# Table of contents

1 Foreword ................................................................................................................................. 5  
  1.1 Notes on the documentation.............................................................................................. 5  
  1.2 Safety instructions ............................................................................................................ 6  

2 Introduction ...................................................................................................................................... 7  

3 M-Bus ........................................................................................................................................ 8  
  3.1 Topology ................................................................................................................................ 9  
  3.2 Bulletin .................................................................................................................................. 9  
    3.2.1 Functionality of the function block................................................................................ 9  
    3.2.2 Long set ..................................................................................................................... 10  
    3.2.3 Primary address ......................................................................................................... 10  
    3.2.4 Secondary address ................................................................................................... 11  

4 Programming ............................................................................................................................... 12  
  4.1 POUs .................................................................................................................................... 15  
    4.1.1 FB_MBUSKL6781 .................................................................................................... 18  
    4.1.2 ABB overview .......................................................................................................... 19  
    4.1.3 Actaris overview ....................................................................................................... 22  
    4.1.4 Aquametro overview ................................................................................................. 26  
    4.1.5 Berg overview ........................................................................................................... 37  
    4.1.6 Brunata overview ....................................................................................................... 41  
    4.1.7 Carlo Gavazzi overview ............................................................................................ 43  
    4.1.8 Cynox ........................................................................................................................ 46  
    4.1.9 Elster overview .......................................................................................................... 48  
    4.1.10 elvaco overview ....................................................................................................... 50  
    4.1.11 EMH overview ......................................................................................................... 52  
    4.1.12 EMU overview ......................................................................................................... 61  
    4.1.13 Engelmann overview ............................................................................................... 71  
    4.1.14 General device blocks .............................................................................................. 74  
    4.1.15 Gossen Metrawatt overview .................................................................................... 90  
    4.1.16 GWF overview ......................................................................................................... 92  
    4.1.17 Hydrometer overview .............................................................................................. 94  
    4.1.18 ista overview ........................................................................................................... 106  
    4.1.19 Itron .......................................................................................................................... 114  
    4.1.20 Janitza overview ..................................................................................................... 116  
    4.1.21 Kamstrup overview ................................................................................................. 119  
    4.1.22 Kundo overview ...................................................................................................... 132  
    4.1.23 Landis & Gyr overview ............................................................................................ 138  
    4.1.24 Metrima overview ................................................................................................... 141  
    4.1.25 NZR overview .......................................................................................................... 146  
    4.1.26 OPTEC overview ...................................................................................................... 150  
    4.1.27 Relay overview ....................................................................................................... 153  
    4.1.28 Saia-Burgess overview ......................................................................................... 164  
    4.1.29 Schlumberger overview ......................................................................................... 169  
    4.1.30 Schneider Electric ................................................................................................. 174
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1.31</td>
<td>Sensus overview</td>
<td>177</td>
</tr>
<tr>
<td>4.1.32</td>
<td>Sontex overview</td>
<td>182</td>
</tr>
<tr>
<td>4.1.33</td>
<td>TIP</td>
<td>184</td>
</tr>
<tr>
<td>4.1.34</td>
<td>Zenner overview</td>
<td>188</td>
</tr>
<tr>
<td>4.1.35</td>
<td>Error codes</td>
<td>194</td>
</tr>
<tr>
<td>4.2</td>
<td>DUTs</td>
<td>197</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Enums</td>
<td>197</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Structures</td>
<td>202</td>
</tr>
<tr>
<td>4.3</td>
<td>GLVs</td>
<td>207</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Globale_Variablen_MBUS</td>
<td>207</td>
</tr>
<tr>
<td>4.4</td>
<td>Integration into TwinCAT</td>
<td>207</td>
</tr>
<tr>
<td>4.4.1</td>
<td>KL6781 with CX5120</td>
<td>207</td>
</tr>
<tr>
<td>5</td>
<td>Appendix</td>
<td>211</td>
</tr>
<tr>
<td>5.1</td>
<td>Support and Service</td>
<td>211</td>
</tr>
</tbody>
</table>
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®, 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!

- **DANGER**
  - Serious risk of injury!
  - Failure to follow the safety instructions associated with this symbol directly endangers the life and health of persons.

- **WARNING**
  - Risk of injury!
  - Failure to follow the safety instructions associated with this symbol endangers the life and health of persons.

- **CAUTION**
  - Personal injuries!
  - Failure to follow the safety instructions associated with this symbol can lead to injuries to persons.

- **NOTE**
  - Damage to the environment or devices
  - Failure to follow the instructions associated with this symbol can lead to damage to the environment or equipment.

- **Tip or pointer**
  - This symbol indicates information that contributes to better understanding.
2 Introduction

The Tc2_MBUs library is a comprehensive TwinCAT PLC library for reading M-Bus devices.

The application of this PLC library significantly simplifies the engineering in these areas of building technical equipment.

The function blocks are object-oriented and characterized by a self-contained, more or less complex function.

The input parameters form the interface to the user. The parameters can be used to adapt the function block to its specific task within the associated system.

Thanks to strongly object-oriented encapsulation of complex system functions within the function blocks, comprehensive system programs can be set up with a few function blocks. The blocks are linked to each other via a small number of PLC variables.

The status of all objects is indicated through a large number of different output variables at the function blocks. The simplifies the connection of HMI and visualization systems.

These features offer the following benefits for system programmers during system setup and for system operators during operation:

- Faster creation of system programs.
- Faster system parameterization and commissioning.
- Guarantee of a very large range of system functions at all times.
- Improved readability of programs (prerequisite for long-term maintainability and expandability of the systems)
- Improved reusability of templates for systems or system components
- Easier familiarization of personnel.
- Easier extension of existing systems.
- Programs are easier to document.

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

- TwinCAT XAE
- PC and network knowledge
- Structure and properties of the Beckhoff Embedded PC and its Bus Terminal system
- Technology of M-Bus devices
- 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_MBUs library is usable on all hardware platforms that support TwinCAT 3.1 or higher.

Hardware documentation in the Beckhoff information system:

https://infosys.beckhoff.com/content/1033/kl6781/index.html

Preferred format: LReal.

M-Bus devices may supply very large values (the DWord value range may be exceeded). They are therefore output in string format. Conversions to Real format may lead to inaccuracies/invalid values. Conversions to LReal format are therefore preferable.
M-Bus = metering bus

The M-Bus is a fieldbus for the recording consumption data (e.g. energy meters). Further details about M-Bus can be found under www.m-bus.com. The M-Bus is European standard and is described in the EN1434 standard. The data are sent serially from a slave (measuring device) to a master (level converter with PC). Master and slave are connected via a two-wire cable that is protected against polarity reversal. With primary addressing up to 250 slaves can be connected in star, strand or tree topologies. Ring structures are not permitted. Devices from different manufacturers can be operated on the same bus.

The master controls the communication on the bus by requesting data from the slaves. The slaves can respond with a fixed or variable data structure. The M-Bus library only evaluates data with variable data structure (low byte first). The slaves do not communicate with each other. The data have to be requested sequentially from the slaves.
3.1 Topology

Star, line and tree topology

Ring topology

- Ring topology not supported
- Ring topology is possible for M-Bus, but not recommended and therefore also not supported by Beckhoff.

3.2 Bulletin

3.2.1 Functionality of the function block

Three methods for reading M-Bus meters are offered:

1. The variable $t\text{MinSendTime}> t\#0s$ of the meter block is used to read the meter automatically once the time has elapsed. The variable is internally preset to $t\#2s$.
2. A positive edge of the variable $b\text{Start}$ of the meter block triggers one reading of the meter.
3. A positive edge of the variables $b\text{Start}$ of the block $FB\_MBUS\_KL6781()$ triggers one reading of all meters.

If several meter blocks receive a start command at the same time, they are started in the order in which they are called in the PLC.

The variable $b\text{Ready}$ becomes TRUE for one cycle once the block has received the data.

$b\text{Error}$ becomes TRUE if an error has occurred. This error is described with $e\text{Error}$.

To read the meter after a start/ restart of the PLC, set the variable $b\text{ReadInit}$ to TRUE; otherwise to FALSE. Internally this variable is preset to TRUE.

$e\text{Baudrate}$: This variable is internally preset to 2400 baud. To read the meter with this baud rate (2400 baud), this variable does not have to be set explicitly. If the baud rate is changed, the KL6781 is adjusted automatically. This makes it possible to read meters with different baud rates in an M-Bus network. The baud rate of the meters is not changed. They must be able to operate with the baud rate specified here. Some meters operate with automatic baud rate detection. For further information please refer to the user guide of the meter.

$b\text{SND\_NKE}$: Internally this variable is preset to TRUE. SND_NKE is a special telegram to the slave. This telegram triggers an initialization of the receiver. This telegram is important for meters, which send several telegrams. These meters respond to a SND_NKE with the first telegram. If TRUE, the SND_NK telegram is sent before the actual query. If FALSE, the SND_NKE telegram is not sent.
**bDisabled = TRUE** can be used to interrupt processing of the block. If a meter query is in progress, it is completed.

### 3.2.2 Long set

Data is sent to the M-bus device with a long set. The long set is composed of a maximum of 255 bytes and is transferred to the counter with the `FB_MBUS_General_Send()` block.

#### Structure of the protocol:

<table>
<thead>
<tr>
<th>Byte</th>
<th>Long set</th>
<th>Description</th>
<th>Assignment in the &quot;FB_MBUS_General_Send&quot; block</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Start character</td>
<td>68hex</td>
<td>Is added in the block</td>
</tr>
<tr>
<td>2</td>
<td>L field</td>
<td>Length of user data plus 3</td>
<td>Is added in the block</td>
</tr>
<tr>
<td>3</td>
<td>L field</td>
<td>Length of user data plus 3</td>
<td>Is added in the block</td>
</tr>
<tr>
<td>4</td>
<td>Start character</td>
<td>68hex</td>
<td>Is added in the block</td>
</tr>
<tr>
<td>5</td>
<td>C field</td>
<td>Function field</td>
<td>Is transferred to the &quot;byC_Field&quot; input variable</td>
</tr>
<tr>
<td>6</td>
<td>A field</td>
<td>Primary address of the M-Bus device</td>
<td>Is transferred to the &quot;usiAddress&quot; input variable</td>
</tr>
<tr>
<td>7</td>
<td>CI Field</td>
<td>Identifier field</td>
<td>Is transferred to the &quot;byCI_Field&quot; input variable</td>
</tr>
<tr>
<td>8..x</td>
<td>User data (0..240)</td>
<td>User data</td>
<td>Are transferred to the ‘arrData’ input variable</td>
</tr>
<tr>
<td>x+1</td>
<td>Checksum</td>
<td>Checksum</td>
<td>Is added in the block</td>
</tr>
<tr>
<td>x+2</td>
<td>Stop character</td>
<td>16hex</td>
<td>Is added in the block</td>
</tr>
</tbody>
</table>

Only the bytes marked in bold letters need to be transferred to the block.

The user data in the ‘arrData’ array must contain ‘16hex’ as the last character. It is important to ensure that the subsequent bytes are empty.

**Sample:** Changing the primary address at address 14, old address is 0

(*Transfer of user data*)

```plaintext
fbSend.arrData[0]:=16#01; (*DIF / Data format 8-bit integer*)
fbSend.arrData[1]:=16#7A; (*VIF / Change address*)
fbSend.arrData[2]:=14; (*New address = 14*)
fbSend.arrData[3]:=16#16; (*Do not transfer stop character/checksum; they will be calculated in the block*)

fbSend.byC_Field:=16#53; (*C field*)
fbSend.byCI_Field:=16#51; (*CI field*)
fbSend.usiAddress:=0; (*Old address*)

fbSend(iComId:=1, (*Block call*)
bStart=bStart,
bInit:=TRUE);
```

Sending is started with the ‘bStart’ variable.

### 3.2.3 Primary address

The counters are addressed via the primary address. This can be set at the device, via manufacturer software, or with the function blocks `FB_MBUS_ChangeAdr()` and `FB_MBUS_General_send()`.
All meters on a level converter/serial interface must have a unique address (0..250).

**Address 0-250:** Addresses of the devices

**Address 251:** not used at present

**Address 252:** not used at present

**Address 253:** Use of secondary addressing

**Address 254:** Send to all M-bus devices with response (E5 hexadecimal). If several devices are connected, all will answer. This leads to data collisions. Therefore, this address should only be used if only one device is connected.

**Address 255:** Send to all M-Bus devices without response.

### 3.2.4 Secondary address

Like the primary address, the secondary address is used to identify the terminal device. Like the primary address, the secondary address is used to identify the terminal device. The identification number alone can be used to form 100 million different values. In addition, it is not necessary to allocate primary addresses.

A secondary address has the following structure, according to the M-Bus standard:

- **Ident no.:** 4 bytes / 8-digit BCD device ID data
- **Manufacturer code:** 2 bytes / manufacturer code
- **Version:** 1 byte / generation number of the manufacturer
- **Medium:** 1 byte / medium

To use secondary addressing, set the primary address to 253.

The secondary address is transferred to the function block via the structure "stSecAdr" (**ST_MBUS_SecAdr**

The manufacturer code, version and medium are internally preset to 16#FF, so that these values do not have to be specified explicitly.

**Sample calls:**

```plaintext
stSecAdr1.udiIdNumber := 16#12345678;
stSecAdr1.uiManufacturer := 16#FFFF;
stSecAdr1.usiMedium := 16#FF;
stSecAdr1.usiVersion := 16#FF;

fbmeter{
    usiAddress := 253,
    stSecAdr.udiIdNumber := stSecAdr1,
    stCom := stComKL6781_1};
```

or

```plaintext
fbmeter.stSecAdr.udiIdNumber := 16#12345678;
fbmeter{
    usiAddress := 253,
    stCom := stComKL6781_1};
```
4 Programming

The manufacturer-specific blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks `FB_MBUS_General()` [76], `FB_MBUS_General_Ext()` [79] or `FB_MBUS_General_Param()` [83] from the folder "General [74]" should be used. The block `FB_MBUS_General_Send()` [85] can be used to send data to the device (e.g. setting of the primary address).
<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Type</th>
<th>Device</th>
<th>Block</th>
</tr>
</thead>
<tbody>
<tr>
<td>general</td>
<td>Communication with KL6781</td>
<td>FB_MBUSKL6781</td>
<td>18</td>
</tr>
<tr>
<td>ABB [19]</td>
<td>Electricity meter</td>
<td>DELTAplus DZ+</td>
<td>FB_MBUS_ABB_DZ[20]</td>
</tr>
<tr>
<td>Actaris [22]</td>
<td>Heat meter</td>
<td>CF-Echo II</td>
<td>FB_MBUS_ACW_CF [22]</td>
</tr>
<tr>
<td></td>
<td>Arithmetic unit</td>
<td>CF-51</td>
<td>FB_MBUS_ACW_CF [22]</td>
</tr>
<tr>
<td></td>
<td>Arithmetic unit</td>
<td>CF-55</td>
<td>FB_MBUS_ACW_CF [22]</td>
</tr>
<tr>
<td>Aquametro [26]</td>
<td>Water meter</td>
<td>SAPHIR</td>
<td>FB_MBUS_AMT_SAPHIR [35]</td>
</tr>
<tr>
<td></td>
<td>Heat meter</td>
<td>CALEC MB</td>
<td>FB_MBUS_AMT_CALEC [31]</td>
</tr>
<tr>
<td></td>
<td>Heat meter</td>
<td>CALEC ST, version C4</td>
<td>FB_MBUS_AMT_CALEC_STC4 [33]</td>
</tr>
<tr>
<td>Berg [37]</td>
<td>Electricity meter</td>
<td>DZ+</td>
<td>FB_MBUS_BEC_DZ [39]</td>
</tr>
<tr>
<td></td>
<td>Electricity meter</td>
<td>DCMi</td>
<td>FB_MBUS_BEC_DCMi [37]</td>
</tr>
<tr>
<td>Brunata [41]</td>
<td>Heat meter</td>
<td>HGQ / HGS</td>
<td>FB_MBUS_BHG_HGx [41]</td>
</tr>
<tr>
<td></td>
<td>Heat meter</td>
<td>Optuna H (775)</td>
<td>FB_MBUS_HYD_Sharky [101], FB_MBUS_HYD_Sharky_00 [103]</td>
</tr>
<tr>
<td>Cynox [46]</td>
<td>Pulse counter</td>
<td>MCount2C</td>
<td>FB_MBUS_CYN_MCount2C [46]</td>
</tr>
<tr>
<td>Elster [48]</td>
<td>Gas meter</td>
<td>Encoder Z6</td>
<td>FB_MBUS_ELS_ENCODERZ6 [48]</td>
</tr>
<tr>
<td>elvaco [50]</td>
<td>Temperature and humidity sensors</td>
<td>CMa10 &amp; CMa20</td>
<td>FB_MBUS_ELV_CMa10_20 [50]</td>
</tr>
<tr>
<td>EMH [52]</td>
<td>Electricity meter</td>
<td>DIZ</td>
<td>FB_MBUS_EMH_DIZ [53]</td>
</tr>
<tr>
<td></td>
<td>Electricity meter</td>
<td>EIZ-E</td>
<td>FB_MBUS_EMH_EIZE [55]</td>
</tr>
<tr>
<td></td>
<td>Electricity meter</td>
<td>EIZ-G</td>
<td>FB_MBUS_EMH_EIZG [57]</td>
</tr>
<tr>
<td></td>
<td>Electricity meter</td>
<td>MIZ</td>
<td>FB_MBUS_EMH_MIZ [59]</td>
</tr>
<tr>
<td>EMU [61]</td>
<td>Electricity meter</td>
<td>EMU32x7</td>
<td>FB_MBUS_EMU_32x7 [61]</td>
</tr>
<tr>
<td>Manufacturer</td>
<td>Type</td>
<td>Device</td>
<td>Block</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------------</td>
<td>-----------------</td>
<td>--------------------------------------------</td>
</tr>
<tr>
<td>Programming</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Electricity meter</td>
<td>EMU32x7</td>
<td>FB_MBUS_EMU_32x7_Option8 [64]</td>
</tr>
<tr>
<td></td>
<td>Electricity meter</td>
<td>Allrounder 3/5</td>
<td>FB_MBUS_EMU_3_Allrounder [67]</td>
</tr>
<tr>
<td></td>
<td>Electricity meter</td>
<td>DHZ 5/63</td>
<td>FB_MBUS_EMU_DHZ_5_63 [69]</td>
</tr>
<tr>
<td>Engelmann [71]</td>
<td>Heat meter</td>
<td>Sensostar 2C</td>
<td>FB_MBUS_EFF_SensoStar2C [72]</td>
</tr>
<tr>
<td>Gossen Metrawatt [90]</td>
<td>Electricity meter</td>
<td>U128x</td>
<td>FB_MBUS_GMC_Electricity [91]</td>
</tr>
<tr>
<td></td>
<td>Electricity meter</td>
<td>U138x</td>
<td>FB_MBUS_GMC_Electricity [91]</td>
</tr>
<tr>
<td></td>
<td>Water meter</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Gas meter</td>
<td>S1</td>
<td>FB_MBUS_GWF_Coder [93]</td>
</tr>
<tr>
<td></td>
<td>Gas meter</td>
<td>Z1</td>
<td>FB_MBUS_GWF_Coder [93]</td>
</tr>
<tr>
<td>Hydrometer [94]</td>
<td>2 pulse inputs</td>
<td>HYDRO-PORT Pulse</td>
<td>FB_MBUS_HYD_PorPulse [99]</td>
</tr>
<tr>
<td></td>
<td>2 analog inputs 1 temperature sensor</td>
<td>HYDRO-PORT Analog</td>
<td>FB_MBUS_HYD_PorAnalog [97]</td>
</tr>
<tr>
<td></td>
<td>Water meter</td>
<td>Flypper</td>
<td>FB_MBUS_HYD_Flypper [95]</td>
</tr>
<tr>
<td></td>
<td>Heat meter</td>
<td>Sharky 773</td>
<td>FB_MBUS_HYD_Sharky [101], FB_MBUS_HYD_Sharky_00 [103]</td>
</tr>
<tr>
<td></td>
<td>Heat meter</td>
<td>Sharky 775</td>
<td>FB_MBUS_HYD_Sharky [101], FB_MBUS_HYD_Sharky_00 [103]</td>
</tr>
<tr>
<td>ista [106]</td>
<td>Water meter</td>
<td>domaqua® m</td>
<td>FB_MBUS_JST_Istameter [106]</td>
</tr>
<tr>
<td></td>
<td>Water meter</td>
<td>istameter® m</td>
<td>FB_MBUS_JST_Istameter [106]</td>
</tr>
<tr>
<td></td>
<td>Water meter</td>
<td>istameter III</td>
<td>FB_MBUS_JST_IstameterIII [108]</td>
</tr>
<tr>
<td></td>
<td>Pulse counter</td>
<td>pulsonic II</td>
<td>FB_MBUS_JST_PulsonicII [110]</td>
</tr>
<tr>
<td></td>
<td>Heat meter</td>
<td>sensonic II</td>
<td>FB_MBUS_JST_SensonicII [112]</td>
</tr>
<tr>
<td>Itron [114]</td>
<td>Energy meter</td>
<td>Integral-V-UltraLite</td>
<td>FB_MBUS_ITR_IntegralVUltraLite [114]</td>
</tr>
<tr>
<td>Janitza [116]</td>
<td>Electricity meter</td>
<td>UMG96S</td>
<td>FB_MBUS_JAN_UMG96S [117]</td>
</tr>
<tr>
<td>Kamstrup [119]</td>
<td>Electricity meter</td>
<td>Kamstrup 162</td>
<td>FB_MBUS_KAM_KamstrupE [120]</td>
</tr>
<tr>
<td></td>
<td>Electricity meter</td>
<td>Kamstrup 351</td>
<td>FB_MBUS_KAM_KamstrupE [120]</td>
</tr>
<tr>
<td></td>
<td>Electricity meter</td>
<td>Kamstrup 382</td>
<td>FB_MBUS_KAM_KamstrupE [120]</td>
</tr>
<tr>
<td></td>
<td>Heat/cold meter</td>
<td>Maxical III</td>
<td>FB_MBUS_KAM_Maxical III [122]</td>
</tr>
<tr>
<td></td>
<td>Heat/cold meter</td>
<td>Multical 401</td>
<td>FB_MBUS_KAM_Multical [124]</td>
</tr>
<tr>
<td></td>
<td>Heat/cold meter</td>
<td>Multical 402</td>
<td>FB_MBUS_KAM_Multical402 [126]</td>
</tr>
<tr>
<td></td>
<td>Water meter</td>
<td>Multical 41</td>
<td>FB_MBUS_KAM_Multical41 [128]</td>
</tr>
<tr>
<td></td>
<td>Heat/cold meter</td>
<td>Multical 601</td>
<td>FB_MBUS_KAM_Multical601 [130]</td>
</tr>
<tr>
<td>KUNDO [132]</td>
<td>Heat/cold meter</td>
<td>Compact WMZ G20</td>
<td>FB_MBUS_KST_G20 [133]</td>
</tr>
<tr>
<td></td>
<td>Heat/cold meter</td>
<td>Compact WMZ G21</td>
<td>FB_MBUS_KST_G20 [133]</td>
</tr>
<tr>
<td></td>
<td>External M-Bus module</td>
<td>him1s</td>
<td>FB_MBUS_KST_him1 [135]</td>
</tr>
<tr>
<td></td>
<td>External M-Bus module</td>
<td>him1plus</td>
<td>FB_MBUS_KST_him1plus [135]</td>
</tr>
<tr>
<td>Landis &amp; Gyr [138]</td>
<td>Heat/cold meter</td>
<td>ULTRAHEAT 2WR5</td>
<td>FB_MBUS_LUG_Heat [139]</td>
</tr>
<tr>
<td></td>
<td>Heat/cold meter</td>
<td>ULTRAHEAT 2WR6</td>
<td>FB_MBUS_LUG_Heat [139]</td>
</tr>
<tr>
<td>Manufacturer</td>
<td>Type</td>
<td>Device</td>
<td>Block</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------------------------</td>
<td>-----------------------------</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td>Metrima [141]</td>
<td>Heat/cold meter</td>
<td>ULTRAHEAT UH50</td>
<td>FB_MBUS_LUG_Heat [139]</td>
</tr>
<tr>
<td>NZR [146]</td>
<td>Heat meter</td>
<td>F22 (standard values)</td>
<td>FB_MBUS_SVM_F22 [141]</td>
</tr>
<tr>
<td>Metrima [141]</td>
<td>Heat meter</td>
<td>F22 (with additional output values)</td>
<td>FB_MBUS_SVM_F22_Ext [144]</td>
</tr>
<tr>
<td>Metrima [141]</td>
<td>Pulse memory module</td>
<td>IC-M2</td>
<td>FB_MBUS_NZR_ICM2 [147]</td>
</tr>
<tr>
<td>Metrima [141]</td>
<td>Pulse memory module</td>
<td>IC-M2C</td>
<td>FB_MBUS_NZR_ICM2 [147]</td>
</tr>
<tr>
<td>Metrima [141]</td>
<td>Water meter</td>
<td>Modularis 2</td>
<td>FB_MBUS_NZR_Modularis2 [149]</td>
</tr>
<tr>
<td>Metrima [141]</td>
<td>Electricity meter</td>
<td>ECS Type 2</td>
<td>FB_MBUS_OPT_ECSType2 [151]</td>
</tr>
<tr>
<td>Metrima [141]</td>
<td>1-4 analog inputs</td>
<td>AnDi 1-4</td>
<td>FB_MBUS_REL_AnDi [154]</td>
</tr>
<tr>
<td>Metrima [141]</td>
<td>1-way pulse adapter</td>
<td>PadPuls M1</td>
<td>FB_MBUS_REL_PadPulsM1 [158]</td>
</tr>
<tr>
<td>Metrima [141]</td>
<td>1-way pulse adapter</td>
<td>PadPuls M1C</td>
<td>FB_MBUS_REL_PadPulsM1 [158]</td>
</tr>
<tr>
<td>Metrima [141]</td>
<td>2-way pulse adapter</td>
<td>PadPuls M2</td>
<td>FB_MBUS_REL_PadPulsM2 [160]</td>
</tr>
<tr>
<td>Metrima [141]</td>
<td>2-way pulse adapter</td>
<td>PadPuls M2C</td>
<td>FB_MBUS_REL_PadPulsM2 [160]</td>
</tr>
<tr>
<td>Metrima [141]</td>
<td>4-way pulse adapter</td>
<td>PadPuls M4</td>
<td>FB_MBUS_REL_PadPulsM4 [162]</td>
</tr>
<tr>
<td>Metrima [141]</td>
<td>4-way pulse adapter</td>
<td>PadPuls M4L</td>
<td>FB_MBUS_REL_PadPulsM4 [162]</td>
</tr>
<tr>
<td>Metrima [141]</td>
<td>Electricity meter</td>
<td>ALD1</td>
<td>FB_MBUS_SBC_ALD1 [165]</td>
</tr>
<tr>
<td>Metrima [141]</td>
<td>Electricity meter</td>
<td>ALE3</td>
<td>FB_MBUS_SBC_ALE3 [167]</td>
</tr>
<tr>
<td>Metrima [141]</td>
<td>Electricity meter</td>
<td>AWD3</td>
<td>FB_MBUS_SBC_ALE3 [167]</td>
</tr>
<tr>
<td>Metrima [141]</td>
<td>Heat meter</td>
<td>Integral-Mk MaXX</td>
<td>FB_MBUS_SLB_MK_MaXX [172]</td>
</tr>
<tr>
<td>Metrima [141]</td>
<td>Heat meter</td>
<td>CF Echo I</td>
<td>FB_MBUS_SLB_CFEchoI [170]</td>
</tr>
<tr>
<td>Metrima [141]</td>
<td>Electricity meter</td>
<td>iEM3135</td>
<td>FB_MBUS_SEC_iEM3135 [175]</td>
</tr>
<tr>
<td>Metrima [141]</td>
<td>Heat/cold meter</td>
<td>PolluTherm</td>
<td>FB_MBUS_SEN_Pollu [178]</td>
</tr>
<tr>
<td>Metrima [141]</td>
<td>Heat/cold meter</td>
<td>PolluCom E</td>
<td>FB_MBUS_SEN_Pollu [178]</td>
</tr>
<tr>
<td>Metrima [141]</td>
<td>Water meter</td>
<td>FB_MBUS_SEN_Water [180]</td>
<td></td>
</tr>
<tr>
<td>Metrima [141]</td>
<td>Heat/cold meter</td>
<td>Supercal 531</td>
<td>FB_MBUS_SON_Supercal531 [182]</td>
</tr>
<tr>
<td>Metrima [141]</td>
<td>Electricity meter</td>
<td>SINUS 85 M</td>
<td>FB_MBUS_TIP_SINUS85M [185]</td>
</tr>
<tr>
<td>Metrima [141]</td>
<td>Arithmetic unit</td>
<td>multidataWR3</td>
<td>FB_MBUS_ZRM_multidataWR3 [189]</td>
</tr>
<tr>
<td>Metrima [141]</td>
<td>Heat meter</td>
<td>zelsiusZR</td>
<td>FB_MBUS_ZRM_zelsiusZR [191]</td>
</tr>
</tbody>
</table>

### 4.1 POU$s$

The manufacturer-specific blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General() [76], FB_MBUS_General_Ext() [79] or FB_MBUS_General_Param() [83] from the folder "General [74]" should be used. The block FB_MBUS_General_Send() [85] can be used to send data to the device (e.g. setting of the primary address).
<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Type</th>
<th>Device</th>
<th>Function block</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td>Communication with KL6781</td>
<td></td>
<td>FB_MBUSKL6781 [18]</td>
</tr>
<tr>
<td>General [74]</td>
<td>Electricity meter</td>
<td>all electricity meters</td>
<td>FB_MBUS_General_Electricity [77]</td>
</tr>
<tr>
<td></td>
<td>Heat meter</td>
<td>all heat meters</td>
<td>FB_MBUS_General_Heat [81]</td>
</tr>
<tr>
<td></td>
<td>Water meter</td>
<td>all water meters</td>
<td>FB_MBUS_General_Water [86]</td>
</tr>
<tr>
<td></td>
<td>Raw data of the first telegram</td>
<td>all</td>
<td>FB_MBUS_RawData [88]</td>
</tr>
<tr>
<td></td>
<td>max. 40 values from the first telegram</td>
<td>all</td>
<td>FB_MBUS_General [76]</td>
</tr>
<tr>
<td></td>
<td>all telegrams for all values</td>
<td>all</td>
<td>FB_MBUS_General_Ext [79]</td>
</tr>
<tr>
<td></td>
<td>values parameterisable</td>
<td>all</td>
<td>FB_MBUS_General_Param [83]</td>
</tr>
<tr>
<td></td>
<td>universal send blocks</td>
<td>all</td>
<td>FB_MBUS_General_Send [85]</td>
</tr>
<tr>
<td></td>
<td>scan block</td>
<td>all</td>
<td>FB_MBUS_Scan [89]</td>
</tr>
<tr>
<td></td>
<td>Change primary address</td>
<td>all</td>
<td>FB_MBUS_ChangeAdr [75]</td>
</tr>
<tr>
<td>ABB [19]</td>
<td>Electricity meter</td>
<td>DELTAplus DZ+</td>
<td>FB_MBUS_ABB_DZ [20]</td>
</tr>
<tr>
<td>Actaris [22]</td>
<td>Heat meter</td>
<td>CF-Echo II</td>
<td>FB_MBUS_ACW_CF [22]</td>
</tr>
<tr>
<td></td>
<td>Arithmetic unit</td>
<td>CF-51</td>
<td>FB_MBUS_ACW_CF [22]</td>
</tr>
<tr>
<td></td>
<td>Arithmetic unit</td>
<td>CF-55</td>
<td>FB_MBUS_ACW_CF [22]</td>
</tr>
<tr>
<td></td>
<td>Water meter</td>
<td>MB +M</td>
<td>FB_MBUS_ACW_PlusM [24]</td>
</tr>
<tr>
<td>Aquametro [26]</td>
<td>Water meter</td>
<td>SAPHIR</td>
<td>FB_MBUS_AMT_SAPHIR [35]</td>
</tr>
<tr>
<td></td>
<td>Heat meter</td>
<td>CALEC MB</td>
<td>FB_MBUS_AMT_CALEC [31]</td>
</tr>
<tr>
<td></td>
<td>Heat meter</td>
<td>CALEC ST, version C4</td>
<td>FB_MBUS_AMT_CALEC STC4 [33]</td>
</tr>
<tr>
<td></td>
<td>Heat meter</td>
<td>AMTRON</td>
<td>FB_MBUS_AMT_AMTRON [29]</td>
</tr>
<tr>
<td></td>
<td>Pulse collector</td>
<td>AMBUS</td>
<td>FB_MBUS_AMT_AMBUS [27]</td>
</tr>
<tr>
<td></td>
<td>Heat meter</td>
<td>AMTRON SONIC D</td>
<td>FB_MBUS_HYD_Sharky [101],</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FB_MBUS_HYD_Sharky_00 [103]</td>
</tr>
<tr>
<td>Berg [37]</td>
<td>Electricity meter</td>
<td>DZ+</td>
<td>FB_MBUS_BEC_DZ [39]</td>
</tr>
<tr>
<td>Brunata [41]</td>
<td>Heat meter</td>
<td>HGQ / HGS</td>
<td>FB_MBUS_BHG_HGx [41]</td>
</tr>
<tr>
<td></td>
<td>Heat meter</td>
<td>Optuna H (775)</td>
<td>FB_MBUS_HYD_Sharky [101],</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FB_MBUS_HYD_Sharky_00 [103]</td>
</tr>
<tr>
<td>Cynox [46]</td>
<td>Pulse counter</td>
<td>MCount2C</td>
<td>FB_MBUS_CYM_MCount2C [46]</td>
</tr>
<tr>
<td>Elster [48]</td>
<td>Gas meter</td>
<td>Encoder Z6</td>
<td>FB_MBUS_ELS_EncoderZ6 [48]</td>
</tr>
<tr>
<td>elvaco [50]</td>
<td>Temperature and humidity sensors</td>
<td>CMa10 &amp; CMa20</td>
<td>FB_MBUS_ELV_CMa10-20 [50]</td>
</tr>
<tr>
<td>EMH [52]</td>
<td>Electricity meter</td>
<td>DIZ</td>
<td>FB_MBUS_EMH_DIZ [53]</td>
</tr>
<tr>
<td></td>
<td>Electricity meter</td>
<td>EIZ-E</td>
<td>FB_MBUS_EMH_EIZ-E [55]</td>
</tr>
<tr>
<td></td>
<td>Electricity meter</td>
<td>EIZ-G</td>
<td>FB_MBUS_EMH_EIZ-G [57]</td>
</tr>
<tr>
<td></td>
<td>Electricity meter</td>
<td>MIZ</td>
<td>FB_MBUS_EMH_MIZ [59]</td>
</tr>
<tr>
<td>EMU [61]</td>
<td>Electricity meter</td>
<td>EMU32x7</td>
<td>FB_MBUS_EMU_32x7_Option8 [61]</td>
</tr>
<tr>
<td></td>
<td>Electricity meter</td>
<td>EMU32x7</td>
<td>FB_MBUS_EMU_32x7_Option8 [64]</td>
</tr>
<tr>
<td>Manufacturer</td>
<td>Type</td>
<td>Device</td>
<td>Function block</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------</td>
<td>-------------------------</td>
<td>-----------------------------------------------------</td>
</tr>
<tr>
<td>Engelmann [71]</td>
<td>Heat meter</td>
<td>Sensostar 2C</td>
<td>FB_MBUS_EFF_SensoStar2C [72]</td>
</tr>
<tr>
<td>Gossen Metrawatt  [90]</td>
<td>Electricity meter</td>
<td>U128x</td>
<td>FB_MBUS_GMC_Electricity [91]</td>
</tr>
<tr>
<td>GWF [92]</td>
<td>Electricity meter</td>
<td>U138x</td>
<td>FB_MBUS_GMC_Electricity [91]</td>
</tr>
<tr>
<td>Hydrometer [94]</td>
<td>2 pulse inputs</td>
<td>HYDRO-PORT Pulse</td>
<td>FB_MBUS_HYD_PortPulse [99]</td>
</tr>
<tr>
<td></td>
<td>2 analog inputs 1</td>
<td>HYDRO-PORT Analog</td>
<td>FB_MBUS_HYD_PortAnalog [97]</td>
</tr>
<tr>
<td></td>
<td>temperature sensor</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GWF</td>
<td>Water meter</td>
<td>Flypper</td>
<td>FB_MBUS_HYD_Flypper [95]</td>
</tr>
<tr>
<td></td>
<td>Heat meter</td>
<td>Sharky 773</td>
<td>FB_MBUS_HYD_Sharky [101], FB_MBUS_HYD_Sharky_00 [103]</td>
</tr>
<tr>
<td></td>
<td>Heat meter</td>
<td>Sharky 775</td>
<td>FB_MBUS_HYD_Sharky [101], FB_MBUS_HYD_Sharky_00 [103]</td>
</tr>
<tr>
<td>Ista [106]</td>
<td>Water meter</td>
<td>domaqua® m</td>
<td>FB_MBUS_IST_Istameter [106]</td>
</tr>
<tr>
<td></td>
<td>Water meter</td>
<td>istameter® m</td>
<td>FB_MBUS_IST_Istameter [106]</td>
</tr>
<tr>
<td></td>
<td>Water meter</td>
<td>istameter® III</td>
<td>FB_MBUS_IST_IstameterIII [108]</td>
</tr>
<tr>
<td></td>
<td>Pulse counter</td>
<td>pulsonic II</td>
<td>FB_MBUS_IST_PulsonicII [110]</td>
</tr>
<tr>
<td></td>
<td>Heat meter</td>
<td>sensonic II</td>
<td>FB_MBUS_IST_SensonicII [112]</td>
</tr>
<tr>
<td>Itron [114]</td>
<td>Energy meter</td>
<td>Integral-V UltraLite</td>
<td>FB_MBUS_ITR_IntegralVUltraLite [114]</td>
</tr>
<tr>
<td>Janitza [116]</td>
<td>Electricity meter</td>
<td>UMG96S</td>
<td>FB_MBUS_JAN_UMG96S [117]</td>
</tr>
<tr>
<td>Kamstrup [119]</td>
<td>Electricity meter</td>
<td>Kamstrup 162</td>
<td>FB_MBUS_KAM_KamstrupE [120]</td>
</tr>
<tr>
<td></td>
<td>Electricity meter</td>
<td>Kamstrup 351</td>
<td>FB_MBUS_KAM_KamstrupE [120]</td>
</tr>
<tr>
<td></td>
<td>Electricity meter</td>
<td>Kamstrup 382</td>
<td>FB_MBUS_KAM_KamstrupE [120]</td>
</tr>
<tr>
<td></td>
<td>Heat/cold meter</td>
<td>Maxical III</td>
<td>FB_MBUS_KAM_Maxical_III [122]</td>
</tr>
<tr>
<td></td>
<td>Heat/cold meter</td>
<td>Multical 401</td>
<td>FB_MBUS_KAM_Multical [124]</td>
</tr>
<tr>
<td></td>
<td>Heat/cold meter</td>
<td>Multical 402</td>
<td>FB_MBUS_KAM_Multical402 [126]</td>
</tr>
<tr>
<td></td>
<td>Water meter</td>
<td>Multical 41</td>
<td>FB_MBUS_KAM_Multical41 [128]</td>
</tr>
<tr>
<td></td>
<td>Heat/cold meter</td>
<td>Multical 601</td>
<td>FB_MBUS_KAM_Multical601 [130]</td>
</tr>
<tr>
<td>KUNDO [132]</td>
<td>Heat/cold meter</td>
<td>Compact WMZ G20</td>
<td>FB_MBUS_KST_G20 [133]</td>
</tr>
<tr>
<td></td>
<td>Heat/cold meter</td>
<td>Compact WMZ G21</td>
<td>FB_MBUS_KST_G20 [133]</td>
</tr>
<tr>
<td></td>
<td>External M-Bus module</td>
<td>him1s</td>
<td>FB_MBUS_KST_him1 [135]</td>
</tr>
<tr>
<td></td>
<td>External M-Bus module</td>
<td>him1plus</td>
<td>FB_MBUS_KST_him1 [135]</td>
</tr>
<tr>
<td></td>
<td>Pulse input</td>
<td>him1plus</td>
<td>FB_MBUS_KST_him1Puls [137]</td>
</tr>
<tr>
<td>Landis &amp; Gyr [138]</td>
<td>Heat/cold meter</td>
<td>ULTRAHEAT 2WR5</td>
<td>FB_MBUS_LUG_Heat [139]</td>
</tr>
<tr>
<td></td>
<td>Heat/cold meter</td>
<td>ULTRAHEAT 2WR6</td>
<td>FB_MBUS_LUG_Heat [139]</td>
</tr>
<tr>
<td></td>
<td>Heat/cold meter</td>
<td>ULTRAHEAT UH50</td>
<td>FB_MBUS_LUG_Heat [139]</td>
</tr>
<tr>
<td>Metrima [141]</td>
<td>Heat meter</td>
<td>F22 (default values)</td>
<td>FB_MBUS_SVM_F22 [141]</td>
</tr>
</tbody>
</table>
4.1.1 **FB_MBUSKL6781**

This function block is used to read M-Bus devices via the Bus Terminal KL6781. The block can only be used in conjunction with at least one **counter block**. An instance of this block required for each KL6781 terminal.
At 2400 baud the maximum task time for this block is 10 ms. If higher task times are required, this block must be processed in a separate fast task.

