.

**VAR_OUTPUT**

- `bBusy`: The `bBusy` output is TRUE as long as values are sent.

- `bError`: This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.

- `eError`: This output outputs an error code in the event of an error (see `E_LON_ERROR`). `bError` goes TRUE at the same time.
**CH**
dwErrorKL: Error ID of the function block FB_LON_KL6401 [27] (see dwErrorKL [631]). In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

**VAR_IN_OUT**

stLON_Com : ST_LON_Communication;

stLON_Com: This structure is used to link FB_LON_KL6401() [27] with the send/receive function (see ST_LON_Communication [587]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.190  **FB_SEND_201_SNVT_geo_loc**

This function block sends the following LON output variable (nvo):

**SNVT Name:** SNVT_geo_loc.

**SNVT number:** 201.

**Description:** Geographical location.

**VAR_INPUT**

wNVIndex : WORD;
stValue : ST_LON_SNVT_geo_loc;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;

wNVIndex: Unique index. This index is required for binding the LON nodes. A maximum of 62 SNVTs are permitted per LON terminal. Values between 0 and 61 are possible.

stValue: Structure of the data to be sent (see ST_LON_SNVT_geo_loc [604]).

bStart: A positive edge starts the send process (irrespective of bAuto).

bSendInit: Automatically sends the value once when the PLC restarts.

bAuto: Automatic sending [630] is selected when a value changes or the time tMaxSendTime has elapsed (polling).

tMinSendTime: Parameter for automatic sending [630]. A new value is sent when this time has elapsed at the earliest, to prevent continuous sending.
tMaxSendTime: Parameter for automatic sending [tMaxSendTime]. The value is sent after this time has elapsed at the latest. The value 0 disables this function.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>BOOL</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>eError</td>
<td>E_LON_ERROR</td>
</tr>
<tr>
<td>dwErrorKL</td>
<td>DWORD</td>
</tr>
</tbody>
</table>

bBusy: The bBusy output is TRUE as long as values are sent.

bError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

eError: This output goes TRUE as soon as an error occurs. This error is described via the variable eError.

dwErrorKL: Error ID of the function block FB_LON_KL6401 [dwErrorKL]. In this case the variable eError has the value eKL6401_Error. bError goes TRUE at the same time.

VAR_IN_OUT

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>stLON_Com</td>
<td>ST_LON_Communication</td>
</tr>
</tbody>
</table>

stLON_Com: This structure is used to link FB_LON_KL6401 [stLON_Com] with the send/receive function (see ST_LON_Communication [stLON_Com]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.4 Error codes

<table>
<thead>
<tr>
<th>Value (hex)</th>
<th>Value (dec)</th>
<th>Value (enum)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0000</td>
<td>0</td>
<td>eLON_no_Error</td>
<td>No error is pending.</td>
</tr>
<tr>
<td>0x0001</td>
<td>1</td>
<td>eLON_Value_out_of_range</td>
<td>The input variable &quot;Value&quot; is outside the permitted range. The value was not sent. &quot;Value&quot; can have different formats with corresponding prefix (e.g. LREAL = lrValue).</td>
</tr>
<tr>
<td>0x0002</td>
<td>2</td>
<td>eLON_Terminal_not_ready</td>
<td>The function block &quot;FB_LON_KL6401&quot; passes through an initialization step chain (query terminal type, query firmware etc.) when the PLC is started. This message is issued as along as the initialization is in progress. If an error is pending after a PLC reset, the controller should be de-energized once.</td>
</tr>
<tr>
<td>0x0003</td>
<td>3</td>
<td>eLON_Wrong_SNVT_Typ</td>
<td>The received SNVT type does not match the SNVT type of the addressed NV index (input variable &quot;wId&quot;).</td>
</tr>
<tr>
<td>0x0004</td>
<td>4</td>
<td>eLON_Wrong_wNVIndex</td>
<td>Incorrect NV index.</td>
</tr>
<tr>
<td>0x0005</td>
<td>5</td>
<td>eKL6401_Wrong_Terminal</td>
<td>No KL6401 was detected.</td>
</tr>
<tr>
<td>0x0006</td>
<td>6</td>
<td>eKL6401_Error</td>
<td>The function block &quot;FB_LON_KL6401&quot; has an error. The error code is shown at output &quot;dwErrorKL&quot;.</td>
</tr>
<tr>
<td>0x0007</td>
<td>7</td>
<td>eKL6401_Terminal_is_not_initialized</td>
<td>The terminal is not initialized. This message usually means that there is no connection to the terminal. Terminal linked to the variables in the System Manager? Terminal plugged in incorrectly? Everything corrected, everything translated and re-read into the System Manager?</td>
</tr>
<tr>
<td>0x0032</td>
<td>50</td>
<td>eLON_L_star_Out_of_range</td>
<td>SNVT 70 / The input variable &quot;stValue.L_star&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0033</td>
<td>51</td>
<td>eLON_A_star_Out_of_range</td>
<td>SNVT 70 / The input variable &quot;stValue.A_star&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0034</td>
<td>52</td>
<td>eLON_B_star_Out_of_range</td>
<td>SNVT 70 / The input variable &quot;stValue.B_star&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0037</td>
<td>55</td>
<td>eLON_eRequest_Out_of_range</td>
<td>SNVT 73 / The input variable &quot;stValue.eRequest&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>Value (hex)</td>
<td>Value (dec)</td>
<td>Value (enum)</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
<td>----------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>0x0042</td>
<td>66</td>
<td>eLON_wYear_Out_of_range</td>
<td>SNVT 084 / 088 / The input variable &quot;stValue.wYear&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0043</td>
<td>67</td>
<td>eLON_wMonth_Out_of_range</td>
<td>SNVT 084 / 088 / The input variable &quot;stValue.wMonth&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0044</td>
<td>68</td>
<td>eLON_wDay_Out_of_range</td>
<td>eLON_wDay_Out_of_range: SNVT 084 / 088 / The input variable &quot;stValue.wDay&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0045</td>
<td>69</td>
<td>eLON_wHour_Out_of_range</td>
<td>SNVT 084 / 088 / The input variable &quot;stValue.wHour&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0046</td>
<td>70</td>
<td>eLON_wMinute_Out_of_range</td>
<td>SNVT 084 / 088 / The input variable &quot;stValue.wMinute&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0047</td>
<td>71</td>
<td>eLON_wSecond_Out_of_range</td>
<td>SNVT 084 / 088 / The input variable &quot;stValue.wSecond&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0048</td>
<td>72</td>
<td>eLON_wMillisecond_Out_of_range</td>
<td>SNVT 73 / The input variable &quot;stValue.wMillisecond&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0050</td>
<td>80</td>
<td>eLON_rZero_Out_of_range</td>
<td>SNVT 085 / The input variable &quot;stValue.rZero&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0051</td>
<td>81</td>
<td>eLON_rSpan_Out_of_range</td>
<td>SNVT 085 / The input variable &quot;stValue.rSpan&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0055</td>
<td>85</td>
<td>eLON_arrValue01_Out_of_range</td>
<td>SNVT 086 / The input variable &quot;arrValue[1]&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0056</td>
<td>86</td>
<td>eLON_arrValue02_Out_of_range</td>
<td>SNVT 086 / The input variable &quot;arrValue[2]&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0057</td>
<td>87</td>
<td>eLON_arrValue03_Out_of_range</td>
<td>SNVT 086 / The input variable &quot;arrValue[3]&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0058</td>
<td>88</td>
<td>eLON_arrValue04_Out_of_range</td>
<td>SNVT 086 / The input variable &quot;arrValue[4]&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0059</td>
<td>89</td>
<td>eLON_arrValue05_Out_of_range</td>
<td>SNVT 086 / The input variable &quot;arrValue[5]&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x005A</td>
<td>90</td>
<td>eLON_arrValue06_Out_of_range</td>
<td>SNVT 086 / The input variable &quot;arrValue[6]&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x005B</td>
<td>91</td>
<td>eLON_arrValue07_Out_of_range</td>
<td>SNVT 086 / The input variable &quot;arrValue[7]&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x005C</td>
<td>92</td>
<td>eLON_arrValue08_Out_of_range</td>
<td>SNVT 086 / The input variable &quot;arrValue[8]&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x005D</td>
<td>93</td>
<td>eLON_arrValue09_Out_of_range</td>
<td>SNVT 086 / The input variable &quot;arrValue[9]&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0064</td>
<td>100</td>
<td>eLON_arrValue10_Out_of_range</td>
<td>SNVT 086 / The input variable &quot;arrValue[10]&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0065</td>
<td>101</td>
<td>eLON_arrValue11_Out_of_range</td>
<td>SNVT 086 / The input variable &quot;arrValue[11]&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0066</td>
<td>102</td>
<td>eLON_arrValue12_Out_of_range</td>
<td>SNVT 086 / The input variable &quot;arrValue[12]&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0067</td>
<td>103</td>
<td>eLON_arrValue13_Out_of_range</td>
<td>SNVT 086 / The input variable &quot;arrValue[13]&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0068</td>
<td>104</td>
<td>eLON_arrValue14_Out_of_range</td>
<td>SNVT 086 / The input variable &quot;arrValue[14]&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0069</td>
<td>105</td>
<td>eLON_arrValue15_Out_of_range</td>
<td>SNVT 086 / The input variable &quot;arrValue[15]&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x006A</td>
<td>106</td>
<td>eLON_arrValue16_Out_of_range</td>
<td>SNVT 086 / The input variable &quot;arrValue[16]&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x006B</td>
<td>107</td>
<td>eLON_arrValue17_Out_of_range</td>
<td>SNVT 086 / The input variable &quot;arrValue[17]&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x006C</td>
<td>108</td>
<td>eLON_arrValue18_Out_of_range</td>
<td>SNVT 086 / The input variable &quot;arrValue[18]&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x006D</td>
<td>109</td>
<td>eLON_arrValue19_Out_of_range</td>
<td>SNVT 086 / The input variable &quot;arrValue[19]&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0073</td>
<td>115</td>
<td>eLON_arrValue20_Out_of_range</td>
<td>SNVT 086 / The input variable &quot;arrValue[20]&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0074</td>
<td>116</td>
<td>eLON_arrValue21_Out_of_range</td>
<td>SNVT 086 / The input variable &quot;arrValue[21]&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0075</td>
<td>117</td>
<td>eLON_arrValue22_Out_of_range</td>
<td>SNVT 086 / The input variable &quot;arrValue[22]&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0076</td>
<td>118</td>
<td>eLON_arrValue23_Out_of_range</td>
<td>SNVT 086 / The input variable &quot;arrValue[23]&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>Value (hex)</td>
<td>Value (dec)</td>
<td>Value (enum)</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>0x0077</td>
<td>119</td>
<td>eLON_arrValue24_Out_of_range</td>
<td>SNVT 086 / The input variable &quot;arrValue[24]&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0078</td>
<td>120</td>
<td>eLON_arrValue25_Out_of_range</td>
<td>SNVT 086 / The input variable &quot;arrValue[25]&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0079</td>
<td>121</td>
<td>eLON_arrValue26_Out_of_range</td>
<td>SNVT 086 / The input variable &quot;arrValue[26]&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x007A</td>
<td>122</td>
<td>eLON_arrValue27_Out_of_range</td>
<td>SNVT 086 / The input variable &quot;arrValue[27]&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x007B</td>
<td>123</td>
<td>eLON_arrValue28_Out_of_range</td>
<td>SNVT 086 / The input variable &quot;arrValue[28]&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x007C</td>
<td>124</td>
<td>eLON_arrValue29_Out_of_range</td>
<td>SNVT 086 / The input variable &quot;arrValue[29]&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0082</td>
<td>130</td>
<td>eLON_arrValue30_Out_of_range</td>
<td>SNVT 086 / The input variable &quot;arrValue[30]&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0083</td>
<td>131</td>
<td>eLON_arrValue31_Out_of_range</td>
<td>SNVT 086 / The input variable &quot;arrValue[31]&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0084</td>
<td>132</td>
<td>eLON_arrValue32_Out_of_range</td>
<td>SNVT 086 / The input variable &quot;arrValue[32]&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0085</td>
<td>133</td>
<td>eLON_arrValue33_Out_of_range</td>
<td>SNVT 086 / The input variable &quot;arrValue[33]&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0086</td>
<td>134</td>
<td>eLON_arrValue34_Out_of_range</td>
<td>SNVT 086 / The input variable &quot;arrValue[34]&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0087</td>
<td>135</td>
<td>eLON_arrValue35_Out_of_range</td>
<td>SNVT 086 / The input variable &quot;arrValue[35]&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0088</td>
<td>136</td>
<td>eLON_arrValue36_Out_of_range</td>
<td>SNVT 086 / The input variable &quot;arrValue[36]&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0089</td>
<td>137</td>
<td>eLON_arrValue37_Out_of_range</td>
<td>SNVT 086 / The input variable &quot;arrValue[37]&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x008A</td>
<td>138</td>
<td>eLON_arrValue38_Out_of_range</td>
<td>SNVT 086 / The input variable &quot;arrValue[38]&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x008B</td>
<td>139</td>
<td>eLON_arrValue39_Out_of_range</td>
<td>SNVT 086 / The input variable &quot;arrValue[39]&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x008C</td>
<td>140</td>
<td>eLON_arrValue40_Out_of_range</td>
<td>SNVT 086 / The input variable &quot;arrValue[40]&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0091</td>
<td>145</td>
<td>eLON_087uiDay_Out_of_range</td>
<td>SNVT 087 / The input variable &quot;stValue.uiDay&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0092</td>
<td>146</td>
<td>eLON_087uiHour_Out_of_range</td>
<td>SNVT 087 / The input variable &quot;stValue.uiHour&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0093</td>
<td>147</td>
<td>eLON_087uiMinute_Out_of_range</td>
<td>SNVT 087 / The input variable &quot;stValue.uiMinute&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0094</td>
<td>148</td>
<td>eLON_087uiSecond_Out_of_range</td>
<td>SNVT 087 / The input variable &quot;stValue.uiSecond&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0095</td>
<td>149</td>
<td>eLON_087uiMillisecond_Out_of_range</td>
<td>SNVT 087 / The input variable &quot;stValue.uiMillisecond&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x009B</td>
<td>155</td>
<td>eLON_ePriority_level_Out_of_range</td>
<td>SNVT 088 / The input variable &quot;stValue.ePriority_level&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x009C</td>
<td>156</td>
<td>eLON_eAlarm_type_Out_of_range</td>
<td>SNVT 088 / The input variable &quot;stValue.eAlarm_type&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x00A0</td>
<td>160</td>
<td>eLON_Currency_Out_of_range</td>
<td>SNVT 089 / The input variable &quot;stValue.Currency&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x00A5</td>
<td>165</td>
<td>eLON_diRw_ptr_Out_of_range</td>
<td>SNVT 090 / The input variable &quot;stValue.diRw_ptr&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x00AA</td>
<td>170</td>
<td>eLON_Object_request_Out_of_range</td>
<td>SNVT 092 / The input variable &quot;stValue.Object_request&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x00AF</td>
<td>175</td>
<td>eLON_094eLearn_Out_of_range</td>
<td>SNVT 094 / The input variable &quot;stValue.eLearn&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x00B0</td>
<td>176</td>
<td>eLON_094uiHour_Out_of_range</td>
<td>SNVT 094 / The input variable &quot;stValue.uiHour&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x00B1</td>
<td>177</td>
<td>eLON_094uiMinute_Out_of_range</td>
<td>SNVT 094 / The input variable &quot;stValue.uiMinute&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x00B2</td>
<td>178</td>
<td>eLON_094uiSecond_Out_of_range</td>
<td>SNVT 094 / The input variable &quot;stValue.uiSecond&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x00B3</td>
<td>179</td>
<td>eLON_094uiMillisecond_Out_of_range</td>
<td>SNVT 094 / The input variable &quot;stValue.uiMillisecond&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>Value (hex)</td>
<td>Value (dec)</td>
<td>Value (enum)</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>0x00B9</td>
<td>185</td>
<td>eLON_095rValue_Out_of_range</td>
<td>SNVT 095 / The input variable &quot;stValue.rValue&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x00BA</td>
<td>186</td>
<td>eLON_095siState_Out_of_range</td>
<td>SNVT 095 / The input variable &quot;stValue.siState&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x00BE</td>
<td>190</td>
<td>eLON_byInterp_pts_0_to_1_Out_of_range</td>
<td>SNVT 096 / The input variable &quot;stValue.byInterp_pts_0_to_1&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x00BF</td>
<td>191</td>
<td>eLON_byInterp_pts_1_to_2_Out_of_range</td>
<td>SNVT 096 / The input variable &quot;stValue.byInterp_pts_1_to_2&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x00C0</td>
<td>192</td>
<td>eLON_byInterp_pts_2_to_3_Out_of_range</td>
<td>SNVT 096 / The input variable &quot;stValue.byInterp_pts_2_to_3&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x00C1</td>
<td>193</td>
<td>eLON_byInterp_pts_3_to_4_Out_of_range</td>
<td>SNVT 096 / The input variable &quot;stValue.byInterp_pts_3_to_4&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x00C2</td>
<td>194</td>
<td>eLON_byInterp_pts_4_to_5_Out_of_range</td>
<td>SNVT 096 / The input variable &quot;stValue.byInterp_pts_4_to_5&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x00C3</td>
<td>195</td>
<td>eLON_byInterp_pts_5_to_6_Out_of_range</td>
<td>SNVT 096 / The input variable &quot;stValue.byInterp_pts_5_to_6&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x00C4</td>
<td>196</td>
<td>eLON_byInterp_pts_6_to_0_Out_of_range</td>
<td>SNVT 096 / The input variable &quot;stValue.byInterp_pts_6_to_0&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x00C8</td>
<td>200</td>
<td>eLON_rOccupied_cool_Out_of_range</td>
<td>SNVT 106 / The input variable &quot;stValue.rOccupied_cool&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x00C9</td>
<td>201</td>
<td>eLON_rStandby_cool_Out_of_range</td>
<td>SNVT 106 / The input variable &quot;stValue.rStandby_cool&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x00CA</td>
<td>202</td>
<td>eLON_rUnoccupied_cool_Out_of_range</td>
<td>SNVT 106 / The input variable &quot;stValue.rUnoccupied_cool&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x00CB</td>
<td>203</td>
<td>eLON_rOccupied_heat_Out_of_range</td>
<td>SNVT 106 / The input variable &quot;stValue.rOccupied_heat&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x00CC</td>
<td>204</td>
<td>eLON_rStandby_heat_Out_of_range</td>
<td>SNVT 106 / The input variable &quot;stValue.rStandby_heat&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x00CD</td>
<td>205</td>
<td>eLON_rUnoccupied_heat_Out_of_range</td>
<td>SNVT 106 / The input variable &quot;stValue.rUnoccupied_heat&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x00D2</td>
<td>210</td>
<td>eLON_111rPercent_Out_of_range</td>
<td>SNVT 111 / The input variable &quot;stValue.rPercent&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x00D3</td>
<td>211</td>
<td>eLON_111siState_Out_of_range</td>
<td>SNVT 111 / The input variable &quot;stValue.siState&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x00D7</td>
<td>215</td>
<td>eLON_eMode_Out_of_range</td>
<td>SNVT 112 / The input variable &quot;stValue.eMode&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x00D8</td>
<td>216</td>
<td>eLON_rHeat_output_primary_Out_of_range</td>
<td>SNVT 112 / The input variable &quot;stValue.rHeat_output_primary&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x00D9</td>
<td>217</td>
<td>eLON_rHeat_output_secondary_Out_of_range</td>
<td>SNVT 112 / The input variable &quot;stValue.rHeat_output_secondary&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x00DA</td>
<td>218</td>
<td>eLON_rCool_output_Out_of_range</td>
<td>SNVT 112 / The input variable &quot;stValue.rCool_output&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x00DB</td>
<td>219</td>
<td>eLON_rEcon_output_Out_of_range</td>
<td>SNVT 112 / The input variable &quot;stValue.rEcon_output&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x00DC</td>
<td>220</td>
<td>eLON_rFan_output_Out_of_range</td>
<td>SNVT 112 / The input variable &quot;stValue.rFan_output&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x00E1</td>
<td>225</td>
<td>eLON_115eFunction_Out_of_range</td>
<td>SNVT 115 / The input variable &quot;stValue.eFunction&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x00E2</td>
<td>226</td>
<td>eLON_eFunction_Out_of_range</td>
<td>SNVT 116 / The input variable &quot;stValue.eFunction&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x00E3</td>
<td>227</td>
<td>eLON_rSetting_Out_of_range</td>
<td>SNVT 116 / The input variable &quot;stValue.rSetting&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x00E4</td>
<td>228</td>
<td>eLON_rRotation_Out_of_range</td>
<td>SNVT 116 / The input variable &quot;stValue.rRotation&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x00E5</td>
<td>229</td>
<td>eLON_rFade_time_Out_of_range</td>
<td>SNVT 116 / The input variable &quot;stValue.rFade_time&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x00E6</td>
<td>230</td>
<td>eLON_rDelay_time_Out_of_range</td>
<td>SNVT 116 / The input variable &quot;stValue.rDelay_time&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x00EB</td>
<td>235</td>
<td>eLON_eChlr_run_mode_Out_of_range</td>
<td>SNVT 127 / The input variable &quot;stValue.eChlr_run_mode&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x00EC</td>
<td>236</td>
<td>eLON_eChlr_op_mode_Out_of_range</td>
<td>SNVT 127 / The input variable &quot;stValue.eChlr_op_mode&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x00F0</td>
<td>240</td>
<td>eLON_eNext_state_Out_of_range</td>
<td>SNVT 128 / The input variable &quot;stValue.eNext_state&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>Value (hex)</td>
<td>Value (dec)</td>
<td>Value (enum)</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>------------</td>
<td>-------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>0x00F1</td>
<td>241</td>
<td>eLON_eCurrent_state_Out_of_range</td>
<td>SNVT 128 / The input variable &quot;stValue.eCurrent_state&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x00F5</td>
<td>245</td>
<td>eLON_diSecond_time_offset_Out_of_range</td>
<td>SNVT 134 / The input variable &quot;stValue.diSecond_time_offset&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x00F6</td>
<td>246</td>
<td>eLON_eType_of_description_Out_of_range</td>
<td>SNVT 134 / The input variable &quot;stValue.eType_of_description&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x00FA</td>
<td>250</td>
<td>eLON_byHour_of_start_DST_Out_of_range</td>
<td>SNVT 134 / The input variable &quot;stValue.byHour_of_start_DST&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x00FB</td>
<td>251</td>
<td>eLON_byMinute_of_start_DST_Out_of_range</td>
<td>SNVT 134 / The input variable &quot;stValue.byMinute_of_start_DST&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x00FC</td>
<td>252</td>
<td>eLON_bySecond_of_start_DST_Out_of_range</td>
<td>SNVT 134 / The input variable &quot;stValue.bySecond_of_start_DST&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0104</td>
<td>260</td>
<td>eLON_byHour_of_end_DST_Out_of_range</td>
<td>SNVT 134 / The input variable &quot;stValue.byHour_of_end_DST&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0105</td>
<td>261</td>
<td>eLON_byMinute_of_end_DST_Out_of_range</td>
<td>SNVT 134 / The input variable &quot;stValue.byMinute_of_end_DST&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0106</td>
<td>262</td>
<td>eLON_bySecond_of_end_DST_Out_of_range</td>
<td>SNVT 134 / The input variable &quot;stValue.bySecond_of_end_DST&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0107</td>
<td>263</td>
<td>eLON_stStart_DST_uiG_day_of_start_DST_Out_of_range</td>
<td>SNVT 134 / The input variable &quot;stValue.stStart_DST.uiG_day_of_start_DST&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0108</td>
<td>264</td>
<td>eLON_stStart_DST_uiJ_day_of_start_DST_Out_of_range</td>
<td>SNVT 134 / The input variable &quot;stValue.stStart_DST.uiJ_day_of_start_DST&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0109</td>
<td>265</td>
<td>eLON_stStart_DST_stM_start_DST_byMonth_of_start_DST_Out_of_range</td>
<td>SNVT 134 / The input variable &quot;stValue.stStart_DST.stM_start_DST.byMonth_of_start_DST&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x010A</td>
<td>266</td>
<td>eLON_stStart_DST_stM_start_DST_byWeek_of_start_DST_Out_of_range</td>
<td>SNVT 134 / The input variable &quot;stValue.stStart_DST.stM_start_DST.byWeek_of_start_DST&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x010B</td>
<td>267</td>
<td>eLON_stStart_DST_stM_start_DST_eDateDay_of_start_DST_Out_of_range</td>
<td>SNVT 134 / The input variable &quot;stValue.stStart_DST.stM_start_DST.eDateDay_of_start_DST&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x010C</td>
<td>268</td>
<td>eLON_stEnd_DST_uiG_day_of_end_DST_Out_of_range</td>
<td>SNVT 134 / The input variable &quot;stValue.stEnd_DST.uiG_day_of_end_DST&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x010D</td>
<td>269</td>
<td>eLON_stEnd_DST_uiJ_day_of_end_DST_Out_of_range</td>
<td>SNVT 134 / The input variable &quot;stValue.stEnd_DST.uiJ_day_of_end_DST&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x010E</td>
<td>270</td>
<td>eLON_stEnd_DST_stM_end_DST_byMonth_of_end_DST_Out_of_range</td>
<td>SNVT 134 / The input variable &quot;stValue.stEnd_DST.stM_end_DST.byMonth_of_end_DST&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x010F</td>
<td>271</td>
<td>eLON_stEnd_DST_stM_end_DST_byWeek_of_end_DST_Out_of_range</td>
<td>SNVT 134 / The input variable &quot;stValue.stEnd_DST.stM_end_DST.byWeek_of_end_DST&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0110</td>
<td>272</td>
<td>eLON_stEnd_DST_stM_end_DST_eDateDay_of_end_DST_Out_of_range</td>
<td>SNVT 134 / The input variable &quot;stValue.stEnd_DST.stM_end_DST.eDateDay_of_end_DST&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0118</td>
<td>280</td>
<td>eLON_byLatitude_deg_Out_of_range</td>
<td>SNVT 135 / The input variable &quot;stValue.byLatitude&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0119</td>
<td>281</td>
<td>eLON_byLatitude_min_Out_of_range</td>
<td>SNVT 135 / The input variable &quot;stValue.byLatitude&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x011A</td>
<td>282</td>
<td>eLON_bylongitude_deg_Out_of_range</td>
<td>SNVT 135 / The input variable &quot;stValue.bylongitude_deg&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x011B</td>
<td>283</td>
<td>eLON_byLongitude_min_Out_of_range</td>
<td>SNVT 135 / The input variable &quot;stValue.byLongitude_min&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0122</td>
<td>290</td>
<td>eLON_byNr_decimals_Out_of_range</td>
<td>SNVT 136 / The input variable &quot;stValue.byNr_decimals&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0123</td>
<td>291</td>
<td>eLON_eUnit_Out_of_range</td>
<td>SNVT 136 / The input variable &quot;stValue.eUnit&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0127</td>
<td>295</td>
<td>eLON_137eUnit_Out_of_range</td>
<td>SNVT 137 / The input variable &quot;stValue.eUnit&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0128</td>
<td>296</td>
<td>eLON_137byNr_decimals_Out_of_range</td>
<td>SNVT 137 / The input variable &quot;stValue.byNr_decimals&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0129</td>
<td>297</td>
<td>eLON_137byStatus_Out_of_range</td>
<td>SNVT 137 / The input variable &quot;stValue.byStatus&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>Value (hex)</td>
<td>Value (dec)</td>
<td>Value (enum)</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>0x012A</td>
<td>298</td>
<td>eLON_137uiYear_Out_of_range</td>
<td>SNVT 137 / The input variable &quot;stValue.uiYear&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x012B</td>
<td>299</td>
<td>eLON_137uiMonth_Out_of_range</td>
<td>SNVT 137 / The input variable &quot;stValue.uiMonth&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x012C</td>
<td>300</td>
<td>eLON_137uiDay_Out_of_range</td>
<td>SNVT 137 / The input variable &quot;stValue.uiDay&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x012D</td>
<td>301</td>
<td>eLON_137uiHour_Out_of_range</td>
<td>SNVT 137 / The input variable &quot;stValue.uiHour&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x012E</td>
<td>302</td>
<td>eLON_137uiMinute_Out_of_range</td>
<td>SNVT 137 / The input variable &quot;stValue.uiMinute&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x012F</td>
<td>303</td>
<td>eLON_137uiSecond_Out_of_range</td>
<td>SNVT 137 / The input variable &quot;stValue.uiSecond&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0130</td>
<td>310</td>
<td>eLON_bySender_prio_Out_of_range</td>
<td>SNVT 148 / The input variable &quot;stValue.bySender_prio&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0131</td>
<td>315</td>
<td>eLON_eStatus_Out_of_range</td>
<td>SNVT 149 / The input variable &quot;stValue.eStatus&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0132</td>
<td>316</td>
<td>eLON_stSender_uiID_Out_of_range</td>
<td>SNVT 149 / The input variable &quot;stValue.stSender.uiID&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0133</td>
<td>317</td>
<td>eLON_stSender_stRange.uiLower_Out_of_range</td>
<td>SNVT 149 / The input variable &quot;stValue.stSender.stRange.uiLower&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0134</td>
<td>318</td>
<td>eLON_stSender_stRange.uiUpper_Out_of_range</td>
<td>SNVT 149 / The input variable &quot;stValue.stSender.stRange.uiUpper&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0135</td>
<td>319</td>
<td>eLON_uiController_id_Out_of_range</td>
<td>SNVT 149 / The input variable &quot;stValue.uiController&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0136</td>
<td>325</td>
<td>eLON_ePan_dir_Out_of_range</td>
<td>SNVT 150 / The input variable &quot;stValue.ePan&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0137</td>
<td>326</td>
<td>eLON_rPan_speed_Out_of_range</td>
<td>SNVT 150 / The input variable &quot;stValue.rPan&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0138</td>
<td>327</td>
<td>eLON_eTilt_dir_Out_of_range</td>
<td>SNVT 150 / The input variable &quot;stValue.eTilt&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0139</td>
<td>328</td>
<td>eLON_rTilt_speed_Out_of_range</td>
<td>SNVT 150 / The input variable &quot;stValue.rTilt&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x013A</td>
<td>329</td>
<td>eLON_eZoom_Out_of_range</td>
<td>SNVT 150 / The input variable &quot;stValue.eZoom&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x013B</td>
<td>330</td>
<td>eLON_rZoom_speed_Out_of_range</td>
<td>SNVT 150 / The input variable &quot;stValue.rZoom&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x013C</td>
<td>335</td>
<td>eLON_eAction_Out_of_range</td>
<td>SNVT 151 / The input variable &quot;stValue.eAction&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x013D</td>
<td>340</td>
<td>eLON_byController_prio_Out_of_range</td>
<td>SNVT 152 / The input variable &quot;stValue.byController_prio&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x013E</td>
<td>341</td>
<td>eLON_152eFunction_Out_of_range</td>
<td>SNVT 152 / The input variable &quot;stValue.152eFunction&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x013F</td>
<td>342</td>
<td>eLON_152eAction_Out_of_range</td>
<td>SNVT 152 / The input variable &quot;stValue.152eAction&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0140</td>
<td>343</td>
<td>eLON_stValue_stAbspos_rZoom_Out_of_range</td>
<td>SNVT 152 / The input variable &quot;stValue.stAbspos.rZoom&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0141</td>
<td>344</td>
<td>eLON_stValue_stAbspos_rTilt_Out_of_range</td>
<td>SNVT 152 / The input variable &quot;stValue.stAbspos.rTilt&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0142</td>
<td>345</td>
<td>eLON_stValue_stAbspos_rPan_Out_of_range</td>
<td>SNVT 152 / The input variable &quot;stValue.stAbspos.rPan&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0143</td>
<td>350</td>
<td>eLON_eMain_pump_Out_of_range</td>
<td>SNVT 156 / The input variable &quot;stValue.eMain_pump&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0144</td>
<td>351</td>
<td>eLON_eBooster_pump_Out_of_range</td>
<td>SNVT 156 / The input variable &quot;stValue.eBooster_pump&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0145</td>
<td>352</td>
<td>eLON_ePriority_level_Out_of_range</td>
<td>SNVT 156 / The input variable &quot;stValue.ePriority_level&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0146</td>
<td>353</td>
<td>eLON_eProcess_ready_Out_of_range</td>
<td>SNVT 156 / The input variable &quot;stValue.eProcess_ready&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0147</td>
<td>354</td>
<td>eLON_eEmergency_stop_activated_Out_of_range</td>
<td>SNVT 156 / The input variable &quot;stValue.eEmergency_stop_activated&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0148</td>
<td>355</td>
<td>eLON_eMain_pump_drive_enabled_Out_of_range</td>
<td>SNVT 156 / The input variable &quot;stValue.eMain_pump_drive_enabled&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0149</td>
<td>356</td>
<td>eLON_eBooster_pump_drive_enabled_Out_of_range</td>
<td>SNVT 156 / The input variable &quot;stValue.eBooster_pump_drive_enabled&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>Value (hex)</td>
<td>Value (dec)</td>
<td>Value (enum)</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>0x0165</td>
<td>357</td>
<td>eLON_eMaintenance_required_Out_of_range</td>
<td>SNVT 156 / The input variable “stValue.eMaintenance_required” is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x016D</td>
<td>365</td>
<td>eLON_eControl_status_Out_of_range</td>
<td>SNVT 157 / The input variable “stValue.eControl_status” is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x016E</td>
<td>366</td>
<td>eLON_stControl_device_addr_byDomain_length_Out_of_range</td>
<td>SNVT 157 / The input variable “stValue.stControl_device_addr_byDomain_length” is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x016F</td>
<td>367</td>
<td>eLON_stControl_device_addr_bySubnet_Out_of_range</td>
<td>SNVT 157 / The input variable “stValue.stControl_device_addr_bySubnet” is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0170</td>
<td>368</td>
<td>eLON_stControl_device_addr_byNode_Out_of_range</td>
<td>SNVT 157 / The input variable “stValue.stControl_device_addr_byNode” is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0177</td>
<td>375</td>
<td>eLON_rExhaust_temperature_Out_of_range</td>
<td>SNVT 158 / The input variable “stValue.rExhaust_temperature” is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0178</td>
<td>376</td>
<td>eLON_rExhaust_pressure_Out_of_range</td>
<td>SNVT 158 / The input variable “stValue.rExhaust_pressure” is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0179</td>
<td>377</td>
<td>eLON_rShaft_seal_purge_pressure_Out_of_range</td>
<td>SNVT 158 / The input variable “stValue.rShaft_seal_purge_pressure” is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x017A</td>
<td>378</td>
<td>eLON_rSupply_voltage_Out_of_range</td>
<td>SNVT 158 / The input variable “stValue.rSupply_voltage” is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x017B</td>
<td>379</td>
<td>eLON_eCoolant_flow_low_Out_of_range</td>
<td>SNVT 158 / The input variable “stValue.eCoolant_flow_low” is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x017C</td>
<td>380</td>
<td>eLON_eDilution_active_Out_of_range</td>
<td>SNVT 158 / The input variable “stValue.eDilution_active” is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x017D</td>
<td>381</td>
<td>eLON_eBallast_dilution_active_Out_of_range</td>
<td>SNVT 158 / The input variable “stValue.eBallast_dilution_active” is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x017E</td>
<td>382</td>
<td>eLON_eInlet_purge_dilution_active_Out_of_range</td>
<td>SNVT 158 / The input variable “stValue.eInlet_purge_dilution_active” is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x017F</td>
<td>383</td>
<td>eLON_eExhaust_dilution_active_Out_of_range</td>
<td>SNVT 158 / The input variable “stValue.eExhaust_dilution_active” is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0180</td>
<td>384</td>
<td>eLON_eDilution_flow_Out_of_range</td>
<td>SNVT 158 / The input variable “stValue.eDilution_flow” is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0181</td>
<td>385</td>
<td>eLON_ePower_supply_on_Out_of_range</td>
<td>SNVT 158 / The input variable “stValue.ePower_supply_on” is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0186</td>
<td>390</td>
<td>eLON_rRotational_speed_Out_of_range</td>
<td>SNVT 159 / The input variable “stValue.rRotational_speed” is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0187</td>
<td>391</td>
<td>eLON_rBody_temperature_Out_of_range</td>
<td>SNVT 159 / The input variable “stValue.rBody” is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0188</td>
<td>392</td>
<td>eLON_rMotor_external_temperature_Out_of_range</td>
<td>SNVT 159 / The input variable “stValue.rMotor_external_temperature” is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0189</td>
<td>393</td>
<td>eLON_eMotor_overloaded_Out_of_range</td>
<td>SNVT 159 / The input variable “stValue.eMotor_overloaded” is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x018A</td>
<td>394</td>
<td>eLON_eMotor_overloaded_Out_of_range</td>
<td>SNVT 159 / The input variable “stValue.eMotor_overloaded” is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x018B</td>
<td>395</td>
<td>eLON_eOil_level_low_Out_of_range</td>
<td>SNVT 159 / The input variable “stValue.eOil_level_low” is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x018C</td>
<td>396</td>
<td>eLON_ePhase imbalance_detected_Out_of_range</td>
<td>SNVT 159 / The input variable “stValue.ePhase imbalance detected” is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x018D</td>
<td>397</td>
<td>eLON_rCurrent_usage_Out_of_range</td>
<td>SNVT 159 / The input variable “stValue.rCurrent_usage” is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x018E</td>
<td>398</td>
<td>eLON_rPower_usage_Out_of_range</td>
<td>SNVT 159 / The input variable “stValue.rPower_usage” is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x018F</td>
<td>399</td>
<td>eLON_eTemperature_control_Out_of_range</td>
<td>SNVT 159 / The input variable “stValue.eTemperature_control” is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0190</td>
<td>400</td>
<td>eLON_eElectromagnetic_brake_active_Out_of_range</td>
<td>SNVT 159 / The input variable “stValue.eElectromagnetic_brake_active” is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0191</td>
<td>401</td>
<td>eLON_eFriction_brake_active_Out_of_range</td>
<td>SNVT 159 / The input variable “stValue.eFriction_brake_active” is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0192</td>
<td>402</td>
<td>eLON_eGas_brake_active_Out_of_range</td>
<td>SNVT 159 / The input variable “stValue.eGas_brake_active” is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x019A</td>
<td>410</td>
<td>eLON_164Milliseconds_Out_of_range</td>
<td>SNVT 164 / The input variable “stValue.164Milliseconds” is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x019B</td>
<td>411</td>
<td>eLON_164Priority_level_Out_of_range</td>
<td>SNVT 164 / The input variable “stValue.164Priority_level” is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x019C</td>
<td>412</td>
<td>eLON_164eAlarm_type_Out_of_range</td>
<td>SNVT 164 / The input variable “stValue.eAlarm” is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>Value (hex)</td>
<td>Value (dec)</td>
<td>Value (enum)</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>0x01A4</td>
<td>420</td>
<td>eLON_byType_scope_Out_of_range</td>
<td>SNVT 166 / The input variable “stValue.byType_scope” is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01A5</td>
<td>421</td>
<td>eLON_uiType_index_Out_of_range</td>
<td>SNVT 166 / The input variable “stValue.uiType_index” is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01A6</td>
<td>422</td>
<td>eLON_eType_category_Out_of_range</td>
<td>SNVT 166 / The input variable “stValue.eType_category” is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01A7</td>
<td>423</td>
<td>eLON_byType_length_Out_of_range</td>
<td>SNVT 166 / The input variable “stValue.byType” is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01AE</td>
<td>430</td>
<td>eLON_eCmd_fb_Out_of_range</td>
<td>SNVT 170 / The input variable “stValue.eCmd_fb” is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01B3</td>
<td>435</td>
<td>eLON_byManufacturer_Out_of_range</td>
<td>SNVT 172 / The input variable “stValue.byManufacturer” is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01B8</td>
<td>440</td>
<td>eLON_eDevice_select_Out_of_range</td>
<td>SNVT 175 / The input variable “stValue.eDevice_select” is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01BD</td>
<td>445</td>
<td>eLON_stPos_eFunction_Out_of_range</td>
<td>SNVT 180 / The input variable “stValue.stPos_eFunction” is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01BE</td>
<td>446</td>
<td>eLON_stPos_rSetting_Out_of_range</td>
<td>SNVT 180 / The input variable “stValue.stPos.rSetting” is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01BF</td>
<td>447</td>
<td>eLON_stPos_rRotation_Out_of_range</td>
<td>SNVT 180 / The input variable “stValue.stPos.rRotation” is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01C0</td>
<td>448</td>
<td>eLON_eCmd_source_Out_of_range</td>
<td>SNVT 180 / The input variable “stValue.eCmd_source” is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01C1</td>
<td>449</td>
<td>eLON_eError_code_Out_of_range</td>
<td>SNVT 180 / The input variable “stValue.eError_code” is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01C7</td>
<td>455</td>
<td>eLON_181stAddr_talk_eAudio_sensor_Type_Out_of_range</td>
<td>SNVT 181 / The input variable “stValue.stAddr_talk.eAudio_sensor&quot; type&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01C8</td>
<td>456</td>
<td>eLON_181stAddr_talk_byCar_id_Out_of_range</td>
<td>SNVT 181 / The input variable “stValue.stAddr_talk.byCar_id” is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01C9</td>
<td>457</td>
<td>eLON_181stAddr_talk_byLocation_Out_of_range</td>
<td>SNVT 181 / The input variable “stValue.stAddr_talk.byLocation” is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01CA</td>
<td>458</td>
<td>eLON_181stAddr_talk_byUnit_id_Out_of_range</td>
<td>SNVT 181 / The input variable “stValue.stAddr_talk.byUnit&quot; id&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01CB</td>
<td>459</td>
<td>eLON_181stAddr_init_eAudio_sensor_Type_Out_of_range</td>
<td>SNVT 181 / The input variable “stValue.stAddr_init.eAudio_sensor&quot; type&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01C2</td>
<td>460</td>
<td>eLON_181stAddr_init_byCar_id_Out_of_range</td>
<td>SNVT 181 / The input variable “stValue.stAddr_init.byCar&quot; id&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01CD</td>
<td>461</td>
<td>eLON_181stAddr_init_byLocation_Out_of_range</td>
<td>SNVT 181 / The input variable “stValue.stAddr_init.byLocation&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01CE</td>
<td>462</td>
<td>eLON_181stAddr_init_byUnit_id_Out_of_range</td>
<td>SNVT 181 / The input variable “stValue.stAddr_init.byUnit&quot; id&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01CF</td>
<td>463</td>
<td>eLON_181eAudio_type_Out_of_range</td>
<td>SNVT 181 / The input variable “stValue.eAudio_type&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01D0</td>
<td>464</td>
<td>eLON_181byAudio_line_Out_of_range</td>
<td>SNVT 181 / The input variable “stValue.byAudio_line&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01D1</td>
<td>465</td>
<td>eLON_181stAddr_dest_stP2p_eAudio_sensor_Type_Out_of_range</td>
<td>SNVT 181 / The input variable “stValue.stAddr_dest.stP2p.eAudio_sensor_type&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01D2</td>
<td>466</td>
<td>eLON_181stAddr_dest_stP2p_byCar_id_Out_of_range</td>
<td>SNVT 181 / The input variable “stValue.stAddr_dest.stP2p.byCar&quot; id&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01D3</td>
<td>467</td>
<td>eLON_181stAddr_dest_stP2p_byLocation_Out_of_range</td>
<td>SNVT 181 / The input variable “stValue.stAddr_dest.stP2p.byLocation&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01D4</td>
<td>468</td>
<td>eLON_181stAddr_dest_stP2p_byUnit_id_Out_of_range</td>
<td>SNVT 181 / The input variable “stValue.stAddr_dest.stP2p.byUnit&quot; id&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01DB</td>
<td>475</td>
<td>eLON_stAddr_dest_stP2p_eAudio_sensor_Type_Out_of_range</td>
<td>SNVT 182 / The input variable “stValue.stAddr_dest.stP2p.eAudio_sensor_type&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01DC</td>
<td>476</td>
<td>eLON_stAddr_dest_stP2p_byCar_id_Out_of_range</td>
<td>SNVT 182 / The input variable “stValue.stAddr_dest.stP2p.byCar&quot; id&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01DD</td>
<td>477</td>
<td>eLON_stAddr_dest_stP2p_byLocation_Out_of_range</td>
<td>SNVT 182 / The input variable “stValue.stAddr_dest.stP2p.byLocation&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01DE</td>
<td>478</td>
<td>eLON_stAddr_dest_stP2p_byUnit_id_Out_of_range</td>
<td>SNVT 182 / The input variable “stValue.stAddr_dest.stP2p.byUnit&quot; id&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01DF</td>
<td>479</td>
<td>eLON_stAddr_init_eAudio_sensor_Type_Out_of_range</td>
<td>SNVT 182 / The input variable “stValue.stAddr_init.eAudio_sensor_type&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01E0</td>
<td>480</td>
<td>eLON_stAddr_init_byCar_id_Out_of_range</td>
<td>SNVT 182 / The input variable “stValue.stAddr_init.byCar_id&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>Value (hex)</td>
<td>Value (dec)</td>
<td>Value (enum)</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>0x01E1</td>
<td>481</td>
<td>eLON_stAddr_init_byLocation_Out_of_range</td>
<td>SNVT 182 / The input variable &quot;stValue.stAddr_init.byLocation&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01E2</td>
<td>482</td>
<td>eLON_stAddr_init_byUnit_id_Out_of_range</td>
<td>SNVT 182 / The input variable &quot;stValue.stAddr_init.byUnit_id&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01E3</td>
<td>483</td>
<td>eLON_eAudio_type_Out_of_range</td>
<td>SNVT 182 / The input variable &quot;stValue.eAudio_type&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01EA</td>
<td>490</td>
<td>eLON_eCycle_Out_of_range</td>
<td>SNVT 184 / The input variable &quot;stValue.eCycle&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01EB</td>
<td>491</td>
<td>eLON_eSubcycle_Out_of_range</td>
<td>SNVT 184 / The input variable &quot;stValue.eSubcycle&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01EC</td>
<td>492</td>
<td>eLON_stFunction_eProgram_Out_of_range</td>
<td>SNVT 184 / The input variable &quot;stValue.stFunction.eProgram&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01ED</td>
<td>493</td>
<td>eLON_stFunction_eLoad_level_Out_of_range</td>
<td>SNVT 184 / The input variable &quot;stValue.stFunction.eLoad_level&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01EE</td>
<td>494</td>
<td>eLON_stFunction_stWash_ePrewash_Out_of_range</td>
<td>SNVT 184 / The input variable &quot;stValue.stFunction.stWash.ePrewash&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01EF</td>
<td>495</td>
<td>eLON_stFunction_stRinse_ePreRepeat_Out_of_range</td>
<td>SNVT 184 / The input variable &quot;stValue.stFunction.stRinse.ePreRepeat&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01F0</td>
<td>496</td>
<td>eLON_stFunction_stRinse_byRepeat_Out_of_range</td>
<td>SNVT 184 / The input variable &quot;stValue.stFunction.stRinse.byRepeat&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01F1</td>
<td>497</td>
<td>eLON_stFunction_stSpin_eHold_Out_of_range</td>
<td>SNVT 184 / The input variable &quot;stValue.stFunction.stSpin.eHold&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01F2</td>
<td>498</td>
<td>eLON_stFunction_stDry_byTemp_Out_of_range</td>
<td>SNVT 184 / The input variable &quot;stValue.stFunction.stDry.byTemp&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01F3</td>
<td>499</td>
<td>eLON_stFunction_stDry_stDuration_eDryness_Out_of_range</td>
<td>SNVT 184 / The input variable &quot;stValue.stFunction.stDry.stDuration.eDryness&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01F9</td>
<td>505</td>
<td>eLON_186eCycle_Out_of_range</td>
<td>SNVT 186 / The input variable &quot;stValue.eCycle&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01FA</td>
<td>506</td>
<td>eLON_186eSubcycle_Out_of_range</td>
<td>SNVT 186 / The input variable &quot;stValue.eSubcycle&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01FB</td>
<td>507</td>
<td>eLON_stWasher_command_data_eCycle_Out_of_range</td>
<td>SNVT 186 / The input variable &quot;stValue.stWasher_command_data.eCycle&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01FC</td>
<td>508</td>
<td>eLON_stWasher_command_data_eSubcycle_Out_of_range</td>
<td>SNVT 186 / The input variable &quot;stValue.stWasher_command_data.eSubcycle&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01FD</td>
<td>509</td>
<td>eLON_stWasher_command_data_stFunction_eProgram_Out_of_range</td>
<td>SNVT 186 / The input variable &quot;stValue.stWasher_command_data.stFunction.eProgram&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01FE</td>
<td>510</td>
<td>eLON_stWasher_command_data_stFunction_stWash_eLoad_level_Out_of_range</td>
<td>SNVT 186 / The input variable &quot;stValue.stWasher_command_data.stWash.eLoadLevel&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x01FF</td>
<td>511</td>
<td>eLON_stWasher_command_data_stFunction_stWash_ePrewash_Out_of_range</td>
<td>SNVT 186 / The input variable &quot;stValue.stWasher_command_data.stWash.ePrewash&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0200</td>
<td>512</td>
<td>eLON_stWasher_command_data_stFunction_stRinse_eRepeat_Out_of_range</td>
<td>SNVT 186 / The input variable &quot;stValue.stWasher_command_data.stFunction.stRinse.eRepeat&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0201</td>
<td>513</td>
<td>eLON_stWasher_command_data_stFunction_stRinse_byRepeat_Out_of_range</td>
<td>SNVT 186 / The input variable &quot;stValue.stWasher_command_data.stFunction.stRinse.byRepeat&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0202</td>
<td>514</td>
<td>eLON_stWasher_command_data_stFunction_stSpin_eHold_Out_of_range</td>
<td>SNVT 186 / The input variable &quot;stValue.stWasher_command_data.stFunction.stSpin.eHold&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0203</td>
<td>515</td>
<td>eLON_stWasher_command_data_stFunction_stDry_byTemp_Out_of_range</td>
<td>SNVT 186 / The input variable &quot;stValue.stWasher_command_data.stFunction.stDry.byTemp&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0204</td>
<td>516</td>
<td>eLON_stWasher_command_data_stFunction_stDry_stDuration_eDryness_Out_of_range</td>
<td>SNVT 186 / The input variable &quot;stValue.stWasher_command_data.stFunction.stDry.stDuration.eDryness&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0206</td>
<td>518</td>
<td>eLON_eState_Out_of_range</td>
<td>SNVT 189 / The input variable &quot;stValue.eState&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>0x0207</td>
<td>519</td>
<td>eLON_stSetting_rValue_Out_of_range</td>
<td>SNVT 189 / The input variable &quot;stValue.stSettings.rValue&quot; is outside the permitted range. The value was not sent.</td>
</tr>
<tr>
<td>Hex Value</td>
<td>Decimal Value</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>------------</td>
<td>---------------</td>
<td>------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>0x0208</td>
<td>520</td>
<td>&quot;eLON_stSetting_rChange_Out_of_range&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNVT 189 / The input variable &quot;stValue.stSettings.rChange&quot; is outside the permitted range. The value was not sent.</td>
<td></td>
</tr>
<tr>
<td>0x0209</td>
<td>521</td>
<td>&quot;eLON_stSetting_rMultiplier_Out_of_range&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNVT 189 / The input variable &quot;stValue.stSettings.rMultiplier&quot; is outside the permitted range. The value was not sent.</td>
<td></td>
</tr>
<tr>
<td>0x020A</td>
<td>522</td>
<td>&quot;eLON_stSetting_iAngle_Out_of_range&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNVT 189 / The input variable &quot;stValue.stSettings.iAngle&quot; is outside the permitted range. The value was not sent.</td>
<td></td>
</tr>
<tr>
<td>0x020B</td>
<td>523</td>
<td>&quot;eLON_stSetting_byGroup_number_Out_of_range&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNVT 189 / The input variable &quot;stValue.stSettings.byGroup_number&quot; is outside the permitted range. The value was not sent.</td>
<td></td>
</tr>
<tr>
<td>0x020C</td>
<td>524</td>
<td>&quot;eLON_stSetting_siFan_level_Out_of_range&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNVT 189 / The input variable &quot;stValue.stSettings.siFan_level&quot; is outside the permitted range. The value was not sent.</td>
<td></td>
</tr>
<tr>
<td>0x020D</td>
<td>525</td>
<td>&quot;eLON_stColor_value_stCIE1931_lumen_rX_Out_of_range&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNVT 190 / The input variable &quot;stValue.stColor_value.stCIE1931_lumen.rX&quot; is outside the permitted range. The value was not sent.</td>
<td></td>
</tr>
<tr>
<td>0x020E</td>
<td>526</td>
<td>&quot;eLON_stColor_value_stCIE1931_lumen_ry_Out_of_range&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNVT 190 / The input variable &quot;stValue.stColor_value.stCIE1931_lumen.ry&quot; is outside the permitted range. The value was not sent.</td>
<td></td>
</tr>
<tr>
<td>0x020F</td>
<td>527</td>
<td>&quot;eLON_stColor_value_stCIE1931_lumen_udiAbsolute_Y_Out_of_range&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNVT 190 / The input variable &quot;stValue.stColor_value.stCIE1931_lumen.udiAbsolute.Y&quot; is outside the permitted range. The value was not sent.</td>
<td></td>
</tr>
<tr>
<td>0x0210</td>
<td>528</td>
<td>&quot;eLON_stColor_value_stCIE1931_percent_rX_Out_of_range&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNVT 190 / The input variable &quot;stValue.stColor_value.stCIE1931_percent.rX&quot; is outside the permitted range. The value was not sent.</td>
<td></td>
</tr>
<tr>
<td>0x0211</td>
<td>529</td>
<td>&quot;eLON_stColor_value_stCIE1931_percent_ry_Out_of_range&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNVT 190 / The input variable &quot;stValue.stColor_value.stCIE1931_percent.ry&quot; is outside the permitted range. The value was not sent.</td>
<td></td>
</tr>
<tr>
<td>0x0212</td>
<td>530</td>
<td>&quot;eLON_stColor_value_stCIE1931_percent_rPercent_Y_Out_of_range&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNVT 190 / The input variable &quot;stValue.stColor_value.stCIE1931_percent.rPercent.Y&quot; is outside the permitted range. The value was not sent.</td>
<td></td>
</tr>
<tr>
<td>0x0213</td>
<td>531</td>
<td>&quot;eLON_stColor_value_uiColor_temperature_Out_of_range&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNVT 190 / The input variable &quot;stValue.stColor_value.uiColor_temperature&quot; is outside the permitted range. The value was not sent.</td>
<td></td>
</tr>
<tr>
<td>0x0217</td>
<td>535</td>
<td>&quot;eLON_191eStatus_Out_of_range&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNVT 191 / The input variable &quot;stValue.Status&quot; is outside the permitted range. The value was not sent.</td>
<td></td>
</tr>
<tr>
<td>0x0218</td>
<td>536</td>
<td>&quot;eLON_uLog_number_Out_of_range&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNVT 191 / The input variable &quot;stValue.uLog_number&quot; is outside the permitted range. The value was not sent.</td>
<td></td>
</tr>
<tr>
<td>0x0219</td>
<td>537</td>
<td>&quot;eLON_rLevel_Out_of_range&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNVT 191 / The input variable &quot;stValue.rLevel&quot; is outside the permitted range. The value was not sent.</td>
<td></td>
</tr>
<tr>
<td>0x021A</td>
<td>538</td>
<td>&quot;eLON_stCurrent_notify_time_rHundredths_Out_of_range&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNVT 191 / The input variable &quot;stValue.stCurrent_notify_time.rHundredths&quot; is outside the permitted range. The value was not sent.</td>
<td></td>
</tr>
<tr>
<td>0x021B</td>
<td>539</td>
<td>&quot;eLON_stPrevious_notify_time_rHundredths_Out_of_range&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNVT 191 / The input variable &quot;stValue.stPrevious_notify_time.rHundredths&quot; is outside the permitted range. The value was not sent.</td>
<td></td>
</tr>
<tr>
<td>0x0221</td>
<td>545</td>
<td>&quot;eLON_rHundredths_Out_of_range&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNVT 192 / The input variable &quot;stValue.rHundredths&quot; is outside the permitted range. The value was not sent.</td>
<td></td>
</tr>
<tr>
<td>0x0226</td>
<td>550</td>
<td>&quot;eLON_stStart_time_rHundredths_Out_of_range&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNVT 193 / The input variable &quot;stValue.stStart_time.rHundredths&quot; is outside the permitted range. The value was not sent.</td>
<td></td>
</tr>
<tr>
<td>0x0227</td>
<td>551</td>
<td>&quot;eLON_stEnd_time_rHundredths_Out_of_range&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNVT 193 / The input variable &quot;stValue.stEnd_time.rHundredths&quot; is outside the permitted range. The value was not sent.</td>
<td></td>
</tr>
<tr>
<td>0x0235</td>
<td>565</td>
<td>&quot;eLON_rComplete_Out_of_range&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNVT 194 / The input variable &quot;stValue.rComplete&quot; is outside the permitted range. The value was not sent.</td>
<td></td>
</tr>
<tr>
<td>0x023A</td>
<td>570</td>
<td>&quot;eLON_stTime_actual_rHundredths_Out_of_range&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNVT 199 / The input variable &quot;stValue.stTime_actual.rHundredths&quot; is outside the permitted range. The value was not sent.</td>
<td></td>
</tr>
<tr>
<td>0x023B</td>
<td>571</td>
<td>&quot;eLON_stTime_previous_rHundredths_Out_of_range&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNVT 199 / The input variable &quot;stValue.stTime_previous.rHundredths&quot; is outside the permitted range. The value was not sent.</td>
<td></td>
</tr>
<tr>
<td>0x0249</td>
<td>585</td>
<td>&quot;eLON_lrEnergy_Out_of_range&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNVT 200 / The input variable &quot;stValue.lrEnergy&quot; is outside the permitted range. The value was not sent.</td>
<td></td>
</tr>
<tr>
<td>0x024A</td>
<td>586</td>
<td>&quot;eLON_rPowerFactor_Out_of_range&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNVT 200 / The input variable &quot;stValue.rPowerFactor&quot; is outside the permitted range. The value was not sent.</td>
<td></td>
</tr>
<tr>
<td>0x024B</td>
<td>587</td>
<td>&quot;eLON_rPower_Out_of_range&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNVT 200 / The input variable &quot;stValue.rPower&quot; is outside the permitted range. The value was not sent.</td>
<td></td>
</tr>
<tr>
<td>0x024C</td>
<td>588</td>
<td>&quot;eLON_rBallastTemp_Out_of_range&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNVT 200 / The input variable &quot;stValue.rBallastTemp&quot; is outside the permitted range. The value was not sent.</td>
<td></td>
</tr>
<tr>
<td>0x0253</td>
<td>595</td>
<td>&quot;eLON_lrLongitude_Out_of_range&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNVT 201 / The input variable &quot;stValue.lrLongitude&quot; is outside the permitted range. The value was not sent.</td>
<td></td>
</tr>
<tr>
<td>0x0254</td>
<td>596</td>
<td>&quot;eLON_lrLatitude_Out_of_range&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNVT 201 / The input variable &quot;stValue.lrLatitude&quot; is outside the permitted range. The value was not sent.</td>
<td></td>
</tr>
</tbody>
</table>
### 4.2 DUTs