**VAR_INPUT**

```plaintext
usiRetries : USINT;
bStart      : BOOL;
bDisabled   : BOOL := FALSE;
```

**usiRetries**: Number of repetitions in the event of errors

**bStart**: A positive edge at this input triggers one reading of all meters.

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

**VAR_OUTPUT**

```plaintext
bBusy       : BOOL;
bReady      : BOOL;
bError      : BOOL;
eError      : E_MBUS_ERROR;
```

**bBusy**: The `bBusy` output is TRUE while the meter is being read.

**bReady**: The `bReady` output is TRUE for one cycle, once meter reading is completed.

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

**eError**: In the event of an error the output issues an error code (see `E_MBUS_ERROR`). `bError` goes TRUE at the same time.

**VAR_IN_OUT**

```plaintext
stComIn     : ST_KL6781inData22B;
stComOut    : ST_KL6781outData22B;
stCom       : ST_MBUS_Communication;
```

**stComIn**: Process image of the inputs (see `ST_KL6781inData22B`).

**stComOut**: Process image of the outputs (see `ST_KL6781outData22B`).

**stCom**: This structure is used to link the block with the meter blocks (see `ST_MBUS_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_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.2 ABB overview**

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks `FB_MBUS_General` or `FB_MBUS_General_Param` from the folder "General" should be used. Note that these blocks do not run BC and BX systems. The block `FB_MBUS_General_Send()` can be used to send data to the device (e.g. setting of the primary address).

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Type</th>
<th>Device</th>
<th>Block</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABB</td>
<td>Electricity meter</td>
<td>DELTApplus DZ+</td>
<td><code>FB_MBUS_ABB_DZ</code></td>
</tr>
</tbody>
</table>


4.1.2.1 **FB_MBUS_ABB_DZ**

This block is used to read electricity meters from ABB:

- **DELTAplus DZ+**

The block can only be executed together with the block **FB_MBUSKL6781()** [18].

**Functionality of the block [9]**

### VAR_INPUT

- **usiAddress**: Primary address [10] of the meter to be read with this block.
- **stSecAdr**: Secondary address [11] of the meter to be read with this block (see **ST_MBUS_SecAdr** [205]).
- **eBaudrate**: 300, 600, 1200, 2400, 4800, 9600 baud (see **E_MBUS_Baudrate** [197]).
- **bStart**: A positive edge of this input triggers one reading of the meter.
- **bSND_NKE**: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).
- **bReadInit**: If the PLC is restarted, the meter is read once.
- **tMinSendTime**: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.
- **usiUnit**: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2=MW(h) / 3=GW(h).
- **bDisabled**: TRUE = deselection of the block.
VAR_OUTPUT

bBusy : BOOL;
bReady : BOOL;
bError : BOOL;
eError : E_MBUS_ERROR;
dwIdNumber : DWORD;
byStatus : BYTE;
byGEN : BYTE;
byCounter : BYTE;
usiRecivedAdr : USINT;
eMedium : E_MBUS_Medium;
eman : STRING(3);
stActiveEnergy : ST_MBus_Info;
stReactiveEnergy : ST_MBus_Info;
stActivePower : ST_MBus_Info;
stReactivePower : ST_MBus_Info;
stCurrentL1 : ST_MBus_Info;
stCurrentL2 : ST_MBus_Info;
stCurrentL3 : ST_MBus_Info;
stPowerFactor : ST_MBus_Info;

bBusy: The bBusy output is TRUE while the meter is being read.

bReady: The bReady output is TRUE for one cycle, once meter reading is completed.

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

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [ 197 ]). bError goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [ 200 ]). 

sMan: Manufacturer code.

stActiveEnergy: Meter value, total active energy (see ST_MBus_Info [ 204 ]).

stReactiveEnergy: Meter value, total reactive energy (see ST_MBus_Info [ 204 ]).

stActivePower: Current consumption value, total effective power (see ST_MBus_Info [ 204 ]).

stReactivePower: Current consumption value, total reactive power (see ST_MBus_Info [ 204 ]).

stCurrentL1: Current L1 (see ST_MBus_Info [ 204 ]).

stCurrentL2: Current L2 (see ST_MBus_Info [ 204 ]).

stCurrentL3: Current L3 (see ST_MBus_Info [ 204 ]).

stPowerFactor: Total power factor (see ST_MBus_Info [ 204 ]).

VAR_IN_OUT

stCom : ST_MBUS_Communication;

stCom: This structure is used to link the block FB_MBUSKL6781 [ 202 ] with the meter blocks (see ST_MBUS_Communication [ 202 ]).
### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

#### 4.1.3 Actaris overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [76], FB_MBUS_General_Ext [79] or FB_MBUS_General_Param [83] from the folder "General [74]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [85] can be used to send data to the device (e.g. setting of the primary address).

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Type</th>
<th>Device</th>
<th>Block</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actaris</td>
<td>Heat meter</td>
<td>CF-Echo II</td>
<td>FB_MBUS_ACW_CF [22]</td>
</tr>
<tr>
<td></td>
<td>Arithmetic unit</td>
<td>CF-51</td>
<td>FB_MBUS_ACW_CF [22]</td>
</tr>
<tr>
<td></td>
<td>Arithmetic unit</td>
<td>CF-55</td>
<td>FB_MBUS_ACW_CF [22]</td>
</tr>
<tr>
<td></td>
<td>Water meter</td>
<td>MB +M</td>
<td>FB_MBUS_ACW_PlusM [24]</td>
</tr>
</tbody>
</table>

#### 4.1.3.1 FB_MBUS_ACW_CF

This block is used to read heat meters from Actaris:

- CF-Echo II
- CF-51
- CF-55

Up to two additional water meters can be connected to this device (optional).
The block can only be executed together with the block FB_MBUSKL6781) [18].

Functionality of the block [9]

VAR_INPUT

usiAddress: USINT;
stSecAdr: ST_MBUS_SecAdr;
eBaudrate: E_MBUS_Baudrate := eMBUS_Baud2400;
bStart: BOOL;
bSND_NKE: BOOL := TRUE;
bReadInit: BOOL := TRUE;
tMinSendTime: TIME := t#2s;
usiUnit: USINT;
bDisabled: BOOL := FALSE;

usiAddress: Primary address [10] of the meter to be read with this block.

stSecAdr: Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAdr [205]).

eBaudrate: 300, 1200, 2400, 9600 Baud.

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy: BOOL;
bReady: BOOL;
bError: BOOL;
eError: E_MBUS_ERROR;
dwldNumber: DWORD;
byStatus: BYTE;
byGEN: BYTE;
byCounter: BYTE;
usiRecivedAdr: USINT;
eMedium: E_MBUS_Medium;
zMan: STRING(3);
stEnergy: ST_MBus_Info;
stPower: ST_MBus_Info;
stVolume: ST_MBus_Info;
stFlow: ST_MBus_Info;
stForwardTemp: ST_MBus_Info;
stReturnTemp: ST_MBus_Info;
stDiffTemp: ST_MBus_Info;
stVolume1: ST_MBus_Info;
stVolume2: ST_MBus_Info;
stColdEnergy: ST_MBus_Info;

bBusy: The bBusy output is TRUE while the meter is being read.

bReady: The bReady output is TRUE for one cycle, once meter reading is completed.

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

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.

dlwdNumber: Serial number of the meter (secondary address).

byStatus: Device status.
byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAddr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [200]).

sMan: Manufacturer code.

stEnergy: Meter reading, energy consumption (see ST_MBus_Info [204]).

stPower: Current energy consumption, power (see ST_MBus_Info [204]).

stVolume: Meter reading, water consumption (see ST_MBus_Info [204]).

stFlow: Current water consumption (see ST_MBus_Info [204]).

stForwardTemp: Flow temperature (see ST_MBus_Info [204]).

stReturnTemp: Return temperature (see ST_MBus_Info [204]).

stDiffTemp: Temperature difference (see ST_MBus_Info [204]).

stVolume1: Meter reading of additional water meter 1 (option) (see ST_MBus_Info [204]).

stVolume2: Meter reading of additional water meter 2 (option) (see ST_MBus_Info [204]).

stColdEnergy: Meter reading, cooling energy consumption (option) (see ST_MBus_Info [204]).

VAR_IN_OUT

stCom: This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Communication [202]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.3.2 FB_MBUS_ACW_PlusM

This block is used to read water meters from Actaris:
The block can only be executed together with the block `FB_MBUSK16781` [18].

**Functionality of the block [9]**

**VAR_INPUT**

```plaintext
usiAddress : USINT;
stSecAdr   : ST_MBUS_SecAdr;
eBaudrate  : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart     : BOOL;
bSND_NKE   : BOOL := TRUE;
bReadInit  : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit    : USINT;
bDisabled  := BOOL := FALSE;
```

- **usiAddress:** Primary address [10] of the meter to be read with this block.
- **stSecAdr:** Secondary address [11] of the meter to be read with this block (see `ST_MBUS_SecAdr` [205]).
- **eBaudrate:** 300, 2400 baud (see `E_MBUS_Baudrate` [197]).
- **bStart:** A positive edge of this input triggers one reading of the meter.
- **bSND_NKE:** TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).
- **bReadInit:** If the PLC is restarted, the meter is read once.
- **tMinSendTime:** Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with `bStart`.
- **usiUnit:** Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).
- **bDisabled:** TRUE = deselection of the block.

**VAR_OUTPUT**

```plaintext
bBusy         : BOOL;
bReady        : BOOL;
bError        : BOOL;
eError        : E_MBUS_ERROR;
dwldNumber    : DWORD;
byStatus      : BYTE;
byGEN         : BYTE;
byCounter     : BYTE;
usiRecivedAdr : USINT;
eMedium       : E_MBUS_Medium;
zMan          : STRING(3);
stVolume      : ST_MBus_Info;
```

- **bBusy:** The `bBusy` output is TRUE while the meter is being read.
- **bReady:** The `bReady` output is TRUE for one cycle, once meter reading is completed.
- **bError:** this output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.
- **eError:** In the event of an error the output issues an error code (see `E_MBUS_ERROR` [197]). `bError` goes TRUE at the same time.
- **dwldNumber:** Serial number of the meter (secondary address).
- **byStatus:** Device status.
- **byGEN:** Software version of the device.
- **byCounter:** Number of times the master has accessed data of the respective slave.
- **usiRecivedAdr:** Received primary address (0-250).
eMedium: Medium (see E_MBUS_Medium [201]).

sMan: Manufacturer code.

stVolume: Meter reading, water consumption (see ST_MBus_Info [204]).

VAR_IN_OUT

stCom : ST_MBUS_Communication;

stCom: This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Communication [202]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.4 Aquametro overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [76], FB_MBUS_General_Ext [79] or FB_MBUS_General_Param [83] from the folder "General [74]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [85] can be used to send data to the device (e.g. setting of the primary address).

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Type</th>
<th>Device</th>
<th>Block</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aquametro</td>
<td>Water meter</td>
<td>SAPHIR</td>
<td>FB_MBUS AMT SAPHIR [35]</td>
</tr>
<tr>
<td></td>
<td>Heat meter</td>
<td>CALEC</td>
<td>FB_MBUS AMT CALEC [31]</td>
</tr>
<tr>
<td></td>
<td>Heat meter</td>
<td>CALEC ST, version C4</td>
<td>FB_MBUS AMT CALEC_ST C4 [33]</td>
</tr>
<tr>
<td></td>
<td>Heat meter</td>
<td>AMTRON</td>
<td>FB_MBUS AMT AMTRON [29]</td>
</tr>
<tr>
<td></td>
<td>Pulse collector</td>
<td>AMBUS</td>
<td>FB_MBUS AMT AMBUS [27]</td>
</tr>
<tr>
<td></td>
<td>Heat meter</td>
<td>AMTRON SONIC D</td>
<td>FB_MBUS_HYD_Sharky [101], FB_MBUS_HYD_Sharky_00 [103]</td>
</tr>
</tbody>
</table>
This block is used to read pulse collectors from Aquametro:

- AMBUS IS

The block can only be executed together with the block FB_MBUSKL6781() [18].

Functionality of the block [9]

VAR_INPUT

usiAddress : USINT;
stSecAddr : ST_MBUS_SecAddr;
eBaudrate : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart : BOOL;
bSND_NKE : BOOL := TRUE;
bReadInit : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiInit : USINT;
bDisabled : BOOL := FALSE;

usiAddress: Primary address [10] of the meter to be read with this block.
stSecAddr: Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAddr [205]).
eBaudrate: (see E_MBUS_baud rate [197]).
bStart: A positive edge of this input triggers one reading of the meter.
bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).
bReadInit: If the PLC is restarted, the meter is read once.
tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.
usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2=MW(h) / 3=GW(h).
bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bReady : BOOL;
bError : BOOL;
eError : E_MBUS_ERROR;
dwIdNumber : DWORD;
byStatus : BYTE;
byGEN : BYTE;
byCounter : BYTE;
usiRecivedAdr    : USINT;
eMedium          : E_MBUS_Medium;
sMan             : STRING(3);
stValue          : ST_MBus_Info;

**bBusy**: The *bBusy* output is TRUE while the meter is being read.

**bReady**: The *bReady* output is TRUE for one cycle, once meter reading is completed.

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

**eError**: In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [200]).

sMan: Manufacturer code.

stValue: Meter reading (see ST_MBus_Info [204]).

**VAR_IN_OUT**

stCom : ST_MBUS_Communication;

**stCom**: This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Communication [202]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.1.4.2 FB_MBUS_AMT_AMTRON

This block is used to read heat meters from Aquametro:

-AMTRON

The block can only be executed together with the block FB_MBUSKL6781() [18].

Functionality of the block [9]

VAR_INPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>usiAddress</td>
<td>USINT</td>
<td>Primary address [10] of the meter to be read with this block.</td>
</tr>
<tr>
<td>stSecAddr</td>
<td>ST_MBUS_SecAdr</td>
<td>Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAdr [205]).</td>
</tr>
<tr>
<td>eBaudrate</td>
<td>E_MBUS_Baudrate := eMBUS_Baud2400</td>
<td>300, 2400 baud (see E_MBUS_Baudrate [197]).</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL</td>
<td>A positive edge of this input triggers one reading of the meter.</td>
</tr>
<tr>
<td>bSND_NKE</td>
<td>BOOL := TRUE</td>
<td>TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).</td>
</tr>
<tr>
<td>bReadInit</td>
<td>BOOL := TRUE</td>
<td>If the PLC is restarted, the meter is read once.</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>TIME := t#2s;</td>
<td>Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.</td>
</tr>
<tr>
<td>usiUnit</td>
<td>USINT</td>
<td>Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL := FALSE</td>
<td>TRUE = deselection of the block.</td>
</tr>
</tbody>
</table>
VAR_OUTPUT

bBusy : BOOL;
bReady : BOOL;
bError : BOOL;
eError : E_MBUS_ERROR;
dwIdNumber : DWORD;
byStatus : BYTE;
byGEN : BYTE;
byCounter : BYTE;
usiRecievedAdr : USINT;
eMedium : E_MBUS_Medium;
sMan : STRING(3);
stEnergy : ST_MBus_Info;
stPower : ST_MBus_Info;
stVolume : ST_MBus_Info;
stFlow : ST_MBus_Info;
stForwardTemp : ST_MBus_Info;
stReturnTemp : ST_MBus_Info;
stDiffTemp : ST_MBus_Info;

bBusy: The bBusy output is TRUE while the meter is being read.

bReady: The bReady output is TRUE for one cycle, once meter reading is completed.

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

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [200]).

sMan: Manufacturer code.

stEnergy: Meter reading, energy consumption (see ST_MBus_Info [204]).

stPower: Current energy consumption, power (see ST_MBus_Info [204]).

stVolume: Meter reading, water consumption (see ST_MBus_Info [204]).

stFlow: Current water consumption (see ST_MBus_Info [204]).

stForwardTemp: Flow temperature (see ST_MBus_Info [204]).

stReturnTemp: Return temperature (see ST_MBus_Info [204]).

stDiffTemp: Temperature difference (see ST_MBus_Info [204]).

VAR_IN_OUT

stCom : ST_MBUS_Communication;

stCom: This structure is used to link the block FB_MBUSKL6781 () [202] with the meter blocks (see ST_MBUS_Communication [202]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.1.4.3 FB_MBUS_AMT_CALEC

This block is used to read heat meters from Aquametro:

- CALEC

The block can only be executed together with the block FB_MBUSKL6781() [18].

Functionality of the block [9]

VAR_INPUT

usiAddress : USINT;
stSecAddr : ST_MBUS_SecAddr;
eBaudrate : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart : BOOL;
bSND_NKE : BOOL := TRUE;
bReadInit : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit : USINT;
bDisabled : BOOL := FALSE;

usiAddress: Primary address [10] of the meter to be read with this block.

stSecAddr: Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAddr [205]).

eBaudrate: 300, 2400 baud (see E_MBUS_Baudrate [197]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.
VAR_OUTPUT

bBusy : BOOL;
bReady : BOOL;
bError : BOOL;
eError : E_MBUS_ERROR;
dwIdNumber : DWORD;
byStatus : BYTE;
byGEN : BYTE;
byCounter : BYTE;
usiRecivedAddr : USINT;
eMedium : E_MBUS_Medium;
eMan : STRING(3);
stEnergy : ST_MBus_Info;
stPower : ST_MBus_Info;
stVolume : ST_MBus_Info;
stFlow : ST_MBus_Info;
stForwardTemp : ST_MBus_Info;
stReturnTemp : ST_MBus_Info;
stDiffTemp : ST_MBus_Info;

**stEnergy**: Meter reading, energy consumption (see ST_MBus_Info [204]).

**stPower**: Current energy consumption, power (see ST_MBus_Info [204]).

**stVolume**: Meter reading, water consumption (see ST_MBus_Info [204]).

**stFlow**: Current water consumption (see ST_MBus_Info [204]).

**stForwardTemp**: Flow temperature (see ST_MBus_Info [204]).

**stReturnTemp**: Return temperature (see ST_MBus_Info [204]).

**stDiffTemp**: Temperature difference (see ST_MBus_Info [204]).

VAR_IN_OUT

stCom : ST_MBUS_Communication;

**stCom**: This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Communication [202]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>
### FB_MBUS_AMT_CALEC_STC4

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>usiAddress</td>
<td>Primary address of the meter to be read with this block.</td>
</tr>
<tr>
<td>stSecAdr</td>
<td>Secondary address of the meter to be read with this block (see ST_MBUS_SecAdr).</td>
</tr>
<tr>
<td>eBaudrate</td>
<td>E_MBUS_Baudrate := eMBUS_Baud2400;</td>
</tr>
<tr>
<td>bStart</td>
<td>A positive edge of this input triggers one reading of the meter.</td>
</tr>
<tr>
<td>bSND_NKE</td>
<td>TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).</td>
</tr>
<tr>
<td>bReadInit</td>
<td>If the PLC is restarted, the meter is read once.</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.</td>
</tr>
</tbody>
</table>

This block is used to read heat meters from Aquametro:

- CALEC ST, version C4

The block can only be executed together with the block FB_MBUSKL6781) [18].

Functionality of the block [9]
usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bReady : BOOL;
bError : BOOL;
eError : E_MBUS_ERROR;
dwIdNumber : DWORD;
byStatus : BYTE;
byGEN : BYTE;
byCounter : BYTE;
usiRecivedAdr : USINT;
eMedium : E_MBUS_Medium;
sMan : STRING(3);
stEnergyPos : ST_MBus_Info;
stVolumePos : ST_MBus_Info;
stEnergyNeg : ST_MBus_Info;
stVolumeNeg : ST_MBus_Info;
stPower : ST_MBus_Info;
stFlow : ST_MBus_Info;
stForwardTemp : ST_MBus_Info;
stReturnTemp : ST_MBus_Info;
stDiffTemp : ST_MBus_Info;
stPulsecounter1 : ST_MBus_Info;
stPulsecounter2 : ST_MBus_Info;

bBusy: The bBusy output is TRUE while the meter is being read.

bReady: The bReady output is TRUE for one cycle, once meter reading is completed.

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

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [200]).

sMan: Manufacturer code.

stEnergyPos: Counter value energy consumption (positive) (see ST_MBus_Info [204]).

stVolumePos: Counter value water consumption (positive) (see ST_MBus_Info [204]).

stEnergyNeg: Counter value energy consumption (negative) (see ST_MBus_Info [204]).

stVolumeNeg: Counter value water consumption (negative) (see ST_MBus_Info [204]).

stPower: Current energy consumption, power (see ST_MBus_Info [204]).

stFlow: Current water consumption (see ST_MBus_Info [204]).

stForwardTemp: Flow temperature (see ST_MBus_Info [204]).

stReturnTemp: Return temperature (see ST_MBus_Info [204]).

stDiffTemp: Temperature difference (see ST_MBus_Info [204]).

stPulsecounter1: Pulse counter 1 (see ST_MBus_Info [204]).
stPulsecounter2: Pulse counter 2 (see ST MBus Info [204]).

VAR_IN_OUT

stCom: ST_MBUS_Communication;

stCom: This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Communication [201]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.4.5 FB_MBUS_AMT_SAPHIR

VAR_INPUT

usiAddress : USINT;
stSecAdr : ST_MBUS_SecAdr;
eBaudrate : E_MBUS_Baudrate := EMBUS_Baud2400;
bStart : BOOL;
bSND_NKE : BOOL := TRUE;
bReadInit : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit : USINT;
bDisabled : BOOL := FALSE;

usiAddress: Primary address [10] of the meter to be read with this block.

stSecAdr: Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAdr [205]).

eBaudrate: (see E_MBUS_baud rate [197]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

This block is used to read water meters from Aquametro.

-Saphir

The block can only be executed together with the block FB_MBUSKL6781() [18].

Functionality of the block [9]
bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

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>bReady</td>
<td>BOOL;</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL;</td>
</tr>
<tr>
<td>eError</td>
<td>E_MBUS_ERROR;</td>
</tr>
<tr>
<td>dwIdNumber</td>
<td>DWORD;</td>
</tr>
<tr>
<td>byStatus</td>
<td>BYTE;</td>
</tr>
<tr>
<td>byGEN</td>
<td>BYTE;</td>
</tr>
<tr>
<td>byCounter</td>
<td>BYTE;</td>
</tr>
<tr>
<td>usiRecivedAdr</td>
<td>USINT;</td>
</tr>
<tr>
<td>eMedium</td>
<td>E_MBUS_Medium;</td>
</tr>
<tr>
<td>sMan</td>
<td>STRING(3);</td>
</tr>
<tr>
<td>stVolume</td>
<td>ST_MBus_Info;</td>
</tr>
<tr>
<td>stFlow</td>
<td>ST_MBus_Info;</td>
</tr>
</tbody>
</table>

bBusy: The bBusy output is TRUE while the meter is being read.

bReady: The bReady output is TRUE for one cycle, once meter reading is completed.

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

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [200]).

sMan: Manufacturer code.

stVolume: Meter reading, water consumption (see ST_MBus_Info [204]).

stFlow: Current water consumption (see ST_MBus_Info [204]).

**VAR_IN_OUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>stCom</td>
<td>ST_MBUS_Communication;</td>
</tr>
</tbody>
</table>

stCom: This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Communication [202]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.1.5 Berg overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [76], FB_MBUS_General_Ext [79] or FB_MBUS_General_Param [83] from the folder "General [74]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [85] can be used to send data to the device (e.g. setting of the primary address).

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Type</th>
<th>Device</th>
<th>Block</th>
</tr>
</thead>
<tbody>
<tr>
<td>Berg</td>
<td>Electricity meter</td>
<td>DZ+</td>
<td>FB_MBUS_BEC_DZ [39]</td>
</tr>
<tr>
<td></td>
<td>Electricity meter</td>
<td>DCMi</td>
<td>FB_MBUS_BEC_DCMi [37]</td>
</tr>
</tbody>
</table>

4.1.5.1 FB_MBUS_BEC_DCMi

This block is used to read electricity meters from Berg:

-DCMi

The block can only be executed together with the block FB_MBUSKL6781 [18].

Functionality of the block [9]

VAR_INPUT

\[
\begin{align*}
\text{usiAddress} & : \text{USINT}; \\
\text{stSecAdr} & : \text{ST_MBUS_SecAdr}; \\
\text{eBaudrate} & : \text{E_MBUS_Baudrate} := \text{eMBUS_Baud2400}; \\
\text{bStart} & : \text{BOOL}; \\
\text{bSND_NKE} & : \text{BOOL} := \text{TRUE}; \\
\text{bReadInit} & : \text{BOOL} := \text{TRUE}; \\
\text{tMinSendTime} & : \text{TIME} := \text{t\#2s}; \\
\text{usiUnit} & : \text{USINT}; \\
\text{bDisabled} & : \text{BOOL} := \text{FALSE};
\end{align*}
\]

\textbf{usiAddress:} Primary address [10] of the meter to be read with this block.

\textbf{stSecAdr:} Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAdr [205]).

\textbf{eBaudrate:} 300, 2400, 9600 baud (see E_MBUS_Baudrate [197]).

\textbf{bStart:} A positive edge of this input triggers one reading of the meter.
bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

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>bReady</td>
<td>BOOL;</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL;</td>
</tr>
<tr>
<td>eError</td>
<td>E_MBUS_ERROR;</td>
</tr>
<tr>
<td>dwIdNumber</td>
<td>DWORD;</td>
</tr>
<tr>
<td>byStatus</td>
<td>BYTE;</td>
</tr>
<tr>
<td>byGEN</td>
<td>BYTE;</td>
</tr>
<tr>
<td>byCounter</td>
<td>BYTE;</td>
</tr>
<tr>
<td>usiRecivedAdr</td>
<td>USINT;</td>
</tr>
<tr>
<td>eMedium</td>
<td>E_MBUS_Medium;</td>
</tr>
<tr>
<td>sMan</td>
<td>STRING(3);</td>
</tr>
<tr>
<td>stEnergy</td>
<td>ST_MBus_Info;</td>
</tr>
<tr>
<td>stPower</td>
<td>ST_MBus_Info;</td>
</tr>
<tr>
<td>stDeviceError</td>
<td>ST_MBus_Info;</td>
</tr>
</tbody>
</table>

bBusy: The bBusy output is TRUE while the meter is being read.

bReady: The bReady output is TRUE for one cycle, once meter reading is completed.

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

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [200]).

sMan: Manufacturer code.

stEnergy: Meter reading, energy consumption (see ST_MBus_Info [204]).

stPower: Current energy consumption, power (see ST_MBus_Info [204]).

stDeviceError: Error message from device (see ST_MBus_Info [204]).

VAR_IN_OUT

stCom: ST_MBUS_Communication;

stCom: This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Communication [202]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.1.5.2 FB_MBUS_BEC_DZ

This block is used to read electricity meters from Berg:

-DZ+

The block can only be executed together with the block FB_MBUSKL6781() ▶18.

Functionality of the block ▶9

VAR_INPUT

usiAddress : USINT;
stSecAdr : ST_MBUS_SecAdr;
eBaudrate : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart : BOOL;
bSND_NKE : BOOL := TRUE;
bReadInit : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit : USINT;
bDisabled : BOOL := FALSE;

usiAddress: Primary address ▶10 of the meter to be read with this block.

stSecAdr: Secondary address ▶11 of the meter to be read with this block (see ST_MBUS_SecAdr ▶205).

eBaudrate: 300, 600, 1200, 2400, 4800, 9600 baud (see E_MBUS_Baudrate ▶197).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.
VAR_OUTPUT

bBusy : BOOL;
bReady : BOOL;
bError : BOOL;
eError : E_MBUS_ERROR;
dwIdNumber : DWORD;
byStatus : BYTE;
byGEN : BYTE;
byCounter : BYTE;
usiRecivedAdr : USINT;
eMedium : E_MBUS_Medium;
sMan : STRING(3);
stActiveEnergy : ST_MBus_Info;
stReactiveEnergy : ST_MBus_Info;
stActivePower : ST_MBus_Info;
stReactivePower : ST_MBus_Info;
stCurrentL1 : ST_MBus_Info;
stCurrentL2 : ST_MBus_Info;
stCurrentL3 : ST_MBus_Info;
stPowerFactor : ST_MBus_Info;

bBusy: The bBusy output is TRUE while the meter is being read.

bReady: The bReady output is TRUE for one cycle, once meter reading is completed.

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

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [200]).

sMan: Manufacturer code.

stActiveEnergy: Meter value, total active energy (see ST_MBus_Info [204]).

stReactiveEnergy: Meter value, total reactive energy (see ST_MBus_Info [204]).

stActivePower: Current consumption value, total effective power (see ST_MBus_Info [204]).

stReactivePower: Current consumption value, total reactive power (see ST_MBus_Info [204]).

stCurrentL1: Current L1 (see ST_MBus_Info [204]).

stCurrentL2: Current L2 (see ST_MBus_Info [204]).

stCurrentL3: Current L3 (see ST_MBus_Info [204]).

stPowerFactor: Total power factor (see ST_MBus_Info [204]).

VAR_IN_OUT

stCom : ST_MBUS_Communication;

stCom: This structure is used to link the block FB_MBUSKL67810 [202] with the meter blocks (see ST_MBUS_Communication [202]).
Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.6 Brunata overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks **FB_MBUS_General** [76], **FB_MBUS_General_Ext** [79] or **FB_MBUS_General_Param** [83] from the folder "General" [74] should be used. Note that these blocks do not run BC and BX systems. The block **FB_MBUS_General_Send()** [85] can be used to send data to the device (e.g. setting of the primary address).

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Type</th>
<th>Device</th>
<th>Block</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brunata</td>
<td>Heat meter</td>
<td>HGQ / HGS</td>
<td><strong>FB_MBUS_BHG_HGx</strong> [41]</td>
</tr>
<tr>
<td></td>
<td>Heat meter</td>
<td>Optuna H (775)</td>
<td><strong>FB_MBUS_HYD_Sharky</strong> [101], <strong>FB_MBUS_HYD_Sharky_00</strong> [103]</td>
</tr>
</tbody>
</table>

### 4.1.6.1 FB_MBUS_BHG_HGx

This block is used to read heat meters from Brunata:

- HGQ
- HGS

The block can only be executed together with the block **FB_MBUSKL6781()** [18].
Functionality of the block [9]

**VAR_INPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>usiAddress</td>
<td>USINT;</td>
<td>Primary address [10] of the meter to be read with this block.</td>
</tr>
<tr>
<td>stSecAdr</td>
<td>ST_MBUS_SecAdr;</td>
<td>Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAdr [205]).</td>
</tr>
<tr>
<td>eBaudrate</td>
<td>E_MBUS_Baudrate := eMBUS_Baud2400;</td>
<td>(see E_MBUS_baud_rate [197]).</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL;</td>
<td>A positive edge of this input triggers one reading of the meter.</td>
</tr>
<tr>
<td>bSND_NKE</td>
<td>BOOL := TRUE;</td>
<td>TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).</td>
</tr>
<tr>
<td>bReadInit</td>
<td>BOOL := TRUE;</td>
<td>If the PLC is restarted, the meter is read once.</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>TIME := t#2s;</td>
<td>Standard t#2s. The meter is read again, once the time set here has elapsed.</td>
</tr>
<tr>
<td>usiUnit</td>
<td>USINT;</td>
<td>Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).</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 while the meter is being read.</td>
</tr>
<tr>
<td>bReady</td>
<td>BOOL;</td>
<td>The bReady output is TRUE for one cycle, once meter reading is completed.</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_MBUS_ERROR;</td>
<td>In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.</td>
</tr>
<tr>
<td>dwIdNumber</td>
<td>DWORD;</td>
<td>Serial number of the meter (secondary address).</td>
</tr>
<tr>
<td>byStatus</td>
<td>BYTE;</td>
<td>Device status.</td>
</tr>
<tr>
<td>byGEN</td>
<td>BYTE;</td>
<td>Software version of the device.</td>
</tr>
<tr>
<td>usiRecivedAdr</td>
<td>USINT;</td>
<td></td>
</tr>
<tr>
<td>eMedium</td>
<td>E_MBUS_Medium;</td>
<td></td>
</tr>
<tr>
<td>sMan</td>
<td>STRING(3);</td>
<td></td>
</tr>
<tr>
<td>stEnergy</td>
<td>ST_MBus_Info;</td>
<td></td>
</tr>
<tr>
<td>stVolume</td>
<td>ST_MBus_Info;</td>
<td></td>
</tr>
<tr>
<td>stVolume2</td>
<td>ST_MBus_Info;</td>
<td></td>
</tr>
<tr>
<td>stForwardTemp</td>
<td>ST_MBus_Info;</td>
<td></td>
</tr>
<tr>
<td>stReturnTemp</td>
<td>ST_MBus_Info;</td>
<td></td>
</tr>
<tr>
<td>stDiffTemp</td>
<td>ST_MBus_Info;</td>
<td></td>
</tr>
<tr>
<td>stFlow</td>
<td>ST_MBus_Info;</td>
<td></td>
</tr>
<tr>
<td>stPower</td>
<td>ST_MBus_Info;</td>
<td></td>
</tr>
<tr>
<td>stPulsecounter1</td>
<td>ST_MBus_Info;</td>
<td></td>
</tr>
<tr>
<td>stPulsecounter2</td>
<td>ST_MBus_Info;</td>
<td></td>
</tr>
</tbody>
</table>

**Notes**

- **usiAddress**: Primary address [10] of the meter to be read with this block.
- **stSecAdr**: Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAdr [205]).
- **eBaudrate**: (see E_MBUS_baud_rate [197]).
- **bStart**: A positive edge of this input triggers one reading of the meter.
- **bSND_NKE**: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).
- **bReadInit**: If the PLC is restarted, the meter is read once.
- **tMinSendTime**: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.
- **usiUnit**: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).
- **bDisabled**: TRUE = deselection of the block.
**byCounter**: Number of times the master has accessed data of the respective slave.