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>E_LON_alarm_type_t</code> [503]</td>
<td>Used by: SNVT_alarm / SNVT_alarm_2</td>
</tr>
<tr>
<td><code>E_LON_appl_cwc_t</code> [505]</td>
<td>Used by: SNVT_clothes_w_c / SNVT_clothes_w_s</td>
</tr>
<tr>
<td><code>E_LON_appl_cwp_t</code> [505]</td>
<td>Used by: SNVT_clothes_w_c</td>
</tr>
<tr>
<td><code>E_LON_appl_cws_t</code> [506]</td>
<td>Used by: SNVT_clothes_w_c / SNVT_clothes_w_s</td>
</tr>
<tr>
<td><code>E_LON_appl_rin_t</code> [507]</td>
<td>Used by: SNVT_clothes_w_c</td>
</tr>
<tr>
<td><code>E_LON_boolean_t</code> [507]</td>
<td>Used by: SCPTautoAnswer / SCPTcoolingResetEnable / SCPTdefrostHold / SCPTdefrostInternalSchedule / SCPTheatingResetEnable / SCPThighLimit1Enable / SCPThighLimit2Enable / SCPTlowLimit1Enable / SCPTlowLimit2Enable / SCPTscheduleInternal / SNVT_clothes_w_c / SNVT_pump_sensor / SNVT_pumpset_mn / SNVT_pumpset_sn</td>
</tr>
<tr>
<td><code>E_LON_calendar_type_t</code> [507]</td>
<td>Used by: SNVT_time_zone</td>
</tr>
<tr>
<td><code>E_LON_cam_act_t</code> [508]</td>
<td>Used by: SNVT_pos_ctrl</td>
</tr>
<tr>
<td><code>E_LON_cam_func_t</code> [508]</td>
<td>Used by: SNVT_pos_ctrl</td>
</tr>
<tr>
<td><code>E_LON_chiller_t</code> [509]</td>
<td>Used by: SNVT_chlr_status</td>
</tr>
<tr>
<td><code>E_LON_color_encoding_t</code> [509]</td>
<td>Used by: SNVT_color_2</td>
</tr>
<tr>
<td><code>E_LON_config_source_t</code> [509]</td>
<td>Used by: SNVT_config_src</td>
</tr>
<tr>
<td><code>E_LON_controlresp_t</code> [510]</td>
<td>Used by: SNVT_ctrl_resp</td>
</tr>
<tr>
<td><code>E_LON_currency_t</code> [510]</td>
<td>Used by: SNVT_currency</td>
</tr>
<tr>
<td><code>E_LON_days_of_week_t</code> [513]</td>
<td>Used by: SCPTtimePeriod / SNVT_date_day / SNVT_time_zone</td>
</tr>
<tr>
<td><code>E_LON_defrost_mode_t</code> [513]</td>
<td>Used by: SNVT_defr_mode</td>
</tr>
<tr>
<td><code>E_LON_defrost_state_t</code> [514]</td>
<td>Used by: SNVT_defr_state</td>
</tr>
<tr>
<td><code>E_LON_defrost_term_t</code> [514]</td>
<td>Used by: SNVT_defr_term</td>
</tr>
<tr>
<td><code>E_LON_device_c_mode_t</code> [515]</td>
<td>Used by: SNVT_dev_c_mode</td>
</tr>
<tr>
<td><code>E_LON_device_select_t</code> [516]</td>
<td>Used by: SNVT_dev_fault / SNVT_dev_maint / SNVT_dev_status</td>
</tr>
<tr>
<td><code>E_LON_discrete_levels_t</code> [516]</td>
<td>Used by: SNVT_clothes_w_c / SNVT_lev_disc</td>
</tr>
<tr>
<td><code>E_LON_emerg_t</code> [517]</td>
<td>Used by: SNVT_hvac_emerg</td>
</tr>
<tr>
<td><code>E_LON_entcmd_t</code> [517]</td>
<td>Used by: SNVT_ent_state</td>
</tr>
<tr>
<td><code>E_LON_ent_opmode_cmd_t</code> [518]</td>
<td>Used by: SNVT_ent_opmode / SNVT_ent_status</td>
</tr>
<tr>
<td><code>E_LON_evap_t</code> [519]</td>
<td>Used by: SNVT_evap_state</td>
</tr>
<tr>
<td><code>E_LON_ex_control_t</code> [520]</td>
<td>Used by: SNVT_ex_control</td>
</tr>
<tr>
<td><code>E_LON_file_request_t</code> [520]</td>
<td>Used by: SNVT_file_req</td>
</tr>
<tr>
<td><code>E_LON_filestatus_t</code> [521]</td>
<td>Used by: SNVT_file_status</td>
</tr>
<tr>
<td><code>E_LON_fire_indicator_t</code> [521]</td>
<td>Used by: SNVT_fire_indcte</td>
</tr>
<tr>
<td><code>E_LON_fire_initiator_t</code> [522]</td>
<td>Used by: SNVT_fire_init</td>
</tr>
<tr>
<td><code>E_LON_fire_test_t</code> [523]</td>
<td>Used by: SNVT_fire_test</td>
</tr>
<tr>
<td><code>E_LON_flow_direction_t</code> [523]</td>
<td>Used by: SNVT_flow_dir</td>
</tr>
<tr>
<td><code>E_LON_gfci_status_t</code> [523]</td>
<td>Used by: SNVT_gfci_status</td>
</tr>
<tr>
<td><code>E_LON_hvac_hvt_t</code> [524]</td>
<td>Used by: SNVT_hvac_type</td>
</tr>
</tbody>
</table>
### Data types

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>E_LON_hvac_overid_t</td>
<td>Used by: SNVT_hvac_overid</td>
</tr>
<tr>
<td>E_LON_hvac_t</td>
<td>Used by: SNVT_chlr_status / SNVT_hvac_mode / SNVT_hvac_status</td>
</tr>
<tr>
<td>E_LON_learn_mode_t</td>
<td>Used by: SNVT_preset</td>
</tr>
<tr>
<td>E_LON_log_status_t</td>
<td>Used by: SCPTLogRecord / SNVT_log_status</td>
</tr>
<tr>
<td>E_LON_motor_state_t</td>
<td>Used by: SNVT_motor_state / SNVT_pumpset_mn</td>
</tr>
<tr>
<td>E_LON_nv_type_category_t</td>
<td>Used by: SNVT_nv_type</td>
</tr>
<tr>
<td>E_LON_object_request_t</td>
<td>Used by: SNVT_obj_request</td>
</tr>
<tr>
<td>E_LON_occup_t</td>
<td>Used by: SNVT_occupancy / SNVT_tod_event</td>
</tr>
<tr>
<td>E_LON_override_t</td>
<td>Used by: SNVT_override</td>
</tr>
<tr>
<td>E_LON_pan_dir_t</td>
<td>Used by: SNVT_ptz</td>
</tr>
<tr>
<td>E_LON_priority_level_t</td>
<td>Used by: SNVT_alarm / SNVT_alarm_2 / SNVT_pumpset_mn</td>
</tr>
<tr>
<td>E_LON_privacyzone_t</td>
<td>Used by: SNVT_privacyzone</td>
</tr>
<tr>
<td>E_LON_rail_audio_sensor_type_t</td>
<td>Used by: SNVT_rac_ctrl / SNVT_rac_req</td>
</tr>
<tr>
<td>E_LON_rail_audio_type_t</td>
<td>Used by: SNVT_rac_ctrl / SNVT_rac_req</td>
</tr>
<tr>
<td>E_LON_reg_val_unit_t</td>
<td>Used by: SNVT_reg_val / SNVT_reg_valTs</td>
</tr>
<tr>
<td>E_LON_sblnd_cmd_source_t</td>
<td>Used by: SNVT_sblnd_state</td>
</tr>
<tr>
<td>E_LON_sblnd_error_t</td>
<td>Used by: SNVT_sblnd_state</td>
</tr>
<tr>
<td>E_LON_scene_config_t</td>
<td>Used by: SNVT_scene_cfg</td>
</tr>
<tr>
<td>E_LON_scene_t</td>
<td>Used by: SNVT_scene</td>
</tr>
<tr>
<td>E_LON_sec_state_t</td>
<td>Used by: SNVT_sec_state</td>
</tr>
<tr>
<td>E_LON_sec_status_t</td>
<td>Used by: SNVT_sec_status</td>
</tr>
<tr>
<td>E_LON_setting_t</td>
<td>Used by: SNVT_setting</td>
</tr>
<tr>
<td>E_LON_switch_state_t</td>
<td>Used by: SNVT_switch_2</td>
</tr>
<tr>
<td>E_LON_telcom_states_t</td>
<td>Used by: SNVT_telcom</td>
</tr>
<tr>
<td>E_LON_therm_mode_t</td>
<td>Used by: SNVT_therm_mode</td>
</tr>
<tr>
<td>E_LON_tilt_dir_t</td>
<td>Used by: SNVT_ptz</td>
</tr>
<tr>
<td>E_LON_unit_temp_t</td>
<td>Used by: SNVT_pump_sensor</td>
</tr>
<tr>
<td>E_LON_valve_mode_t</td>
<td>Used by: SNVT_valve_mode</td>
</tr>
<tr>
<td>E_LON_zoom_t</td>
<td>Used by: SNVT_ptz</td>
</tr>
</tbody>
</table>

### Hardware Types

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_Parameter_IN_36B</td>
<td>Process Image of the inputs</td>
</tr>
<tr>
<td>ST_LON_Parameter_OUT_36B</td>
<td>Process Image of the outputs</td>
</tr>
</tbody>
</table>

### LON_TYPES

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>E_LON_Parameter_Datatypes</td>
<td>Enums SNVT types</td>
</tr>
</tbody>
</table>

### Structure/AuxiliaryStructure/SNVT_chlr_status

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_chlr_state</td>
<td>Used by: SNVT_chlr_status</td>
</tr>
</tbody>
</table>
### Data types

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_action</td>
<td>Used by: SNVT_clothes_w_c</td>
</tr>
<tr>
<td>ST_LON_dry</td>
<td>Used by: SNVT_clothes_w_c</td>
</tr>
<tr>
<td>ST_LON_duration</td>
<td>Used by: SNVT_clothes_w_c</td>
</tr>
<tr>
<td>ST_LON_function</td>
<td>Used by: SNVT_clothes_w_c</td>
</tr>
<tr>
<td>ST_LON_rinse</td>
<td>Used by: SNVT_clothes_w_c</td>
</tr>
<tr>
<td>ST_LON_spin</td>
<td>Used by: SNVT_clothes_w_c</td>
</tr>
<tr>
<td>ST_LON_wash</td>
<td>Used by: SNVT_clothes_w_c</td>
</tr>
</tbody>
</table>

### Data types

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_alarm</td>
<td>Used by: SNVT_clothes_w_s</td>
</tr>
</tbody>
</table>

### Data types

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_CIE1931_lumen</td>
<td>Used by: SNVT_color_2</td>
</tr>
<tr>
<td>ST_LON_CIE1931_percent</td>
<td>Used by: SNVT_color_2</td>
</tr>
<tr>
<td>ST_LON_color_value</td>
<td>Used by: SNVT_color_2</td>
</tr>
<tr>
<td>ST_LON_RGB</td>
<td>Used by: SNVT_color_2</td>
</tr>
</tbody>
</table>

### Data types

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_range</td>
<td>Used by: SNVT_ctrl_resp</td>
</tr>
<tr>
<td>ST_LON_sender</td>
<td>Used by: SNVT_ctrl_resp</td>
</tr>
</tbody>
</table>

### Data types

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_Dev_type1</td>
<td>Used by: SNVT_dev_fault</td>
</tr>
<tr>
<td>ST_LON_pump_ctrl1</td>
<td>Used by: SNVT_dev_fault</td>
</tr>
<tr>
<td>ST_LON_valve_pos1</td>
<td>Used by: SNVT_dev_fault</td>
</tr>
</tbody>
</table>

### Data types

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_Dev_type2</td>
<td>Used by: SNVT_dev_maint</td>
</tr>
<tr>
<td>ST_LON_pump_ctrl2</td>
<td>Used by: SNVT_dev_maint</td>
</tr>
<tr>
<td>ST_LON_valve_pos2</td>
<td>Used by: SNVT_dev_maint</td>
</tr>
</tbody>
</table>

### Data types

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_Dev_type3</td>
<td>Used by: SNVT_dev_status</td>
</tr>
<tr>
<td>ST_LON_pump_ctrl3</td>
<td>Used by: SNVT_dev_status</td>
</tr>
<tr>
<td>ST_LON_valve_pos3</td>
<td>Used by: SNVT_dev_status</td>
</tr>
</tbody>
</table>
### Structure/AuxiliaryStructure/SNVT_ex_control

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_Control_device_addr [572]</td>
<td>Used by: SNVT_ex_control</td>
</tr>
</tbody>
</table>

### Structure/AuxiliaryStructure/SNVT_file_req

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_addrt [572]</td>
<td>Used by: SNVT_file_req</td>
</tr>
<tr>
<td>ST_LON_dest_address [573]</td>
<td>Used by: SNVT_file_req</td>
</tr>
<tr>
<td>ST_LON_qp [573]</td>
<td>Used by: SNVT_file_req</td>
</tr>
<tr>
<td>ST_LON_sn [573]</td>
<td>Used by: SNVT_file_req</td>
</tr>
</tbody>
</table>

### Structure/AuxiliaryStructure/SNVT_file_status

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_address [574]</td>
<td>Used by: FB_Write_Address_Table / FB_Read_Address_Table</td>
</tr>
<tr>
<td>ST_LON_adr [574]</td>
<td>Used by: SNVT_file_status</td>
</tr>
<tr>
<td>ST_LON_descriptor [575]</td>
<td>Used by: SNVT_file_status</td>
</tr>
</tbody>
</table>

### Structure/AuxiliaryStructure/SNVT_lamp_status

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_Alarm_actual [575]</td>
<td>Used by: SNVT_lamp_status</td>
</tr>
<tr>
<td>ST_LON_alarm_previous [577]</td>
<td>Used by: SNVT_lamp_status</td>
</tr>
</tbody>
</table>

### Structure/AuxiliaryStructure/SNVT_pos_ctrl

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_abspos [579]</td>
<td>Used by: SNVT_pos_ctrl</td>
</tr>
<tr>
<td>ST_LON_Value [579]</td>
<td>Used by: SNVT_pos_ctrl</td>
</tr>
</tbody>
</table>

### Structure/AuxiliaryStructure/SNVT_rac_ctrl

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_addr_dest [580]</td>
<td>Used by: SNVT_rac_req / SNVT_rac_ctrl</td>
</tr>
<tr>
<td>ST_LON_addr_init [580]</td>
<td>Used by: SNVT_rac_req / SNVT_rac_ctrl</td>
</tr>
<tr>
<td>ST_LON_addr_talk [581]</td>
<td>Used by: SNVT_rac_ctrl</td>
</tr>
<tr>
<td>ST_LON_p2m [581]</td>
<td>Used by: SNVT_rac_req / SNVT_rac_ctrl</td>
</tr>
<tr>
<td>ST_LON_p2p [582]</td>
<td>Used by: SNVT_rac_req / SNVT_rac_ctrl</td>
</tr>
</tbody>
</table>

### Structure/AuxiliaryStructure/SNVT_rac_req

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_rac_req_addr_dest [582]</td>
<td></td>
</tr>
<tr>
<td>ST_LON_rac_req_addr_init [582]</td>
<td></td>
</tr>
</tbody>
</table>

### Structure/AuxiliaryStructure/SNVT_switch_2

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_setting [583]</td>
<td>Used by: SNVT_switch_2</td>
</tr>
</tbody>
</table>
Structure/AuxiliaryStructure/SNVT_time_zone

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_end_DST [584]</td>
<td>Used by: SNVT_time_zone</td>
</tr>
<tr>
<td>ST_LON_M_end_DST [584]</td>
<td>Used by: SNVT_time_zone</td>
</tr>
<tr>
<td>ST_LON_M_start_DST [585]</td>
<td>Used by: SNVT_time_zone</td>
</tr>
<tr>
<td>ST_LON_start_DST [585]</td>
<td>Used by: SNVT_time_zone</td>
</tr>
</tbody>
</table>

Structure

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_KL6401 [589]</td>
<td>Structure for the configuration</td>
</tr>
<tr>
<td>ST_LON_AddressTable [589]</td>
<td>Used by: FB_Write_Address_Table / FB_Read_Address_Table</td>
</tr>
<tr>
<td>ST_LON_ConfigTable [590]</td>
<td>Used by: FB_Write_Config_Table / FB_Read_Config_Table</td>
</tr>
<tr>
<td>ST_LON_DomainTable [591]</td>
<td>Used by: FB_Write_Domain_Table / FB_Read_Domain_Table</td>
</tr>
<tr>
<td>ST_LON_SNVT_alarm [591]</td>
<td>Used by: SNVT_alarm</td>
</tr>
<tr>
<td>ST_LON_SNVT_alarm_2 [592]</td>
<td>Used by: SNVT_alarm_2</td>
</tr>
<tr>
<td>ST_LON_SNVT_chlr_status [593]</td>
<td>Used by: SNVT_chlr_status</td>
</tr>
<tr>
<td>ST_LON_SNVT_clothes_w_a [593]</td>
<td>Used by: SNVT_clothes_w_a</td>
</tr>
<tr>
<td>ST_LON_SNVT_clothes_w_c [595]</td>
<td>Used by: SNVT_clothes_w_c</td>
</tr>
<tr>
<td>ST_LON_SNVT_clothes_w_m [595]</td>
<td>Used by: SNVT_clothes_w_m</td>
</tr>
<tr>
<td>ST_LON_SNVT_clothes_w_s [595]</td>
<td>Used by: SNVT_clothes_w_s</td>
</tr>
<tr>
<td>ST_LON_SNVT_color [596]</td>
<td>Used by: SNVT_color</td>
</tr>
<tr>
<td>ST_LON_SNVT_color_2 [596]</td>
<td>Used by: SNVT_color_2</td>
</tr>
<tr>
<td>ST_LON_SNVT_ctrl_req [596]</td>
<td>Used by: SNVT_ctrl_req</td>
</tr>
<tr>
<td>ST_LON_SNVT_ctrl_resp [597]</td>
<td>Used by: SNVT_ctrl_resp</td>
</tr>
<tr>
<td>ST_LON_SNVT_currency [597]</td>
<td>Used by: SNVT_currency</td>
</tr>
<tr>
<td>ST_LON_SNVT_date_event [598]</td>
<td>Used by: SNVT_date_event</td>
</tr>
<tr>
<td>ST_LON_SNVT_dev_fault [598]</td>
<td>Used by: SNVT_dev_fault</td>
</tr>
<tr>
<td>ST_LON_SNVT_dev_maint [598]</td>
<td>Used by: SNVT_dev_maint</td>
</tr>
<tr>
<td>ST_LON_SNVT_dev_status [599]</td>
<td>Used by: SNVT_dev_status</td>
</tr>
<tr>
<td>ST_LON_SNVT_earth_pos [599]</td>
<td>Used by: SNVT_earth_pos</td>
</tr>
<tr>
<td>ST_LON_SNVT_elapsed_tm [599]</td>
<td>Used by: SNVT_elapsed_tm</td>
</tr>
<tr>
<td>ST_LON_SNVT_ent_status [600]</td>
<td>Used by: SNVT_ent_status</td>
</tr>
<tr>
<td>ST_LON_SNVT_environment [601]</td>
<td>Used by: SNVT_environment</td>
</tr>
<tr>
<td>ST_LON_SNVT_ex_control [602]</td>
<td>Used by: SNVT_ex_control</td>
</tr>
<tr>
<td>ST_LON_SNVT_file_pos [602]</td>
<td>Used by: SNVT_file_pos</td>
</tr>
<tr>
<td>ST_LON_SNVT_file_req [603]</td>
<td>Used by: SNVT_file_req</td>
</tr>
<tr>
<td>ST_LON_SNVT_file_status [603]</td>
<td>Used by: SNVT_file_status</td>
</tr>
<tr>
<td>ST_LON_SNVT_geo_loc [604]</td>
<td>Used by: SNVT_geo_loc</td>
</tr>
<tr>
<td>ST_LON_SNVT_hvac_overid [604]</td>
<td>Used by: SNVT_hvac_overid</td>
</tr>
<tr>
<td>ST_LON_SNVT_hvac_satsts [604]</td>
<td>Used by: SNVT_hvac_satsts</td>
</tr>
<tr>
<td>ST_LON_SNVT_hvac_status [605]</td>
<td>Used by: SNVT_hvac_status</td>
</tr>
<tr>
<td>ST_LON_SNVT_lamp_status [606]</td>
<td>Used by: SNVT_lamp_status</td>
</tr>
</tbody>
</table>
### Data types

<table>
<thead>
<tr>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_SNVT_log_fx_request</td>
<td>Used by: SNVT_log_fx_request</td>
</tr>
<tr>
<td>ST_LON_SNVT_log_fx_status</td>
<td>Used by: SNVT_log_fx_status</td>
</tr>
<tr>
<td>ST_LON_SNVT_log_status</td>
<td>Used by: SNVT_log_status</td>
</tr>
<tr>
<td>ST_LON_SNVT_muldiv</td>
<td>Used by: SNVT_muldiv</td>
</tr>
<tr>
<td>ST_LON_SNVT_nv_type</td>
<td>Used by: SNVT_nv_type</td>
</tr>
<tr>
<td>ST_LON_SNVT_obj_request</td>
<td>Used by: SNVT_obj_request</td>
</tr>
<tr>
<td>ST_LON_SNVT_obj_status</td>
<td>Used by: SNVT_obj_status</td>
</tr>
<tr>
<td>ST_LON_SNVT_pos_ctrl</td>
<td>Used by: SNVT_pos_ctrl</td>
</tr>
<tr>
<td>ST_LON_SNVT_preset</td>
<td>Used by: SNVT_preset</td>
</tr>
<tr>
<td>ST_LON_SNVT_privacyzone</td>
<td>Used by: SNVT_privacyzone</td>
</tr>
<tr>
<td>ST_LON_SNVT_ptz</td>
<td>Used by: SNVT_ptz</td>
</tr>
<tr>
<td>ST_LON_SNVT_pump_sensor</td>
<td>Used by: SNVT_pump_sensor</td>
</tr>
<tr>
<td>ST_LON_SNVT_pumpset_mn</td>
<td>Used by: SNVT_pumpset_mn</td>
</tr>
<tr>
<td>ST_LON_SNVT_pumpset_sn</td>
<td>Used by: SNVT_pumpset_sn</td>
</tr>
<tr>
<td>ST_LON_SNVT_rac_ctrl</td>
<td>Used by: SNVT_rac_ctrl</td>
</tr>
<tr>
<td>ST_LON_SNVT_rac_req</td>
<td>Used by: SNVT_rac_req</td>
</tr>
<tr>
<td>ST_LON_SNVT_req_val</td>
<td>Used by: SNVT_rac_val</td>
</tr>
<tr>
<td>ST_LON_SNVT_req_val_ts</td>
<td>Used by: SNVT_rac_val_ts</td>
</tr>
<tr>
<td>ST_LON_SNVT_sblnd_state</td>
<td>Used by: SNVT_sblnd_state</td>
</tr>
<tr>
<td>ST_LON_SNVT_scene</td>
<td>Used by: SNVT_scene</td>
</tr>
<tr>
<td>ST_LON_SNVT_scene_cfg</td>
<td>Used by: SNVT_scene_cfg</td>
</tr>
<tr>
<td>ST_LON_SNVT_setting</td>
<td>Used by: SNVT_setting</td>
</tr>
<tr>
<td>ST_LON_SNVT_str_int</td>
<td>Used by: SNVT_str_int</td>
</tr>
<tr>
<td>ST_LON_SNVT_switch</td>
<td>Used by: SNVT_switch</td>
</tr>
<tr>
<td>ST_LON_SNVT_switch_2</td>
<td>Used by: SNVT_switch_2</td>
</tr>
<tr>
<td>ST_LON_SNVT_temp_setpt</td>
<td>Used by: SNVT_temp_setpt</td>
</tr>
<tr>
<td>ST_LON_SNVT_time_zone</td>
<td>Used by: SNVT_time_zone</td>
</tr>
<tr>
<td>ST_LON_SNVT_tod_event</td>
<td>Used by: SNVT_tod_event</td>
</tr>
<tr>
<td>ST_LON_SNVT_trans_table</td>
<td>Used by: SNVT_trans_table</td>
</tr>
<tr>
<td>ST_LON_SNVT_zerospan</td>
<td>Used by: SNVT_zerospan</td>
</tr>
<tr>
<td>str_AddressTable</td>
<td>Address table</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>E_LON_ERROR</td>
<td>Error messages</td>
</tr>
<tr>
<td>ST_ExplicitMessage</td>
<td>Explicit Message</td>
</tr>
<tr>
<td>ST_LON_Communication</td>
<td>Connection between &quot;FB_LON_KL6401&quot; and the send/receive function blocks</td>
</tr>
<tr>
<td>ST_LON_ParameterInterface</td>
<td>LON parameter interface</td>
</tr>
<tr>
<td>ST_LON_WriteData</td>
<td>Structure of the send buffer</td>
</tr>
<tr>
<td>ST_Prm</td>
<td>Structure for configuration</td>
</tr>
</tbody>
</table>
### Enums

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>E_LON_alarm_type_t [503]</td>
<td>Used by: SNVT_alarm / SNVT_alarm_2</td>
</tr>
<tr>
<td>E_LON_appl_cwc_t [505]</td>
<td>Used by: SNVT_clothes_w_c / SNVT_clothes_w_s</td>
</tr>
<tr>
<td>E_LON_appl_cwp_t [505]</td>
<td>Used by: SNVT_clothes_w_c</td>
</tr>
<tr>
<td>E_LON_appl_cws_t [506]</td>
<td>Used by: SNVT_clothes_w_c / SNVT_clothes_w_s</td>
</tr>
<tr>
<td>E_LON_appl_rin_t [507]</td>
<td>Used by: SNVT_clothes_w_c</td>
</tr>
<tr>
<td>E_LON_boolean_t [507]</td>
<td>Used by: SCPTautoAnswer / SCPTcoolingResetEnable / SCPTdefrostHold / SCPTdefrostInternalSchedule / SCPTThreadingResetEnable / SCPTThighLimit1Enable / SCPTThighLimit2Enable / SCPTTlowLimit1Enable / SCPTTlowLimit2Enable / SCPTScheduleInternal / SNVT_clothes_w_c / SNVT_pump_sensor / SNVT_pumpset_mn / SNVT_pumpset_sn</td>
</tr>
<tr>
<td>E_LON_calendar_type_t [507]</td>
<td>Used by: SNVT_time_zone</td>
</tr>
<tr>
<td>E_LON_cam_act_t [508]</td>
<td>Used by: SNVT_pos_ctrl</td>
</tr>
<tr>
<td>E_LON_cam_func_t [508]</td>
<td>Used by: SNVT_pos_ctrl</td>
</tr>
<tr>
<td>E_LON_chiller_t [509]</td>
<td>Used by: SNVT_chlr_status</td>
</tr>
<tr>
<td>E_LON_color_encoding_t [509]</td>
<td>Used by: SNVT_color_2</td>
</tr>
<tr>
<td>E_LON_config_source_t [509]</td>
<td>Used by: SNVT_config_src</td>
</tr>
<tr>
<td>E_LON_control_resp_t [510]</td>
<td>Used by: SNVT_ctrl_resp</td>
</tr>
<tr>
<td>E_LON_currency_t [510]</td>
<td>Used by: SNVT_currency</td>
</tr>
<tr>
<td>E_LON_days_of_week_t [513]</td>
<td>Used by: SCPTtimePeriod / SNVT_date_day / SNVT_time_zone</td>
</tr>
<tr>
<td>E_LON_defrost_mode_t [513]</td>
<td>Used by: SNVT_defr_mode</td>
</tr>
<tr>
<td>E_LON_defrost_state_t [514]</td>
<td>Used by: SNVT_defr_state</td>
</tr>
<tr>
<td>E_LON_defrost_term_t [514]</td>
<td>Used by: SNVT_defr_term</td>
</tr>
<tr>
<td>E_LON_device_c_mode_t [515]</td>
<td>Used by: SNVT_dev_c_mode</td>
</tr>
<tr>
<td>E_LON_device_select_t [516]</td>
<td>Used by: SNVT_dev_fault / SNVT_dev_maint / SNVT_dev_status</td>
</tr>
<tr>
<td>E_LON_discrete_levels_t [516]</td>
<td>Used by: SNVT_clothes_w_c / SNVT_lev_disc</td>
</tr>
<tr>
<td>E_LON_emerg_t [517]</td>
<td>Used by: SNVT_hvac_emerg</td>
</tr>
<tr>
<td>E_LON_ent_cmd_t [517]</td>
<td>Used by: SNVT_ent_state</td>
</tr>
<tr>
<td>E_LON_ent_opmode_cmd_t [518]</td>
<td>Used by: SNVT_ent_opmode / SNVT_ent_status</td>
</tr>
<tr>
<td>E_LON_evap_t [519]</td>
<td>Used by: SNVT_evap_state</td>
</tr>
<tr>
<td>E_LON_ex_control_t [520]</td>
<td>Used by: SNVT_ex_control</td>
</tr>
<tr>
<td>E_LON_file_request_t [520]</td>
<td>Used by: SNVT_file_req</td>
</tr>
<tr>
<td>E_LON_file_status_t [521]</td>
<td>Used by: SNVT_file_status</td>
</tr>
<tr>
<td>E_LON_fire_indicator_t [521]</td>
<td>Used by: SNVT_fire_indcte</td>
</tr>
<tr>
<td>E_LON_fire_initiator_t [522]</td>
<td>Used by: SNVT_fire_init</td>
</tr>
<tr>
<td>E_LON_fire_test_t [523]</td>
<td>Used by: SNVT_fire_test</td>
</tr>
<tr>
<td>E_LON_flow_direction_t [523]</td>
<td>Used by: SNVT_flow_dir</td>
</tr>
<tr>
<td>E_LON_gfci_status_t [524]</td>
<td>Used by: SNVT_gfci_status</td>
</tr>
<tr>
<td>E_LON_hvac_hvt_t [524]</td>
<td>Used by: SNVT_hvac_type</td>
</tr>
<tr>
<td>E_LON_hvac_overid_t [525]</td>
<td>Used by: SNVT_hvac_overid</td>
</tr>
</tbody>
</table>
### Data types

<table>
<thead>
<tr>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>E_LON_hvac_t</td>
<td>Used by: SNVT_chlr_status / SNVT_hvac_mode / SNVT_hvac_status</td>
</tr>
<tr>
<td>E_LON_learn_mode_t</td>
<td>Used by: SNVT_preset</td>
</tr>
<tr>
<td>E_LON_log_status_t</td>
<td>Used by: SCPTLogRecord / SNVT_log_status</td>
</tr>
<tr>
<td>E_LON_motor_state_t</td>
<td>Used by: SNVT_motor_state / SNVT_pumpset_mn</td>
</tr>
<tr>
<td>E_LON_nv_type_category_t</td>
<td>Used by: SNVT_nv_type</td>
</tr>
<tr>
<td>E_LON_object_request_t</td>
<td>Used by: SNVT_obj_request</td>
</tr>
<tr>
<td>E_LON_occup_t</td>
<td>Used by: SNVT_occupancy / SNVT_tod_event</td>
</tr>
<tr>
<td>E_LON_pan_dir_t</td>
<td>Used by: SNVT_ptz</td>
</tr>
<tr>
<td>E_LON_priority_level_t</td>
<td>Used by: SNVT_alarm / SNVT_alarm_2 / SNVT_pumpset_mn</td>
</tr>
<tr>
<td>E_LON_privacyzone_t</td>
<td>Used by: SNVT_privacyzone</td>
</tr>
<tr>
<td>E_LON_rail_audio_sensor_type_t</td>
<td>Used by: SNVT_rac_ctrl / SNVT_rac_req</td>
</tr>
<tr>
<td>E_LON_rail_audio_type_t</td>
<td>Used by: SNVT_rac_ctrl / SNVT_rac_req</td>
</tr>
<tr>
<td>E_LON_reg_val_unit_t</td>
<td>Used by: SNVT_reg_val / SNVT_reg_val_ts</td>
</tr>
<tr>
<td>E_LON_sblnd_cmd_source_t</td>
<td>Used by: SNVT_sblnd_state</td>
</tr>
<tr>
<td>E_LON_sblnd_error_t</td>
<td>Used by: SNVT_sblnd_state</td>
</tr>
<tr>
<td>E_LON_scene_config_t</td>
<td>Used by: SNVT_scene_cfg</td>
</tr>
<tr>
<td>E_LON_scene_t</td>
<td>Used by: SNVT_scene</td>
</tr>
<tr>
<td>E_LON_sec_state_t</td>
<td>Used by: SNVT_sec_state</td>
</tr>
<tr>
<td>E_LON_sec_status_t</td>
<td>Used by: SNVT_sec_status</td>
</tr>
<tr>
<td>E_LON_setting_t</td>
<td>Used by: SNVT_setting</td>
</tr>
<tr>
<td>E_LON_switch_state_t</td>
<td>Used by: SNVT_switch_2</td>
</tr>
<tr>
<td>E_LON_telcom_states_t</td>
<td>Used by: SNVT_telcom</td>
</tr>
<tr>
<td>E_LON_therm_mode_t</td>
<td>Used by: SNVT_therm_mode</td>
</tr>
<tr>
<td>E_LON_tilt_dir_t</td>
<td>Used by: SNVT_ptz</td>
</tr>
<tr>
<td>E_LON_unit_temp_t</td>
<td>Used by: SNVT_pump_sensor</td>
</tr>
<tr>
<td>E_LON_valve_mode_t</td>
<td>Used by: SNVT_valve_mode</td>
</tr>
<tr>
<td>E_LON_zoom_t</td>
<td>Used by: SNVT_ptz</td>
</tr>
</tbody>
</table>

### 4.2.1.1 E_LON_ERROR

#### Library error messages

The NV index in the PLC is not compared with the NV index (column Id) in the KS2000 during sending. Wrong/invalid values can be sent if the indices do not match.

Sending without binding does not result in an error message.

```plaintext
TYPE E_LON_ERROR :
{
  eLON_no_Error := 0,
  eLON_Value_out_of_range := 1,
  eLON_Terminal_not_ready := 2,
  eLON_Wrong_SNVT_Typ := 3,
  eLON_Wrong_wNVIndex := 4,
  eKL6401_Wrong_Terminal := 5,
  eKL6401_Error := 6,
  eKL6401_Terminal_is_not_initialized := 7,
}```
eLON_L_star_Out_of_range := 50,
eLON_A_star_Out_of_range := 51,
eLON_B_star_Out_of_range := 52,
eLON_eRequest_Out_of_range := 55,
eLON_wYear_Out_of_range := 66,
eLON_wMonth_Out_of_range := 67,
eLON_wDay_Out_of_range := 68,
eLON_wHour_Out_of_range := 69,
eLON_wMinute_Out_of_range := 70,
eLON_wSecond_Out_of_range := 71,
eLON_wMillisecond_Out_of_range := 72,
eLON_rZero_Out_of_range := 80,
eLON_rSpan_Out_of_range := 81,
eLON_arrValue01_Out_of_range := 85,
eLON_arrValue02_Out_of_range := 86,
eLON_arrValue03_Out_of_range := 87,
eLON_arrValue04_Out_of_range := 88,
eLON_arrValue05_Out_of_range := 89,
eLON_arrValue06_Out_of_range := 90,
eLON_arrValue07_Out_of_range := 91,
eLON_arrValue08_Out_of_range := 92,
eLON_arrValue09_Out_of_range := 93,
eLON_arrValue10_Out_of_range := 100,
eLON_arrValue11_Out_of_range := 101,
eLON_arrValue12_Out_of_range := 102,
eLON_arrValue13_Out_of_range := 103,
eLON_arrValue14_Out_of_range := 104,
eLON_arrValue15_Out_of_range := 105,
eLON_arrValue16_Out_of_range := 106,
eLON_arrValue17_Out_of_range := 107,
eLON_arrValue18_Out_of_range := 108,
eLON_arrValue19_Out_of_range := 109,
eLON_arrValue20_Out_of_range := 115,
eLON_arrValue21_Out_of_range := 116,
eLON_arrValue22_Out_of_range := 117,
eLON_arrValue23_Out_of_range := 118,
eLON_arrValue24_Out_of_range := 119,
eLON_arrValue25_Out_of_range := 120,
eLON_arrValue26_Out_of_range := 121,
eLON_arrValue27_Out_of_range := 122,
eLON_arrValue28_Out_of_range := 123,
eLON_arrValue29_Out_of_range := 124,
eLON_arrValue30_Out_of_range := 130,
eLON_arrValue31_Out_of_range := 131,
eLON_arrValue32_Out_of_range := 132,
eLON_arrValue33_Out_of_range := 133,
eLON_arrValue34_Out_of_range := 134,
eLON_arrValue35_Out_of_range := 135,
eLON_arrValue36_Out_of_range := 136,
eLON_arrValue37_Out_of_range := 137,
eLON_arrValue38_Out_of_range := 138,
eLON_arrValue39_Out_of_range := 139,
eLON_arrValue40_Out_of_range := 140,
eLON_087uiDay_Out_of_range := 145,
eLON_087uiHour_Out_of_range := 146,
eLON_087uiMinute_Out_of_range := 147,
eLON_087uiSecond_Out_of_range := 148,
eLON_087uiMillisecond_Out_of_range := 149,
eLON_ePriority_level_Out_of_range := 155,
eLON_eAlarm_type_Out_of_range := 156,
eLON_Currency_Out_of_range := 160,
eLON_diRw_ptr_Out_of_range := 165,
eLON_Object_request_Out_of_range := 170,
eLON_094eLearn_Out_of_range := 175,
eLON_094uiHour_Out_of_range := 176,
eLON_094uiMinute_Out_of_range := 177,
eLON_094uiSecond_Out_of_range := 178,
eLON_094uiMillisecond_Out_of_range := 179,
eLON_095rValue_Out_of_range := 185,
eLON_095siState_Out_of_range := 186,
eLON_byInterp_pts_0_to_1_Out_of_range := 190,
eLON_byInterp_pts_1_to_2_Out_of_range := 191,
eLON_byInterp_pts_2_to_3_Out_of_range := 192,
eLON_byInterp_pts_3_to_4_Out_of_range := 193,
eLON_byInterp_pts_4_to_5_Out_of_range := 194,
eLON_byInterp_pts_5_to_6_Out_of_range := 195,
eLON_byInterp_pts_6_to_0_Out_of_range := 196,
eLON_rOccupied_cool_Out_of_range := 200,
eLON_rStandby_cool_Out_of_range := 201,
eLON_rUnoccupied_cool_Out_of_range := 202,
eLON_rOccupied_heat_Out_of_range := 203,
eLON_rStandby_heat_Out_of_range := 204,
eLON_rUnoccupied_heat_Out_of_range := 205,
eLON_111rPercent_Out_of_range := 210,
eLON_111eState_Out_of_range := 211,
eLON_rMode_Out_of_range := 215,
eLON_rHeat_output_primary_Out_of_range := 216,
eLON_rHeat_output_secondary_Out_of_range := 217,
eLON_rCool_output_Out_of_range := 218,
eLON_rEcon_output_Out_of_range := 219,
eLON_rFan_output_Out_of_range := 220,
eLON_115eFunction_Out_of_range := 225,
eLON_eFunction_Out_of_range := 226,
eLON_rSetting_Out_of_range := 227,
eLON_rRotation_Out_of_range := 228,
eLON_rFade_time_Out_of_range := 229,
eLON_rDelay_time_Out_of_range := 230,
eLON_eChlr_run_mode_Out_of_range := 235,
eLON_eChlr_op_mode_Out_of_range := 236,
eLON_eNext_state_Out_of_range := 240,
eLON_eCurrent_state_Out_of_range := 241,
eLON_diSecond_time_offset_Out_of_range := 245,
eLON_eType_of_description_Out_of_range := 246,
eLON_byHour_of_start_DST_Out_of_range := 250,
eLON_byMinute_of_start_DST_Out_of_range := 251,
eLON_bySecond_of_start_DST_Out_of_range := 252,
eLON_byHour_of_end_DST_Out_of_range := 260,
eLON_byMinute_of_end_DST_Out_of_range := 261,
eLON_bySecond_of_end_DST_Out_of_range := 262,
eLON_stStart_DST_uiG_day_of_start_DST_Out_of_range := 263,
eLON_stStart_DST_uiJ_day_of_start_DST_Out_of_range := 264,
eLON_stStart_DST_stM_start_DST_byWeek_of_start_DST_Out_of_range := 266,
eLON_stStart_DST_stM_start_DST_byMonth_of_start_DST_Out_of_range := 267,
eLON_stStart_DST_stM_start_DST_byDateday_of_start_DST_Out_of_range := 268,
eLON_stStart_DST_stM_start_DST_byDateday_of_start_DST_Out_of_range := 268,
eLON_stStart_DST_stM_start_DST_byMonth_of_start_DST_Out_of_range := 269,
eLON_stStart_DST_stM_start_DST_byWeek_of_start_DST_Out_of_range := 270,
eLON_stEnd_DST_uiG_day_of_end_DST_Out_of_range := 271,
eLON_stEnd_DST_uiJ_day_of_end_DST_Out_of_range := 272,
eLON_stEnd_DST_stM_end_DST_byWeek_of_end_DST_Out_of_range := 273,
eLON_stEnd_DST_stM_end_DST_byMonth_of_end_DST_Out_of_range := 274,
eLON_stEnd_DST_stM_end_DST_byDateday_of_end_DST_Out_of_range := 275,
eLON_byLatitude_deg_Out_of_range := 280,
eLON_rLatitude_min_Out_of_range := 281,
eLON_bylongitude_deg_Out_of_range := 282,
eLON_rLongitude_min_Out_of_range := 283,
eLON_byNr_decimals_Out_of_range := 290,
eLON_eUnit_Out_of_range := 291,
eLON_137eUnit_Out_of_range := 295,
eLON_137byNr_decimals_Out_of_range := 296,
eLON_137byStatus_Out_of_range := 297,
eLON_137uiYear_Out_of_range := 298,
eLON_137uiMonth_Out_of_range := 299,
eLON_137uiDay_Out_of_range := 300,
eLON_137uiHour_Out_of_range := 301,
eLON_137uiMinute_Out_of_range := 302,
eLON_137uiSecond_Out_of_range := 303,
eLON_bySender_prio_Out_of_range := 310,
eLON_eStatus_Out_of_range := 315,
eLON_stSender_uiID_Out_of_range := 316,
eLON_stSender_stRange_uiLower_Out_of_range := 317,
eLON_stSender_stRange_uiUpper_Out_of_range := 318,
eLON_uiController_id_Out_of_range := 319,
eLON_ePan_dir_Out_of_range := 325,
eLON_rPan_speed_Out_of_range := 326,
eLON_eTilt_dir_Out_of_range := 327,
eLON_rTilt_speed_Out_of_range := 328,
eLON_eZoom_Out_of_range := 329,
eLON_rZoom_speed_Out_of_range := 330,
eLON_eAction_Out_of_range := 335,
eLON_byController_prio_Out_of_range := 340,
eLON_152eFunction_Out_of_range := 341,
eLON_152eAction_Out_of_range := 342,
eLON_stValue_stAbspos_rZoom_Out_of_range := 343,
eLON_stValue_stAbspos_rTilt_Out_of_range := 344,
eLON_stValue_stAbspos_rPan_Out_of_range := 345,
eLON_eMain_pump_Out_of_range := 350,
eLON_eBooster_pump_Out_of_range := 351,
eLON_ePriority_level_Out_of_range := 352,
eLON_eProcess_ready_Out_of_range := 353,
eLON_eEmergency_stop_activated_Out_of_range := 354,
eLON_eMain_pump_drive_enabled_Out_of_range := 355,
eLON_eBooster_pump_drive_enabled_Out_of_range := 356,
eLON_eMaintenance_required_Out_of_range := 357,
eLON_eControl_status_Out_of_range := 365,
eLON_stControl_device_addr_byDomain_length_Out_of_range := 366,
eLON_stControl_device_addr_bySubnet_Out_of_range := 367,
eLON_stControl_device_addr_byNode_Out_of_range := 368,
eLON_rExhaust_temperature_Out_of_range := 375,
eLON_rExhaust_pressure_Out_of_range := 376,
eLON_rShaft_seal_purge_pressure_Out_of_range := 377,
eLON_rSupply_voltage_Out_of_range := 378,
eLON_eCoolant_flow_low_Out_of_range := 379,
eLON_eDilution_active_Out_of_range := 380,
eLON_eBallast_dilution_active_Out_of_range := 381,
eLON_eInlet_purge_dilution_active_Out_of_range := 382,
eLON_eExhaust_dilution_active_Out_of_range := 383,
eLON_eDilution_flow_Out_of_range := 384,
eLON_ePower_supply_on_Out_of_range := 385,
eLON_rRotational_speed_Out_of_range := 390,
eLON_rBody_temperature_Out_of_range := 391,
eLON_rMotor_external_temperature_Out_of_range := 392,
eLON_rMotor_internal_temperature_Out_of_range := 393,
eLON_eMotor_overloaded_Out_of_range := 394,
eLON_eOil_level_low_Out_of_range := 395,
eLON_eMotor_overloaded_Out_of_range := 396,
eLON_ePower_usage_Out_of_range := 397,
eLON_eTemperature_control_Out_of_range := 398,
eLON_eElectromagnetic_brake_active_Out_of_range := 399,
eLON_eFriction_brake_active_Out_of_range := 400,
eLON_eGas_brake_active_Out_of_range := 401,
eLON_rSupply_voltage_Out_of_range := 402,
eLON_rShaft_seal_purge_pressure_Out_of_range := 403,
eLON_rExhaust_pressure_Out_of_range := 404,
eLON_rExhaust_temperature_Out_of_range := 405,
eLON_stControl_device_addr_byNode_Out_of_range := 406,
eLON_stSender_stRange_uiUpper_Out_of_range := 407,
eLON_stSender_stRange_uiLower_Out_of_range := 408,
eLON_uiController_id_Out_of_range := 409,
eLON_eError_Out_of_range := 410,
eLON_164ePriority_level_Out_of_range := 411,
eLON_164eAlarm_type_Out_of_range := 412,
eLON_byType_scope_Out_of_range := 420,
eLON_uiType_index_Out_of_range := 421,
eLON_eType_category_Out_of_range := 422,
eLON_byType_length_Out_of_range := 423,
eLON_eStatus_Out_of_range := 430,
eLON_byManufacturer_Out_of_range := 435,
eLON_eDevice_select_Out_of_range := 440,
Version: 1.5
eLON_no_Error: No error is pending.

eLON_Value_out_of_range: The input variable "Value" is outside the permitted range. The value was not sent. "Value" can have different formats with corresponding prefix (e.g. LREAL = lrValue).

eLON_Terminal_not_ready: The function block "FB_LON_KL6401" passes through an initialization step chain (query terminal type, query firmware etc.) when the PLC is started. This message is issued as along as the initialization is in progress. If an error is pending after a PLC reset, the controller should be de-energized once.

eLON_Wrong_SNVT_Typ: The received SNVT type does not match the SNVT type of the addressed NV index (input variable "wId").

eLON_Wrong_wNVIndex: Incorrect NV index.

eKL6401_Wrong_Terminal: No KL6401 was detected.

eKL6401_Error: The function block "FB_LON_KL6401" has an error. The error code is shown at output "dwErrorKL".

eKL6401_Terminal_is_not_initialized: The terminal is not initialized. This message usually means that there is no connection to the terminal. Terminal linked to the variables in the System Manager? Terminal plugged in incorrectly? Everything corrected, everything translated and re-read into the System Manager?

eLON_L_star_Out_of_range: SNVT 70 / The input variable "stValue.L_star" is outside the permitted range. The value was not sent.

eLON_A_star_Out_of_range: SNVT 70 / The input variable "stValue.A_star" is outside the permitted range. The value was not sent.

eLON_B_star_Out_of_range: SNVT 70 / The input variable "stValue.B_star" is outside the permitted range. The value was not sent.

eLON_eRequest_Out_of_range: SNVT 73 / The input variable "stValue.eRequest" is outside the permitted range. The value was not sent.

eLON_wYear_Out_of_range: SNVT 084 / 088 / The input variable "stValue.wYear" is outside the permitted range. The value was not sent.

eLON_wMonth_Out_of_range: SNVT 084 / 088 / The input variable "stValue.wMonth" is outside the permitted range. The value was not sent.

eLON_wDay_Out_of_range: SNVT 084 / 088 / The input variable "stValue.wDay" is outside the permitted range. The value was not sent.

eLON_wHour_Out_of_range: SNVT 084 / 088 / The input variable "stValue.wHour" is outside the permitted range. The value was not sent.

eLON_wMinute_Out_of_range: SNVT 084 / 088 / The input variable "stValue.wMinute" is outside the permitted range. The value was not sent.

eLON_wSecond_Out_of_range: SNVT 084 / 088 / The input variable "stValue.wSecond" is outside the permitted range. The value was not sent.
eLON_wMilliscond_Out_of_range: SNVT 73 / The input variable "stValue.wMilliscond" is outside the permitted range. The value was not sent.

eLON_rZero_Out_of_range: SNVT 085 / The input variable "stValue.rZero" is outside the permitted range. The value was not sent.

eLON_rSpan_Out_of_range: SNVT 085 / The input variable "stValue.rSpan" is outside the permitted range. The value was not sent.

eLON_arrValue01_Out_of_range: SNVT 086 / The input variable "arrValue[1]" is outside the permitted range. The value was not sent.

eLON_arrValue02_Out_of_range: SNVT 086 / The input variable "arrValue[2]" is outside the permitted range. The value was not sent.

eLON_arrValue03_Out_of_range: SNVT 086 / The input variable "arrValue[3]" is outside the permitted range. The value was not sent.

eLON_arrValue04_Out_of_range: SNVT 086 / The input variable "arrValue[4]" is outside the permitted range. The value was not sent.

eLON_arrValue05_Out_of_range: SNVT 086 / The input variable "arrValue[5]" is outside the permitted range. The value was not sent.

eLON_arrValue06_Out_of_range: SNVT 086 / The input variable "arrValue[6]" is outside the permitted range. The value was not sent.

eLON_arrValue07_Out_of_range: SNVT 086 / The input variable "arrValue[7]" is outside the permitted range. The value was not sent.

eLON_arrValue08_Out_of_range: SNVT 086 / The input variable "arrValue[8]" is outside the permitted range. The value was not sent.

eLON_arrValue09_Out_of_range: SNVT 086 / The input variable "arrValue[9]" is outside the permitted range. The value was not sent.

eLON_arrValue10_Out_of_range: SNVT 086 / The input variable "arrValue[10]" is outside the permitted range. The value was not sent.

eLON_arrValue11_Out_of_range: SNVT 086 / The input variable "arrValue[11]" is outside the permitted range. The value was not sent.

eLON_arrValue12_Out_of_range: SNVT 086 / The input variable "arrValue[12]" is outside the permitted range. The value was not sent.

eLON_arrValue13_Out_of_range: SNVT 086 / The input variable "arrValue[13]" is outside the permitted range. The value was not sent.

eLON_arrValue14_Out_of_range: SNVT 086 / The input variable "arrValue[14]" is outside the permitted range. The value was not sent.

eLON_arrValue15_Out_of_range: SNVT 086 / The input variable "arrValue[15]" is outside the permitted range. The value was not sent.

eLON_arrValue16_Out_of_range: SNVT 086 / The input variable "arrValue[16]" is outside the permitted range. The value was not sent.

eLON_arrValue17_Out_of_range: SNVT 086 / The input variable "arrValue[17]" is outside the permitted range. The value was not sent.

eLON_arrValue18_Out_of_range: SNVT 086 / The input variable "arrValue[18]" is outside the permitted range. The value was not sent.

eLON_arrValue19_Out_of_range: SNVT 086 / The input variable "arrValue[19]" is outside the permitted range. The value was not sent.

eLON_arrValue20_Out_of_range: SNVT 086 / The input variable "arrValue[20]" is outside the permitted range. The value was not sent.
eLON_arrValue21_Out_of_range: SNVT 086 / The input variable "arrValue[21]" is outside the permitted range. The value was not sent.

eLON_arrValue22_Out_of_range: SNVT 086 / The input variable "arrValue[22]" is outside the permitted range. The value was not sent.

eLON_arrValue23_Out_of_range: SNVT 086 / The input variable "arrValue[23]" is outside the permitted range. The value was not sent.

eLON_arrValue24_Out_of_range: SNVT 086 / The input variable "arrValue[24]" is outside the permitted range. The value was not sent.

eLON_arrValue25_Out_of_range: SNVT 086 / The input variable "arrValue[25]" is outside the permitted range. The value was not sent.

eLON_arrValue26_Out_of_range: SNVT 086 / The input variable "arrValue[26]" is outside the permitted range. The value was not sent.

eLON_arrValue27_Out_of_range: SNVT 086 / The input variable "arrValue[27]" is outside the permitted range. The value was not sent.

eLON_arrValue28_Out_of_range: SNVT 086 / The input variable "arrValue[28]" is outside the permitted range. The value was not sent.

eLON_arrValue29_Out_of_range: SNVT 086 / The input variable "arrValue[29]" is outside the permitted range. The value was not sent.

eLON_arrValue30_Out_of_range: SNVT 086 / The input variable "arrValue[30]" is outside the permitted range. The value was not sent.

eLON_arrValue31_Out_of_range: SNVT 086 / The input variable "arrValue[31]" is outside the permitted range. The value was not sent.

eLON_arrValue32_Out_of_range: SNVT 086 / The input variable "arrValue[32]" is outside the permitted range. The value was not sent.

eLON_arrValue33_Out_of_range: SNVT 086 / The input variable "arrValue[33]" is outside the permitted range. The value was not sent.

eLON_arrValue34_Out_of_range: SNVT 086 / The input variable "arrValue[34]" is outside the permitted range. The value was not sent.

eLON_arrValue35_Out_of_range: SNVT 086 / The input variable "arrValue[35]" is outside the permitted range. The value was not sent.

eLON_arrValue36_Out_of_range: SNVT 086 / The input variable "arrValue[36]" is outside the permitted range. The value was not sent.

eLON_arrValue37_Out_of_range: SNVT 086 / The input variable "arrValue[37]" is outside the permitted range. The value was not sent.

eLON_arrValue38_Out_of_range: SNVT 086 / The input variable "arrValue[38]" is outside the permitted range. The value was not sent.

eLON_arrValue39_Out_of_range: SNVT 086 / The input variable "arrValue[39]" is outside the permitted range. The value was not sent.

eLON_arrValue40_Out_of_range: SNVT 086 / The input variable "arrValue[40]" is outside the permitted range. The value was not sent.

eLON_087uiDay_Out_of_range: SNVT 087 / The input variable "stValue.uiDay" is outside the permitted range. The value was not sent.

eLON_087uiHour_Out_of_range: SNVT 087 / The input variable "stValue.uiHour" is outside the permitted range. The value was not sent.

eLON_087uiMinute_Out_of_range: SNVT 087 / The input variable "stValue.uiMinute" is outside the permitted range. The value was not sent.
eLON_087uiSecond_Out_of_range: SNVT 087 / The input variable "stValue.uiSecond" is outside the permitted range. The value was not sent.

eLON_087uiMillisecond_Out_of_range: SNVT 087 / The input variable "stValue.uiMillisecond" is outside the permitted range. The value was not sent.

eLON_ePriority_level_Out_of_range: SNVT 088 / The input variable "stValue.ePriority_level" is outside the permitted range. The value was not sent.

eLON_eAlarm_type_Out_of_range: SNVT 088 / The input variable "stValue.eAlarm_type" is outside the permitted range. The value was not sent.

eLON_Currency_Out_of_range: SNVT 089 / The input variable "stValue.Currency" is outside the permitted range. The value was not sent.

eLON_diRw_ptr_Out_of_range: SNVT 090 / The input variable "stValue.diRw_ptr" is outside the permitted range. The value was not sent.

eLON_Object_request_Out_of_range: SNVT 092 / The input variable "stValue.Object_request" is outside the permitted range. The value was not sent.

eLON_094eLearn_Out_of_range: SNVT 094 / The input variable "stValue.eLearn" is outside the permitted range. The value was not sent.

eLON_094uiHour_Out_of_range: SNVT 094 / The input variable "stValue.uiHour" is outside the permitted range. The value was not sent.

eLON_094uiMinute_Out_of_range: SNVT 094 / The input variable "stValue.uiMinute" is outside the permitted range. The value was not sent.

eLON_094uiSecond_Out_of_range: SNVT 094 / The input variable "stValue.uiSecond" is outside the permitted range. The value was not sent.

eLON_094uiMillisecond_Out_of_range: SNVT 094 / The input variable "stValue.uiMillisecond" is outside the permitted range. The value was not sent.

eLON_095rValue_Out_of_range: SNVT 095 / The input variable "stValue.rValue" is outside the permitted range. The value was not sent.

eLON_095siState_Out_of_range: SNVT 095 / The input variable "stValue.siState" is outside the permitted range. The value was not sent.

eLON_byInterp_pts_0_to_1_Out_of_range: SNVT 096 / The input variable "stValue.byInterp_pts_0_to_1" is outside the permitted range. The value was not sent.

eLON_byInterp_pts_1_to_2_Out_of_range: SNVT 096 / The input variable "stValue.byInterp_pts_1_to_2" is outside the permitted range. The value was not sent.

eLON_byInterp_pts_2_to_3_Out_of_range: SNVT 096 / The input variable "stValue.byInterp_pts_2_to_3" is outside the permitted range. The value was not sent.

eLON_byInterp_pts_3_to_4_Out_of_range: SNVT 096 / The input variable "stValue.byInterp_pts_3_to_4" is outside the permitted range. The value was not sent.

eLON_byInterp_pts_4_to_5_Out_of_range: SNVT 096 / The input variable "stValue.byInterp_pts_4_to_5" is outside the permitted range. The value was not sent.

eLON_byInterp_pts_5_to_6_Out_of_range: SNVT 096 / The input variable "stValue.byInterp_pts_5_to_6" is outside the permitted range. The value was not sent.

eLON_byInterp_pts_6_to_0_Out_of_range: SNVT 096 / The input variable "stValue.byInterp_pts_6_to_0" is outside the permitted range. The value was not sent.

eLON_rOccupied_cool_Out_of_range: SNVT 106 / The input variable "stValue.rOccupied_cool" is outside the permitted range. The value was not sent.

eLON_rStandby_cool_Out_of_range: SNVT 106 / The input variable "stValue.rStandby_cool" is outside the permitted range. The value was not sent.
eLON_rUnoccupied_cool_Out_of_range: SNVT 106 / The input variable "stValue.rUnoccupied_cool" is outside the permitted range. The value was not sent.

eLON_rOccupied_heat_Out_of_range: SNVT 106 / The input variable "stValue.rOccupied_heat" is outside the permitted range. The value was not sent.

eLON_rStandby_heat_Out_of_range: SNVT 106 / The input variable "stValue.rStandby_heat" is outside the permitted range. The value was not sent.

eLON_rUnoccupied_heat_Out_of_range: SNVT 106 / The input variable "stValue.rUnoccupied_heat" is outside the permitted range. The value was not sent.

eLON_111rPercent_Out_of_range: SNVT 111 / The input variable "stValue.rPercent" is outside the permitted range. The value was not sent.

eLON_111eState_Out_of_range: SNVT 111 / The input variable "stValue.eState" is outside the permitted range. The value was not sent.

eLON_eMode_Out_of_range: SNVT 112 / The input variable "stValue.eMode" is outside the permitted range. The value was not sent.

eLON_rHeat_output_primary_Out_of_range: SNVT 112 / The input variable "stValue.rHeat_output_primary" is outside the permitted range. The value was not sent.

eLON_rHeat_output_secondary_Out_of_range: SNVT 112 / The input variable "stValue.rHeat_output_secondary" is outside the permitted range. The value was not sent.

eLON_rCool_output_Out_of_range: SNVT 112 / The input variable "stValue.rCool_output" is outside the permitted range. The value was not sent.

eLON_rEcon_output_Out_of_range: SNVT 112 / The input variable "stValue.rEcon_output" is outside the permitted range. The value was not sent.

eLON_rFan_output_Out_of_range: SNVT 112 / The input variable "stValue.rFan_output" is outside the permitted range. The value was not sent.

eLON_115eFunction_Out_of_range: SNVT 115 / The input variable "stValue.eFunction" is outside the permitted range. The value was not sent.

eLON_eFunction_Out_of_range: SNVT 116 / 117 / The input variable "stValue.eFunction" is outside the permitted range. The value was not sent.

eLON_rSetting_Out_of_range: SNVT 116 / 117 / The input variable "stValue.rSetting" is outside the permitted range. The value was not sent.

eLON_rRotation_Out_of_range: SNVT 116 / 117 / The input variable "stValue.rRotation" is outside the permitted range. The value was not sent.

eLON_rFade_time_Out_of_range: SNVT 116 / The input variable "stValue.rFade_time" is outside the permitted range. The value was not sent.

eLON_rDelay_time_Out_of_range: SNVT 116 / The input variable "stValue.rDelay_time" is outside the permitted range. The value was not sent.

eLON_eChlr_run_mode_Out_of_range: SNVT 127 / The input variable "stValue.eChlr_run_mode" is outside the permitted range. The value was not sent.

eLON_eChlr_op_mode_Out_of_range: SNVT 127 / The input variable "stValue.eChlr_op_mode" is outside the permitted range. The value was not sent.

eLON_eNext_state_Out_of_range: SNVT 128 / The input variable "stValue.eNext_state" is outside the permitted range. The value was not sent.

eLON_eCurrent_state_Out_of_range: SNVT 128 / The input variable "stValue.eCurrent_state" is outside the permitted range. The value was not sent.

eLON_diSecond_time_offset_Out_of_range: SNVT 134 / The input variable "stValue.diSecond_time_offset" is outside the permitted range. The value was not sent.
eLON_eType_of_description_Out_of_range: SNVT 134 / The input variable "stValue.eType_of_description" is outside the permitted range. The value was not sent.

eLON_byHour_of_start_DST_Out_of_range: SNVT 134 / The input variable "stValue.byHour_of_start_DST" is outside the permitted range. The value was not sent.

eLON_byMinute_of_start_DST_Out_of_range: SNVT 134 / The input variable "stValue.byMinute_of_start_DST" is outside the permitted range. The value was not sent.

eLON_bySecond_of_start_DST_Out_of_range: SNVT 134 / The input variable "stValue.bySecond_of_start_DST" is outside the permitted range. The value was not sent.

eLON_byHour_of_end_DST_Out_of_range: SNVT 134 / The input variable "stValue.byHour_of_end_DST" is outside the permitted range. The value was not sent.

eLON_byMinute_of_end_DST_Out_of_range: SNVT 134 / The input variable "stValue.byMinute_of_end_DST" is outside the permitted range. The value was not sent.

eLON_bySecond_of_end_DST_Out_of_range: SNVT 134 / The input variable "stValue.bySecond_of_end_DST" is outside the permitted range. The value was not sent.

eLON_stStart_DST_uiG_day_of_start_DST_Out_of_range: SNVT 134 / The input variable "stValue.stStart_DST.uiG_day_of_start_DST" is outside the permitted range. The value was not sent.

eLON_stStart_DST_uiJ_day_of_start_DST_Out_of_range: SNVT 134 / The input variable "stValue.stStart_DST.uiJ_day_of_start_DST" is outside the permitted range. The value was not sent.

eLON_stStart_DST_stM_start_DST_byMonth_of_start_DST_Out_of_range: SNVT 134 / The input variable "stValue.stStart_DST.stM_start_DST.byMonth_of_start_DST" is outside the permitted range. The value was not sent.

eLON_stStart_DST_stM_start_DST_byWeek_of_start_DST_Out_of_range: SNVT 134 / The input variable "stValue.stStart_DST.stM_start_DST.byWeek_of_start_DST" is outside the permitted range. The value was not sent.

eLON_stStart_DST_stM_start_DST_eDateday_of_start_DST_Out_of_range: SNVT 134 / The input variable "stValue.stStart_DST.stM_start_DST.eDateday_of_start_DST" is outside the permitted range. The value was not sent.

eLON_stEnd_DST_uiG_day_of_end_DST_Out_of_range: SNVT 134 / The input variable "stValue.stEnd_DST.uiG_day_of_end_DST" is outside the permitted range. The value was not sent.

eLON_stEnd_DST_uiJ_day_of_end_DST_Out_of_range: SNVT 134 / The input variable "stValue.stEnd_DST.uiJ_day_of_end_DST" is outside the permitted range. The value was not sent.

eLON_stEnd_DST_stM_end_DST_byMonth_of_end_DST_Out_of_range: SNVT 134 / The input variable "stValue.stEnd_DST.stM_end_DST.byMonth_of_end_DST" is outside the permitted range. The value was not sent.

eLON_stEnd_DST_stM_end_DST_byWeek_of_end_DST_Out_of_range: SNVT 134 / The input variable "stValue.stEnd_DST.stM_end_DST.byWeek_of_end_DST" is outside the permitted range. The value was not sent.

eLON_stEnd_DST_stM_end_DST_eDateday_of_end_DST_Out_of_range: SNVT 134 / The input variable "stValue.stEnd_DST.stM_end_DST.eDateday_of_end_DST" is outside the permitted range. The value was not sent.

eLON_byLatitude_deg_Out_of_range: SNVT 135 / The input variable "stValue.byLatitude" is outside the permitted range. The value was not sent.

eLON_rLatitude_min_Out_of_range: SNVT 135 / The input variable "stValue.rLatitude" is outside the permitted range. The value was not sent.

eLON_bylongitude_deg_Out_of_range: SNVT 135 / The input variable "stValue.bylongitude_deg" is outside the permitted range. The value was not sent.

eLON_rLongitude_min_Out_of_range: SNVT 135 / The input variable "stValue.rLongitude_min" is outside the permitted range. The value was not sent.
eLON_byNr_decimals_Out_of_range: SNVT 136 / The input variable "stValue.byNr_decimals" is outside the permitted range. The value was not sent.
eLON_eUnit_Out_of_range: SNVT 136 / The input variable "stValue.eUnit" is outside the permitted range. The value was not sent.
eLON_137_eUnit_Out_of_range: SNVT 137 / The input variable "stValue.eUnit" is outside the permitted range. The value was not sent.
eLON_137_byNr_decimals_Out_of_range: SNVT 137 / The input variable "stValue.byNr_decimals" is outside the permitted range. The value was not sent.
eLON_137_byStatus_Out_of_range: SNVT 137 / The input variable "stValue.byStatus" is outside the permitted range. The value was not sent.
eLON_137_uiYear_Out_of_range: SNVT 137 / The input variable "stValue.uiYear" is outside the permitted range. The value was not sent.
eLON_137_uiMonth_Out_of_range: SNVT 137 / The input variable "stValue.uiMonth" is outside the permitted range. The value was not sent.
eLON_137_uiDay_Out_of_range: SNVT 137 / The input variable "stValue.uiDay" is outside the permitted range. The value was not sent.
eLON_137_uiHour_Out_of_range: SNVT 137 / The input variable "stValue.uiHour" is outside the permitted range. The value was not sent.
eLON_137_uiMinute_Out_of_range: SNVT 137 / The input variable "stValue.uiMinute" is outside the permitted range. The value was not sent.
eLON_137_uiSecond_Out_of_range: SNVT 137 / The input variable "stValue.uiSecond" is outside the permitted range. The value was not sent.
eLON_bySender_prio_Out_of_range: SNVT 148 / The input variable "stValue.bySender_prio" is outside the permitted range. The value was not sent.
eLON_eStatus_Out_of_range: SNVT 149 / The input variable "stValue.eStatus" is outside the permitted range. The value was not sent.
eLON_stSender_uiID_Out_of_range: SNVT 149 / The input variable "stValue.stSender.uiID" is outside the permitted range. The value was not sent.
eLON_stSender_stRange_uiLower_Out_of_range: SNVT 149 / The input variable "stValue.stSender.stRange.uiLower" is outside the permitted range. The value was not sent.
eLON_stSender_stRange_uiUpper_Out_of_range: SNVT 149 / The input variable "stValue.stSender.stRange.uiUpper" is outside the permitted range. The value was not sent.
eLON_uiController_id_Out_of_range: SNVT 149 / The input variable "stValue.uiController" is outside the permitted range. The value was not sent.
eLON_ePan_dir_Out_of_range: SNVT 150 / The input variable "stValue.ePan_dir" is outside the permitted range. The value was not sent.
eLON_rPan_speed_Out_of_range: SNVT 150 / The input variable "stValue.rPan_speed" is outside the permitted range. The value was not sent.
eLON_eTilt_dir_Out_of_range: SNVT 150 / The input variable "stValue.eTilt_dir" is outside the permitted range. The value was not sent.
eLON_rTilt_speed_Out_of_range: SNVT 150 / The input variable "stValue.rTilt_speed" is outside the permitted range. The value was not sent.
eLON_eZoom_Out_of_range: SNVT 150 / The input variable "stValue.eZoom" is outside the permitted range. The value was not sent.
eLON_rZoom_speed_Out_of_range: SNVT 150 / The input variable "stValue.rZoom_speed" is outside the permitted range. The value was not sent.
eLON_eAction_Out_of_range: SNVT 151 / The input variable "stValue.eAction" is outside the permitted range. The value was not sent.

eLON_byController_prio_Out_of_range: SNVT 152 / The input variable "stValue.byController" is outside the permitted range. The value was not sent.

eLON_152eFunction_Out_of_range: SNVT 152 / The input variable "stValue.eFunction" is outside the permitted range. The value was not sent.

eLON_152eAction_Out_of_range: SNVT 152 / The input variable "stValue.eAction" is outside the permitted range. The value was not sent.

eLON_stValue_stAbspos_rZoom_Out_of_range: SNVT 152 / The input variable "stValue.stValue.stAbspos.rZoom" is outside the permitted range. The value was not sent.

eLON_stValue_stAbspos_rTilt_Out_of_range: SNVT 152 / The input variable "stValue.stValue.stAbspos.rTilt" is outside the permitted range. The value was not sent.

eLON_stValue_stAbspos_rPan_Out_of_range: SNVT 152 / The input variable "stValue.stValue.stAbspos.rPan" is outside the permitted range. The value was not sent.

eLON_eMain_pump_Out_of_range: SNVT 156 / The input variable "stValue.eMain_pump" is outside the permitted range. The value was not sent.

eLON_eBooster_pump_Out_of_range: SNVT 156 / The input variable "stValue.eBooster_pump" is outside the permitted range. The value was not sent.

eLON_ePriority_level_Out_of_range: SNVT 156 / The input variable "stValue.ePriority_level" is outside the permitted range. The value was not sent.

eLON_eProcess_ready_Out_of_range: SNVT 156 / The input variable "stValue.eProcess_ready" is outside the permitted range. The value was not sent.

eLON_eEmergency_stop_activated_Out_of_range: SNVT 156 / The input variable "stValue.eEmergency_stop_activated" is outside the permitted range. The value was not sent.

eLON_eMain_pump_drive_enabled_Out_of_range: SNVT 156 / The input variable "stValue.eMain_pump_drive_enabled" is outside the permitted range. The value was not sent.

eLON_eBooster_pump_drive_enabled_Out_of_range: SNVT 156 / The input variable "stValue.eBooster_pump_drive_enabled" is outside the permitted range. The value was not sent.

eLON_eMaintenance_required_Out_of_range: SNVT 156 / The input variable "stValue.eMaintenance_required" is outside the permitted range. The value was not sent.

eLON_eControl_status_Out_of_range: SNVT 157 / The input variable "stValue.eControl_status" is outside the permitted range. The value was not sent.

eLON_stControl_device_addr_byDomain_length_Out_of_range: SNVT 157 / The input variable "stValue.stControl_device_addr.byDomain_length" is outside the permitted range. The value was not sent.

eLON_stControl_device_addr_bySubnet_Out_of_range: SNVT 157 / The input variable "stValue.stControl_device_addr.bySubnet" is outside the permitted range. The value was not sent.

eLON_stControl_device_addr_byNode_Out_of_range: SNVT 157 / The input variable "stValue.stControl_device_addr.byNode" is outside the permitted range. The value was not sent.

eLON_rExhaust_temperature_Out_of_range: SNVT 158 / The input variable "stValue.rExhaust_temperature" is outside the permitted range. The value was not sent.

eLON_rExhaust_pressure_Out_of_range: SNVT 158 / The input variable "stValue.rExhaust_pressure" is outside the permitted range. The value was not sent.

eLON_rShaft_seal_purge_pressure_Out_of_range: SNVT 158 / The input variable "stValue.rShaft_seal_purge_pressure" is outside the permitted range. The value was not sent.

eLON_rSupply_voltage_Out_of_range: SNVT 158 / The input variable "stValue.rSupply_voltage" is outside the permitted range. The value was not sent.
eLON_eCoolant_flow_low_Out_of_range: SNVT 158 / The input variable "stValue.eCoolant_flow_low" is outside the permitted range. The value was not sent.

eLON_eDilution_active_Out_of_range: SNVT 158 / The input variable "stValue.eDilution_active" is outside the permitted range. The value was not sent.

eLON_eBallast_dilution_active_Out_of_range: SNVT 158 / The input variable "stValue.eBallast_dilution_active" is outside the permitted range. The value was not sent.

eLON_eInlet_purge_dilution_active_Out_of_range: SNVT 158 / The input variable "stValue.eInlet_purge_dilution_active" is outside the permitted range. The value was not sent.

eLON_eExhaust_dilution_active_Out_of_range: SNVT 158 / The input variable "stValue.eExhaust_dilution_active" is outside the permitted range. The value was not sent.

eLON_eDilution_flow_Out_of_range: SNVT 158 / The input variable "stValue.eDilution_flow" is outside the permitted range. The value was not sent.

eLON_ePower_supply_on_Out_of_range: SNVT 158 / The input variable "stValue.ePower_supply_on" is outside the permitted range. The value was not sent.

eLON_rRotational_speed_Out_of_range: SNVT 159 / The input variable "stValue.rRotational_speed" is outside the permitted range. The value was not sent.

eLON_rBody_temperature_Out_of_range: SNVT 159 / The input variable "stValue.rBody" is outside the permitted range. The value was not sent.

eLON_rMotor_external_temperature_Out_of_range: SNVT 159 / The input variable "stValue.rMotor_external_temperature" is outside the permitted range. The value was not sent.

eLON_rMotor_internal_temperature_Out_of_range: SNVT 159 / The input variable "stValue.eMotor_overloaded" is outside the permitted range. The value was not sent.

eLON_eMotor_overloaded_Out_of_range: SNVT 159 / The input variable "stValue.eMotor_overloaded" is outside the permitted range. The value was not sent.

eLON_eOil_level_low_Out_of_range: SNVT 159 / The input variable "stValue.ePhase_imbalance_detected" is outside the permitted range. The value was not sent.

eLON_ePhase_imbalance_detected_Out_of_range: SNVT 159 / The input variable "stValue.ePhase_imbalance_detected" is outside the permitted range. The value was not sent.

eLON_rCurrent_usage_Out_of_range: SNVT 159 / The input variable "stValue.rCurrent_usage" is outside the permitted range. The value was not sent.

eLON_rPower_usage_Out_of_range: SNVT 159 / The input variable "stValue.ePower_supply_on" is outside the permitted range. The value was not sent.

eLON_eTemperature_control_Out_of_range: SNVT 159 / The input variable "stValue.eElectromagnetic_brake_active" is outside the permitted range. The value was not sent.

eLON_eElectromagnetic_brake_active_Out_of_range: SNVT 159 / The input variable "stValue.eElectromagnetic_brake_active" is outside the permitted range. The value was not sent.

eLON_eFriction_brake_active_Out_of_range: SNVT 159 / The input variable "stValue.eFriction_brake_active" is outside the permitted range. The value was not sent.

eLON_eGas_brake_active_Out_of_range: SNVT 159 / The input variable "stValue.eGas_brake_active" is outside the permitted range. The value was not sent.

eLON_164iMilliseconds_Out_of_range: SNVT 164 / The input variable "stValue.iMilliseconds" is outside the permitted range. The value was not sent.

eLON_164ePriority_level_Out_of_range: SNVT 164 / The input variable "stValue.ePriority_level" is outside the permitted range. The value was not sent.

eLON_164eAlarm_type_Out_of_range: SNVT 164 / The input variable "stValue.eAlarm" is outside the permitted range. The value was not sent.
eLON_byType_scope_Out_of_range: SNVT 166 / The input variable "stValue.byType_scope" is outside the permitted range. The value was not sent.
eLON_uiType_index_Out_of_range: SNVT 166 / The input variable "stValue.uiType_index" is outside the permitted range. The value was not sent.
eLON_eType_category_Out_of_range: SNVT 166 / The input variable "stValue.eType_category" is outside the permitted range. The value was not sent.
eLON_byType_length_Out_of_range: SNVT 166 / The input variable "stValue.byType" is outside the permitted range. The value was not sent.
eLON_eCmd_fb_Out_of_range: SNVT 170 / The input variable "stValue.eCmd_fb" is outside the permitted range. The value was not sent.
eLON_byManufacturer_Out_of_range: SNVT 172 / The input variable "stValue.byManufacturer" is outside the permitted range. The value was not sent.
eLON_eDevice_select_Out_of_range: SNVT 175 / The input variable "stValue.eDevice_select" is outside the permitted range. The value was not sent.
eLON_stPos_eFunction_Out_of_range: SNVT 180 / The input variable "stValue.stPos_eFunction" is outside the permitted range. The value was not sent.
eLON_stPos_rSetting_Out_of_range: SNVT 180 / The input variable "stValue.stPos.rSetting" is outside the permitted range. The value was not sent.
eLON_stPos_rRotation_Out_of_range: SNVT 180 / The input variable "stValue.stPos.rRotation" is outside the permitted range. The value was not sent.
eLON_eCmd_source_Out_of_range: SNVT 180 / The input variable "stValue.eCmd_source" is outside the permitted range. The value was not sent.
eLON_eError_code_Out_of_range: SNVT 180 / The input variable "stValue._eError_code" is outside the permitted range. The value was not sent.
eLON_181stAddr_talk_eAudio_sensor_type_Out_of_range: SNVT 181 / The input variable "stValue.stAddr_talk.eAudio_sensor_type" is outside the permitted range. The value was not sent.
eLON_181stAddr_talk_byCar_id_Out_of_range: SNVT 181 / The input variable "stValue.stAddr_talk.byCar_id" is outside the permitted range. The value was not sent.
eLON_181stAddr_talk_byLocation_Out_of_range: SNVT 181 / The input variable "stValue.stAddr_talk.byLocation" is outside the permitted range. The value was not sent.
eLON_181stAddr_talk_byUnit_id_Out_of_range: SNVT 181 / The input variable "stValue.stAddr_talk.byUnit" is outside the permitted range. The value was not sent.
eLON_181stAddr_init_eAudio_sensor_type_Out_of_range: SNVT 181 / The input variable "stValue.stAddr_init.eAudio_sensor_type" is outside the permitted range. The value was not sent.
eLON_181stAddr_init_byCar_id_Out_of_range: SNVT 181 / The input variable "stValue.stAddr_init.byCar_id" is outside the permitted range. The value was not sent.
eLON_181stAddr_init_byLocation_Out_of_range: SNVT 181 / The input variable "stValue.stAddr_init.byLocation" is outside the permitted range. The value was not sent.
eLON_181stAddr_init_byUnit_id_Out_of_range: SNVT 181 / The input variable "stValue.stAddr_init.byUnit" is outside the permitted range. The value was not sent.
eLON_181eAudio_type_Out_of_range: SNVT 181 / The input variable "stValue.eAudio_type" is outside the permitted range. The value was not sent.
eLON_181byAudio_line_Out_of_range: SNVT 181 / The input variable "stValue.byAudio_line" is outside the permitted range. The value was not sent.
eLON_181stAddr_dest_stP2p_eAudio_sensor_type_Out_of_range: SNVT 181 / The input variable "stValue.stAddr_dest.stP2p.eAudio_sensor_type" is outside the permitted range. The value was not sent.
eLON_181stAddr_dest_stP2p_byCar_id_Out_of_range: SNVT 181 / The input variable "stValue.stAddr_dest.stP2p.byLocation" is outside the permitted range. The value was not sent.

eLON_181stAddr_dest_stP2p_byLocation_Out_of_range: SNVT 181 / The input variable "stValue." is outside the permitted range. The value was not sent.

eLON_181stAddr_dest_stP2p_byUnit_id_Out_of_range: SNVT 181 / The input variable "stValue.stAddr_dest.stP2p.byUnit_id" is outside the permitted range. The value was not sent.

eLON_stAddr_dest_stP2p_eAudio_sensor_type_Out_of_range: SNVT 182 / The input variable "stValue.stAddr_dest.stP2p.eAudio_sensor_type" is outside the permitted range. The value was not sent.

eLON_stAddr_dest_stP2p_byCar_id_Out_of_range: SNVT 182 / The input variable "stValue.stAddr_dest.stP2p.byCar" is outside the permitted range. The value was not sent.

eLON_stAddr_dest_stP2p_byLocation_Out_of_range: SNVT 182 / The input variable "stValue.stAddr_dest.stP2p.byLocation" is outside the permitted range. The value was not sent.

eLON_stAddr_dest_stP2p_byUnit_id_Out_of_range: SNVT 182 / The input variable "stValue.stAddr_dest.stP2p.byUnit_id" is outside the permitted range. The value was not sent.

eLON_stAddr_init_eAudio_sensor_type_Out_of_range: SNNT 182 / The input variable "stValue.stAddr_init.byCar_id" is outside the permitted range. The value was not sent.

eLON_stAddr_init_byCar_id_Out_of_range: SNVT 182 / The input variable "stValue." is outside the permitted range. The value was not sent.

eLON_stAddr_init_byLocation_Out_of_range: SNVT 182 / The input variable "stValue.stAddr_init.byLocation" is outside the permitted range. The value was not sent.

eLON_stAddr_init_byUnit_id_Out_of_range: SNVT 182 / The input variable "stValue.stAddr_init.byUnit_id" is outside the permitted range. The value was not sent.

eLON_eAudio_type_Out_of_range: SNVT 182 / The input variable "stValue.eAudio_type" is outside the permitted range. The value was not sent.

eLON_eCycle_Out_of_range: SNVT 184 / The input variable "stValue.eCycle" is outside the permitted range. The value was not sent.

eLON_eSubcycle_Out_of_range: SNVT 184 / The input variable "stValue.eSubcycle" is outside the permitted range. The value was not sent.

eLON_stFunction_eProgram_Out_of_range: SNVT 184 / The input variable "stValue.stFunction.eProgram" is outside the permitted range. The value was not sent.

eLON_stFunction_stWash_eLoad_level_Out_of_range: SNVT 184 / The input variable "stValue.stFunction.stWash.eLoad_level" is outside the permitted range. The value was not sent.

eLON_stFunction_stWash_ePrewash_Out_of_range: SNVT 184 / The input variable "stValue.stFunction.stWash.ePrewash" is outside the permitted range. The value was not sent.

eLON_stFunction_stRinse_eOption_Out_of_range: SNVT 184 / The input variable "stValue.stFunction.stRinse.eOption" is outside the permitted range. The value was not sent.

eLON_stFunction_stRinse_byRepeat_Out_of_range: SNVT 184 / The input variable "stValue.stFunction.stRinse.byRepeat" is outside the permitted range. The value was not sent.

eLON_stFunction_stSpin_eHold_Out_of_range: SNVT 184 / The input variable "stValue.stFunction.stSpin.eHold" is outside the permitted range. The value was not sent.

eLON_stFunction_stDry_byTemp_Out_of_range: SNVT 184 / The input variable "stValue.stFunction.stDry.byTemp" is outside the permitted range. The value was not sent.

eLON_stFunction_stDry_stDuration_eDryness_Out_of_range: SNVT 184 / The input variable "stValue.stFunction.stDry.stDuration.eDryness" is outside the permitted range. The value was not sent.

eLON_186eCycle_Out_of_range: SNVT 186 / The input variable "stValue.eCycle" is outside the permitted range. The value was not sent.
eLON_186eSubcycle_Out_of_range: SNVT 186 / The input variable "stValue.eSubcycle" is outside the permitted range. The value was not sent.

eLON_stWasher_command_data_eCycle_Out_of_range: SNVT 186 / The input variable "stValue.stWasher_command_data.eCycle" is outside the permitted range. The value was not sent.

eLON_stWasher_command_data_eSubcycle_Out_of_range: SNVT 186 / The input variable "stValue.stWasher_command_data.eSubcycle" is outside the permitted range. The value was not sent.

eLON_stWasher_command_data_stFunction_eProgram_Out_of_range: SNVT 186 / The input variable "stValue.stWasher_command_data.stFunction.eProgram" is outside the permitted range. The value was not sent.

eLON_stWasher_command_data_stFunction_stWash_eLoad_level_Out_of_range: SNVT 186 / The input variable "stValue.stWasher_command_data.stFunction.stWash.eLoad" is outside the permitted range. The value was not sent.

eLON_stWasher_command_data_stFunction_stWash_ePrewash_Out_of_range: SNVT 186 / The input variable "stValue.stWasher_command_data.stFunction.stWash.ePrewash" is outside the permitted range. The value was not sent.

eLON_stWasher_command_data_stFunction_stRinse_eOption_Out_of_range: SNVT 186 / The input variable "stValue.stWasher_command_data.stFunction.stRinse.eOption" is outside the permitted range. The value was not sent.

eLON_stWasher_command_data_stFunction_stRinse_byRepeat_Out_of_range: SNVT 186 / The input variable "stValue.stWasher_command_data.stFunction.stRinse.byRepeat" is outside the permitted range. The value was not sent.

eLON_stWasher_command_data_stFunction_stSpin_eHold_Out_of_range: SNVT 186 / The input variable "stValue.stWasher_command_data.stFunction.stSpin.eHold" is outside the permitted range. The value was not sent.

eLON_stWasher_command_data_stFunction_stDry_byTemp_Out_of_range: SNVT 186 / The input variable "stValue.stWasher_command_data.stFunction.stDry.byTemp" is outside the permitted range. The value was not sent.

eLON_stWasher_command_data_stFunction_stDry_stDuration_eDryness_Out_of_range: SNVT 186 / The input variable "stValue.stWasher_command_data.stFunction.stDry.stDuration.eDryness" is outside the permitted range. The value was not sent.

eLON_eState_Out_of_range: SNVT 189 / The input variable "stValue.eState" is outside the permitted range. The value was not sent.

eLON_stSetting_rValue_Out_of_range: SNVT 189 / The input variable "stValue.stSettings.rValue" is outside the permitted range. The value was not sent.

eLON_stSetting_rChange_Out_of_range: SNVT 189 / The input variable "stValue.stSettings.rChange" is outside the permitted range. The value was not sent.

eLON_stSetting_rMultiplier_Out_of_range: SNVT 189 / The input variable "stValue.stSettings.rMultiplier" is outside the permitted range. The value was not sent.

eLON_stSetting_iAngle_Out_of_range: SNVT 189 / The input variable "stValue.stSettings.iAngle" is outside the permitted range. The value was not sent.

eLON_stSetting_byGroup_number_Out_of_range: SNVT 189 / The input variable "stValue.stSettings.byGroup_number" is outside the permitted range. The value was not sent.

eLON_stSetting_siFan_level_Out_of_range: SNVT 189 / The input variable "stValue.stSettings.siFan_level" is outside the permitted range. The value was not sent.

eLON_stColor_value_stCIE1931_lumen_rX_Out_of_range: SNVT 190 / The input variable "stValue.stColor_value.stCIE1931_lumen.rX" is outside the permitted range. The value was not sent.

eLON_stColor_value_stCIE1931_lumen_rY_Out_of_range: SNVT 190 / The input variable "stValue.stColor_value.stCIE1931_lumen.rY" is outside the permitted range. The value was not sent.
eLON_stColor_value_stCIE1931_lumen_udiAbsolute_Y_Out_of_range: SNVT 190 / The input variable "stValue.stColor_value.stCIE1931_lumen.udiAbsolute_Y" is outside the permitted range. The value was not sent.

eLON_stColor_value_stCIE1931_percent_rX_Out_of_range: SNVT 190 / The input variable "stValue.stColor_value.stCIE1931_percent.rX" is outside the permitted range. The value was not sent.

eLON_stColor_value_stCIE1931_percent_rY_Out_of_range: SNVT 190 / The input variable "stValue.stColor_value.stCIE1931_percent.rY" is outside the permitted range. The value was not sent.

eLON_stColor_value_stCIE1931_percent_rPercent_Y_Out_of_range: SNVT 190 / The input variable "stValue.stColor_value.stCIE1931_percent.rPercent_Y" is outside the permitted range. The value was not sent.

eLON_stColor_value_uiColor_temperature_Out_of_range: SNVT 190 / The input variable "stValue.stColor_value.uiColor_temperature" is outside the permitted range. The value was not sent.

eLON_191eStatus_Out_of_range: SNVT 191 / The input variable "stValue.Status" is outside the permitted range. The value was not sent.

eLON_uiLog_number_Out_of_range: SNVT 191 / The input variable "stValue.uiLog_number" is outside the permitted range. The value was not sent.

eLON_rLevel_Out_of_range: SNVT 191 / The input variable "stValue.rLevel" is outside the permitted range. The value was not sent.

eLON_stCurrent_notify_time_rHundredths_Out_of_range: SNVT 191 / The input variable "stValue.stCurrent_notify_time.rHundredths" is outside the permitted range. The value was not sent.

eLON_stPrevious_notify_time_rHundredths_Out_of_range: SNVT 191 / The input variable "stValue.stPrevious_notify_time.rHundredths" is outside the permitted range. The value was not sent.

eLON_rHundredths_Out_of_range: SNVT 192 / The input variable "stValue.rHundredths" is outside the permitted range. The value was not sent.

eLON_stStart_time_rHundredths_Out_of_range: SNVT 193 / The input variable "stValue.stStart_time.rHundredths" is outside the permitted range. The value was not sent.

eLON_stEnd_time_rHundredths_Out_of_range: SNVT 193 / The input variable "stValue.stEnd_time.rHundredths" is outside the permitted range. The value was not sent.

eLON_rComplete_Out_of_range: SNVT 194 / The input variable "stValue.rComplete" is outside the permitted range. The value was not sent.

eLON_stTime_actual_rHundredths_Out_of_range: SNVT 199 / The input variable "stValue.stTime_actual.rHundredths" is outside the permitted range. The value was not sent.

eLON_stTime_previous_rHundredths_Out_of_range: SNVT 199 / The input variable "stValue.stTime_previous.rHundredths" is outside the permitted range. The value was not sent.

eLON_lrEnergy_Out_of_range: SNVT 200 / The input variable "stValue.lrEnergy" is outside the permitted range. The value was not sent.

eLON_rPowerFactor_Out_of_range: SNVT 200 / The input variable "stValue.rPowerFactor" is outside the permitted range. The value was not sent.

eLON_rPower_Out_of_range: SNVT 200 / The input variable "stValue.rPower" is outside the permitted range. The value was not sent.

eLON_rBallastTemp_Out_of_range: SNVT 200 / The input variable "stValue.rBallastTemp" is outside the permitted range. The value was not sent.

eLON_lrLongitude_Out_of_range: SNVT 201 / The input variable "stValue.lrLongitude" is outside the permitted range. The value was not sent.

eLON_lrLatitude_Out_of_range: SNVT 201 / The input variable "stValue.lrLatitude" is outside the permitted range. The value was not sent.
### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.2.1.2 E_LON_Parameter_Datatypes

**Enums SNVT types**

```plaintext
TYPE E_LON_Parameter_Datatypes :
{
  eEmpty             := 0,
  eSNVT_amp          := 1,
  eSNVT_amp_mil      := 2,
  eSNVT_angle        := 3,
  eSNVT_angle_vel    := 4,
  eSNVT_btu_kilo     := 5,
  eSNVT_btu_mega     := 6,
  eSNVT_char_ascii   := 7,
  eSNVT_count        := 8,
  eSNVT_count_inc    := 9,
  eSNVT_date_cal     := 10,
  eSNVT_date_day     := 11,
  eSNVT_date_time    := 12,
  eSNVT_elec_kwh     := 13,
  eSNVT_elec_whr     := 14,
  eSNVT_flow         := 15,
  eSNVT_flow_mil     := 16,
  eSNVT_length       := 17,
  eSNVT_length_kilo  := 18,
  eSNVT_length_micr  := 19,
  eSNVT_length_mil   := 20,
  eSNVT_lev_cont     := 21,
  eSNVT_lev_disc     := 22,
  eSNVT_mass         := 23,
  eSNVT_mass_kilo    := 24,
  eSNVT_mass_mega    := 25,
  eSNVT_mass_mil     := 26,
  eSNVT_power        := 27,
  eSNVT_power_kilo   := 28,
  eSNVT_ppm          := 29,
  eSNVT_press        := 30,
  eSNVT_res          := 31,
  eSNVT_res_kilo     := 32,
  eSNVT_sound_db     := 33,
  eSNVT_speed        := 34,
  eSNVT_speed_mil    := 35,
  eSNVT_str_asc      := 36,
  eSNVT_str_int      := 37,
  eSNVT_telcom       := 38,
  eSNVT_temp         := 39,
  eSNVT_time_passed  := 40,
  eSNVT_vol          := 41,
  eSNVT_vol_kilo     := 42,
  eSNVT_vol_mil      := 43,
  eSNVT_volt         := 44,
  eSNVT_volt_dbmv    := 45,
  eSNVT_volt_kilo    := 46,
  eSNVT_volt_mil     := 47,
  eSNVT_amp_f        := 48,
  eSNVT_angle_f      := 49,
  eSNVT_angle_vel_f  := 50,
  eSNVT_count_f      := 51,
  eSNVT_count_inc_f  := 52,
  eSNVT_flow_f       := 53,
  eSNVT_length_f     := 54,
  eSNVT_lev_cont_f   := 55,
  eSNVT_mass_f       := 56,
  eSNVT_power_f      := 57,
  eSNVT_ppm_f        := 58,
  eSNVT_press_f      := 59,
  eSNVT_res_f        := 60,
  eSNVT_sound_db_f   := 61,
  eSNVT_speed_f      := 62,
  eSNVT_temp_f       := 63,
  eSNVT_time_f       := 64,
}
```
```
:::eSNVT_vol_f := 65,
:::eSNVT_volt_f := 66,
:::eSNVT_btu_f := 67,
:::eSNVT_elec_whr_f := 68,
:::eSNVT_config_src := 69,
:::eSNVT_color := 70,
:::eSNVT_grammage := 71,
:::eSNVT_grammage_f := 72,
:::eSNVT_file_req := 73,
:::eSNVT_file_status := 74,
:::eSNVT_freq_f := 75,
:::eSNVT_freq_hz := 76,
:::eSNVT_freq_kilohz := 77,
:::eSNVT_freq_milhz := 78,
:::eSNVT_lux := 79,
:::eSNVT_ISO_7811 := 80,
:::eSNVT_lev_percent := 81,
:::eSNVT_multiplier := 82,
:::eSNVT_state := 83,
:::eSNVT_time_stamp := 84,
:::eSNVT_zerospan := 85,
:::eSNVT_magcard := 86,
:::eSNVT_elapsed_tm := 87,
:::eSNVT_alarm := 88,
:::eSNVT_currency := 89,
:::eSNVT_file_pos := 90,
:::eSNVT_multiplier := 91,
:::eSNVT_obj_request := 92,
:::eSNVT_obj_status := 93,
:::eSNVT_preset := 94,
:::eSNVT_switch := 95,
:::eSNVT_trans_table := 96,
:::eSNVT_override := 97,
:::eSNVT_pwr_fact := 98,
:::eSNVT_pwr_fact_f := 99,
:::eSNVT_density := 100,
:::eSNVT_density_f := 101,
:::eSNVT_rpm := 102,
:::eSNVT_hvac_emerg := 103,
:::eSNVT_angle_deg := 104,
:::eSNVT_temp_p := 105,
:::eSNVT_temp_setpt := 106,
:::eSNVT_time_sec := 107,
:::eSNVT_hvac_mode := 108,
:::eSNVT_occupancy := 109,
:::eSNVT_area := 110,
:::eSNVT_hvac_overid := 111,
:::eSNVT_hvac_status := 112,
:::eSNVT_press_p := 113,
:::eSNVT_address := 114,
:::eSNVT_scene := 115,
:::eSNVT_scene_cfg := 116,
:::eSNVT_setting := 117,
:::eSNVT_evap_state := 118,
:::eSNVT_therm_mode := 119,
:::eSNVT_defr_mode := 120,
:::eSNVT_defr_term := 121,
:::eSNVT_defr_state := 122,
:::eSNVT_time_min := 123,
:::eSNVT_time_hour := 124,
:::eSNVT_ph := 125,
:::eSNVT_ph_f := 126,
:::eSNVT_chlr_status := 127,
:::eSNVT_tod_event := 128,
:::eSNVT smo obscur := 129,
:::eSNVT_fire_test := 130,
:::eSNVT_temp_ror := 131,
:::eSNVT_fire_init := 132,
:::eSNVT_fire_indcte := 133,
:::eSNVT_time_zone := 134,
:::eSNVT_earth_pos := 135,
:::eSNVT_reg_val := 136,
:::eSNVT_reg_val_ts := 137,
:::eSNVT_volt_ac := 138,
:::eSNVT_amps_ac := 139,
:::eSNVT_turbidity := 143,
:::eSNVT_turbidity_f := 144,
:::eSNVT_hvac_type := 145,
:::eSNVT_elec_kwh_l := 146,
```
eSNVT_temp_diff_p := 147,
eSNVT_ctrl_req := 148,
eSNVT_ctrl_resp := 149,
eSNVT_ptz := 150,
eSNVT_privacyzone := 151,
eSNVT_pos_ctrl := 152,
eSNVT_enthalpy := 153,
eSNVT_gfci_status := 154,
eSNVT_motor_state := 155,
eSNVT_pumpset_mm := 156,
eSNVT_ex_control := 157,
eSNVT_pumpset_sn := 158,
eSNVT_pump_sensor := 159,
eSNVT_abs_humid := 160,
eSNVT_flow_p := 161,
eSNVT_dev_c_mode := 162,
eSNVT_valve_mode := 163,
eSNVT_alarm_2 := 164,
eSNVT_state_64 := 165,
eSNVT_nv_type := 166,
eSNVT_ent_opmode := 168,
eSNVT_ent_state := 169,
eSNVT_flow_dir := 171,
eSNVT_hvac_satsts := 172,
eSNVT_dev_status := 173,
eSNVT_dev_fault := 174,
eSNVT_dev_maint := 175,
eSNVT_date_event := 176,
eSNVT_sched_val := 177,
eSNVT_out_state := 178,
eSNVT_out_status := 179,
eSNVT_sblnd_state := 180,
eSNVT_rac_ctrl := 181,
eSNVT_rac_req := 182,
eSNVT_count_32 := 183,
eSNVT_clothes_w_c := 184,
eSNVT_clothes_w_m := 185,
eSNVT_clothes_w_s := 186,
eSNVT_clothes_w_a := 187,
eSNVT_multiplier_s := 188,
eSNVT_switch_2 := 189,
eSNVT_color_2 := 190,
eSNVT_log_status := 191,
eSNVT_time_hour_p := 192,
eSNVT_log_fx_request := 193,
eSNVT_log_fx_status := 194,
eSNVT_log_request := 195,
eSNVT_enthalpy_d := 196,
eSNVT_amp_ac_mil := 197,
eSNVT_time_hour := 198,
eSNVT_lamp_status := 199,
eSNVT_environment := 200,
eSNVT_geo_loc := 201
)
END_TYPE

eEmpty:
eSNVT_amp: SNVT_amp
eSNVT_amp_mil: SNVT_amp_mil
eSNVT_angle: SNVT_angle
eSNVT_angle_vel: SNVT_angle_vel
eSNVT_btu_kilo: SNVT_btu_kilo
eSNVT_btu_mega: SNVT_btu_mega
eSNVT_char_ascii: SNVT_char_ascii
eSNVT_count: SNVT_count
eSNVT_count_inc: SNVT_count_inc
eSNVT_date_cal: SNVT_date_cal
eSNVT_date_day: SNVT_date_day
eSNVT_date_time: SNVT_date_time
eSNVT_elec_kwh: SNVT_elec_kwh
eSNVT_elec_whr: SNVT_elec_whr
eSNVT_flow: SNVT_flow
eSNVT_flow_mil: SNVT_flow_mil
eSNVT_length: SNVT_length
eSNVT_length_kilo: SNVT_length_kilo
eSNVT_length_micr: SNVT_length_micr
eSNVT_length_mil: SNVT_length_mil
eSNVT_lev_cont: SNVT_lev_cont
eSNVT_lev_disc: SNVT_lev_disc
eSNVT_mass: SNVT_mass
eSNVT_mass_kilo: SNVT_mass_kilo
eSNVT_mass_mega: SNVT_mass_mega
eSNVT_mass_mil: SNVT_mass_mil
eSNVT_power: SNVT_power
eSNVT_power_kilo: SNVT_power_kilo
eSNVT_ppm: SNVT_ppm
eSNVT_press: SNVT_press
eSNVT_res: SNVT_res
eSNVT_res_kilo: SNVT_res_kilo
eSNVT_sound_db: SNVT_sound_db
eSNVT_speed: SNVT_speed
eSNVT_speed_mil: SNVT_speed_mil
eSNVT_str_asc: SNVT_str_asc
eSNVT_str_int: SNVT_str_int
eSNVT_telcom: SNVT_telcom
eSNVT_temp: SNVT_temp
eSNVT_time_passed: SNVT_time_passed
eSNVT_vol: SNVT_vol
eSNVT_vol_kilo: SNVT_vol_kilo
eSNVT_vol_mil: SNVT_vol_mil
eSNVT_volt: SNVT_volt
eSNVT_volt_dbmv: SNVT_volt_dbmv
eSNVT_volt_kilo: SNVT_volt_kilo
eSNVT_volt_mil: SNVT_volt_mil
eSNVT_amp_f: SNVT_amp_f
eSNVT_angle_f: SNVT_angle_f
eSNVT_angle_vel_f: SNVT_angle_vel_f
eSNVT_count_f: SNVT_count_f
eSNVT_count_inc_f: SNVT_count_inc_f
eSNVT_flow_f: SNVT_flow_f
eSNVT_length_f: SNVT_length_f
eSNVT_lev_cont_f: SNVT_lev_cont_f
eSNVT_mass_f: SNVT_mass_f
eSNVT_power_f: SNVT_power_f
eSNVT_ppm_f: SNVT_ppm_f
eSNVT_press_f: SNVT_press_f
eSNVT_res_f: SNVT_res_f
eSNVT_sound_db_f: SNVT_sound_db_f
eSNVT_speed_f: SNVT_speed_f
eSNVT_temp_f: SNVT_temp_f
eSNVT_time_f: SNVT_time_f
eSNVT_vol_f: SNVT_vol_f
eSNVT_volt_f: SNVT_volt_f
eSNVT_btu_f: SNVT_btu_f
eSNVT_elec_whr_f: SNVT_elec_whr_f
eSNVT_config_src: SNVT_config_src
eSNVT_color: SNVT_color
eSNVT_grammage: SNVT_grammage
eSNVT_grammage_f: SNVT_grammage_f
eSNVT_file_req: SNVT_file_req
eSNVT_file_status: SNVT_file_status
eSNVT_freq_f: SNVT_freq_f
eSNVT_freq_hz: SNVT_freq_hz
eSNVT_freq_kilohz: SNVT_freq_kilohz
eSNVT_freq_milhz: SNVT_freq_milhz
eSNVT_lux: SNVT_lux
eSNVT_ISO_7811: SNVT_ISO_7811
eSNVT_lev_percent: SNVT_lev_percent
eSNVT_multiplier: SNVT_multiplier
eSNVT_state: SNVT_state
eSNVT_time_stamp: SNVT_time_stamp
eSNVT_zerospan: SNVT_zerospan
eSNVT_magcard: SNVT_magcard
eSNVT_elapsed_tm: SNVT_elapsed_tm
eSNVT_alarm: SNVT_alarm
eSNVT_currency: SNVT_currency
eSNVT_file_pos: SNVT_file_pos
eSNVT_muldiv: SNVT_muldiv
eSNVT_obj_request: SNVT_obj_request
eSNVT_obj_status: SNVT_obj_status
eSNVT_preset: SNVT_preset
eSNVT_switch: SNVT_switch
eSNVT_trans_table: SNVT_trans_table
eSNVT_override: SNVT_override
eSNVT_pwr_fact: SNVT_pwr_fact
eSNVT_pwr_fact_f: SNVT_pwr_fact_f
eSNVT_density: SNVT_density
eSNVT_density_f: SNVT_density_f
eSNVT_rpm: SNVT_rpm
eSNVT_hvac_emerg: SNVT_hvac_emerg
eSNVT_angle_deg: SNVT_angle_deg
eSNVT_temp_p: SNVT_temp_p
eSNVT_temp_setpt: SNVT_temp_setpt
eSNVT_time_sec: SNVT_time_sec
eSNVT_hvac_mode: SNVT_hvac_mode
eSNVT_occupancy: SNVT_occupancy
eSNVT_area: SNVT_area
eSNVT_hvac_overid: SNVT_hvac_overid
eSNVT_hvac_status: SNVT_hvac_status
eSNVT_press_p: SNVT_press_p
eSNVT_address: SNVT_address
eSNVT_scene: SNVT_scene
eSNVT_scene_cfg: SNVT_scene_cfg
eSNVT_setting: SNVT_setting
eSNVT_evap_state: SNVT_evap_state
eSNVT_therm_mode: SNVT_therm_mode
eSNVT_defr_mode: SNVT_defr_mode
eSNVT_defr_term: SNVT_defr_term
eSNVT_defr_state: SNVT_defr_state
eSNVT_time_min: SNVT_time_min
eSNVT_time_hour: SNVT_time_hour
eSNVT_ppm: SNVT_ph
eSNVT_ph_f: SNVT_ph_f
eSNVT_chlr_status: SNVT_chlr_status
eSNVT_tod_event: SNVT_tod_event
eSNVT_smo_obscur: SNVT_smo_obscur
eSNVT_fire_test: SNVT_fire_test
eSNVT_temp_ror: SNVT_temp_ror
eSNVT_fire_init: SNVT_fire_init
eSNVT_fire_indcte: SNVT_fire_indcte
eSNVT_time_zone: SNVT_time_zone
eSNVT_earth_pos: SNVT_earth_pos
eSNVT_reg_val: SNVT_reg_val
eSNVT_reg_val_ts: SNVT_reg_val_ts
eSNVT_volt_ac: SNVT_volt_ac
eSNVT_amp_ac: SNVT_amp_ac
eSNVT_turbidity: SNVT_turbidity
eSNVT_turbidity_f: SNVT_turbidity_f
eSNVT_hvac_type: SNVT_hvac_type
eSNVT_elec_kwh_l: SNVT_elec_kwh_l
eSNVT_temp_diff_p: SNVT_temp_diff_p
eSNVT_ctrl_req: SNVT_ctrl_req
eSNVT_ctrl_resp: SNVT_ctrl_resp
eSNVT_ptz: SNVT_ptz
eSNVT_privacyzone: SNVT_privacyzone
eSNVT_pos_ctrl: SNVT_pos_ctrl
eSNVT_enthalpy: SNVT_enthalpy
eSNVT_gfci_status: SNVT_gfci_status
eSNVT_motor_state: SNVT_motor_state
eSNVT_pumpset_mn: SNVT_pumpset_mn
eSNVT_ex_control: SNVT_ex_control
eSNVT_pumpset_sn: SNVT_pumpset_sn
eSNVT_pump_sensor: SNVT_pump_sensor
eSNVT_abs_humid: SNVT_abs_humid
eSNVT_flow_p: SNVT_flow_p
eSNVT_dev_c_mode: SNVT_dev_c_mode
eSNVT_valve_mode: SNVT_valve_mode
eSNVT_alarm_2: SNVT_alarm_2
eSNVT_state_64: SNVT_state_64
eSNVT_nv_type: SNVT_nv_type
eSNVT_ent_opmode: SNVT_ent_opmode
eSNVT_ent_state: SNVT_ent_state
eSNVT_ent_status: SNVT_ent_status
eSNVT_flow_dir: SNVT_flow_dir
eSNVT_hvac_satsts: SNVT_hvac_satsts
eSNVT_dev_status: SNVT_dev_status
eSNVT_dev_fault: SNVT_dev_fault
eSNVT_dev_maint: SNVT_dev_maint
eSNVT_date_event: SNVT_date_event
eSNVT_sched_val: SNVT_sched_val
eSNVT_sec_state: SNVT_sec_state
eSNVT_sec_status: SNVT_sec_status
eSNVT_sblnd_state: SNVT_sblnd_state
eSNVT_rac_ctrl: SNVT_rac_ctrl
eSNVT_rac_req: SNVT_rac_req
eSNVT_count_32: SNVT_count_32
eSNVT_clothes_w_c: SNVT_clothes_w_c
eSNVT_clothes_w_m: SNVT_clothes_w_m
eSNVT_clothes_w_s: SNVT_clothes_w_s
eSNVT_clothes_w_a: SNVT_clothes_w_a
eSNVT_multiplier_s: SNVT_multiplier_s
eSNVT_switch_2: SNVT_switch_2
eSNVT_color_2: SNVT_color_2
eSNVT_log_status: SNVT_log_status
eSNVT_time_stamp_p: SNVT_time_stamp_p
eSNVT_log_fx_request: SNVT_log_fx_request
**eSNVT_log_fx_status:** SNVT_log_fx_status

**eSNVT_log_request:** SNVT_log_request

**eSNVT_enthalpy_d:** SNVT_enthalpy_d

**eSNVT_amp_ac_mil:** SNVT_amp_ac_mil

**eSNVT_time_hour_p:** SNVT_time_hour_p

**eSNVT_lamp_status:** SNVT_lamp_status

**eSNVT_environment:** SNVT_environment

**eSNVT_geo_loc:** SNVT_geo_loc

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.2.1.3 E_LON_alarm_type_t**

*Used by: SNVT_alarm / SNVT_alarm_2*