**usiRecivedAdr**: Received primary address (0-250).

**eMedium**: Medium (see E_MBUS_Medium [200]).

**sMan**: Manufacturer code.

**stEnergy**: Meter reading, energy consumption (see ST_MBus_Info [204]).

**stVolume**: Meter reading, water consumption (see ST_MBus_Info [204]).

**stVolume2**: Volume from flow sensor (see ST_MBus_Info [204]).

**stForwardTemp**: Flow temperature (see ST_MBus_Info [204]).

**stReturnTemp**: Return temperature (see ST_MBus_Info [204]).

**stDiffTemp**: Temperature difference (see ST_MBus_Info [204]).

**stFlow**: Current water consumption (see ST_MBus_Info [204]).

**stPower**: Current energy consumption, power (see ST_MBus_Info [204]).

**stPulsecounter1**: Pulse counter 1 (see ST_MBus_Info [204]).

**stPulsecounter2**: Pulse counter 2 (see ST_MBus_Info [204]).

**VAR_IN_OUT**

```
VAR_IN_OUT

stCom : ST_MBUS_Communication;
```

**stCom**: This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Communication [202]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.7 Carlo Gavazzi overview**

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [76], FB_MBUS_General Ext [79] or FB_MBUS_General Param [83] from the folder "General [74]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Sendi [85] can be used to send data to the device (e.g. setting of the primary address).

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Type</th>
<th>Device</th>
<th>Block</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carlo Gavazzi</td>
<td>Energy meter</td>
<td>EM24</td>
<td>FB_MBUS_GAV EM24</td>
</tr>
</tbody>
</table>

[200] [202] [76] [79] [83] [74] [85]
4.1.7.1 **FB_MBUS_GAV_EM24**

This block is used to read energy meters from Carl Gavazzi.

The block can only be executed together with the block **FB_MBUSKL6781**.

**Functionality of the block**

**VAR_INPUT**

- **usiAddress**: Primary address of the meter to be read with this block.
- **stSecAdr**: Secondary address of the meter to be read with this block (see **ST_MBUS_SecAdr**).
- **eBaudrate**: 300, 2400, 9600 baud (see **E_MBUS_Baudrate**).
- **bStart**: A positive edge of this input triggers one reading of the meter.
- **bSND_NKE**: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).
- **bReadInit**: If the PLC is restarted, the meter is read once.
- **tMinSendTime**: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with **bStart**.
- **usiUnit**: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).
- **bDisabled**: TRUE = deselection of the block.

**VAR_OUTPUT**

- **bBusy**: BOOLEAN;
- **bReady**: BOOLEAN;
- **bError**: BOOLEAN;
- **eError**: E_MBUS_ERROR;
- **dwIdNumber**: DWORD;
- **byStatus**: BYTE;
VAR_IN_OUT

stCom : ST_MBUS_Communication;

VAR_IN_OUT

stCom: This structure is used to link the block FB_MBUSKL6781() with the meter blocks (see ST_MBUS_Communication).

bBusy: The bBusy output is TRUE while the meter is being read.

bReady: The bReady output is TRUE for one cycle, once meter reading is completed.

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

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR). bError goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium).

sMan: Manufacturer code.

stKWh_TOT: See manufacturer information (see ST_MBus_Info).

stKWh_L1: See manufacturer information (see ST_MBus_Info).

stKWh_L2: See manufacturer information (see ST_MBus_Info).

stKWh_L3: See manufacturer information (see ST_MBus_Info).

stW_Sum: See manufacturer information (see ST_MBus_Info).

stV_L_N_Sum: See manufacturer information (see ST_MBus_Info).

stV_L_L_Sum: See manufacturer information (see ST_MBus_Info).

stKWh_TOT: See manufacturer information (see ST_MBus_Info).

stKWh_L1: See manufacturer information (see ST_MBus_Info).

stKWh_L2: See manufacturer information (see ST_MBus_Info).

stKWh_L3: See manufacturer information (see ST_MBus_Info).

stW_Sum: See manufacturer information (see ST_MBus_Info).

stV_L_N_Sum: See manufacturer information (see ST_MBus_Info).

stV_L_L_Sum: See manufacturer information (see ST_MBus_Info).

VAR_IN_OUT

stCom : ST_MBUS_Communication;
4.1.8 Cynox

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [76], FB_MBUS_General_Ext [79] or FB_MBUS_General_Param [83] from the folder "General [74]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [85] can be used to send data to the device (e.g. setting of the primary address).

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Type</th>
<th>Device</th>
<th>Function block</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cynox</td>
<td>Pulse counter</td>
<td>MCount2C</td>
<td>FB_MBUS_CYN_MCount2C [46]</td>
</tr>
</tbody>
</table>

### 4.1.8.1 FB_MBUS_CYN_MCount2C

![FB_MBUS_CYN_MCount2C Diagram]

This function block is used to read pulse counters from Cynox.

The function block can only be executed together with the function block FB_MBUSKL6781() [18].

**Functionality of the function block [9]**

**VAR_INPUT**

- **usiAddress**: Primary address [10] of the meter to be read with this block.
- **stSecAdr**: Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAdr [205]).

```plaintext
usiAddress : USINT;
stSecAdr   : ST_MBUS_SecAdr;
eBaudrate  : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart     : BOOL;
bSND_NKE   : BOOL := TRUE;
bReadInit  : BOOL := TRUE;
tMinSendTime: TIME := t#2s;
bDisabled  : BOOL := FALSE;
```
eBaudrate: 2400 baud (see E_MBus_Baudrate [197]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.

bDisabled: TRUE = deselection of the function block.

**VAR_OUTPUT**

```plaintext
VAR_OUTPUT
  bBusy : BOOL;
  bReady : BOOL;
  bError : BOOL;
  eError : E_MBUS_ERROR;
  dwIdNumber : DWORD;
  byStatus   : BYTE;
  byGEN      : BYTE;
  byCounter  : BYTE;
  usiRecivedAdr : USINT;
  eMedium   : E_MBUS_Medium;
  stMan     : STRING(3);
  stCurrent1 : ST_MBus_Info;
  stCurrent2 : ST_MBus_Info;
  stCurrent3 : ST_MBus_Info;
  stCurrent4 : ST_MBus_Info;
  stHistorical1 : ST_MBus_Info;
  stHistorical2 : ST_MBus_Info;
  stHistorical3 : ST_MBus_Info;
  stHistorical4 : ST_MBus_Info;
  stNextDeadline : ST_MBus_Info;
  stLastDeadline : ST_MBus_Info;
  stCurrentTime : ST_MBus_Info;
  stOperatingTime : ST_MBus_Info;
```

bBusy: The bBusy output is TRUE while the meter is being read.

bReady: The bReady output is TRUE for one cycle, once meter reading is completed.

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

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [200]).

sMan: Manufacturer code.

stCurrent1: Current value 1 (see ST_MBus_Info [204]).

stCurrent2: Current value 2 (see ST_MBus_Info [204]).

stCurrent3: Current value 3 (see ST_MBus_Info [204]).

stCurrent4: Current value 4 (see ST_MBus_Info [204]).

stHistorical1: Historical value 1 (see ST_MBus_Info [204]).
stHistorical2: Historical value 2 (see ST_MBUs_Info [204]).

stHistorical3: Historical value 3 (see ST_MBUs_Info [204]).

stHistorical4: Historical value 4 (see ST_MBUs_Info [204]).

stNextDeadline: Next reporting date (see ST_MBUs_Info [204]).

stLastDeadline: Last reporting date (see ST_MBUs_Info [204]).

stCurrentTime: Current time (see ST_MBUs_Info [204]).

stOperatingTime: Operating time (see ST_MBUs_Info [204]).

VAR_IN_OUT

stCom : ST_MBUS_Communication;

stCom: This structure is used to link the block FB_MBUSKL6781 [202] with the meter blocks (see ST_MBUS_Communication [202]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.32</td>
<td>Tc2_MBus from 3.4.6.0</td>
</tr>
</tbody>
</table>

4.1.9 Elster overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT”. If more or all data are required, the blocks FB_MBUS_General [76], FB_MBUS_General_Ext [79] or FB_MBUS_General_Param [83] from the folder "General [74]” should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [85] can be used to send data to the device (e.g. setting of the primary address).

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Type</th>
<th>Device</th>
<th>Block</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elster</td>
<td>Gas meter</td>
<td>Encoder Z6</td>
<td>FB_MBUS_ELS_EncoderZ6 [48]</td>
</tr>
</tbody>
</table>

4.1.9.1 FB_MBUS_ELS_EncoderZ6

This block is used to read meters from Elster:
The block can only be executed together with the block FB_MBUSKL6781 [18].

Functionality of the block [9]

**VAR_INPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>usiAddress</td>
<td>Primary address [10] of the meter to be read with this block.</td>
</tr>
<tr>
<td>stSecAdr</td>
<td>Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAdr [205]).</td>
</tr>
<tr>
<td>eBaudrate</td>
<td>300, 2400 baud (see E_MBUS_Baudrate [197]).</td>
</tr>
<tr>
<td>bStart</td>
<td>A positive edge of this input triggers one reading of the meter.</td>
</tr>
<tr>
<td>bSND_NKE</td>
<td>TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).</td>
</tr>
<tr>
<td>bReadInit</td>
<td>If the PLC is restarted, the meter is read once.</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>Standard t#2s. The meter is read again, once the time set here has elapsed.</td>
</tr>
<tr>
<td>usiUnit</td>
<td>Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).</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 while the meter is being read.</td>
</tr>
<tr>
<td>bReady</td>
<td>The bReady output is TRUE for one cycle, once meter reading is completed.</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>In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.</td>
</tr>
<tr>
<td>dwIdNumber</td>
<td>Serial number of the meter (secondary address).</td>
</tr>
<tr>
<td>byStatus</td>
<td>Device status.</td>
</tr>
<tr>
<td>byGEN</td>
<td>Software version of the device.</td>
</tr>
<tr>
<td>byCounter</td>
<td>Number of times the master has accessed data of the respective slave.</td>
</tr>
<tr>
<td>usiRecivedAdr</td>
<td>Received primary address (0-250).</td>
</tr>
</tbody>
</table>

**Notes:**

- Encoder Z6

- Encoder Z6

The block can only be executed together with the block FB_MBUSKL6781 [18].

Functionality of the block [9]

**VAR_INPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>usiAddress</td>
<td>USINT;</td>
</tr>
<tr>
<td>stSecAdr</td>
<td>ST_MBUS_SecAdr;</td>
</tr>
<tr>
<td>eBaudrate</td>
<td>E_MBUS_Baudrate := eMBUS_Baud2400;</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL;</td>
</tr>
<tr>
<td>bSND_NKE</td>
<td>BOOL := TRUE;</td>
</tr>
<tr>
<td>bReadInit</td>
<td>BOOL := TRUE;</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>TIME := t#2s;</td>
</tr>
<tr>
<td>usiUnit</td>
<td>USINT;</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL := FALSE;</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>bReady</td>
<td>BOOL;</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL;</td>
</tr>
<tr>
<td>eError</td>
<td>E_MBUS_ERROR;</td>
</tr>
<tr>
<td>dwIdNumber</td>
<td>DWORD;</td>
</tr>
<tr>
<td>byStatus</td>
<td>BYTE;</td>
</tr>
<tr>
<td>byGEN</td>
<td>BYTE;</td>
</tr>
<tr>
<td>byCounter</td>
<td>BYTE;</td>
</tr>
<tr>
<td>usiRecivedAdr</td>
<td>USINT;</td>
</tr>
<tr>
<td>eMedium</td>
<td>E_MBUS_Medium;</td>
</tr>
<tr>
<td>sMan</td>
<td>STRING(3);</td>
</tr>
<tr>
<td>stVolume</td>
<td>ST_MBus_Info;</td>
</tr>
</tbody>
</table>

**Notes:**

- Encoder Z6

The block can only be executed together with the block FB_MBUSKL6781 [18].

Functionality of the block [9]

**VAR_INPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>usiAddress</td>
<td>USINT;</td>
</tr>
<tr>
<td>stSecAdr</td>
<td>ST_MBUS_SecAdr;</td>
</tr>
<tr>
<td>eBaudrate</td>
<td>E_MBUS_Baudrate := eMBUS_Baud2400;</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL;</td>
</tr>
<tr>
<td>bSND_NKE</td>
<td>BOOL := TRUE;</td>
</tr>
<tr>
<td>bReadInit</td>
<td>NULL := TRUE;</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>TIME := t#2s;</td>
</tr>
<tr>
<td>usiUnit</td>
<td>USINT;</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL := FALSE;</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>bReady</td>
<td>BOOL;</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL;</td>
</tr>
<tr>
<td>eError</td>
<td>E_MBUS_ERROR;</td>
</tr>
<tr>
<td>dwIdNumber</td>
<td>DWORD;</td>
</tr>
<tr>
<td>byStatus</td>
<td>BYTE;</td>
</tr>
<tr>
<td>byGEN</td>
<td>BYTE;</td>
</tr>
<tr>
<td>byCounter</td>
<td>BYTE;</td>
</tr>
<tr>
<td>usiRecivedAdr</td>
<td>USINT;</td>
</tr>
<tr>
<td>eMedium</td>
<td>E_MBUS_Medium;</td>
</tr>
<tr>
<td>sMan</td>
<td>STRING(3);</td>
</tr>
<tr>
<td>stVolume</td>
<td>ST_MBus_Info;</td>
</tr>
</tbody>
</table>

**Notes:**

- Encoder Z6

The block can only be executed together with the block FB_MBUSKL6781 [18].

Functionality of the block [9]

**VAR_INPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>usiAddress</td>
<td>USINT;</td>
</tr>
<tr>
<td>stSecAdr</td>
<td>ST_MBUS_SecAdr;</td>
</tr>
<tr>
<td>eBaudrate</td>
<td>E_MBUS_Baudrate := eMBUS_Baud2400;</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL;</td>
</tr>
<tr>
<td>bSND_NKE</td>
<td>BOOL := TRUE;</td>
</tr>
<tr>
<td>bReadInit</td>
<td>NULL := TRUE;</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>TIME := t#2s;</td>
</tr>
<tr>
<td>usiUnit</td>
<td>USINT;</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL := FALSE;</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>bReady</td>
<td>BOOL;</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL;</td>
</tr>
<tr>
<td>eError</td>
<td>E_MBUS_ERROR;</td>
</tr>
<tr>
<td>dwIdNumber</td>
<td>DWORD;</td>
</tr>
<tr>
<td>byStatus</td>
<td>BYTE;</td>
</tr>
<tr>
<td>byGEN</td>
<td>BYTE;</td>
</tr>
<tr>
<td>byCounter</td>
<td>BYTE;</td>
</tr>
<tr>
<td>usiRecivedAdr</td>
<td>USINT;</td>
</tr>
<tr>
<td>eMedium</td>
<td>E_MBUS_Medium;</td>
</tr>
<tr>
<td>sMan</td>
<td>STRING(3);</td>
</tr>
<tr>
<td>stVolume</td>
<td>ST_MBus_Info;</td>
</tr>
</tbody>
</table>
eMedium: Medium (see E_MBUS_Medium [200]).

sMan: Manufacturer code.

stVolume: Meter reading, water consumption (see ST_MBus_Info [204]).

**VAR_IN_OUT**

stCom : ST_MBUS_Communication;

stCom: This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Communication [202]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.10 elvaco overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [76], FB_MBUS_General_Ext [79] or FB_MBUS_General_Param [83] from the folder "General [74]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [85] can be used to send data to the device (e.g. setting of the primary address).

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Type</th>
<th>Device</th>
<th>Block</th>
</tr>
</thead>
<tbody>
<tr>
<td>elvaco</td>
<td>Temperature and humidity sensors</td>
<td>CMa10 / CMa20</td>
<td>FB_MBUS ELV_CMa10_20 [50]</td>
</tr>
</tbody>
</table>

### 4.1.10.1 FB_MBUS_ELV_CMa10_20

This block is used to read temperature and humidity sensors from elvaco.

Can be used with the sensors CMa10 and CMa20.

The block can only be executed together with the block FB_MBUSKL6781() [18].
Functionality of the block [► 9]

VAR_INPUT

usiAddress : USINT;
stSecAdr : ST_MBUS_SecAdr;
eBaudrate : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart : BOOL;
bSND_NKE : BOOL := TRUE;
bReadInit : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit : USINT;
bDisabled : BOOL := FALSE;

usiAddress: Primary address [► 10] of the meter to be read with this block.

stSecAdr: Secondary address [► 11] of the meter to be read with this block (see ST_MBUS_SecAdr [► 205]).

eBaudrate: 300, 2400, 9600 baud (see E_MBUS_Baudrate [► 197]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bReady : BOOL;
bError : BOOL;
eError : E_MBUS_ERROR;
dwIdNumber : DWORD;
byStatus : BYTE;
byGEN : BYTE;
byCounter : BYTE;
usiRecivedAdr : USINT;
eMedium : E_MBUS_Medium;
strRelHumi : STRING(3);
strRelHumiMin : ST_MBus_Info;
strRelHumiMax : ST_MBus_Info;
strTemp : ST_MBus_Info;
strTempMin : ST_MBus_Info;
strTempMax : ST_MBus_Info;
strTempAvg1h : ST_MBus_Info;
strTempAvg24h : ST_MBus_Info;

bBusy: The bBusy output is TRUE while the meter is being read.

bReady: The bReady output is TRUE for one cycle, once meter reading is completed.

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

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [► 197]). bError goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.
usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [201]).

sMan: Manufacturer code.

strRelHumi: Current relative humidity (see ST_MBus_Info [204]).

strRelHumiMin: Lowest relative humidity since the last min/max reset command (see ST_MBus_Info [204]).

strRelHumiMax: Highest relative humidity since the last min/max reset command (see ST_MBus_Info [204]).

strTemp: Current temperature (see ST_MBus_Info [204]).

strTempMin: Lowest temperature since the last min/max reset command (see ST_MBus_Info [204]).

strTempMax: Highest temperature since the last min/max reset command (see ST_MBus_Info [204]).

strTempAvg1h: 1-hour mean value for temperature (see ST_MBus_Info [204]).

strTempAvg24h: 24-hour mean value for temperature (see ST_MBus_Info [204]).

VAR_IN_OUT

stCom : ST_MBUS_Communication;

stCom: This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Communication [201]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.11 EMH overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [76], FB_MBUS_General_Ext [79] or FB_MBUS_General_Param [83] from the folder "General [74]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [85] can be used to send data to the device (e.g. setting of the primary address).

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Type</th>
<th>Device</th>
<th>Block</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMH</td>
<td>Electricity meter</td>
<td>DIZ</td>
<td>FB_MBUS_EMH_DIZ [53]</td>
</tr>
<tr>
<td></td>
<td>Electricity meter</td>
<td>EIZ-E</td>
<td>FB_MBUS_EMH_EIZE [55]</td>
</tr>
<tr>
<td></td>
<td>Electricity meter</td>
<td>EIZ-G</td>
<td>FB_MBUS_EMH_EIZG [57]</td>
</tr>
<tr>
<td></td>
<td>Electricity meter</td>
<td>MIZ</td>
<td>FB_MBUS_EMH_MIZ [59]</td>
</tr>
</tbody>
</table>
4.1.11.1   FB_MBUS_EMH_DIZ

This block is used to read electricity meters from EMH:

-DIZ

Unidirectional tariff meter only

The block can only be executed together with the block FB_MBUSKL6781() [18].

Functionality of the block [9]

VAR_INPUT

usiAddress : USINT;
stSecAdr : ST_MBUS_SecAdr;
eBaudrate : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart : BOOL;
bSND_NKE : BOOL := TRUE;
bReadInit : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit : USINT;
bDisabled : BOOL := FALSE;

usiAddress: Primary address [10] of the meter to be read with this block.

stSecAdr: Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAdr [205]).

eBaudrate: 300, 2400, 9600 baud (see E_MBUS_Baudrate [197]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bReady : BOOL;
bError : BOOL;

bBusy:
### Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>eError</td>
<td>E_MBUS_ERROR</td>
<td>Error code</td>
</tr>
<tr>
<td>dwIdNumber</td>
<td>DWORD</td>
<td>Serial number of the meter (secondary address)</td>
</tr>
<tr>
<td>byStatus</td>
<td>BYTE</td>
<td>Device status</td>
</tr>
<tr>
<td>byGEN</td>
<td>BYTE</td>
<td>Software version of the device</td>
</tr>
<tr>
<td>byCounter</td>
<td>BYTE</td>
<td>Number of times the master has accessed data of the respective slave</td>
</tr>
<tr>
<td>usiRecivedAdr</td>
<td>USINT</td>
<td>Received primary address (0-250)</td>
</tr>
<tr>
<td>eMedium</td>
<td>E_MBUS_Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>sMan</td>
<td>STRING(3)</td>
<td>Manufacturer code</td>
</tr>
<tr>
<td>stEnergy</td>
<td>ST_MBus_Info</td>
<td>Meter reading, energy consumption</td>
</tr>
<tr>
<td>stPower</td>
<td>ST_MBus_Info</td>
<td>Current energy consumption, power</td>
</tr>
<tr>
<td>stDeviceError</td>
<td>ST_MBus_Info</td>
<td>Error message from device</td>
</tr>
</tbody>
</table>

### Notes

- **bBusy**: The `bBusy` output is TRUE while the meter is being read.
- **bReady**: The `bReady` output is TRUE for one cycle, once meter reading is completed.
- **bError**: This output goes TRUE as soon as an error occurs. This error is described via the variable `eError`.
- **eError**: In the event of an error the output issues an error code (see E_MBUS_ERROR). `bError` goes TRUE at the same time.
- **dwIdNumber**: Serial number of the meter (secondary address).
- **byStatus**: Device status.
- **byGEN**: Software version of the device.
- **byCounter**: Number of times the master has accessed data of the respective slave.
- **usiRecivedAdr**: Received primary address (0-250).
- **eMedium**: Medium (see E_MBUS_Medium).
- **sMan**: Manufacturer code.
- **stEnergy**: Meter reading, energy consumption (see ST_MBus_Info).
- **stPower**: Current energy consumption, power (see ST_MBus_Info).
- **stDeviceError**: Error message from device (see ST_MBus_Info).

### VAR_IN_OUT

- **stCom**: This structure is used to link the block FB_MBUSKL6781() with the meter blocks (see ST_MBUS_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_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.1.11.2 FB_MBUS_EMH_EIZE

This block is used to read electricity meters from EMH:

-EIZ-E

The block can only be executed together with the block FB_MBUSKL6781() [18].

Functionality of the block [9]

VAR_INPUT

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>usiAddress</td>
<td>Primary address [10] of the meter to be read with this block.</td>
</tr>
<tr>
<td>stSecAdr</td>
<td>Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAdr [205]).</td>
</tr>
<tr>
<td>eBaudrate</td>
<td>300, 2400, 9600 baud (see E_MBUS_Baudrate [197]).</td>
</tr>
<tr>
<td>bStart</td>
<td>A positive edge of this input triggers one reading of the meter.</td>
</tr>
<tr>
<td>bSND_NKE</td>
<td>TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).</td>
</tr>
<tr>
<td>bReadInit</td>
<td>If the PLC is restarted, the meter is read once.</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.</td>
</tr>
<tr>
<td>usuUnit</td>
<td>Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).</td>
</tr>
<tr>
<td>bDisabled</td>
<td>TRUE = deselection of the block.</td>
</tr>
</tbody>
</table>

VAR_OUTPUT

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td></td>
</tr>
<tr>
<td>bReady</td>
<td></td>
</tr>
<tr>
<td>bError</td>
<td></td>
</tr>
<tr>
<td>eError</td>
<td></td>
</tr>
<tr>
<td>dwIdNumber</td>
<td></td>
</tr>
</tbody>
</table>
byStatus : BYTE;
byGEN : BYTE;
byCounter : BYTE;
usiRecivedAdr : USINT;
eMedium : E_MBUS_Medium;
sMan : STRING(3);
stEnergy : ST_MBus_Info;
stPower : ST_MBus_Info;
stDeviceError : ST_MBus_Info;

**bBusy:** The *bBusy* output is TRUE while the meter is being read.

**bReady:** The *bReady* output is TRUE for one cycle, once meter reading is completed.

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

**eError:** In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). *bError* goes TRUE at the same time.

**dwIdNumber:** Serial number of the meter (secondary address).

**byStatus:** Device status.

**byGEN:** Software version of the device.

**byCounter:** Number of times the master has accessed data of the respective slave.

**usiRecivedAdr:** Received primary address (0-250).

**eMedium:** Medium (see E_MBUS_Medium [200]).

**sMan:** Manufacturer code.

**stEnergy:** Meter reading, energy consumption (see ST_MBus_Info [204]).

**stPower:** Current energy consumption, power (see ST_MBus_Info [204]).

**stDeviceError:** Error message from device (see ST_MBus_Info [204]).

**VAR_IN_OUT**

**stCom :** ST_MBUS_Communication;

**stCom:** This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Communication [202]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>
### 4.1.11.3 FB_MBUS_EMH_EIZG

This block is used to read electricity meters from EMH:

-EIZ-G

The block can only be executed together with the block FB_MBUSKL6781) [18].

**Functionality of the block [19]**

#### VAR_INPUT

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>usiAddress</td>
<td>Primary address [10] of the meter to be read with this block.</td>
<td></td>
</tr>
<tr>
<td>stSecAdr</td>
<td>Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAdr [205]).</td>
<td></td>
</tr>
<tr>
<td>eBaudrate</td>
<td>300, 2400, 9600 baud (see E_MBUS_Baudrate [197]).</td>
<td></td>
</tr>
<tr>
<td>bStart</td>
<td>A positive edge of this input triggers one reading of the meter.</td>
<td></td>
</tr>
<tr>
<td>bSND_NKE</td>
<td>TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).</td>
<td></td>
</tr>
<tr>
<td>bReadInit</td>
<td>If the PLC is restarted, the meter is read once.</td>
<td></td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.</td>
<td></td>
</tr>
<tr>
<td>usiUnit</td>
<td>Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).</td>
<td></td>
</tr>
<tr>
<td>bDisabled</td>
<td>TRUE = deselection of the block.</td>
<td></td>
</tr>
</tbody>
</table>

#### VAR_OUTPUT

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td></td>
<td>BOOL;</td>
</tr>
<tr>
<td>bReady</td>
<td></td>
<td>BOOL;</td>
</tr>
<tr>
<td>bError</td>
<td></td>
<td>BOOL;</td>
</tr>
<tr>
<td>eError</td>
<td></td>
<td>E_MBUS_ERROR;</td>
</tr>
<tr>
<td>dwIdNumber</td>
<td></td>
<td>DWORD;</td>
</tr>
</tbody>
</table>

---

This block is used to read electricity meters from EMH:

-EIZ-G

The block can only be executed together with the block FB_MBUSKL6781) [18].

**Functionality of the block [19]**

#### VAR_INPUT

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>usiAddress</td>
<td>Primary address [10] of the meter to be read with this block.</td>
<td></td>
</tr>
<tr>
<td>stSecAdr</td>
<td>Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAdr [205]).</td>
<td></td>
</tr>
<tr>
<td>eBaudrate</td>
<td>300, 2400, 9600 baud (see E_MBUS_Baudrate [197]).</td>
<td></td>
</tr>
<tr>
<td>bStart</td>
<td>A positive edge of this input triggers one reading of the meter.</td>
<td></td>
</tr>
<tr>
<td>bSND_NKE</td>
<td>TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).</td>
<td></td>
</tr>
<tr>
<td>bReadInit</td>
<td>If the PLC is restarted, the meter is read once.</td>
<td></td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.</td>
<td></td>
</tr>
<tr>
<td>usiUnit</td>
<td>Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).</td>
<td></td>
</tr>
<tr>
<td>bDisabled</td>
<td>TRUE = deselection of the block.</td>
<td></td>
</tr>
</tbody>
</table>

#### VAR_OUTPUT

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td></td>
<td>BOOL;</td>
</tr>
<tr>
<td>bReady</td>
<td></td>
<td>BOOL;</td>
</tr>
<tr>
<td>bError</td>
<td></td>
<td>BOOL;</td>
</tr>
<tr>
<td>eError</td>
<td></td>
<td>E_MBUS_ERROR;</td>
</tr>
<tr>
<td>dwIdNumber</td>
<td></td>
<td>DWORD;</td>
</tr>
</tbody>
</table>
byStatus : BYTE;
byGEN : BYTE;
byCounter : BYTE;
usiRecivedAdr : USINT;
eMedium : E_MBUS_Medium;
sMan : STRING(3);
stEnergy : ST_MBus_Info;
stPower : ST_MBus_Info;
stDeviceError : ST_MBus_Info;

bBusy: The bBusy output is TRUE while the meter is being read.

bReady: The bReady output is TRUE for one cycle, once meter reading is completed.

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

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [200]).

sMan: Manufacturer code.

stEnergy: Meter reading, energy consumption (see ST_MBus_Info [204]).

stPower: Current energy consumption, power (see ST_MBus_Info [204]).

stDeviceError: Error message from device (see ST_MBus_Info [204]).

VAR_IN_OUT
stCom : ST_MBUS_Communication;

stCom: This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Communication [202]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.1.11.4   FB_MBUS_EMH_MIZ

This block is used to read electricity meters from EMH:

-MIZ

The block can only be executed together with the block FB_MBUSKL6781() \[\ref{FB_MBUSKL6781}\].

Functionality of the block \[\ref{FB_MBUS_EMH_MIZ}\]

VAR_INPUT

\begin{verbatim}
usiAddress   : USINT;
stSecadr     : ST_MBUS_Secadr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit      : USINT;
bDisabled    : BOOL := FALSE;
\end{verbatim}

usiAddress: Primary address \[\ref{usiAddress}\] of the meter to be read with this block.

stSecadr: Secondary address \[\ref{stSecadr}\] of the meter to be read with this block (see ST_MBUS_Secadr \[\ref{ST_MBUS_Secadr}\]).

eBaudrate: 300, 2400, 9600 baud (see E_MBUS_Baudrate \[\ref{E_MBUS_Baudrate}\]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.
VAR_OUTPUT

bBusy : BOOL;
bReady : BOOL;
bError : BOOL;
eError : E_MBUS_ERROR;
dwIdNumber : DWORD;
byStatus : BYTE;
byGEN : BYTE;
byCounter : BYTE;
usiRecivedAddr : USINT;
eMedium : E_MBUS_Medium;
sm : STRING(3);
stEnergy : ST_MBus_Info;
stPower : ST_MBus_Info;
stVoltage : ST_MBus_Info;
stCurrent : ST_MBus_Info;
stFrequency : ST_MBus_Info;
stPowerfactor : ST_MBus_Info;
stDeviceError : ST_MBus_Info;

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

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

eError: In the event of an error the output issues an error code (see `E_MBUS_ERROR` [197]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAddr: Received primary address (0-250).

eMedium: Medium (see `E_MBUS_Medium` [200]).

sm: Manufacturer code.

stEnergy: Meter reading, energy consumption (see `ST_MBus_Info` [204]).

stPower: Current energy consumption, power (see `ST_MBus_Info` [204]).

stVoltage: Current voltage (see `ST_MBus_Info` [204]).

stCurrent: Current current (see `ST_MBus_Info` [204]).

stFrequency: Current frequency (see `ST_MBus_Info` [204]).

stPowerfactor: Power factor (see `ST_MBus_Info` [204]).

stDeviceError: Error message from device (see `ST_MBus_Info` [204]).

VAR_IN_OUT

stCom : ST_MBUS_Communication;

stCom: This structure is used to link the block `FB_MBUSKL6781()` [202] with the meter blocks (see `ST_MBUS_Communication` [202]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 function</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT v3.1</td>
<td>TF8000</td>
<td>TC3 BA Connectivity Library from v1.0.0.0</td>
</tr>
</tbody>
</table>
4.1.12  EMU overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [76], FB_MBUS_General_Ext [79] or FB_MBUS_General_Param [83] from the folder "General [74]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [85] can be used to send data to the device (e.g. setting of the primary address).

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Type</th>
<th>Device</th>
<th>Block</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMU</td>
<td>Electricity</td>
<td>EMU32x7</td>
<td>FB_MBUS_EMU_32x7 [61]</td>
</tr>
<tr>
<td></td>
<td>meter</td>
<td>EMU32x7</td>
<td>FB_MBUS EMU 32x7 Opti</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>on8 [64]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Allrounder 3/5</td>
<td>FB_MBUS EMU 3 5 Allrounder [67]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DHZ 5/63</td>
<td>FB_MBUS EMU DHZ 5 63 [69]</td>
</tr>
</tbody>
</table>

4.1.12.1  FB_MBUS_EMU_32x7

This block is used to read electricity meters from EMU:

-EMU32.x7

Only the standard meter data will be read. The meter transmits this data in the standard EMU parameterization, or if the parameter set is set in the device to 00000 hexadecimal. Please refer to the meter documentation for further information regarding this.

In the normal version, the current consumption of the M-Bus interface is equivalent to 3 standard loads. If an M-Bus master interface is used that is designed, for example, for up to 120 standard loads, a maximum of 40 EMU M-Bus meters can be connected. The meter can optionally be supplied with 230 V. The current consumption of the M-Bus interface is then equivalent to one standard load.

The transmission of data from the EMU meter to the M-Bus protocol computer only works if the EMU meter is connected to at least two phases of the mains voltage network.
The EMU meter transmits current data to the device’s M-Bus interface every 40 seconds, so that the readout data is approx. 40 - 45 seconds old.

The block can only be executed together with the block FB_MBUSKL6781) [18].

Functionality of the block [9]

VAR_INPUT

usiAddress : USINT;
stSecAdr : ST_MBUS_SecAdr;
eBaudrate : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart : BOOL;
bSND_NKE : BOOL := TRUE;
bReadInit : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit : USINT;
bDisabled : BOOL := FALSE;

usiAddress: Primary address [10] of the meter to be read with this block.

stSecAdr: Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAdr [205]).

eBaudrate: 300, 600, 1200, 2400, 4800, 9600 baud (see E_MBUS_Baudrate [197]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bReady : BOOL;
bError : BOOL;
eError : E_MBUS_ERROR;
dwIdNumber : DWORD;
byStatus : BYTE;
byGEN : BYTE;
byCounter : BYTE;
usiRecivedAdr : USINT;
eMedium : E_MBUS_Medium;
zman : STRING(3);
stOperatingHours : ST_MBus_Info;
stEnergy : ST_MBus_Info;
stPower : ST_MBus_Info;
istInitCounter : ST_MBus_Info;
istDeviceError : ST_MBus_Info;

bBusy: The bBusy output is TRUE while the meter is being read.

bReady: The bReady output is TRUE for one cycle, once meter reading is completed.

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

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.
byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [200]).

sMan: Manufacturer code.

stOperatingHours: Operating hours of the EMU meter (see ST_MBus_Info [204]).

stEnergy: Meter reading, energy consumption (see ST_MBus_Info [204]).

stPower: Current energy consumption, power (see ST_MBus_Info [204]).

stInitCounter: Number of power failures at the EMU meter (see ST_MBus_Info [204]).

stDeviceError: Error message from device (see ST_MBus_Info [204]).

VAR_IN_OUT

stCom : ST_MBUS_Communication;

stCom: This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Communication [202]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.12.2 FB_MBUS_EMU_32x7_Option8

This block is used to read electricity meters from EMU:
-EMU32.x7

The parameter set must be set in the device to 70000 hexadecimal (variant 8) in order to read out this data. Please refer to the meter documentation for further information regarding this.

In the normal version, the current consumption of the M-Bus interface is equivalent to 3 standard loads. If an M-Bus master interface is used that is designed, for example, for up to 120 standard loads, a maximum of 40 EMU M-Bus meters can be connected. The meter can optionally be supplied with 230 V. The current consumption of the M-Bus interface is then equivalent to one standard load.

The transmission of data from the EMU meter to the M-Bus protocol computer only works if the EMU meter is connected to at least two phases of the mains voltage network.
The EMU meter transmits current data to the device’s M-Bus interface every 40 seconds, so that the readout data is approx. 40 - 45 seconds old.

The block can only be executed together with the block FB_MBUSKL6781([18]).