```
TYPE E_LON_alarm_type_t : 
{
  eLON_AL_HEADER := -13,
  eLON_AL_FOOTER := -12,
  eLON_AL_DEBUG := -11,
  eLON_AL_INFO := -10,
  eLON_AL_SYSTEM_INFO := -6,
  eLON_AL_VALUE_INVALID := -5,
  eLON_AL_CONSTANT := -4,
  eLON_AL_OFFLINE := -3,
  eLON_AL_UNKNOWN := -2,
  eLON_AL_NUL := -1,
  eLON_AL_NO_CONDITION := 0,
  eLON_AL_ALM_CONDITION := 1,
  eLON_AL_TOT_SVC_ALM_1 := 2,
  eLON_AL_TOT_SVC_ALM_2 := 3,
  eLON_AL_TOT_SVC_ALM_3 := 4,
  eLON_AL_LOW_LMT_CLR_1 := 5,
  eLON_AL_LOW_LMT_CLR_2 := 6,
  eLON_AL_HIGH_LMT_CLR_1 := 7,
  eLON_AL_HIGH_LMT_CLR_2 := 8,
  eLON_AL_LOW_LMT_ALM_1 := 9,
  eLON_AL_LOW_LMT_ALM_2 := 10,
  eLON_AL_HIGH_LMT_ALM_1 := 11,
  eLON_AL_HIGH_LMT_ALM_2 := 12,
  eLON_AL_FIR_ALM := 13,
  eLON_AL_FIR_PRE_ALM := 14,
  eLON_AL_FIR_TRBL := 15,
  eLON_AL_FIR_SUPV := 16,
  eLON_AL_FIR_TEST_ALM := 17,
  eLON_AL_FIR_TEST_PRE_ALM := 18,
  eLON_AL_FIR_ENVCOMP_MAX := 19,
  eLON_AL_FIR_MONITOR_COND := 20,
  eLON_AL_FIR_MAINT_ALERT := 21,
  eLON_AL_FATAL_ERROR := 30,
  eLON_AL_ERROR := 31,
  eLON_AL_WARNING := 32
}
END_TYPE
```

**eLON_AL_HEADER:** Update sequence header

**eLON_AL_FOOTER:** Update sequence footer

**eLON_AL_DEBUG:** Debug information (not an alarm)

**eLON_AL_INFO:** Information update (not an alarm)
eLON_AL_SYSTEM_INFO: System information (not an alarm)

eLON_AL_VALUE_INVALID: The value is invalid

eLON_AL_CONSTANT: The value is a constant value (not an alarm)

eLON_AL_OFFLINE: The device is offline

eLON_AL_UNKNOWN: Alarm condition unknown (may be due to a communication failure or hardware failure)

eLON_AL_NUL: Invalid alarm type value (alarm condition not specified)

eLON_AL_NO_CONDITION: No alarm condition present

eLON_AL_ALM_CONDITION: Unspecified alarm condition present

eLON_AL_TOT_SVC_ALM_1: Total/service interval alarm 1 (component requires service or maintenance)

eLON_AL_TOT_SVC_ALM_2: Total/service interval alarm 2

eLON_AL_TOT_SVC_ALM_3: Total/service interval alarm 3

eLON_AL_LOW_LMT_CLR_1: Alarm low limit alarm clear 1

eLON_AL_LOW_LMT_CLR_2: Alarm low limit alarm clear 2

eLON_AL_HIGH_LMT_CLR_1: Alarm high limit alarm clear 1

eLON_AL_HIGH_LMT_CLR_2: Alarm high limit alarm clear 2

eLON_AL_LOW_LMT_ALM_1: Alarm low limit alarm 1

eLON_AL_LOW_LMT_ALM_2: Alarm low limit alarm 2

eLON_AL_HIGH_LMT_ALM_1: Alarm high limit alarm 1

eLON_AL_HIGH_LMT_ALM_2: Alarm high limit alarm 2

eLON_AL_FIR_ALM: Fire alarm condition

eLON_AL_FIR_PRE_ALM: Fire pre-alarm condition

eLON_AL_FIR_TRBL: Fire-related trouble (fault) condition

eLON_AL_FIR_SUPV: Fire-related supervisory condition (e.g., sprinkler pressure)

eLON_AL_FIR_TEST_ALM: Fire-related test-mode alarm condition

eLON_AL_FIR_TEST_PRE_ALM: Fire-related test-mode pre-alarm condition

eLON_AL_FIR_ENVCOMP_MAX: Fire-related maximum environmental compensation level reached

eLON_AL_FIR_MONITOR_COND: Fire-related abnormal input condition

eLON_AL_FIR_MAINT_ALERT: Fire-related maintenance alert

eLON_AL_FATAL_ERROR: Fatal application error

eLON_AL_ERROR: Other error condition

eLON_AL_WARNING: Other warning condition

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.2.1.4 E_LON_appl_cwc_t

Used by: SNVT_clothes_w_c / SNVT_clothes_w_s

TYPE E_LON_appl_cwc_t :
{
  eLON_CWC_NUL := -1,
  eLON_CWC_WASH := 0,
  eLON_CWC_RINSE := 1,
  eLON_CWC_SPIN := 2,
  eLON_CWC_DRY := 3
}
END_TYPE

eLON_CWC_NUL: Invalid Value

eLON_CWC_WASH: Wash

eLON_CWC_RINSE: Rinse

eLON_CWC_SPIN: Spin

eLON_CWC_DRY: Dry

Requirements

Development environment | required TC3 PLC library
------------------------|--------------------------
TwinCAT from v3.1.4020.14 | Tc2_LON from 3.3.4.0

4.2.1.5 E_LON_appl_cwp_t

Used by: SNVT_clothes_w_c

TYPE E_LON_appl_cwp_t :
{
  eLON_CWP_NUL := -1,
  eLON_CWP_GENERAL := 0,
  eLON_CWP_BOIL := 1,
  eLON_CWP_FAST_WASH := 2,
  eLON_CWP_LINGERIE := 3,
  eLON_CWP_WOOL := 4,
  eLON_CWP_TOWEL := 5,
  eLON_CWP_BED_LINENS := 6,
  eLON_CWP_CURTAIN := 7,
  eLON_CWP_RINSE_SPIN_ONLY := 8,
  eLON_CWP_DELICATE_RINSE := 9,
  eLON_CWP_SPIN_ONLY := 10,
  eLON_CWP_DRY_ONLY := 11
}
END_TYPE

eLON_CWP_NUL: Invalid Value

eLON_CWP_GENERAL: Normal Wash

eLON_CWP_BOIL: Boil

eLON_CWP_FAST_WASH: Fast Wash

eLON_CWP_LINGERIE: Lingerie

eLON_CWP_WOOL: Wool

eLON_CWP_TOWEL: Towel

eLON_CWP_BED_LINENS: Bed Linens

eLON_CWP_CURTAIN: Curtain

eLON_CWP_RINSE_SPIN_ONLY: Rinse and Spin Only

eLON_CWP_DELICATE_RINSE: Delicate Rinse
eLON_CWP_SPIN_ONLY: Spin Only

eLON_CWP_DRY_ONLY: Dry Only

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.1.6 E_LON_appl_cws_t

Used by: SNVT_clothes_w_c / SNVT_clothes_w_s

TYPE E_LON_appl_cws_t :
{
  eLON_CWS_NUL := -1,
  eLON_CWS_LOAD_SENSING := 0,
  eLON_CWS_WETTING := 1,
  eLON_CWS_DETERGENT := 2,
  eLON_CWS_WASHING := 3,
  eLON_CWS_WATERING := 4,
  eLON_CWS_RINSING := 5,
  eLON_CWS_ARRANGING := 6,
  eLON_CWS_DRAIN := 7,
  eLON_CWS_SPINNING := 8,
  eLON_CWS_FINAL_SPINNING := 9,
  eLON_CWS_FLUFFING := 10,
  eLON_CWS_DRYING := 11,
  eLON_CWS_COOLING := 12
}
END_TYPE

eLON_CWS_NUL: Invalid Value
eLON_CWS_LOAD_SENSING: Sensing Load
eLON_CWS_WETTING: Wetting
eLON_CWS_DETERGENT: Detergent
eLON_CWS_WASHING: Washing
eLON_CWS_WATERING: Watering
eLON_CWS_RINSING: Rinsing
eLON_CWS_ARRANGING: Arranging
eLON_CWS_DRAIN: Drain
eLON_CWS_SPINNING: Spinning
eLON_CWS_FINAL_SPINNING: In Final Spin
eLON_CWS_FLUFFING: Fluffling
eLON_CWS_DRYING: Drying
eLON_CWS_COOLING: Cooling

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.2.1.7  E_LON_appl_rin_t

Used by: SNVT_clothes_w_c

TYPE E_LON_appl_rin_t :
{
  eLON_RIN_NUL := -1,
  eLON_RIN_PRE_WASH := 0,
  eLON_RIN_WATER_PLUS := 1,
  eLON_RIN_DETERGENT_PLUS := 2,
  eLON_RIN_RINSE_HOLD := 3
}
END_TYPE

eLON_RIN_NUL: Invalid Value

eLON_RIN_PRE_WASH: Pre-wash

eLON_RIN_WATER_PLUS: Water Plus

eLON_RIN_DETERGENT_PLUS: Detergent Plus

eLON_RIN_RINSE_HOLD: Rinse Hold

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.1.8  E_LON_boolean_t

Used by: SCPTautoAnswer / SCPTcoolingResetEnable / SCPTdefrostHold / SCPTdefrostInternalSchedule / SCPTheatingResetEnable / SCPThighLimit1Enable / SCPTlowLimit1Enable / SCPTlowLimit2Enable / SCPTscheduleInternal / SNVT_clothes_w_c / SNVT_pump_sensor / SNVT_pumpset_mn / SNVT_pumpset_sn

TYPE E_LON_boolean_t :
{
  eLON_BOOL_NUL := -1,
  eLON_BOOL_FALSE := 0,
  eLON_BOOL_TRUE := 1
}
END_TYPE

eLON_BOOL_NUL: Invalid Value

eLON_BOOL_FALSE: False

eLON_BOOL_TRUE: True

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.1.9  E_LON_calendar_type_t

Used by: SNVT_time_zone

TYPE E_LON_calendar_type_t :
{
  eLON_CAL_NUL := -1,
  eLON_CAL_GREG := 0,
  eLON_CAL_JUL := 1,
  eLON_CAL_MEU := 2
}
END_TYPE
eLON_CAL_NUL: Invalid Value
eLON_CAL_GREG: Gregorian calendar
eLON_CAL_JUL: Julian calendar
eLON_CAL_MEU: Calendar Method European/US "MEU"

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.1.10 E_LON_cam_act_t

Used by: SNVT_pos_ctrl

```
TYPE E_LON_cam_act_t :
{
  eLON_CMA_NUL := -1,
  eLON_CMA_SAVE := 0,
  eLON_CMA_CALL := 1,
  eLON_CMA_READ := 2
}
END_TYPE
```

eLON_CMA_NUL: Invalid action call response
eLON_CMA_SAVE: Save the values defined by the function
eLON_CMA_CALL: Preposition tour tables
eLON_CMA_READ: Absolute positions

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.1.11 E_LON_cam_func_t

Used by: SNVT_pos_ctrl

```
TYPE E_LON_cam_func_t :
{
  eLON_CMF_NUL := -1,
  eLON_CMF_REL := 0,
  eLON_CMF_TOUR := 1,
  eLON_CMF_ABS := 2
}
END_TYPE
```

eLON_CMF_NUL: Invalid function call response
eLON_CMF_REL: Relative positions, prepositions
eLON_CMF_TOUR: Preposition tour tables
eLON_CMF_ABS: Absolute positions

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.2.1.12  **E_LON_chiller_t**

Used by: SNVT_chlr_status

```c
TYPE E_LON_chiller_t : 
{
    eLON_CHLR_NUL := -1,
    eLON_CHLR_OFF := 0,
    eLON_CHLR_START := 1,
    eLON_CHLR_RUN := 2,
    eLON_CHLR_PRESHUTDN := 3,
    eLON_CHLR_SERVICE := 4
}
END_TYPE
```

eLON_CHLR_NUL: Invalid Value
eLON_CHLR_OFF: Chiller off
eLON_CHLR_START: Chiller in start mode
eLON_CHLR_RUN: Chiller in run mode
eLON_CHLR_PRESHUTDN: Chiller in pre shutdown mode
eLON_CHLR_SERVICE: Chiller in service mode

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.1.13  **E_LON_color_encoding_t**

Used by: SNVT_color_2

```c
TYPE E_LON_color_encoding_t :
{
    eLON_COLOR_NUL := -1,
    eLON_COLOR_CIE31_LUMEN := 0,
    eLON_COLOR_CIE31_PERCENT := 1,
    eLON_COLOR_RGB := 2,
    eLON_COLOR_TEMPERATURE := 3
}
END_TYPE
```

eLON_COLOR_NUL: Invalid value
eLON_COLOR_CIE31_LUMEN: CIE 1931 color space; Y output in lumen
eLON_COLOR_CIE31_PERCENT: CIE 1931 color space; Y output in percent of maximum lumen output of the lamp
eLON_COLOR_RGB: No color space, RGB color value
eLON_COLOR_TEMPERATURE: Color temperature

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.1.14  **E_LON_config_source_t**

Used by: SNVT_config_src
TYPE E_LON_config_source_t :
{
    eLON_CFG_NUL := -1,
    eLON_CFG_LOCAL := 0,
    eLON_CFG_EXTERNAL := 1
}
END_TYPE

eLON_CFG_NUL: Invalid Value

eLON_CFG_LOCAL: Device will use self-installation functions to set its own network image

eLON_CFG_EXTERNAL: Device's network image will be set by an outside source

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.1.15 E_LON_control_resp_t

Used by: SNVT_ctrl_resp

TYPE E_LON_control_resp_t :
{
    eLON_CTRLR_NUL := -1,
    eLON_CTRLR_NO := 0,
    eLON_CTRLR_PEND := 1,
    eLON_CTRLR_REL := 2,
    eLON_CTRLR_QUERY := 3,
    eLON_CTRLR_RES := 4,
    eLON_CTRLR_ERR := 5
}
END_TYPE

eLON_CTRLR_NUL: Invalid value

eLON_CTRLR_NO: Number of current controller

eLON_CTRLR_PEND: Request pending due to control query to current operator

eLON_CTRLR_REL: Current control released

eLON_CTRLR_QUERY: Query to current controller

eLON_CTRLR_RES: Controllable device has been reset

eLON_CTRLR_ERR: Error in control

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.1.16 E_LON_currency_t

Used by: SNVT_currency

TYPE E_LON_currency_t :
{
    eLON_CU_NUL := -1,
    eLON_CU_ARGENTINA_PESO := 0,
    eLON_CU_AUSTRALIA_DOLLAR := 1,
    eLON_CU_AUSTRIA_SCHILLING := 2,
    eLON_CU_BAHRAIN_DINAR := 3,
    eLON_CU_BELGIUM_FRANC := 4,
    eLON_CU_BRAZIL_CRUZEIRO_REAL := 5,
    eLON_CU_BRITAIN_POUND := 6,
    eLON_CU_CANADA_DOLLAR := 7,


```plaintext
eLON_CU_CZECH_KORUNA := 8,
eLON_CU_CHILE_PESO := 9,
eLON_CU_CHINA_RENMINBI := 10,
eLON_CU_COLOMBIA_PESO := 11,
eLON_CU_DENMARK_KRONE := 12,
eLON_CU_ECUADOR_SUCRE := 13,
eLON_CU_EUROPEAN_CURRENCY_UNIT := 14,
eLON_CU_FINLAND_MARKKA := 15,
eLON_CU_FRANCE_FRANC := 16,
eLON_CU_GERMANY_MARK := 17,
eLON_CU_GREECE_DRACHMA := 18,
eLON_CU_HONG_KONG_DOLLAR := 19,
eLON_CU_HUNGARY_FORINT := 20,
eLON_CU_INDIA_RUPEE := 21,
eLON_CU_INDONESIA_RUPIAH := 22,
eLON_CU_IRLAEAND_PUNT := 23,
eLON_CU_ISRAEL_SHEKEL := 24,
eLON_CU_ITALY_LIRA := 25,
eLON_CU_JAPAN_YEN := 26,
eLON_CU_JORDAN_DINAR := 27,
eLON_CU_KUWAIT_DINAR := 28,
eLON_CU_LEBANON_POUND := 29,
eLON_CU_MALAYSIA_RINGGIT := 30,
eLON_CU_MALTA_LIRA := 31,
eLON_CU_MEXICO_PESO := 32,
eLON_CU_NETHERLANDS_GUILDER := 33,
eLON_CU_NEW_ZEALAND_DOLLAR := 34,
eLON_CU_NORWAY_KRONE := 35,
eLON_CU_PAKISTAN_RUPEE := 36,
eLON_CU_PERU_NEW_SOL := 37,
eLON_CU_PHILIPPINES_PESO := 38,
eLON_CU_POLAND_ZLOTY := 39,
eLON_CU_PORTUGAL_ESCUDO := 40,
eLON_CU_SAUDI_ARABIA_RIYAL := 41,
eLON_CU_SINGAPORE_DOLLAR := 42,
eLON_CU_SLOVAK_KORUNA := 43,
eLON_CU_SOUTH_AFRICA Rand := 44,
eLON_CU_SOUTH_KOREA WON := 45,
eLON_CU_SPAIN_PESETA := 46,
eLON_CU_SPECIAL_DRAWING_RIGHTS := 47,
eLON_CU_SWEDEN_KRONA := 48,
eLON_CU_SWITZERLAND_FRENCH := 49,
eLON_CU_TAIWAN_DOLLAR := 50,
eLON_CU_THAILAND_BAHT := 51,
eLON_CU_TURKEY_LIRA := 52,
eLON_CU_UNITED_ARAB_DIRHAM := 53,
eLON_CU_UNITED_STATES_DOLLAR := 54,
eLON_CU URUGUAY NEW PESO := 55,
eLON_CU/VENEZUELA_BOLIVAR := 56
\)

END_TYPE

eLON_CU_NUL: Invalid Value

eLON_CU_ARGENTINA_PESO: Argentine Peso

eLON_CU_AUSTRALIA_DOLLAR: Australian Dollar

eLON_CU_AUSTRIA_SCHILLING: Austrian Schilling

eLON_CU_BAHRAIN_DINAR: Bahraini Dinar

eLON_CU_BELGIUM_SCHILLING: Belgian Franc

eLON_CU_BRAZIL_CRUZERIO_REAL: Brazilian Cruzeiro Real

eLON_CU_BRITAIN_POUND: British Pound

eLON_CU_CANADA_DOLLAR: Canadian Dollar

eLON_CU_CZECH_KORUNA: Czechoslovakian Koruna

eLON_CU_CHILE_PESO: Chilean Peso

eLON_CU_CHINA_RENMINBI: Chinese Renminbi Yuan

eLON_CU_COLOMBIA_PESO: Colombian Peso
```
eLON_CU_DENMARK_KRONE: Danish Krone
eLON_CU_ECUADOR_SUCRE: Ecuadorian Sucre
eLON_CU EUROPEAN_CURRENCY_UNIT: European Euro
eLON_CU_FINLAND_MARKKA: Finnish Markka
eLON_CU_FRANCE.FRANC: French Franc
eLON_CU_GERMANY_MARK: German Mark
eLON_CU_GREECE_DRAHAM: Greek Drachma
eLON_CU_HONG_KONG_DOLLAR: Hong Kong Dollar
eLON_CU_HUNGARY_FORINT: Hungarian Forint
eLON_CU_INDIA_RUPEE: Indian Rupee
eLON_CU_INDONESIA_RUPIAH: Indonesian Rupiah
eLON_CU_IRELAND_PUNT: Irish Punt
eLON_CU_ISRAEL_SHEKEL: Israeli Shekel
eLON_CU_ITALY_LIRA: Italian Lira
eLON_CU_JAPAN_YEN: Japanese Yen
eLON_CU_JORDAN_DINAR: Jordanian Dinar
eLON_CU_KUWAIT_DINAR: Kuwaiti Dinar
eLON_CU_LEBANON_POUND: Lebanese Pound
eLON_CU_MALAYSIA_RINGGIT: Malaysian Ringgit
eLON_CU_MALTA_LIRA: Maltese Lira
eLON_CU_MEXICO_PESO: Mexican New Peso
eLON_CU_NETHERLANDS_GUILDER: Netherlands Guilder
eLON_CU_NEW_ZEALAND_DOLLAR: New Zealand Dollar
eLON_CU_NORWAY_KRONE: Norwegian Krone
eLON_CU_PAKISTAN_RUPEE: Pakistani Rupee
eLON_CU_PERU_NEW_SOL: Peruvian New Sol
eLON_CU_PHILIPPINES_PESO: Philippine Peso
eLON_CU_POLAND_ZLOTY: Polish Zloty
eLON_CU_PORTUGAL_ESCU: Portuguese Escudo
eLON_CU_SAUDI_ARABIA_RIYAL: Saudi Arabian Riyal
eLON_CU_SINGAPORE_DOLLAR: Singaporean Dollar
eLON_CU_SLOVAK_KORUNA: Slavic Koruna
eLON_CU_SOUTH_AFRICA_RAND: South African Rand
eLON_CU_SOUTH_KOREA_WON: South Korean Won
eLON_CU_SPAIN_PESETA: Spanish Peseta
eLON_CU_SPECIAL_DRAWING_RIGHTS: international governmental exchange
Programming

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.1.17 E_LON_days_of_week_t

Used by: SCPTtimePeriod / SNVT_date_day / SNVT_time_zone

```c
TYPE E_LON_days_of_week_t :
{
    eLON_DAY_NUL := -1,
    eLON_DAY_SUN := 0,
    eLON_DAY_MON := 1,
    eLON_DAY_TUE := 2,
    eLON_DAY_WED := 3,
    eLON_DAY_THU := 4,
    eLON_DAY_FRI := 5,
    eLON_DAY_SAT := 6
}
END_TYPE
```

eLON_DAY_NUL: Invalid Value
eLON_DAY_SUN: Sunday
eLON_DAY_MON: Monday
eLON_DAY_TUE: Tuesday
eLON_DAY_WED: Wednesday
eLON_DAY_THU: Thursday
eLON_DAY_FRI: Friday
eLON_DAY_SAT: Saturday

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.1.18 E_LON_defrost_mode_t

Used by: SNVT_defr_mode

```c
TYPE E_LON_defrost_mode_t :
{
    eLON_DFM_NUL := -1,
```
eLON_DFM_NUL: Invalid Value

eLON_DFM_MODE_AMBIENT: No forced heating required

eLON_DFM_MODE_FORCED: Start-up after defrost ignored

eLON_DFM_MODE_SYNC: Synchronized

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.1.19 E_LON_defrost_state_t

Used by: SNVT_defr_state

TYPE E_LON_defrost_state_t :
{
  eLON_DFS_NUL := -1,
  eLON_DFS_STANDBY := 0,
  eLON_DFS_PUMPDOWN := 1,
  eLON_DFS_DEFROST := 2,
  eLON_DFS_DRAINDOWN := 3,
  eLON_DFS_INJECT_DLY := 4
}

END_TYPE

eLON_DFS_NUL: Invalid Value

eLON_DFS_STANDBY: Defrost in standby

eLON_DFS_PUMPDOWN: Defrost in pump-down mode

eLON_DFS_DEFROST: In defrost mode

eLON_DFS_DRAINDOWN: Defrost in drain-down

eLON_DFS_INJECT_DLY: Defrost in injection delay

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.1.20 E_LON_defrost_term_t

Used by: SNVT_defr_term

TYPE E_LON_defrost_term_t :
{
  eLONT_DFT_NUL := -1,
  eLONT_DFT_TERM_TEMP := 0,
  eLONT_DFT_TERM_TIME := 1,
  eLONT_DFT_TERM_FIRST := 2,
  eLONT_DFT_TERM_LAST := 3,
  eLONT_DFT_TERM_SENSOR := 4,
  eLONT_DFT_TERM_DISCHARGE := 5,
  eLONT_DFT_TERM_RETURN := 6,
  eLONT_DFT_TERM_SW_OPEN := 7,
  eLONT_DFT_TERM_SW_CLOSE := 8,
}

END_TYPE
eLON_DFT_TERM_MANUF := 100
)
END_TYPE

eLON_DFT_NUL: Invalid Value
eLON_DFT_TERM_TEMP: Terminate on temperature
eLON_DFT_TERM_TIME: Terminate on time
eLON_DFT_TERM_FIRST: Terminate on first occurring
eLON_DFT_TERM_LAST: Terminate on last occurring
eLON_DFT_TERM_SENSOR: Terminate on sensor
eLON_DFT_TERM_DISCHARGE: Terminate on discharge
eLON_DFT_TERM_RETURN: Terminate on return
eLON_DFT_TERM_SW_OPEN: Terminate on "Switch Open"
eLON_DFT_TERM_SW_CLOSE: Terminate on "Switch Closed"
eLON_DFT_TERM_MANUF: Manufacturer-Defined termination state

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.1.21 E_LON_device_c_mode_t

Used by: SNVT_dev_c_mode

TYPE E_LON_device_c_mode_t :
{
    eLON_DCM_NUL := -1,
    eLON_DCM_SPEED_CONST := 0,
    eLON_DCM_PRESS_CONST := 1,
    eLON_DCM_PRESS_COMP := 2,
    eLON_DCM_FLOW_CONST := 3,
    eLON_DCM_FLOW_COMP := 4,
    eLON_DCM_TEMP_CONST := 5,
    eLON_DCM_TEMP_COMP := 6,
    eLON_DCM_PRESS_AUTO := 7,
    eLON_DCM_QUICK_OPEN := 20,
    eLON_DCM_LINEAR := 21,
    eLON_DCM_EQUAL_PERCENT := 22,
    eLON_DCM_QUADRATIC := 23,
    eLON_DCM_FREE_DEFINED := 24,
    eLON_DCM_ZWAY_VALUE := 27,
    eLON_DCM_MIXING_VALUE := 28,
    eLON_DCM_DIVERTING_VALUE := 29,
    eLON_DCM_INVFNC_QCK_OPN := 30,
    eLON_DCM_INVFNC_EQL_PERC := 31,
    eLON_DCM_INVFNC_QUAD := 32
}
END_TYPE

eLON_DCM_NUL: Invalid Value
eLON_DCM_SPEED_CONST:
eLON_DCM_PRESS_CONST:
eLON_DCM_PRESS_COMP:
eLON_DCM_FLOW_CONST:
eLON_DCM_FLOW_COMP:
### Requirements

**Development environment**
- TwinCAT from v3.1.4020.14
- Tc2_LON from 3.3.4.0

#### E_LON_device_select_t

**Used by:** SNVT_dev_fault / SNVT_dev_maint / SNVT_dev_status

```c
TYPE E_LON_device_select_t :
{
    eLON_DV_NUL      := -1,
    eLON_DV_PUMP_CTRL := 0,
    eLON_DV_VALVE_POS := 1
}
END_TYPE
```

- **eLON_DV_NUL:** Invalid value
- **eLON_DV_PUMP_CTRL:** Use union for SFPTpumpController values
- **eLON_DV_VALVE_POS:** Use union for SFPTvalvePositioner values

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

#### E_LON_discrete_levels_t

**Used by:** SNVT_clothes_w_c / SNVT_lev_disc

```c
TYPE E_LON_discrete_levels_t :
{
    eLON_ST_NUL  := -1,
    eLON_ST_OFF  := 0,
    eLON_ST_LOW  := 1,
    eLON_ST_MED  := 2,
    eLON_ST_HIGH := 3,
}
```

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.2.1.24 E_LON_emerg_t

Used by: SNVT_hvac_emerg

TYPE E_LON_emerg_t :
{
    eLON_EMERG_NUL := -1,
    eLON_EMERG_NORMAL := 0,
    eLON_EMERG_PRESSURIZE := 1,
    eLON_EMERG_DEPRESSURIZE := 2,
    eLON_EMERG_PURGE := 3,
    eLON_EMERG_SHUTDOWN := 4,
    eLON_EMERG_FIRE := 5
}
END_TYPE

eLON_EMERG_NUL: Invalid Value

eLON_EMERG_NORMAL: No emergency mode

eLON_EMERG_PRESSURIZE: Emergency pressurize mode

eLON_EMERG_DEPRESSURIZE: Emergency depressurize mode

eLON_EMERG_PURGE: Emergency purge mode

eLON_EMERG_SHUTDOWN: Emergency shutdown mode

eLON_EMERG_FIRE: Emergency fire mode

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.1.25 E_LON_ent_cmd_t

Used by: SNVT_ent_state

TYPE E_LON_ent_cmd_t :
{
    eLON_ES_NUL := -1,
    eLON_ES_UNDEFINED := 0,
    eLON_ES_OPEN_PULS := 1,
    eLON_ES_OPEN := 2,
    eLON_ES_CLOSE := 3,
    eLON_ES_STOP := 4,
}
END_TYPE
eLON_ES_STOP_RESUME := 5,
eLON_ES_ENTRY_REQ := 6,
eLON_ES_EXIT_REQ := 7,
eLON_ES_KEY_REQ := 8,
eLON_ES_SAFETY_EXT_REQ := 9,
eLON_ES_EMERGENCY_REQ := 10,
eLON_ES_UPDATE_STATE := 11,
eLON_ES_SAFETY_EXT_RESUME := 12,
eLON_ES_EMERGENCY_REQ := 13,
)
END_TYPE

eLON_ES_NUL: Invalid Value

eLON_ES_UNDEFINED: State is not yet defined

eLON_ES_OPEN_PULS: Open the device and close it when back in normal position

eLON_ES_OPEN: Open the device if not locked

eLON_ES_CLOSE: Close the device

eLON_ES_STOP: Stop the device

eLON_ES_STOP_RESUME: Continue after stop command

eLON_ES_ENTRY_REQ: Entry request, access in to the area

eLON_ES_EXIT_REQ: Exit request, access out from the area

eLON_ES_KEY_REQ: Exit request, access out from the area

eLON_ES_SAFETY_EXT_REQ: Safety request, the device will go to a pre-defined safety position/mode

eLON_ES_EMERGENCY_REQ: Emergency request, the device will go to an pre-defined emergency position/mode

eLON_ES_UPDATE_STATE: Update the current state and mode

eLON_ES_SAFETY_EXT_RESUME: Resume after Safety function

eLON_ES_EMERGENCY_RESUME: Resume after Emergency function

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.1.26 E_LON_ent_opmode_cmd_t

Used by: SNVT_ent_opmode / SNVT_ent_status

TYPE E_LON_ent_opmode_cmd_t :
{
  eLON_EM_NUL := -1,
eLON_EM_UNDEFINED := 0,
eLON_EM_AUTO := 1,
eLON_EM_AUTO_RED := 2,
eLON_EM_CLOSE_LOCK := 3,
eLON_EM_CLOSE_UNLOCK := 4,
eLON_EM_EXIT_ONLY := 5,
eLON_EM_OPEN := 6,
eLON_EM_OPEN_ONCE := 7,
eLON_EM_MANUAL := 8,
eLON_EM_FIRE := 9,
eLON_EM_EVAC := 10,
eLON_EM_WEATHER := 11,
eLON_EM_DAY_LOCKING := 12,
eLON_EM_NIGHT_LOCKING := 13,
eLON_EM_BLOCKED := 14,
eLON_EM_SERVICE := 15,
eLON_EM_NUL: Invalid Value

eLON_EM_UNDEFINED: Operation mode is not defined

eLON_EM_AUTO: Operation mode is AUTOMATIC

eLON_EM_AUTO_RED: Operation mode is AUTOMATIC with reduced width

eLON_EM_CLOSE_LOCK: Operation mode is CLOSE AND LOCK

eLON_EM_CLOSE_UNLOCK: Operation mode is CLOSE AND UNLOCK

eLON_EM_EXIT_ONLY: Operation mode is EXIT ONLY

eLON_EM_OPEN: Operation mode is OPEN

eLON_EM_OPEN_ONCE: Operation mode is OPEN AND CLOSE ONCE

eLON_EM_MANUAL: Operation mode is MANUAL

eLON_EM_FIRE: Operation mode is FIRE

eLON_EM_EVAC: Operation mode is EVACUATION

eLON_EM_WEATHER: Operation mode is WEATHER MODE

eLON_EM_DAY_LOCKING: Operation mode is DAY_LOCKING, locking with reduced level of security

eLON_EM_NIGHT_LOCKING: Operation mode is NIGHT_LOCKING, locking with maximum level of security

eLON_EM_BROKEN: Operation mode is BROKEN, no operations is allowed

eLON_EM_SERVICE: Operation mode is SERVICE

eLON_EM_ENTRY_ONLY: Operation mode is ENTRY_ONLY

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.1.27 E_LON_evap_t

Used by: SNVT_evap_state

```
TYPE E_LON_evap_t :
{
    eLON_EVAP_NUL := -1,
    eLON_EVAP_NO_COOLING := 0,
    eLON_EVAP_COOLING := 1,
    eLON_EVAP_EMERG_COOLING := 2
}
END_TYPE
```

eLON_EVAP_NUL: Invalid Value

eLON_EVAP_NO_COOLING: Object not performing cooling (off cycle or disabled)

eLON_EVAP_COOLING: Object currently cooling

eLON_EVAP_EMERG_COOLING: Object performing emergency cooling
Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.2.1.28 E_LON_ex_control_t

Used by: SNVT_ex_control

```c
TYPE E_LON_ex_control_t :
{
  eLON_EX_CONTROL_NUL := -1,
  eLON_EX_CONTROL_NONE := 0,
  eLON_EX_CONTROL_OTHER := 1,
  eLON_EX_CONTROL_THIS_ADDR := 2
}
END_TYPE
```

- **eLON_EX_CONTROL_NUL**: The control status of the item is unknown
- **eLON_EX_CONTROL_NONE**: Nothing has control of the item.
- **eLON_EX_CONTROL_OTHER**: Some unidentified entity has control of the item.
- **eLON_EX_CONTROL_THIS_ADDR**: A device has control of the item. The network address of this device is specified in the control_device_addr

### 4.2.1.29 E_LON_file_request_t

Used by: SNVT_file_req

```c
TYPE E_LON_file_request_t :
{
  eLON_FR_NUL             := -1,
  eLON_FR_OPEN_TO_SEND    := 0,
  eLON_FR_OPEN_TO_RECEIVE := 1,
  eLON_FR_CLOSE_FILE      := 2,
  eLON_FR_CLOSE_DELETE_FILE := 3,
  eLON_FR_DIRECTORY_LOOKUP := 4,
  eLON_FR_OPEN_TO_SEND_RA := 5,
  eLON_FR_OPEN_TO_RECEIVE_RA := 6
}
END_TYPE
```

- **eLON_FR_NUL**: Invalid Value
- **eLON_FR_OPEN_TO_SEND**: Sequential access read
- **eLON_FR_OPEN_TO_RECEIVE**: Sequential access write
- **eLON_FR_CLOSE_FILE**: Close and save file
- **eLON_FR_CLOSE_DELETE_FILE**: Close and delete file
- **eLON_FR_DIRECTORY_LOOKUP**: Retrieve directory entry
- **eLON_FR_OPEN_TO_SEND_RA**: Random access read
- **eLON_FR_OPEN_TO_RECEIVE_RA**: Random access write
Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.1.30  E_LON_file_status_t

Used by: SNVT_file_status

```c
TYPE E_LON_file_status_t :
{
    eLON_FS_NUL := -1,
    eLON_FS_XFER_OK := 0,
    eLON_FS_LOOKUP_OK := 1,
    eLON_FS_OPEN_FAIL := 2,
    eLON_FS_LOOKUP_ERR := 3,
    eLON_FS_XFER_UNDERWAY := 4,
    eLON_FS_IO_ERR := 5,
    eLON_FS_TIMEOUT_ERR := 6,
    eLON_FS_WINDOW_ERR := 7,
    eLON_FS_AUTH_ERR := 8,
    eLON_FS_ACCESS_UNAVAIL := 9,
    eLON_FS_SEEK_INVALID := 10,
    eLON_FS_SEEK_WAIT := 11
} END_TYPE
```

eLON_FS_NUL:  Invalid Value
eLON_FS_XFER_OK: File transfer successful
eLON_FS_LOOKUP_OK: Directory lookup successful
eLON_FS_OPEN_FAIL: Error on opening file
eLON_FS_LOOKUP_ERR: Error on directory lookup
eLON_FS_XFER_UNDERWAY: File transfer in progress
eLON_FS_IO_ERR: Error on reading/writing file
eLON_FS_TIMEOUT_ERR: File transfer timed out
eLON_FS_WINDOW_ERR: Window sequence error
eLON_FS_AUTH_ERR: Authentication failure
eLON_FS_ACCESS_UNAVAIL: Access mode not supported
eLON_FS_SEEK_INVALID: Random access beyond EOF
eLON_FS_SEEK_WAIT:

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.1.31  E_LON_fire_indicator_t

Used by: SNVT_fire_indcte

```c
TYPE E_LON_fire_indicator_t :
{
    eLON_FN_NUL := -1,
    eLON_FN_UNDEFINED := 0,
    eLON_FN_STROBE_U := 1,
    eLON_FN_STROBE_S := 2,
    eLON_FN_HORN := 3,
} END_TYPE
```
eLON_FN_CHIME := 4,
eLON_FN_BELL := 5,
eLON_FN_SOUNDER := 6,
eLON_FN_SPEAKER := 7,
eLON_FN_UNIVERSAL := 8
)
END_TYPE

eLON_FN_NUL: Invalid Value

eLON_FN_UNDEFINED: Undefined indicator

eLON_FN_STROBE_U: The indicator is un-synchronized

eLON_FN_STROBE_S: The indicator is synchronized

eLON_FN_HORN: The indicator is a DC input, pre coded Horn

eLON_FN_CHIME: The indicator is a DC input, pre coded Chime

eLON_FN_BELL: The indicator is a DC input

eLON_FN_SOUNDER: The indicator is powered from the device

eLON_FN_SPEAKER: The indicator is an AC input for the speaker

eLON_FN_UNIVERSAL: General purpose indicator

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.1.32 E_LON_fire_initiator_t

Used by: SNVT_fire_init

TYPE E_LON_fire_initiator_t :
{
    eLON_FI_NUL := -1,
eLON_FI_UNDEFINED := 0,
eLON_FI_THERMAL_FIXED := 1,
eLON_FI_SMOKE_ION := 2,
eLON_FI_MULTI_ION_THERMAL := 3,
eLON_FI_SMOKE_PHOTO := 4,
eLON_FI_MULTI_PHOTO_THERMAL := 5,
eLON_FI_MULTI_PHOTO_ION := 6,
eLON_FI_MULTI_PHOTO_ION_THERMAL := 7,
eLON_FI_THERMAL_ROR := 8,
eLON_FI_MULTI_THERMAL_ROR := 9,
eLON_FI_MANUAL_PULL := 10,
eLON_FI_WATER_FLOW := 11,
eLON_FI_WATER_FLOW_TAMPER := 12,
eLON_FI_STATUS_ONLY := 13,
eLON_FI_MANUAL_CALL := 14,
eLON_FI_FIREMAN_CALL := 15,
eLON_FI_UNIVERSAL := 16
)
END_TYPE

eLON_FI_NUL: Invalid Value

eLON_FI_UNDEFINED: Initiator is undefined

eLON_FI_THERMAL_FIXED: Initiator is thermal fixed (heat)
eLON_FI_SMOKE_ION: Initiator is smoke and ion

eLON_FI_MULTI_ION_THERMAL: Initiator is multi-ion and thermal

eLON_FI_SMOKE_PHOTO: Initiator is smoke and photo
eLON_FI_MULTI_PHOTO_THERMAL: Initiator is multi-photo and thermal

eLON_FI_MULTI_PHOTO_ION: Initiator is multi-photo and ion

eLON_FI_MULTI_PHOTO_ION_THERMAL: Initiator is multi-photo, ion and thermal

eLON_FI_THERMAL_ROR: Initiator is thermal fixed and Rate of Rise

eLON_FI_MULTI_THERMAL_ROR: Initiator is multi-thermal and Rate of Rise

eLON_FI_MANUAL_PULL: Initiator is manual pull

eLON_FI_WATER_FLOW: Initiator is water flow

eLON_FI_WATER_FLOW_TAMPER: Initiator is water flow and tamper

eLON_FI_STATUS_ONLY: Initiator is status only

eLON_FI_MANUAL_CALL: Initiator is a manual call point

eLON_FI_FIREMAN_CALL: Initiator is a fireman call point

eLON_FI_UNIVERSAL: General purpose initiator definition

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.1.33 E_LON_fire_test_t

Used by: SNVT_fire_test

TYPE E_LON_fire_test_t :
{
    eLON_FT_NUL := -1,
    eLON_FT_NORMAL := 0,
    eLON_FT_RESET := 1,
    eLON_FT_TEST := 2,
    eLON_FT_NOTEST := 3
}
END_TYPE

eLON_FT_NUL: Invalid Value

eLON_FT_NORMAL: Return object to normal status

eLON_FT_RESET: Perform a RESET function (for smoke detectors)

eLON_FT_TEST: Go into TEST mode

eLON_FT_NOTEST: Exit TEST mode

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.1.34 E_LON_flow_direction_t

Used by: SNVT_flow_dir

TYPE E_LON_flow_direction_t :
{
    eLON_FD_NUL := -1,
    eLON_FD_NONE := 0,
    eLON_FD_OUT := 1,
    eLON_FD_IN := 2,
}
eLON_FD_ANY := 3

END_TYPE

eLON_FD_NUL: Invalid Value

eLON_FD_NONE: No flow/movement allowed

eLON_FD_OUT: Exit/out/away direction only

eLON_FD_IN: Entry/in/toward direction only

eLON_FD_ANY: No restriction on flow/movement

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.2.1.35 E_LON_gfci_status_t

Used by: SNVT_gfci_status

```c
TYPE E_LON_gfci_status_t :
{
  eLON_GFCI_NUL := -1,
  eLON_GFCI_UNKNOWN := 0,
  eLON_GFCI_NORMAL := 1,
  eLON_GFCI_TRIPPED := 2,
  eLON_GFCI_TEST_FAILED := 3,
  eLON_GFCI_TEST_PASSED := 4,
  eLON_GFCI_TEST_NOW := 5
}
END_TYPE
```

eLON_GFCI_NUL: Invalid Value

eLON_GFCI_UNKNOWN: Unknown response

eLON_GFCI_NORMAL: Normal GFCI operating condition

eLON_GFCI_TRIPPED: A ground-fault has caused the GFCI to interrupt the circuit

eLON_GFCI_TEST_FAILED: The GFCI failed testing

eLON_GFCI_TEST_PASSED: The GFCI passed testing

eLON_GFCI_TEST_NOW: The GFCI needs to be tested

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.2.1.36 E_LON_hvac_hvt_t

Used by: SNVT_hvac_type

```c
TYPE E_LON_hvac_hvt_t :
{
  eLON_HVT_NUL := -1,
  eLON_HVT_GENERIC := 0,
  eLON_HVT_FAN_COIL := 1,
  eLON_HVT_VAV := 2,
  eLON_HVT_HEAT_PUMP := 3,
  eLON_HVT_ROOFTOP := 4,
  eLON_HVT_UNIT_VENT := 5,
  eLON_HVT_CHILL_CEIL := 6,
  eLON_HVT_RADIATOR := 7,
}
```
Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.1.37 E_LON_hvac_overid_t

Used by: SNVT_hvac_overid

```
TYPE E_LON_hvac_overid_t :
|
  eLON_HVO_NUL := -1,
  eLON_HVO_OFF := 0,
  eLON_HVO_POSITION := 1,
  eLON_HVO_FLOW_VALUE := 2,
  eLON_HVO_FLOW_PERCENT := 3,
  eLON_HVO_OPEN := 4,
  eLON_HVO_CLOSE := 5,
  eLON_HVO_MINIMUM := 6,
  eLON_HVO_MAXIMUM := 7,
  eLON_HVO UNUSED8 := 8,
  eLON_HVO UNUSED9 := 9,
  eLON_HVO UNUSED10 := 10,
  eLON_HVO UNUSED11 := 11,
  eLON_HVO UNUSED12 := 12,
  eLON_HVO UNUSED13 := 13,
  eLON_HVO UNUSED14 := 14,
  eLON_HVO UNUSED15 := 15,
  eLON_HVO UNUSED16 := 16,
  eLON_HVO POSITION_1 := 17,
  eLON_HVO_FLOW_VALUE_1 := 18,
  eLON_HVO_FLOW_PERCENT_1 := 19,
  eLON_HVO_OPEN_1 := 20,
  eLON_HVO_CLOSE_1 := 21,
  eLON_HVO_MINIMUM_1 := 22,
  eLON_HVO_MAXIMUM_1 := 23,
  eLON_HVO UNUSED24 := 24,
  eLON_HVO UNUSED25 := 25,
  eLON_HVO UNUSED26 := 26,
  eLON_HVO UNUSED27 := 27,
  eLON_HVO UNUSED28 := 28,
  eLON_HVO UNUSED29 := 29,
  eLON_HVO UNUSED30 := 30,
  eLON_HVO UNUSED31 := 31,
  eLON_HVO UNUSED32 := 32,
  eLON_HVO POSITION_2 := 33,
```
eLON_HVO_NULL: Invalid Value

eLON_HVO_OFF: Not overridden

eLON_HVO_POSITION:

eLON_HVO_FLOW_VALUE: Override flow in liters/sec - use flow field

eLON_HVO_FLOW_PERCENT: Override flow percentage - use percent field

eLON_HVO_OPEN: Override to position = 100%

eLON_HVO_CLOSE: Override to position = 0%

eLON_HVO_MINIMUM: Override to configured minimum

eLON_HVO_MAXIMUM: Override to configured maximum

eLON_HVO_UNUSED8:

eLON_HVO_UNUSED9:

eLON_HVO_UNUSED10:

eLON_HVO_UNUSED11:

eLON_HVO_UNUSED12:

eLON_HVO_UNUSED13:

eLON_HVO_UNUSED14:

eLON_HVO_UNUSED15:

eLON_HVO_UNUSED16:

eLON_HVO_POSITION_1:

eLON_HVO_FLOW_VALUE_1: Override flow in liters/sec - use flow field

eLON_HVO_FLOW_PERCENT_1: Override flow percentage - use percent field

eLON_HVO_OPEN_1: Override to position = 100%

eLON_HVO_CLOSE_1: Override to position = 0%

eLON_HVO_MINIMUM_1: Override to configured minimum

eLON_HVO_MAXIMUM_1: Override to configured maximum

eLON_HVO_UNUSED24:

eLON_HVO_UNUSED25:

eLON_HVO_UNUSED26:
eLON_HVO_UNUSED27:
eLON_HVO_UNUSED28:
eLON_HVO_UNUSED29:
eLON_HVO_UNUSED30:
eLON_HVO_UNUSED31:
eLON_HVO_UNUSED32:
eLON_HVO_POSITION_2:
eLON_HVO_FLOW_VALUE_2: Override flow in liters/sec - use flow field
eLON_HVO_FLOW_PERCENT_2: Override flow percentage - use percent field
eLON_HVO_OPEN_2: Override to position = 100%
eLON_HVO_CLOSE_2: Override to position = 0%
eLON_HVO_MINIMUM_2: Override to configured minimum
eLON_HVO_MAXIMUM_2: Override to configured maximum

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.1.38  E_LON_hvac_t

Used by: SNVT_chlr_status / SNVT_hvac_mode / SNVT_hvac_status

TYPE E_LON_hvac_t :
{
  eLON_HVAC_NUL := -1,
  eLON_HVAC_AUTO := 0,
  eLON_HVAC_HEAT := 1,
  eLON_HVAC_MRNG_WRMUP := 2,
  eLON_HVAC_COOL := 3,
  eLON_HVAC_NIGHT_PURGE := 4,
  eLON_HVAC_PRE_COOL := 5,
  eLON_HVAC_OFF := 6,
  eLON_HVAC_TEST := 7,
  eLON_HVAC_EMERG_HEAT := 8,
  eLON_HVAC_FAN_ONLY := 9,
  eLON_HVAC_FREE_COOL := 10,
  eLON_HVAC_FAN := 11,
  eLON_HVAC_MAX_HEAT := 12,
  eLON_HVAC_ECONOMY := 13,
  eLON_HVAC_DEHUMID := 14,
eLON_HVAC_CALIBRATE := 15,
eLON_HVAC_EMERG_COOL := 16,
eLON_HVAC_EMERG_STEAM := 17,
eLON_HVAC_MAX_COOL := 18,
eLON_HVAC_HVC_LOAD := 19,
eLON_HVAC_NO_LOAD := 20
END_TYPE

eLON_HVAC_NUL: Invalid value

eLON_HVAC_AUTO: Controller automatically changes between application modes

eLON_HVAC_HEAT: Heating only

eLON_HVAC_MRNG_WRMUP: Application-specific morning warm-up

eLON_HVAC_COOL: Cooling only

eLON_HVAC_NIGHT_PURGE: Application-specific night purge

eLON_HVAC_PRE_COOL: Application-specific pre-cool

eLON_HVAC_OFF: Controller not controlling outputs

eLON_HVAC_TEST: Equipment being tested

eLON_HVAC_EMERG_HEAT: Emergency heat mode (heat pump)
eLON_HVAC_FAN_ONLY: Air not conditioned, fan turned on

eLON_HVAC_FREE_COOL: Cooling with compressor not running

eLON_HVAC_ICE: Ice-making mode

eLON_HVAC_MAX_HEAT: Maximum heating mode

eLON_HVAC_ECONOMY: Economic Heat/Cool mode

eLON_HVAC_DEHUMID: Dehumidification mode

eLON_HVAC_CALIBRATE: Calibration mode

eLON_HVAC_EMERG_COOL: Emergency cool mode

eLON_HVAC_EMERG_STEAM: Emergency steam mode

eLON_HVAC_MAX_COOL:
eLON_HVAC_HVC_LOAD:
eLON_HVAC_NO_LOAD:

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.1.39 E_LON_learn_mode_t

Used by: SNVT_preset

```plaintext
TYPE E_LON_learn_mode_t :
{
eLON_LN_NUL := -1,
eLON_LN_RECALL := 0,
eLON_LN_LEARN_CURRENT := 1,
eLON_LN_LEARN_VALUE := 2,
eLON_LN_REPORT_VALUE := 3
} END_TYPE
```
eLON_LN_NUL: Invalid Value

eLON_LN_RECALL: Recall

eLON_LN_LEARN_CURRENT: Learn present value

eLON_LN_LEARN_VALUE: Learn given value

eLON_LN_REPORT_VALUE: Report the value

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

#### 4.2.1.40  E_LON_log_status_t

Used by: SCPTlogRecord / SNVT_log_status

TYPE E_LON_log_status_t :
{
  eLON_LS_NUL            := -1,
  eLON_LS_ENABLED       := 0,
  eLON_LS_DISABLED      := 1,
  eLON_LS_FULL          := 2,
  eLON_LS_OVERFLOW_ERR  := 3,
  eLON_LS_INVALID_LOG_ERR := 4,
  eLON_LS_APP_ERR       := 5
}
END_TYPE

- **eLON_LS_NUL**: Invalid value
- **eLON_LS_ENABLED**: Log enabled
- **eLON_LS_DISABLED**: Log disabled
- **eLON_LS_FULL**: Log enabled and full
- **eLON_LS_OVERFLOW_ERR**: Log enabled, overflow occurred
- **eLON_LS_INVALID_LOG_ERR**: Invalid log selected
- **eLON_LS_APP_ERR**: Other application error

#### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

#### 4.2.1.41  E_LON_motor_state_t

Used by: SNVT_motor_state / SNVT_pumpset_mn

TYPE E_LON_motor_state_t :
{
  eLON_MOTOR_NUL            := -1,
  eLON_MOTOR_STOPPED       := 0,
  eLON_MOTOR_STARTING      := 1,
  eLON_MOTOR_ACCELERATING  := 2,
  eLON_MOTOR_AT_STANDBY    := 3,
  eLON_MOTOR_AT_NORMAL     := 4,
  eLON_MOTOR_AT_REFERENCE  := 5,
  eLON_MOTOR_DECELERATING  := 6,
  eLON_MOTOR_STOPPING      := 7
}
END_TYPE
eLON_MOTOR_NUL: The state of the motor is unknown (invalid value)
eLON_MOTOR_STOPPED: The motor is not running
eLON_MOTOR_STARTING: The motor is performing its start-up sequence
eLON_MOTOR_ACCELERATING: The motor is running. Speed is increasing.
eLON_MOTOR_AT_STANDBY: The motor is running in its standby mode
eLON_MOTOR_AT_NORMAL: The motor is running in its normal operational mode
eLON_MOTOR_AT_REFERENCE: The motor is running at its reference speed.
eLON_MOTOR_DECELERATING: The motor is running. Speed is decreasing.
eLON_MOTOR_STOPPING: The motor is running, beginning its shutdown sequence.

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.1.42 E_LON_nv_type_category_t

Used by: SNVT_nv_type

```
TYPE E_LON_nv_type_category_t :
{
    eLON_NV_TYPE_CAT_NUL := -1,
    eLON_NV_TYPE_CAT_INITIAL := 0,
    eLON_NV_TYPE_CAT_SIGNED_CHAR := 1,
    eLON_NV_TYPE_CAT_UNSIGNED_CHAR := 2,
    eLON_NV_TYPE_CAT_SIGNED_SHORT := 3,
    eLON_NV_TYPE_CAT_UNSIGNED_SHORT := 4,
    eLON_NV_TYPE_CAT_SIGNED_LONG := 5,
    eLON_NV_TYPE_CAT_UNSIGNED_LONG := 6,
    eLON_NV_TYPE_CAT_ENUM := 7,
    eLON_NV_TYPE_CAT_ARRAY := 8,
    eLON_NV_TYPE_CAT_STRUCT := 9,
    eLON_NV_TYPE_CAT_UNION := 10,
    eLON_NV_TYPE_CAT_BITFIELD := 11,
    eLON_NV_TYPE_CAT_FLOAT := 12,
    eLON_NV_TYPE_CAT_SIGNED_QUAD := 13,
    eLON_NV_TYPE_CAT_REFERENCE := 14
}
END_TYPE
```
eLON_NV_TYPE_CAT_NUL: Invalid Value
eLON_NV_TYPE_CAT_INITIAL:
eLON_NV_TYPE_CAT_SIGNED_CHAR: 8-bit signed character
eLON_NV_TYPE_CAT_UNSIGNED_CHAR: 8-bit unsigned character
eLON_NV_TYPE_CAT_SIGNED_SHORT: 8-bit signed integer
eLON_NV_TYPE_CAT_UNSIGNED_SHORT: 8-bit unsigned integer
eLON_NV_TYPE_CAT_SIGNED_LONG: 16-bit signed integer
eLON_NV_TYPE_CAT_UNSIGNED_LONG: 16-bit unsigned integer
eLON_NV_TYPE_CAT_ENUM: 8-bit enumeration
eLON_NV_TYPE_CAT_ARRAY: Array
eLON_NV_TYPE_CAT_STRUCT: Structure
eLON_NV_TYPE_CAT_UNION: Union
eLON_NVT_CAT_BITFIELD: Bitfield

eLON_NVT_CAT_FLOAT: 32-bit IEC 60559 (IEEE 754) floating-point value

eLON_NVT_CAT_SIGNED_QUAD: 32-bit signed integer

eLON_NVT_CAT_REFERENCE: Reference type

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.1.43 E_LON_object_request_t

Used by: SNVT_obj_request

TYPE E_LON_object_request_t :
{
  eLON_RQ_NUL := -1,
  eLON_RQ_NORMAL := 0,
  eLON_RQ_DISABLED := 1,
  eLON_RQ_UPDATE_STATUS := 2,
  eLON_RQ_SELF_TEST := 3,
  eLON_RQ_UPDATE_ALARM := 4,
  eLON_RQ_REPORT_MASK := 5,
  eLON_RQ_OVERRIDE := 6,
  eLON_RQ_ENABLE := 7,
  eLON_RQ_RMV_OVERRIDE := 8,
  eLON_RQ_CLEAR_STATUS := 9,
  eLON_RQ_CLEAR_ALARM := 10,
  eLON_RQ_ALARM_NOTIFY_ENABLED := 11,
  eLON_RQ_ALARM_NOTIFY_DISABLED := 12,
  eLON_RQ_MANUAL_CTRL := 13,
  eLON_RQ_REMOTE_CTRL := 14,
  eLON_RQ_PROGRAM := 15,
  eLON_RQ_CLEAR_RESET := 16,
  eLON_RQ_RESET := 17,
  eLON_RQ_CLEAR_LOG := 18
}
END_TYPE

eLON_RQ_NUL: Invalid Value

eLON_RQ_NORMAL: Enable object and remove override

eLON_RQ_DISABLED: Disable object

eLON_RQ_UPDATE_STATUS: Report object status

eLON_RQ_SELF_TEST: Perform object self-test

eLON_RQ_UPDATE_ALARM: Update alarm status

eLON_RQ_REPORT_MASK: Report status bit mask

eLON_RQ_OVERRIDE: Override object

eLON_RQ_ENABLE: Enable object

eLON_RQ_RMV_OVERRIDE: Remove object override

eLON_RQ_CLEAR_STATUS: Clear object status

eLON_RQ_CLEAR_ALARM: Clear object alarm

eLON_RQ_ALARM_NOTIFY_ENABLED: Enable alarm notification

eLON_RQ_ALARM_NOTIFY_DISABLED: Disable alarm notification

eLON_RQ_MANUAL_CTRL: Enable object for manual control
eLON_RQ_REMOTE_CTRL: Enable object for remote control

eLON_RQ_PROGRAM: Enable programming of special configuration properties

eLON_RQ_CLEAR_RESET: Clear reset-complete flag (reset_complete)

eLON_RQ_RESET: Execute reset-sequence of object

eLON_RQ_CLEAR_LOG: Clear data log

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.1.44  E_LON_occu_p_t

Used by: SNVT_occupancy / SNVT_tod_event

```
TYPE E_LON_occu_p_t :
{
  eLON_OC_NUL := -1,
  eLON_OC_OCCUPIED := 0,
  eLON_OC_UNOCCUPIED := 1,
  eLON_OC_BYPASS := 2,
  eLON_OC_STANDBY := 3
}
END_TYPE
```

eLON_OC_NUL: Invalid Value

eLON_OC_OCCUPIED: Area is occupied

eLON_OC_UNOCCUPIED: Area is unoccupied

eLON_OC_BYPASS: Area is temporarily occupied for the bypass period

eLON_OC_STANDBY: Area is temporarily unoccupied

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.1.45  E_LON Override_t

Used by: SNVT_override

```
TYPE E_LON_override_t :
{
  eLON_OV_NUL := -1,
  eLON_OV_RETAIN := 0,
  eLON_OV_SPECIFIED := 1,
  eLON_OV_DEFAULT := 2
}
END_TYPE
```

eLON_OV_NUL: Invalid Value

eLON_OV_RETAIN: Retain current level

eLON_OV_SPECIFIED: Go to specified level

eLON_OV_DEFAULT: Go to default level
Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.2.1.46 E_LON_pan_dir_t

Used by: SNVT_ptz

```c
typedef enum {
    eLON_PAN_NUL   := -1,
    eLON_PAN_STOP  := 0,
    eLON_PAN_RIGHT := 1,
    eLON_PAN_LEFT  := 2
} E_LON_pan_dir_t;

E_LON_PAN_NUL: Invalid Value
E_LON_PAN_STOP: Stop panning
E_LON_PAN_RIGHT: Pan to the right
E_LON_PAN_LEFT: Pan to the left
```

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.2.1.47 E_LON_priority_level_t

Used by: SNVT_alarm / SNVT_alarm_2 / SNVT_pumpset_mn

```c
typedef enum {
    eLON_PR_NUL       := -1,
    eLON_PR_LEVEL_0   := 0,
    eLON_PR_LEVEL_1   := 1,
    eLON_PR_LEVEL_2   := 2,
    eLON_PR_LEVEL_3   := 3,
    eLON_PR_1         := 4,
    eLON_PR_2         := 5,
    eLON_PR_3         := 6,
    eLON_PR_4         := 7,
    eLON_PR_5         := 8,
    eLON_PR_8         := 9,
    eLON_PR_10        := 10,
    eLON_PR_16        := 11
} E_LON_priority_level_t;
```

E_LON_PR_NUL: Invalid Value
E_LON_PR_LEVEL_0: Lowest alarm priority level
E_LON_PR_LEVEL_1:
E_LON_PR_LEVEL_2:
E_LON_PR_LEVEL_3: Highest alarm priority level
E_LON_PR_1: Life Safety Fire Alarms (BACnet Priority 2)
E_LON_PR_2: Property Safety Fire Alarms (BACnet Priority 3)
E_LON_PR_3: Fire Supervisory Alarm (BACnet Priority 4)
eLON_PR_4: Fire Trouble/Fault (Display) (BACnet Priority 5)
eLON_PR_6: Fire Pre-Alarm, HVAC Critical Equipment Alarm (BACnet Priority 6)
eLON_PR_8: HVAC Alarms (BACnet Priority 8)
eLON_PR_10: HVAC Critical Equipment RTN, Fire RTN (Display) (BACnet Priority 10)
eLON_PR_16: HVAC RTN (lowest priority) (BACnet Priority 16)

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.1.48 E_LON_privacyzone_t

Used by: SNVT_privacyzone

TYPE E_LON_privacyzone_t :
{
    eLON_PZ_NUL := -1,
    eLON_PZ_DISABLE := 0,
    eLON_PZ_ENABLE := 1,
    eLON_PZ_UPPER_LEFT := 2,
    eLON_PZ_LOWER_RIGHT := 3,
    eLON_PZ_ENTER := 4,
    eLON_PZ_EXIT := 5
}
END_TYPE

eLON_PZ_NUL: Invalid value
eLON_PZ_DISABLE: Disable privacy zone warning
eLON_PZ_ENABLE: Enable privacy zone warning
eLON_PZ_UPPER_LEFT: Set upper left corner
eLON_PZ_LOWER_RIGHT: Set lower right corner
eLON_PZ_ENTER: Privacy zone enter warning
eLON_PZ_EXIT: Privacy zone exit message

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.1.49 E_LON_rail_audio_sensor_type_t

Used by: SNVT_rac_ctrl / SNVT_rac_req

TYPE E_LON_rail_audio_sensor_type_t :
{
    eLON_RAST_NUL := -1,
    eLON_RAST_CU_TYPE_1 := 0,
    eLON_RAST_CU_TYPE_2 := 1,
    eLON_RAST_CU_TYPE_3 := 2,
    eLON_RAST_CU_TYPE_4 := 3,
    eLON_RAST_LS_LINE_1 := 4,
    eLON_RAST_LS_LINE_2 := 5,
    eLON_RAST_LS_LINE_3 := 6,
    eLON_RAST_LS_LINE_4 := 7,
    eLON_RAST_LS_LINE_5 := 8,
    eLON_RAST_LS_LINE_6 := 9,
    eLON_RAST_LS_LINE_7 := 10,
    eLON_RAST_LS_LINE_8 := 11,
eLON_RAST_PAU := 12,
eLON_RAST_CFA_TYPE_1 := 13,
eLON_RAST_CFA_TYPE_2 := 14,
eLON_RAST_CFA_TYPE_3 := 15,
eLON_RAST_CFA_TYPE_4 := 16,
eLON_RAST_DVA := 17,
eLON_RAST_ET_TYPE_1 := 18,
eLON_RAST_ET_TYPE_2 := 19,
eLON_RAST_USERDEF_TYPE_1 := 20,
eLON_RAST_USERDEF_TYPE_2 := 21,
eLON_RAST_USERDEF_TYPE_3 := 22,
eLON_RAST_USERDEF_TYPE_4 := 23
)

END_TYPE

eLON_RAST_NUL: Invalid Value

eLON_RAST_CU_TYPE_1: CU Type 1

eLON_RAST_CU_TYPE_2: CU Type 2

eLON_RAST_CU_TYPE_3:

eLON_RAST_CU_TYPE_4: CU Type 4

eLON_RAST_LS_LINE_1: LS Line 1

eLON_RAST_LS_LINE_2: LS Line 2

eLON_RAST_LS_LINE_3: LS Line 3

eLON_RAST_LS_LINE_4: LS Line 4

eLON_RAST_LS_LINE_5: LS Line 5

eLON_RAST_LS_LINE_6: LS Line 6

eLON_RAST_LS_LINE_7: LS Line 7

eLON_RAST_LS_LINE_8: LS Line 8

eLON_RAST_PAU: Public-Address Unit

eLON_RAST_CFA_TYPE_1: CFA Type 1

eLON_RAST_CFA_TYPE_2: CFA Type 2

eLON_RAST_CFA_TYPE_3: CFA Type 3

eLON_RAST_CFA_TYPE_4: CFA Type 4

eLON_RAST_DVA: DVA

eLON_RAST_ET_TYPE_1: ET Type 1

eLON_RAST_ET_TYPE_2: ET Type 2

eLON_RAST_USERDEF_TYPE_1: User-defined Type 1

eLON_RAST_USERDEF_TYPE_2: User-defined Type 2

eLON_RAST_USERDEF_TYPE_3: User-defined Type 3

eLON_RAST_USERDEF_TYPE_4: User-defined Type 4

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.2.1.50 E_LON_rail_audio_type_t

Used by: SNVT_rac_ctrl / SNVT_rac_req

```
TYPE E_LON_rail_audio_type_t :
{
  eLON_RAT_NUL := -1,
eLON_RAT_IC_REQ := 0,
eLON_RAT_IC_JOIN := 1,
eLON_RAT_IC_QUIT := 2,
eLON_RAT_IC_END := 3,
eLON_RAT_HW_RADIO_REQ := 4,
eLON_RAT_HW_RADIO_END := 5,
eLON_RAT_HW_PA_REQ := 6,
eLON_RAT_HW_PA_END := 7,
eLON_RAT_SW_PA_REQ := 8,
eLON_RAT_SW_PA_END := 9,
eLON_RAT_SW_PA_OR_REQ := 10,
eLON_RAT_SW_PA_OR_END := 11,
eLON_RAT_PAU_REQ := 12,
eLON_RAT_PAU_ACCEPT := 13,
eLON_RAT_PAU_CALL := 14,
eLON_RAT_PAU_END := 15,
eLON_RAT_ENTERT_REQ := 16,
eLON_RAT_ENTERT_END := 17
}
END_TYPE
```

eLON_RAT_NUL:
eLON_RAT_IC_REQ:
eLON_RAT_IC_JOIN:
eLON_RAT_IC_QUIT:
eLON_RAT_IC_END:
eLON_RAT_HW_RADIO_REQ:
eLON_RAT_HW_RADIO_END:
eLON_RAT_HW_PA_REQ:
eLON_RAT_HW_PA_END:
eLON_RAT_SW_PA_REQ:
eLON_RAT_SW_PA_END:
eLON_RAT_SW_PA_OR_REQ:
eLON_RAT_SW_PA_OR_END:
eLON_RAT_PAU_REQ:
eLON_RAT_PAU_ACCEPT:
eLON_RAT_PAU_CALL:
eLON_RAT_PAU_END:
eLON_RAT_ENTERT_REQ:
eLON_RAT_ENTERT_END:

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.2.1.51  E_LON_reg_val_unit_t

Used by: SNVT_reg_val / SNVT_reg_val_ts

TYPE E_LON_reg_val_unit_t:
{
  eLON_RVU_NUL := -1,
  eLON_RVU_NONE := 0,
  eLON_RVU_W := 1,
  eLON_RVU_KW := 2,
  eLON_RVU_MW := 3,
  eLON_RVU_GW := 4,
  eLON_RVU_VAR := 5,
  eLON_RVU_KVAR := 6,
  eLON_RVU_MVAR := 7,
  eLON_RVU_GVAR := 8,
  eLON_RVU_WH := 9,
  eLON_RVU_MWH := 10,
  eLON_RVU_GWH := 11,
  eLON_RVU_VARH := 12,
  eLON_RVU_KVARH := 13,
  eLON_RVU_MVARH := 14,
  eLON_RVU_GVARH := 15,
  eLON_RVU_V := 16,
  eLON_RVU_A := 17,
  eLON_RVU_COSF := 18,
  eLON_RVU_M3 := 19,
  eLON_RVU_L := 20,
  eLON_RVU_ML := 21,
  eLON_RVU_USGAL := 22,
  eLON_RVU_GJ := 23,
  eLON_RVU_MJ := 24,
  eLON_RVU_MCAL := 25,
  eLON_RVU_KCAL := 26,
  eLON_RVU_MBTU := 27,
  eLON_RVU_KBTU := 28,
  eLON_RVU_MJH := 29,
  eLON_RVU_MLS := 30,
  eLON_RVU_M3S := 31,
  eLON_RVU_C := 32,
  eLON_RVU_LH := 33,
  eLON_RVU_VA := 34,
  eLON_RVU_KVA := 35,
  eLON_RVU_MVA := 36,
  eLON_RVU_GVA := 37,
  eLON_RVU_VAH := 38,
  eLON_RVU_KVAH := 39,
  eLON_RVU_MVAH := 40,
  eLON_RVU_GVAH := 41,
  eLON_RVU_VARH := 42,
  eLON_RVU_GVARH := 43
}
END_TYPE

eLON_RVU_NUL: invalid unit of measure (INVALID)
eLON_RVU_NONE: no units specified ( )
eLON_RVU_W: Watts (W)
eLON_RVU_KW: kiloWatts (kW)
eLON_RVU_MW: megaWatts (MW)
eLON_RVU_GW: gigaWatts (GW)
eLON_RVU_VAR: Volt-Amperes reactive (var)
eLON_RVU_KVAR: kilo-Volt-Amperes reactive (kvar)
eLON_RVU_MVAR: mega-Volt-Amperes reactive (Mvar)
eLON_RVU_GVAR: giga-Volt-Amperes reactive (Gvar)
eLON_RVU_WH: Watt-hour (Wh)
eLON_RVU_KWH: kiloWatt-hour (kWh)
eLON_RVU_MWH: megaWatt-hour (MWh)
eLON_RVU_GWH: gigawatt-hour (GWh)
eLON_RVU_VARH: Volt-Amperes reactive -hour (varh)
eLON_RVU_KVARH: kilo-Volt-Amperes reactive -hour (kvarh)
eLON_RVU_MVARH: mega-Volt-Amperes reactive -hour (Mvarh)
eLON_RVU_GVARH: giga-Volt-Amperes reactive -hour (Gvarh)
eLON_RVU_V: Volts (V)
eLON_RVU_A: Amps (A)
eLON_RVU_COSF: (cosf)
eLON_RVU_M3: cubic meters (m³)(cu.m)
eLON_RVU_L: liters (l)
eLON_RVU_ML: milliliters (ml)
eLON_RVU_USGAL: U.S. Gallons (USG)
eLON_RVU_GJ: giga-Joules (GJ)
eLON_RVU_MJ: mega-Joules (MJ)
eLON_RVU_MCAL: megacalories (Mcal)
eLON_RVU_KCAL: kilocalories (kcal) / Calories (Cal)
eLON_RVU_MBTU: mega-British thermal units (mBtu)
eLON_RVU_KBTU: kilo-British thermal units (kBtu)
eLON_RVU_MJH: mega-Joules per hour (MJ/h)
eLON_RVU_MLS: milliliters per second (ml/s)
eLON_RVU_LS: liters per second (l/s)
eLON_RVU_M3S: cubic-meters per second (m³/s) (cu.m/s)
eLON_RVU_C: (°C)
eLON_RVU_LH: liters per hour (l/h)
eLON_RVU_VA: Volt-Amperes (VA)
eLON_RVU_KVA: kiloVolt-Amperes (kVA)
eLON_RVU_MVA: megaVolt-Amperes (MVA)
eLON_RVU_GVA: gigaVolt-Amperes (GVA)
eLON_RVU_VAH: Volt-Ampere hours (VAh)
eLON_RVU_KVAH: kiloVolt-Ampere hours (kVAh)
eLON_RVU_MVAH: megaVolt-Ampere hours (MV Ah)
eLON_RVU_GVAH: giga-Volt-Ampere hours (GVAh)

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.2.1.52 \hspace{1em} \texttt{E\_LON\_sblnd\_cmd\_source\_t}

Used by: \texttt{SNVT\_sblnd\_state}

```c
TYPE \texttt{E\_LON\_sblnd\_cmd\_source\_t}:
{ 
  \texttt{eLON\_SBCS\_NUL} := -1,
  \texttt{eLON\_SBCS\_LOCAL} := 0,
  \texttt{eLON\_SBCS\_GROUP} := 1,
  \texttt{eLON\_SBCS\_WIND\_SPEED} := 2,
  \texttt{eLON\_SBCS\_SUN\_LUX} := 3,
  \texttt{eLON\_SBCS\_RAIN} := 4,
  \texttt{eLON\_SBCS\_FROST} := 5,
  \texttt{eLON\_SBCS\_DAWN} := 6,
  \texttt{eLON\_SBCS\_DUSK} := 7,
  \texttt{eLON\_SBCS\_OUTSIDE\_TEMP} := 8,
  \texttt{eLON\_SBCS\_INDOOR\_TEMP} := 9,
  \texttt{eLON\_SBCS\_OUTDOOR\_RH} := 10,
  \texttt{eLON\_SBCS\_INDOOR\_RH} := 11,
  \texttt{eLON\_SBCS\_ILLUM\_LEVEL} := 12,
  \texttt{eLON\_SBCS\_SCENE} := 13,
  \texttt{eLON\_SBCS\_GLOBAL} := 14,
  \texttt{eLON\_SBCS\_WINDOW\_CONTACT} := 15,
  \texttt{eLON\_SBCS\_AUTOMODE\_CHANGED} := 16,
  \texttt{eLON\_SBCS\_OVERRIDE} := 17,
  \texttt{eLON\_SBCS\_EMERGENCY} := 18,
  \texttt{eLON\_SBCS\_MAINTENANCE} := 19,
  \texttt{eLON\_SBCS\_INTRUSION} := 20,
  \texttt{eLON\_SBCS\_TERMINAL\_LOAD} := 21,
  \texttt{eLON\_SBCS\_ALARM} := 22,
  \texttt{eLON\_SBCS\_OCC\_SENSOR} := 23,
  \texttt{eLON\_SBCS\_OCC\_MAN\_CMD} := 24,
  \texttt{eLON\_SBCS\_GLARE} := 25,
  \texttt{eLON\_SBCS\_ALARM\_2} := 26,
  \texttt{eLON\_SBCS\_NOTIFY} := 27,
  \texttt{eLON\_SBCS\_ELEVATION} := 28,
  \texttt{eLON\_SBCS\_AZIMUTH} := 29,
  \texttt{eLON\_SBCS\_SET\_OVERRIDE} := 30,
  \texttt{eLON\_SBCS\_SET\_MAINTENANCE} := 31,
  \texttt{eLON\_SBCS\_TIMER} := 32,
  \texttt{eLON\_SBCS\_UNKNOWN} := 127
}
END_TYPE
```

\texttt{eLON\_SBCS\_NUL}: Invalid value

\texttt{eLON\_SBCS\_LOCAL}: Local

\texttt{eLON\_SBCS\_GROUP}: Group

\texttt{eLON\_SBCS\_WIND\_SPEED}: Wind speed

\texttt{eLON\_SBCS\_SUN\_LUX}: Sun lux level

\texttt{eLON\_SBCS\_RAIN}: Rain

\texttt{eLON\_SBCS\_FROST}: Frost

\texttt{eLON\_SBCS\_DAWN}: Dawn

\texttt{eLON\_SBCS\_DUSK}: Dusk

\texttt{eLON\_SBCS\_OUTSIDE\_TEMP}: Outside temperature

\texttt{eLON\_SBCS\_INDOOR\_TEMP}: Indoor temperature

\texttt{eLON\_SBCS\_OUTDOOR\_RH}: Outdoor relative humidity

\texttt{eLON\_SBCS\_INDOOR\_RH}: Indoor relative humidity

\texttt{eLON\_SBCS\_ILLUM\_LEVEL}: Illumination level

\texttt{eLON\_SBCS\_SCENE}: Scene

\texttt{eLON\_SBCS\_GLOBAL}: Global
**eLON_SBCS_WINDOW_CONTACT**: Window contact

**eLON_SBCS_AUTOMODE_CHANGED**: Auto-mode changed

**eLON_SBCS_OVERRIDE**: Override

**eLON_SBCS_EMERGENCY**: Emergency

**eLON_SBCS_MAINTENANCE**: Maintenance

**eLON_SBCS_INTRUSION**: Intrusion

**eLON_SBCS_TERMINAL_LOAD**: Terminal load

**eLON_SBCS_ALARM**: Alarm

**eLON_SBCS_OCC_SENSOR**: Occupancy sensor

**eLON_SBCS_OCC_MAN_CMD**: Occupancy manual command

**eLON_SBCS_GLARE**: Glare

**eLON_SBCS_ALARM_2**: Alarm 2

**eLON_SBCS_NOTIFY**: Notify

**eLON_SBCS_ELEVATION**: Elevation

**eLON_SBCS_AZIMUTH**: Azimuth

**eLON_SBCS_SET_OVERRIDE**: Set override

**eLON_SBCS_SET_MAINTENANCE**: Set maintenance

**eLON_SBCS_TIMER**: Timer

**eLON_SBCS_UNKNOWN**: Unknown command source

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

#### 4.2.1.53 **E_LON_sblnd_error_t**

Used by: SNVT_sblnd_state

```c
TYPE E_LON_sblnd_error_t :
{
    eLON_SBE_NUL := -1,
    eLON_SBE_NO_ERROR := 0,
    eLON_SBE_IN_PROGRESS := 1,
    eLON_SBE_LIMITS := 2,
    eLON_SBE_OBSTACLE_UP := 3,
    eLON_SBE_OBSTACLE_DOWN := 4,
    eLON_SBE_OVERHEAT := 5,
    eLON_SBE_POWER := 6,
    eLON_SBE_SENSOR := 7,
    eLON_SBE_MOTOR_CIRCUIT := 8,
    eLON_SBE_FUSE := 9,
    eLON_SBE_REFERENCE_LOST := 10,
    eLON_SBE_HOST_COMM := 11,
    eLON_SBE_VOLTAGE_1 := 12,
    eLON_SBE_VOLTAGE_2 := 13,
    eLON_SBE_CONTROLLER := 14
}
END_TYPE
```

**eLON_SBE_NUL**: Invalid Value

**eLON_SBE_NO_ERROR**: No error
eLON_SBE_IN_PROGRESS: In progress
eLON_SBE_LIMITS: Limits
eLON_SBE_OBSTACLE_UP: Obstacle up
eLON_SBE_OBSTACLE_DOWN: Obstacle down
eLON_SBE_OVERHEAT: Overheat
eLON_SBE_POWER: Power
eLON_SBE_SENSOR: Sensor
eLON_SBE_MOTOR_CIRCUIT: Motor circuit
eLON_SBE_FUSE: Fuse
eLON_SBE_REFERENCE_LOST: Reference lost
eLON_SBE_HOST_COMM: Host communication
eLON_SBE_VOLTAGE_1: Voltage 1
eLON_SBE_VOLTAGE_2: Voltage 2
eLON_SBE_CONTROLLER: Controller

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.1.54 E_LON_scene_config_t

Used by: SNVT_scene_cfg

```
TYPE E_LON_scene_config_t :
{
  eLON_SCF_NUL ::= -1,
  eLON_SCF_SAVE ::= 0,
  eLON_SCF_CLEAR ::= 1,
  eLON_SCF_REPORT ::= 2,
  eLON_SCF_SIZE ::= 3,
  eLON_SCF_FREE ::= 4
}
END_TYPE
```

eLON_SCF_NUL: Invalid Value

eLON_SCF_SAVE: Overwrite this scene with new data

eLON_SCF_CLEAR: Delete this scene from the list

eLON_SCF_REPORT: Display this scene's data

eLON_SCF_SIZE: Report the number of programmed scenes

eLON_SCF_FREE: Report the number of free scene storage spaces

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.2.1.55  E_LON_scene_t

Used by: SNVT_scene

TYPE E_LON_scene_t :
{
  eLON_SC_NUL := -1,
  eLON_SC_RECALL := 0,
  eLON_SC_LEARN := 1,
  eLON_SC_DISPLAY := 2,
  eLON_SC_GROUP_OFF := 3,
  eLON_SC_GROUP_ON := 4,
  eLON_SC_STATUS_OFF := 5,
  eLON_SC_STATUS_ON := 6,
  eLON_SC_STATUS_MIXED := 7,
  eLON_SC_GROUP_STATUS := 8,
  eLON_SC_FLICK := 9,
  eLON_SC_TIMEOUT := 10,
  eLON_SC_TIMEOUT_FLICK := 11,
  eLON_SC_DELAYOFF := 12,
  eLON_SC_DELAYOFF_FLICK := 13,
  eLON_SC_DELAYON := 14,
  eLON_SC延迟OFF := 15,
  eLON_SC_DISABLE_GROUP := 16,
  eLON_SC_CLEANON := 17,
  eLON_SC_CLEANOFF := 18,
  eLON_SC_WINK := 19,
  eLON_SC_RESET := 20,
  eLON_SC_MODE1 := 21,
  eLON_SC_MODE2 := 22,
  eLON_SC_MODE3 := 23
}
END_TYPE

eLON_SC_NUL: Invalid value

eLON_SC_RECALL: Recall a specified scene.

eLON_SC_LEARN: Store the current setting in the specified scene.

eLON_SC_DISPLAY: Display the current scene.

eLON_SC_GROUP_OFF: Report current group is off.

eLON_SC_GROUP_ON: Report current group is on.

eLON_SC_STATUS_OFF: Report current status is off.

eLON_SC_STATUS_ON: Report current status is on.

eLON_SC_STATUS_MIXED: Report current status is mixed.

eLON_SC_GROUP_STATUS: Get group status.

eLON_SC_FLICK: Toggle state off and then on.

eLON_SC_TIMEOUT: Report a timeout occurred.

eLON_SC_TIMEOUT_FLICK: Report a timeout occurred for a flick warning.

eLON_SC_DELAYOFF: Set the state to off after a delay.

eLON_SC_DELAYOFF_FLICK: Flick and then set the state to off after a delay.

eLON_SC_DELAYON: Set the state to on after a delay.

eLON_SC_ENABLE_GROUP: Enable the current group.

eLON_SC_DISABLE_GROUP: Disable the current group.

eLON_SC_CLEANON: Recall the cleaning scene.

eLON_SC_CLEANOFF: Restore the previous scene.

eLON_SC_WINK: Toggle to the opposite state and then restore the state.
eLON_SC_RESET: Restore the factory default scene table.

eLON_SC_MODE1: Manufacturer-specific mode 1.

eLON_SC_MODE2: Manufacturer-specific mode 2.

eLON_SC_MODE3: Manufacturer-specific mode 3.

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.1.56  E_LON_sec_state_t

Used by: SNVT_sec_state

TYPE E_LON_sec_state_t :
{
eLON_SSE_NUL := -1,
eLON_SSE_OFF := 0,
eLON_SSE_ON := 1,
eLON_SSE_INHIBIT_RESET := 2,
eLON_SSE_INHIBIT := 3,
eLON_SSE_WALK_TEST_OFF := 4,
eLON_SSE_WALK_TEST_ON := 5,
eLON_SSE_TEST_MODE_OFF := 6,
eLON_SSE_TEST_MODE_ON := 7,
eLON_SSE_POLL_STATUS := 8,
eLON_SSE_POLL_STATE := 9,
eLON_SSE_CONFIRM_ALARM_RESET := 10,
eLON_SSE_CONFIRM_ALARM := 11,
eLON_SSE_CONFIRM_TAMPER_RESET := 12,
eLON_SSE_CONFIRM_TAMPER := 13,
eLON_SSE_CONFIRM_MAINTENANCE := 14,
eLON_SSE_CONFIRM_TROUBLE := 15,
eLON_SSE_CONFIRM_FAULT := 16,
eLON_SSE_CONFIRM_RECOVERED_SENSOR := 17,
eLON_SSE_CONFIRM_UNSUPPORTED := 18,
eLON_SSE_CONFIRM_UNSUPPORTED := 19
}
END_TYPE

eLON_SSE_NUL:
eLON_SSE_OFF:
eLON_SSE_ON:
eLON_SSE_INHIBIT_RESET:
eLON_SSE_INHIBIT:
eLON_SSE_WALK_TEST_OFF:
eLON_SSE_WALK_TEST_ON:
eLON_SSE_TEST_MODE_OFF:
eLON_SSE_TEST_MODE_ON:
eLON_SSE_POLL_STATUS:
eLON_SSE_POLL_STATE:
eLON_SSE_CONFIRM_ALARM_RESET:
eLON_SSE_CONFIRM_ALARM:
eLON_SSE_CONFIRM_TAMPER_RESET:
eLON_SSE_CONFIRM_TAMPER:
eLON_SSE_CONFIRM_MAINTENANCE:
eLON_SSE_CONFIRM_TROUBLE:
eLON_SSE_CONFIRM_FAULT:
eLON_SSE_CONFIRM_RECOVERED_SENSOR:
eLON_SSE_LOST_SENSOR:
eLON_SSE_CONFIRM_UNSUPPORTED:

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.1.57 E_LON_sec_status_t

Used by: SNVT_sec_status

TYPE E_LON_sec_status_t :
{
    eLON_SSS_NUL := -1,
    eLON_SSS_POWER_UP := 0,
    eLON_SSS_ALARM_RESET := 1,
    eLON_SSS_ALARM := 2,
    eLON_SSS_TAMPER_RESET := 3,
    eLON_SSS_TAMPER := 4,
    eLON_SSS_MAINTENANCE := 5,
    eLON_SSS_TROUBLE := 6,
    eLON_SSS_FAULT := 7,
    eLON_SSS_RECOVERED_SENSOR := 8,
    eLON_SSS_LOST_SENSOR := 9,
    eLON_SSS_POLL_ACTIVE := 10,
    eLON_SSS_POLL_INACTIVE := 11,
    eLON_SSS_POLL_TAMPER := 12,
    eLON_SSS_POLL_ON := 13,
    eLON_SSS_POLL_OFF := 14,
    eLON_SSS_POLL_INHIBIT := 15,
    eLON_SSS_POLL_TEST := 16,
    eLON_SSS_CONFIRM_OFF := 17,
    eLON_SSS_CONFIRM_ON := 18,
    eLON_SSS_CONFIRM_INHIBIT_RESET := 19,
    eLON_SSS_CONFIRM_INHIBIT := 20,
    eLON_SSS_CONFIRM_WAKE_TEST_OFF := 21,
    eLON_SSS_CONFIRM_WAKE_TEST_ON := 22,
    eLON_SSS_CONFIRM_TEST_MODE_OFF := 23,
    eLON_SSS_CONFIRM_TEST_MODE_ON := 24,
    _LON_SSS_CONFIRM_UNSUPPORTED := 25
} END_TYPE

eLON_SSS_NUL:
eLON_SSS_POWER_UP:
eLON_SSS_ALARM_RESET:
eLON_SSS_ALARM:
eLON_SSS_TAMPER_RESET:
eLON_SSS_TAMPER:
eLON_SSS_MAINTENANCE:
eLON_SSS_TROUBLE:
eLON_SSS_FAULT:
eLON_SSS_RECOVERED_SENSOR:
eLON_SSS_LOST_SENSOR:
eLON_SSS_POLL_ACTIVE:
eLON_SSS_POLL_INACTIVE:
eLON_SSS_POLL_TAMPER:
eLON_SSS_POLL_ON:
eLON_SSS_POLL_OFF:
eLON_SSS_POLL_INHIBIT:
eLON_SSS_POLL_TEST:
eLON_SSS_CONFIRM_OFF:
eLON_SSS_CONFIRM_ON:
eLON_SSS_CONFIRM_INHIBIT_RESET:
eLON_SSS_CONFIRM_INHIBIT:
eLON_SSS_CONFIRM_WALK_TEST_OFF:
eLON_SSS_CONFIRM_WALK_TEST_ON:
eLON_SSS_CONFIRM_TEST_MODE_OFF:
eLON_SSS_CONFIRM_TEST_MODE_ON:
eLON_SSS_CONFIRM_UNSUPPORTED:

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.1.58  E_LON_setting_t

Used by: SNVT_setting

TYPE E_LON_setting_t :
{
  eLON_SET_NUL := -1,
  eLON_SET_OFF := 0,
  eLON_SET_ON := 1,
  eLON_SET_DOWN := 2,
  eLON_SET_UP := 3,
  eLON_SET_STOP := 4,
  eLON_SET_STATE := 5
}
END_TYPE

eLON_SET_NUL: Invalid value
eLON_SET_OFF: Change state to off
eLON_SET_ON: Change state to on, restoring the last on setting
eLON_SET_DOWN: Decrease the setting by the offset supplied in the setting field
eLON_SET_UP: Increase the setting by the offset supplied in the setting field
eLON_SET_STOP: Stop any motion, for example for blinds
eLON_SET_STATE: Change the setting to the value specified
### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

#### 4.2.1.59 E_LON_switch_state_t

Used by: SNVT_switch_2

```plaintext
TYPE E_LON_switch_state_t :
{
    eLON_SW_NUL := -1,
    eLON_SW_SET_OFF := 0,
    eLON_SW_SET_ON := 1,
    eLON_SW_REPORT_OFF := 2,
    eLON_SW_REPORT_ON := 3,
    eLON_SW_TOGGLE_STATE := 4,
    eLON_SW_SET_LEVEL := 5,
    eLON_SW_INCREMENT_LEVEL := 6,
    eLON_SW_DECREMENT_LEVEL := 7,
    eLON_SW_REPORT_OFF := 8,
    eLON_SW_RECALL_SCENE := 9,
    eLON_SW_STORE_SCENE := 10,
    eLON_SW_SET_OCCUPIED := 11,
    eLON_SW_SET_UNOCCUPIED := 12,
    eLON_SW_ENABLE_GROUP := 13,
    eLON_SW_DISABLE_GROUP := 14,
    eLON_SW_WINK := 15,
    eLON_SW_RESET := 16,
    eLON_SW_RESET_ENERGY_USAGE := 17,
    eLON_SW_RESET_RUNTIME := 18,
    eLON_SW_INCREASE_HUE := 19,
    eLON_SW_DECREASE_HUE := 20,
    eLON_SW_SET_BUTTON := 21,
    eLON_SW_SET_FAN_UP := 22,
    eLON_SW_SET_FAN_DOWN := 23,
    eLON_SW_TOGGLE_FAN_DIRECTION := 24,
    eLON_SW_INCREASE_FAN_LEVEL := 25,
    eLON_SW_DECREASE_FAN_LEVEL := 26,
    eLON_SW_SET_FAN_ON := 27,
    eLON_SW_SET_FAN_OFF := 28,
    eLON_SW_TOGGLE_FAN_STATE := 29,
    eLON_SW_MOVE_OPEN := 30,
    eLON_SW_MOVE_CLOSED := 31,
    eLON_SW_SET_ANGLE := 32,
    eLON_SW_ROTATE_OPEN := 33,
    eLON_SW_ROTATE_CLOSED := 34,
    eLON_SW_STOP := 35,
    eLON_SW_SET_STANDBY := 36,
    eLON_SW_TOGGLE_STANDBY := 37,
    eLON_SW_SET_POSITION := 38,
    eLON_SW_REPORT_POSITION := 39,
    eLON_SW_REPORT_FAN_LEVEL := 40
}
END_TYPE
```

**eLON_SW_NUL:** Invalid value

**eLON_SW_SET_OFF:** Set the state to off; ignored for blinds, drapes, shades, and fans

**eLON_SW_SET_ON:** Set the state to on; ignored for blinds, drapes, shades, and fans

**eLON_SW_REPORT_OFF:** Report that the state is off; output only; ignored for input

**eLON_SW_REPORT_ON:** Report that the state is on; output only; ignored for input

**eLON_SW_TOGGLE_STATE:** Toggle on-off state; same action as SW_SET_OFF if the on/off state was on, and SW_SET_ON if the on/off state was off; ignored for blinds, drapes, shades, and fans

**eLON_SW_SET_LEVEL:** Set the level to the specified value; ignored for blinds, drapes, shades, and fans

**eLON_SW_INCREMENT_LEVEL:** Increase the level by the specified value; ignored for blinds, drapes, shades, and fans
eLON_SW_DECREASE_LEVEL: Decrease the level by the specified amount; ignored for blinds, drapes, shades, and fans

eLON_SW_RECALL_SCENE: Recall the state and level from the specified scene

eLON_SW_STORE_SCENE: Store setting for the specified scene

eLON_SW_LEARN_SCENE: Learn setting for the specified scene

eLON_SW_SET_OCCUPIED: Set the occupancy state

eLON_SW_SET_UNOCCUPIED: Clear the occupancy state

eLON_SW_SET_MULTIPLIER: Set a multiplier for the level for 60 minutes; ignored for blinds, drapes, shades, and fans

eLON_SW_ENABLE_GROUP: Enable a group; all groups are enabled by default

eLON_SW_DISABLE_GROUP: Disable a group

eLON_SW_WINK: Blink state (toggle on-off state; pause; toggle on-off state again

eLON_SW_RESET: Reset scene definitions, multiplier, occupancy state, group enable flags, and settings to factory defaults

eLON_SW_RESET_ENERGY_USAGE: Reset energy usage value to zero

eLON_SW_RESET_RUNTIME: Reset runtime value to zero

eLON_SW_INCREASE_HUE: Increase color hue

eLON_SW_DECREASE_HUE: Decrease color hue

eLON_SW_SET_BUTTON: Trigger the actions for pressing and releasing the button specified in the value field

eLON_SW_SET_FAN_UP: Set ceiling fan direction to up, with specified level

eLON_SW_SET_FAN_DOWN: Set ceiling fan direction to down, with specified level

eLON_SW_TOGGLE_FAN_DIRECTION: Toggle fan up-down direction

eLON_SW_INCREASE_FAN_LEVEL: Increase fan speed by the setting

eLON_SW_DECREASE_FAN_LEVEL: Decrease fan speed by the setting

eLON_SW_SET_FAN_ON: Set the fan state to on

eLON_SW_SET_FAN_OFF: Set the fan state to off

eLON_SW_TOGGLE_FAN_STATE: Toggle the fan on-off state

eLON_SW_MOVE_OPEN: Move blinds, drapes, or shades open by the setting

eLON_SW_MOVE_CLOSED: Move blinds, drapes, or shades closed by the setting

eLON_SW_SET_ANGLE: Set the rotation angle of blinds to the setting

eLON_SW_ROTATE_OPEN: Rotate blinds open by the setting

eLON_SW_ROTATE_CLOSED: Rotate blinds closed by the setting

eLON_SW_STOP: Stop any motion of blinds, drapes, or shades

eLON_SW_SETSTANCEBY: Set Standby mode

eLON_SW_TOGGLESTANCEBY: Toggle the standby state

eLON_SW_SET_POSITION: Set blinds, drapes, or shades to the specified position; 100% is fully open, 0% is fully closed
eLON_SW_REPORT_POSITION: Report the position of blinds, drapes, or shades output only; ignored for input

eLON_SW_REPORT_FAN_LEVEL: Report the fan speed in percent of full level output only; ignored for input

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.2.1.60 E_LON_telcom_states_t

Used by: SNVT_telcom

```
TYPE E_LON_telcom_states_t :
{
  eLON_TEL_NUL       := -1,
  eLON_TEL_NOTINUSE  := 0,
  eLON_TEL_OFFHOOK   := 1,
  eLON_TEL_DIALING   := 2,
  eLON_TEL.DialCOMP  := 3,
  eLON_TEL_RINGBACK  := 4,
  eLON_TEL_INCOMING  := 5,
  eLON_TEL_RINGING   := 6,
  eLON_TEL_ANSWERED  := 7,
  eLON_TEL_CONNECTED := 8,
  eLON_TEL_TALKING   := 9,
  eLON_TEL_HANGINGUP := 10,
  eLON_TEL_HUNGUPX   := 11,
  eLON_TEL_HOLD      := 12,
  eLON_TEL_UNHOLD    := 13,
  eLON_TEL_RELEASE   := 14,
  eLON_TEL_FULLDUP   := 15,
  eLON_TEL_BLOCKED   := 16,
  eLON_TEL_CWAIT     := 17,
  eLON_TEL_DESTBUSY  := 18,
  eLON_TEL_NETBUSY   := 19,
  eLON_TEL_ERROR     := 20
}
END_TYPE
```

eLON_TEL_NUL: Invalid Value

eLON_TEL_NOTINUSE: "Null State (U0)" not in use

eLON_TEL_OFFHOOK: "Call Initiated (U1)"

eLON_TEL_DIALING: "Overlap Sending (U2)"

eLON_TEL.DialCOMP: "Outgoing Call Proceeding (U3)"

eLON_TEL_RINGBACK: "Call Delivered (U4)" hearing ringback

eLON_TEL_INCOMING: "Call Present (U6)" incoming call has not yet started ringing (only on ISDN line)

eLON_TEL_RINGING: "Call Received (U7)" incoming call when the user has indicated alerting but has not yet answered

eLON_TEL_ANSWERED: "Connect Request (U8)" user has answered the call and is waiting to be awarded the call

eLON_TEL_CONNECTED:

eLON_TEL_TALKING: "Active (U10)" two parties are exchanging data

eLON_TEL_HANGINGUP: "Disconnect Request (U11)" user has hung up

eLON_TEL_HUNGUPX: "Disconnect Indication (U12)" the other side hung up

eLON_TEL_HOLD: "Suspend Request (U15)" user has requested the network suspend the call
eLON_TEL_UNHOLD: "Resume Request (U17)" resume a held call (usually go back to TEL_TALKING)

eLON_TEL_RELEASE: "Release Request (U19)" user has requested the network to release

eLON_TEL_FULLDUP: "Overlap Receiving (U25)" user has acknowledged the call and is prepared to receive additional

eLON_TEL_BLOCKED: connection with blocking, (call-waiting disabled)

eLON_TEL_CWAIT: call-waiting coming in

eLON_TEL_DESTBUSY: destination busy

eLON_TEL_NETBUSY: problem, network

eLON_TEL_ERROR: problem, non-network

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.1.61 E_LON_therm_mode_t

Used by: SNVT_therm_mode

```
TYPE E_LON_therm_mode_t :
{
  eLON_THERM_NUL := -1,
  eLON_THERM_NO_CONTROL := 0,
  eLON_THERM_IN_OUT := 1,
  eLON_THERM_MODULATING := 2
}
END_TYPE
```

- eLON_THERM_NUL: Invalid Value
- eLON_THERM_NO_CONTROL: Thermostat disabled
- eLON_THERM_IN_OUT: Cut in/out control
- eLON_THERM_MODULATING: Modulating control

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.1.62 E_LON_tilt_dir_t

Used by: SNVT_plz