Functionality of the block ([9])

**VAR_INPUT**

<table>
<thead>
<tr>
<th>usiAddress</th>
<th>USINT;</th>
</tr>
</thead>
<tbody>
<tr>
<td>stSecAdr</td>
<td>ST_MBUS_SecAdr;</td>
</tr>
<tr>
<td>eBaudrate</td>
<td>E_MBUS_Baudrate := eMBUS_Baud2400;</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL;</td>
</tr>
<tr>
<td>bSND_NKE</td>
<td>BOOL := TRUE;</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>TIME := t#2s;</td>
</tr>
<tr>
<td>usiUnit</td>
<td>USINT;</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL := FALSE;</td>
</tr>
</tbody>
</table>

**usiAddress**: Primary address ([10]) of the meter to be read with this block.

**stSecAdr**: Secondary address ([11]) of the meter to be read with this block (see ST_MBUS_SecAdr ([205]).

**eBaudrate**: 300, 600, 1200, 2400, 4800, 9600 baud (see E_MBUS_Baudrate ([197]).

**bStart**: A positive edge of this input triggers one reading of the meter.

**bSND_NKE**: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

**bReadInit**: If the PLC is restarted, the meter is read once.

**tMinSendTime**: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.

**usiUnit**: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

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

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>bBusy</th>
<th>BOOL;</th>
</tr>
</thead>
<tbody>
<tr>
<td>bReady</td>
<td>BOOL;</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL;</td>
</tr>
<tr>
<td>eError</td>
<td>E_MBUS_ERROR;</td>
</tr>
<tr>
<td>dwIdNumber</td>
<td>DWORD;</td>
</tr>
<tr>
<td>byStatus</td>
<td>BYTE;</td>
</tr>
<tr>
<td>byGEN</td>
<td>BYTE;</td>
</tr>
<tr>
<td>byCounter</td>
<td>BYTE;</td>
</tr>
<tr>
<td>usiRecivedAdr</td>
<td>USINT;</td>
</tr>
<tr>
<td>eMedium</td>
<td>E_MBUS_Medium;</td>
</tr>
<tr>
<td>sMan</td>
<td>STRING{3};</td>
</tr>
<tr>
<td>stOperatingHours</td>
<td>ST_MBUs_Info;</td>
</tr>
<tr>
<td>stActiveEnergyT1</td>
<td>ST_MBUs_Info;</td>
</tr>
<tr>
<td>stActiveEnergyT2</td>
<td>ST_MBUs_Info;</td>
</tr>
<tr>
<td>stReactiveEnergyT1</td>
<td>ST_MBUs_Info;</td>
</tr>
<tr>
<td>stReactiveEnergyT2</td>
<td>ST_MBUs_Info;</td>
</tr>
<tr>
<td>stTariff</td>
<td>ST_MBUs_Info;</td>
</tr>
<tr>
<td>stActivePowerL1</td>
<td>ST_MBUs_Info;</td>
</tr>
<tr>
<td>stActivePowerL2</td>
<td>ST_MBUs_Info;</td>
</tr>
<tr>
<td>stActivePowerL3</td>
<td>ST_MBUs_Info;</td>
</tr>
<tr>
<td>stReactivePowerL1</td>
<td>ST_MBUs_Info;</td>
</tr>
<tr>
<td>stReactivePowerL2</td>
<td>ST_MBUs_Info;</td>
</tr>
<tr>
<td>stReactivePowerL3</td>
<td>ST_MBUs_Info;</td>
</tr>
<tr>
<td>stActivePowerMaxT1</td>
<td>ST_MBUs_Info;</td>
</tr>
<tr>
<td>stActivePowerMaxT2</td>
<td>ST_MBUs_Info;</td>
</tr>
<tr>
<td>stVoltageL1</td>
<td>ST_MBUs_Info;</td>
</tr>
<tr>
<td>stVoltageL2</td>
<td>ST_MBUs_Info;</td>
</tr>
<tr>
<td>stVoltageL3</td>
<td>ST_MBUs_Info;</td>
</tr>
<tr>
<td>stCurrentL1</td>
<td>ST_MBUs_Info;</td>
</tr>
<tr>
<td>stCurrentL2</td>
<td>ST_MBUs_Info;</td>
</tr>
<tr>
<td>stCurrentL3</td>
<td>ST_MBUs_Info;</td>
</tr>
<tr>
<td>stFrequency</td>
<td>ST_MBUs_Info;</td>
</tr>
<tr>
<td>stPowerFactorL1</td>
<td>ST_MBUs_Info;</td>
</tr>
<tr>
<td>stPowerFactorL2</td>
<td>ST_MBUs_Info;</td>
</tr>
</tbody>
</table>
**bBusy**: The `bBusy` output is TRUE while the meter is being read.

**bReady**: The `bReady` output is TRUE for one cycle, once meter reading is completed.

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

**eError**: In the event of an error the output issues an error code (see `E_MBUS_ERROR`). `bError` goes TRUE at the same time.

**dwIdNumber**: Serial number of the meter (secondary address).

**byStatus**: Device status.

**byGEN**: Software version of the device.

**byCounter**: Number of times the master has accessed data of the respective slave.

**usiRecivedAdr**: Received primary address (0-250).

**eMedium**: Medium (see `E_MBUS_Medium`).

**sMan**: Manufacturer code.

**stOperatingHours**: Operating hours of the EMU meter (see `ST_MBus_Info`).

**stActiveEnergyT1**: Meter reading, active energy tariff 1 (see `ST_MBus_Info`).

**stActiveEnergyT2**: Meter reading, active energy tariff 2 (see `ST_MBus_Info`).

**stReactiveEnergyT1**: Meter reading, reactive energy tariff 1 (see `ST_MBus_Info`).

**stReactiveEnergyT2**: Meter reading, reactive energy tariff 2 (see `ST_MBus_Info`).

**stTariff**: Current tariff (see `ST_MBus_Info`).

**stActivePowerL1**: Instantaneous consumption, effective power L1 (see `ST_MBus_Info`).

**stActivePowerL2**: Instantaneous consumption, effective power L2 (see `ST_MBus_Info`).

**stActivePowerL3**: Instantaneous consumption, effective power L3 (see `ST_MBus_Info`).

**stReactivePowerL1**: Instantaneous consumption, reactive power L1 (see `ST_MBus_Info`).

**stReactivePowerL2**: Instantaneous consumption, reactive power L2 (see `ST_MBus_Info`).

**stReactivePowerL3**: Instantaneous consumption, reactive power L3 (see `ST_MBus_Info`).

**stActivePowerMaxT1**: Maximum effective power tariff 1 (see `ST_MBus_Info`).

**stActivePowerMaxT2**: Maximum effective power tariff 2 (see `ST_MBus_Info`).

**stVoltageL1**: Current voltage L1 (see `ST_MBus_Info`).

**stVoltageL2**: Current voltage L2 (see `ST_MBus_Info`).

**stVoltageL3**: Current voltage L3 (see `ST_MBus_Info`).

**stCurrentL1**: Current current L1 (see `ST_MBus_Info`).

**stCurrentL2**: Current current L2 (see `ST_MBus_Info`).

**stCurrentL3**: Current current L3 (see `ST_MBus_Info`).

**stFrequency**: Current mains frequency (see `ST_MBus_Info`).
stPowerFactorL1: Current form factor phase L1 (cos Phi) (see ST_MBus_Info [p. 204]).
stPowerFactorL2: Current form factor phase L2 (cos Phi) (see ST_MBus_Info [p. 204]).
stPowerFactorL3: Current form factor phase L3 (cos Phi) (see ST_MBus_Info [p. 204]).
stInitCounter: Number of power failures at the EMU meter (see ST_MBus_Info [p. 204]).
stDeviceError: Error message from device (see ST_MBus_Info [p. 204]).

VAR_IN_OUT
stCom : ST_MBUS_Communication;

stCom: This structure is used to link the block FB_MBUSKL6781() [p. 202] with the meter blocks (see ST_MBUS_Communication [p. 202]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.12.3 FB_MBUS_EMU_3_5_Allrounder

This block is used to read electricity meters from EMU.
The block can only be executed together with the block FB_MBUSKL6781() [p. 18].

Functionality of the block [p. 9]

VAR_INPUT
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit       : BOOL := TRUE;
tMinSendTime    : TIME := t#2s;
bDisabled       : BOOL := FALSE;

usiAddress: Primary address [10] of the meter to be read with this block.

stSecAddr: Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAdr [205]).

eBaudrate: 300..9600 baud (see E_MBUS_Baudrate [197]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy       : BOOL;
bReady      : BOOL;
bError      : BOOL;
eError      : E_MBUS_ERROR;
dwIdNumber  : DWORD;
byStatus    : BYTE;
byGEN       : BYTE;
byCounter   : BYTE;
usiRecivedAdr: USINT;
eMedium     : E_MBUS_Medium;
sMan        : STRING(3);

stActiveEnergyTariff1 : ST_MBus_Info;
stActiveEnergyTariff2 : ST_MBus_Info;
stActivePowerL1       : ST_MBus_Info;
stActivePowerL2       : ST_MBus_Info;
stActivePowerL3       : ST_MBus_Info;
stActivePowerTotal    : ST_MBus_Info;
stVoltageL1           : ST_MBus_Info;
stVoltageL2           : ST_MBus_Info;
stVoltageL3           : ST_MBus_Info;
stCurrentL1           : ST_MBus_Info;
stCurrentL2           : ST_MBus_Info;
stCurrentL3           : ST_MBus_Info;
stCurrentTotal        : ST_MBus_Info;
stDeviceError         : ST_MBus_Info;

bBusy: The bBusy output is TRUE while the meter is being read.

bReady: The bReady output is TRUE for one cycle, once meter reading is completed.

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

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [200]).

sMan: Manufacturer code.
stActiveEnergyTariff1: Active energy tariff 1 (see ST_MBus_Info [204]).

stActiveEnergyTariff2: Active energy tariff 2 (see ST_MBus_Info [204]).

stActivePowerL1: Effective power L1 (see ST_MBus_Info [204]).

stActivePowerL2: Effective power L2 (see ST_MBus_Info [204]).

stActivePowerL3: Effective power L3 (see ST_MBus_Info [204]).

stActivePowerTotal: Total effective power (see ST_MBus_Info [204]).

stVoltageL1: Voltage L1 (see ST_MBus_Info [204]).

stVoltageL2: Voltage L2 (see ST_MBus_Info [204]).

stVoltageL3: Voltage L3 (see ST_MBus_Info [204]).

stCurrentL1: Current intensity L1 (see ST_MBus_Info [204]).

stCurrentL2: Current intensity L2 (see ST_MBus_Info [204]).

stCurrentL3: Current intensity L3 (see ST_MBus_Info [204]).

stCurrentTotal: Total current intensity (see ST_MBus_Info [204]).

stDeviceError: Error message from device (see ST_MBus_Info [204]).

VAR_IN_OUT

stCom : ST_MBUS_Communication;

stCom: This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Communication [202]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.12.4 FB_MBUS_EMU_DHZ_5_63

This block is used to read electricity meters from EMU.

The block can only be executed together with the block FB_MBUSKL6781() [18].
Functionality of the block [9]

VAR_INPUT

usiAddress : USINT;
stSecAdr   : ST_MBUS_SecAddr;
eBaudrate  : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart     : BOOL;
bSND_NKE   : BOOL := TRUE;
bReadInit  : BOOL := TRUE;
tMinSendTime: TIME := t#2s;
usiUnit    : USINT;
bDisabled  : BOOL := FALSE;

usiAddress: Primary address [10] of the meter to be read with this block.

stSecAdr: Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAddr [205]).

eBaudrate: 300..9600 baud (see E_MBUS_Baudrate [197]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy        : BOOL;
bReady       : BOOL;
bError       : BOOL;
eError       : E_MBUS_ERROR;
dwIdNumber   : DWORD;
byStatus     : BYTE;
byGEN        : BYTE;
byCounter    : BYTE;
usiRecivedAdr: USINT;
eMedium      : E_MBUS_Medium;
strMan       : STRING(3);
stEnergy1    : ST_MBus_Info;
stEnergy2    : ST_MBus_Info;
stVoltage    : ST_MBus_Info;
stCurrent    : ST_MBus_Info;
stPower      : ST_MBus_Info;

bBusy: The bBusy output is TRUE while the meter is being read.

bReady: The bReady output is TRUE for one cycle, once meter reading is completed.

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

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [200]).
sMan: Manufacturer code.

stEnergy1: Effective energy 1 (see ST_MBus_Info ▶ 204).

stEnergy2: Effective energy 2 (resettable) (see ST_MBus_Info ▶ 204).

stVoltage: Mains voltage (see ST_MBus_Info ▶ 204).

stCurrent: Instantaneous current (see ST_MBus_Info ▶ 204).

stPower: Instantaneous active power (see ST_MBus_Info ▶ 204).

**VAR_IN_OUT**

stCom: ST_MBUS_Communication;

stCom: This structure is used to link the block FB_MBUSKL6781() ▶ 202 with the meter blocks (see ST_MBUS_Communication ▶ 202).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.13 Engelmann overview**

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General ▶ 76, FB_MBUS_General_Ext ▶ 79 or FB_MBUS_General_Param ▶ 83 from the folder "General ▶ 74" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() ▶ 85 can be used to send data to the device (e.g. setting of the primary address).

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Type</th>
<th>Device</th>
<th>Block</th>
</tr>
</thead>
<tbody>
<tr>
<td>Engelmann</td>
<td>Heat meter</td>
<td>Sensostar 2C</td>
<td>FB_MBUS_EFF_SensoStar2C ▶ 72</td>
</tr>
</tbody>
</table>
4.1.13.1 FB_MBUS_EFE_SensoStar2C

This block is used to read heat meters from Engelmann:
- SENSOSTAR 2C

The block can only be executed together with the block FB_MBUSKL6781() [18].

Functionality of the block [9]

VAR_INPUT

usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit      : USINT;
bDisabled    : BOOL := FALSE;

usiAddress: Primary address [10] of the meter to be read with this block.

stSecAdr: Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAdr [205]).

eBaudrate: (see E_MBUS_baud rate [197]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.
**tMinSendTime:** Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.

**usiUnit:** Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

**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>bReady</td>
<td>BOOL;</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL;</td>
</tr>
<tr>
<td>eError</td>
<td>E_MBUS_ERROR;</td>
</tr>
<tr>
<td>dwIdNumber</td>
<td>DWORD;</td>
</tr>
<tr>
<td>byStatus</td>
<td>BYTE;</td>
</tr>
<tr>
<td>byGEN</td>
<td>BYTE;</td>
</tr>
<tr>
<td>usiRecivedAdr</td>
<td>USINT;</td>
</tr>
<tr>
<td>eMedium</td>
<td>E_MBUS_Medium;</td>
</tr>
<tr>
<td>sMan</td>
<td>STRING(3);</td>
</tr>
<tr>
<td>stEnergy</td>
<td>ST_MBus_Info;</td>
</tr>
<tr>
<td>stColdEnergy</td>
<td>ST_MBus_Info;</td>
</tr>
<tr>
<td>stPower</td>
<td>ST_MBus_Info;</td>
</tr>
<tr>
<td>stVolume</td>
<td>ST_MBus_Info;</td>
</tr>
<tr>
<td>stFlow</td>
<td>ST_MBus_Info;</td>
</tr>
<tr>
<td>stForwardTemp</td>
<td>ST_MBus_Info;</td>
</tr>
<tr>
<td>stReturnTemp</td>
<td>ST_MBus_Info;</td>
</tr>
<tr>
<td>stDiffTemp</td>
<td>ST_MBus_Info;</td>
</tr>
<tr>
<td>stTariff1</td>
<td>ST_MBus_Info;</td>
</tr>
<tr>
<td>stTariff2</td>
<td>ST_MBus_Info;</td>
</tr>
<tr>
<td>stPulsecounter1</td>
<td>ST_MBus_Info;</td>
</tr>
<tr>
<td>stPulsecounter2</td>
<td>ST_MBus_Info;</td>
</tr>
<tr>
<td>stDeviceError</td>
<td>ST_MBus_Info;</td>
</tr>
</tbody>
</table>

**bBusy:** The bBusy output is TRUE while the meter is being read.

**bReady:** The bReady output is TRUE for one cycle, once meter reading is completed.

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

**eError:** In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.

**dwIdNumber:** Serial number of the meter (secondary address).

**byStatus:** Device status.

**byGEN:** Software version of the device.

**byCounter:** Number of times the master has accessed data of the respective slave.

**usiRecivedAdr:** Received primary address (0-250).

**eMedium:** Medium (see E_MBUS_Medium [201]).

**sMan:** Manufacturer code.

**stEnergy:** Meter reading, energy consumption (see ST_MBus_Info [204]).

**stColdEnergy:** Energy consumption meter reading (see ST_MBus_Info [204]).

**stPower:** Current energy consumption, power (see ST_MBus_Info [204]).

**stVolume:** Meter reading, water consumption (see ST_MBus_Info [204]).

**stFlow:** Current water consumption (see ST_MBus_Info [204]).

**stForwardTemp:** Flow temperature (see ST_MBus_Info [204]).

**stReturnTemp:** Return temperature (see ST_MBus_Info [204]).

**stDiffTemp:** Temperature difference (see ST_MBus_Info [204]).
stTariff1: Tariff register 1 (see ST_MBus_Info [ 204]).

stTariff2: Tariff register 2 (see ST_MBus_Info [ 204]).

stPulsecounter1: Pulse counter 1 (see ST_MBus_Info [ 204]).

stPulsecounter2: Pulse counter 2 (see ST_MBus_Info [ 204]).

stDeviceError: Error message from device (see ST_MBus_Info [ 204]).

VAR_IN_OUT

stCom : ST_MBUS_Communication;

stCom: This structure is used to link the block FB_MBUSL6781() [ 202] with the meter blocks (see ST_MBUS_Communication [ 202]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.14 General device blocks

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Type</th>
<th>Device</th>
<th>Block</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td>Electricity meter</td>
<td>all electricity meters</td>
<td>FB_MBUS_General_Electricity [ 77]</td>
</tr>
<tr>
<td></td>
<td>Heat meter</td>
<td>all heat meters</td>
<td>FB_MBUS_General_Heat [ 81]</td>
</tr>
<tr>
<td></td>
<td>Water meter</td>
<td>all water meters</td>
<td>FB_MBUS_General_Water [ 86]</td>
</tr>
<tr>
<td></td>
<td>Raw data of the first telegram</td>
<td>all</td>
<td>FB_MBUS_RawData [ 88]</td>
</tr>
<tr>
<td></td>
<td>max. 40 values from the first telegram</td>
<td>all</td>
<td>FB_MBUS_General [ 76]</td>
</tr>
<tr>
<td></td>
<td>All telegrams, all values</td>
<td>all</td>
<td>FB_MBUS_General_Ext [ 79]</td>
</tr>
<tr>
<td></td>
<td>values parameterizable</td>
<td>all</td>
<td>FB_MBUS_General_Param [ 83]</td>
</tr>
<tr>
<td></td>
<td>universal send blocks</td>
<td>all</td>
<td>FB_MBUS_General_Send [ 85]</td>
</tr>
<tr>
<td></td>
<td>scan block</td>
<td>all</td>
<td>FB_MBUS_Scan [ 89]</td>
</tr>
<tr>
<td></td>
<td>Change address</td>
<td>all</td>
<td>FB_MBUS_ChangeAdr [ 75]</td>
</tr>
</tbody>
</table>
4.1.14.1 FB_MBUS_ChangeAdr

This block can be used to change the primary address.

The block can only be executed together with the block FB_MBUSKL6781 () [18].

VAR_INPUT
usiAdrOld : USINT;
usiAdrNew : USINT;
eBaudrate : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart : BOOL;
bDisabled : BOOL:=FALSE;

usiAdrOld: Old primary address.
usiAdrNew: New primary address.
eBaudrate: 300, 600, 1200, 2400, 4800, 9600 baud (see E_MBUS_Baudrate [197]).
bStart: A positive edge at this input triggers a change of the primary address of the counter.
bDisabled: TRUE = deselection of the block.

VAR_OUTPUT
bBusy : BOOL;
bReady : BOOL;
bError : BOOL;
eError : E_MBUS_ERROR;

bBusy: The bBusy output is TRUE while the meter is being read.
bReady: The bReady output is TRUE for one cycle, once meter reading is completed.
bError: this output goes TRUE as soon as an error occurs. This error is described via the variable eError.
eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.

VAR_IN_OUT
stCom : ST_MBUS_Communication;

stCom: This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Communication [202]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.1.14.2    FB_MBUS_General

This block is used for reading any M-Bus devices. The variable `arrData` supplies a maximum of `cMBUS_MaxData` values for the first telegram. String values and manufacturer-specific information are not shown correctly.

The block can only be executed together with the block `FB_MBUSKL6781`.

Functionality of the block [9]

**VAR_INPUT**

```plaintext
usiAddress : USINT;
stSecAdr : ST_MBUS_SecAdr;
eBaudrate : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart : BOOL;
bSND_NKE : BOOL := TRUE;
bReadInit : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit : USINT;
bDisabled : BOOL := FALSE;
```

`usiAddress`: Primary address [10] of the meter to be read with this block.

`stSecAdr`: Secondary address [11] of the meter to be read with this block (see `ST_MBUS_SecAdr` [205]).

`eBaudrate`: 300, 600, 1200, 2400, 4800, 9600 baud (see `E_MBUS_Baudrate` [197]).

`bStart`: A positive edge of this input triggers one reading of the meter.

`bSND_NKE`: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

`bReadInit`: If the PLC is restarted, the meter is read once.

`tMinSendTime`: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with `bStart`.

`usiUnit`: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

`bDisabled`: TRUE = deselection of the block.

**VAR_OUTPUT**

```plaintext
bBusy : BOOL;
bReady : BOOL;
bError : BOOL;
eError : E_MBUS_ERROR;
dwIdNumber : DWORD;
byStatus : BYTE;
byGEN : BYTE;
```

This block is used for reading any M-Bus devices. The variable `arrData` supplies a maximum of `cMBUS_MaxData` values for the first telegram. String values and manufacturer-specific information are not shown correctly.

The block can only be executed together with the block `FB_MBUSKL6781`. [18]
byCounter        : BYTE;
usiRecivedAddr   : USINT;
eMedium          : E_MBUS_Medium;
sMan             : STRING(3);
arrData          : ARRAY [1..cMBUS_MaxData] OF ST_MBus_Data;

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

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

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAddr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [200]).

sMan: Manufacturer code.

arrData: Up to cMBUS_MaxData [207] values of the first telegram. The meaning of the values is explained in the M-Bus protocol for the device.

VAR_IN_OUT

stCom : ST_MBUS_Communication;

stCom: This structure is used to link the block *FB_MBUSKL6781* [202] with the meter blocks (see ST_MBUS_Communication [202]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.14.3  **FB_MBUS_General_Electricity**

This block is used to read electricity meters.
Not all electricity meters automatically send power data. In this case the corresponding structure remains empty.

The block can only be executed together with the block FB_MBUSKL6781 [18].

Functionality of the block [9]

**VAR_INPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>usiAddress</td>
<td>Primary address [10] of the meter to be read with this block.</td>
</tr>
<tr>
<td>stSecAdr</td>
<td>Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAdr [205]).</td>
</tr>
<tr>
<td>eBaudrate</td>
<td>300, 600, 1200, 2400, 4800, 9600 baud (see E_MBUS_Baudrate [197]).</td>
</tr>
<tr>
<td>bStart</td>
<td>A positive edge of this input triggers one reading of the meter.</td>
</tr>
<tr>
<td>bSND_NKE</td>
<td>TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.</td>
</tr>
<tr>
<td>usiUnit</td>
<td>Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).</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 while the meter is being read.</td>
</tr>
<tr>
<td>bReady</td>
<td>The bReady output is TRUE for one cycle, once meter reading is completed.</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>In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.</td>
</tr>
<tr>
<td>dwIdNumber</td>
<td>Serial number of the meter (secondary address).</td>
</tr>
<tr>
<td>byStatus</td>
<td>Device status.</td>
</tr>
<tr>
<td>byGEN</td>
<td>Software version of the device.</td>
</tr>
</tbody>
</table>
byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [200]).

sMan: Manufacturer code.

stEnergy: Meter reading, energy consumption (see ST_MBus_Info [204]).

stPower: Current energy consumption, power (see ST_MBus_Info [204]).

**VAR_IN_OUT**

stCom : ST_MBUS_Communication;

stCom: This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Communication [202]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.14.4 FB_MBUS_General_Ext**

Some M-Bus devices send values distributed over several telegrams. This block can be used to read all telegrams from any M-Bus devices.

The variable arrTelegram[1..cMBUS_MaxTelegrams].arrData[1..cMBUS_MaxData] supplies a maximum of cMBUS_MaxData [207] data from a maximum of cMBUS_MaxTelegrams [207] telegrams. String values and manufacturer-specific information are not shown correctly.

The number of telegrams to be read can be changed with the constant cMBUS_MaxTelegrams [207].

The number of data per telegram to be read can be changed with the constant cMBUS_MaxData [207].

The block can only be executed together with the block FB_MBUSKL6781() [18].

Functionality of the block [9]

**VAR_INPUT**

usiAddress : USINT;

stSecAdr : ST_MBUS_SecAdr;

eBaudrate : E_MBUS_Baudrate := eMBUS_Baud2400;
**bStart**: A positive edge of this input triggers one reading of the meter.

**bSND_NKE**: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

**bReadInit**: If the PLC is restarted, the meter is read once.

**tMinSendTime**: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.

**usiUnit**: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

**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 while the meter is being read.</td>
</tr>
<tr>
<td>bReady</td>
<td>BOOL</td>
<td>The bReady output is TRUE for one cycle, once meter reading is completed.</td>
</tr>
<tr>
<td>bError</td>
<td>E_MBUS_ERROR</td>
<td>This output goes TRUE as soon as an error occurs. This error is described via variable eError.</td>
</tr>
<tr>
<td>dwIdNumber</td>
<td>DWORD</td>
<td>Serial number of the meter (secondary address).</td>
</tr>
<tr>
<td>byStatus</td>
<td>BYTE</td>
<td>Device status.</td>
</tr>
<tr>
<td>byGEN</td>
<td>BYTE</td>
<td>Software version of the device.</td>
</tr>
<tr>
<td>byCounter</td>
<td>BYTE</td>
<td>Number of times the master has accessed data of the respective slave.</td>
</tr>
<tr>
<td>usiRecivedAdr</td>
<td>USINT</td>
<td>Received primary address (0-250).</td>
</tr>
<tr>
<td>eMedium</td>
<td>E_MBUS_Medium</td>
<td>Medium (see E_MBUS_Medium [200]).</td>
</tr>
<tr>
<td>sMan</td>
<td>STRING(3)</td>
<td>Manufacturer code.</td>
</tr>
<tr>
<td>arrTelegram</td>
<td>ARRAY [1..cMBUS_MaxTelegrams] OF ST_MBus_Data2</td>
<td>Up to cMBUS_MaxTelegrams [207] telegrams (see ST_MBus_data [204]). The meaning of the values is explained in the M-Bus protocol for the device.</td>
</tr>
</tbody>
</table>

**VAR_IN_OUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stCom</td>
<td>ST_MBUS_Communication;</td>
<td></td>
</tr>
</tbody>
</table>
**stCom**: This structure is used to link the block **FB_MBUSKL6781() [202]** with the meter blocks (see **ST_MBUS_Communication [202])**.

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.14.5 **FB_MBUS_General_Heat**

This block is used to read heat meters. Many heat meters do not send all values. In this case the corresponding structures remain empty.

The block can only be executed together with the block **FB_MBUSKL6781() [18]**.

**Functionality of the block [9]**

**VAR_INPUT**

- `usiAddress` : USINT;
- `stSecAdr` : ST_MBUS_SecAdr;
- `eBaudrate` : E_MBUS_Baudrate := eMBUS_Baud2400;
- `bStart` : BOOL;
- `bSND_NKE` : BOOL := TRUE;
- `bReadInit` : BOOL := TRUE;
- `tMinSendTime` : TIME := t#2s;
- `usiUnit` : USINT;
- `bDisabled` : BOOL := FALSE;

- `usiAddress`: Primary address [10] of the meter to be read with this block.

- `stSecAdr`: Secondary address [11] of the meter to be read with this block (see **ST_MBUS_SecAdr [205]**).

- `eBaudrate`: 300, 600, 1200, 2400, 4800, 9600 baud (see **E_MBUS_Baudrate [197]**).

- `bStart`: A positive edge of this input triggers one reading of the meter.
**bSND_NKE**: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

**bReadInit**: If the PLC is restarted, the meter is read once.

**tMinSendTime**: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with **bStart**.

**usiUnit**: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2=MW(h) / 3=GW(h).

**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 <strong>bBusy</strong> output is TRUE while the meter is being read.</td>
</tr>
<tr>
<td>bReady</td>
<td>The <strong>bReady</strong> output is TRUE for one cycle, once meter reading is completed.</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>In the event of an error the output issues an error code (see <strong>E_MBUS_ERROR</strong>) <strong>bError</strong> goes TRUE at the same time.</td>
</tr>
<tr>
<td>dwIdNumber</td>
<td>Serial number of the meter (secondary address).</td>
</tr>
<tr>
<td>byStatus</td>
<td>Device status.</td>
</tr>
<tr>
<td>byGEN</td>
<td>Software version of the device.</td>
</tr>
<tr>
<td>byCounter</td>
<td>Number of times the master has accessed data of the respective slave.</td>
</tr>
<tr>
<td>usiRecivedAdr</td>
<td>Received primary address (0-250).</td>
</tr>
<tr>
<td>eMedium</td>
<td>Medium (see <strong>E_MBUS_Medium</strong>).</td>
</tr>
<tr>
<td>sMan</td>
<td>Manufacturer code.</td>
</tr>
<tr>
<td>stEnergy</td>
<td>Meter reading, energy consumption (see <strong>ST_MBus_Info</strong>).</td>
</tr>
<tr>
<td>stPower</td>
<td>Current energy consumption, power (see <strong>ST_MBus_Info</strong>).</td>
</tr>
<tr>
<td>stVolume</td>
<td>Meter reading, water consumption (see <strong>ST_MBus_Info</strong>).</td>
</tr>
<tr>
<td>stFlow</td>
<td>Current water consumption (see <strong>ST_MBus_Info</strong>).</td>
</tr>
<tr>
<td>stForwardTemp</td>
<td>Flow temperature (see <strong>ST_MBus_Info</strong>).</td>
</tr>
<tr>
<td>stReturnTemp</td>
<td>Return temperature (see <strong>ST_MBus_Info</strong>).</td>
</tr>
<tr>
<td>stDiffTemp</td>
<td>Temperature difference (see <strong>ST_MBus_Info</strong>).</td>
</tr>
</tbody>
</table>
VAR_IN_OUT

stCom : ST_MBUS_Communication;

stCom: This structure is used to link the block FB_MBUSKL6781() \[202\] with the meter blocks (see ST_MBUS_Communication \[202\]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.14.6 FB_MBUS_General_Param

<table>
<thead>
<tr>
<th>VAR_IN_OUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>usiAddress : USINT;</td>
</tr>
<tr>
<td>stSecAdr : ST_MBUS_SecAdr;</td>
</tr>
<tr>
<td>eBaudrate : E_MBUS_Baudrate := eMBUS_Baud2400;</td>
</tr>
<tr>
<td>bStart : BOOL;</td>
</tr>
<tr>
<td>bSND_NKE : BOOL := TRUE;</td>
</tr>
<tr>
<td>bReadInit : BOOL := TRUE;</td>
</tr>
<tr>
<td>tMinSendTime : TIME := t#2s;</td>
</tr>
<tr>
<td>usiUnit : USINT;</td>
</tr>
<tr>
<td>bDisabled : BOOL := FALSE;</td>
</tr>
<tr>
<td>arrConfigData : ARRAY [1..cMBUS_MaxDataParam] OF WORD;</td>
</tr>
</tbody>
</table>

This block is used for reading any M-Bus devices. The variable arrData \[203\] supplies cMBUS_MaxDataParam \[207\] values.

These values can be parameterized in the input array arrConfigData. String values and manufacturer-specific information are not shown correctly.

The block can only be executed together with the block FB_MBUSKL6781() \[18\].

Functionality of the block \[9\]

VAR_INPUT

usiAddress: Primary address \[10\] of the meter to be read with this block.

stSecAdr: Secondary address \[11\] of the meter to be read with this block (see ST_MBUS_SecAdr \[205\]).

eBaudrate: 300, 600, 1200, 2400, 4800, 9600 baud (see E_MBUS_Baudrate \[197\]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).
bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard 2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2=MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

arrConfigData: Up to cMBUS_MaxDataParam input parameters for specifying which values are to be displayed in the output array arrData (see Global_Variables_MBUS [207]).

VAR_OUTPUT

bBusy : BOOL;
bReady : BOOL;
bError : BOOL;
eError : E_MBUS_ERROR;
dwIdNumber : DWORD;
byStatus : BYTE;
byGEN : BYTE;
byCounter : BYTE;
usiRecivedAdr : USINT;
eMedium : E_MBUS_Medium;
sMan : STRING(3);
arrData : ARRAY [1..cMBUS_MaxDataParam] OF ST_MBus_Data;

bBusy: The bBusy output is TRUE while the meter is being read.

bReady: The bReady output is TRUE for one cycle, once meter reading is completed.

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

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [200]).

sMan: Manufacturer code.

arrData: Up to cMBUS_MaxDataParam values (see Global_Variables_MBUS [207]). The values can be configured via the input variable arrConfigData. The meaning of the values is explained in the M-Bus protocol for the device.

VAR_IN_OUT

stCom : ST_MBUS_Communication;

stCom: This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Communication [202]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.1.14.7 FB_MBUS_General_Send

This block serves to send data to any M-Bus devices. (for example, the primary address of the meter can be changed with this block)

The block can only be executed together with the block FB_MBUSKL6781() [18].

Functionality of the function block [9]

VAR_INPUT

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>usiAddress</td>
<td>USINT;</td>
</tr>
<tr>
<td>stSecAdr</td>
<td>ST_MBUS_SecAdr;</td>
</tr>
<tr>
<td>eBaudrate</td>
<td>E_MBUS_Baudrate := eMBUS_Baud2400;</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL;</td>
</tr>
<tr>
<td>bSND_NKE</td>
<td>BOOL := TRUE;</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL := FALSE;</td>
</tr>
<tr>
<td>byC_Field</td>
<td>USINT := 16#53;</td>
</tr>
<tr>
<td>byCI_Field</td>
<td>USINT := 16#51;</td>
</tr>
<tr>
<td>arrData</td>
<td>ARRAY [0..240] OF BYTE;</td>
</tr>
</tbody>
</table>

usiAddress: Primary address [10] of the meter to be read with this block.

stSecAdr: Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAdr [205]).

eBaudrate: 300, 600, 1200, 2400, 4800, 9600 baud (see E_MBUS_Baudrate [197]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bDisabled: TRUE = deselection of the block.

byC_Field: C-field / function field.

byCI_Field: CI-field / ID field.

arrData: The data to be sent must be written to this variable (see long block [10]).

VAR_OUTPUT

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>BOOL;</td>
</tr>
<tr>
<td>bReady</td>
<td>BOOL;</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL;</td>
</tr>
<tr>
<td>eError</td>
<td>E_MBUS_ERROR;</td>
</tr>
</tbody>
</table>

bBusy: The bBusy output is TRUE while the meter is being read.

bReady: The bReady output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable eError.
eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.

VAR_IN_OUT

stCom : ST_MBUS_Communication;

stCom: This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Communication [202]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.14.8 FB_MBUS_General_Water

This block is used to read water meters.

Not all water meters automatically send the flow rate. In this case the corresponding structure remains empty.

The block can only be executed together with the block FB_MBUSKL6781() [18].

Functionality of the block [9]

VAR_INPUT

usiAddress : USINT;
stSecAdr : ST_MBUS_SecAdr;
eBaudrate : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart : BOOL;
bSND_NKE : BOOL := TRUE;
bReadInit : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
bDisabled : BOOL := FALSE;

usiAddress: Primary address [10] of the meter to be read with this block.

stSecAdr: Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAdr [205]).

eBaudrate: 300, 600, 1200, 2400, 4800, 9600 baud (see E_MBUS_Baudrate [197]).

bStart: A positive edge of this input triggers one reading of the meter.
bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bReady : BOOL;
bError : BOOL;
eError : E_MBUS_ERROR;
dwIdNumber : DWORD;
byStatus : BYTE;
byGEN : BYTE;
byCounter : BYTE;
usiRecivedAdr : USINT;
eMedium : E_MBUS_Medium;
sMan : STRING(3);
stVolume : ST_MBus_Info;
stFlow : ST_MBus_Info;

bBusy: The bBusy output is TRUE while the meter is being read.

bReady: The bReady output is TRUE for one cycle, once meter reading is completed.

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

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [200]).

sMan: Manufacturer code.

stVolume: Meter reading, water consumption (see ST_MBus_Info [204]).

stFlow: Current water consumption (see ST_MBus_Info [204]).

VAR_IN_OUT

stComIn : ST_KL6781inData22B;
stComOut : ST_KL6781outData22B;
stCom : ST_MBUS_Communication;

stComIn: Process image of the inputs (see ST_KL6781inData22B [202]).

stComOut: Process image of the outputs (see ST_KL6781outData22B [202]).

stCom: This structure is used to link the block with the meter blocks (see ST_MBUS_Communication [202]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.1.14.9  FB_MBUS_RawData

This block is used for reading any M-Bus devices. The variable arrData supplies the raw data of the M-Bus device. Only the first telegram is evaluated.

The block can only be executed together with the block FB_MBUSKL6781() [18].

Functionality of the block [9]

VAR_INPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>usiAddress</td>
<td>Primary address [10] of the meter to be read with this block.</td>
</tr>
<tr>
<td>stSecAdr</td>
<td>Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAdr [205]).</td>
</tr>
<tr>
<td>eBaudrate</td>
<td>300, 600, 1200, 2400, 4800, 9600 baud (see E_MBUS_Baudrate [197]).</td>
</tr>
<tr>
<td>bStart</td>
<td>A positive edge of this input triggers one reading of the meter.</td>
</tr>
<tr>
<td>bSND_NKE</td>
<td>TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).</td>
</tr>
<tr>
<td>bReadInit</td>
<td>If the PLC is restarted, the meter is read once.</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.</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>bReady</td>
<td>BOOL;</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL;</td>
</tr>
<tr>
<td>eError</td>
<td>E_MBUS_ERROR;</td>
</tr>
<tr>
<td>dwIdNumber</td>
<td>DWORD;</td>
</tr>
<tr>
<td>byStatus</td>
<td>BYTE;</td>
</tr>
<tr>
<td>byGEN</td>
<td>BYTE;</td>
</tr>
<tr>
<td>byCounter</td>
<td>BYTE;</td>
</tr>
<tr>
<td>usiRecivedAdr</td>
<td>USINT;</td>
</tr>
<tr>
<td>eMedium</td>
<td>E_MBUS_Medium;</td>
</tr>
</tbody>
</table>
sMan           : STRING[3];
iLen            : INT;
arrData         : ARRAY [0..259] OF BYTE;

iLen: Number of transferred bytes.

arrData: Raw data of the first telegram.