```
TYPE E_LON_tilt_dir_t :
{
  eLON_TILT_NUL := -1,
  eLON_TILT_STOP := 0,
  eLON_TILT_UP := 1,
  eLON_TILT_DOWN := 2
}
END_TYPE
```

- eLON_TILT_NUL: Invalid Value
- eLON_TILT_STOP: Stop tilting
- eLON_TILT_UP: Tilt up
- eLON_TILT_DOWN: Tilt down
Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.1.63 E_LON_unit_temp_t

Used by: SNVT_pump_sensor

```c
TYPE E_LON_unit_temp_t :
{
  eLON_TEMP_NUL := -1,
  eLON_TEMP_INACTIVE := 0,
  eLON_TEMP_AT_DESIRED := 1,
  eLON_TEMP_TOO_HOT := 2,
  eLON_TEMP_TOO_COLD := 3
}
END_TYPE
```

- **eLON_TEMP_NUL**: The status of the apparatus or unit is unknown, or not applicable (Invalid Value).
- **eLON_TEMP_INACTIVE**: The temperature-sensing apparatus is present, but not currently operating.
- **eLON_TEMP_AT_DESIRED**: The unit temperature is within the desired range.
- **eLON_TEMP_TOO_HOT**: The unit temperature is above the upper limit of the desired range.
- **eLON_TEMP_TOO_COLD**: The unit temperature is below the lower limit of the desired range.

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.1.64 E_LON_valve_mode_t

Used by: SNVT_valve_mode

```c
TYPE E_LON_valve_mode_t :
{
  eLON_VALVE_NUL := -1,
  eLON_VALVE_NORMAL := 0,
  eLON_VALVE_COOLING := 1,
  eLON_VALVE_HEATING := 2,
  eLON_VALVE_EMERGENCY := 3,
  eLON_VALVE_STROKE_ADP := 4,
  eLON_VALVE_STROKE_SYN := 5,
  eLON_VALVE_ERROR := 6,
  eLON_VALVE_OVERRIDDEN := 7
}
END_TYPE
```

- **eLON_VALVE_NUL**: Invalid value
- **eLON_VALVE_NORMAL**: Valve works as normal valve
- **eLON_VALVE_COOLING**: Valve works as cooling valve only
- **eLON_VALVE_HEATING**: Valve works as heating valve only
- **eLON_VALVE_EMERGENCY**: Valve works in emergency operation
- **eLON_VALVE_STROKE_ADP**: Valve adapt its stroke and its end positions
- **eLON_VALVE_STROKE_SYN**: Valve resynchronizes its position
- **eLON_VALVE_ERROR**: Valve is in error mode
- **eLON_VALVE_OVERRIDDEN**: Value is overridden
Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.1.65 E_LON_zoom_t

Used by: SNVT_ptz

```cpp
TYPE E_LON_zoom_t :
    eLON_ZOOM_NUL := -1,
    eLON_ZOOM_STOP := 0,
    eLON_ZOOM_TELE := 1,
    eLON_ZOOM_WIDE := 2
END_TYPE
```

eLON_ZOOM_NUL: Invalid Value

eLON_ZOOM_STOP: Stop zooming

eLON_ZOOM_TELE: Telephoto zoom / zoom in

eLON_ZOOM_WIDE: Wide zoom / zoom out

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2 Structures

AuxiliaryStructure/SNVT_chlr_status

**Data types**

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_chlr_state [558]</td>
</tr>
</tbody>
</table>

Used by: SNVT_chlr_status

AuxiliaryStructure/SNVT_clothes_w_c

**Data types**

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_action [558]</td>
</tr>
<tr>
<td>ST_LON_dry [559]</td>
</tr>
<tr>
<td>ST_LON_duration [559]</td>
</tr>
<tr>
<td>ST_LON_function [559]</td>
</tr>
<tr>
<td>ST_LON_rinse [560]</td>
</tr>
<tr>
<td>ST_LON_spin [560]</td>
</tr>
<tr>
<td>ST_LON_wash [560]</td>
</tr>
</tbody>
</table>

Used by: SNVT_clothes_w_c

AuxiliaryStructure/SNVT_clothes_w_s

**Data types**

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_alarm [561]</td>
</tr>
</tbody>
</table>

Used by: SNVT_clothes_w_s
AuxiliaryStructure/SNVT_color_2

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_CIE1931_lumen [p.563]</td>
<td>Used by: SNVT_color_2</td>
</tr>
<tr>
<td>ST_LON_CIE1931_percent [p.563]</td>
<td>Used by: SNVT_color_2</td>
</tr>
<tr>
<td>ST_LON_color_value [p.564]</td>
<td>Used by: SNVT_color_2</td>
</tr>
<tr>
<td>ST_LON_RGB [p.564]</td>
<td>Used by: SNVT_color_2</td>
</tr>
</tbody>
</table>

AuxiliaryStructure/SNVT_ctrl_resp

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_range [p.565]</td>
<td>Used by: SNVT_ctrl_resp</td>
</tr>
<tr>
<td>ST_LON_sender [p.565]</td>
<td>Used by: SNVT_ctrl_resp</td>
</tr>
</tbody>
</table>

AuxiliaryStructure/SNVT_dev_fault

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_Dev_type1 [p.565]</td>
<td>Used by: SNVT_dev_fault</td>
</tr>
<tr>
<td>ST_LON_pump_ctrl1 [p.566]</td>
<td>Used by: SNVT_dev_fault</td>
</tr>
<tr>
<td>ST_LON_valve_pos1 [p.567]</td>
<td>Used by: SNVT_dev_fault</td>
</tr>
</tbody>
</table>

AuxiliaryStructure/SNVT_dev_maint

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_Dev_type2 [p.568]</td>
<td>Used by: SNVT_dev_maint</td>
</tr>
<tr>
<td>ST_LON_pump_ctrl2 [p.568]</td>
<td>Used by: SNVT_dev_maint</td>
</tr>
<tr>
<td>ST_LON_valve_pos2 [p.569]</td>
<td>Used by: SNVT_dev_maint</td>
</tr>
</tbody>
</table>

AuxiliaryStructure/SNVT_dev_status

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_Dev_type3 [p.570]</td>
<td>Used by: SNVT_dev_status</td>
</tr>
<tr>
<td>ST_LON_pump_ctrl3 [p.570]</td>
<td>Used by: SNVT_dev_status</td>
</tr>
<tr>
<td>ST_LON_valve_pos3 [p.571]</td>
<td>Used by: SNVT_dev_status</td>
</tr>
</tbody>
</table>

AuxiliaryStructure/SNVT_ex_control

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_Control_device_addr [p.572]</td>
<td>Used by: SNVT_ex_control</td>
</tr>
</tbody>
</table>

AuxiliaryStructure/SNVT_file_req

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_addrt [p.572]</td>
<td>Used by: SNVT_file_req</td>
</tr>
<tr>
<td>ST_LON_dest_address [p.573]</td>
<td>Used by: SNVT_file_req</td>
</tr>
<tr>
<td>ST_LON_gp [p.573]</td>
<td>Used by: SNVT_file_req</td>
</tr>
<tr>
<td>ST_LON_sn [p.573]</td>
<td>Used by: SNVT_file_req</td>
</tr>
</tbody>
</table>
## AuxiliaryStructure/SNVT_file_status

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_address</td>
<td>Used by: FB_Write_Address_Table / FB_Read_Address_Table</td>
</tr>
<tr>
<td>ST_LON_addr</td>
<td>Used by: SNVT_file_status</td>
</tr>
<tr>
<td>ST_LON_descriptor</td>
<td>Used by: SNVT_file_status</td>
</tr>
</tbody>
</table>

## AuxiliaryStructure/SNVT_lamp_status

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_Alarm_actual</td>
<td>Used by: SNVT_lamp_status</td>
</tr>
<tr>
<td>ST_LON_alarm_previous</td>
<td>Used by: SNVT_lamp_status</td>
</tr>
</tbody>
</table>

## AuxiliaryStructure/SNVT_pos_ctrl

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_abspos</td>
<td>Used by: SNVT_pos_ctrl</td>
</tr>
<tr>
<td>ST_LON_Value</td>
<td>Used by: SNVT_pos_ctrl</td>
</tr>
</tbody>
</table>

## AuxiliaryStructure/SNVT_rac_ctrl

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_addr_dest</td>
<td>Used by: SNVT_rac_req / SNVT_rac_ctrl</td>
</tr>
<tr>
<td>ST_LON_addr_init</td>
<td>Used by: SNVT_rac_req / SNVT_rac_ctrl</td>
</tr>
<tr>
<td>ST_LON_addr_talk</td>
<td>Used by: SNVT_rac_ctrl</td>
</tr>
<tr>
<td>ST_LON_p2m</td>
<td>Used by: SNVT_rac_req / SNVT_rac_ctrl</td>
</tr>
<tr>
<td>ST_LON_p2p</td>
<td>Used by: SNVT_rac_req / SNVT_rac_ctrl</td>
</tr>
</tbody>
</table>

## AuxiliaryStructure/SNVT_rac_req

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_rac_req_addr_dest</td>
<td></td>
</tr>
<tr>
<td>ST_LON_rac_req_addr_init</td>
<td></td>
</tr>
</tbody>
</table>

## AuxiliaryStructure/SNVT_switch_2

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_setting</td>
<td>Used by: SNVT_switch_2</td>
</tr>
</tbody>
</table>

## AuxiliaryStructure/SNVT_time_zone

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_end_DST</td>
<td>Used by: SNVT_time_zone</td>
</tr>
<tr>
<td>ST_LON_M_end_DST</td>
<td>Used by: SNVT_time_zone</td>
</tr>
<tr>
<td>ST_LON_M_start_DST</td>
<td>Used by: SNVT_time_zone</td>
</tr>
<tr>
<td>ST_LON_start_DST</td>
<td>Used by: SNVT_time_zone</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_KL6401</td>
<td>Structure for configuration</td>
</tr>
<tr>
<td>ST_LON_AddressTable</td>
<td>Used by: FB_Write_Address_Table / FB_Read_Address_Table</td>
</tr>
<tr>
<td>Data types</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>---------------------------------------</td>
</tr>
<tr>
<td>ST_LON_ConfigTable [590]</td>
<td>Used by: FB_Write_Config_Table /</td>
</tr>
<tr>
<td></td>
<td>FB_Read_Config_Table</td>
</tr>
<tr>
<td>ST_LON_DomainTable [591]</td>
<td>Used by: FB_Write_Domain_Table /</td>
</tr>
<tr>
<td></td>
<td>FB_Read_Domain_Table</td>
</tr>
<tr>
<td>ST_LON_SNVT_alarm [591]</td>
<td>Used by: SNVT_alarm</td>
</tr>
<tr>
<td>ST_LON_SNVT_alarm_2 [592]</td>
<td>Used by: SNVT_alarm_2</td>
</tr>
<tr>
<td>ST_LON_SNVT_chlr_status [593]</td>
<td>Used by: SNVT_chlr_status</td>
</tr>
<tr>
<td>ST_LON_SNVT_clothes_w_a [593]</td>
<td>Used by: SNVT_clothes_w_a</td>
</tr>
<tr>
<td>ST_LON_SNVT_clothes_w_c [595]</td>
<td>Used by: SNVT_clothes_w_c</td>
</tr>
<tr>
<td>ST_LON_SNVT_clothes_w_m [595]</td>
<td>Used by: SNVT_clothes_w_m</td>
</tr>
<tr>
<td>ST_LON_SNVT_clothes_w_s [595]</td>
<td>Used by: SNVT_clothes_w_s</td>
</tr>
<tr>
<td>ST_LON_SNVT_color [596]</td>
<td>Used by: SNVT_color</td>
</tr>
<tr>
<td>ST_LON_SNVT_color_2 [596]</td>
<td>Used by: SNVT_color_2</td>
</tr>
<tr>
<td>ST_LON_SNVT_ctrl_req [596]</td>
<td>Used by: SNVT_ctrl_req</td>
</tr>
<tr>
<td>ST_LON_SNVT_ctrl_resp [597]</td>
<td>Used by: SNVT_ctrl_resp</td>
</tr>
<tr>
<td>ST_LON_SNVT_currency [597]</td>
<td>Used by: SNVT_currency</td>
</tr>
<tr>
<td>ST_LON_SNVT_date_event [598]</td>
<td>Used by: SNVT_date_event</td>
</tr>
<tr>
<td>ST_LON_SNVT_dev_fault [598]</td>
<td>Used by: SNVT_dev_fault</td>
</tr>
<tr>
<td>ST_LON_SNVT_dev_maint [598]</td>
<td>Used by: SNVT_dev_maint</td>
</tr>
<tr>
<td>ST_LON_SNVT_dev_status [598]</td>
<td>Used by: SNVT_dev_status</td>
</tr>
<tr>
<td>ST_LON_SNVT_earth_pos [599]</td>
<td>Used by: SNVT_earth_pos</td>
</tr>
<tr>
<td>ST_LON_SNVT_elapsed_tm [599]</td>
<td>Used by: SNVT_elapsed_tm</td>
</tr>
<tr>
<td>ST_LON_SNVT_ent_status [600]</td>
<td>Used by: SNVT_ent_status</td>
</tr>
<tr>
<td>ST_LON_SNVT_environment [601]</td>
<td>Used by: SNVT_environment</td>
</tr>
<tr>
<td>ST_LON_SNVT_ex_control [602]</td>
<td>Used by: SNVT_ex_control</td>
</tr>
<tr>
<td>ST_LON_SNVT_file_pos [602]</td>
<td>Used by: SNVT_file_pos</td>
</tr>
<tr>
<td>ST_LON_SNVT_file_req [602]</td>
<td>Used by: SNVT_file_req</td>
</tr>
<tr>
<td>ST_LON_SNVT_file_status [603]</td>
<td>Used by: SNVT_file_status</td>
</tr>
<tr>
<td>ST_LON_SNVT_geo_loc [604]</td>
<td>Used by: SNVT_geo_loc</td>
</tr>
<tr>
<td>ST_LON_SNVT_hvac_overid [604]</td>
<td>Used by: SNVT_hvac_overid</td>
</tr>
<tr>
<td>ST_LON_SNVT_hvac_satsts [604]</td>
<td>Used by: SNVT_hvac_satsts</td>
</tr>
<tr>
<td>ST_LON_SNVT_hvac_status [605]</td>
<td>Used by: SNVT_hvac_status</td>
</tr>
<tr>
<td>ST_LON_SNVT_lamp_status [606]</td>
<td>Used by: SNVT_lamp_status</td>
</tr>
<tr>
<td>ST_LON_SNVT_log_fx_request [606]</td>
<td>Used by: SNVT_log_fx_request</td>
</tr>
<tr>
<td>ST_LON_SNVT_log_fx_status [607]</td>
<td>Used by: SNVT_log_fx_status</td>
</tr>
<tr>
<td>ST_LON_SNVT_log_status [607]</td>
<td>Used by: SNVT_log_status</td>
</tr>
<tr>
<td>ST_LON_SNVT_muldiv [608]</td>
<td>Used by: SNVT_muldiv</td>
</tr>
<tr>
<td>ST_LON_SNVT_nv_type [608]</td>
<td>Used by: SNVT_nv_type</td>
</tr>
<tr>
<td>ST_LON_SNVT_obj_request [609]</td>
<td>Used by: SNVT_obj_request</td>
</tr>
<tr>
<td>ST_LON_SNVT_obj_status [609]</td>
<td>Used by: SNVT_obj_status</td>
</tr>
<tr>
<td>ST_LON_SNVT_pos_ctrl [610]</td>
<td>Used by: SNVT_pos_ctrl</td>
</tr>
<tr>
<td>ST_LON_SNVT_preset [611]</td>
<td>Used by: SNVT_preset</td>
</tr>
<tr>
<td>ST_LON_SNVT_privacyzone [611]</td>
<td>Used by: SNVT_privacyzone</td>
</tr>
<tr>
<td>ST_LON_SNVT_ptz [612]</td>
<td>Used by: SNVT_ptz</td>
</tr>
</tbody>
</table>
### Data types

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_SNVT_pump_sensor[612]</td>
<td>Used by: SNVT_pump_sensor</td>
</tr>
<tr>
<td>ST_LON_SNVT_pumpset_mn[613]</td>
<td>Used by: SNVT_pumpset_mn</td>
</tr>
<tr>
<td>ST_LON_SNVT_pumpset_sn[614]</td>
<td>Used by: SNVT_pumpset_sn</td>
</tr>
<tr>
<td>ST_LON_SNVT_rac_ctrl[614]</td>
<td>Used by: SNVT_rac_ctrl</td>
</tr>
<tr>
<td>ST_LON_SNVT_rac_req[615]</td>
<td>Used by: SNVT_rac_req</td>
</tr>
<tr>
<td>ST_LON_SNVT_rac_val[615]</td>
<td>Used by: SNVT_rac_val</td>
</tr>
<tr>
<td>ST_LON_SNVT_reg_val[615]</td>
<td>Used by: SNVT_reg_val</td>
</tr>
<tr>
<td>ST_LON_SNVT_reg_val_ts[616]</td>
<td>Used by: SNVT_reg_val_ts</td>
</tr>
<tr>
<td>ST_LON_SNVT_sblind_state[616]</td>
<td>Used by: SNVT_sblind_state</td>
</tr>
<tr>
<td>ST_LON_SNVT_scene[617]</td>
<td>Used by: SNVT_scene</td>
</tr>
<tr>
<td>ST_LON_SNVT_scene_cfg[617]</td>
<td>Used by: SNVT_scene_cfg</td>
</tr>
<tr>
<td>ST_LON_SNVT_setting[618]</td>
<td>Used by: SNVT_setting</td>
</tr>
<tr>
<td>ST_LON_SNVT_str_int[618]</td>
<td>Used by: SNVT_str_int</td>
</tr>
<tr>
<td>ST_LON_SNVT_switch[618]</td>
<td>Used by: SNVT_switch</td>
</tr>
<tr>
<td>ST_LON_SNVT_switch_2[619]</td>
<td>Used by: SNVT_switch_2</td>
</tr>
<tr>
<td>ST_LON_SNVT_temp_setpt[619]</td>
<td>Used by: SNVT_temp_setpt</td>
</tr>
<tr>
<td>ST_LON_SNVT_time_zone[620]</td>
<td>Used by: SNVT_time_zone</td>
</tr>
<tr>
<td>ST_LON_SNVT_tod_event[620]</td>
<td>Used by: SNVT_tod_event</td>
</tr>
<tr>
<td>ST_LON_SNVT_trans_table[621]</td>
<td>Used by: SNVT_trans_table</td>
</tr>
<tr>
<td>ST_LON_SNVT_zerospan[621]</td>
<td>Used by: SNVT_zerospan</td>
</tr>
<tr>
<td>str_AddressTable[621]</td>
<td>adress table</td>
</tr>
</tbody>
</table>

### 4.2.2.1 AuxiliaryStructure

#### SNVT_chlr_status

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_chlr_state[558]</td>
<td>Used by: SNVT_chlr_status</td>
</tr>
</tbody>
</table>

#### SNVT_clothes_w_c

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_action[558]</td>
<td>Used by: SNVT_clothes_w_c</td>
</tr>
<tr>
<td>ST_LON_dry[559]</td>
<td>Used by: SNVT_clothes_w_c</td>
</tr>
<tr>
<td>ST_LON_duration[559]</td>
<td>Used by: SNVT_clothes_w_c</td>
</tr>
<tr>
<td>ST_LON_function[559]</td>
<td>Used by: SNVT_clothes_w_c</td>
</tr>
<tr>
<td>ST_LON_rinse[560]</td>
<td>Used by: SNVT_clothes_w_c</td>
</tr>
<tr>
<td>ST_LON_spin[560]</td>
<td>Used by: SNVT_clothes_w_c</td>
</tr>
<tr>
<td>ST_LON_wash[560]</td>
<td>Used by: SNVT_clothes_w_c</td>
</tr>
</tbody>
</table>

#### SNVT_clothes_w_s

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_alarm[561]</td>
<td>Used by: SNVT_clothes_w_s</td>
</tr>
</tbody>
</table>
### SNVT_color_2

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST LON CIE1931_lumen [563]</td>
<td>Used by: SNVT_color_2</td>
</tr>
<tr>
<td>ST LON CIE1931_percent [563]</td>
<td>Used by: SNVT_color_2</td>
</tr>
<tr>
<td>ST LON color_value [564]</td>
<td>Used by: SNVT_color_2</td>
</tr>
<tr>
<td>ST LON RGB [564]</td>
<td>Used by: SNVT_color_2</td>
</tr>
</tbody>
</table>

### SNVT_ctrl_resp

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST LON range [565]</td>
<td>Used by: SNVT_ctrl_resp</td>
</tr>
<tr>
<td>ST LON sender [565]</td>
<td>Used by: SNVT_ctrl_resp</td>
</tr>
</tbody>
</table>

### SNVT_dev_fault

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST LON Dev_type1 [565]</td>
<td>Used by: SNVT_dev_fault</td>
</tr>
<tr>
<td>ST LON pump_ctrl1 [566]</td>
<td>Used by: SNVT_dev_fault</td>
</tr>
<tr>
<td>ST LON valve_pos1 [567]</td>
<td>Used by: SNVT_dev_fault</td>
</tr>
</tbody>
</table>

### SNVT_dev_maint

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST LON Dev_type2 [568]</td>
<td>Used by: SNVT_dev_maint</td>
</tr>
<tr>
<td>ST LON pump_ctrl2 [568]</td>
<td>Used by: SNVT_dev_maint</td>
</tr>
<tr>
<td>ST LON valve_pos2 [569]</td>
<td>Used by: SNVT_dev_maint</td>
</tr>
</tbody>
</table>

### SNVT_dev_status

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST LON Dev_type3 [570]</td>
<td>Used by: SNVT_dev_status</td>
</tr>
<tr>
<td>ST LON pump_ctrl3 [570]</td>
<td>Used by: SNVT_dev_status</td>
</tr>
<tr>
<td>ST LON valve_pos3 [571]</td>
<td>Used by: SNVT_dev_status</td>
</tr>
</tbody>
</table>

### SNVT_ex_control

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST LON Control_device_addr [572]</td>
<td>Used by: SNVT_ex_control</td>
</tr>
</tbody>
</table>

### SNVT_file_req

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST LON addr [572]</td>
<td>Used by: SNVT_file_req</td>
</tr>
<tr>
<td>ST LON dest_address [573]</td>
<td>Used by: SNVT_file_req</td>
</tr>
<tr>
<td>ST LON gp [573]</td>
<td>Used by: SNVT_file_req</td>
</tr>
<tr>
<td>ST LON sn [573]</td>
<td>Used by: SNVT_file_req</td>
</tr>
</tbody>
</table>
### SNVT_file_status

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_address [574]</td>
<td>Used by: FB_Write_Address_Table / FB_Read_Address_Table</td>
</tr>
<tr>
<td>ST_LON_addr [574]</td>
<td>Used by: SNVT_file_status</td>
</tr>
<tr>
<td>ST_LON_descriptor [575]</td>
<td>Used by: SNVT_file_status</td>
</tr>
</tbody>
</table>

### SNVT_lamp_status

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_Alarm_actual [575]</td>
<td>Used by: SNVT_lamp_status</td>
</tr>
<tr>
<td>ST_LON_alarm_previous [577]</td>
<td>Used by: SNVT_lamp_status</td>
</tr>
</tbody>
</table>

### SNVT_pos_ctrl

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_abspos [579]</td>
<td>Used by: SNVT_pos_ctrl</td>
</tr>
<tr>
<td>ST_LON_Value [579]</td>
<td>Used by: SNVT_pos_ctrl</td>
</tr>
</tbody>
</table>

### SNVT_rac_ctrl

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_addr_dest [580]</td>
<td>Used by: SNVT_rac_req / SNVT_rac_ctrl</td>
</tr>
<tr>
<td>ST_LON_addr_init [580]</td>
<td>Used by: SNVT_rac_req / SNVT_rac_ctrl</td>
</tr>
<tr>
<td>ST_LON_addr_talk [581]</td>
<td>Used by: SNVT_rac_ctrl</td>
</tr>
<tr>
<td>ST_LON_p2m [581]</td>
<td>Used by: SNVT_rac_req / SNVT_rac_ctrl</td>
</tr>
<tr>
<td>ST_LON_p2p [582]</td>
<td>Used by: SNVT_rac_req / SNVT_rac_ctrl</td>
</tr>
</tbody>
</table>

### SNVT_rac_req

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_rac_req_addr_dest [582]</td>
<td></td>
</tr>
<tr>
<td>ST_LON_rac_req_addr_init [582]</td>
<td></td>
</tr>
</tbody>
</table>

### SNVT_switch_2

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_setting [583]</td>
<td>Used by: SNVT_switch_2</td>
</tr>
</tbody>
</table>

### SNVT_time_zone

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_end_DST [584]</td>
<td>Used by: SNVT_time_zone</td>
</tr>
<tr>
<td>ST_LON_M_end_DST [584]</td>
<td>Used by: SNVT_time_zone</td>
</tr>
<tr>
<td>ST_LON_M_start_DST [585]</td>
<td>Used by: SNVT_time_zone</td>
</tr>
<tr>
<td>ST_LON_start_DST [585]</td>
<td>Used by: SNVT_time_zone</td>
</tr>
</tbody>
</table>
4.2.2.1.1 SNVT_chlr_status

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_chlr_state</td>
<td>Used by: SNVT_chlr_status</td>
</tr>
</tbody>
</table>

**ST_LON_chlr_state**

Used by: SNVT_chlr_status

```plaintext
TYPE ST_LON_chlr_state :
  STRUCT
    bIn_alarm  : BOOL;
    bRun_enabled : BOOL;
    bLocal : BOOL;
    bLimited : BOOL;
    bChw_flow : BOOL;
    bCondw_flow : BOOL;
END_STRUCT
END_TYPE
```

- **bIn_alarm**: Alarm flag (boolean).
- **bRun_enabled**: Run-enabled flag (boolean).
- **bLocal**: Locally-controlled flag (boolean).
- **bLimited**: Limited-condition flag (boolean). Conditions may exist that prevent reaching the setpoint.
- **bChw_flow**: Chiller-water-flow flag (boolean).
- **bCondw_flow**: Condenser-water-flow flag (boolean).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.1.2 SNVT_clothes_w_c

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_action</td>
<td>Used by: SNVT_clothes_w_c</td>
</tr>
<tr>
<td>ST_LON_dry</td>
<td>Used by: SNVT_clothes_w_c</td>
</tr>
<tr>
<td>ST_LON_duration</td>
<td>Used by: SNVT_clothes_w_c</td>
</tr>
<tr>
<td>ST_LON_function</td>
<td>Used by: SNVT_clothes_w_c</td>
</tr>
<tr>
<td>ST_LON_rinse</td>
<td>Used by: SNVT_clothes_w_c</td>
</tr>
<tr>
<td>ST_LON_spin</td>
<td>Used by: SNVT_clothes_w_c</td>
</tr>
<tr>
<td>ST_LON_wash</td>
<td>Used by: SNVT_clothes_w_c</td>
</tr>
</tbody>
</table>

**ST_LON_action**

Used by: SNVT_clothes_w_c

```plaintext
TYPE ST_LON_action :
  STRUCT
    bPower_on  : BOOL;
    bRun_mode : BOOL;
    byRsvd2_7 : BYTE;
END_STRUCT
END_TYPE
```

- **bPower_on**: 
### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### ST_LON_dry

*Used by: SNVT_clothes_w_c*

```plaintext
TYPE ST_LON_dry :
STRUCT
  byTemp  : BYTE;
  stDuration : ST_LON_Duration;
END_STRUCT
END_TYPE
```

- **byTemp:** Min: 0 / Max: 1
- **stDuration:** (see [ST_LON_Duration](#))

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### ST_LON_duration

*Used by: SNVT_clothes_w_c*

```plaintext
TYPE ST_LON_duration :
STRUCT
  byTime   : BYTE;
  eDryness : E_LON_discrete_levels_t;
END_STRUCT
END_TYPE
```

- **byTime:** Min: 0 / Max: 255
- **eDryness:** (see [E_LON_discrete_levels_t](#))

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### ST_LON_function

*Used by: SNVT_clothes_w_c*

```plaintext
TYPE ST_LON_function :
STRUCT
  eProgram : E_LON_appl_cwp_t;
  stWash   : ST_LON_wash;
  stRinse  : ST_LON_rinse;
  stSpin   : ST_LON_spin;
  stDry    : ST_LON_dry;
END_STRUCT
END_TYPE
```

- **eProgram:** (see [E_LON_appl_cwp_t](#))

---

T1000  
Version: 1.5 
559
**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**ST_LON_rinse**

Used by: SNVT_clothes_w_c

```c
TYPE ST_LON_rinse :
STRUCT
    byTemp : BYTE;
    byRepeat : BYTE;
    eOption : E_LON_appl_rin_t;
END_STRUCT
END_TYPE
```

- **byTemp**: Min: 0 / Max: 255
- **byRepeat**: Min: 0 / Max: 9
- **eOption**: (see E_LON_appl_rin_t [507])

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**ST_LON_spin**

Used by: SNVT_clothes_w_c

```c
TYPE ST_LON_spin :
STRUCT
    uiSpeed : UINT;
    byTime : BYTE;
    eHold : E_LON_boolean_t;
END_STRUCT
END_TYPE
```

- **uiSpeed**: Min: 0 / Max: 65535
- **byTime**: Min: 0 / Max: 255
- **eHold**: (see E_LON_boolean_t [507])

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**ST_LON_wash**

Used by: SNVT_clothes_w_c
TYPE ST_LON_wash :
STRUCT
  eLoad_level  : E_LON_discrete_levels_t;
  byTemp       : BYTE;
  byTime       : BYTE;
  ePrewash     : E_LON_boolean_t;
END_STRUCT
END_TYPE

eLoad_level: (see E_LON_discrete_levels_t [ 516])

byTemp: Min: 0 / Max: 255

byTime: Min: 0 / Max: 255

ePrewash: (see E_LON_boolean_t [ 507])

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.1.3 SNVT_clothes_w_s

Data types | Description
---|---
ST_LON_alarm [ 561] | Used by: SNVT_clothes_w_s

ST_LON_alarm

Used by: SNVT_clothes_w_s

TYPE ST_LON_alarm :
STRUCT
  bAlarm_reset          : BOOL;
  bWar_water_supply     : BOOL;
  bWar_drain_slow       : BOOL;
  bWar_door_open        : BOOL;
  bWar_load_unbalanced  : BOOL;
  bWar_filter_cleaning  : BOOL;
  bWar_hoses_reversed  : BOOL;
  bWar_voltage_low      : BOOL;
  bWar_power_failure    : BOOL;
  bWar_drain_open       : BOOL;
  bWar_execute_fail     : BOOL;
  bWar_door_locked      : BOOL;
  bWar_service          : BOOL;
  bWar_rsrvd5           : BOOL;
  bWar_rsrvd6           : BOOL;
  bWar_rsrvd7           : BOOL;
  bErr_motor_stall      : BOOL;
  bErr_water_temp       : BOOL;
  bErr_pressure         : BOOL;
  bErr_overflow         : BOOL;
  bErr_water_heat       : BOOL;
  bErr_water_leak       : BOOL;
  bErr_motor_speed      : BOOL;
  bErr_wash_thermistor  : BOOL;
  bErr_dry_thermistor   : BOOL;
  bErr_dry_overheat     : BOOL;
  bErr_dry_fan          : BOOL;
  bErr_rsrvd4           : BOOL;
  bErr_rsrvd5           : BOOL;
  bErr_rsrvd6           : BOOL;
  bErr_rsrvd7           : BOOL;
  byErr_rsrvd0_7        : BYTE;
  byErr_manuf_code      : BYTE;
END_STRUCT
END_TYPE
bAlarm_reset:
bWar_water_supply:
bWar_drain_slow:
bWar_door_open:
bWar_load_unbalanced:
bWar_filter_cleaning:
bWar_hoses_reversed:
bWar_voltage_low:
bWar_power_failure:
bWar_drain_open:
bWar_execute_fail:
bWar_door_locked:
bWar_service:
bWar_rsrvd5:
bWar_rsrvd6:
bWar_rsrvd7:
bErr_motor_stall:
bErr_water_temp:
bErr_pressure:
bErr_overflow:
bErr_water_heat:
bErr_water_leak:
bErr_motor_speed:
bErr_wash_thermistor:
bErr_dry_thermistor:
bErr_dry_overheat:
bErr_dry_heating:
bErr_dry_fan:
bErr_rsrvd4:
bErr_rsrvd5:
bErr_rsrvd6:
bErr_rsrvd7:
byErr_rsrvd0_7:
byErr_manuf_code: Min: 0 / Max: 255
Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.1.4 SNVT_color_2

Data types

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_CIE1931_lumen [563]</td>
<td>Used by: SNVT_color_2</td>
</tr>
<tr>
<td>ST_LON_CIE1931_percent [563]</td>
<td>Used by: SNVT_color_2</td>
</tr>
<tr>
<td>ST_LON_color_value [564]</td>
<td>Used by: SNVT_color_2</td>
</tr>
<tr>
<td>ST_LON_RGB [564]</td>
<td>Used by: SNVT_color_2</td>
</tr>
</tbody>
</table>

**ST_LON_CIE1931_lumen**

Used by: SNVT_color_2

```plaintext
TYPE ST_LON_CIE1931_lumen :
STRUCT
  rX : REAL;
  rY : REAL;
  udiAbsolute_Y : UDINT;
END_STRUCT
END_TYPE
```

- **rX**: Min: 0.0 / Max: 0.740 / Invalid: 1.275 / CIE 1931 x value (CIE 1931 color space coordinate). CIE 1931 x-axis color value
- **rY**: Min: 0.0 / Max: 0.840 / Invalid: 1.275 / CIE 1931 y value (CIE 1931 color space coordinate). CIE 1931 y-axis color value
- **udiAbsolute_Y**: Min: 0 / Max: 6553400 / Invalid: 6553500 / Absolute luminance (lumen). Absolute luminance

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**ST_LON_CIE1931_percent**

Used by: SNVT_color_2

```plaintext
TYPE ST_LON_CIE1931_percent :
STRUCT
  rX : REAL;
  rY : REAL;
  rPercent_Y : REAL;
END_STRUCT
END_TYPE
```

- **rX**: Min: 0.0 / Max: 0.740 / Invalid: 1.275 / CIE 1931 x value (CIE 1931 color space coordinate). CIE 1931 x-axis color value
- **rY**: Min: 0.0 / Max: 0.840 / Invalid: 1.275 / CIE 1931 y value (CIE 1931 color space coordinate). CIE 1931 y-axis color value
- **rPercent_Y**: Min: 0.0 / Max: 100.0 / Invalid: 655.35 / Luminance (% of full level). Y output in percent of maximum lumen output of the lamp
Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**ST_LON_color_value**

Used by: SNVT_color_2

```plaintext
TYPE ST_LON_color_value :
  STRUCT
    stCIE1931_lumen : ST_LON_CIE1931_lumen;
    stCIE1931_percent : ST_LON_CIE1931_percent;
    stRGB : ST_LON_RGB;
    uiColor_temperature : UINT;
  END_STRUCT
END_TYPE
```

- **stCIE1931_lumen**: CIE 1931 color space with lumen. CIE 1931 color space with Y output in lumen (see ST_LON_CIE1931_lumen [563]).
- **stCIE1931_percent**: CIE 1931 color space with percent. CIE 1931 color space with Y output in percent of maximum lumen output of the lamp (see ST_LON_CIE1931_percent [563]).
- **stRGB**: RGB color value (see ST_LON_RGB [564]).
- **uiColor_temperature**: Min: 2800 / Max: 7500 / Invalid: 12750 / Color temperature (degrees Kelvin).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**ST_LON_RGB**

Used by: SNVT_color_2

```plaintext
TYPE ST_LON_RGB :
  STRUCT
    byRed : BYTE;
    byGreen : BYTE;
    byBlue : BYTE;
  END_STRUCT
END_TYPE
```

- **byRed**: Min: 0 / Max: 250 / Red component. Red component for RGB color
- **byGreen**: Min: 0 / Max: 250 / Green component. Green component for RGB color
- **byBlue**: Min: 0 / Max: 250 / Blue component. Blue component for RGB color

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.2.2.1.5   SNVT_ctrl_resp**

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_range</td>
<td>Used by: SNVT_ctrlResp</td>
</tr>
<tr>
<td>ST_LON_sender</td>
<td>Used by: SNVT_ctrlResp</td>
</tr>
</tbody>
</table>
ST_LON_range

Used by: SNVT_ctrl_resp

```
TYPE ST_LON_range :
  STRUCT
    uiLower : UINT;
    uiUpper : UINT;
  END_STRUCT
END_TYPE
```

**uiLower:** Min: 1 / Max: 65535 / Invalid: 65535 / Sender range lower ID (ID number).

**uiUpper:** Min: 1 / Max: 65535 / Invalid: 65535 / Sender range upper ID (ID number).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

ST_LON_sender

Used by: SNVT_ctrl_resp

```
TYPE ST_LON_sender :
  STRUCT
    uiID    : UINT;
    stRange : ST_LON_range;
  END_STRUCT
END_TYPE
```

**uiID:** Min: 1 / Max: 65535 / Invalid: 65535 / Sender ID (ID number).

**stRange:** Sender ID range (lower, upper) (see ST_LON_range).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.1.6 SNVT_dev_fault

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_Dev_type1</td>
<td>Used by: SNVT_dev_fault</td>
</tr>
<tr>
<td>ST_LON_pump_ctrl1</td>
<td>Used by: SNVT_dev_fault</td>
</tr>
<tr>
<td>ST_LON_valve_pos1</td>
<td>Used by: SNVT_dev_fault</td>
</tr>
</tbody>
</table>

ST_LON_Dev_type1

Used by: SNVT_dev_fault

```
TYPE ST_LON_Dev_type1 :
  STRUCT
    stPump_ctrl : ST_LON_pump_ctrl1;
    stValvePos  : ST_LON_valve_pos1;
  END_STRUCT
END_TYPE
```

**stPump_ctrl:** Pump controller device fault information (see ST_LON_pump_ctrl1).

**stValvePos:** Valve positioner device fault information (see ST_LON_valve_pos1).
Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**ST_LON_pump_ctrl1**

Used by: SNVT_dev_fault

```plaintext
TYPE ST_LON_pump_ctrl1 :

  STRUCT
    bSf_voltage_low       : BOOL;
    bSf_voltage_high      : BOOL;
    bSf_phase             : BOOL;
    bSf_no_fluid          : BOOL;
    bSf_press_low         : BOOL;
    bSf_press_high        : BOOL;
    bSf_generalFault      : BOOL;
    bSf_reserved1_7       : BOOL;
    bDf_motor_temp        : BOOL;
    bDf_motor_failure     : BOOL;
    bDf_pump_blocked      : BOOL;
    bDf_elect_temp        : BOOL;
    bDf_elect_failure_nf  : BOOL;
    bDf_elect_failure     : BOOL;
    bDf_sensor_failure    : BOOL;
    bDf_generalFault      : BOOL;
    byReserved3_0_7       : BYTE;
  END_STRUCT

END_TYPE
```

- **bSf_voltage_low**: Supply fault - low voltage (boolean). Supply voltage is too low.
- **bSf_voltage_high**: Supply fault - high voltage (boolean). Supply voltage is too high.
- **bSf_phase**: Supply fault - power phase (boolean). Supply power is missing phase.
- **bSf_no_fluid**: Supply fault - no fluid (boolean). There is no fluid in the pump.
- **bSf_press_low**: Supply fault - low pressure (boolean). System pressure is too low.
- **bSf_press_high**: Supply fault - high pressure (boolean). System pressure is too high.
- **bSf_general_fault**: General supply fault.
- **bSf_reserved1_7**: 
- **bDf_motor_temp**: Device fault - motor temperature (boolean). Motor temperature is too high.
- **bDf_motor_failure**: Device fault - motor fatal failure (boolean). Motor has encountered a fatal failure.
- **bDf_pump_blocked**: Device fault - pump blocked (boolean). Pump is presently blocked.
- **bDf_elect_temp**: Device fault - electronics temperature (boolean). Temperature of the electronic circuitry is too high.
- **bDf_elect_failure_nf**: Device fault - electronics failure (boolean). Electronic circuitry has encountered a non-fatal failure.
- **bDf_elect_failure**: Device fault - electronics fatal failure (boolean). Electronic circuitry has encountered a fatal failure.
- **bDf_sensor_failure**: Device fault - sensor failure (boolean). Sensor has failed on the device.
- **bDf_general_fault**: General device fault.
- **byReserved3_0_7**: 


**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**ST_LON_valve_pos1**

Used by: SNVT_dev_fault

TYPE ST_LON_valve_pos1 :

STRUCT
    bDf_valve_blocked           : BOOL;
    bDf_blocked_direction_open  : BOOL;
    bDf_blocked_direction_close : BOOL;
    bDf_position_error          : BOOL;
    bDf_stroke_Out_of_range     : BOOL;
    bDf_initialization          : BOOL;
    bDf_vibration_cavitation    : BOOL;
    bDf_edTooHigh               : BOOL;
    byReserved1_0_2             : BYTE;
    bEe_oscillating             : BOOL;
    bEe_valve_too_large         : BOOL;
    bEe_valve_too_small         : BOOL;
    byReserved2_6_7             : BYTE;
    bReserved3_0_7              : BOOL;
    bSf_voltage_Out_of_range    : BOOL;
    bSf_electronic_high_temp    : BOOL;
    bSf_frictional_resistance   : BOOL;
    byReserved4_4_6             : BYTE;
    bGeneral_fault              : BOOL;
END_STRUCT
END_TYPE

**bDf_valve_blocked**: Device fault - valve blocked. The valve is presently blocked.

**bDf_blocked_direction_open**: Device fault - blocked direction open. The device is blocked while attempting to open.

**bDf_blocked_direction_close**: Device fault - blocked direction close. The device is blocked while attempting to close.

**bDf_position_error**: Device fault - position error. The valve position is not correct

**bDf_stroke_Out_of_range**: Device fault - stroke out of range. The valve stroke is out of operating range

**bDf_initialization**: Device fault - initialization error. The was an error during initialization of the device

**bDf_vibration_cavitation**: Device fault - vibration / cavitation. There are excessive vibrations or cavitations detected

**bDf_edTooHigh**: Device fault - ED too high. The ED is too high

**byReserved1_0_2**: This field is reserved.. This field is reserved.

**bEe_oscillating**: Engineering error - oscillating. There is an oscillating error

**bEe_valve_too_large**: Engineering error - valve too big. The valve size is too large

**bEe_valve_too_small**: Engineering error - valve too small. The valve size is too small

**byReserved2_6_7**: This field is reserved.. This field is reserved.

**bReserved3_0_7**: This field is reserved.. This field is reserved.

**bSf_voltage_Out_of_range**: Supply fault - voltage out of range. The voltage is out of the specified acceptable range

**bSf_electronic_high_temp**: Supply fault - electronics temperature. The temperature of the electronics is too high

**bSf_frictional_resistance**: Supply fault - frictional resistance. Resistance due to friction is detected
byReserved4_4_6: This field is reserved. This field is reserved.

bGeneral_fault: General Fault. A General Fault has occurred. Please consult the documentation or contact the valve-controller manufacturer.

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.2.2.1.7 SNVT_dev_maint

#### Data types

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_Dev_type2</td>
<td>Used by: SNVT_dev_maint</td>
</tr>
<tr>
<td>ST_LON_pump_ctrl2</td>
<td>Used by: SNVT_dev_maint</td>
</tr>
<tr>
<td>ST_LON_valve_pos2</td>
<td>Used by: SNVT_dev_maint</td>
</tr>
</tbody>
</table>

#### ST_LON_Dev_type2

Used by: SNVT_dev_maint

```plaintext
TYPE ST_LON_Dev_type2
    STRUCT
        stPump_ctrl : ST_LON_pump_ctrl2;
        stValvePos  : ST_LON_valve_pos2;
    END_STRUCT
END_TYPE
```

**stPump_ctrl:** Pump controller device maintenance state (see ST_LON_pump_ctrl2 [568]).

**stValvePos:** Valve positioner device maintenance information (see ST_LON_valve_pos2 [569]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

#### ST_LON_pump_ctrl2

Used by: SNVT_dev_maint

```plaintext
TYPE ST_LON_pump_ctrl2
    STRUCT
        bService_required   : BOOL;
        bBearings_change    : BOOL;
        bBearings_lubricate : BOOL;
        bShaftseal_change   : BOOL;
        byReserved1_4_7      : BYTE;
        byReserved2_0_7      : BYTE;
        byReserved3_0_7      : BYTE;
    END_STRUCT
END_TYPE
```

**bService_required:** Service required (boolean). Service/maintenance is required

**bBearings_change:** Change bearings (boolean). Bearings need to be replaced

**bBearings_lubricate:** Lubricate bearings (boolean). Bearings need to be greased

**bShaftseal_change:** Change shaft seal (boolean). Seal on the shaft needs to be replaced

**byReserved1_4_7:** Reserve
byReserved2_0_7: Reserve
byReserved3_0_7: Reserve

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**ST_LON_valve_pos2**

Used by: SNVT_dev_maint

TYPE ST_LON_valve_pos2 :
  STRUCT
    bMotor_maint       : BOOL;
    bPacking_change    : BOOL;
    bElectronics_check : BOOL;
    bPositioning_check : BOOL;
    bLubrication_check : BOOL;
    bReturn_check      : BOOL;
    bReserved2_0_7     : BYTE;
    byReserved3_0_7    : BYTE;
    bGeneral_maint     : BOOL;
  END_STRUCT
END_TYPE

bMotor_maint: Motor Maintenance. The motor requires servicing
bPacking_change: Packing Change. The packing needs to be controlled or changed
bElectronics_check: Check Electronics. The electronics need to be checked (temperature too high)
bPositioning_check: Check Position. The positioning needs to be checked (mechanical or electronic)
bLubrication_check: Check Lubrication. The lubrication need to be checked
bReturn_check: Check Spring-Return Function. The spring-return function needs to be checked
battery_check: Check battery. The battery needs to be checked
bReserved1_7: This field is reserved. This field is reserved.
byReserved2_0_7: This field is reserved. This field is reserved.
byReserved3_0_6: This field is reserved. This field is reserved.
bGeneral_maint: General Maintenance. General Maintenance needs to be performed. Please consult the documentation or your Maintenance Department.

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.1.8 **SNVT_dev_status**

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_Dev_type3 [570]</td>
<td>Used by: SNVT_dev_status</td>
</tr>
<tr>
<td>ST_LON_pump_ctrl3 [570]</td>
<td>Used by: SNVT_dev_status</td>
</tr>
<tr>
<td>ST_LON_valve_pos3 [571]</td>
<td>Used by: SNVT_dev_status</td>
</tr>
</tbody>
</table>
**ST_LON_Dev_type3**

Used by: SNVT_dev_status

```plaintext
TYPE ST_LON_Dev_type3 :
STRUCT
  stPump_ctrl : ST_LON_pump_ctrl3;
  stValvePos : ST_LON_valve_pos3;
END_STRUCT
END_TYPE
```

**stPump_ctrl**: Pump controller device status (see ST_LON_pump_ctrl3 [570]).

**stValvePos**: Valve positioner device status (see ST_LON_valve_pos3 [571]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**ST_LON_pump_ctrl3**

Used by: SNVT_dev_status

```plaintext
TYPE ST_LON_pump_ctrl3 :
STRUCT
  bDevice_fault : BOOL;
  bSupply_fault : BOOL;
  bReserved1_2 : BOOL;
  bSpeed_low    : BOOL;
  bSpeed_high   : BOOL;
  bReserved1_5 : BOOL;
  bSetpt_Out_of_range : BOOL;
  bReserved1_7 : BOOL;
  bLocal_control : BOOL;
  bReserved2_1 : BOOL;
  bRunning     : BOOL;
  bReserved2_3 : BOOL;
  bRemote_press : BOOL;
  bRemote_flow : BOOL;
  bRemote_temp : BOOL;
  bReserved2_7 : BOOL;
  byReserved3_0_7 : BYTE;
END_STRUCT
END_TYPE
```

**bDevice_fault**: Pump controller fault (boolean). See SNVT_pump_fault network variable declaration on device

**bSupply_fault**: Supply fault (boolean). No electrical power, no fluid in pump, etc. See SNVT_pump_fault network variable declaration on device.

**bReserved1_2**: 

**bSpeed_low**: Low-speed limit of pump (boolean). Pump is running at the lowest possible speed, therefore the requested performance is not possible.

**bSpeed_high**: High-speed limit of pump (boolean). Pump is running at the highest possible speed, therefore the requested performance is not possible.

**bReserved1_5**: 

**bSetpt_Out_of_range**: Setpoint out of range (boolean). Chosen override setpoint value is lower than the manufacturer-defined low-setpoint limit or higher than the manufacturer-defined high-setpoint limit.

**bReserved1_7**: 

**bLocal_control**: Locally controlled pump (boolean). Pump is locally operated (hardware override)

**bReserved2_1**: 

**bRunning**: 

**bReserved2_3**: 

**bRemote_press**: 

**bRemote_flow**: 

**bRemote_temp**: 

**bReserved2_7**: 

**byReserved3_0_7**:
bRunning: Running pump (boolean). Pump is presently running

bReserved2_3:

bRemote_press: Remote pressure sensor (boolean). Pump controller is using a remote pressure sensor

bRemote_flow: Remote flow sensor (boolean). Pump controller is using a remote flow sensor

bRemote_temp: Remote temperature sensor (boolean). Pump controller is using a remote temperature sensor

bReserved2_7:

byReserved3_0_7:

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**ST_LON_valve_pos3**

Used by: SNVT_dev_status

TYPE ST_LON_valve_pos3 :

STRUCT

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>bRunning</td>
<td>BOOL</td>
</tr>
<tr>
<td>bAdapting</td>
<td>BOOL</td>
</tr>
<tr>
<td>bInitializing</td>
<td>BOOL</td>
</tr>
<tr>
<td>bLocal_control</td>
<td>BOOL</td>
</tr>
<tr>
<td>bSetpt_Out_of_range</td>
<td>BOOL</td>
</tr>
<tr>
<td>bRemote_ctrl_signal</td>
<td>BOOL</td>
</tr>
<tr>
<td>byReserved1_6_7</td>
<td>BYTE</td>
</tr>
<tr>
<td>bHw_emergency</td>
<td>BOOL</td>
</tr>
<tr>
<td>bSw_emergency</td>
<td>BOOL</td>
</tr>
<tr>
<td>byReserved2_2_7</td>
<td>BYTE</td>
</tr>
<tr>
<td>byReserved3_0_7</td>
<td>BYTE</td>
</tr>
</tbody>
</table>

END_STRUCT

END_TYPE

bRunning: Valve Running. Valve is presently being positioned.

bAdapting: Adapting. Valve is presently adapting.

bInitializing: Initializing. Valve is presently initializing.

bLocal_control: Local Control. The valve operation is being locally controlled.

bSetpt_Out_of_range: Setpoint out of range. Chosen override setpoint value is lower than the manufacturer-defined low-setpoint limit or higher than the manufacturer-defined high-setpoint limit.

bRemote_ctrl_signal: Remote Control Signal. The remote-control signal is active.

byReserved1_6_7: This field is reserved. This field is reserved.

bHw_emergency: Hardware Emergency. The hardware-emergency state is active

bSw_emergency: Software Emergency. The software-emergency state is active

byReserved2_2_7: This field is reserved. This field is reserved.

byReserved3_0_7: This field is reserved. This field is reserved.

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
### 4.2.2.1.9 SNVT_ex_control

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_Control_device_addr</td>
<td>Used by: SNVT_ex_control</td>
</tr>
</tbody>
</table>

#### ST_LON_Control_device_addr

Used by: SNVT_ex_control

```plaintext
TYPE ST_LON_Control_device_addr :
STRUCT
  arrDomain_id : ARRAY [0..5] OF BYTE;
  byDomain_length : BYTE;
  bySubnet : BYTE;
  byNode : BYTE;
END_STRUCT
END_TYPE
```

- **arrDomain_id**: Domain ID (array of 6 bytes). ANSI/CEA-709.1 domain ID
- **byDomain_length**: Domain length (ANSI/CEA-709.1 domain length). Valid domain lengths are 0, 1, 3, and 6.
- **bySubnet**: Min: 1 / Max: 255 / Subnet (subnet number). There can be 255 subnets (1-255) in a domain.
- **byNode**: Min: 1 / Max: 127 / Node (node number). There can be 127 nodes (1-127) in a subnet.

#### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.2.2.1.10 SNVT_file_req

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_addrt [572]</td>
<td>Used by: SNVT_file_req</td>
</tr>
<tr>
<td>ST_LON_dest_address [573]</td>
<td>Used by: SNVT_file_req</td>
</tr>
<tr>
<td>ST_LON_gp [573]</td>
<td>Used by: SNVT_file_req</td>
</tr>
<tr>
<td>ST_LON_sn [573]</td>
<td>Used by: SNVT_file_req</td>
</tr>
</tbody>
</table>

#### ST_LON_addrt

Used by: SNVT_file_req

```plaintext
TYPE ST_LON_addrt :
STRUCT
  byType : BYTE;
  uiIndex : UINT;
END_STRUCT
END_TYPE
```

- **byType**: Min: 0 / Max: 33 / Address type (8-bit unsigned value). The address-table address type is 33 (0x21).
- **uiIndex**: Min: 0 / Max: 65535 / Address table index (16-bit unsigned value).

#### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
**ST_LON_dest_address**

Used by: SNVT_file_req

```plaintext
TYPE ST_LON_dest_address :
STRUCT
  stAddrt : ST_LON_addrt;
  stSn    : ST_LON_sn;
  stGp    : ST_LON_gp;
END_STRUCT
END_TYPE
```

**stAddrt**: Address table entry (Address table entry). ANSI/CEA-709.1 address in device's internal address table entry (see ST_LON_addrt [572]).

**stSn**: Subnet-node address (LonWorks subnet-node address) (see ST_LON_sn [573]).

**stGp**: Group address (LonWorks group address) (see ST_LON_gp [573]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**ST_LON_gp**

Used by: SNVT_file_req

```plaintext
TYPE ST_LON_gp :
STRUCT
  byType : BYTE;
  bySize : BYTE;
  bDomain : BOOL;
  byUnused : BYTE;
  byRetry : BYTE;
  byTx_timer : BYTE;
  byGroup : BYTE;
END_STRUCT
END_TYPE
```

**byType**: Min: 0 / Max: 1 / Address type (boolean). The group address type is 1.

**bySize**: Min: 0 / Max: 65 / Size (LonWorks group size). An acknowledged group can have from 0-64 addressees, plus the sender.

**bDomain**: Min: 0 / Max: 1 / Domain (LonWorks domain index).

**byUnused**: Unused field. This field is reserved.

**byRetry**: Min: 0 / Max: 15 / Retry count (number of retries).

**byTx_timer**: Min: 0 / Max: 15 / Transaction timer (timer code value).

**byGroup**: Min: 0 / Max: 255 / Group. There can be 256 groups (0-255) in a domain.

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**ST_LON_sn**

Used by: SNVT_file_req

```plaintext
TYPE ST_LON_sn :
STRUCT
  byType : BYTE;
```

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
byType: Min: 1 / Max: 1 / Address type (8-bit unsigned value). The subnet-node address type is 1.

bDomain: Domain (LonWorks domain index).

byNode: Min: 0 / Max: 127 / Node (node number). There can be 127 nodes (1-127) in a subnet.

byRetry: Min: 0 / Max: 15 / Retry count (number of retries).

byTx_timer: Min: 0 / Max: 15 / Transaction timer (timer code value).

bySubnet: Min: 0 / Max: 255 / Subnet (subnet number). There can be 255 subnets (1-255) in a domain.

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.11 SNVT_file_status

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_address [574]</td>
<td>Used by: FB_Write_Address_Table / FB_Read_Address_Table</td>
</tr>
<tr>
<td>ST_LON_adr [574]</td>
<td>Used by: SNVT_file_status</td>
</tr>
<tr>
<td>ST_LON_descriptor [575]</td>
<td>Used by: SNVT_file_status</td>
</tr>
</tbody>
</table>

ST_LON_address

Used by: FB_Write_Address_Table / FB_Read_Address_Table

TYPE ST_LON_address :
  STRUCT
    arrDomain_id : ARRAY [0..5] OF BYTE;
    byDomain_length : BYTE;
    bySubnet : BYTE;
    byNode : BYTE;
  END_STRUCT
END_TYPE

arrDomain_id: Domain ID (array of 6 bytes). ANSI/CEA-709.1 domain ID

byDomain_length: Min: 0 / Max: 6 / Domain length (ANSI/CEA-709.1 domain length). Valid domain lengths are 0, 1, 3, and 6.

bySubnet: Min: 0 / Max: 255 / Subnet (subnet number). There can be 255 subnets (1-255) in a domain.

byNode: Min: 0 / Max: 127 / Node (node number). There can be 127 nodes (1-127) in a subnet.

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

ST_LON_adr

Used by: SNVT_file_status
TYPE ST_LON_adr:
STRUCT
    stDescriptor : ST_LON_descriptor;
    stAddress : ST_LON_address;
END_STRUCT
END_TYPE

stDescriptor: Descriptor (see ST_LON_descriptor [575]).

stAddress: Address (see ST_LON_address [574]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

ST_LON_descriptor

Used by: SNVT_file_status

TYPE ST_LON_descriptor:
STRUCT
    sFile_info : STRING(16);
    udiSize : UDINT;
    uiType : UINT;
END_STRUCT
END_TYPE

sFile_info: File info (array of 16 characters)

udiSize: Min: 0 / Max: 2147483647 / Size (bytes)

uiType: Min: 0 / Max: 65535 / Type

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.1.12 SNVT_lamp_status

Data types

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_Alarm_actual [575]</td>
<td>Used by: SNVT_lamp_status</td>
</tr>
<tr>
<td>ST_LON_alarm_previous [577]</td>
<td>Used by: SNVT_lamp_status</td>
</tr>
</tbody>
</table>

ST_LON_Alarm_actual

Used by: SNVT_lamp_status

TYPE ST_LON_Alarm_actual:
STRUCT
    bLamp_current_high : BOOL;
    bLamp_current_low : BOOL;
    bMain_current_high : BOOL;
    bMain_current_low : BOOL;
    bLamp_voltage_high : BOOL;
    bLamp_voltage_low : BOOL;
    bMain_voltage_high : BOOL;
    bMain_voltage_low : BOOL;
    bOLC_temp_high : BOOL;
    bPower_factor_low : BOOL;
    bPower_high : BOOL;
    bPower_low : BOOL;
    bRelay_failure : BOOL;
    bCap_failure : BOOL;
bLamp_failure : BOOL;
bBallast_failure : BOOL;
bInter_com_failure : BOOL;
bExter_com_failure : BOOL;
bMain_volt_below_spec : BOOL;
bLamp_restart_count : BOOL;
bFading_ready : BOOL;
bBallast_temp_high : BOOL;
bDigi_in_A : BOOL;
bDigi_in_B : BOOL;
bBit_25_res : BOOL;
bBit_26_res : BOOL;
bBit_27_res : BOOL;
bBit_28_res : BOOL;
bBit_29_res : BOOL;
bBit_30_res : BOOL;
bBit_31_res : BOOL;
bBit_32_res : BOOL;
bBit_33_res : BOOL;
bBit_34_res : BOOL;
bBit_35_res : BOOL;
bBit_36_res : BOOL;
bBit_37_res : BOOL;
bBit_38_res : BOOL;
bBit_39_res : BOOL;
bBit_40_res : BOOL;
END_STRUCT
END_TYPE

bLamp_current_high: Lamp current too high.
bLamp_current_low: Lamp current too low.
bMain_current_high: Main current too high.
bMain_current_low: Main current too low.
bLamp_voltage_high: Lamp voltage too high.
bLamp_voltage_low: Lamp voltage too low.
bMain_voltage_high: Main voltage too high.
bMain_voltage_low: Main voltage too low.
bPowerfactor_low: Powerfactor too low.
bOLC_temp_high: Temperature too high.
bPower_high: Power too high.
bPower_low: Power too low.
bRelay_failure: Relay failure.
bCap_failure: Capacitor failure.
bLamp_failure: Lamp failure.
bBallast_failure: Ballast failure.
bInter_com_failure: Internal communication failure.
bExter_com_failur: External communication failure.
bMain_volt_below_spec: Main voltage below performance specification.
bLamp_restart_count: Lamp restart retry counter / cycling failure.
bFading_ready: Fading ready.
bBallast_temp_high: Ballast temperature too high.
bDigi_in_A: digital input A active.
bDigi_in_B: digital input B active.

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

ST_LON_alarm_previous

Used by: SNVT_lamp_status

```plaintext
TYPE ST_LON_alarm_previous :
STRUCT
  bLamp_current_high : BOOL;
  bLamp_current_low : BOOL;
  bMain_current_high : BOOL;
  bMain_current_low : BOOL;
  bLamp_voltage_high : BOOL;
  bLamp_voltage_low : BOOL;
  bMain_voltage_high : BOOL;
  bMain_voltage_low : BOOL;
  bPowerfactor_low : BOOL;
  bOLC_temp_high : BOOL;
  bPower_high : BOOL;
  bPower_low : BOOL;
  bRelay_failure : BOOL;
  bCap_failure : BOOL;
  bLamp_failure : BOOL;
  bBallast_failure : BOOL;
  bInter_com_failure : BOOL;
  bExter_com_failure : BOOL;
  bMain_volt_below_spec : BOOL;
  bLamp_restart_count : BOOL;
  bFading_ready : BOOL;
  bBallast_temp_high : BOOL;
  bDigi_in_A : BOOL;
  bDigi_in_B : BOOL;
  bBit_25_res : BOOL;
  bBit_26_res : BOOL;
  bBit_27_res : BOOL;
```
bBit_28_res : BOOL;
bBit_29_res : BOOL;
bBit_30_res : BOOL;
bBit_31_res : BOOL;
bBit_32_res : BOOL;
bBit_33_res : BOOL;
bBit_34_res : BOOL;
bBit_35_res : BOOL;
bBit_36_res : BOOL;
bBit_37_res : BOOL;
bBit_38_res : BOOL;
bBit_39_res : BOOL;
bBit_40_res : BOOL;
END_STRUCT
END_TYPE

bLamp_current_high: Lamp current too high.
bLamp_current_low: Lamp current too low.
bMain_current_high: Main current too high.
bMain_current_low: Main current too low.
bLamp_voltage_high: Lamp voltage too high.
bLamp_voltage_low: Lamp voltage too low.
bMain_voltage_high: Main voltage too high.
bMain_voltage_low: Main voltage too low.
bPowerfactor_low: Powerfactor too low.
bOLC_temp_high: Temperature too high.
bPower_high: Power too high.
bPower_low: Power too low.
bRelay_failure: Relay failure.
bCap_failure: Capacitor failure.
bLamp_failure: Lamp failure.
bBallast_failure: Ballast failure.
bInter_com_failure: Internal communication failure.
bExter_com_failure: External communication failure.
bMain_volt_below_spec: Main voltage below performance specification.
bLamp_restart_count: Lamp restart retry counter / cycling failure.
bFading_ready: Fading ready.
bBallast_temp_high: Ballast temperature too high.
bDigi_in_A: digital input A active.
bDigi_in_B: digital input B active.
bBit_25_res:
bBit_26_res:
bBit_27_res:
bBit_28_res:
bBit_29_res:
### Requirements

#### Development environment

| TwinCAT from v3.1.4020.14 | Tc2_LON from 3.3.4.0 |

### 4.2.2.1.13 SNVT_pos_ctrl

#### Data types

<table>
<thead>
<tr>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_abspos [579]</td>
<td>Used by: SNVT_pos_ctrl</td>
</tr>
<tr>
<td>ST_LON_Value [579]</td>
<td>Used by: SNVT_pos_ctrl</td>
</tr>
</tbody>
</table>

#### ST_LON_abspos

Used by: SNVT_pos_ctrl

**TYPE** ST_LON_abspos :

```plaintext
STRUCT
    rPan  : REAL;
    rTilt : REAL;
    rZoom : REAL;
END_STRUCT
END_TYPE
```

<table>
<thead>
<tr>
<th>rPan</th>
<th>Min: -359.98 / Max: 360 / Pan position.</th>
</tr>
</thead>
<tbody>
<tr>
<td>rTilt</td>
<td>Min: -359.98 / Max: 360 / Tilt position.</td>
</tr>
<tr>
<td>rZoom</td>
<td>Min: -163.84 / Max: 163.835 / Zoom position.</td>
</tr>
</tbody>
</table>

#### Requirements

| TwinCAT from v3.1.4020.14 | Tc2_LON from 3.3.4.0 |

### ST_LON_Value

Used by: SNVT_pos_ctrl

**TYPE** ST_LON_Value :

```plaintext
STRUCT
    byNumber : BYTE;
END_TYPE
```
byNumber: Min: 0 / Max: 255 / Action number (action number).

stAbspos: Function absolute values (pan, tilt, zoom) (see ST_LON_abspos [579]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.1.14 SNVT_rac_ctrl

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_addr_dest [580]</td>
<td>Used by: SNVT_rac_req / SNVT_rac_ctrl</td>
</tr>
<tr>
<td>ST_LON_addr_init [580]</td>
<td>Used by: SNVT_rac_req / SNVT_rac_ctrl</td>
</tr>
<tr>
<td>ST_LON_addr_talk [581]</td>
<td>Used by: SNVT_rac_ctrl</td>
</tr>
<tr>
<td>ST_LON_p2p [582]</td>
<td>Used by: SNVT_rac_req / SNVT_rac_ctrl</td>
</tr>
<tr>
<td>ST_LON_p2m [581]</td>
<td>Used by: SNVT_rac_req / SNVT_rac_ctrl</td>
</tr>
</tbody>
</table>

ST_LON_addr_dest

Used by: SNVT_rac_req / SNVT_rac_ctrl

```c
TYPE ST_LON_addr_dest :
STRUCT
    stP2p : ST_LON_p2p;
    stP2m : ST_LON_p2m;
END_STRUCT
END_TYPE
```

stP2p: (see ST_LON_p2p [582]).

stP2m: (see ST_LON_p2m [581]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

ST_LON_addr_init

Used by: SNVT_rac_req / SNVT_rac_ctrl

```c
TYPE ST_LON_addr_init :
STRUCT
    byUnit_id : BYTE;
    byLocation : BYTE;
    byCar_id : BYTE;
    byReserved : BYTE;
    eAudio_sensor_type : E_LON_rail_audio_sensor_type_t;
END_STRUCT
END_TYPE
```

byUnit_id: Min: 0 / Max: 15

byLocation: Min: 0 / Max: 15

byCar_id: Min: 0 / Max: 31
byReserved:

**eAudio_sensor_type**: (see `E_LON_rail_audio_sensor_type_t`).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**ST_LON_addr_talk**

Used by: `SNVT_rac_ctrl`

```plaintext
TYPE ST_LON_addr_talk :
  STRUCT
    byUnit_id : BYTE;
    byLocation : BYTE;
    byCar_id : BYTE;
    byReserved : BYTE;
    eAudio_sensor_type : E_LON_rail_audio_sensor_type_t;
  END_STRUCT
END_TYPE
```

**byUnit_id**: Min: 0 / Max: 15

**byLocation**: Min: 0 / Max: 15

**byCar_id**: Min: 0 / Max: 31

**byReserved**:  

**eAudio_sensor_type**: (see `E_LON_rail_audio_sensor_type_t`).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**ST_LON_p2m**

Used by: `SNVT_rac_req / SNVT_rac_ctrl`

```plaintext
TYPE ST_LON_p2m :
  STRUCT
    byMask_unit : BYTE;
    arrMask_car : ARRAY[0..3] OF BYTE;
    arrMask_location : ARRAY[0..1] OF BYTE;
    arrMask_audio : ARRAY[0..2] OF BYTE;
  END_STRUCT
END_TYPE
```

**byMask_unit**: Min: 0 / Max: 255

**arrMask_car**: unsigned char [4]

**arrMask_location**: unsigned char [2]

**arrMask_audio**: unsigned char [3]

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_L_ON from 3.3.4.0</td>
</tr>
</tbody>
</table>
**ST_LON_p2p**

**Used by:** SNVT_rac_req / SNVT_rac_ctrl

```c
TYPE ST_LON_p2p :
  STRUCT
    byUnit_id : BYTE;
    byLocation : BYTE;
    byCar_id : BYTE;
    byReserved : BYTE;
    eAudio_sensor_type : E_LON_rail_audio_sensor_type_t;
  END_STRUCT
  END_TYPE
```

- **byUnit_id:** Min: 0 / Max: 15
- **byLocation:** Min: 0 / Max: 15
- **byCar_id:** Min: 0 / Max: 31
- **byReserved:**
- **eAudio_sensor_type:** (see E_LON_rail_audio_sensor_type_t [534]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.1.15 **SNVT_rac_req**

**Data types**

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_rac_req_addr_dest [582]</td>
<td></td>
</tr>
<tr>
<td>ST_LON_rac_req_addr_init [582]</td>
<td></td>
</tr>
</tbody>
</table>

**ST_LON_rac_req_addr_dest**

```c
TYPE ST_LON_rac_req_addr_dest :
  STRUCT
    stP2p : ST_LON_p2p;
    stP2m : ST_LON_p2m;
  END_STRUCT
  END_TYPE
```

- **stP2p:** (see ST_LON_p2p [582]).
- **stP2m:** (see ST_LON_p2m [581]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

**ST_LON_rac_req_addr_init**

```c
TYPE ST_LON_rac_req_addr_init :
  STRUCT
    byUnit_id : BYTE;
    byLocation : BYTE;
    byCar_id : BYTE;
    byReserved : BYTE;
    eAudio_sensor_type : E_LON_rail_audio_sensor_type_t;
  END_STRUCT
  END_TYPE
```
byUnit_id:
byLocation:
byCar_id:
byReserved:

eAudio_sensor_type: (see $E\_LON\_rail\_audio\_sensor\_type\_t$ [534]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.1.16 SNVT_switch_2

Data types | Description
--- | ---
ST_LON_setting [583] | Used by: SNVT_switch_2

ST_LON_setting

Used by: SNVT_switch_2

```plaintext
TYPE ST_LON_setting :
STRUCT
  rValue : REAL;
  rChange : REAL;
  byDelay : BYTE;
  byGroup_number : BYTE;
  rMultiplier : REAL;
  iAngle : INT;
  siFan_level : SINT;
  byButton_number : BYTE;
END_STRUCT
END_TYPE
```

rValue: Min: 0 / Max: 100 / Value. Percent of full level when state is on. Reports last level for outputs when state is off.

rChange: Min: 0 / Max: 100 / Percent change. Percent change to level.

byDelay: Min: 0 / Max: 255 / On or off delay (seconds) (seconds). Time delay before changing state to on or off.

byGroup_number: Min: 0 / Max: 63 / Group number. Group number that is enabled or disabled by the SW_ENABLE_GROUP and SW_DISABLE_GROUP states in the state field; if 0, all groups are enabled or disabled.

rMultiplier: Min: 0 / Max: 2.54 / Factor (percent). Multiplier for the level

iAngle: Min: -180 / Max: 180 / Rotation angle (degrees). Rotation angle for devices that support a rotation setting such as blinds.

siFan_level: Min: -100 / Max: 100 / Fan level. Percent of full level fan speed when state is on. Reports last fan speed for outputs when state is off. Positive values represent the down direction, and negative values represent the up direction.

byButton_number: Min: 0 / Max: 255 / Button number. The button number to activate when the state field is set to SW_SET_BUTTON, no invalid value

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
### 4.2.2.1.17 SNVT_time_zone

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_LON_end_DST [584]</td>
<td>Used by: SNVT_time_zone</td>
</tr>
<tr>
<td>ST_LON_M_end_DST [584]</td>
<td>Used by: SNVT_time_zone</td>
</tr>
<tr>
<td>ST_LON_M_start_DST [585]</td>
<td>Used by: SNVT_time_zone</td>
</tr>
<tr>
<td>ST_LON_start_DST [585]</td>
<td>Used by: SNVT_time_zone</td>
</tr>
</tbody>
</table>

#### ST_LON_end_DST

Used by: SNVT_time_zone

```
TYPE ST_LON_end_DST :
STRUCT
  uiG_day_of_end_DST  : UINT;
  uiJ_day_of_end_DST  : UINT;
  stM_end_DST         : ST_LON_M_end_DST;
END_STRUCT
END_TYPE
```

- **uiG_day_of_end_DST**: Min: 0 / Max: 365 / Gregorian calendar day of end DST (days).
- **uiJ_day_of_end_DST**: Min: 1 / Max: 365 / Julian calendar day of end DST (days).
- **stM_end_DST**: Mu calendar day of end DST (month, week, dateday) (see ST_LON_M_end_DST [584]).

#### ST_LON_M_end_DST

Used by: SNVT_time_zone

```
TYPE ST_LON_M_end_DST :
STRUCT
  byMonth_of_end_DST   : BYTE;
  byWeek_of_end_DST    : BYTE;
  eDateday_of_end_DST  : E_LON_days_of_week_t;
END_STRUCT
END_TYPE
```

- **byMonth_of_end_DST**: Min: 1 / Max: 12 / Month of end DST (months).
- **byWeek_of_end_DST**: Min: 1 / Max: 5 / Week of end DST (weeks).
- **eDateday_of_end_DST**: Day of week (day names).

#### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### Development environment

- **TwinCAT** from v3.1.4020.14
- **Tc2_LON** from 3.3.4.0
ST_LON_M_start_DST

Used by: SNVT_time_zone

```
TYPE ST_LON_M_start_DST :
   STRUCT
      byMonth_of_start_DST   : BYTE;
      byWeek_of_start_DST    : BYTE;
      eDateday_of_start_DST  : E_LON_days_of_week_t;
   END_STRUCT
END_TYPE
```

- `byMonth_of_start_DST`: Min: 1 / Max: 12 / Month of start DST (months).
- `byWeek_of_start_DST`: Min: 1 / Max: 5 / Week of start DST (weeks).
- `eDateday_of_start_DST`: Day of week (day names) (see `E_LON_days_of_week_t`).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

ST_LON_start_DST

Used by: SNVT_time_zone

```
TYPE ST_LON_start_DST :
   STRUCT
      uiG_day_of_start_DST  : UINT;
      uiJ_day_of_start_DST  : UINT;
      stM_start_DST         : ST_LON_M_start_DST;
   END_STRUCT
END_TYPE
```

- `uiG_day_of_start_DST`: Min: 0 / Max: 365 / Gregorian calendar day of start DST (days).
- `uiJ_day_of_start_DST`: Min: 1 / Max: 365 / Julian calendar day of start DST (days).
- `stM_start_DST`: Meu calendar day of start DST (month, week, dateday) (see `ST_LON_M_start_DST`).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.2 ST_LON_Parameter_IN_36B

Process image of the inputs

This variable must be linked with the KL6401.

```
TYPE ST_LON_Parameter_IN_36B :
   STRUCT
      wParameterStatus    : WORD;
      wDummy              : WORD;
      stParameterReadValue: ST_LON_ParameterInterface;
      byParameterType     : BYTE;
      byLONStatus         : BYTE;
   END_STRUCT
END_TYPE
```
wParameterStatus: Status word.
wDummy: Dummy for CX9000.
stParameterReadValue: Data structure.
stParameterReadValue. arrParameterInterface: 8 word data (see ST_LON_ParameterInterface [588]).
byParameterType: Parameter type
byLONStatus: LON status.

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.3 ST_LON_Parameter_OUT_36B

Process image of the outputs

This variable must be linked with the KL6401.

TYPE ST_LON_Parameter_OUT_36B :
  STRUCT
    wParameterControl : WORD;
    wDummy : WORD;
    stParametervalue : ST_LON_ParameterInterface;
    byCMD : BYTE;
    byIdx : BYTE;
  END_STRUCT
END_TYPE

wParameterControl: Control word.
wDummy: Dummy for CX9000.
stParametervalue: Data structure.
stParametervalue. arrParameterInterface: 8 word data (see ST_LON_ParameterInterface [588]).
byCMD: LON command
byIdx: LON index

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.4 ST_ExplicitMessage

Explicit Message

TYPE ST_ExplicitMessage :
  STRUCT
    byEcpM_1 : BYTE;
    byEcpM_2 : BYTE;
    byLen : BYTE;
    byAddressingTyp : BYTE;
    byRetry : BYTE;
    byRptTimer : BYTE;
    byTxTimer : BYTE;
    byRcvTimer : BYTE;
    bySubNet : BYTE;
    byDestSubNet : BYTE;
    byNode : BYTE;
    arrNeuronId : ARRAY [0..7] OF BYTE;
  END_STRUCT
END_TYPE
### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

#### 4.2.2.5 ST_LON_Communication

Connection between "FB_LON_KL6401" and the send/receive function blocks

```plaintext
TYPE ST_LON_Communication :
  STRUCT
    arrWriteLONdata : ARRAY[0..iLONBufferSize] OF ST_LON_WriteData;
    bWrite : BOOL;
    bWriteBusy : BOOL;
    bReadBusy : BOOL;
    arrParameterReadValue : ARRAY[1..32] OF BYTE;
    wNV_Index : WORD;
    eParameterDataType : E_LON_Parameter_Datatypes;
    bWriteLONdataToTable : BOOL;
    bTerminalOk : BOOL;
    byActBuffer : BYTE;
    rActBuffer : REAL;
    rMaxBuffer : REAL;
  END_STRUCT
END_TYPE
```

- **arrWriteLONdata**: Send buffer.
- **bWrite**: Data is being sent.
- **bWriteBusy**: Write-OnChange active.
- **bReadBusy**: Read-OnChange active.
- **arrParameterReadValue**: 32 data bytes.
- **wNV_Index**: NV index.
eParameterDataType: Data type of the LON variable (SNVT) (see E_LON parameter data types [495]).

bWriteLONdataToTable: Writing of the data in the table is active.

bTerminalOk: Initialization of the KL6401 was completed successfully.

byActBuffer: Number of instructions in the send buffer.

rActBuffer: Current utilization of the send buffer in percent.

rMaxBuffer: Maximum utilization of the send buffer in percent. The value can be cancelled with the input variable “bResetMaxBuffer”.

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.6 ST_LON_ParameterInterface

LON parameter interface

```plaintext
TYPE ST_LON_ParameterInterface :
  STRUCT
    arrParameterInterface : ARRAY[1..8] OF DWORD;
  END_STRUCT
END_TYPE
arrParameterInterface: 8 word data.
```

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.7 ST_LON_WriteData

Structure of the send buffer

```plaintext
TYPE ST_LON_WriteData :
  STRUCT
    wNVIndex            : WORD;
    udiSrcAddrWriteValue : PVOID;
    uiLenWriteValue     : UINT;
    udiAdrBusy          : PVOID;
    udiAdrErrorKL       : PVOID;
  END_STRUCT
END_TYPE
```

wNVIndex: NV index.

udiSrcAddrWriteValue: Address of the value to be sent.

uiLenWriteValue: Length of the value to be sent.

udiAdrBusy: Address of output “bBusy”.

udiAdrErrorKL: Address of output “dwErrorKL”.

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.2.2.8      ST_Prm

Structure for configuration

```delphi
TYPE ST_Prm :
    STRUCT
        byCMD            : BYTE;
        byIDX           : BYTE;
        wControl        : WORD;
        wStatus         : WORD;
        wParameterControl : WORD;
        wError          : WORD;
        arrParameterInterface : ARRAY[1..8] OF DWORD;
    END_STRUCT
END_TYPE
```

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>byCMD:</td>
<td>LON Command.</td>
</tr>
<tr>
<td>byIDX:</td>
<td>LON Index.</td>
</tr>
<tr>
<td>wControl:</td>
<td>Control Word.</td>
</tr>
<tr>
<td>wStatus:</td>
<td>Status Word.</td>
</tr>
<tr>
<td>wParameterControl:</td>
<td>Control Word.</td>
</tr>
<tr>
<td>wError:</td>
<td>Error information.</td>
</tr>
<tr>
<td>arrParameterInterface:</td>
<td>8 Data Word.</td>
</tr>
</tbody>
</table>

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.9      ST_KL6401

Structure for the configuration

```delphi
TYPE ST_KL6401 :
    STRUCT
        wStatus            : WORD;
        wControl          : WORD;
        dwPointer_IN      : PVOID;
        dwPointer_OUT     : PVOID;
        arrParameterInterface : ARRAY[1..8] OF DWORD;
    END_STRUCT
END_TYPE
```

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wStatus:</td>
<td>Status Word.</td>
</tr>
<tr>
<td>wControl:</td>
<td>Control Word.</td>
</tr>
<tr>
<td>dwPointer_IN:</td>
<td>Pointer In.</td>
</tr>
<tr>
<td>dwPointer_OUT:</td>
<td>Pointer Out.</td>
</tr>
<tr>
<td>arrParameterInterface:</td>
<td>8 Word data.</td>
</tr>
</tbody>
</table>

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.10     ST_LON_AddressTable

Used by: FB_Write_Address_Table / FB_Read_Address_Table
TYPE ST_LON_AddressTable :
STRUCT
  bType : BOOL;
  usiNode : USINT;
  bDomain : BOOL;
  usiMember : USINT;
  usiRPT_Timer : USINT;
  usiRetry : USINT;
  usiRCV_Timer : USINT;
  usiTx_Timer : USINT;
  byGroup : BYTE;
END_STRUCT
END_TYPE

bType: Type.

usiNode: Node.

bDomain: Domain.

usiMember: Member.

usiRPT_Timer: RPT Timer.

usiRetry: Retry.

usiRCV_Timer: Rcv timer.

usiTx_Timer: Tx timer.

byGroup: group.

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.11 ST_LON_ConfigTable

Used by: FB_Write_Config_Table / FB_Read_Config_Table

TYPE ST_LON_ConfigTable :
STRUCT
  bPriority : BOOL;
  bDirection : BOOL;
  wSelector : WORD;
  bTurnaround : BOOL;
  usiService : USINT;
  bAuth : BOOL;
  usiAddrIndex : USINT;
END_STRUCT
END_TYPE

bPriority: priority.

bDirection: direction.

wSelector: selector.

bTurnaround: turn around.

usiService: service.

bAuth: auth.

usiAddrIndex: Address index.
Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.12 **ST_LON_DomainTable**

Used by: FB_Write_Domain_Table / FB_Read_Domain_Table

```plaintext
TYPE ST_LON_DomainTable :
    STRUCT
        arrDomainID : ARRAY[0..5] OF BYTE;
        bySubNet : BYTE;
        bCloneDomainBit : BOOL;
        byNode : BYTE;
        byLen : BYTE;
        arrKey : ARRAY[0..5] OF BYTE := 16#FF;
    END_STRUCT
END_TYPE
```

- **arrDomainID**: domain id.
- **bySubNet**: sub net.
- **bCloneDomainBit**: Clone domain bit.
- **byNode**: node.
- **byLen**: Len.
- **arrKey**: key.

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.13 **ST_LON_SNVT_alarm**

Used by: SNVT_alarm

```plaintext
TYPE ST_LON_SNVT_alarm :
    STRUCT
        arrLocation : ARRAY[0..5] OF BYTE;
        uiObject_Id : UINT;
        eAlarm_type : E_LON_alarm_type_t;
        ePriority_level : E_LON_priority_level_t;
        uiIndex_To_SNVT : WORD;
        arrValue : ARRAY[0..3] OF BYTE;
        uiYear : UINT;
        uiMonth : UINT;
        uiDay : UINT;
        uiHour : UINT;
        uiMinute : UINT;
        uiSecond : UINT;
        uiMilliseconds : UINT;
        arrAlarm_limit : ARRAY[0..3] OF BYTE;
    END_STRUCT
END_TYPE
```

- **arrLocation**: Location (array of 6 bytes). Location code for the node
- **uiObject_Id**: Object ID (object index). ID of object within node
- **eAlarm_type**: Alarm type (alarm type names) (see E_LON_alarm_type_t [503]).
- **ePriority_level**: Priority level (priority level names) (see E_LON_priority_level_t [533]).
- **uiIndex_To_SNVT**: Index of NV (index of NV causing alarm).
**arrValue**: Value (array of 4 bytes). The type of this field is dependent on the NV causing the alarm condition.

**uiYear**: Year (years). Zero (0) means year not specified.

**uiMonth**: Month (months). Zero (0) means month not specified.

**uiDay**: Day (days). Zero (0) means day not specified.

**uiHour**: Hour (hours). This field uses a 24-hour value.

**uiMinute**: Minute (minutes).

**uiSecond**: Second (seconds).

**uiMilliseconds**: Millisecond (milliseconds).

**arrAlarm_limit**: Alarm limit (array of 4 bytes). The type of this field is dependent on the NV causing the alarm condition.

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

#### 4.2.2.14 ST_LON_SNVT_alarm_2

**Used by**: SNVT_alarm_2

```plaintext
TYPE ST_LON_SNVT_alarm_2 :
  STRUCT
    eAlarm_type : E_LON_alarm_type_t;
    ePriority_level : E_LON_priority_level_t;
    udiAlarm_time : UDINT;
    iMilliseconds : INT;
    bySequence_number : BYTE;
    sDescription : STRING(22);
  END_STRUCT
END_TYPE
```

**eAlarm_type**: Alarm type (alarm type names). Alarm condition reported by this update (see E_LON_alarm_type_t [503]).

**ePriority_level**: Priority level (priority level names). Priority level of the alarm reported by this update (see E_LON_priority_level_t [533]).

**udiAlarm_time**: Alarm time (seconds). Alarm time in seconds since 2000-01-01T00:00:00Z (the 0 hour of 1 January 2000, Coordinated Universal Time)

**iMilliseconds**: Milliseconds (milliseconds). Alarm time in milliseconds since the second specified by the alarm_time field

**bySequence_number**: Sequence number (count). Sequence number for this update. Incremented by one for each update from an alarm source. Wraps to zero after reaching 255. An alarm receiver can use the sequence number to detect missed alarm messages.

**sDescription**: Description (array of 22 characters). Alarm description with NUL terminator. The terminator is not required if the description requires 22 characters. May include a reference to a language string, delimited by a 0x80 value.

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.2.2.15  ST_LON_SNVT_chlr_status

Used by: SNVT_chlr_status

```
TYPE ST_LON_SNVT_chlr_status :
  STRUCT
    eChlr_run_mode : E_LON_chiller_t;
    echlr_op_mode  : E_LON_hvac_t;
    stChlr_state   : ST_LON_chlr_state;
  END_STRUCT
  END_TYPE
```

- **eChlr_run_mode**: Chiller run mode (chiller run mode names) (see `E_LON_chiller_t`).
- **echlr_op_mode**: Chiller operating mode (HVAC mode names) (see `E_LON_hvac_t`).
- **stChlr_state**: Chiller state flags (alarm, enabled, local, limited, chiller water flow, condenser water flow) (see `ST_LON_chlr_state`).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.16  ST_LON_SNVT_clothes_w_a

Used by: SNVT_clothes_w_a

```
TYPE ST_LON_SNVT_clothes_w_a :
  STRUCT
    bAlarm_reset          : BOOL;
    bWar_water_supply     : BOOL;
    bWar_drain_slow       : BOOL;
    bWar_door_open        : BOOL;
    bWar_load_unbalanced  : BOOL;
    bWar_filter_cleaning  : BOOL;
    bWar_hoses_reversed   : BOOL;
    bWar_voltage_low      : BOOL;
    bWar_power_failure    : BOOL;
    bWar_drain_open       : BOOL;
    bWar_execute_fail     : BOOL;
    bWar_door_locked      : BOOL;
    bWar_service          : BOOL;
    bRsrvd5               : BOOL;
    bRsrvd6               : BOOL;
    bRsrvd7               : BOOL;
    bErr_motor_stall      : BOOL;
    bErr_water_temp       : BOOL;
    bErr_pressure         : BOOL;
    bErr_overflow         : BOOL;
    bErr_water_heat       : BOOL;
    bErr_water_leak       : BOOL;
    bErr_motor_speed      : BOOL;
    bErr_wash_thermistor  : BOOL;
    bErr_dry_thermistor   : BOOL;
    bErr_dry_overheat     : BOOL;
    bErr_dry_heating      : BOOL;
    bErr_dry_fan          : BOOL;
    bErr_rsrvd4           : BOOL;
    bErr_rsrvd5           : BOOL;
    bErr_rsrvd6           : BOOL;
    bErr_rsrvd7           : BOOL;
    byErr_rsrvd0_7        : BYTE;
    byManuf_code          : BYTE;
  END_STRUCT
  END_TYPE
```

- **bAlarm_reset**:
- **bWar_water_supply**:
- **bWar_drain_slow**: 

---

**BECKHOFF**

Programming
bWar_door_open:
bWar_load_unbalanced:
bWar_filter_cleaning:
bWar_hoses_reversed:
bWar_voltage_low:
bWar_power_failure:
bWar_drain_open:
bWar_execute_fail:
bWar_door_locked:
bWar_service:
bRsrvd5:
bRsrvd6:
bRsrvd7:
bErr_motor_stall:
bErr_water_temp:
bErr_pressure:
bErr_overflow:
bErr_water_heat:
bErr_water_leak:
bErr_motor_speed:
bErr_wash_thermistor:
bErr_dry_thermistor:
bErr_dry_overheat:
bErr_dry_heating:
bErr_dry_fan:
bErr_rsrvd4:
bErr_rsrvd5:
bErr_rsrvd6:
bErr_rsrvd7:
byErr_rsrvd0_7:
byManuf_code: Min: 0 / Max: 255

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
### 4.2.2.17  ST_LON_SNVT_clothes_w_c

*Used by: SNVT_clothes_w_c*

Type `ST_LON_SNVT_clothes_w_c`:

```c
TYPE ST_LON_SNVT_clothes_w_c :
STRUCT
  eCycle             : E_LON_appl_cwc_t;
  eSubcycle          : E_LON_appl_cws_t;
  byRervd            : BYTE;
  station            : ST_LON_action;
  stFunction         : ST_LON_function;
  uiTime_remaining   : UINT;
END_STRUCT
END_TYPE
```

- **eCycle**: (see `E_LON_appl_cwc_t`).
- **eSubcycle**: (see `E_LON_appl_cws_t`).
- **byRervd**: Reserve.
- **stAction**: (see `ST_LON_action`).
- **stFunction**: (see `ST_LON_function`).
- **uiTime_remaining**: Min: 0 / Max: 65535.

#### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.2.2.18  ST_LON_SNVT_clothes_w_m

*Used by: SNVT_clothes_w_m*

Type `ST_LON_SNVT_clothes_w_m`:

```c
TYPE ST_LON_SNVT_clothes_w_m :
STRUCT
  bDoor_ajar   : BOOL;
  bDrain_on   : BOOL;
  byReserved  : BYTE;
END_STRUCT
END_TYPE
```

- **bDoor_ajar**: Door/Lid Ajar. The door/lid of the washer is not fully closed.
- **bDrain_on**: Drain On. The drain is on.
- **byReserved**: Reserve.

#### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.2.2.19  ST_LON_SNVT_clothes_w_s

*Used by: SNVT_clothes_w_s*

Type `ST_LON_SNVT_clothes_w_s`:

```c
TYPE ST_LON_SNVT_clothes_w_s :
STRUCT
  eCycle : E_LON_appl_cwc_t;
  eSubcycle : E_LON_appl_cws_t;
  stWasher_command_data : ST_LON_SNVT_clothes_w_c;
  uiTime_remaining : UINT;
  stAlarm : ST_LON_alarm;
END_STRUCT
END_TYPE
```

T1000  Version: 1.5  595
eCycle: (see E_LON_appl_cwc_t [505]).

eSubcycle: (see E_LON_appl_cws_t [506]).

stWasher_command_data: (see ST_LON_SNVT_clothes_w_c [595]).

uiTime_remaining: Min: 0 / Max: 65535

stAlarm: (see ST_LON_alarm [561]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.20  ST_LON_SNVT_color

Used by: SNVT_color

TYPE ST_LON_SNVT_color :

  STRUCT
    rL_star : REAL;
    rA_star : REAL;
    rB_star : REAL;
  END_STRUCT
END_TYPE

rL_star: Min: 0 / Max: 100.0 / L*
rA_star: Min: -200.0 / Max: 200.0 / a*
rB_star: Min: -200.0 / Max: 200.0 / b*

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.21  ST_LON_SNVT_color_2

Used by: SNVT_color_2

TYPE ST_LON_SNVT_color_2 :

  STRUCT
    eEncoding : E_LON_color_encoding_t;
    stColor_value : ST_LON_color_value;
  END_STRUCT
END_TYPE

eEncoding: Color encoding. Color encoding specified by the color_value union; additional encodings may be added (see E_LON_color_encoding_t [509]).

stColor_value: Color value. Color value encoded as specified by the encoding field (see ST_LON_color_value [564]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.22  ST_LON_SNVT_ctrl_req

Used by: SNVT_ctrl_req
Programming

TYPE ST_LON_SNVT_ctrl_req :
   STRUCT
      uiReceiver_id : UINT;
      uiSender_id : UINT;
      bySender_prio : BYTE;
   END_STRUCT
END_TYPE

uiReceiver_id: Min: 1 / Max: 65535 / Invalid / 0 / Receiver ID (ID number).
uiSender_id: Min: 1 / Max: 65535 / Invalid / 65535 / Sender ID (ID number).
bySender_prio: Min: 0 / Max: 200 / Sender priority (priority value).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.23 ST_LON_SNVT_ctrl_resp

Used by: SNVT_ctrl_resp

TYPE ST_LON_SNVT_ctrl_resp :
   STRUCT
      eStatus : E_LON_control_resp_t;
      stSender : ST_LON_sender;
      uiController_id : UINT;
   END_STRUCT
END_TYPE

eStatus: Control response type (control response type names) (see E_LON_control_resp_t [510]).
stSender: Sender ID (see ST_LON_sender [565]).
uiController_id: Min: 1 / Max: 65535 / Invalid: 65535 / Controller ID (ID number).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.24 ST_LON_SNVT_currency

Used by: SNVT_currency

TYPE ST_LON_SNVT_currency :
   STRUCT
      eCurrency : E_LON_currency_t;
      siPower_of_10 : SINT;
      diValue : DINT;
   END_STRUCT
END_TYPE

eCurrency: Currency (currency names) (see E_LON_currency_t [510]).
diValue: Min: -2147483648 / Max: 2147483647 / Value (currency value). Credit is positive, debit is negative.

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.2.2.25  ST_LON_SNVT_date_event

Used by: SNVT_date_event

TYPE ST_LON_SNVT_date_event :
  STRUCT
    iDays_to_active   : INT;
    iDays_to_inactive : INT;
    sName            : STRING(22);
  END_STRUCT
END_TYPE

iDays_to_active: Min: -32768 / Max: 32767 / Invalid: 32767 / Days to active (days). Number of days until this schedule will be active. Positive if a schedule is inactive; zero or negative if a schedule is active.

iDays_to_inactive: Min: -32768 / Max: 32767 / Invalid: -32768 / Days to inactive (days). Number of days until this schedule will be inactive. Positive if a schedule is active; zero or negative if a schedule is inactive.

sName: 22 characters / Schedule name (array of 22 characters). Nul-terminated schedule name. The nul terminator is not required if the name is 22 characters.

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.26  ST_LON_SNVT_dev_fault

Used by: SNVT_dev_fault

TYPE ST_LON_SNVT_dev_fault :
  STRUCT
    eDevice_select  : E_LON_device_select_t;
    stDev_type      : ST_LON_Dev_type1;
  END_STRUCT
END_TYPE

eDevice_select: Device selection (device selection names) (see E_LON_device_select_t [516]). Determines the interpretation of the network-variable content.

stDev_type: Union of device fault structures for various devices (see ST_LON_Dev_type1 [565]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.27  ST_LON_SNVT_dev_maint

Used by: SNVT_dev_maint

TYPE ST_LON_SNVT_dev_maint :
  STRUCT
    eDevice_select : E_LON_device_select_t;
    stDev_type     : ST_LON_Dev_type2;
  END_STRUCT
END_TYPE

eDevice_select: Device selection (device selection names) (see E_LON_device_select_t [516]). Determines the interpretation of the network-variable content.

stDev_type: Union of device maintenance state structures for various devices (see ST_LON_Dev_type2 [568]).
### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

#### 4.2.2.28 ST_LON_SNVT_dev_status

**Used by:** SNVT_dev_status

**TYPE ST_LON_SNVT_dev_status :**

<table>
<thead>
<tr>
<th>STRUCT</th>
</tr>
</thead>
<tbody>
<tr>
<td>eDevice_select : E_LON_device_select_t;</td>
</tr>
<tr>
<td>stDev_type : ST_LON_Dev_type3;</td>
</tr>
</tbody>
</table>

**eDevice_select:** Device selection (device selection names) (see E_LON_device_select_t [516]). Determines the interpretation of the network-variable content.

**stDev_type:** Union of device status for various devices (see ST_LON_Dev_type3 [570]).

#### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

#### 4.2.2.29 ST_LON_SNVT_earth_pos

**Used by:** SNVT_earth_pos

**TYPE ST_LON_SNVT_earth_pos :**

<table>
<thead>
<tr>
<th>STRUCT</th>
</tr>
</thead>
<tbody>
<tr>
<td>bLatitude_direction : BOOL;</td>
</tr>
<tr>
<td>bLongitude_direction : BOOL;</td>
</tr>
<tr>
<td>byLatitude_deg : BYTE;</td>
</tr>
<tr>
<td>rLatitude_min : REAL;</td>
</tr>
<tr>
<td>bylongitude_deg : BYTE;</td>
</tr>
<tr>
<td>rLongitude_min : REAL;</td>
</tr>
</tbody>
</table>

**bLatitude_direction:** FALSE = South latitude, TRUE = North latitude.

**bLongitude_direction:** FALSE = East longitude, TRUE = West longitude.

**byLatitude_deg:** Min: 0 / Max: 90 / Invalid: 255 / Latitude degrees (degrees).

**rLatitude_min:** Min: 0.0 / Max: 59.999 / Invalid: 65.535/ Latitude minutes (minutes).

**bylongitude_deg:** Min: 0.0 / Max: 180 / Invalid: 255 / Longitude degrees (degrees).

**rLongitude_min:** Min: 0.0 / Max: 59.999 / Invalid: 65.535/ Longitude minutes (minutes).

**rHeight_above_sea:** Min: -3.40E+38 / Max: 3.40E+38 / Height above sea level (meters).

#### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

#### 4.2.2.30 ST_LON_SNVT_elapsed_tm

**Used by:** SNVT_elapsed_tm
TYPE ST_LON_SNVT_elapsed_tm :
  STRUCT
    uiDay : UINT;
    uiHour : UINT;
    uiMinute : UINT;
    uiSecond : UINT;
    uiMillisecond : UINT;
  END_STRUCT
END_TYPE

uiDay: Min: 0 / Max: 65535 / Days (days). The value 65535 represents NULL or unknown elapsed time.

uiHour: Min: 0 / Max: 23 / Hours (hours). This field uses a 24-hour value.

uiMinute: Min: 0 / Max: 59 / Minutes (minutes).

uiSecond: Min: 0 / Max: 59 / Seconds (seconds).

uiMillisecond: Min: 0 / Max: 999 / Milliseconds (milliseconds).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.31 ST_LON_SNVT_ent_status

Used by: SNVT_ent_status

TYPE ST_LON_SNVT_ent_status :
  STRUCT
    bUnlocked : BOOL;
    bLocked : BOOL;
    bSecurity_locked : BOOL;
    bClosed : BOOL;
    bOpen : BOOL;
    bIn_alarm : BOOL;
    bIn_error_cond : BOOL;
    bOpen_pre_alarm : BOOL;
    bOpen_alarm : BOOL;
    bService_alarm : BOOL;
    bTamper : BOOL;
    bEntry_req : BOOL;
    bExit_req : BOOL;
    bKey_req : BOOL;
    bSafety_ext_req : BOOL;
    bEmergency_req : BOOL;
    bUnable_lock : BOOL;
    bUnable_unlock : BOOL;
    bStuck : BOOL;
    bForced_open : BOOL;
    bForced_close : BOOL;
    bOpening : BOOL;
    bClosing : BOOL;
    bMoving : BOOL;
    bStopped : BOOL;
    bSafety_alarm : BOOL;
    bUnknown_state : BOOL;
eCmd_fb : E_LON_ent_opmode_cmd_t;
  END_STRUCT
END_TYPE

bUnlocked: Unlocked device (boolean). Device is in unlocked position.

bLocked: Locked device (boolean). Device is in locked position.

bSecurity_locked: Security locked (boolean). Device is in a security-driven locked position.

bClosed: Closed device (boolean). Device is in a closed position.

bOpen: Open device (boolean). Device is in an open position.

bIn_alarm: In alarm state (boolean). The device is in the alarm state.
bIn_error_cond: In error condition (boolean). Device has an error condition.

bOpen_pre_alarm: Open device, pre-alarm (boolean). Device is open, and in warning state.

bOpen_alarm: Open device, alarm state (boolean). Device is open, and in not-closed alarm state.

bService_alarm: Service alarm (boolean). Device needs service.

bTamper: Tamper mode (boolean). Device has detected tamper.

bEntry_req: Entry request pending (boolean). Device has a pending entry request.

bExit_req: Exit request pending (boolean). Device has a pending exit request.

bKey_req: Key request pending (boolean). Device has a pending key request.

bSafety_ext_req: Safety-exit request pending (boolean). Device has a pending safety-exit request.

bEmergency_req: Emergency-exit request pending (boolean). Device has a pending emergency-exit request.

bUnable_lock: Unable to lock (boolean). Device is unable to close and/or lock.

bUnable_unlock: Unable to unlock (boolean). Device is unable to open and/or unlock.

bStuck: Device is stuck (boolean). Device is unable to move.

bForced_open: Forced-open Device (boolean). Device is/was forced to go to an open position.

bForced_close: Forced-closed Device (boolean). Device is/was forced to go to a closed position.

bOpening: Device is opening (boolean). Device is currently opening from a closed position.

bClosing: Device is closing (boolean). Device is currently closing from an open position.

bMoving: Device is in motion (boolean). Device is currently changing position.

bStopped: Device Stopped (boolean). The device is stopped and can be moved manually.

bSafety_alarm: Safety-alarm (boolean). Device is in a safety-alarm state.

bUnknown_state: Unknown state (boolean). The state of the device is currently unknown.

eCmd_fb: Command feedback (entry command names) (see E_LON_ent_opmode_cmd_t [p 518]). Feedback of requested-operation-mode of device.

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.32 ST_LON_SNVT_environment

Used by: SNVT_environment

```plaintext
TYPE ST_LON_SNVT_environment :
    STRUCT
        uiLampCurrent : UINT;
        uiLampVoltage : UINT;
        uiSupplyVoltage : UINT;
        uiSupplyCurrent : UINT;
        rBallastTemp : REAL;
        rPower : REAL;
        rPowerFactor : REAL;
        udiRunHours : UDINT;
        lrEnergy : LREAL;
    END_STRUCT
END_TYPE
```
uiLampCurrent: Min: 0 / Max: 65534 / Invalid: 65535 / Lamp current (milliAmperes). This is the current the lamp consumes.

uiLampVoltage: Min: 0 / Max: 65535 / Lamp Voltage (Volts). This is the lamp voltage.

uiSupplyVoltage: Min: 0 / Max: 65535 / Supply Voltage (Volts). This is the luminaire supply voltage.

uiSupplyCurrent: Min: 0 / Max: 65534 / Invalid: 65535 / Supply Current (milliAmperes). This is the luminaire supply current.

rBallastTemp: Min: -273.17 / Max: 327.67 / Ballast temperature (degrees Celsius). This is the temperature at the ballast.

rPower: Min: 0 / Max: 6553.5 / Power (Watts). The value shows the at this moment consumed power of the ballast and the luminaire.

rPowerFactor: Min: -1 / Max: 1 / Power factor. This is the luminaire power-factor.

udiRunHours: Min: 0 / Max: 4294967294 / Run Hours (hours). This are the run hours since the last maintenance.

lrEnergy: Min: -214748364.8 / Max: 214748364.7 / Energy (kiloWatt-hours). This is the energy the luminair has consumed since the last maintenance.

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.33  ST_LON_SNVT_ex_control

Used by: SNVT_ex_control

TYPE ST_LON_SNVT_ex_control :

  STRUCT
    eControl_status : E_LON_ex_control_t;
    stControl_device_addr : ST_LON_Control_device_addr;
  END_STRUCT
END_TYPE

    eControl_status: Control type (control type names) (see E_LON_ex_control_t [520]).
    stControl_device_addr: Control device address (LonWorks subnet-node address) (see ST_LON_Control_device_addr [572]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.34  ST_LON_SNVT_file_pos

Used by: SNVT_file_pos

TYPE ST_LON_SNVT_file_pos :

  STRUCT
    diRw_ptr : DINT;
    uiRw_length : UINT;
  END_STRUCT
END_TYPE

diRw_ptr: Min: 0 / Max: 2147483647 / Read/Write pointer (file byte address).

uiRw_length: Min: 0 / Max: 65535 / Read/Write length (number of bytes).
### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

#### 4.2.2.35 ST_LON_SNVT_file_req

*Used by: SNVT_file_req*

**TYPE ST_LON_SNVT_file_req :**

**STRUCT**

- eRequest : E_LON_file_request_t;
- uiIndex : UINT;
- uiReceive_timeout : UINT;
- stDest_address : ST_LON_dest_address;
- byAuth_on : BYTE;
- byPrio_on : BYTE;

**END_STRUCT**

**END_TYPE**

**eRequest:** Request (file request names) (see [E_LON_file_request_t](#) [520]).

**uiIndex:** Min: 0 / Max: 65535 / Index (file index).

**uiReceive_timeout:** Min: 0 / Max: 65535 / Receive timeout (milliseconds).

**stDest_address:** Destination address (LonWorks address) (see [ST_LON_dest_address](#) [573]).

**byAuth_on:** Min: 0 / Max: 1 / Authentication on (boolean). This field specifies whether the message requires authentication.

**byPrio_on:** Min: 0 / Max: 1 / Priority on (boolean). This field specifies whether the message is to be sent with priority.

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

#### 4.2.2.36 ST_LON_SNVT_file_status

*Used by: SNVT_file_status*

**TYPE ST_LON_SNVT_file_status :**

**STRUCT**

- eStatus : E_LON_file_status_t;
- uiNumber_of_files : UINT;
- uiSelected_file : UINT;

**END_STRUCT**

**END_TYPE**

**eStatus:** Status (file status names) (see [E_LON_file_status_t](#) [521]).

**uiNumber_of_files:** Min: 0 / Max: 65535 / Number of files (count).

**uiSelected_file:** Min: 0 / Max: 65535 / Selected file (file index).

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
### 4.2.2.37 ST_LON_SNVT_geo_loc

Used by: SNVT_geo_loc

```plaintext
TYPE ST_LON_SNVT_geo_loc :
  STRUCT
    lrLongitude : LREAL;
    lrLatitude  : LREAL;
    rElevation  : REAL;
    sName      : STRING(19);
  END_STRUCT
END_TYPE
```

**lrLongitude**: Min: -180.0 / Max: 180.0 / Longitude. Longitude is given as an angular measurement ranging from 0° at the prime meridian to +180° eastward and -180° westward.

**lrLatitude**: Min: -90.0 / Max: 90.0 / Latitude. Latitude is given as an angular measurement ranging from 0° at the equator to +90° northward and -90° southward.

**rElevation**: Min: -3.40E+51 / Max: 3.40E+51 / Elevation (meters).

**sName**:

#### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.2.2.38 ST_LON_SNVT_hvac_overid

Used by: SNVT_hvac_overid

```plaintext
TYPE ST_LON_SNVT_hvac_overid :
  STRUCT
    eState   : E_LON_hvac_overid_t;
    rPercent : REAL;
    uiFlow   : UINT;
  END_STRUCT
END_TYPE
```

**eState**: HVAC override state (override state names) (see E_LON_hvac_overid_t [\[ 525\]]).

**rPercent**: Min: -163.84 / Max: 163.835 / Percent (% of full scale). Position or flow override value.

**uiFlow**: Min: 0 / Max: 65535 / Flow (liters/second).

#### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.2.2.39 ST_LON_SNVT_hvac_satsts

Used by: SNVT_hvac_satsts

```plaintext
TYPE ST_LON_SNVT_hvac_satsts :
  STRUCT
    bPri_heat         : BOOL;
    bSec_heat         : BOOL;
    bPri_cool         : BOOL;
    bSec_cool         : BOOL;
    bPri_duct_starved : BOOL;
    bSec_duct_starved : BOOL;
    bReserved1        : BOOL;
    bReserved2        : BOOL;
    byReserved1       : BYTE;
  END_STRUCT
END_TYPE
```

#### Requirements
bPri_heat: Primary heating saturation status (boolean). A value of 0 indicates primary heating is not saturated. A value of 1 indicates primary heating is saturated.

bSec_heat: Secondary heating saturation status (boolean). A value of 0 indicates secondary heating is not saturated. A value of 1 indicates secondary heating is saturated.

bPri_cool: Primary cooling saturation status (boolean). A value of 0 indicates primary cooling is not saturated. A value of 1 indicates primary cooling is saturated.

bSec_cool: Secondary cooling saturation status (boolean). A value of 0 indicates secondary cooling is not saturated. A value of 1 indicates secondary cooling is saturated.

bPri_duct_starved: Primary duct saturation status (boolean). A value of 0 indicates primary duct is not saturated (starved). A value of 1 indicates primary duct is saturated (starved).

bSec_duct_starved: Secondary duct saturation status (boolean). A value of 0 indicates secondary duct is not saturated (starved). A value of 1 indicates secondary duct is saturated (starved).

bReserved1:

bReserved2:

byReserved1: Min: 0 / Max: 15

byManufacturer_defined: Min: 0 / Max: 15 / Manufacturer defined (boolean). Four manufacturer-defined bits — please see product documentation for proper interpretation of these bits

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.40 ST_LON_SNVT_hvac_status

Used by: SNVT_hvac_status

TYPE ST_LON_SNVT_hvac_status :
    STRUCT
        eMode            : E_LON_hvac_t;
        rHeat_output_primary   : REAL;
        rHeat_output_secondary : REAL;
        rCool_output        : REAL;
        rEcon_output       : REAL;
        rFan_output        : REAL;
        byIn_alarm         : BYTE;
    END_STRUCT
    END_TYPE

eMode: HVAC status mode (HVAC mode names) (see E_LON_hvac_t [527]).

rHeat_output_primary: Min: -163.8400 / Max: 163.8350 / Primary heat output (% of full scale).


rCool_output: Min: -163.8400 / Max: 163.8350 / Cooling output (% of full scale).

rEcon_output: Min: -163.8400 / Max: 163.8350 / Economizer output (% of full scale).

rFan_output: Min: -163.8400 / Max: 163.8350 / Fan output (% of full scale).

byIn_alarm: Min: 0 / Max: 255 / in_alarm
Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.2.2.41 ST_LON_SNVT_lamp_status

**Used by:** SNVT_lamp_status

```plaintext
TYPE ST_LON_SNVT_lamp_status :
    STRUCT
        stTime_actual : TIMESTRUCT;
        stAlarm_actual : ST_LON_Alarm_actual;
        stTime_previous : TIMESTRUCT;
        stAlarm_previous : ST_LON_alarm_previous;
    END_STRUCT
    END_TYPE

stTime_actual: Actual alarm message. This is the time stamp for the actual alarm message (see TIMESTRUCT).

stAlarm_actual: (see `ST_LON_Alarm_actual`) [575].

stTime_previous: Time Stamp Old. This is the time stamp for an old alarm message (see TIMESTRUCT).

stAlarm_previous: (see `ST_LON_alarm_previous`) [577].