VAR_IN_OUT

stCom : ST_MBUS_Communication;

stCom: This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Communication [202]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.14.10  FB_MBUS_Scan

This block can be used to scan the M-Bus bus. All primary addresses (0..250) are queried successively. The array arrDevice is used to show certain device information.

Only the primary address is used for scanning.

The primary address [10] of all devices must be set.

The block can only be executed together with the block FB_MBUSKL6781() [18].

VAR_INPUT

bStart          : BOOL;
bStop           : BOOL;
eBaudrate       : E_MBUS_Baudrate := eMBUS_Baud2400;
bDisabled       : BOOL := FALSE;

bStart: The search is initiated with a positive edge at this input.

bStop: The search is stopped with a positive edge at this input.

eBaudrate: 300, 600, 1200, 2400, 4800, 9600 Baud.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy           : BOOL;
bReady          : BOOL;
bError          : BOOL;
eError          : E_MBUS_ERROR;
usiAddress      : USINT;
usiCount        : USINT;
arrDevice       : ARRAY [0..250] OF ST_MBus_Scan;
bBusy: The bBusy output is TRUE while the meter is being read.

bReady: The bReady output is TRUE for one cycle, once meter reading is completed.

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

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.

usiAddress: Primary address [10] of the meter to be read with this block.

usiCount: Number of detected valid devices.

arrDevice: Information about the detected devices (see ST_MBus_scan [205]).

VAR_IN_OUT

stCom : ST_MBUS_Communication;

stCom: This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Communication [202]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.15 Gossen Metrawatt overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [76], FB_MBUS_General_Ext [79] or FB_MBUS_General_Param [83] from the folder "General [74]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [85] can be used to send data to the device (e.g. setting of the primary address).

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Type</th>
<th>Device</th>
<th>Block</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gossen Metrawatt</td>
<td>Electricity meter</td>
<td>U128x</td>
<td>FB_MBUS_GMC_Electricity [91]</td>
</tr>
<tr>
<td></td>
<td>Electricity meter</td>
<td>U138x</td>
<td>FB_MBUS_GMC_Electricity [91]</td>
</tr>
</tbody>
</table>
4.1.15.1 FB_MBUS_GMC_Electricity

This block is used to read electricity meters from Gossen Metrawatt:

-U128x

-U138x

The block can only be executed together with the block FB_MBUSKL6781() [18].

Functionality of the block [9]

VAR_INPUT

usiAddress : USINT;
stSecAdr : ST_MBUS_SecAdr;
eBaudrate : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart : BOOL;
bSND_NKE : BOOL := TRUE;
bReadInit : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit : USINT;
bDisabled : BOOL := FALSE;

usiAddress: Primary address [10] of the meter to be read with this block.

stSecAdr: Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAdr [205]).

eBaudrate: 300, 600, 1200, 2400, 4800, 9600 baud (see E_MBUS_Baudrate [197]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard #2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bReady : BOOL;
bError : BOOL;

bBusy: Busy signal.
eError : E_MBUS_ERROR;
dwIdNumber : DWORD;
byStatus : BYTE;
byGEN : BYTE;
byCounter : BYTE;
usiRecivedAdr : USINT;
eMedium : E_MBUS_Medium;
man : STRING(3);
stEnergy : ST_MBus_Info;
stPower : ST_MBus_Info;
stDeviceError : ST_MBus_Info;

bBusy: The bBusy output is TRUE while the meter is being read.

bReady: The bReady output is TRUE for one cycle, once meter reading is completed.

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

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [200]).

sMan: Manufacturer code.

stEnergy: Meter reading, energy consumption (see ST_MBus_Info [204]).

stPower: Current energy consumption, power (see ST_MBus_Info [204]).

stDeviceError: Error message from device (see ST_MBus_Info [204]).

VAR_IN_OUT

stCom : ST_MBUS_Communication;

stCom: This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Communication [202]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.16 GWF overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [76], FB_MBUS_General_Ext [79] or FB_MBUS_General_Param [83] from the folder "General [74]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [85] can be used to send data to the device (e.g. setting of the primary address).
This block is used to read meters from GWF:

- Water meter
- Gas meter S1
- Gas meter Z1

The block can only be executed together with the block \texttt{FB_MBUSKL6781()} \cite{18}.

**Functionality of the block** \cite{9}

**VAR_INPUT**

\begin{verbatim}
usiAddress : USINT;
stSecAdr   : ST_MBUS_SecAdr;
eBaudrate  : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart     : BOOL;
bSND_NKE   : BOOL := TRUE;
bReadInit  : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
bDisabled  : BOOL := FALSE;
\end{verbatim}

\textbf{usiAddress:} Primary address \cite{10} of the meter to be read with this block.

\textbf{stSecAdr:} Secondary address \cite{11} of the meter to be read with this block (see \texttt{ST_MBUS_SecAdr} \cite{205}).

\textbf{eBaudrate:} 300, 2400, 9600 baud (see \texttt{E_MBUS_Baudrate} \cite{197}).

\textbf{bStart:} A positive edge of this input triggers one reading of the meter.

\textbf{bSND_NKE:} TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

\textbf{bReadInit:} If the PLC is restarted, the meter is read once.
tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.

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>bReady</td>
<td>BOOL</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>eError</td>
<td>E_MBUS_ERROR</td>
</tr>
<tr>
<td>dwIdNumber</td>
<td>DWORD</td>
</tr>
<tr>
<td>byStatus</td>
<td>BYTE</td>
</tr>
<tr>
<td>byGEN</td>
<td>BYTE</td>
</tr>
<tr>
<td>byCounter</td>
<td>BYTE</td>
</tr>
<tr>
<td>usiRecivedAdr</td>
<td>USINT</td>
</tr>
<tr>
<td>eMedium</td>
<td>E_MBUS_Medium</td>
</tr>
<tr>
<td>sMan</td>
<td>STRING(3)</td>
</tr>
<tr>
<td>stVolume</td>
<td>ST_MBus_Info</td>
</tr>
</tbody>
</table>

**bBusy:** The bBusy output is TRUE while the meter is being read.

**bReady:** The bReady output is TRUE for one cycle, once meter reading is completed.

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

**eError:** In the event of an error the output issues an error code (see E_MBUS_ERROR). bError goes TRUE at the same time.

**dwIdNumber:** Serial number of the meter (secondary address).

**byStatus:** Device status.

**byGEN:** Software version of the device.

**byCounter:** Number of times the master has accessed data of the respective slave.

**usiRecivedAdr:** Received primary address (0-250).

**eMedium:** Medium (see E_MBUS_Medium).

**sMan:** Manufacturer code.

**stVolume:** Meter reading, water consumption (see ST_MBus_Info).

**VAR_IN_OUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>stCom</td>
<td>ST_MBUS_Communication;</td>
</tr>
</tbody>
</table>

**stCom:** This structure is used to link the block FB_MBUSKL6781() with the meter blocks (see ST_MBUS_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_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.17 Hydrometer overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General, FB_MBUS_General_Ext or FB_MBUS_General_Param from the folder "General" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send can be used to send data to the device (e.g. setting of the primary address).
<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Type</th>
<th>Device</th>
<th>Block</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hydrometer</td>
<td>2 pulse inputs</td>
<td>HYDRO-PORT Pulse</td>
<td>FB_MBUS_HYD_PortPulse</td>
</tr>
<tr>
<td></td>
<td>2 analog inputs + 1 temperature sensor</td>
<td>HYDRO-PORT Analog</td>
<td>FB_MBUS_HYD_PortAnalog</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Water meter</td>
<td>FB_MBUS_HYD_Flypper</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Heat meter</td>
<td>FB_MBUS_HYD_Sharky</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FB_MBUS_HYD_Sharky_00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Heat meter</td>
<td>FB_MBUS_HYD_Sharky</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FB_MBUS_HYD_Sharky_00</td>
</tr>
</tbody>
</table>

4.1.17.1 FB_MBUS_HYD_Flypper

This block is used to read water meters from Hydrometer:
-Flypper

The block can only be executed together with the block FB_MBUSKL6781() [18].

Functionality of the block [9]

VAR_INPUT
usiAddress : USINT;
stSecAdr : ST_MBUS_SecAdr;
eBaudrate : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart : BOOL;
bSND_NKE : BOOL := TRUE;
bReadInit : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
bDisabled : BOOL := FALSE;

usiAddress: Primary address [10] of the meter to be read with this block.

stSecAdr: Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAdr [205]).
eBaudrate: 300, 2400 (see E_MBUS_Baudrate [197]).
**bStart:** A positive edge of this input triggers one reading of the meter.

**bSND_NKE:** TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

**bReadInit:** If the PLC is restarted, the meter is read once.

**tMinSendTime:** Standard #2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.

**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 while the meter is being read.</td>
</tr>
<tr>
<td>bReady</td>
<td>The bReady output is TRUE for one cycle, once meter reading is completed.</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>dwIdNumber</td>
<td>Serial number of the meter (secondary address).</td>
</tr>
<tr>
<td>byStatus</td>
<td>Device status.</td>
</tr>
<tr>
<td>byGEN</td>
<td>Software version of the device.</td>
</tr>
<tr>
<td>byCounter</td>
<td>Number of times the master has accessed data of the respective slave.</td>
</tr>
<tr>
<td>usiRecivedAdr</td>
<td>Received primary address (0-250).</td>
</tr>
<tr>
<td>eMedium</td>
<td>Medium (see E_MBUS_Medium [200]).</td>
</tr>
<tr>
<td>sMan</td>
<td>Manufacturer code.</td>
</tr>
<tr>
<td>stVolume</td>
<td>Meter reading, water consumption (see ST_MBus_Info [204]).</td>
</tr>
<tr>
<td>stFlow</td>
<td>Current water consumption (see ST_MBus_Info [204]).</td>
</tr>
</tbody>
</table>

### VAR_IN_OUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stCom</td>
<td>This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Communication [202]).</td>
</tr>
</tbody>
</table>

### Requirements

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Development environment</td>
<td>required TC3 PLC library</td>
</tr>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.1.17.2 FB_MBUS_HYD_PortAnalog

This block is used for reading energy meters with analog output from Hydrometer:
- HYDRO-PORT analog (2x0/4-20 mA / 1xPT temperature sensor)

The block can only be executed together with the block FB_MBUSKL6781) [18].

Functionality of the block [9]

VAR_INPUT

| siAddress | usINT; |
| stSecAdr | ST_MBUS_SecAdr; |
| eBaudrate | E_MBUS_Baudrate := eMBUS_Baud2400; |
| bStart | BOOL := TRUE; |
| bSND_NKE | BOOL := TRUE; |
| tMinSendTime | TIME := t#2s; |
| usiUnit | USINT; |
| bDisabled | BOOL := FALSE; |

usiAddress: Primary address [10] of the meter to be read with this block.

stSecAdr: Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAdr [205]).

eBaudrate: 300, 2400, 9600 baud (see E_MBUS_Baudrate [197]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

| bBusy | BOOL; |
| bReady | BOOL; |
| bError | BOOL; |
| eError | E_MBUS_ERROR; |
| dwIdNumber | DWORD; |

bBusy:
byStatus : BYTE;
byGEN : BYTE;
byCounter : BYTE;
usiRecivedAdr : USINT;
eMedium : E_MBUS_Medium;
sMan : STRING(3);
stValue1 : ST_MBus_Info;
stValue2 : ST_MBus_Info;
stTemperatur : ST_MBus_Info;

bBusy: The bBusy output is TRUE while the meter is being read.

bReady: The bReady output is TRUE for one cycle, once meter reading is completed.

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

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes
TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [201]).

sMan: Manufacturer code.

stValue1: Meter reading 1 (see ST_MBus_Info [204]).

stValue2: Meter reading 2 (see ST_MBus_Info [204]).

stTemperatur: temperature (see ST_MBus_Info [204]).

VAR_IN_OUT
stCom : ST_MBUS_Communication;

stCom: This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see
ST_MBUS_Communication [202]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.1.17.3  FB_MBUS_HYD_PortPulse

This block is used for reading energy meters with pulse output from Hydrometer:

- HYDRO-PORT Pulse

The block can only be executed together with the block FB_MBUSKL6781() [18].

Functionality of the block [9]

VAR_INPUT

usiAddress : USINT;
stSecAdr : ST_MBUS_SecAdr;
eBaudrate : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart : BOOL;
bSND_NKE : BOOL := TRUE;
bReadInit : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit : USINT;
bDisabled : BOOL := FALSE;

usiAddress: Primary address [10] of the meter to be read with this block.

stSecAdr: Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAdr [205]).

eBaudrate: 300, 2400, 9600 baud (see E_MBUS_Baudrate [197]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2=MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bReady : BOOL;
bError : BOOL;
eError : E_MBUS_ERROR;
dwIdNumber : DWORD;
byStatus : BYTE;
byGEN : BYTE;
byCounter : BYTE;
usiRecivedAdr : USINT;
eMedium : E_MBUS_Medium;
sMan : STRING(3);
stValue1 : ST_MBus_Info;
stValue2 : ST_MBus_Info;

**bBusy**: The *bBusy* output is TRUE while the meter is being read.

**bReady**: The *bReady* output is TRUE for one cycle, once meter reading is completed.

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

**eError**: In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). *bError* goes TRUE at the same time.

**dwIdNumber**: Serial number of the meter (secondary address).

**byStatus**: Device status.

**byGEN**: Software version of the device.

**byCounter**: Number of times the master has accessed data of the respective slave.

**usiRecivedAdr**: Received primary address (0-250).

**eMedium**: Medium (see E_MBUS_Medium [200]).

**sMan**: Manufacturer code.

**stValue1**: Meter reading 1 (see ST_MBus_Info [204]).

**stValue2**: Meter reading 2 (see ST_MBus_Info [204]).

**VAR_IN_OUT**

**stCom**: This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Comunication [202]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.1.17.4 FB_MBUS_HYD_Sharky

This block is used to read energy meters from:

Hydrometer:
- Sharky 773
- Sharky 775
- ENERGY INT 6

Brunata:
- Brunata Optuna H (775)

Aquametro:
- AMNTRONIC SONIC D

Only the most common values (see "VAR_OUTPUT") of the telegrams:

00 (Application Reset-Subcode 00 / All)
10 (Application Reset-Subcode 10 / User data)
20 (Application Reset-Subcode 20 / Simple billing)
30 (Application Reset-Subcode 30 / Enhanced billing)
40 (Application Reset-Subcode 40 / Multi tariff billing)
or 50 (Application Reset-Subcode 50 / Instant values)

are read.

The device is not switched to these telegrams; it must be set to one of these telegrams.

The block FB_MBUS_HYD_Sharky_000 [103] can be used if further data are required, or the block FB_MBUS_General_Send() [85] can be used to select the required telegram, and the block FB_MBUS_General() [76] can be used to read all data of the respective telegram.

The block can only be executed together with the block FB_MBUSKL6781 [18].

Functionality of the block [9]
VAR_INPUT

usiAddress : USINT;
stSecAdr : ST_MBUS_SecAdr;
eBaudrate : E_MBUS_Baudrate := MBUS_Baud2400;
bStart : BOOL;
bSND_NKE : BOOL := TRUE;
bReadInit : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit : USINT;
bDisabled : BOOL := FALSE;

usiAddress: Primary address of the meter to be read with this block.

stSecAdr: Secondary address of the meter to be read with this block (see ST_MBUS_SecAdr).

eBaudrate: 300, 2400 baud (see E_MBUS_Baudrate).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2=MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bReady : BOOL;
bError : BOOL;
eError : E_MBUS_ERROR;
dwIdNumber : DWORD;
byStatus : BYTE;
byGEN : BYTE;
byCounter : BYTE;
usiRecivedAdr : USINT;
eMedium : E_MBUS_Medium;
sMan : STRING[3];
stEnergy : ST_MBus_Info;
stVolume : ST_MBus_Info;
stFlow : ST_MBus_Info;
stForwardTemp : ST_MBus_Info;
stReturnTemp : ST_MBus_Info;
stTariff1 : ST_MBus_Info;

bBusy: The bBusy output is TRUE while the meter is being read.

bReady: The bReady output is TRUE for one cycle, once meter reading is completed.

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

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR). bError goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium).

sMan: Manufacturer code.
stEnergy: Meter reading, energy consumption (see ST_MBus_Info [204]).

stVolume: Meter reading, water consumption (see ST_MBus_Info [204]).

stFlow: Current water consumption (see ST_MBus_Info [204]).

stForwardTemp: Flow temperature (see ST_MBus_Info [204]).

stReturnTemp: Return temperature (see ST_MBus_Info [204]).

stTariff1: Energy consumption tariff 1 (see ST_MBus_Info [204]).

VAR_IN_OUT

stCom : ST_MBUS_Communication;

stCom: This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Communication [202]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.17.5 FB_MBUS_HYD_Sharky_00

This block is used to read energy meters from:
Hydrometer:
- Sharky 773
- Sharky 775
- ENERGY INT 6

Brunata:
- Brunata Optuna H (775)

Aquametro:
- AMNTRONIC SONIC D

All values of telegram 00 (application reset subcode 00 / All) are read. The device automatically switches to the corresponding mode.

stPulsecounter1 and stPulsecounter2 are only output if the pulse module is connected.

If further telegrams are required, the block FB_MBUS_General_Send() can be used to select the required telegram, and the block FB_MBUS_General() can be used to read all data of the respective telegram.

The block can only be executed together with the block FB_MBUSKL6781().

Functionality of the block

VAR_INPUT
usiAddress : USINT;
stSecAdr : ST_MBUS_SecAdr;
eBaudrate : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart : BOOL;
bSND_NKE : BOOL := TRUE;
bReadInit : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit : USINT;
bDisabled : BOOL := FALSE;

usiAddress: Primary address of the meter to be read with this block.

stSecAdr: Secondary address of the meter to be read with this block (see ST_MBUS_SecAdr).

eBaudrate: 300, 2400 baud (see E_MBUS_Baudrate).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT
bBusy : BOOL;
bReady : BOOL;
bError : BOOL;
eError : E_MBUS_ERROR;
dwIdNumber : DWORD;
byStatus : BYTE;
byGEN : BYTE;
byCounter : BYTE;
usiRecivedAdr : USINT;
bBusy: The bBusy output is TRUE while the meter is being read.

bReady: The bReady output is TRUE for one cycle, once meter reading is completed.

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

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [200]).

sMan: Manufacturer code.

stEnergy: Meter reading, energy consumption (see ST_MBus_Info [204]).

stTariff1: Meter reading tariff 1 (see ST_MBus_Info [204]).

stTariff2: Meter reading tariff 2 (see ST_MBus_Info [204]).

stVolume: Meter reading, water consumption (see ST_MBus_Info [204]).

stPower: Current energy consumption, power (see ST_MBus_Info [204]).

stFlow: Current water consumption (see ST_MBus_Info [204]).

stForwardTemp: Flow temperature (see ST_MBus_Info [204]).

stReturnTemp: Return temperature (see ST_MBus_Info [204]).

stDiffTemp: Temperature difference (see ST_MBus_Info [204]).

stOperatingHours: Current operating hours (see ST_MBus_Info [204]).

stDueDate: Current date, time (see ST_MBus_Info [204]).

stDueDay1: Values cutoff date 1 (see ST_MBUS_DueDayHYD1 [206]).

stDueDay2: Values cutoff date 2 (see ST_MBUS_DueDayHYD1 [206]).

stPulsecounter1: Meter reading pulse counter 1 (see ST_MBus_Info [204]).

stPulsecounter2: Meter reading pulse counter 2 (see ST_MBus_Info [204]).
VAR_IN_OUT

stCom : ST_MBUS_Communication;

stCom: This structure is used to link the block FB_MBUSKL6781() \[202\] with the meter blocks (see ST_MBUS_Communication \[202\]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.18 ista overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General \[76\], FB_MBUS_General_Ext \[79\] or FB_MBUS_General_Param \[83\] from the folder "General \[74\]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() \[85\] can be used to send data to the device (e.g. setting of the primary address).

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Type</th>
<th>Device 1</th>
<th>Block</th>
</tr>
</thead>
<tbody>
<tr>
<td>ista</td>
<td>Water meter</td>
<td>domaqua® m</td>
<td>FB_MBUS_IST_Istameter [106]</td>
</tr>
<tr>
<td></td>
<td>Water meter</td>
<td>istameter® m</td>
<td>FB_MBUS_IST_Istameter [106]</td>
</tr>
<tr>
<td></td>
<td>Water meter</td>
<td>istameter III</td>
<td>FB_MBUS_IST_IstameterIII [108]</td>
</tr>
<tr>
<td></td>
<td>Pulse counter</td>
<td>pulsonic II</td>
<td>FB_MBUS_IST_PulsonicII [110]</td>
</tr>
<tr>
<td></td>
<td>Heat meter</td>
<td>sensonic II</td>
<td>FB_MBUS_IST_SensonicII [112]</td>
</tr>
</tbody>
</table>

4.1.18.1 FB_MBUS_IST_Istameter

This block is used to read water meters from Ista:

- istameter® m
- domaqua® m
The block can only be executed together with the block FB_MBUSKl6781() [18].

The devices are supplied from a battery. The number of read operations is therefore limited. An internal meter prevents communication exceeding 96 times per day on average. The user must make sure that excessive queries are prevented.

**Functionality of the block [9]**

**VAR_INPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>usiAddress</td>
<td>USINT</td>
<td>Primary address [10] of the meter to be read with this block.</td>
</tr>
<tr>
<td>stSecAdr</td>
<td>ST_MBUS_SecAdr</td>
<td>Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAdr [205]).</td>
</tr>
<tr>
<td>eBaudrate</td>
<td>E_MBUS_Baudrate</td>
<td>300, 2400 baud (see E_MBUS_Baudrate [197]).</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL</td>
<td>A positive edge of this input triggers one reading of the meter.</td>
</tr>
<tr>
<td>bSND_NKE</td>
<td>BOOL</td>
<td>TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).</td>
</tr>
<tr>
<td>bReadInit</td>
<td>BOOL</td>
<td>If the PLC is restarted, the meter is read once.</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>TIME</td>
<td>Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.</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>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>BOOL</td>
<td>The bBusy output is TRUE while the meter is being read.</td>
</tr>
<tr>
<td>bReady</td>
<td>BOOL</td>
<td>The bReady output is TRUE for one cycle, once meter reading is completed.</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_MBUS_ERROR</td>
<td>In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.</td>
</tr>
<tr>
<td>dwlIdNumber</td>
<td>DWORD</td>
<td>Serial number of the meter (secondary address).</td>
</tr>
<tr>
<td>byStatus</td>
<td>BYTE</td>
<td>Device status.</td>
</tr>
<tr>
<td>byGEN</td>
<td>BYTE</td>
<td>Software version of the device.</td>
</tr>
<tr>
<td>byCounter</td>
<td>BYTE</td>
<td>Number of times the master has accessed data of the respective slave.</td>
</tr>
<tr>
<td>usiRecivedAdr</td>
<td>USINT</td>
<td>Received primary address (0-250).</td>
</tr>
<tr>
<td>eMedium</td>
<td>E_MBUS_Medium</td>
<td></td>
</tr>
<tr>
<td>sMan</td>
<td>STRING(3)</td>
<td></td>
</tr>
<tr>
<td>stVolume</td>
<td>ST_MBus_Info</td>
<td></td>
</tr>
<tr>
<td>stDeviceError</td>
<td>ST_MBus_Info</td>
<td></td>
</tr>
</tbody>
</table>
Programming

\textbf{eMedium:} Medium (see \texttt{E\_MBUS\_Medium})

\textbf{sMan:} Manufacturer code.

\textbf{stVolume:} Meter reading, water consumption (see \texttt{ST\_MBus\_Info}).

\textbf{stDeviceError:} Error message from device (see \texttt{ST\_MBus\_Info}).

\textbf{VAR\_IN\_OUT}

\texttt{stCom} : \texttt{ST\_MBUS\_Communication};

\texttt{stCom:} This structure is used to link the block \texttt{FB\_MBUSKL6781()} with the meter blocks (see \texttt{ST\_MBUS\_Communication}).

\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_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.18.2 \texttt{FB\_MBUS\_IST\_IstameterIll}

\begin{verbatim}
VAR\_INPUT
usiAddress : USINT;
stSecAdr : ST\_MBUS\_SecAdr;
eBaudrate : E\_MBUS\_Baudrate := eMBUS\_Baud2400;
bStart : BOOL;
bSND\_NKE : BOOL := TRUE;
bReadInit : BOOL := TRUE;
tMinSendTime : TIME := t\#2s;
bDisabled : BOOL := FALSE;

eMedium:

sMan:

stVolume:

stDeviceError:
\end{verbatim}

This block is used to read water meters from Ista:
- istameter III

The block can only be executed together with the block \texttt{FB\_MBUSKL6781()}.

The devices are supplied from a battery. The number of read operations is therefore limited. An internal meter prevents communication exceeding 96 times per day on average. The user must make sure that excessive queries are prevented.

\textbf{Functionality of the block}

\textbf{VAR\_INPUT}

\begin{verbatim}
usiAddress : USINT;
stSecAdr : ST\_MBUS\_SecAdr;
eBaudrate : E\_MBUS\_Baudrate := eMBUS\_Baud2400;
bStart : BOOL;
bSND\_NKE : BOOL := TRUE;
bReadInit : BOOL := TRUE;
tMinSendTime : TIME := t\#2s;
bDisabled : BOOL := FALSE;
\end{verbatim}
usiAddress: Primary address [10] of the meter to be read with this block.

stSecAdr: Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAdr [205]).

eBaudrate: 300, 2400 baud (see E_MBUS_Baudrate [197]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bReady : BOOL;
bError : BOOL;
eError : E_MBUS_ERROR;
dwIdNumber : DWORD;
byStatus : BYTE;
byGEN : BYTE;
byCounter : BYTE;
usiRecivedAdr : USINT;
eMedium : E_MBUS_Medium;
sMan : STRING(3);
stVolume : ST_MBus_Info;
stFlow : ST_MBus_Info;
stDeviceError : ST_MBus_Info;

bBusy: The bBusy output is TRUE while the meter is being read.

bReady: The bReady output is TRUE for one cycle, once meter reading is completed.

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

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [201]).

sMan: Manufacturer code.

stVolume: Meter reading, water consumption (see ST_MBus_Info [204]).

stFlow: Current water consumption (see ST_MBus_Info [204]).

stDeviceError: Error message from device (see ST_MBus_Info [204]).

VAR_IN_OUT

stCom : ST_MBUS_Communication;

stCom: This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Communication [202]).
Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.18.3 FB_MBUS IST_PulsonicII

This block is used to read energy meters with pulse output from Ista:

-Pulsonic II

The block can only be executed together with the block FB_MBUSKL6781() [18].

Maximum number of readings

The devices are supplied from a battery. The number of read operations is therefore limited. An internal counter prevents communication exceeding 96 times per day on average. The user must make sure that excessive queries are prevented.

Functionality of the block [9]

VAR_INPUT

- `usiAddress`: Primary address [10] of the meter to be read with this block.
- `stSecAdr`: Secondary address [11] of the meter to be read with this block (see `ST_MBUS_SecAdr [205]`).
- `eBaudrate`: 300, 2400 baud (see `E_MBUS_Baudrate [197]`).
- `bStart`: A positive edge of this input triggers one reading of the meter.
- `bSND_NKE`: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).
- `bReadInit`: If the PLC is restarted, the meter is read once.
**tMinSendTime:** Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.

**usiUnit:** Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

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

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>bBusy</th>
<th>: BOOL;</th>
</tr>
</thead>
<tbody>
<tr>
<td>bReady</td>
<td>: BOOL;</td>
</tr>
<tr>
<td>bError</td>
<td>: BOOL;</td>
</tr>
<tr>
<td>eError</td>
<td>: E_MBUS_ERROR;</td>
</tr>
<tr>
<td>dwIdNumber</td>
<td>: DWORD;</td>
</tr>
<tr>
<td>byStatus</td>
<td>: BYTE;</td>
</tr>
<tr>
<td>byGEN</td>
<td>: BYTE;</td>
</tr>
<tr>
<td>byCounter</td>
<td>: BYTE;</td>
</tr>
<tr>
<td>usiRecivedAddr</td>
<td>: USINT;</td>
</tr>
<tr>
<td>eMedium</td>
<td>: E_MBUS_Medium;</td>
</tr>
<tr>
<td>sMan</td>
<td>: STRING(3);</td>
</tr>
<tr>
<td>stValue</td>
<td>: ST_MBus_Info;</td>
</tr>
<tr>
<td>stCurrentValue</td>
<td>: ST_MBus_Info;</td>
</tr>
<tr>
<td>stDeviceError</td>
<td>: ST_MBus_Info;</td>
</tr>
</tbody>
</table>

**bBusy:** The bBusy output is TRUE while the meter is being read.

**bReady:** The bReady output is TRUE for one cycle, once meter reading is completed.

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

**eError:** In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.

**dwIdNumber:** Serial number of the meter (secondary address).

**byStatus:** Device status.

**byGEN:** Software version of the device.

**byCounter:** Number of times the master has accessed data of the respective slave.

**usiRecivedAddr:** Received primary address (0-250).

**eMedium:** Medium (see E_MBUS_Medium [200]).

**sMan:** Manufacturer code.

**stValue:** Current consumption value (see ST_MBus_Info [204]).

**stCurrentValue:** Current flow rate / power (see ST_MBus_Info [204]).

**stDeviceError:** Error message from device (see ST_MBus_Info [204]).

**VAR_IN_OUT**

| stCom      | : ST_MBUS_Communication; |

**stCom:** This structure is used to link the block FB_MBUSKL6781 () [202] with the meter blocks (see ST_MBUS_Communication [202]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>
This block is used to read heat meters from Ista:

-Sensonic II

The block can only be executed together with the block FB_MBUS_KL6781 [18].

Maximum number of readings

The devices are supplied from a battery. The number of read operations is therefore limited. An internal counter prevents communication exceeding 96 times per day on average. The user must make sure that excessive queries are prevented.

Functionality of the block [9]

VAR_INPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>usiAddress</td>
<td>Primary address [10] of the meter to be read with this block.</td>
</tr>
<tr>
<td>stSecAdr</td>
<td>Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAdr [205]).</td>
</tr>
<tr>
<td>eBaudrate</td>
<td>300, 2400 baud (see E_MBUS_Baudrate [197]).</td>
</tr>
<tr>
<td>bStart</td>
<td>A positive edge of this input triggers one reading of the meter.</td>
</tr>
<tr>
<td>bSND_NKE</td>
<td>TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).</td>
</tr>
<tr>
<td>bReadInit</td>
<td>If the PLC is restarted, the meter is read once.</td>
</tr>
</tbody>
</table>

This block is used to read heat meters from Ista:

-Sensonic II

The block can only be executed together with the block FB_MBUS_KL6781 [18].

Maximum number of readings

The devices are supplied from a battery. The number of read operations is therefore limited. An internal counter prevents communication exceeding 96 times per day on average. The user must make sure that excessive queries are prevented.

Functionality of the block [9]

VAR_INPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>usiAddress</td>
<td>Primary address [10] of the meter to be read with this block.</td>
</tr>
<tr>
<td>stSecAdr</td>
<td>Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAdr [205]).</td>
</tr>
<tr>
<td>eBaudrate</td>
<td>300, 2400 baud (see E_MBUS_Baudrate [197]).</td>
</tr>
<tr>
<td>bStart</td>
<td>A positive edge of this input triggers one reading of the meter.</td>
</tr>
<tr>
<td>bSND_NKE</td>
<td>TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).</td>
</tr>
<tr>
<td>bReadInit</td>
<td>If the PLC is restarted, the meter is read once.</td>
</tr>
</tbody>
</table>
tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2=MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bReady : BOOL;
bError : BOOL;
eError : E_MBUS_ERROR;
dwIdNumber : DWORD;
byStatus : BYTE;
byGEN : BYTE;
byCounter : BYTE;
usiRecivedAdr : UINT;
eMedium : E_MBUS_Medium;
sMan : STRING(3);
stEnergy : ST_MBus_Info;
stColdEnergy : ST_MBus_Info;
stPower : ST_MBus_Info;
stVolume : ST_MBus_Info;
stFlow : ST_MBus_Info;
stForwardTemp : ST_MBus_Info;
stReturnTemp : ST_MBus_Info;
stDiffTemp : ST_MBus_Info;
stDeviceError : ST_MBus_Info;

bBusy: The bBusy output is TRUE while the meter is being read.

bReady: The bReady output is TRUE for one cycle, once meter reading is completed.

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

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [200]).

sMan: Manufacturer code.

stEnergy: Meter reading, energy consumption (see ST_MBus_Info [204]).

stColdEnergy: Meter reading, cooling energy consumption (see ST_MBus_Info [204]).

stPower: Current energy consumption, power (see ST_MBus_Info [204]).

stVolume: Meter reading, water consumption (see ST_MBus_Info [204]).

stFlow: Current water consumption (see ST_MBus_Info [204]).

stForwardTemp: Flow temperature (see ST_MBus_Info [204]).

stReturnTemp: Return temperature (see ST_MBus_Info [204]).

stDiffTemp: Temperature difference (see ST_MBus_Info [204]).

stDeviceError: Error message from device (see ST_MBus_Info [204]).
VAR_IN_OUT

stCom : ST_MBUS_Communication;

stCom: This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Communication [202]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.19 Itron

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [76], FB_MBUS_General_Ext [79] or FB_MBUS_General_Param [83] from the folder "General [74]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [85] can be used to send data to the device (e.g. setting of the primary address).

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Type</th>
<th>Device</th>
<th>Function block</th>
</tr>
</thead>
<tbody>
<tr>
<td>Itron</td>
<td>Energy meter</td>
<td>Integral-V UltraLite</td>
<td>FB_MBUS_ITR_IntegralVUltraLite [114]</td>
</tr>
</tbody>
</table>

4.1.19.1 FB_MBUS_ITR_IntegralVUltraLite

This function block is used to read energy meters from Itron.

The function block can only be executed together with the function block FB_MBUSKL6781() [18].

Functionality of the function block [9]
### VAR_INPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>usiAddress</td>
<td>Primary address of the meter to be read with this block.</td>
</tr>
<tr>
<td>stSecAdr</td>
<td>Secondary address of the meter to be read with this block (see ST_MBUS_SecAdr).</td>
</tr>
<tr>
<td>eBaudrate</td>
<td>300, 2400 baud (see E_MBUS_Baudrate).</td>
</tr>
<tr>
<td>bStart</td>
<td>A positive edge of this input triggers one reading of the meter.</td>
</tr>
<tr>
<td>bSND_NKE</td>
<td>TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).</td>
</tr>
<tr>
<td>bReadInit</td>
<td>If the PLC is restarted, the meter is read once.</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>Standard t#2s. The meter is read again, once the time set here has elapsed.</td>
</tr>
<tr>
<td>bDisabled</td>
<td>TRUE = deselection of the function 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 while the meter is being read.</td>
</tr>
<tr>
<td>bReady</td>
<td>The bReady output is TRUE for one cycle, once meter reading is completed.</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>In the event of an error the output issues an error code (see E_MBUS_ERROR).</td>
</tr>
<tr>
<td>dwIdNumber</td>
<td>Serial number of the meter (secondary address).</td>
</tr>
<tr>
<td>byStatus</td>
<td>Device status.</td>
</tr>
<tr>
<td>byGEN</td>
<td>Software version of the device.</td>
</tr>
<tr>
<td>byCounter</td>
<td>Number of times the master has accessed data of the respective slave.</td>
</tr>
<tr>
<td>usiRecivedAdr</td>
<td>Received primary address (0-250).</td>
</tr>
<tr>
<td>eMedium</td>
<td>Medium (see E_MBUS_Medium).</td>
</tr>
<tr>
<td>sMan</td>
<td>Manufacturer code.</td>
</tr>
</tbody>
</table>

Primary address [10] of the meter to be read with this block.
Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAdr [205]).
300, 2400 baud (see E_MBUS_Baudrate [197]).
A positive edge of this input triggers one reading of the meter.
If the PLC is restarted, the meter is read once.
Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.
TRUE = deselection of the function block.

The bBusy output is TRUE while the meter is being read.
The bReady output is TRUE for one cycle, once meter reading is completed.
this output goes TRUE as soon as an error occurs. This error is described via the variable eError.
In the event of an error the output issues an error code (see E_MBUS_ERROR). bError goes TRUE at the same time.
Serial number of the meter (secondary address).
Device status.
Software version of the device.
Number of times the master has accessed data of the respective slave.
Received primary address (0-250).
Medium (see E_MBUS_Medium).
Manufacturer code.
stEnergy: Current energy (see ST_MBus_Info [204]).

stVolume: Current volume (see ST_MBus_Info [204]).

stPower: Current output (see ST_MBus_Info [204]).

stFlow: Current flow rate (see ST_MBus_Info [204]).

stTempFlow: Current flow temperature (see ST_MBus_Info [204]).

stTempReturn: Current return temperature (see ST_MBus_Info [204]).

stTempDifference: Current temperature difference (see ST_MBus_Info [204]).

stDateTime: Date and time (see ST_MBus_Info [204]).

stOperatingTime: Operating time (see ST_MBus_Info [204]).

VAR_IN_OUT

stCom : ST_MBUS_Communication;

stCom: This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Communication [202]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.32</td>
<td>Tc2_MBus from 3.4.6.0</td>
</tr>
</tbody>
</table>

4.1.20 Janitza overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [76], FB_MBUS_General_Ext [79] or FB_MBUS_General_Param [83] from the folder "General [74]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [85] can be used to send data to the device (e.g. setting of the primary address).

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Type</th>
<th>Device</th>
<th>Block</th>
</tr>
</thead>
<tbody>
<tr>
<td>Janitza</td>
<td>Electricity meter</td>
<td>UMG96S</td>
<td>FB_MBUS_JAN_UMG96S [117]</td>
</tr>
</tbody>
</table>
This block is used to read electricity meters from Janitza:

-UMG96S

The block can only be executed together with the block FB_MBUSKL6781.[18]

**Functionality of the block[9]**

**VAR_INPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>usiAddress</td>
<td>USINT;</td>
</tr>
<tr>
<td>stSecAdr</td>
<td>ST_MBUS_SecAdr;</td>
</tr>
<tr>
<td>eBaudrate</td>
<td>E_MBUS_Baudrate := eMBUS_Baud2400;</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL;</td>
</tr>
<tr>
<td>bSND_NKE</td>
<td>BOOL := TRUE;</td>
</tr>
<tr>
<td>bReadInit</td>
<td>BOOL := TRUE;</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>TIME := t#2s;</td>
</tr>
<tr>
<td>usiUnit</td>
<td>USINT;</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL := FALSE;</td>
</tr>
</tbody>
</table>

**usiAddress:** Primary address[10] of the meter to be read with this block.

**stSecAdr:** Secondary address[11] of the meter to be read with this block (see ST_MBUS_SecAdr[205]).
**eBaudrate:** 300, 2400, 9600 baud (see __E_MBUS_Baudrate [197]).

**bStart:** A positive edge of this input triggers one reading of the meter.

**bSND_NKE:** TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

**bReadInit:** If the PLC is restarted, the meter is read once.

**tMinSendTime:** Standard #2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with __bStart__.

**usiUnit:** Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2=MW(h) / 3=GW(h).

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

**VAR_OUTPUT**

```plaintext
bBusy : BOOL;
bReady : BOOL;
bError : BOOL;
eError : E_MBUS_ERROR;
dwIdNumber : DWORD;
byStatus : BYTE;
byGEN : BYTE;
byCounter : BYTE;
usiRecivedAdr : UINT;
eMedium : E_MBUS_Medium;
msMan : STRING(3);
stActiveEnergy : ST_MBus_Info;
stActiveEnergyTariff1 : ST_MBus_Info;
stActiveEnergyTariff2 : ST_MBus_Info;
stReactiveEnergy : ST_MBus_Info;
stReactiveEnergyTariff1 : ST_MBus_Info;
stReactiveEnergyTariff2 : ST_MBus_Info;
stApparentEnergy : ST_MBus_Info;
stActivePower : ST_MBus_Info;
stActivePowerL1 : ST_MBus_Info;
stActivePowerL2 : ST_MBus_Info;
stActivePowerL3 : ST_MBus_Info;
stApparentPower : ST_MBus_Info;
stApparentPowerL1 : ST_MBus_Info;
stApparentPowerL2 : ST_MBus_Info;
stApparentPowerL3 : ST_MBus_Info;
stCurrent : ST_MBus_Info;
stCurrentL1 : ST_MBus_Info;
stCurrentL2 : ST_MBus_Info;
stCurrentL3 : ST_MBus_Info;
stVoltageL1 : ST_MBus_Info;
stVoltageL2 : ST_MBus_Info;
stVoltageL3 : ST_MBus_Info;
```

**stActiveEnergy:** Active energy without backstop (telegram 2, data point 14) (see __ST_MBus_Info [204]).

**stActiveEnergyTariff1:** Active energy, relative (telegram 2, data point 15) (see __ST_MBus_Info [204]).

**stActiveEnergyTariff2:** Active energy, supplied (telegram 2, data point 16) (see __ST_MBus_Info [204]).

**stReactiveEnergy:** Reactive energy, inductive (telegram 2, data point 17) (see __ST_MBus_Info [204]).

**stReactiveEnergyTariff1:** Reactive energy, capacitive (telegram 2, data point 18) (see __ST_MBus_Info [204]).

**stReactiveEnergyTariff2:** Reactive energy, without backstop (telegram 2, data point 19) (see __ST_MBus_Info [204]).

**stApparentEnergy:** Apparent energy (telegram 2, data point 20) (see __ST_MBus_Info [204]).

**stActivePower:** Instantaneous consumption, power, total (telegram 2, data point 29) (see __ST_MBus_Info [204]).

**stActivePowerL1:** Instantaneous consumption, power, phase L1 (telegram 2, data point 38) (see __ST_MBus_Info [204]).
stActivePowerL2: Instantaneous consumption, power, phase L2 (telegram 2, data point 39) (see ST_MBus_Info [204]).

stActivePowerL3: Instantaneous consumption, power, phase L3 (telegram 2, data point 40) (see ST_MBus_Info [204]).

stReactivePower: Reactive power (telegram 2, data point 30) (see ST_MBus_Info [204]).

stApparentPower: Apparent power (telegram 2, data point 31) (see ST_MBus_Info [204]).

stCurrent: Current (telegram 2, data point 28) (see ST_MBus_Info [204]).

stCurrentL1: Current L1 (telegram 2, data point 35) (see ST_MBus_Info [204]).

stCurrentL2: Current L2 (telegram 2, data point 36) (see ST_MBus_Info [204]).

stCurrentL3: Current L3 (telegram 2, data point 37) (see ST_MBus_Info [204]).

stVoltageL1: Voltage L1 (telegram 2, data point 32) (see ST_MBus_Info [204]).

stVoltageL2: Voltage L2 (telegram 2, data point 33) (see ST_MBus_Info [204]).

stVoltageL3: Voltage L3 (telegram 2, data point 34) (see ST_MBus_Info [204]).

VAR_IN_OUT

stCom : ST_MBUS_Communication;

stCom: This structure is used to link the block FB_MBUS_KL6781() [202] with the meter blocks (see ST_MBUS_Communication [202]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.21 Kamstrup overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [76], FB_MBUS_General_Ext [79] or FB_MBUS_General_Param [83] from the folder "General [74]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [85] can be used to send data to the device (e.g. setting of the primary address).

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Type</th>
<th>Device</th>
<th>Block</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kamstrup</td>
<td>Electricity</td>
<td>Kamstrup 162</td>
<td>FB_MBUS_KAM_KamstrupE [120]</td>
</tr>
<tr>
<td></td>
<td>Electricity</td>
<td>Kamstrup 351</td>
<td>FB_MBUS_KAM_KamstrupE [120]</td>
</tr>
<tr>
<td></td>
<td>Electricity</td>
<td>Kamstrup 382</td>
<td>FB_MBUS_KAM_KamstrupE [120]</td>
</tr>
<tr>
<td></td>
<td>Heat/cold</td>
<td>Maxical III</td>
<td>FB_MBUS_KAM_Maxical_III [122]</td>
</tr>
<tr>
<td></td>
<td>Heat/cold</td>
<td>Multical 401</td>
<td>FB_MBUS_KAM_Multical [124]</td>
</tr>
<tr>
<td></td>
<td>Heat/cold</td>
<td>Multical 402</td>
<td>FB_MBUS_KAM_Multical402 [126]</td>
</tr>
<tr>
<td></td>
<td>Water</td>
<td>Multical 41</td>
<td>FB_MBUS_KAM_Multical41 [128]</td>
</tr>
<tr>
<td></td>
<td>Heat/cold</td>
<td>Multical 601</td>
<td>FB_MBUS_KAM_Multical601 [130]</td>
</tr>
</tbody>
</table>
This block is used to read electricity meters from Kamstrup:

-Kamstrup 162
-Kamstrup 351
-Kamstrup 382

The block can only be executed together with the block `FB_MBUSKL6781`) [⬦ 18].

**Functionality of the block [⬦ 9]**

**VAR_INPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>usiAddress</td>
<td>Primary address [⬦ 10] of the meter to be read with this block.</td>
</tr>
<tr>
<td>stSecAdr</td>
<td>Secondary address [⬦ 11] of the meter to be read with this block (see <code>ST_MBUS_SecAdr [⬦ 205]</code>).</td>
</tr>
<tr>
<td>eBaudrate</td>
<td>300, 2400 baud (see <code>E_MBUS_Baudrate [⬦ 197]</code>).</td>
</tr>
<tr>
<td>bStart</td>
<td>A positive edge of this input triggers one reading of the meter.</td>
</tr>
<tr>
<td>bSND_NKE</td>
<td>TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).</td>
</tr>
<tr>
<td>bReadInit</td>
<td>If the PLC is restarted, the meter is read once.</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with <code>bStart</code>.</td>
</tr>
<tr>
<td>usiUnit</td>
<td>Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).</td>
</tr>
<tr>
<td>bDisabled</td>
<td>TRUE = deselection of the block.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Input</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>usiAddress</td>
<td>Primary address [⬦ 10] of the meter to be read with this block.</td>
</tr>
<tr>
<td>eBaudrate</td>
<td>300, 2400 baud (see <code>E_MBUS_Baudrate [⬦ 197]</code>).</td>
</tr>
<tr>
<td>bStart</td>
<td>A positive edge of this input triggers one reading of the meter.</td>
</tr>
<tr>
<td>bSND_NKE</td>
<td>TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).</td>
</tr>
<tr>
<td>bReadInit</td>
<td>If the PLC is restarted, the meter is read once.</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with <code>bStart</code>.</td>
</tr>
<tr>
<td>usiUnit</td>
<td>Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).</td>
</tr>
<tr>
<td>bDisabled</td>
<td>TRUE = deselection of the block.</td>
</tr>
</tbody>
</table>
VAR_OUTPUT

bBusy   : BOOL;
bReady  : BOOL;
bError  : BOOL;
eError  : E_MBUS_ERROR;
dwIdNumber : DWORD;
byStatus : BYTE;
byGEN    : BYTE;
byCounter: BYTE;
usiRecivedAdr : USINT;
eMedium : E_MBUS_Medium;
sMan     : STRING(3);
stEnergy : ST_MBus_Info;
stPower  : ST_MBus_Info;

bBusy: The bBusy output is TRUE while the meter is being read.

bReady: The bReady output is TRUE for one cycle, once meter reading is completed.

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

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [200]).

sMan: Manufacturer code.

stEnergy: Meter reading, energy consumption (see ST_MBus_Info [204]).

stPower: Current energy consumption, power (see ST_MBus_Info [204]).

VAR_IN_OUT

stCom : ST_MBUS_Communication;

stCom: This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Communication [202]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>
This block is used for reading heat/cold meters from Kamstrup:

- Maxical III

The block can only be executed together with the block FB_MBUSKL6781() \[18\].

Functionality of the block \[9\]

**VAR_INPUT**

<table>
<thead>
<tr>
<th>usiAddress     : USINT;</th>
<th>bBusy            : BOOL;</th>
</tr>
</thead>
<tbody>
<tr>
<td>stSecAdr       : ST_MBUS_SecAdr;</td>
<td>bReady            : BOOL;</td>
</tr>
<tr>
<td>eBaudrate      : E_MBUS_Baudrate := eMBUS_Baud2400;</td>
<td>bError            : BOOL;</td>
</tr>
<tr>
<td>bStart         : BOOL := TRUE;</td>
<td>nBaudrate         : E_MBUS_Baudrate := eMBUS_Baud2400;</td>
</tr>
<tr>
<td>bSND_NKE       : BOOL := TRUE;</td>
<td>dwIdNumber        : DWORD;</td>
</tr>
<tr>
<td>bReadInit      : BOOL := TRUE;</td>
<td>byStatus          : BYTE;</td>
</tr>
<tr>
<td>tMinSendTime   : TIME := t#2s;</td>
<td>byGen             : BYTE;</td>
</tr>
<tr>
<td>usiUnit        : USINT;</td>
<td>sMan              : STRING;</td>
</tr>
<tr>
<td>bDisabled      : BOOL := FALSE;</td>
<td>sEnergy           : STRING;</td>
</tr>
<tr>
<td>stCom          : eMedium;</td>
<td>stPower           : STRING;</td>
</tr>
<tr>
<td>stUnit         : USINT;</td>
<td>stVolume          : STRING;</td>
</tr>
<tr>
<td></td>
<td>stFlow            : STRING;</td>
</tr>
</tbody>
</table>

usiAddress: Primary address \[10\] of the meter to be read with this block.

stSecAdr: Secondary address \[11\] of the meter to be read with this block (see ST_MBUS_SecAdr \[205\]).

eBaudrate: 300, 2400 baud (see E_MBUS_Baudrate \[197\]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2=MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.
VAR_OUTPUT

bBusy : BOOL;
bReady : BOOL;
bError : BOOL;
eError : E_MBUS_ERROR;
dwIdNumber : DWORD;
byStatus : BYTE;
byGEN : BYTE;
byCounter : BYTE;
usiRecivedAddr : USINT;
eMedium : E_MBUS_Medium;
sMan : STRING(3);
stEnergy : ST_MBus_Info;
stPower : ST_MBus_Info;
stVolume : ST_MBus_Info;
stFlow : ST_MBus_Info;
stForwardTemp : ST_MBus_Info;
stReturnTemp : ST_MBus_Info;
stDiffTemp : ST_MBus_Info;

bBusy: The bBusy output is TRUE while the meter is being read.

bReady: The bReady output is TRUE for one cycle, once meter reading is completed.

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

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAddr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [200]).

sMan: Manufacturer code.

stEnergy: Meter reading, energy consumption (see ST_MBus_Info [204]).

stPower: Current energy consumption, power (see ST_MBus_Info [204]).

stVolume: Meter reading, water consumption (see ST_MBus_Info [204]).

stFlow: Current water consumption (see ST_MBus_Info [204]).

stForwardTemp: Flow temperature (see ST_MBus_Info [204]).

stReturnTemp: Return temperature (see ST_MBus_Info [204]).

stDiffTemp: Temperature difference (see ST_MBus_Info [204]).

VAR_IN_OUT

stCom : ST_MBUS_Communication;

stCom: This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Communication [202]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.1.21.3 FB_MBUS_KAM_Multical

This block is used for reading heat/cold meters from Kamstrup:

- Multical 401
- Multical 601

The block can only be executed together with the block FB_MBUSKL6781() [18].

Functionality of the block [9]

**VAR_INPUT**

```plaintext
usiAddress : USINT;
stSecAdr : ST_MBUS_SecAdr;
eBaudrate : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart : BOOL;
bSND_NKE : BOOL := TRUE;
bReadInit : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit : USINT;
bDisabled : BOOL := FALSE;
```

**usiAddress**: Primary address [10] of the meter to be read with this block.

**stSecAdr**: Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAdr [205]).

**eBaudrate**: 300, 2400 baud (see E_MBUS_Baudrate [197]).

**bStart**: A positive edge of this input triggers one reading of the meter.

**bSND_NKE**: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

**bReadInit**: If the PLC is restarted, the meter is read once.

**tMinSendTime**: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.

**usiUnit**: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

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

bBusy : BOOL;
bReady : BOOL;
bError : BOOL;
eError : E_MBUS_ERROR;
dwIdNumber : DWORD;
byStatus : BYTE;
byGEN : BYTE;
byCounter : BYTE;
usiRecivedAdr : USINT;
eMedium : E_MBUS_Medium;
eMan : STRING(3);
stEnergy : ST_MBus_Info;
stPower : ST_MBus_Info;
stVolume : ST_MBus_Info;
stFlow : ST_MBus_Info;
stForwardTemp : ST_MBus_Info;
stReturnTemp : ST_MBus_Info;
stDiffTemp : ST_MBus_Info;

VAR_IN_OUT

stCom : ST_MBUS_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_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>
This block is used to read energy meters from Kamstrup.

The block can only be executed together with the block `FB_MBUSKL6781` [18].

Functionality of the block [9]

**VAR_INPUT**

```
usiAddress : USINT;
stSecAdr : ST_MBUS_SecAdr;
eBaudrate : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart : BOOL;
bSND_NKE : BOOL := TRUE;
bReadInit : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
bDisabled : BOOL := FALSE;
```

**usiAddress**: Primary address [10] of the meter to be read with this block.

**stSecAdr**: Secondary address [11] of the meter to be read with this block (see `ST_MBUS_SecAdr` [205]).

**eBaudrate**: 300, 2400, 9600 baud (see `E_MBUS_Baudrate` [197]).

**bStart**: A positive edge of this input triggers one reading of the meter.

**bSND_NKE**: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).
bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT
bBusy : BOOL;
bReady : BOOL;
eError : E_MBUS_ERROR;
dwIdNumber : DWORD;
byStatus : BYTE;
byGEN : BYTE;
usiRecvAdr : USINT;
eMedium : E_MBUS_Medium;
byStatus : BYTE;
stEnergyHeating : ST_MBus_Info;
stVolume : ST_MBus_Info;
stOnTime : ST_MBus_Info;
stTempFlow : ST_MBus_Info;
stTempReturn : ST_MBus_Info;
stTempDiff : ST_MBus_Info;
stPowerActual : ST_MBus_Info;
stPowerMax : ST_MBus_Info;
stFlowActual : ST_MBus_Info;
stFlowMax : ST_MBus_Info;
stTariff2 : ST_MBus_Info;
stTariff3 : ST_MBus_Info;
stPulseInputA : ST_MBus_Info;
stPulseInputB : ST_MBus_Info;
stEnergyCooling : ST_MBus_Info;
stDateTime : ST_MBus_Info;
stEnergyHeatingTarget : ST_MBus_Info;
stVolumeTarget : ST_MBus_Info;
stPowerMaxTarget : ST_MBus_Info;
stFlowMaxTarget : ST_MBus_Info;
stTariff2Target : ST_MBus_Info;
stTariff3Target : ST_MBus_Info;
stPulseInputATarget : ST_MBus_Info;
stPulseInputBTarget : ST_MBus_Info;
stEnergyCoolingTarget : ST_MBus_Info;
stDateTarget : ST_MBus_Info;

stEnergyHeating: Heat energy (see ST_MBus_Info [204]).
stVolume: Water consumption from district heating (see ST_MBus_Info [204]).
stOnTime: Operating hours (see ST_MBus_Info [204]).
stTempFlow: Current flow temperature (see ST_MBus_Info [204]).
stTempReturn: Current return temperature (see ST_MBus_Info [204]).
stTempDiff: Current temperature difference (see ST_MBus_Info [204]).
stPowerActual: Current effective power (see ST_MBus_Info [204]).
stPowerMax: Effective power (max.) (see ST_MBus_Info [204]).
stFlowActual: Current flow rate (see ST_MBus_Info [204]).
stFlowMax: Flow rate (max.) (see ST_MBus_Info [204]).
stTariff2: Tariff 2 (see ST_MBus_Info [204]).
stTariff3: Tariff 3 (see ST_MBus_Info [204]).
**stPulseInputA:** Pulse input A (see [ST_MBus_Info](#) [204]).

**stPulseInputB:** Pulse input B (see [ST_MBus_Info](#) [204]).

**stEnergyCooling:** Cooling energy (see [ST_MBus_Info](#) [204]).

**stDateTime:** Date and time (see [ST_MBus_Info](#) [204]).

**stEnergyHeatingTarget:** See manufacturer information (see [ST_MBus_Info](#) [204]).

**stVolumeTarget:** See manufacturer information (see [ST_MBus_Info](#) [204]).

**stPowerMaxTarget:** See manufacturer information (see [ST_MBus_Info](#) [204]).

**stFlowMaxTarget:** See manufacturer information (see [ST_MBus_Info](#) [204]).

**stTariff2Target:** See manufacturer information (see [ST_MBus_Info](#) [204]).

**stTariff3Target:** See manufacturer information (see [ST_MBus_Info](#) [204]).

**stPulseInputATarget:** See manufacturer information (see [ST_MBus_Info](#) [204]).

**stPulseInputBTarget:** See manufacturer information (see [ST_MBus_Info](#) [204]).

**stEnergyCoolingTarget:** See manufacturer information (see [ST_MBus_Info](#) [204]).

**stDateTarget:** See manufacturer information (see [ST_MBus_Info](#) [204]).

**VAR_IN_OUT**

```plaintext
stCom : ST_MBUS_Communication;
```

**stCom:** This structure is used to link the block `FB_MBUSKAM_Multical41` (202) with the meter blocks (see [ST_MBUS_Communication](#) [202]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.21.5 FB_MBUS_KAM_Multical41

This block is used to read water meters from Kamstrup:
The block can only be executed together with the block FB_MBUSKL6781 [18].

**Functionality of the block [9]**

**VAR_INPUT**

- **usiAddress**: Primary address [10] of the meter to be read with this block.
- **stSecAddr**: Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAdr [205]).
- **eBaudrate**: 300, 2400 baud (see E_MBUS_Baudrate [197]).
- **bStart**: A positive edge of this input triggers one reading of the meter.
- **bSND_NKE**: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).
- **bReadInit**: If the PLC is restarted, the meter is read once.
- **tMinSendTime**: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.
- **bDisabled**: TRUE = deselection of the block.

**VAR_OUTPUT**

- **bBusy**: The bBusy output is TRUE while the meter is being read.
- **bReady**: The bReady output is TRUE for one cycle, once meter reading is completed.
- **bError**: this output goes TRUE as soon as an error occurs. This error is described via the variable eError.
- **eError**: In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.
- **dwIdNumber**: Serial number of the meter (secondary address).
- **byStatus**: Device status.
- **byGEN**: Software version of the device.
- **byCounter**: Number of times the master has accessed data of the respective slave.
- **usiRecivedAdr**: Received primary address (0-250).
- **eMedium**: Medium (see E_MBUS_Medium [200]).
sMan: Manufacturer code.

stVolume: Meter reading, water consumption (see ST_MBus_Info [204]).

stFlow: Current water consumption (see ST_MBus_Info [204]).

VAR_IN_OUT

stCom : ST_MBUS_Communication;

stCom: This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Communication [202]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.121.6 FB_MBUS_KAM_Multical601

This block is used for reading heat/cold meters from Kamstrup:

-Multical 601

The block can only be executed together with the block FB_MBUSKL6781() [18].

Functionality of the block [9]

VAR_INPUT

usiAddress : USINT;
stSecAdr : ST_MBUS_SecAdr;
eBaudrate : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart : BOOL;
bSND_NKE : BOOL := TRUE;
bReadInit : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit : USINT;
bDisabled : BOOL := FALSE;

usiAddress: Primary address [► 10] of the meter to be read with this block.

stSecAdr: Secondary address [►11] of the meter to be read with this block (see ST_MBUS_SecAdr [►205]).

eBaudrate: 300, 2400, 9600 baud (see E_MBUS_Baudrate [►197]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the
meter is not read and can be read manually with bStart.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bReady : BOOL;
bError : BOOL;
eError : E_MBUS_ERROR;
dwIdNumber : DWORD;
byStatus : BYTE;
byGEN : BYTE;
byCounter : BYTE;
usiRecivedAdr : USINT;
eMedium : E_MBUS_Medium;
sMan : STRING(3);
stEnergy : ST_MBus_Info;
stPower : ST_MBus_Info;
stVolume : ST_MBus_Info;
stFlow : ST_MBus_Info;
stForwardTemp : ST_MBus_Info;
stReturnTemp : ST_MBus_Info;
stDiffTemp : ST_MBus_Info;
stCoolingEnergy : ST_MBus_Info;
stEnergyT2 : ST_MBus_Info;
stEnergyT3 : ST_MBus_Info;
stPulsecounter1 : ST_MBus_Info;
stPulsecounter2 : ST_MBus_Info;

bBusy: The bBusy output is TRUE while the meter is being read.

bReady: The bReady output is TRUE for one cycle, once meter reading is completed.

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

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [►197]). bError goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [►201]).
sMan: Manufacturer code.

stEnergy: Meter reading, energy consumption (see ST_MBus_Info [204]).

stPower: Current energy consumption, power (see ST_MBus_Info [204]).

stVolume: Meter reading, water consumption (see ST_MBus_Info [204]).

stFlow: Current water consumption (see ST_MBus_Info [204]).

stForwardTemp: Flow temperature (see ST_MBus_Info [204]).

stReturnTemp: Return temperature (see ST_MBus_Info [204]).

stDiffTemp: Temperature difference (see ST_MBus_Info [204]).

stCoolingEnergy: Meter reading, cooling energy consumption (see ST_MBus_Info [204]).

stEnergyT2: Meter reading, energy consumption, tariff 2 (see ST_MBus_Info [204]).

stEnergyT3: Meter reading, energy consumption, tariff 3 (see ST_MBus_Info [204]).

stPulsecounter1: Pulse counter 1 (see ST_MBus_Info [204]).

stPulsecounter2: Pulse counter 2 (see ST_MBus_Info [204]).

VAR_IN_OUT

stCom: ST_MBUS_Communication;

stCom: This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Communication [202]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.22 Kundo overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [76], FB_MBUS_General_Ext [79] or FB_MBUS_General_Param [83] from the folder "General [74]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [85] can be used to send data to the device (e.g. setting of the primary address).

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Type</th>
<th>Device</th>
<th>Block</th>
</tr>
</thead>
<tbody>
<tr>
<td>KUNDO</td>
<td>Heat/cold meter</td>
<td>Compact WMZ G20</td>
<td>FB_MBUS KST G20 [133]</td>
</tr>
<tr>
<td></td>
<td>Heat/cold meter</td>
<td>Compact WMZ G21</td>
<td>FB_MBUS KST G20 [133]</td>
</tr>
<tr>
<td></td>
<td>External M-Bus module</td>
<td>him1s</td>
<td>FB_MBUS KST him1 [135]</td>
</tr>
<tr>
<td></td>
<td>External M-Bus module</td>
<td>him1plus</td>
<td>FB_MBUS KST him1 [135]</td>
</tr>
<tr>
<td></td>
<td>Pulse input</td>
<td>him1plus</td>
<td>FB_MBUS KST him1Puls [137]</td>
</tr>
</tbody>
</table>
This block is used for reading heat/cold meters from KUNDO System Technik:

- Kompakt WZM G20 (with internal M-Bus module)
- Kompakt WZM G21 (with internal M-Bus module)

The block can only be executed together with the block FB_MBUSKL6781().

**Functionality of the block [➔ 9]**

**VAR_INPUT**

- **usiAddress**: Primary address [➔ 10] of the meter to be read with this block.
- **stSecAdr**: Secondary address [➔ 11] of the meter to be read with this block (see ST_MBUS_SecAddr [➔ 205]).
- **eBaudrate**: 300, 2400, 9600 baud (see E_MBUS_Baudrate [➔ 197]).
- **bStart**: A positive edge of this input triggers one reading of the meter.
- **bSND_NKE**: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).
- **bReadInit**: If the PLC is restarted, the meter is read once.
- **tMinSendTime**: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.
- **usiUnit**: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2=MW(h) / 3=GW(h).
- **bDisabled**: TRUE = deselection of the block.
VAR_OUTPUT

bBusy : BOOL;
bReady : BOOL;
bError : BOOL;
eError : E_MBUS_ERROR;
dwIdNumber : DWORD;
byStatus : BYTE;
byGEN : BYTE;
byCounter : BYTE;
usiRecivedAdr : USINT;
eMedium : E_MBUS_Medium;
sMan : STRING(3);
stEnergy : ST_MBus_Info;
stPower : ST_MBus_Info;
stVolume : ST_MBus_Info;
stFlow : ST_MBus_Info;
stForwardTemp : ST_MBus_Info;
stReturnTemp : ST_MBus_Info;
stDiffTemp : ST_MBus_Info;

**bBusy:** The bBusy output is TRUE while the meter is being read.

**bReady:** The bReady output is TRUE for one cycle, once meter reading is completed.

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

**eError:** In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.

**dwIdNumber:** Serial number of the meter (secondary address).

**byStatus:** Device status.

**byGEN:** Software version of the device.

**byCounter:** Number of times the master has accessed data of the respective slave.

**usiRecivedAdr:** Received primary address (0-250).

**eMedium:** Medium (see E_MBUS_Medium [200]).

**sMan:** Manufacturer code.

**stEnergy:** Meter reading, energy consumption (see ST_MBus_Info [204]).

**stPower:** Current energy consumption, power (see ST_MBus_Info [204]).

**stVolume:** Meter reading, water consumption (see ST_MBus_Info [204]).

**stFlow:** Current water consumption (see ST_MBus_Info [204]).

**stForwardTemp:** Flow temperature (see ST_MBus_Info [204]).

**stReturnTemp:** Return temperature (see ST_MBus_Info [204]).

**stDiffTemp:** Temperature difference (see ST_MBus_Info [204]).

VAR_IN_OUT

stCom : ST_MBUS_Communication;

**stCom:** This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Communication [202]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>
This block is used for reading M-Bus modules from KUNDO System Technik:

- him1s
- him1plus

These modules can be used for reading consumption data from a KUNDO arithmetic unit.

The block can only be executed together with the block FB_MBUSKL6781 [18].

**Functionality of the block [9]**

**VAR_INPUT**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>usiAddress</td>
<td>USINT</td>
<td>Primary address [10] of the meter to be read with this block.</td>
</tr>
<tr>
<td>stSecAdr</td>
<td>ST_MBUS_SecAdr</td>
<td>Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAdr [205]).</td>
</tr>
<tr>
<td>eBaudrate</td>
<td>E_MBUS_Baudrate</td>
<td>300, 2400 baud (see E_MBUS_Baudrate [197]).</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL</td>
<td>A positive edge of this input triggers one reading of the meter.</td>
</tr>
<tr>
<td>bSND_NKE</td>
<td>BOOL</td>
<td>TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).</td>
</tr>
<tr>
<td>bReadInit</td>
<td>BOOL</td>
<td>If the PLC is restarted, the meter is read once.</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>TIME</td>
<td>Standard t#2s. The meter is read again, once the time set here has elapsed.</td>
</tr>
<tr>
<td>usiUnit</td>
<td>USINT</td>
<td>Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2=MW(h) / 3=GW(h).</td>
</tr>
</tbody>
</table>
bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bReady : BOOL;
bError : BOOL;
eError : E_MBUS_ERROR;
dwIdNumber : DWORD;
byStatus : BYTE;
byGEN : BYTE;
byCounter : BYTE;
usiRecivedAdr : USINT;
eMedium : E_MBUS_Medium;
sMan : STRING(3);
stEnergy : ST_MBus_Info;
stPower : ST_MBus_Info;
stVolume : ST_MBus_Info;
stFlow : ST_MBus_Info;
stForwardTemp : ST_MBus_Info;
stReturnTemp : ST_MBus_Info;
stDiffTemp : ST_MBus_Info;

bBusy: The bBusy output is TRUE while the meter is being read.

bReady: The bReady output is TRUE for one cycle, once meter reading is completed.

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

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [200]).

sMan: Manufacturer code.

stEnergy: Meter reading, energy consumption (see ST_MBus_Info [204]).

stPower: Current energy consumption, power (see ST_MBus_Info [204]).

stVolume: Meter reading, water consumption (see ST_MBus_Info [204]).

stFlow: Current water consumption (see ST_MBus_Info [204]).

stForwardTemp: Flow temperature (see ST_MBus_Info [204]).

stReturnTemp: Return temperature (see ST_MBus_Info [204]).

stDiffTemp: Temperature difference (see ST_MBus_Info [204]).

VAR_IN_OUT

stCom : ST_MBUS_Communication;

stCom: This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Communication [202]).
Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.22.3 FB_MBUS_KST_him1Puls

This block is used for reading M-Bus modules from KUNDO System Technik:

-him1plus (pulse input)

The block can only be executed together with the block FB_MBUSKL6781) [18].

Functionality of the block [9]

VAR_INPUT

<table>
<thead>
<tr>
<th>siAddress</th>
<th>USINT;</th>
</tr>
</thead>
<tbody>
<tr>
<td>stSecAdr</td>
<td>ST_MBUS_SecAdr;</td>
</tr>
<tr>
<td>eBaudrate</td>
<td>E_MBUS_Baudrate := eMBUS_Baud2400;</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL;</td>
</tr>
<tr>
<td>bSND_NKE</td>
<td>BOOL := TRUE;</td>
</tr>
<tr>
<td>bReadInit</td>
<td>BOOL := TRUE;</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>TIME := t#2s;</td>
</tr>
<tr>
<td>usiUnit</td>
<td>USINT;</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL := FALSE;</td>
</tr>
</tbody>
</table>

usiAddress: Primary address [10] of the meter to be read with this block.

stSecAdr: Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAdr [205]).

eBaudrate: 300, 2400 baud (see E_MBUS_Baudrate [197]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2=MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.
VAR_OUTPUT

VAR_OUTPUT

bBusy : BOOL;
bReady : BOOL;
bError : BOOL;
eError : E_MBUS_ERROR;
dwIdNumber : DWORD;
byStatus : BYTE;
byGEN : BYTE;
byCounter : BYTE;
usiRecivedAdr : USINT;
eMedium : E_MBUS_Medium;
sMan : STRING(3);
stValue : ST_MBus_Info;

bBusy: The bBusy output is TRUE while the meter is being read.

bReady: The bReady output is TRUE for one cycle, once meter reading is completed.

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

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [200]).

sMan: Manufacturer code.

stValue: Meter reading (see ST_MBus_Info [204]).

VAR_IN_OUT

VAR_IN_OUT

stCom : ST_MBUS_Communication;

stCom: This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Communication [202]).

Requirements

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.23 Landis & Gyr overview

4.1.23 Landis & Gyr overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [76], FB_MBUS_General_Ext [79] or FB_MBUS_General_Param [83] from the folder "General [74]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [85] can be used to send data to the device (e.g. setting of the primary address).
This block is used for reading heat/cold meters from Landis & Gyr:

- 2WR5
- 2WR6
- UH50

The block can only be executed together with the block FB_MBUSKL6781[18].

**Functionality of the block[9]**

**VAR_INPUT**

- **usiAddress**: Primary address[10] of the meter to be read with this block.
- **stSecAdr**: Secondary address[11] of the meter to be read with this block (see ST_MBUS_SecAdr[205]).
- **eBaudrate**: 300, 1200, 2400, 4800 baud (see E_MBUS_Baudrate[197]).
**bStart:** A positive edge of this input triggers one reading of the meter.

**bSND_NKE:** TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

**bReadInit:** If the PLC is restarted, the meter is read once.

**tMinSendTime:** Standard #s2. The meter is read again, once the time set here has elapsed. If t<0 the meter is not read and can be read manually with bStart.

**usiUnit:** Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2=MW(h) / 3=GW(h).

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

**VAR_OUTPUT**

```plaintext
bBusy: BOOL;
bReady: BOOL;
bError: BOOL;
eError: E_MBUS_ERROR;
dwIdNumber: DWORD;
byStatus: BYTE;
byGEN: BYTE;
byCounter: BYTE;
usiRecivedAdr: USINT;
eMedium: E_MBUS_Medium;
sMan: STRING(3);
stEnergy: ST_MBus_Info;
stPower: ST_MBus_Info;
stVolume: ST_MBus_Info;
stFlow: ST_MBus_Info;
stForwardTemp: ST_MBus_Info;
stReturnTemp: ST_MBus_Info;
stDiffTemp: ST_MBus_Info;
```

**bBusy:** The bBusy output is TRUE while the meter is being read.

**bReady:** The bReady output is TRUE for one cycle, once meter reading is completed.

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

**eError:** In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.

**dwIdNumber:** Serial number of the meter (secondary address).

**byStatus:** Device status.

**byGEN:** Software version of the device.

**byCounter:** Number of times the master has accessed data of the respective slave.

**usiRecivedAdr:** Received primary address (0-250).

**eMedium:** Medium (see E_MBUS_Medium [200]).

**sMan:** Manufacturer code.

**stEnergy:** Meter reading, energy consumption (see ST_MBus_Info [204]).

**stPower:** Current energy consumption, power (see ST_MBus_Info [204]).

**stVolume:** Meter reading, water consumption (see ST_MBus_Info [204]).

**stFlow:** Current water consumption (see ST_MBus_Info [204]).

**stForwardTemp:** Flow temperature (see ST_MBus_Info [204]).

**stReturnTemp:** Return temperature (see ST_MBus_Info [204]).

**stDiffTemp:** Temperature difference (see ST_MBus_Info [204]).
VAR_IN_OUT

stCom : ST_MBUS_Communication;

stCom: This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Communication [202]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.24 Metrima overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [76], FB_MBUS_General_Ext [79] or FB_MBUS_General_Param [83] from the folder "General [74]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [85] can be used to send data to the device (e.g. setting of the primary address).