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.2.2.42 ST_LON_SNVT_log_fx_request

**Used by:** SNVT_log_fx_request

```plaintext
TYPE ST_LON_SNVT_log_fx_request :
    STRUCT
        uiRequested_log : UINT;
        udiRecord_count : UDINT;
        stStart_time    : TIMESTRUCT;
        stEnd_time      : TIMESTRUCT;
    END_STRUCT
    END_TYPE
```

uiRequested_log: Min: 1 / Max: 65535 / Requested log number. The log number of the data log to be transferred. Logs are numbered from 1 to number_of_logs.

udiRecord_count: Min: 0 / Max: 4294967295 / Record count. The maximum number of log records to be transferred.

stStart_time: Start time. (seconds) (see TIMESTRUCT). Timestamp of first record to be transferred. If no records exist with this timestamp, the first record with a timestamp after this timestamp is the starting record.

stEnd_time: End time. (seconds) (see TIMESTRUCT). Timestamp of last record to be transferred. If no records exist with this timestamp, the last record with a timestamp before this timestamp is the ending record.

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.2.2.43 ST_LON_SNVT_log_fx_status

Used by: SNVT_log_fx_status

```plaintext
TYPE ST_LON_SNVT_log_fx_status : 
  STRUCT
    byRequestor_subnet : BYTE;
    byRequestor_node : BYTE;
    uiLog_number : UINT;
    rComplete : REAL;
  END_STRUCT
END_TYPE
```

- **byRequestor_subnet**: Min: 1 / Max: 255 / Requestor subnet ID. Subnet ID of the device that requested the current log file transfer. Invalid if a file transfer is not active.

- **byRequestor_node**: Min: 1 / Max: 255 / Requestor node ID. Node ID of the device that requested the current log file transfer. Invalid if a file transfer is not active.

- **uiLog_number**: Min: 1 / Max: 65535 / Data log number. Log number for the log file currently being transferred via FTP. Invalid if none.

- **rComplete**: Min: 0 / Max: 100 / Data log file transfer percent complete. Percent of the current data log file transfer that has been completed. Invalid if none.

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.44 ST_LON_SNVT_log_status

Used by: SNVT_log_status

```plaintext
TYPE ST_LON_SNVT_log_status : 
  STRUCT
    eStatus : E_LON_log_status_t;
    uiLog_number : UINT;
    rLevel : REAL;
    diRecord_count : DINT;
    diByte_count : DINT;
    diTotal_record_count : DINT;
    diRecords_since_notification : DINT;
    stCurrent_notify_time : TIMESTRUCT;
    stPrevious_notify_time : TIMESTRUCT;
  END_STRUCT
END_TYPE
```

- **eStatus**: Log state. State of the selected data log (see E_LON_log_status_t [529]).

- **uiLog_number**: Min: 1 / Max: 65535 / Selected log number. The log number of the reported data log. Logs are numbered from 1 to number_of_logs.

- **rLevel**: Min: 0 / Max: 100 / Log level (Percent). The percent of maximum records in the selected data log.

- **diRecord_count**: Min: -2147483648 / Max: 2147483647 / Record count. Number of records in the selected data log. A record is a logged value and any associated data such as a timestamp.

- **diByte_count**: Min: -2147483648 / Max: 2147483647 / Byte count. Number of bytes in the selected data log.

- **diTotal_record_count**: Min: -2147483648 / Max: 2147483647 / Total record count. Total records collected in the selected data log since the data log was created. Wraps to 0 on overflow.

- **diRecords_since_notification**: Min: -2147483648 / Max: 2147483647 / Records since notification. The number of log records collected since the last notification.

- **stCurrent_notify_time**: Current notify time (see TIMESTRUCT). Timestamp of the most recently collected data point.
**stPrevious_notify_time:** Previous notify time (seconds) (see TIMESTRUCT). Timestamp of the most recently collected data point in the previous update to the log status.

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

#### 4.2.2.45  **ST_LON_SNVT_muldiv**

Used by: SNVT_muldiv

```
TYPE ST_LON_SNVT_muldiv :
   STRUCT
      uiMultiplier : UINT;
      uiDivisor    : UINT;
   END_STRUCT
END_TYPE
```

- **uiMultiplier:** Min: 0 / Max: 65535 / Multiplier (16-bit unsigned value).
- **uiDivisor:** Min: 0 / Max: 65535 / Divisor (16-bit unsigned value).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

#### 4.2.2.46  **ST_LON_SNVT_nv_type**

Used by: SNVT_nv_type

```
TYPE ST_LON_SNVT_nv_type :
   STRUCT
      arrType_program_ID : ARRAY[0..7] OF BYTE;
      byType_scope       : BYTE;
      uiType_index       : UINT;
      eType_category     : E_LON_nv_type_category_t;
      byType_length      : BYTE;
      iScaling_factor_a  : INT;
      iScaling_factor_b  : INT;
      iScaling_factor_c  : INT;
   END_STRUCT
END_TYPE
```

- **arrType_program_ID:** Min: 0 / Max: 255 / Type program ID. Program ID template of the resource file containing the network variable type definition.
- **byType_scope:** Min: 0 / Max: 6 / Type scope (file scope). Scope of the resource file containing the network variable type definition.
- **uiType_index:** Min: 1 / Max: 65535 / Type index (type index). Index within the specified resource file of the network variable type definition.
- **eType_category:** Type category (type category names) (see E_LON_nv_type_category_t [530]). Type category of the network variable type.
- **byType_length:** Min: 1 / Max: 31 / Type length (bytes). Length of the network variable type.
- **iScaling_factor_a:** Min: -32768 / Max: 32767 / Scaling factor a. Scaling multiplier 'a' where ScaledValue = a*(10**b)*(RawValue+c)
- **iScaling_factor_b:** Min: -32768 / Max: 32767 / Scaling factor b. Exponent 'b' where ScaledValue = a*(10**b)*(RawValue+c)
iScaling_factor_c: Min: -32768 / Max: 32767 / Scaling Factor c. Offset 'c' where ScaledValue = a*(10**b)*(RawValue+c)

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.47 ST_LON_SNVT_obj_request

Used by: SNVT_obj_request

TYPE ST_LON_SNVT_obj_request :

  STRUCT
    uiObject_id     : UINT;
    eObject_request : E_LON_object_request_t;
  END_STRUCT

END_TYPE

uiObject_id: Min: 0 / Max: 65535 / Object ID (object index).

eObject_request: Object request (object request names) (see E_LON_object_request_t [531]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.48 ST_LON_SNVT_obj_status

Used by: SNVT_obj_status

TYPE ST_LON_SNVT_obj_status :

  STRUCT
    uiObject_id            : UINT;
    bInvalid_id            : BOOL;
    bInvalid_request       : BOOL;
    bDisabled              : BOOL;
    bOut_of_limits        : BOOL;
    bOpen_circuit          : BOOL;
    bOut_of_service        : BOOL;
    bMechanical_fault      : BOOL;
    bFeedback_failure      : BOOL;
    bOver_range            : BOOL;
    bUnder_range           : BOOL;
    bElectrical_fault      : BOOL;
    bUnable_to_measure     : BOOL;
    bComm_failure          : BOOL;
    bFail_self_test        : BOOL;
    bSelf_test_in_progress : BOOL;
    bLocked_out            : BOOL;
    bManual_control        : BOOL;
    bIn_alarm              : BOOL;
    bIn_override           : BOOL;
    bReport_mask           : BOOL;
    bProgramming_mode      : BOOL;
    bProgramming_fail      : BOOL;
    bAlarm_notify_disabled : BOOL;
    bReset_complete        : BOOL;
    byReserved2            : BYTE;
  END_STRUCT

END_TYPE

uiObject_id: Min: 0 / Max: 65535 / Object ID (object index).

bInvalid_id: Invalid-ID flag (boolean).

bInvalid_request: Invalid-request flag (boolean).
bDisabled: Disabled flag (boolean).
bOpen_circuit: Open-circuit flag (boolean).
bMechanical_fault: Mechanical-fault flag (boolean).
bFeedback_failure: Feedback-failure flag (boolean).
bOver_range: Over-range flag (boolean).
bUnder_range: Under-range flag (boolean).
bElectrical_fault: Electrical-fault flag (boolean).
bUnable_to_measure: Unable-to-measure flag (boolean).
bComm_failure: Communications-failure flag (boolean).
bFail_self_test: Failed-self-test flag (boolean).
bSelf_test_in_progress: Self-test-in-progress flag (boolean).
bLocked_out: Locked-out flag (boolean).
bln_alarm: Input-alarm flag (boolean).
bln_override: Input-override flag (boolean).
bProgramming_mode: Programming-mode flag (boolean).
bProgramming_fail: Programming-fail flag (boolean).
bAlarm_notify_disabled: Alarm-notify-disabled flag (boolean).
bReset_complete: Reset (boolean).
byReserved2: This field is reserved.

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.49 ST_LON_SNVT_pos_ctrl

Used by: SNVT_pos_ctrl

```plaintext
TYPE ST_LON_SNVT_pos_ctrl :
STRUCT
   uiReceiver_id   : UINT;
   uiController_id : UINT;
   byController_prio : BYTE;
   eFunction      : E_LON_cam_func_t;
   eAction        : E_LON_cam_act_t;
   stValue        : ST_LON_Value;
END_STRUCT
END_TYPE
```

uiReceiver_id: Min: 0 / Max: 65535 / Receiver ID (ID number).

uiController_id: Min: 0 / Max: 65535 / Controller ID (ID number).
byController_prio: Min: 0 / Max: 100 / Controller priority (priority value).

eFunction: Camera function (camera function names) (see E_LON_cam_func_t [508]).

eAction: Camera action (camera action names) (see E_LON_cam_act_t [508]).

stValue: Function value (see ST_LON_Value [579]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.2.2.50 ST_LON_SNVT_preset

Used by: SNVT_preset

```
TYPE ST_LON_SNVT_preset :
  STRUCT
    eLearn        : E_LON_learn_mode_t;
    uiSelector    : UINT;
    arrValue      : ARRAY[0..3] OF BYTE;
    uiDay         : UINT;
    uiHour        : UINT;
    uiMinute      : UINT;
    uiSecond      : UINT;
    uiMillisecond : UINT;
  END_STRUCT
END_TYPE
```

eLearn: Learn mode (learn mode names) (see E_LON_learn_mode_t [528]).

uiSelector: Min: 0 / Max: 65535 / Selector (16-bit unsigned value). The selector is used to choose which preset.

arrValue: Value (array of 4 bytes).

uiDay: Min: 0 / Max: 65535 / Days (days). The value 65535 represents NULL or unknown elapsed time.

uiHour: Min: 0 / Max: 23 / Hours (hours). This field uses a 24-hour value.

uiMinute: Min: 0 / Max: 59 / Minutes (minutes).

uiSecond: Min: 0 / Max: 59 / Seconds (seconds).

uiMillisecond: Min: 0 / Max: 999 / Milliseconds (milliseconds).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.2.2.51 ST_LON_SNVT_privacyzone

Used by: SNVT_privacyzone

```
TYPE ST_LON_SNVT_privacyzone :
  STRUCT
    eAction     : E_LON_privacyzone_t;
    byNumber    : BYTE;
    uiCamera_id : UINT;
  END_STRUCT
END_TYPE
```

eAction: Privacy zone action type (privacy zone action type names) (see E_LON_privacyzone_t [534]).

byNumber: Min: 0 / Max: 255 / Zone number (zone number).
uiCamera_id: Min: 0 / Max: 65535 / Camera ID (ID number).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.52 ST_LON_SNVT_ptz

Used by: SNVT_ptz

```plaintext
TYPE ST_LON_SNVT_ptz :
STRUCT
    ePan_dir : E_LON_pan_dir_t;
    rPan_speed : REAL;
    eTilt_dir : E_LON_tilt_dir_t;
    rTilt_speed : REAL;
    eZoom : E_LON_zoom_t;
    rZoom_speed : REAL;
END_STRUCT
END_TYPE
```

ePan_dir: Pan direction (pan direction names) (see E_LON_pan_dir_t [\ref E_LON_pan_dir_t]).
rPan_speed: Min: 0 / Max: 100 / Pan speed (% of full level).

eTilt_dir: Tilt direction (tilt direction names) (see E_LON_tilt_dir_t [\ref E_LON_tilt_dir_t]).
rTilt_speed: Min: 0 / Max: 100 / Tilt speed (% of full level).

eZoom: Zoom direction (zoom direction names) (see E_LON_zoom_t [\ref E_LON_zoom_t]).
rZoom_speed: Min: 0 / Max: 100 / Zoom speed (% of full level).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.53 ST_LON_SNVT_pump_sensor

Used by: SNVT_pump_sensor

```plaintext
TYPE ST_LON_SNVT_pump_sensor :
STRUCT
    rRotational_speed : REAL;
    rBody_temperature : REAL;
    rMotor_external_temperature : REAL;
    rMotor_internal_temperature : REAL;
    eMotor_overloaded : E_LON_boolean_t;
    eOil_level_low : E_LON_boolean_t;
    ePhase_imbalance_detected : E_LON_boolean_t;
    rCurrent_usage : REAL;
    rPower_usage : REAL;
    eTemperature_control : E_LON_unit_temp_t;
    eElectromagnetic_brake_active : E_LON_boolean_t;
    eFriction_brake_active : E_LON_boolean_t;
    eGas_brake_active : E_LON_boolean_t;
END_STRUCT
END_TYPE
```

rRotational_speed: Min: 0 / Max: 6553.5 / Rotational speed.
rBody_temperature: Min: -274 / Max: 6279.5 / Body temperature.
rMotor_external_temperature: Min: -274 / Max: 6279.5 / Motor external temp.
rMotor_internal_temperature: Min: -274 / Max: 6279.5 / Motor internal temp.
eMotor_overloaded: Motor overloaded (boolean) (see E_LON_boolean_t).[507]).
eOil_level_low: Oil level low (boolean) (see E_LON_boolean_t).[507]).
ePhase_imbalance_detected: Phase imbalance (boolean) (see E_LON_boolean_t).[507]).
rCurrent_usage: Min: -3276.8 / Max: 3276.7 / Current usage.
rPower_usage: Min: 0 / Max: 6553.5 / Power usage.
eTemperature_control: Pump body temp control status (temperature control status names) (see E_LON_unit_temp_t).[550]).
eElectromagnetic_brake_active: Electromagnetic brake active (boolean) (see E_LON_boolean_t).[507]).
eFriction_brake_active: Friction brake active (boolean) (see E_LON_boolean_t).[507]).
eGas_brake_active: Gas brake active (boolean) (see E_LON_boolean_t).[507]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.54 ST_LON_SNVT_pumpset_mn

Used by: SNVT_pumpset_mn

TYPE ST_LON_SNVT_pumpset_mn :

STRUCT
  eMain_pump : E_LON_motor_state_t;
  eBooster_pump : E_LON_motor_state_t;
  ePriority_level : E_LON_Priority_level_t;
  eProcess_ready : E_LON_boolean_t;
  eEmergency_stop_activated : E_LON_boolean_t;
  eMain_pump_drive_enabled : E_LON_boolean_t;
  eBooster_pump_drive_enabled : E_LON_boolean_t;
  eMaintenance_required : E_LON_boolean_t;
END_STRUCT
END_TYPE

eMain_pump: Main pump state (motor state names) (see E_LON_motor_state_t).[529]).
eBooster_pump: Booster pump state (motor state names) (see E_LON_motor_state_t).[529]).
ePriority_level: Priority level (priority level names) (see E_LON_Priority_level_t).[533]).
eProcess_ready: Process ready (boolean) (see E_LON_boolean_t).[507]).
eEmergency_stop_activated: Emergency stop (boolean) (see E_LON_boolean_t).[507]).
eMain_pump_drive_enabled: Main pump enabled (boolean) (see E_LON_boolean_t).[507]).
eBooster_pump_drive_enabled: Booster pump enabled (boolean) (see E_LON_boolean_t).[507]).
eMaintenance_required: Maintenance required (boolean) (see E_LON_boolean_t).[507]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.2.2.55 ST_LON_SNVT_pumpset_sn

Used by: SNVT_pumpset_sn

TYPE ST_LON_SNVT_pumpset_sn :
STRUCT
  uiTotal_dilution_flow : UINT;
rExhaust_temperature  : REAL;
rExhaust_pressure    : REAL;
rShaft_seal_purge_pressure : REAL;
rInlet_vacuum       : REAL;
rSupply_voltage     : REAL;
uiCoolant_flow      : UINT;
eCoolant_flow_low  : E_LON_boolean_t;
eDilution_active   : E_LON_boolean_t;
eBallast_dilution_active : E_LON_boolean_t;
eInlet_purge_dilution_active : E_LON_boolean_t;
eExhaust_dilution_active : E_LON_boolean_t;
eDilution_flow_Out_of_range : E_LON_boolean_t;
ePower_supply_On : E_LON_boolean_t;
END_STRUCT
END_TYPE

uiTotal_dilution_flow: Min: 0 / Max: 65535 / Dilution gas flow.

rExhaust_temperature: Min: 274 / Max: 6279.5 / Exhaust line external temperature.

rExhaust_pressure: Min: -3276.8 / Max: 3276.7 / Exhaust line pressure.

rShaft_seal_purge_pressure: Min: -3276.8 / Max: 3276.7 / Shaft seal purge pressure.

rInlet_vacuum: Min: -3.40E+38 / Max: 3.40E+38 / Process gas inlet pressure.

rSupply_voltage: Min: -3276.8 / Max: 3276.7 / Pumpset power supply voltage.

uiCoolant_flow: Min: 0 / Max: 65535 / Total coolant flow.

eCoolant_flow_low: Coolant flow too low (boolean) (see E_LON_boolean_t [507]).

eDilution_active: Coolant flow too low (boolean) (see E_LON_boolean_t [507]).

eBallast_dilution_active: Dilution gas being used as ballast (boolean) (see E_LON_boolean_t [507]).

eInlet_purge_dilution_active: Dilution gas being used to purge process gas (boolean) (see E_LON_boolean_t [507]).

eExhaust_dilution_active: Dilution gas being used to dilute exhaust (boolean) (see E_LON_boolean_t [507]).

eDilution_flow_Out_of_range: Dilution gas flow outside normal range (boolean) (see E_LON_boolean_t [507]).

ePower_supply_on: Dilution gas flow outside normal range (boolean) (see E_LON_boolean_t [507]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
byAudio_line: Min: 0 / Max: 7

bDuplex_full:

bDest_p2p:

byReserved:

eAudio_type: (see E_LON_rail_audio_type_t [536]).

stAddr_init: (see ST_LON_addr_init [580]).

stAddr_talk: (see ST_LON_addr_talk [581]).

stAddr_dest: (see ST_LON_addr_dest [580]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.57 ST_LON_SNVT_rac_req

Used by: SNVT_rac_req

TYPE ST_LON_SNVT_rac_req :
  STRUCT
    bDest_def   : BOOL;
    bDest_p2p   : BOOL;
    byReserved  : BYTE;
    eAudio_type : E_LON_rail_audio_type_t;
    stAddr_init : ST_LON_addr_init;
    stAddr_talk : ST_LON_addr_talk;
    stAddr_dest : ST_LON_addr_dest;
  END_STRUCT
  END_TYPE

bDest_def:

bDest_p2p:

byReserved:

eAudio_type: (see E_LON_rail_audio_type_t [536]).

stAddr_init: (see ST_LON_addr_init [580]).

stAddr_dest: (see ST_LON_addr_dest [580]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.58 ST_LON_SNVT_reg_val

Used by: SNVT_rac_val

TYPE ST_LON_SNVT_reg_val :
  STRUCT
    diRaw            : DINT;
  END_STRUCT
  END_TYPE
eUnit : E_LON_reg_val_unit_t;
byNr_decimals : BYTE;
END_STRUCT
END_TYPE

diRaw: Raw value.

eUnit: Unit code (defines unit of measure) (see E_LON_reg_val_unit_t [537]).

byNr_decimals: Number of decimals (digits to right of decimal point).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.59 ST_LON_SNVT_reg_val_ts

Used by: SNVT_rac_val_ts

TYPE ST_LON_SNVT_reg_val_ts :
STRUCT
  diRaw : DINT;
eUnit : E_LON_reg_val_unit_t;
byNr_decimals : BYTE;
byStatus : BYTE;
bReg_state : BOOL;
uiYear : UINT;
uiMonth : UINT;
uiDay : UINT;
uiHour : UINT;
uiMinute : UINT;
uiSecond : UINT;
END_STRUCT
END_TYPE

diRaw: Min: -2147483648 / Max: 2147483647 / Raw value.

eUnit: Unit code (unit names) (see E_LON_reg_val_unit_t [537]).

byNr_decimals: Min: 0 / Max: 7 / Number of decimals (digits to right of decimal point).

byStatus: Min: 0 / Max: 15 / Status (status or error during measuring period).

bReg_state: Activation state (activation state of register).

uiYear: Min: -1 / Max: 3000 / Year (years). Zero (0) means year not specified. Minus one (-1) represents NULL date.

uiMonth: Min: 0 / Max: 12 / Month (months). Zero (0) means month not specified.

uiDay: Min: 0 / Max: 31 / Day (days). Zero (0) means day not specified.

uiHour: Min: 0 / Max: 23 / Hour (hours). This field uses a 24-hour value.

uiMinute: Min: 0 / Max: 59 / Minute (minutes).

uiSecond: Min: 0 / Max: 59 / Second (seconds).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.60 ST_LON_SNVT_sblnd_state

Used by: SNVT_sblnd_state
TYPE ST_LON_SNVT_sblnd_state :
STRUCT
  stPos : ST_LON_SNVT_setting;
  eCmd_source : E_LON_sblnd_cmd_source_t;
  eError_code : E_LON_sblnd_error_t;
END_STRUCT
END_TYPE

stPos: (see ST_LON_SNVT_setting [618]).

eCmd_source: (see E_LON_sblnd_cmd_source_t [539]).

eError_code: (see E_LON_sblnd_error_t [540]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.61 ST_LON_SNVT_scene

Used by: SNVT_scene

TYPE ST_LON_SNVT_scene :
STRUCT
  eFunction : E_LON_Scene_t;
  byScene_number : BYTE;
END_STRUCT
END_TYPE

eFunction: Scene control function (scene control function names) (see E_LON_Scene_t [542]).

byScene_number: Min: 0 / Max: 255 / Scene number.

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.62 ST_LON_SNVT_scene_cfg

Used by: SNVT_scene_cfg

TYPE ST_LON_SNVT_scene_cfg :
STRUCT
  eFunction : E_LON_Scene_config_t;
  byScene_number : BYTE;
  rSetting : REAL;
  rRotation : REAL;
  rFade_time : REAL;
  rDelay_time : REAL;
  scene_priority : BYTE;
END_STRUCT
END_TYPE

eFunction: Scene configuration function (scene configuration function names) (see E_LON_Scene_config_t [541]).

byScene_number: Min: 0 / Max: 255 / Scene number.

rSetting: Min: 0 / Max: 100.0 Invalid: 255 / Scene setting level (% of full level).

rRotation: Min: -359.98 / Max: 360.00 / Scene rotation angle (degrees).

rFade_time: Min: 0 / Max: 6553.5 / Scene fade time (seconds).

rDelay_time: Min: 0 / Max: 6553.5 / Scene delay time (seconds).
scene_priority: Min: 0 / Max: 255 / scene_priority

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.2.2.63  ST_LON_SNVT_setting

Used by: SNVT_setting

```
TYPE ST_LON_SNVT_setting :
  STRUCT
    eFunction : E_LON_setting_t;
    rSetting  : REAL;
    rRotation : REAL;
  END_STRUCT
END_TYPE
```

- **eFunction**: Setting control function (setting control function names) (see E_LON_setting_t [p. 545]).
- **rSetting**: Min: 0 / Max: 100 / Scene setting level (% of full level).
- **rRotation**: Min: -359.98 / Max: 360.00 / Rotation angle (degrees).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.2.2.64  ST_LON_SNVT_str_int

Used by: SNVT_str_int

```
TYPE ST_LON_SNVT_str_int :
  STRUCT
    byChar_set  : BYTE;
    arrWide_char : ARRAY[0..14] OF UINT;
  END_STRUCT
END_TYPE
```

- **byChar_set**: Min: 0 / Max: 255 / Locale code (code value).
- **arrWide_char**: Min: 0 / Max: 65535 / Wide character string (array of 15 wide characters).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.2.2.65  ST_LON_SNVT_switch

Used by: SNVT_switch

```
TYPE ST_LON_SNVT_switch :
  STRUCT
    rValue  : REAL;
    siState : SINT;
  END_STRUCT
END_TYPE
```

- **rValue**: Min: 0 / Max: 100 / Value (% of full level).
- **siState**: Min: -1 / Max: 1 / State (state code). This field can either be -1 (NULL), 0 (OFF), or 1 (ON).
Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.2.2.66  ST_LON_SNVT_switch_2

**Used by:** SNVT_switch_2

```plaintext
type ST_LON_SNVT_switch_2 :
struct
  eState : E_LON_switch_state_t;
  stSetting : ST_LON_setting;
  byScene_number : BYTE;
end_STRUCT
end_TYPE
```

**eState:** Switch state (see E_LON_switch_state_t [546]). Switch state; maybe a state of the switch or other switch properties such as scene, occupancy state, and level multiplier.

**stSetting:** Switch setting. Sets or reports the level, change, or angle for a switch (see ST_LON_setting [583]).

**byScene_number:** Min: 1 / Max: 255 / Scene number. Scene number that is applied based on the function specified in the state field.

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.2.2.67  ST_LON_SNVT_temp_setpt

**Used by:** SNVT_temp_setpt

```plaintext
type ST_LON_SNVT_temp_setpt :
struct
  rOccupied_cool : REAL;
  rStandby_cool  : REAL;
  rUnoccupied_cool : REAL;
  rOccupied_heat : REAL;
  rStandby_heat  : REAL;
  rUnoccupied_heat : REAL;
end_STRUCT
end_TYPE
```

**rOccupied_cool:** Min: -273,17 / Max: 237,67 / Occupied cooling setpoint (degrees Celsius).

**rStandby_cool:** Min: -273,17 / Max: 237,67 / Standby cooling setpoint (degrees Celsius).

**rUnoccupied_cool:** Min: -273,17 / Max: 237,67 / Unoccupied cooling setpoint (degrees Celsius).

**rOccupied_heat:** Min: -273,17 / Max: 237,67 / Occupied heating setpoint (degrees Celsius).

**rStandby_heat:** Min: -273,17 / Max: 237,67 / Standby heating setpoint (degrees Celsius).

**rUnoccupied_heat:** Min: -273,17 / Max: 237,67 / Unoccupied heating setpoint (degrees Celsius).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.2.2.68 ST_LON_SNVT_time_zone

Used by: SNVT_time_zone

```c
TYPE ST_LON_SNVT_time_zone :
STRUCT
  diSecond_time_offset : DINT;
  eType_of_description : E_LON_calendar_type_t;
  byHour_of_start_DST : BYTE;
  byMinute_of_start_DST : BYTE;
  bySecond_of_start_DST : BYTE;
  stStart_DST : ST_LON_start_DST;
  byHour_of_end_DST : BYTE;
  byMinute_of_end_DST : BYTE;
  bySecond_of_end_DST : BYTE;
  stEnd_DST : ST_LON_end_DST;
END_STRUCT
END_TYPE
```

diSecond_time_offset: Min: -86400 / Max: 86400 / Offset from GMT (seconds). West direction is negative offset.

eType_of_description: Calendar type (calendar type names) (see E_LON_calendar_type_t [507]).

byHour_of_start_DST: Min: 0 / Max: 23 / DST start hour (hours).

byMinute_of_start_DST: Min: 0 / Max: 59 / DST start minute (minutes).

bySecond_of_start_DST: Min: 0 / Max: 59 / DST start second (seconds).

stStart_DST: DST start day (day descriptor). Daylight savings time start day (see ST_LON_start_DST [585]).

byHour_of_end_DST: Min: 0 / Max: 23 / DST end hour (hours).

byMinute_of_end_DST: Min: 0 / Max: 59 / DST end minute (minutes).

bySecond_of_end_DST: Min: 0 / Max: 59 / DST end second (seconds).

stEnd_DST: DST end day (day descriptor). Daylight savings time end day (see ST_LON_end_DST [584]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.69 ST_LON_SNVT_tod_event

Used by: SNVT_tod_event

```c
TYPE ST_LON_SNVT_tod_event :
STRUCT
  eCurrent_state : E_LON_occup_t;
  eNext_state : E_LON_occup_t;
  uiTime_to_next_state : UINT;
END_STRUCT
END_TYPE
```

eCurrent_state: Occupancy, current (occupancy code names) (see E_LON_occup_t [532]).

eNext_state: Occupancy, next (occupancy code names) (see E_LON_occup_t [532]).

uiTime_to_next_state: Min: 0 / Max: 65535 / Time to next state (minutes).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.2.2.70  ST_LON_SNVT_trans_table

Used by: SNVT_trans_table

Used by: SNVT_trans_table

```
TYPE ST_LON_SNVT_trans_table :
STRUCT
    arrPoint : ARRAY[0..6] OF REAL;
    byInterp_pts_0_to_1 : BYTE;
    byInterp_pts_1_to_2 : BYTE;
    byInterp_pts_2_to_3 : BYTE;
    byInterp_pts_3_to_4 : BYTE;
    byInterp_pts_4_to_5 : BYTE;
    byInterp_pts_5_to_6 : BYTE;
    byInterp_pts_6_to_0 : BYTE;
END_STRUCT
END_TYPE
```

**arrPoint:** Points (array of 7 points).

**byInterp_pts_0_to_1:** Min: 0 / Max: 1 / Interpolation for point 0 to point 1 (interpolation method code).

**byInterp_pts_1_to_2:** Min: 0 / Max: 1 / Interpolation for point 1 to point 2 (interpolation method code).

**byInterp_pts_2_to_3:** Min: 0 / Max: 1 / Interpolation for point 2 to point 3 (interpolation method code).

**byInterp_pts_3_to_4:** Min: 0 / Max: 1 / Interpolation for point 3 to point 4 (interpolation method code).

**byInterp_pts_4_to_5:** Min: 0 / Max: 1 / Interpolation for point 4 to point 5 (interpolation method code).

**byInterp_pts_5_to_6:** Min: 0 / Max: 1 / Interpolation for point 5 to point 6 (interpolation method code).

**byInterp_pts_6_to_0:** Min: 0 / Max: 1 / Interpolation for point 6 to point 0 (interpolation method code). This field is used when multiple interpolation tables are linked.

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.71  ST_LON_SNVT_zerospan

Used by: SNVT_zerospan

```
TYPE ST_LON_SNVT_zerospan :
STRUCT
    rZero : REAL;
    rSpan : REAL;
END_STRUCT
END_TYPE
```

**rZero:** Min: -163.840 / Max: 163.835 / Zero-term (16-bit signed value).

**rSpan:** Min: 0.0 / Max: 32.7675 / Span-factor (16-bit unsigned value).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.72  str_AddressTable

Address table

```
TYPE str_AddressTable :
STRUCT
    bType : BOOL;
    Node : USINT;
    bDomain : BOOL;
END_STRUCT
```

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>
bType: Type

Node: Node

bDomain: Domain

Member: Member

RPT_Timer: RPT_Timer

Retry: Retry

RCV_Timer: RCV_Timer

Tx_Timer: Tx_Timer

Group: Group

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_LON from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.3 Resources

4.3.1 Globale_Variablen_LON

Default values for all send function blocks

VAR_GLOBAL CONSTANT

  tMinSendTimeDefault := t#1000ms,
  tMaxSendTimeDefault := t#0s,
  bAutoDefault := FALSE,
  bSendInitDefault := FALSE,
END_VAR

\textbf{tMinSendTimeDefault}: Default value for all send function blocks Applies to \textit{automatic mode} \[\text{630}\]. The modified value is sent once after this time has elapsed.

\textbf{tMaxSendTimeDefault}: Default value for all send function blocks Applies to \textit{automatic mode} \[\text{630}\]. The value is sent once this time has elapsed, irrespective of any change in value.

\textbf{bAutoDefault}: Default value for all send function blocks Switching on \textit{automatic mode} \[\text{630}\].

\textbf{bSendInitDefault}: Default value for all send function blocks After switching on the data is sent once.

4.4 Integration into TwinCAT

4.4.1 KL6401 with CX5120

This example explains how to write a simple PLC program for LON in TwinCAT and how to link it with the hardware. The task is to change the state of a switching output with a button.
Example: https://infosys.beckhoff.com/content/1033/tcplclib_tc2_lon/Resources/zip/6164908043.zip

**Hardware**

**Setting up the components**
- 1x CX5120 Embedded PC
- 1x KL1104 four-channel digital input terminal (for the Set/Reset function)
- 1x KL6401 LON terminal
- 1x KL9010 end terminal

Set up the hardware and the LON components as described in the respective documents.

This example assumes that a Set button was linked to the first KL1104 input and a Reset button to the second.

**Software**

**Creation of the PLC program**

Create a new “TwinCAT XAE project” and a “Standard PLC project”.
Add the library Tc2_LON in the PLC project under “References”.
Generate a global variable list with the name GVL_LON and create the following variables:

```
VAR_GLOBAL
   bSet AT %I* : BOOL;
   bReset AT %I* : BOOL
   stParameter_IN AT %I* : ST_LON_Parameter_IN_36B;
   stParameter_OUT AT %Q* : ST_LON_Parameter_OUT_36B;
   stLON_Com : ST_LON_Communication
END_VAR
```

- **bSet**: Input variable for the Set button.
- **bReset**: Input variable for the Reset button.
- **stParameter_IN**: Input structure for the LON terminal (see ST_LON_Parameter_IN_36B).
- **stParameter_OUT**: Output structures for the LON terminal (see ST_LON_Parameter_OUT_36B).
- **stLON_Com**: Structure for the internal communication with LON (see ST_LON_Communication).

All LON function blocks must be called in the same task.

Create a LON program (CFC) in which the function blocks `FB_LON_KL6401` and `FB_SEND_095_SNVT_switch` are called up. At the communication block link the variables `stParameter_IN`, `stParameter_Out` and `stLON_Com`. 
Link the local variable \( b_{Switch} \) with the global variables \( b_{Set} \) and \( b_{Reset} \), then with the selected input. Link the local variable \( stValue \) (see \( ST\_LON\_SNVT\_switch \) [618]) with the selected output, then with the input \( stValue \) of the send block.

Go to the task configuration and give the task a lower interval time.

Further conditions can be found in the description of the function block \( FB\_LON\_KL6401 \) [27].

**I/O configuration**

Select the CX as target system and initiate a search for its hardware. In the project instance within the PLC section, you can see that the input and output variables are assigned to the input and output variables of the task.
Now link the variables with the inputs and outputs of the Bus Terminals.

The linking of the LON variables is described in detail below.

Right-click the variable `wParameterStatus` of the input structure and select "Change link".

Select the terminal under "I/O Configuration", click "Parameter Status" and confirm with "OK".
In the next step, right-click the structure `stParameterReadValue` within the input structure and select "Change Link".

`wDummy` is not linked.
Select "All Types" and "Continuous", then select "Parameter Input Data 1" to "Parameter Input Data 8" with the left mouse button and the >SHIFT< key. Then click "OK".
Now also link the variables `byParameterType` and `byLONStatus`.

You can now check the connection. To this end, select the KL6401 in the "I/O Configuration" and open it. All terminal data should now show a small arrow. If that is the case, then proceed in exactly the same way with the outputs.
5 Appendix

5.1 Automatic send

Automatic sending is enabled with the input variable bAuto. The variable must be TRUE during the whole interval over which the block is to send independently.

The following three parameters (VAR_INPUT) can be used to influence automatic sending.

MaxSendTime : TIME;

This value enables transfers of values at regular intervals. The value is sent once the time has elapsed, irrespective of any change in value.

ValueLimit : This value can have the following formats: (r/lr/ui/i)ValueLimit : REAL / LREAL / UINT / INT.

The value is only sent if the absolute value of the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value (even very small changes).

Please note: The variable "ValueLimit" is not used for "Enums" and "Structures". In this case sending takes place after each change in value.

tMinSendTime : TIME;

The function block starts sending after tMinSendTime at the earliest. This parameter can be used to limit the number of telegrams in situations where the values change very quickly (to reduce the network load). If the value is t#0s, sending takes place after each change in value (see ValueLimit).
## 5.2 dwErrorKL

<table>
<thead>
<tr>
<th>Return parameter dwErrorKL</th>
<th>FW</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000 0000&lt;sub&gt;hex&lt;/sub&gt;</td>
<td></td>
<td>No error</td>
</tr>
<tr>
<td>0000 0001&lt;sub&gt;hex&lt;/sub&gt;</td>
<td></td>
<td>Write access to a read parameter</td>
</tr>
<tr>
<td>0000 0004&lt;sub&gt;hex&lt;/sub&gt;</td>
<td></td>
<td>Undefined parameter</td>
</tr>
<tr>
<td>0000 0005&lt;sub&gt;hex&lt;/sub&gt;</td>
<td></td>
<td>Illegal value for parameter</td>
</tr>
<tr>
<td>0000 0007&lt;sub&gt;hex&lt;/sub&gt;</td>
<td></td>
<td>Undefined slot</td>
</tr>
<tr>
<td>0000 0008&lt;sub&gt;hex&lt;/sub&gt;</td>
<td></td>
<td>Error when reading the NV parameter</td>
</tr>
<tr>
<td>0000 0009&lt;sub&gt;hex&lt;/sub&gt;</td>
<td></td>
<td>Checksum error when downloading the LON configuration</td>
</tr>
<tr>
<td>0000 0010&lt;sub&gt;hex&lt;/sub&gt;</td>
<td>from FW5</td>
<td>OUTGOING_MSG_FAILED</td>
</tr>
<tr>
<td>0000 0011&lt;sub&gt;hex&lt;/sub&gt;</td>
<td>from FW5</td>
<td>OUTGOING_MSG_LATE_ACK</td>
</tr>
<tr>
<td>0000 0012&lt;sub&gt;hex&lt;/sub&gt;</td>
<td>from FW5</td>
<td>OUTGOING_MSG_MALFORMED</td>
</tr>
<tr>
<td>0000 0013&lt;sub&gt;hex&lt;/sub&gt;</td>
<td>from FW5</td>
<td>NEURON_QUERY_FAILED</td>
</tr>
<tr>
<td>0000 0014&lt;sub&gt;hex&lt;/sub&gt;</td>
<td>from FW5</td>
<td>NEURON_UPDATE_FAILED</td>
</tr>
<tr>
<td>0000 000A&lt;sub&gt;hex&lt;/sub&gt;</td>
<td></td>
<td>Fault with the upload of a LON configuration</td>
</tr>
<tr>
<td>0000 0Cxx&lt;sub&gt;hex&lt;/sub&gt;</td>
<td></td>
<td>An SNVT variable that is to be written has not arrived, xx corresponds to the SNVT index number</td>
</tr>
<tr>
<td>0000 0Exx&lt;sub&gt;hex&lt;/sub&gt;</td>
<td></td>
<td>NV index is not an output. The PLC attempts to write to an NV index that was not defined as an output with the KS2000 (nvo). xx corresponds to the SNVT index number</td>
</tr>
<tr>
<td>0000 0Fx&lt;sub&gt;hex&lt;/sub&gt;</td>
<td>from FW5</td>
<td>Confirmation of the LON telegram to be written has arrived too late (&gt; 1500 ms), xx corresponds to the SNVT index number</td>
</tr>
<tr>
<td>0000 400x&lt;sub&gt;hex&lt;/sub&gt;</td>
<td>from FW5</td>
<td>NEURON_MGMT_ERROR. x corresponds to the API code</td>
</tr>
</tbody>
</table>

## 5.3 Samples

<table>
<thead>
<tr>
<th>Sample</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="https://infosys.beckhoff.com/content/1033/tcplclib_tc2_lon/Resources/zip/9007199900867851.zip">https://infosys.beckhoff.com/content/1033/tcplclib_tc2_lon/Resources/zip/9007199900867851.zip</a></td>
<td>TwinCAT 3 project (all SNVT in ST)</td>
</tr>
<tr>
<td><a href="https://infosys.beckhoff.com/content/1033/tcplclib_tc2_lon/Resources/zip/9007199900865547.zip">https://infosys.beckhoff.com/content/1033/tcplclib_tc2_lon/Resources/zip/9007199900865547.zip</a></td>
<td>TwinCAT 3 project (all SNVT in FBD)</td>
</tr>
<tr>
<td><a href="https://infosys.beckhoff.com/content/1033/tcplclib_tc2_lon/Resources/zip/646081419.zip">https://infosys.beckhoff.com/content/1033/tcplclib_tc2_lon/Resources/zip/646081419.zip</a></td>
<td>Terminal configuration with the KS2000</td>
</tr>
</tbody>
</table>

## 5.4 SNVT-variables (OFF)

<table>
<thead>
<tr>
<th>Description</th>
<th>INPUT/OUTPUT</th>
<th>SNVT ID</th>
<th>Length</th>
<th>NV ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>nviSwitch00</td>
<td>INPUT</td>
<td>95</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>nviSwitch01</td>
<td>INPUT</td>
<td>95</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

https://infosys.beckhoff.com/content/1033/tcplclib_tc2_lon/Resources/zip/150431883.zip

https://infosys.beckhoff.com/content/1033/tcplclib_tc2_lon/Resources/zip/150434827.zip

Implemented SNVT variables (KL6401_OFF)
<table>
<thead>
<tr>
<th>Description</th>
<th>INPUT/OUTPUT</th>
<th>SNVT ID</th>
<th>Length</th>
<th>NV ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>nviSwitch02</td>
<td>INPUT</td>
<td>95</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>nviSwitch03</td>
<td>INPUT</td>
<td>95</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>nviSwitch04</td>
<td>INPUT</td>
<td>95</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>nviSwitch05</td>
<td>INPUT</td>
<td>95</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>nviSwitch06</td>
<td>INPUT</td>
<td>95</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>nviSwitch07</td>
<td>INPUT</td>
<td>95</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>nviSwitch08</td>
<td>INPUT</td>
<td>95</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>nviSwitch09</td>
<td>INPUT</td>
<td>95</td>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>nviSwitch10</td>
<td>INPUT</td>
<td>95</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>nviSwitch11</td>
<td>INPUT</td>
<td>95</td>
<td>2</td>
<td>11</td>
</tr>
<tr>
<td>nviSetting0</td>
<td>INPUT</td>
<td>117</td>
<td>4</td>
<td>12</td>
</tr>
<tr>
<td>nviSetting1</td>
<td>INPUT</td>
<td>117</td>
<td>4</td>
<td>13</td>
</tr>
<tr>
<td>nviSetting2</td>
<td>INPUT</td>
<td>117</td>
<td>4</td>
<td>14</td>
</tr>
<tr>
<td>nviSetting3</td>
<td>INPUT</td>
<td>117</td>
<td>4</td>
<td>15</td>
</tr>
<tr>
<td>nviTemp0</td>
<td>INPUT</td>
<td>105</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>nviTemp1</td>
<td>INPUT</td>
<td>105</td>
<td>2</td>
<td>17</td>
</tr>
<tr>
<td>nviTemp2</td>
<td>INPUT</td>
<td>105</td>
<td>2</td>
<td>18</td>
</tr>
<tr>
<td>nviTemp3</td>
<td>INPUT</td>
<td>105</td>
<td>2</td>
<td>19</td>
</tr>
<tr>
<td>nviTemp4</td>
<td>INPUT</td>
<td>105</td>
<td>2</td>
<td>20</td>
</tr>
<tr>
<td>nviHvacStatus0</td>
<td>INPUT</td>
<td>112</td>
<td>12</td>
<td>21</td>
</tr>
<tr>
<td>nviHvacStatus1</td>
<td>INPUT</td>
<td>112</td>
<td>12</td>
<td>22</td>
</tr>
<tr>
<td>nviHvacStatus2</td>
<td>INPUT</td>
<td>112</td>
<td>12</td>
<td>23</td>
</tr>
<tr>
<td>nviHvacStatus3</td>
<td>INPUT</td>
<td>112</td>
<td>12</td>
<td>24</td>
</tr>
<tr>
<td>nviHvacStatus4</td>
<td>INPUT</td>
<td>112</td>
<td>12</td>
<td>25</td>
</tr>
<tr>
<td>nvoSwitch00</td>
<td>OUTPUT</td>
<td>95</td>
<td>2</td>
<td>26</td>
</tr>
<tr>
<td>nvoSwitch01</td>
<td>OUTPUT</td>
<td>95</td>
<td>2</td>
<td>27</td>
</tr>
<tr>
<td>nvoSwitch02</td>
<td>OUTPUT</td>
<td>95</td>
<td>2</td>
<td>28</td>
</tr>
<tr>
<td>nvoSwitch03</td>
<td>OUTPUT</td>
<td>95</td>
<td>2</td>
<td>29</td>
</tr>
<tr>
<td>nvoSwitch04</td>
<td>OUTPUT</td>
<td>95</td>
<td>2</td>
<td>30</td>
</tr>
<tr>
<td>nvoSwitch05</td>
<td>OUTPUT</td>
<td>95</td>
<td>2</td>
<td>31</td>
</tr>
<tr>
<td>nvoSwitch06</td>
<td>OUTPUT</td>
<td>95</td>
<td>2</td>
<td>32</td>
</tr>
<tr>
<td>nvoSwitch07</td>
<td>OUTPUT</td>
<td>95</td>
<td>2</td>
<td>33</td>
</tr>
<tr>
<td>nvoSwitch08</td>
<td>OUTPUT</td>
<td>95</td>
<td>2</td>
<td>34</td>
</tr>
<tr>
<td>nvoSwitch09</td>
<td>OUTPUT</td>
<td>95</td>
<td>2</td>
<td>35</td>
</tr>
<tr>
<td>nvoSwitch10</td>
<td>OUTPUT</td>
<td>95</td>
<td>2</td>
<td>36</td>
</tr>
<tr>
<td>nvoSwitch11</td>
<td>OUTPUT</td>
<td>95</td>
<td>2</td>
<td>37</td>
</tr>
<tr>
<td>nvoSwitch12</td>
<td>OUTPUT</td>
<td>95</td>
<td>2</td>
<td>38</td>
</tr>
<tr>
<td>nvoSwitch13</td>
<td>OUTPUT</td>
<td>95</td>
<td>2</td>
<td>39</td>
</tr>
<tr>
<td>nvoSwitch14</td>
<td>OUTPUT</td>
<td>95</td>
<td>2</td>
<td>40</td>
</tr>
<tr>
<td>nvoSwitch15</td>
<td>OUTPUT</td>
<td>95</td>
<td>2</td>
<td>41</td>
</tr>
<tr>
<td>nvoSwitch16</td>
<td>OUTPUT</td>
<td>95</td>
<td>2</td>
<td>42</td>
</tr>
<tr>
<td>nvoSwitch17</td>
<td>OUTPUT</td>
<td>95</td>
<td>2</td>
<td>43</td>
</tr>
<tr>
<td>nvoSetting0</td>
<td>OUTPUT</td>
<td>117</td>
<td>4</td>
<td>44</td>
</tr>
<tr>
<td>nvoSetting1</td>
<td>OUTPUT</td>
<td>117</td>
<td>4</td>
<td>45</td>
</tr>
<tr>
<td>nvoSetting2</td>
<td>OUTPUT</td>
<td>117</td>
<td>4</td>
<td>46</td>
</tr>
<tr>
<td>nvoSetting3</td>
<td>OUTPUT</td>
<td>117</td>
<td>4</td>
<td>47</td>
</tr>
<tr>
<td>nvoLevP0</td>
<td>OUTPUT</td>
<td>81</td>
<td>2</td>
<td>48</td>
</tr>
<tr>
<td>nvoLevP1</td>
<td>OUTPUT</td>
<td>81</td>
<td>2</td>
<td>49</td>
</tr>
</tbody>
</table>
5.5 Support and Service

Beckhoff and their partners around the world offer comprehensive support and service, making available fast and competent assistance with all questions related to Beckhoff products and system solutions.

Beckhoff’s branch offices and representatives

Please contact your Beckhoff branch office or representative for local support and service on Beckhoff products!

The addresses of Beckhoff’s branch offices and representatives round the world can be found on her internet pages: https://www.beckhoff.com

You will also find further documentation for Beckhoff components there.

Beckhoff Support

Support offers you comprehensive technical assistance, helping you not only with the application of individual Beckhoff products, but also with other, wide-ranging services:

- support
- design, programming and commissioning of complex automation systems
- and extensive training program for Beckhoff system components

Hotline: +49 5246 963 157
Fax: +49 5246 963 9157
e-mail: support@beckhoff.com

Beckhoff Service

The Beckhoff Service Center supports you in all matters of after-sales service:

- on-site service
- repair service
- spare parts service
- hotline service

Hotline: +49 5246 963 460
Fax: +49 5246 963 479
e-mail: service@beckhoff.com

Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG
Huelshorstweg 20
33415 Verl
Germany

Phone: +49 5246 963 0
Fax: +49 5246 963 198
e-mail: info@beckhoff.com
web: https://www.beckhoff.com