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Type</th>
<th>Device</th>
<th>Block</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metrima</td>
<td>Heat meter</td>
<td>F22 (standard values)</td>
<td>FB_MBUS_SVM_F22 [141]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>F22 (with additional output values)</td>
<td>FB_MBUS_SVM_F22_Ext [144]</td>
</tr>
</tbody>
</table>

4.1.24.1 FB_MBUS_SVM_F22

This block is used to read heat meters from Metrima:

-F22

-usiAddress
-istSecAdr
-eBaudrate
-tStart
-tSND_NKE
tReadInit
-tMinSendTime
-usiUnit
-tDisabled
-exstCom
-ebusy
-brdy
-bError
eError
dwldNumber
-byStatus
-byGEN
-byCounter
-usiRecevAd
-emedium
-sMan
-estEnergy
-stVolume
-stVolume2
-stForwardTemp
-stReturnTemp
-stDiffTemp
-stFlow
-stPower
-stPulsecounter1
-stPulsecounter2
The block can only be executed together with the block `FB_MBUSKL6781` [► 18].

**Functionality of the block [► 9]**

### VAR_INPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>usiAddress</td>
<td>Primary address [► 10] of the meter to be read with this block.</td>
</tr>
<tr>
<td>stSecAdr</td>
<td>Secondary address [► 11] of the meter to be read with this block (see ST_MBUS_SecAdr [► 205]).</td>
</tr>
<tr>
<td>eBaudrate</td>
<td>300, 2400, 9600 baud (see E_MBUS_Baudrate [► 197]).</td>
</tr>
<tr>
<td>bStart</td>
<td>A positive edge of this input triggers one reading of the meter.</td>
</tr>
<tr>
<td>bSND_NKE</td>
<td>TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).</td>
</tr>
<tr>
<td>bReadInit</td>
<td>If the PLC is restarted, the meter is read once.</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>Standard t#2s. The meter is read again, once the time set here has elapsed.</td>
</tr>
<tr>
<td>usiUnit</td>
<td>Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).</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 <em>bBusy</em> output is TRUE while the meter is being read.</td>
</tr>
<tr>
<td>bReady</td>
<td>The <em>bReady</em> output is TRUE for one cycle, once meter reading is completed.</td>
</tr>
<tr>
<td>bError</td>
<td>this output goes TRUE as soon as an error occurs. This error is described via the variable <em>eError</em>.</td>
</tr>
<tr>
<td>eError</td>
<td>In the event of an error the output issues an error code (see E_MBUS_ERROR [► 197]). <em>bError</em> goes TRUE at the same time.</td>
</tr>
<tr>
<td>dwIdNumber</td>
<td>Serial number of the meter (secondary address).</td>
</tr>
<tr>
<td>byStatus</td>
<td>Device status.</td>
</tr>
<tr>
<td>byGEN</td>
<td></td>
</tr>
<tr>
<td>byCounter</td>
<td></td>
</tr>
<tr>
<td>usiRecivedAdr</td>
<td></td>
</tr>
<tr>
<td>eMedium</td>
<td></td>
</tr>
<tr>
<td>sMan</td>
<td>STRING(3);</td>
</tr>
<tr>
<td>stEnergy</td>
<td>ST_Mbus_Info;</td>
</tr>
<tr>
<td>stVolume</td>
<td>ST_Mbus_Info;</td>
</tr>
<tr>
<td>stVolume2</td>
<td>ST_Mbus_Info;</td>
</tr>
<tr>
<td>stForwardTemp</td>
<td>ST_Mbus_Info;</td>
</tr>
<tr>
<td>stReturnTemp</td>
<td>ST_Mbus_Info;</td>
</tr>
<tr>
<td>stDiffTemp</td>
<td>ST_Mbus_Info;</td>
</tr>
<tr>
<td>stFlow</td>
<td>ST_Mbus_Info;</td>
</tr>
<tr>
<td>stPower</td>
<td>ST_Mbus_Info;</td>
</tr>
<tr>
<td>stPulsecounter1</td>
<td>ST_Mbus_Info;</td>
</tr>
<tr>
<td>stPulsecounter2</td>
<td>ST_Mbus_Info;</td>
</tr>
</tbody>
</table>

**bBusy**: The *bBusy* output is TRUE while the meter is being read.

**bReady**: The *bReady* output is TRUE for one cycle, once meter reading is completed.

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

**eError**: In the event of an error the output issues an error code (see E_MBUS_ERROR [► 197]). *bError* goes TRUE at the same time.

**dwIdNumber**: Serial number of the meter (secondary address).

**byStatus**: Device status.
byGEN: Software version of the device.
byCounter: Number of times the master has accessed data of the respective slave.
usiRecivedAdr: Received primary address (0-250).
eMedium: Medium (see E_MBUS_Medium [200]).
sMan: Manufacturer code.

stEnergy: Meter reading, energy consumption (see ST_MBus_Info [204]).
stVolume: Meter reading, water consumption (see ST_MBus_Info [204]).
stVolume2: Accumulated volume. Energy calculation (see ST_MBus_Info [204]).
stForwardTemp: Flow temperature (see ST_MBus_Info [204]).
stReturnTemp: Return temperature (see ST_MBus_Info [204]).
stDiffTemp: Temperature difference (see ST_MBus_Info [204]).
stFlow: Current water consumption (see ST_MBus_Info [204]).
stPower: Current energy consumption, power (see ST_MBus_Info [204]).
stPulsecounter1: Pulse counter 1 (see ST_MBus_Info [204]).
stPulsecounter2: Pulse counter 2 (see ST_MBus_Info [204]).

VAR_IN_OUT
stCom: ST_MBUS_Communication;

stCom: This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Communication [202]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.1.24.2  FB_MBUS_SVM_F22_Ext

This block is used to read heat meters from Metrima:

-F22 (as FB_MBUS_SVM_F22() [141], but with the extended output values arrAccountAccums and arrMonthlyAccums.)

This block is not suitable for BC/BX.

The block can only be executed together with the block FB_MBUSKL6781() [18].

Functionality of the block [9]

VAR_INPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>usiAddress</td>
<td>USINT;</td>
</tr>
<tr>
<td>stSecAdr</td>
<td>ST_MBUS_SecAdr;</td>
</tr>
<tr>
<td>eBaudrate</td>
<td>E_MBUS_Baudrate := eMBUS_Baud2400;</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL;</td>
</tr>
<tr>
<td>bSND_NKE</td>
<td>BOOL := TRUE;</td>
</tr>
<tr>
<td>bReadInit</td>
<td>BOOL := TRUE;</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>TIME := t#2s;</td>
</tr>
<tr>
<td>usiUnit</td>
<td>USINT;</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL := FALSE;</td>
</tr>
<tr>
<td>bMonthstorages</td>
<td>BOOL;</td>
</tr>
<tr>
<td>byMonthstorages</td>
<td>BYTE;</td>
</tr>
</tbody>
</table>

usiAddress: Primary address [10] of the meter to be read with this block.

stSecAdr: Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAdr [205]).

eBaudrate: 300, 2400, 9600 baud (see E_MBUS_Baudrate [197]).
bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard #2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

bMonthstorages: =False, cutoff dates and monthly values are not read (arrAccountAccums and arrMonthlyAccums) / =TRUE, cutoff dates and monthly values are read (arrAccountAccums and arrMonthlyAccums). The number of monthly values (arrMonthlyAccums) can be changed and depends on the variable byMonthstorages.

byMonthstorages: Number of monthly values (arrMonthlyAccums), maximum 37 values. Only applies if byMonthstorages =TRUE.

VAR_OUTPUT

bBusy : BOOL;
bReady : BOOL;
bError : BOOL;
eError : E_MBUS_ERROR;
dwIdNumber : DWORD;
byStatus : BYTE;
byGEN : BYTE;
byCounter : BYTE;
usiRecivedAdr : USINT;
eMedium : _MBUS_Medium;
sMan : STRING(3);
stEnergy : ST_MBus_Info;
stVolume : ST_MBus_Info;
stVolume2 : ST_MBus_Info;
stForwardTemp : ST_MBus_Info;
stReturnTemp : ST_MBus_Info;
stDiffTemp : ST_MBus_Info;
stFlow : ST_MBus_Info;
stPowerr : ST_MBus_Info;
stPulsecounter1 : ST_MBus_Info;
stPulsecounter2 : ST_MBus_Info;
arrAccountAccums : ARRAY [1..2] OF ST_MBus_F22;
arrMonthlyAccums : ARRAY [1..37] OF ST_MBus_F22;

bBusy: The bBusy output is TRUE while the meter is being read.

bReady: The bReady output is TRUE for one cycle, once meter reading is completed.

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

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see _MBUS_Medium [200]).

sMan: Manufacturer code.

stEnergy: Meter reading, energy consumption (see ST_MBus_Info [204]).
**stVolume**: Meter reading, water consumption (see ST_MBus_Info [»204]).

**stVolume2**: Accumulated volume. Energy calculation (see ST_MBus_Info [»204]).

**stForwardTemp**: Flow temperature (see ST_MBus_Info [»204]).

**stReturnTemp**: Return temperature (see ST_MBus_Info [»204]).

**stDiffTemp**: Temperature difference (see ST_MBus_Info [»204]).

**stFlow**: Current water consumption (see ST_MBus_Info [»204]).

**stPower**: Current energy consumption, power (see ST_MBus_Info [»204]).

**stPulsecounter1**: Pulse counter 1 (see ST_MBus_Info [»204]).

**stPulsecounter2**: Pulse counter 2 (see ST_MBus_Info [»204]).

**arrAccountAccums**: 2 cutoff date values (energy, volume 1, volume 2, pulse counter 1, pulse counter 2, date). Values are only read if bMonthstorages = TRUE (see ST_MBus_F22 [»206]).

**arrMonthlyAccums**: Maximum 37 monthly values (energy, volume 1, volume 2, pulse counter 1, pulse counter 2, date). Values are only read if bMonthstorages = TRUE. The number of values depends on the variable byMonthstorages (see ST_MBus_F22 [»206]).

**VAR_IN_OUT**

```plaintext
stCom : ST_MBUS_Communication;
```

**stCom**: This structure is used to link the block FB_MBUSKL6781() [»202] with the meter blocks (see ST_MBUS_Communication [»202]).

### Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.25 NZR overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [»76], FB_MBUS_General_Ext [»79] or FB_MBUS_General_Param [»83] from the folder "General [»74]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [»85] can be used to send data to the device (e.g. setting of the primary address).

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Type</th>
<th>Device</th>
<th>Block</th>
</tr>
</thead>
<tbody>
<tr>
<td>NZR</td>
<td>2-way pulse adapter</td>
<td>IC-M2</td>
<td>FB_MBUS_NZR_ICM2 [»147]</td>
</tr>
<tr>
<td></td>
<td>2-way pulse adapter</td>
<td>IC-M2C</td>
<td>FB_MBUS_NZR_ICM2 [»147]</td>
</tr>
<tr>
<td></td>
<td>Water meter</td>
<td>Modularis 2</td>
<td>FB_MBUS_NZR_Modularis 2 [»149]</td>
</tr>
</tbody>
</table>
4.1.25.1  FB_MBUS_NZR_ICM2

This block is used to read energy meters with pulse output from NZR:

-IC-M2

-IC-M2C

The block can only be executed together with the block FB_MBUSKL6781()

Up to 2 pulse generators can be connected to an IC-M2/IC-M2C at the same time. The IC-M2/IC-M2C behaves like 2 independent slaves.

Functionality of the block

VAR_INPUT

usiAddress : USINT;
stSecAdr : ST_MBUS_SecAdr;
eBaudrate : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart : BOOL;
bSND_NKE : BOOL := TRUE;
bReadInit : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit : USINT;
bDisabled : BOOL := FALSE;

usiAddress: Primary address of the meter to be read with this block.
stSecAdr: Secondary address of the meter to be read with this block (see ST_MBUS_SecAdr).
eBaudrate: 300, 2400 baud (see E_MBUS_Baudrate).
bStart: A positive edge of this input triggers one reading of the meter.
bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).
bReadInit: If the PLC is restarted, the meter is read once.
tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.
usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).
bDisabled: TRUE = deselection of the block.
bBusy: The bBusy output is TRUE while the meter is being read.

bReady: The bReady output is TRUE for one cycle, once meter reading is completed.

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

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [200]).

sMan: Manufacturer code.

stValue: Meter reading (see ST_MBus_Info [204]).

VAR_IN_OUT

stCom: This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Communication [202]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>
This block is used to read water meters from NZR:

- Modularis 2

The block can only be executed together with the block FB_MBUSKL6781() [18].

Functionality of the block [9]

VAR_INPUT

usiAddress : USINT;
stSecAdr : ST_MBUS_SecAdr;
eBaudrate : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart : BOOL;
bSND_NKE : BOOL := TRUE;
bReadInit : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
bDisabled : BOOL := FALSE;

usiAddress: Primary address [10] of the meter to be read with this block.

stSecAdr: Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAdr [205]).

eBaudrate: 300, 2400 baud (see E_MBUS_Baudrate [197]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bReady : BOOL;
bError : BOOL;
eError : E_MBUS_ERROR;
dwIdNumber : DWORD;
byStatus : BYTE;
byGEN : BYTE;
byCounter : BYTE;
usiRecivedAdr : USINT;
eMedium : E_MBUS_Medium;
sMan : STRING(3);
stVolume : ST_MBus_Info;

bBusy: The bBusy output is TRUE while the meter is being read.

bReady: The bReady output is TRUE for one cycle, once meter reading is completed.

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

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.

dwldNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [200]).

sMan: Manufacturer code.

stVolume: Meter reading, water consumption (see ST_MBus_Info [204]).

VAR_IN_OUT

stCom : ST_MBUS_Communication;

stCom: This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Communication [202]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.26 OPTEC overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [76], FB_MBUS_General_Ext [79] or FB_MBUS_General_Param [83] from the folder "General [74]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [85] can be used to send data to the device (e.g. setting of the primary address).

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Type</th>
<th>Device</th>
<th>Block</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPTEC</td>
<td>Electricity meter</td>
<td>ECS Type 2</td>
<td>FB_MBUS_OPT_ECSType2 [151]</td>
</tr>
</tbody>
</table>
This block is used to read electricity meters from OPTEC:

-ECS (default readout data type 2)

The block can only be executed together with the block `FB_MBUSEKL6781()`.  

Functionality of the block [9]

**VAR_INPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>usiAddress</td>
<td>Primary address [10] of the meter to be read with this block.</td>
</tr>
<tr>
<td>stSecAdr</td>
<td>Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAdr [205]).</td>
</tr>
<tr>
<td>eBaudrate</td>
<td>300, 600, 1200, 2400, 4800, 9600 baud (see E_MBUS_Baudrate [197]).</td>
</tr>
<tr>
<td>bStart</td>
<td>A positive edge of this input triggers one reading of the meter.</td>
</tr>
<tr>
<td>bSND_NKE</td>
<td>TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).</td>
</tr>
<tr>
<td>bReadInit</td>
<td>If the PLC is restarted, the meter is read once.</td>
</tr>
</tbody>
</table>

### FB_MBUS_OPT_ECSType2

```plaintext
FB_MBUS_OPT_ECSType2

usiAddress     bEbusy
stSecAdr       bReady
eBaudrate      bError
bStart         eError
bSND_NKE       dwIdNumber
bReadInit      byStatus
tMinSendTime   byGEN
usiUnit        byCounter
bDisabled      usiReadedAd
stCom          eMedium
sMan
stEnergyT1_L1
stEnergyT1_L2
stEnergyT1_L3
stEnergyT1_Total
stEnergyT2_L1
stEnergyT2_L2
stEnergyT2_L3
stEnergyT2_Total
stPowerL1
stPower2
stPowerL3
stPowerTotal
stActiveTariff
stStatusByte4
```
tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bReady : BOOL;
bError : BOOL;
eError : E_MBUS_ERROR;
dwIdNumber : DWORD;
byStatus : BYTE;
byGEN : BYTE;
byCounter : BYTE;
usiRecivedAdr : USINT;
eMedium : E_MBUS_Medium;
sMan : STRING(3);
stEnergyT1_L1 : ST_MBus_Info;
stEnergyT1_L2 : ST_MBus_Info;
stEnergyT1_L3 : ST_MBus_Info;
stEnergyT1_Total : ST_MBus_Info;
stEnergyT2_L1 : ST_MBus_Info;
stEnergyT2_L2 : ST_MBus_Info;
stEnergyT2_L3 : ST_MBus_Info;
stEnergyT2_Total : ST_MBus_Info;
stPowerL1 : ST_MBus_Info;
stPowerL2 : ST_MBus_Info;
stPowerL3 : ST_MBus_Info;
stPowerTotal : ST_MBus_Info;
stActiveTariff : ST_MBus_Info;
stStatusByte4 : ST_MBus_Info;

bBusy: The bBusy output is TRUE while the meter is being read.

bReady: The bReady output is TRUE for one cycle, once meter reading is completed.

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

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [201]).

sMan: Manufacturer code.

stEnergyT1_L1: Counter value, active energy consumption, tariff 1, phase L1 (see ST_MBus_Info [204]).

stEnergyT1_L2: Meter reading, active energy consumption, tariff 1, phase L2 (see ST_MBus_Info [204]).

stEnergyT1_L3: Meter reading, active energy consumption, tariff 1, phase L3 (see ST_MBus_Info [204]).

stEnergyT1_Total: Meter reading, active energy consumption, tariff 1, total (see ST_MBus_Info [204]).

stEnergyT2_L1: Meter reading, active energy consumption, tariff 2, phase L1 (see ST_MBus_Info [204]).

stEnergyT2_L2: Meter reading, active energy consumption, tariff 2, phase L2 (see ST_MBus_Info [204]).

stEnergyT2_L3: Meter reading, active energy consumption, tariff 2, phase L3 (see ST_MBus_Info [204]).
stEnergyT2_Total: Meter reading, active energy consumption, tariff 2, total (see ST_MBus_Info [204]).

stPowerL1: Instantaneous consumption, power, phase L1 (see ST_MBus_Info [204]).

stPowerL2: Instantaneous consumption, power, phase L2 (see ST_MBus_Info [204]).

stPowerL3: Instantaneous consumption, power, phase L3 (see ST_MBus_Info [204]).

stPowerTotal: Instantaneous consumption, power, total (see ST_MBus_Info [204]).

stActiveTariff: Current tariff (see ST_MBus_Info [204]).

stStatusByte4: Range overflow alarms (see ST_MBus_Info [204]).

VAR_IN_OUT

stCom: ST_MBUS_Communication;

 VAR_IN_OUT

stCom: This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Communication [202]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.27 Relay overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [76], FB_MBUS_General_Ext [79] or FB_MBUS_General_Param [83] from the folder "General [74]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [85] can be used to send data to the device (e.g. setting of the primary address).

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Type</th>
<th>Device</th>
<th>Block</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relay</td>
<td>1-4 analog inputs</td>
<td>AnDi 1-4</td>
<td>FB_MBUS_REL_AnDi [154]</td>
</tr>
<tr>
<td></td>
<td>4 digital inputs</td>
<td>PadIn 4</td>
<td>FB_MBUS_REL_PadIn4 [156]</td>
</tr>
<tr>
<td></td>
<td>1-way pulse adapter</td>
<td>PadPuls M1</td>
<td>FB_MBUS_REL_PadPulsM1 [158]</td>
</tr>
<tr>
<td></td>
<td>1-way pulse adapter</td>
<td>PadPuls M1C</td>
<td>FB_MBUS_REL_PadPulsM1 [158]</td>
</tr>
<tr>
<td></td>
<td>2-way pulse adapter</td>
<td>PadPuls M2</td>
<td>FB_MBUS_REL_PadPulsM2 [160]</td>
</tr>
<tr>
<td></td>
<td>2-way pulse adapter</td>
<td>PadPuls M2C</td>
<td>FB_MBUS_REL_PadPulsM2 [160]</td>
</tr>
<tr>
<td></td>
<td>4-way pulse adapter</td>
<td>PadPuls M4</td>
<td>FB_MBUS_REL_PadPulsM4 [162]</td>
</tr>
<tr>
<td></td>
<td>4-way pulse adapter</td>
<td>PadPuls M4L</td>
<td>FB_MBUS_REL_PadPulsM4 [162]</td>
</tr>
</tbody>
</table>
4.1.27.1 FB_MBUS_REL_AnDi

This block is used for reading analog converters from Relay:

- AnDi 1 (1x 0/4-20 mA or 0-10 V)
- AnDi 2 (2x 0/4-20 mA or 0-10 V)
- AnDi 3 (3x 0/4-20 mA or 0-10 V)
- AnDi 4 (4x 0/4-20 mA or 0-10 V)

The block can only be executed together with the block FB_MBUSKL6781.\[18\].

Up to 4 sensors can be connected to an AnDi 4 at the same time. The AnDi 4 behaves like 4 independent slaves.

**Functionality of the block** [9]

**VAR_INPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>usiAddress</td>
<td>USINT;</td>
</tr>
<tr>
<td>stSecAdr</td>
<td>ST_MBUS_SecAdr;</td>
</tr>
<tr>
<td>eBaudrate</td>
<td>E_MBUS_Baudrate := eMBUS_Baud2400;</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL;</td>
</tr>
<tr>
<td>bSND_NKE</td>
<td>BOOL := TRUE;</td>
</tr>
<tr>
<td>bReadInit</td>
<td>BOOL := TRUE;</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>TIME := t#2s;</td>
</tr>
<tr>
<td>usiUnit</td>
<td>USINT;</td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL := FALSE;</td>
</tr>
</tbody>
</table>

**usiAddress**: Primary address [10] of the meter to be read with this block.

**stSecAdr**: Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAdr [205]).

**eBaudrate**: 300, 2400 baud (see E_MBUS_Baudrate [197]).

**bStart**: A positive edge of this input triggers one reading of the meter.

**bSND_NKE**: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

**bReadInit**: If the PLC is restarted, the meter is read once.

**tMinSendTime**: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.
**usiUnit:** Unit of the energy values to be output by the block. \(0 = \text{W(h)} / 1 = \text{KW(h)} / 2 = \text{MW(h)} / 3 = \text{GW(h)}\).

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

### VAR_OUTPUT

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>BOOL</td>
</tr>
<tr>
<td>bReady</td>
<td>BOOL</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>eError</td>
<td>E_MBUS_ERROR</td>
</tr>
<tr>
<td>dwIdNumber</td>
<td>DWORD</td>
</tr>
<tr>
<td>byStatus</td>
<td>BYTE</td>
</tr>
<tr>
<td>byGEN</td>
<td>BYTE</td>
</tr>
<tr>
<td>byCounter</td>
<td>BYTE</td>
</tr>
<tr>
<td>usiRecivedAdr</td>
<td>USINT</td>
</tr>
<tr>
<td>eMedium</td>
<td>E_MBUS_Medium</td>
</tr>
<tr>
<td>sMan</td>
<td>STRING(3)</td>
</tr>
<tr>
<td>stValue</td>
<td>ST_MBus_Info</td>
</tr>
<tr>
<td>stMax</td>
<td>ST_MBus_Info</td>
</tr>
<tr>
<td>stOffset</td>
<td>ST_MBus_Info</td>
</tr>
<tr>
<td>byInfo</td>
<td>BYTE</td>
</tr>
</tbody>
</table>

**bBusy:** The \(b\text{Busy}\) output is TRUE while the meter is being read.

**bReady:** The \(b\text{Ready}\) output is TRUE for one cycle, once meter reading is completed.

**bError:** this output goes TRUE as soon as an error occurs. This error is described via the variable \(e\text{Error}\).

**eError:** In the event of an error the output issues an error code (see \(E\text{_MBUS\_ERROR}\)). \(b\text{Error}\) goes TRUE at the same time.

**dwIdNumber:** Serial number of the meter (secondary address).

**byStatus:** Device status.

**byGEN:** Software version of the device.

**byCounter:** Number of times the master has accessed data of the respective slave.

**usiRecivedAdr:** Received primary address (0-250).

**eMedium:** Medium (see \(E\text{_MBUS\_Medium}\)).

**sMan:** Manufacturer code.

**stValue:** Counter value.

**stMax:** Maximum value.

**stOffset:** Offset.

**byInfo:** Information byte.

nBit7-4: Information about the A/D modules installed in AnDi4
nBit3: Protection bit (1: protection enabled)
nBit2-1: no. of the current measuring input (0: Port1 ... 3: Port4)
nBit0: I/V measurement (1: current measurement)

### VAR_IN_OUT

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>stCom</td>
<td>ST_MBUS_Communication</td>
</tr>
</tbody>
</table>

**stCom:** This structure is used to link the block \(FB\_MBUSKL67810\) with the meter blocks (see \(ST\_MBUS\_Communication\)).

### 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_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.1.27.2 FB_MBUS_REL_PadIn4

This block is used for reading digital inputs from Relay:

- PadIn 4 (4 digital inputs)

The block can only be executed together with the block \texttt{FB_MBUSKL6781}[18].

Functionality of the block [9]

\textbf{VAR_INPUT}

\begin{verbatim}
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL := TRUE;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
bDisabled    : BOOL := FALSE;
\end{verbatim}

\textbf{usiAddress:} Primary address [10] of the meter to be read with this block.

\textbf{stSecAdr:} Secondary address [11] of the meter to be read with this block (see \texttt{ST_MBUS_SecAdr}[205]).

\textbf{eBaudrate:} 300, 2400, 9600 baud (see \texttt{E_MBUS_Baudrate}[197]).

\textbf{bStart:} A positive edge of this input triggers one reading of the meter.

\textbf{bSND_NKE:} TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

\textbf{bReadInit:} If the PLC is restarted, the meter is read once.

\textbf{tMinSendTime:} Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with \textit{bStart}.

\textbf{bDisabled:} TRUE = deselection of the block.

\textbf{VAR_OUTPUT}

\begin{verbatim}
bBusy         : BOOL;
bReady        : BOOL;
bError        : BOOL;
eError        : E_MBUS_ERROR;
dwIdNumber    : DWORD;
byStatus      : BYTE;
\end{verbatim}
byGEN : BYTE;
byCounter : BYTE;
usiRecivedAdr : USINT;
eMedium : E_MBUS_Medium;
sMan : STRING(3);
bDataIn1 : BOOL;
bDataIn2 : BOOL;
bDataIn3 : BOOL;
bDataIn4 : BOOL;

bBusy: The bBusy output is TRUE while the meter is being read.

bReady: The bReady output is TRUE for one cycle, once meter reading is completed.

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

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [200]).

sMan: Manufacturer code.

bDataIn1: Digital input 1

bDataIn2: Digital input 2

bDataIn3: Digital input 3

bDataIn4: Digital input 4

VAR_IN_OUT

stCom : ST_MBUS_Communication;

stCom: This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Communication [202]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>
This block is used for reading energy meters with pulse output from Relay:

- PadPuls M1
- PadPuls M1C

The block can only be executed together with the block FB_MBUSKL6781() \[18\].

Functionality of the block \[9\]

**VAR_INPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>usiAddress</td>
<td>Primary address [10] of the meter to be read with this block.</td>
</tr>
<tr>
<td>stSecAdr</td>
<td>Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAdr [205]).</td>
</tr>
<tr>
<td>eBaudrate</td>
<td>300, 2400, 9600 baud (see E_MBUS_Baudrate [197]).</td>
</tr>
<tr>
<td>bStart</td>
<td>A positive edge of this input triggers one reading of the meter.</td>
</tr>
<tr>
<td>bSND_NKE</td>
<td>TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).</td>
</tr>
<tr>
<td>bReadInit</td>
<td>If the PLC is restarted, the meter is read once.</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.</td>
</tr>
<tr>
<td>usiUnit</td>
<td>Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).</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>bReady</td>
<td></td>
</tr>
<tr>
<td>bError</td>
<td></td>
</tr>
</tbody>
</table>

usiAddress: Primary address \[10\] of the meter to be read with this block.

stSecAdr: Secondary address \[11\] of the meter to be read with this block (see ST_MBUS_SecAdr \[205\]).
bBusy: The bBusy output is TRUE while the meter is being read.

bReady: The bReady output is TRUE for one cycle, once meter reading is completed.

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

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [200]).

sMan: Manufacturer code.

stValue: Meter reading (see ST_MBus_Info [204]).

byNumerator: Pulse value numerator (range 1..255).

byDenominator: Pulse value denominator (range 1..255).

VAR_IN_OUT

stCom: ST_MBUS_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_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.1.27.4 FB_MBUS_REL_PadPulsM2

This block is used for reading energy meters with pulse output from Relay:

- PadPuls M2
- PadPuls M2C

The block can only be executed together with the block FB_MBUSKL6781() [18].

Up to 2 pulse generators can be connected to a PadPuls 2/PadPuls 2C at the same time. The PadPuls 2/PadPuls 2C behaves like 2 independent slaves.

Functionality of the block [9]

**VAR_INPUT**

usiAddress : USINT;
stSecAdr : ST_MBUS_SecAdr;
eBaudrate : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart : BOOL := TRUE;
bSND_NKE : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit : USINT;
bDisabled : BOOL := FALSE;

usiAddress: Primary address [10] of the meter to be read with this block.

stSecAdr: Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAdr [205]).

eBaudrate: 300, 2400, 9600 baud (see E_MBUS_Baudrate [197]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.
tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bReady : BOOL;
bError : BOOL;
eError : E_MBUS_ERROR;
dwIdNumber : DWORD;
byStatus : BYTE;
byGEN : BYTE;
byCounter : BYTE;
usiRecivedAdr : USINT;
eMedium : E_MBUS_Medium;
sMan : STRING(3);
stValue : ST_MBus_Info;
stDateTime : ST_MBus_Info;
stValueDueDay : ST_MBus_Info;
stDateDueDay : ST_MBus_Info;
stDateFutureDueDay : ST_MBus_Info;
byInfo : BYTE;
byNumerator : BYTE;
byDenominator : BYTE;
byPStat : BYTE;

bBusy: The bBusy output is TRUE while the meter is being read.

bReady: The bReady output is TRUE for one cycle, once meter reading is completed.

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

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [200]).

sMan: Manufacturer code.

stValue: Meter reading (see ST_MBus_Info [204]).

stDateTime: Current date (see ST_MBus_Info [204]).

stValueDueDay: Cutoff date meter reading (see ST_MBus_Info [204]).

stDateDueDay: Last cutoff date (see ST_MBus_Info [204]).

stDateFutureDueDay: Future cutoff date (see ST_MBus_Info [204]).

byInfo: Information byte (tariff and sampling method).

byNumerator: Pulse value numerator (1..99).

byDenominator: Pulse value denominator (1..255, 0 -> 256).

byPStat: Port status (current contact state at the port inputs).
VAR_IN_OUT

stCom : ST_MBUS_Communication;

stCom: This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Communication [202]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.27.5 FB_MBUS_REL_PadPulsM4

This block is used for reading energy meters with pulse output from Relay:

-PadPuls M4

-PadPuls M4L

The block can only be executed together with the block FB_MBUSKL6781() [18].

Up to 4 pulse generators can be connected to a PadPuls 4/PadPuls 4L at the same time. The PadPuls 4/PadPuls 4L behaves like 4 independent slaves.

Functionality of the block [9]

VAR_INPUT

usiAddress : USINT;
stSecAdr : ST_MBUS_SecAdr;
eBaudrate : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart : BOOL;
bSND_NKE : BOOL := TRUE;
bReadInit : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit : USINT;
bDisabled : BOOL := FALSE;
usiAddress:  Primary address [10] of the meter to be read with this block.

stSecAdr:  Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAdr [205]).

eBaudrate:  300, 2400, 9600 baud (see E_MBUS_Baudrate [197]).

bStart:  A positive edge of this input triggers one reading of the meter.

bSND_NKE:  TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit:  If the PLC is restarted, the meter is read once.

tMinSendTime:  Standard t=2s. The meter is read again, once the time set here has elapsed. If t=0s the meter is not read and can be read manually with bStart.

usiUnit:  Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2=MW(h) / 3=GW(h).

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>bReady</td>
<td>BOOL;</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL;</td>
</tr>
<tr>
<td>eError</td>
<td>E_MBUS_ERROR;</td>
</tr>
<tr>
<td>dwIdNumber</td>
<td>DWORD;</td>
</tr>
<tr>
<td>byStatus</td>
<td>BYTE;</td>
</tr>
<tr>
<td>byGEN</td>
<td>BYTE;</td>
</tr>
<tr>
<td>byCounter</td>
<td>BYTE;</td>
</tr>
<tr>
<td>usiRecivedAdr</td>
<td>USINT;</td>
</tr>
<tr>
<td>eMedium</td>
<td>E_MBUS_Medium;</td>
</tr>
<tr>
<td>sMan</td>
<td>STRING(3);</td>
</tr>
<tr>
<td>stValue</td>
<td>ST_MBus_Info;</td>
</tr>
<tr>
<td>stDateTime</td>
<td>ST_MBus_Info;</td>
</tr>
<tr>
<td>stValueDueDay</td>
<td>ST_MBus_Info;</td>
</tr>
<tr>
<td>stDateDueDay</td>
<td>ST_MBus_Info;</td>
</tr>
<tr>
<td>stDateFutureDueDay</td>
<td>ST_MBus_Info;</td>
</tr>
<tr>
<td>byInfo</td>
<td>BYTE;</td>
</tr>
<tr>
<td>byNumerator</td>
<td>BYTE;</td>
</tr>
<tr>
<td>byDenominator</td>
<td>BYTE;</td>
</tr>
<tr>
<td>byPStat</td>
<td>BYTE;</td>
</tr>
</tbody>
</table>

bBusy:  The bBusy output is TRUE while the meter is being read.

bReady:  The bReady output is TRUE for one cycle, once meter reading is completed.

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

eError:  In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.

dwIdNumber:  Serial number of the meter (secondary address).

byStatus:  Device status.

byGEN:  Software version of the device.

byCounter:  Number of times the master has accessed data of the respective slave.

usiRecivedAdr:  Received primary address (0-250).

eMedium:  Medium (see E_MBUS_Medium [200]).

sMan:  Manufacturer code.

stValue:  Meter reading (see ST_MBus_Info [204]).

stDateTime:  Current date (see ST_MBus_Info [204]).

stValueDueDay:  Cutoff date meter reading (see ST_MBus_Info [204]).
stDateDueDay: Last cutoff date (see ST_MBus_Info [204]).

stDateFutureDueDay: Future cutoff date (see ST_MBus_Info [204]).

byInfo: Information byte (tariff and sampling method).

byNumerator: Pulse value numerator (1..99).

byDenominator: Pulse value denominator (1..255, 0 -> 256).

byPStat: Port status (current contact state at the port inputs).

VAR_IN_OUT

stCom : ST_MBUS_Communication;

stCom: This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Communication [202]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.28 Saia-Burgess overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [76], FB_MBUS_General_Ext [79] or FB_MBUS_General_Param [83] from the folder "General [74]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [85] can be used to send data to the device (e.g. setting of the primary address).

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Type</th>
<th>Device</th>
<th>Block</th>
</tr>
</thead>
<tbody>
<tr>
<td>Saia-Burgess</td>
<td>Electricity meter</td>
<td>ALD1</td>
<td>FB_MBUS_SBC_ALD1 [165]</td>
</tr>
<tr>
<td></td>
<td>Electricity meter</td>
<td>ALE3</td>
<td>FB_MBUS_SBC_ALE3 [167]</td>
</tr>
<tr>
<td></td>
<td>Electricity meter</td>
<td>AWD3</td>
<td>FB_MBUS_SBC_ALE3 [167]</td>
</tr>
</tbody>
</table>
This block is used to read electricity meters from Saia-Burgess:

- ALD1

The block can only be executed together with the block FB_MBUSKL6781() [18].

Functionality of the block [9]

**VAR_INPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>usiAddress</td>
<td>Primary address [10] of the meter to be read with this block.</td>
</tr>
<tr>
<td>stSecAdr</td>
<td>Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAdr [205]).</td>
</tr>
<tr>
<td>eBaudrate</td>
<td>300, 2400, 9600 baud (see E_MBUS_Baudrate [197]).</td>
</tr>
<tr>
<td>bStart</td>
<td>A positive edge of this input triggers one reading of the meter.</td>
</tr>
<tr>
<td>bSND_NKE</td>
<td>TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).</td>
</tr>
<tr>
<td>bReadInit</td>
<td>If the PLC is restarted, the meter is read once.</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.</td>
</tr>
<tr>
<td>usiUnit</td>
<td>Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).</td>
</tr>
<tr>
<td>bDisabled</td>
<td>TRUE = deselection of the block.</td>
</tr>
</tbody>
</table>

This block is used to read electricity meters from Saia-Burgess:

- ALD1

The block can only be executed together with the block FB_MBUSKL6781() [18].

Functionality of the block [9]

**VAR_INPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>usiAddress</td>
<td>Primary address [10] of the meter to be read with this block.</td>
</tr>
<tr>
<td>stSecAdr</td>
<td>Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAdr [205]).</td>
</tr>
<tr>
<td>eBaudrate</td>
<td>300, 2400, 9600 baud (see E_MBUS_Baudrate [197]).</td>
</tr>
<tr>
<td>bStart</td>
<td>A positive edge of this input triggers one reading of the meter.</td>
</tr>
<tr>
<td>bSND_NKE</td>
<td>TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).</td>
</tr>
<tr>
<td>bReadInit</td>
<td>If the PLC is restarted, the meter is read once.</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.</td>
</tr>
<tr>
<td>usiUnit</td>
<td>Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).</td>
</tr>
<tr>
<td>bDisabled</td>
<td>TRUE = deselection of the block.</td>
</tr>
</tbody>
</table>
VAR_OUTPUT

bBusy : BOOL;
bReady : BOOL;
bError : BOOL;
eError : E_MBUS_ERROR;
dwIdNumber : DWORD;
byStatus : BYTE;
byGEN : BYTE;
byCounter : BYTE;
usiRecivedAdr : USINT;
eMedium : E_MBUS_Medium;
sMan : STRING(3);
stEnergyTotal : ST_MBus_Info;
stEnergyPartial : ST_MBus_Info;
stVoltage : ST_MBus_Info;
stCurrent : ST_MBus_Info;
stPower : ST_MBus_Info;
stReactivPower : ST_MBus_Info;

bBusy: The bBusy output is TRUE while the meter is being read.

bReady: The bReady output is TRUE for one cycle, once meter reading is completed.

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

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [200]).

sMan: Manufacturer code.

stEnergyTotal: Meter reading, energy total (see ST_MBus_Info [204]).

stEnergyPartial: Meter reading, partial energy consumption. This value is resettable (see ST_MBus_Info [204]).

stVoltage: Voltage (see ST_MBus_Info [204]).

stCurrent: Current (see ST_MBus_Info [204]).

stPower: Power (see ST_MBus_Info [204]).

stReactivPower: Reactive power (see ST_MBus_Info [204]).

VAR_IN_OUT

stCom : ST_MBUS_Communication;

stCom: This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Communication [202]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>
This block is used to read electricity meters from Saia-Burgess:

- ALE3
- AWD3

The block can only be executed together with the block FB_MBUSKL6781() [18].

Functionality of the block [9]

VAR_INPUT

usiAddress : USINT;
stSecAdr : ST_MBUS_SecAdr;
eBaudrate : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart : BOOL;
bSEND_NKE : BOOL := TRUE;
bReadInit : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit : USINT;
bDisabled : BOOL := FALSE;

usiAddress: Primary address [10] of the meter to be read with this block.
**stSecAdr:** Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAdr [205]).

**eBaudrate:** 300, 2400, 9600 baud (see E_MBUS_Baudrate [197]).

**bStart:** A positive edge of this input triggers one reading of the meter.

**bSND_NKE:** TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

**bReadInit:** If the PLC is restarted, the meter is read once.

**tMinSendTime:** Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.

**usiUnit:** Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2=MW(h) / 3=GW(h).

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

**VAR_OUTPUT**

```plaintext
bBusy : BOOL;
bReady : BOOL;
bError : BOOL;
eError : E_MBUS_ERROR;
dwIdNumber : DWORD;
byStatus : BYTE;
byGEN : BYTE;
byCounter : BYTE;
usiRecivedAdr : USINT;
eMedium : E_MBUS_Medium;
sMan : STRING(3);
stEnergyT1total : ST_MBus_Info;
stEnergyT1partial : ST_MBus_Info;
stEnergyT2total : ST_MBus_Info;
stEnergyT2partial : ST_MBus_Info;
stVoltageL1 : ST_MBus_Info;
stCurrentL1 : ST_MBus_Info;
stPowerL1 : ST_MBus_Info;
stReactivPowerL1 : ST_MBus_Info;
stVoltageL2 : ST_MBus_Info;
stCurrentL2 : ST_MBus_Info;
stPowerL2 : ST_MBus_Info;
stReactivPowerL2 : ST_MBus_Info;
stVoltageL3 : ST_MBus_Info;
stCurrentL3 : ST_MBus_Info;
stPowerL3 : ST_MBus_Info;
stReactivPowerL3 : ST_MBus_Info;
stCurrTransFactor : ST_MBus_Info;
stPowerTotal : ST_MBus_Info;
stReactivePowerTotal : ST_MBus_Info;
stTariff : ST_MBus_Info;
```

**bBusy:** The bBusy output is TRUE while the meter is being read.

**bReady:** The bReady output is TRUE for one cycle, once meter reading is completed.

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

**eError:** In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.

**dwIdNumber:** Serial number of the meter (secondary address).

**byStatus:** Device status.

**byGEN:** Software version of the device.

**byCounter:** Number of times the master has accessed data of the respective slave.

**usiRecivedAdr:** Received primary address (0-250).

**eMedium:** Medium (see E_MBUS_Medium [200]).

**sMan:** Manufacturer code.
**stEnergyT1total:** Meter reading, energy total tariff 1 (see [ST_MBus_Info](#) [204]).

**stEnergyT1partial:** Meter reading, partial energy consumption tariff 1. This value is resettable (see [ST_MBus_Info](#) [204]).

**stEnergyT2total:** Meter reading, energy total tariff 2 (see [ST_MBus_Info](#) [204]).

**stEnergyT2partial:** Meter reading, partial energy consumption tariff 2. This value is resettable (see [ST_MBus_Info](#) [204]).

**stVoltageL1:** Voltage phase L1 (see [ST_MBus_Info](#) [204]).

**stCurrentL1:** Current phase L1 (see [ST_MBus_Info](#) [204]).

**stPowerL1:** Power phase L1 (see [ST_MBus_Info](#) [204]).

**stReactivPowerL1:** Reactive power phase L1 (see [ST_MBus_Info](#) [204]).

**stVoltageL2:** Voltage phase L2 (see [ST_MBus_Info](#) [204]).

**stCurrentL2:** Current phase L2 (see [ST_MBus_Info](#) [204]).

**stPowerL2:** Power phase L2 (see [ST_MBus_Info](#) [204]).

**stReactivPowerL2:** Reactive power phase L2 (see [ST_MBus_Info](#) [204]).

**stVoltageL3:** Voltage phase L3 (see [ST_MBus_Info](#) [204]).

**stCurrentL3:** Current phase L3 (see [ST_MBus_Info](#) [204]).

**stPowerL3:** Power phase L3 (see [ST_MBus_Info](#) [204]).

**stReactivPowerL3:** Reactive power phase L3 (see [ST_MBus_Info](#) [204]).

**stCurrTransFactor:** Transformer ratio (=0 for ALE3 devices) (see [ST_MBus_Info](#) [204]).

**stPowerTotal:** Total power (see [ST_MBus_Info](#) [204]).

**stReactivePowerTotal:** Total reactive power (see [ST_MBus_Info](#) [204]).

**stTariff:** Current tariff (=0 for AWD3 devices) (see [ST_MBus_Info](#) [204]).

**VAR_IN_OUT**

```c
stCom : ST_MBUS_Communication;
```

**VAR_IN_OUT**

```c
stCom: This structure is used to link the block [FB_MBUSKL6781](#) [202] with the meter blocks (see [ST_MBUS_Communication](#) [202]).
```

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.1.29 Schlumberger overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks [FB_MBUS_General](#) [76], [FB_MBUS_General_Ext](#) [79] or [FB_MBUS_General_Param](#) [83] from the folder "General" [74] should be used. Note that these blocks do not run BC and BX systems. The block [FB_MBUS_General_Send](#) [85] can be used to send data to the device (e.g. setting of the primary address).
<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Type</th>
<th>Device</th>
<th>Block</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schlumberger</td>
<td>Heat meter</td>
<td>Integral-Mk MaXX</td>
<td>FB_MBUS_SLB_MK_MaXX</td>
</tr>
<tr>
<td></td>
<td>Heat meter</td>
<td>CF Echo I</td>
<td>FB_MBUS_SLB_CFEchoI</td>
</tr>
</tbody>
</table>

### 4.1.29.1 FB_MBUS_SLB_CFEchoI

This block is used to read heat meters from Schlumberger:

- CF Echo I

The block can only be executed together with the block `FB_MBUSKL6781() [18]`.

**Functionality of the block [9]**

**VAR_INPUT**

```plaintext
usiAddress : USINT;
stSecAdr   : ST_MBUS_SecAdr;
eBaudrate  : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart     : BOOL := FALSE;
bSND_NKE   : BOOL := TRUE;
bReadInit  : BOOL := TRUE;
tMinSendTime: TIME := t#2s;
usiUnit: USINT;
bDisabled : BOOL := FALSE;
```

**usiAddress**: Primary address [10] of the meter to be read with this block.

**stSecAdr**: Secondary address [11] of the meter to be read with this block (see `ST_MBUS_SecAdr [205]`).

**eBaudrate**: 300, 2400 baud (see `E_MBUS_Baudrate [197]`).

**bStart**: A positive edge of this input triggers one reading of the meter.

**bSND_NKE**: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

**bReadInit**: If the PLC is restarted, the meter is read once.
tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bReady : BOOL;
bError : BOOL;
eError : E_MBUS_ERROR;
dwIdNumber : DWORD;
byStatus : BYTE;
byGEN : BYTE;
usiRecivedAdr : USINT;
eMedium : E_MBUS_Medium;
sMan : STRING(3);
stEnergy : ST_MBus_Info;
stPower : ST_MBus_Info;
stVolume : ST_MBus_Info;
stFlow : ST_MBus_Info;
stForwardTemp : ST_MBus_Info;
stReturnTemp : ST_MBus_Info;
stDiffTemp : ST_MBus_Info;

bBusy: The bBusy output is TRUE while the meter is being read.

bReady: The bReady output is TRUE for one cycle, once meter reading is completed.

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

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [200]).

sMan: Manufacturer code.

stEnergy: Meter reading, energy consumption (see ST_MBus_Info [204]).

stPower: Current energy consumption, power (see ST_MBus_Info [204]).

stVolume: Meter reading, water consumption (see ST_MBus_Info [204]).

stFlow: Current water consumption (see ST_MBus_Info [204]).

stForwardTemp: Flow temperature (see ST_MBus_Info [204]).

stReturnTemp: Return temperature (see ST_MBus_Info [204]).

stDiffTemp: Temperature difference (see ST_MBus_Info [204]).

VAR_IN_OUT

stCom : ST_MBUS_Communication;

stCom: This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Communication [202]).
Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.29.2   FB_MBUS_SLB_MK_MaXX

This block is used to read heat meters from Schlumberger:

- Integral-MK Maxx / Up to 4 additional water meters can be connected to this device.

The block can only be executed together with the block FB_MBUSKL6781() [18].

Functionality of the block [9]

VAR_INPUT

<table>
<thead>
<tr>
<th>var</th>
<th>type</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>usiAddress</td>
<td>USINT</td>
<td>Primary address [10] of the meter to be read with this block.</td>
</tr>
<tr>
<td>stSecAdr</td>
<td>ST_MBUS_SecAdr</td>
<td>Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAdr [205]).</td>
</tr>
<tr>
<td>eBaudrate</td>
<td>E_MBUS_Baudrate</td>
<td>300, 2400 baud (see E_MBUS_Baudrate [197]).</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL</td>
<td>A positive edge of this input triggers one reading of the meter.</td>
</tr>
<tr>
<td>bSND_NKE</td>
<td>BOOL</td>
<td>TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).</td>
</tr>
<tr>
<td>bReadInit</td>
<td>BOOL</td>
<td>TRUE</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>TIME</td>
<td>2 seconds</td>
</tr>
<tr>
<td>usiUnit</td>
<td>USINT</td>
<td></td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL</td>
<td>FALSE</td>
</tr>
</tbody>
</table>
bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard #2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bReady : BOOL;
bError : BOOL;
eError : E_MBUS_ERROR;
dwIdNumber : DWORD;
byStatus : BYTE;
byGEN : BYTE;
usiRecivedAdr : USINT;
eMedium : E_MBUS_Medium;
sMan : STRING(3);
stEnergy : ST_MBus_Info;
stVolume : ST_MBus_Info;
stFlow : ST_MBus_Info;
stForwardTemp : ST_MBus_Info;
stReturnTemp : ST_MBus_Info;
stDiffTemp : ST_MBus_Info;
stVolume1 : ST_MBus_Info;
stVolume2 : ST_MBus_Info;
stVolume3 : ST_MBus_Info;
stVolume4 : ST_MBus_Info;

bBusy: The bBusy output is TRUE while the meter is being read.

bReady: The bReady output is TRUE for one cycle, once meter reading is completed.

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

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [200]).

sMan: Manufacturer code.

stEnergy: Meter reading, energy consumption (see ST_MBus_Info [204]).

stVolume: Meter reading, water consumption (see ST_MBus_Info [204]).

stFlow: Current water consumption (see ST_MBus_Info [204]).

stForwardTemp: Flow temperature (see ST_MBus_Info [204]).

stReturnTemp: Return temperature (see ST_MBus_Info [204]).

stDiffTemp: Temperature difference (see ST_MBus_Info [204]).

stVolume1: Meter reading of additional water meter 1 (see ST_MBus_Info [204]).

stVolume2: Meter reading of additional water meter 2 (see ST_MBus_Info [204]).
stVolume3: Meter reading of additional water meter 3 (see ST_MBus_Info [p. 204]).

stVolume4: Meter reading of additional water meter 4 (see ST_MBus_Info [p. 204]).

VAR_IN_OUT

stCom : ST_MBUS_Communication;

stCom: This structure is used to link the block FB_MBUSKL6781[202] with the meter blocks (see ST_MBUS_Communication[202]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.30 Schneider Electric

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General[76], FB_MBUS_General_Ext[79] or FB_MBUS_General_Param[83] from the folder "General[74]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send()[85] can be used to send data to the device (e.g. setting of the primary address).

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Type</th>
<th>Device</th>
<th>Function block</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schneider Electric</td>
<td>Electricity meter</td>
<td>iEM3135</td>
<td>FB_MBUS_SEC_iEM3135 [p. 175]</td>
</tr>
</tbody>
</table>
This function block is used to read electricity meters from Schneider Electric.

The function block can only be executed together with the function block FB_MBUSKL6781().

Functionality of the function block

**VAR_INPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>usiAddress</td>
<td>USINT</td>
<td>Primary address of the meter to be read with this block.</td>
</tr>
<tr>
<td>stSecAdr</td>
<td>ST_MBUS_SecAdr</td>
<td>Secondary address of the meter to be read with this block (see ST_MBUS_SecAdr).</td>
</tr>
<tr>
<td>eBaudrate</td>
<td>E_MBUS_Baudrate := eMBUS_Baud2400;</td>
<td>300..9600 baud (see E_MBUS_Baudrate).</td>
</tr>
<tr>
<td>bStart</td>
<td>BOOL</td>
<td>A positive edge of this input triggers one reading of the meter.</td>
</tr>
<tr>
<td>bSND_NKE</td>
<td>BOOL := TRUE;</td>
<td></td>
</tr>
<tr>
<td>bReadInit</td>
<td>BOOL := TRUE;</td>
<td></td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>TIME := t#2s;</td>
<td></td>
</tr>
<tr>
<td>bDisabled</td>
<td>BOOL := FALSE;</td>
<td></td>
</tr>
</tbody>
</table>

**usiAddress:** Primary address of the meter to be read with this block.

**stSecAdr:** Secondary address of the meter to be read with this block (see ST_MBUS_SecAdr).

**eBaudrate:** 300..9600 baud (see E_MBUS_Baudrate).

**bStart:** A positive edge of this input triggers one reading of the meter.
**bSND_NKE:** TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

**bReadInit:** If the PLC is restarted, the meter is read once.

**tMinSendTime:** Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with **bStart**.

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

### VAR_OUTPUT

- **bBusy:** The **bBusy** output is TRUE while the meter is being read.
- **bReady:** The **bReady** output is TRUE for one cycle, once meter reading is completed.
- **bError:** This output goes TRUE as soon as an error occurs. This error is described via the variable **eError**.
- **eError:** In the event of an error the output issues an error code (see **E_MBUS_ERROR** [197]). **bError** goes TRUE at the same time.
- **dwIdNumber:** Serial number of the meter (secondary address).
- **byStatus:** Device status.
- **byGEN:** Software version of the device.
- **byCounter:** Number of times the master has accessed data of the respective slave.
- **usiReceivedAdr:** Received primary address (0-250).
- **eMedium:** Medium (see **E_MBUS_Medium** [200]).
- **sMan:** Manufacturer code.
- **stCurrentPhase1:** Current intensity phase 1 (see **ST_MBus_Info** [204]).
- **stCurrentPhase2:** Current intensity phase 2 (see **ST_MBus_Info** [204]).
- **stCurrentPhase3:** Current intensity phase 3 (see **ST_MBus_Info** [204]).
- **stCurrentAverage:** Current average (see **ST_MBus_Info** [204]).
- **stVoltageL1N:** Voltage L1N (see **ST_MBus_Info** [204]).
- **stVoltageL2N:** Voltage L2N (see **ST_MBus_Info** [204]).
- **stVoltageL3N:** Voltage L3N (see **ST_MBus_Info** [204]).
- **stVoltageAverageLN:** Voltage average LN (see **ST_MBus_Info** [204]).
- **stVoltageL1L2:** Voltage L1L2 (see **ST_MBus_Info** [204]).
- **stVoltageL2L3:** Voltage L2L3 (see **ST_MBus_Info** [204]).
- **stVoltageL3L1:** Voltage L3L1 (see **ST_MBus_Info** [204]).
- **stVoltageAverageLL:** Voltage average LL (see **ST_MBus_Info** [204]).
- **stPowerActivePhase1:** Active power phase 1 (see **ST_MBus_Info** [204]).
- **stPowerActivePhase2:** Active power phase 2 (see **ST_MBus_Info** [204]).
- **stPowerActivePhase3:** Active power phase 3 (see **ST_MBus_Info** [204]).
- **stPowerActiveTotal:** Active power total (see **ST_MBus_Info** [204]).
- **stPowerReactiveTotal:** Reactive power total (see **ST_MBus_Info** [204]).
- **stPowerApparentTotal:** Apparent power total (see **ST_MBus_Info** [204]).
- **stPowerFactorTotal:** Power factor total (see **ST_MBus_Info** [204]).
- **stFrequency:** Frequency (see **ST_MBus_Info** [204]).
**stCurrentAverage**: Average value of current intensity (see ST_MBus_Info [204]).

**stVoltageL1N**: Voltage L1-N (see ST_MBus_Info [204]).

**stVoltageL2N**: Voltage L2-N (see ST_MBus_Info [204]).

**stVoltageL3N**: Voltage L3-N (see ST_MBus_Info [204]).

**stVoltageAverageLN**: Average value of voltage L-N (see ST_MBus_Info [204]).

**stVoltageL1L2**: Voltage L1-L2 (see ST_MBus_Info [204]).

**stVoltageL2L3**: Voltage L2-L3 (see ST_MBus_Info [204]).

**stVoltageL3L1**: Voltage L3-L1 (see ST_MBus_Info [204]).

**stVoltageAverageLL**: Average value of voltage L-L (see ST_MBus_Info [204]).

**stPowerActivePhase1**: Effective power phase 1 (see ST_MBus_Info [204]).

**stPowerActivePhase2**: Effective power phase 2 (see ST_MBus_Info [204]).

**stPowerActivePhase3**: Effective power phase 3 (see ST_MBus_Info [204]).

**stPowerActiveTotal**: Total effective power (see ST_MBus_Info [204]).

**stPowerReactiveTotal**: Total reactive power (see ST_MBus_Info [204]).

**stPowerApparentTotal**: Total apparent power (see ST_MBus_Info [204]).

**stFrequency**: Frequency (see ST_MBus_Info [204]).

**VAR_IN_OUT**

```c
stCom : ST_MBUS_Communication;
```

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.32</td>
<td>Tc2_MBus from 3.4.6.0</td>
</tr>
</tbody>
</table>

### 4.1.31 Sensus overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [76], FB_MBUS_General_Ext [79] or FB_MBUS_General_Param [83] from the folder "General [74]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [85] can be used to send data to the device (e.g. setting of the primary address).
### 4.1.31.1 FB_MBUS SEN Pollu

This block is used for reading heat/cold meters from Sensus:

- PolluStat E
- PolluCom E
- PolluTherm

The block can only be executed together with the block FB_MBUSKL6781().

**Functionality of the block [9]**

**VAR_INPUT**

```plaintext
usiAddress : USINT;
stSecAdr   : ST_MBUS_SecAdr;
eBaudrate  : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart     : BOOL;
bSNDF_NKE  : BOOL := TRUE;
bReadInit  : BOOL := TRUE;
tMinSendTime: TIME := t#2s;
usiUnit    : USINT;
bDisabled  : BOOL := FALSE;
```

**usiAddress:** Primary address [10] of the meter to be read with this block.
**stSecAdr:** Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAdr [205]).

**eBaudrate:** 300, 2400, 9600 baud (see E_MBUS_Baudrate [197]).

**bStart:** A positive edge of this input triggers one reading of the meter.

**bSND_NKE:** TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

**bReadInit:** If the PLC is restarted, the meter is read once.

**tMinSendTime:** Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.

**usiUnit:** Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2=MW(h) / 3=GW(h).

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

### VAR_OUTPUT

- bBusy : BOOL;
- bReady : BOOL;
- bError : BOOL;
- eError : E_MBUS_ERROR;
- dwIdNumber : DWORD;
- byStatus : BYTE;
- byGEN : BYTE;
- byCounter : BYTE;
- usiRecivedAdr : USINT;
- eMedium : E_MBUS_Medium;
- sMan : STRING(3);
- stEnergy : ST_MBus_Info;
- stPower : ST_MBus_Info;
- stVolume : ST_MBus_Info;
- stFlow : ST_MBus_Info;
- stForwardTemp : ST_MBus_Info;
- stReturnTemp : ST_MBus_Info;
- stDiffTemp : ST_MBus_Info;

**bBusy:** The bBusy output is TRUE while the meter is being read.

**bReady:** The bReady output is TRUE for one cycle, once meter reading is completed.

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

**eError:** In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.

**dwIdNumber:** Serial number of the meter (secondary address).

**byStatus:** Device status.

**byGEN:** Software version of the device.

**byCounter:** Number of times the master has accessed data of the respective slave.

**usiRecivedAdr:** Received primary address (0-250).

**eMedium:** Medium (see E_MBUS_Medium [201]).

**sMan:** Manufacturer code.

**stEnergy:** Meter reading, energy consumption (see ST_MBus_Info [204]).

**stPower:** Current energy consumption, power (see ST_MBus_Info [204]).

**stVolume:** Meter reading, water consumption (see ST_MBus_Info [204]).

**stFlow:** Current water consumption (see ST_MBus_Info [204]).

**stForwardTemp:** Flow temperature (see ST_MBus_Info [204]).
Programming

stReturnTemp: Return temperature (see ST_MBus_Info [204]).

stDiffTemp: Temperature difference (see ST_MBus_Info [204]).

VAR_IN_OUT

stCom: This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Communication [202]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.31.2 FB_MBUS_SEN_Water

This block is used to read water meters from Sensus.

The block can only be executed together with the block FB_MBUSKL6781() [18].

Functionality of the block [9]

VAR_INPUT

usiAddress : USINT;
stSecAdr : ST_MBUS_SecAdr;
eBaudrate : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart : BOOL;
bSND_NKE : BOOL := TRUE;
bReadInit : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
bDisabled : BOOL := FALSE;

usiAddress: Primary address [10] of the meter to be read with this block.

stSecAdr: Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAdr [205]).

eBaudrate: 300, 2400, 9600 baud (see E_MBUS_Baudrate [197]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).
bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the
meter is not read and can be read manually with bStart.

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 while the meter is being read.</td>
</tr>
<tr>
<td>bReady</td>
<td>The bReady output is TRUE for one cycle, once meter reading is completed.</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>In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.</td>
</tr>
<tr>
<td>dwIdNumber</td>
<td>Serial number of the meter (secondary address).</td>
</tr>
<tr>
<td>byStatus</td>
<td>Device status.</td>
</tr>
<tr>
<td>byGEN</td>
<td>Software version of the device.</td>
</tr>
<tr>
<td>byCounter</td>
<td>Number of times the master has accessed data of the respective slave.</td>
</tr>
<tr>
<td>usiRecivedAddr</td>
<td>Received primary address (0-250).</td>
</tr>
<tr>
<td>eMedium</td>
<td>Medium (see E_MBUS_Medium [200]).</td>
</tr>
<tr>
<td>sMan</td>
<td>Manufacturer code.</td>
</tr>
<tr>
<td>stVolume</td>
<td>Meter reading, water consumption (see ST_MBus_Info [204]).</td>
</tr>
<tr>
<td>stFlow</td>
<td>Current water consumption (see ST_MBus_Info [204]).</td>
</tr>
</tbody>
</table>

**VAR_IN_OUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stCom</td>
<td>This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Communication [202]).</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_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.1.32 Sontex overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [76], FB_MBUS_General_Ext [79] or FB_MBUS_General_Param [83] from the folder "General [74]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [85] can be used to send data to the device (e.g. setting of the primary address).

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Type</th>
<th>Device</th>
<th>Block</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sontex</td>
<td>Heat/cold meter</td>
<td>Supercal 531 (standard values)</td>
<td>FB_MBUS_SON_Supercal531</td>
</tr>
</tbody>
</table>

4.1.32.1 FB_MBUS_SON_Supercal531

This block is used for reading heat/cold meters from Sontex:
- Supercal 531

The block can only be executed together with the block FB_MBUSKL6781() [18].

Functionality of the block [9]
VAR_INPUT

usiAddress : USINT;
stSecAdr : ST_MBUS_SecAdr;
eBaudrate : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart : BOOL;
bSND_NKE : BOOL := TRUE;
bReadInit : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit : USINT;
bDisabled : BOOL := FALSE;

usiAddress: Primary address [10] of the meter to be read with this block.

stSecAdr: Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAdr [205]).

eBaudrate: 300, 2400, 9600 baud (see E_MBUS_Baudrate [197]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with bStart.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2=MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

bBusy : BOOL;
bReady : BOOL;
bError : BOOL;
eError : E_MBUS_ERROR;
dwIdNumber : DWORD;
byStatus : BYTE;
byGEN : BYTE;
byCounter : BYTE;
usiRecivedAdr : USINT;
eMedium : E_MBUS_Medium;
mMan : STRING(3);
stEnergy : ST_Mbus_Info;
stPower : ST_Mbus_Info;
stVolume : ST_Mbus_Info;
stFlow : ST_Mbus.Info;
stForwardTemp : ST_Mbus_Info;
stReturnTemp : ST_Mbus_Info;
stEnergyTariff1 : ST_Mbus_Info;
stVolumeTariff1 : ST_Mbus_Info;
stEnergyTariff2 : ST_Mbus_Info;
stVolumeTariff2 : ST_Mbus_Info;
stTypTariff1 : ST_Mbus_Info;
stLimitLowTariff1 : ST_Mbus_Info;
stLimitHighTariff1 : ST_Mbus_Info;
stTypTariff2 : ST_Mbus_Info;
stLimitLowTariff2 : ST_Mbus_Info;
stLimitHighTariff2 : ST_Mbus_Info;
stDeviceError : ST_Mbus_Info;

bBusy: The bBusy output is TRUE while the meter is being read.

bReady: The bReady output is TRUE for one cycle, once meter reading is completed.

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

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). bError goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.
byGEN: Software version of the device.
byCounter: Number of times the master has accessed data of the respective slave.
usiRecivedAdr: Received primary address (0-250).
eMedium: Medium (see E_MBUS_Medium [200]).
sMan: Manufacturer code.
stEnergy: Meter reading, energy consumption (see ST_MBus_Info [204]).
stPower: Current energy consumption, power (see ST_MBus_Info [204]).
stVolume: Meter reading, water consumption (see ST_MBus_Info [204]).
stFlow: Current water consumption (see ST_MBus_Info [204]).
stForwardTemp: Flow temperature (see ST_MBus_Info [204]).
stReturnTemp: Return temperature (see ST_MBus_Info [204]).
stEnergyTariff1: Meter reading, energy consumption, tariff 1 (see ST_MBus_Info [204]).
stVolumeTariff1: Meter reading, water consumption, tariff 1 (see ST_MBus_Info [204]).
stEnergyTariff2: Meter reading, energy consumption, tariff 2 (see ST_MBus_Info [204]).
stVolumeTariff2: Meter reading, water consumption, tariff 2 (see ST_MBus_Info [204]).
stTypTariff1: Type tariff 1 (see ST_MBus_Info [204]).
stLimitLowTariff1: Lower limit value tariff 1 (see ST_MBus_Info [204]).
stLimitHighTariff1: Upper limit value tariff 1 (see ST_MBus_Info [204]).
stTypTariff2: Type tariff 2 (see ST_MBus_Info [204]).
stLimitLowTariff2: Lower limit value tariff 2 (see ST_MBus_Info [204]).
stLimitHighTariff2: Upper limit value tariff 2 (see ST_MBus_Info [204]).

**VAR_IN_OUT**

```c
stCom : ST_MBUS_Communication;
```

**stCom**: This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Communication [202]).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

**4.1.33 TIP**

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [76], FB_MBUS_General_Ext [79] or FB_MBUS_General_Param [83] from the folder "General [74]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [85] can be used to send data to the device (e.g. setting of the primary address).
4.1.33.1 **FB_MBUS_TIP_SINUS85M**

This function block is used to read electricity meters from Thüringer Industrie Produkte GmbH.

The function block can only be executed together with the function block **FB_MBUSKL6781() [» 18]**.

**Functionality of the function block [» 9]**

**VAR_INPUT**

<table>
<thead>
<tr>
<th>varname</th>
<th>type</th>
</tr>
</thead>
<tbody>
<tr>
<td>usiAddress</td>
<td>USINT;</td>
</tr>
<tr>
<td>stSecAddr</td>
<td>ST_MBUS_SecAddr;</td>
</tr>
<tr>
<td>eBaudrate</td>
<td>E_MBUS_Baudrate := eMBUS_Baud2400;</td>
</tr>
</tbody>
</table>
**bStart**: BOOL
**bSND_NKE**: BOOL := TRUE;
**bReadInit**: BOOL := TRUE;
**tMinSendTime**: TIME := t#2s;
**bDisabled**: BOOL := FALSE;

**usiAddress**: Primary address [10] of the meter to be read with this block.

**stSecAdr**: Secondary address [11] of the meter to be read with this block (see ST_MBUS_SecAdr [205]).

**eBaudrate**: 300, 600, 1200, 2400, 4800, 9600 baud (see E_MBUS_Baudrate [197]).

**bStart**: A positive edge of this input triggers one reading of the meter.

**bSND_NKE**: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

**bReadInit**: If the PLC is restarted, the meter is read once.

**tMinSendTime**: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with **bStart**.

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

**VAR_OUTPUT**

**bBusy**: BOOL
**bReady**: BOOL
**bError**: BOOL
**eError**: E_MBUS_ERROR
**dwIdNumber**: DWORD
**byStatus**: BYTE
**byGEN**: BYTE
**byCounter**: BYTE
**usiRecivedAdr**: USINT
**eMedium**: E_MBUS_Medium
**sMan**: STRING(3)
**stActivePowerTariff1**: ST_MBus_Info
**stActivePowerTariff2**: ST_MBus_Info
**stApparentPowerL1**: ST_MBus_Info
**stApparentPowerL2**: ST_MBus_Info
**stApparentPowerL3**: ST_MBus_Info
**stApparentPowerTotal**: ST_MBus_Info
**stActivePowerL1**: ST_MBus_Info
**stActivePowerL2**: ST_MBus_Info
**stActivePowerL3**: ST_MBus_Info
**stActivePowerTotal**: ST_MBus_Info
**stReactivePowerL1**: ST_MBus_Info
**stReactivePowerL2**: ST_MBus_Info
**stReactivePowerL3**: ST_MBus_Info
**stReactivePowerTotal**: ST_MBus_Info
**stVoltageL1**: ST_MBus_Info
**stVoltageL2**: ST_MBus_Info
**stVoltageL3**: ST_MBus_Info
**stPowerLineFrequency**: ST_MBus_Info
**stCurrentTariff**: ST_MBus_Info
**stCurrentL1**: ST_MBus_Info
**stCurrentL2**: ST_MBus_Info
**stCurrentL3**: ST_MBus_Info
**stCurrentTotal**: ST_MBus_Info
**stPowerFactorL1**: ST_MBus_Info
**stPowerFactorL2**: ST_MBus_Info
**stPowerFactorL3**: ST_MBus_Info
**stPowerFactorTotal**: ST_MBus_Info

**bBusy**: The **bBusy** output is TRUE while the meter is being read.

**bReady**: The **bReady** output is TRUE for one cycle, once meter reading is completed.

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

**eError**: In the event of an error the output issues an error code (see E_MBUS_ERROR [197]). **bError** goes TRUE at the same time.

**dwIdNumber**: Serial number of the meter (secondary address).
byStatus: Device status.
byGEN: Software version of the device.
byCounter: Number of times the master has accessed data of the respective slave.
usiRecivedAdr: Received primary address (0-250).
eMedium: Medium (see E_MBUS_Medium [p. 201]).
sMan: Manufacturer code.

stActivePowerTariff1: Active energy import tariff 1 (see ST_MBus_Info [p. 204]).
stActivePowerTariff2: Active energy import tariff 2 (see ST_MBus_Info [p. 204]).
stApparentPowerL1: Current apparent power L1 (see ST_MBus_Info [p. 204]).
stApparentPowerL2: Current apparent power L2 (see ST_MBus_Info [p. 204]).
stApparentPowerL3: Current apparent power L3 (see ST_MBus_Info [p. 204]).
stApparentPowerTotal: Current total apparent power (see ST_MBus_Info [p. 204]).

stActivePowerL1: Current effective power phase L1 (see ST_MBus_Info [p. 204]).
stActivePowerL2: Current effective power phase L2 (see ST_MBus_Info [p. 204]).
stActivePowerL3: Current effective power phase L3 (see ST_MBus_Info [p. 204]).
stActivePowerTotal: Current total effective power (see ST_MBus_Info [p. 204]).

stReactivePowerL1: Current reactive power phase L1 (see ST_MBus_Info [p. 204]).
stReactivePowerL2: Current reactive power phase L2 (see ST_MBus_Info [p. 204]).
stReactivePowerL3: Current reactive power phase L3 (see ST_MBus_Info [p. 204]).
stReactivePowerTotal: Current total reactive power (see ST_MBus_Info [p. 204]).

stVoltageL1: Current voltage phase L1 (see ST_MBus_Info [p. 204]).
stVoltageL2: Current voltage phase L2 (see ST_MBus_Info [p. 204]).
stVoltageL3: Current voltage phase L3 (see ST_MBus_Info [p. 204]).

stPowerLineFrequency: Current mains frequency (see ST_MBus_Info [p. 204]).

stCurrentTariff: Current tariff (see ST_MBus_Info [p. 204]).

stCurrentL1: Current phase L1 current (see ST_MBus_Info [p. 204]).

stCurrentL2: Current phase L2 current (see ST_MBus_Info [p. 204]).

stCurrentL3: Current phase L3 current (see ST_MBus_Info [p. 204]).

stCurrentTotal: Current total current (see ST_MBus_Info [p. 204]).

stPowerFactorL1: Current form factor phase L1 (cos Phi) (see ST_MBus_Info [p. 204]).

stPowerFactorL2: Current form factor phase L2 (cos Phi) (see ST_MBus_Info [p. 204]).

stPowerFactorL3: Current form factor phase L3 (cos Phi) (see ST_MBus_Info [p. 204]).

stPowerFactorTotal: Current total form factor (cos Phi) (see ST_MBus_Info [p. 204]).
VAR_IN_OUT

stCom : ST_MBUS_Communication;

stCom: This structure is used to link the block `FB_MBUSKL6781()` with the meter blocks (see `ST_MBUS_Communication`).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.32</td>
<td>Tc2_MBus from 3.4.6.0</td>
</tr>
</tbody>
</table>

4.1.34 Zenner overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks `FB_MBUS_General`, `FB_MBUS_General_Ext` or `FB_MBUS_General_Param` from the folder "General" should be used. Note that these blocks do not run BC and BX systems. The block `FB_MBUS_General_Send()` can be used to send data to the device (e.g. setting of the primary address).

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Type</th>
<th>Device</th>
<th>Block</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zenner</td>
<td>Arithmetic unit</td>
<td>multidataWR3</td>
<td><code>FB_MBUS_ZRM_multidataWR3</code></td>
</tr>
<tr>
<td></td>
<td>Heat meter</td>
<td>zelsiusZR</td>
<td><code>FB_MBUS_ZRM_zelsiusZR</code></td>
</tr>
</tbody>
</table>
This block is used to read arithmetic units from Zenner.

The block can only be executed together with the block \texttt{FB_MBUSKL6781} [18].

**Functionality of the block [9]**

**VAR_INPUT**

\begin{tabular}{ll}
\texttt{usiAddress} & : USINT; \\
\texttt{stSecAdr} & : \texttt{ST_MBUS\_SecAdr}; \\
\texttt{eBaudrate} & : \texttt{E\_MBUS\_Baudrate} := \texttt{eMBUS\_Baud2400}; \\
\texttt{bStart} & : BOOL; \\
\texttt{bSND\_NKE} & : BOOL := \text{TRUE}; \\
\texttt{bReadInit} & : BOOL := \text{TRUE}; \\
\texttt{tMinSendTime} & : \texttt{TIME} := t\#2s; \\
\texttt{bDisabled} & : BOOL := \text{FALSE}; \\
\end{tabular}

\texttt{usiAddress}: Primary address [10] of the meter to be read with this block.

\texttt{stSecAdr}: Secondary address [11] of the meter to be read with this block (see \texttt{ST\_MBUS\_SecAdr} [205]).

\texttt{eBaudrate}: 300, 2400, 9600 baud (see \texttt{E\_MBUS\_Baudrate} [197]).

\texttt{bStart}: A positive edge of this input triggers one reading of the meter.

\texttt{bSND\_NKE}: \text{TRUE} initializes the meter for each read operation and sets the meter to the first telegram (SND\_NKE).

\texttt{bReadInit}: If the PLC is restarted, the meter is read once.
tMinSendTime: Standard #12s. The meter is read again, once the time set here has elapsed. If #0s the meter is not read and can be read manually with bStart.

bDisabled: TRUE = deselection of the block.

**VAR_OUTPUT**

- bBusy: The bBusy output is TRUE while the meter is being read.
- bReady: The bReady output is TRUE for one cycle, once meter reading is completed.
- bError: this output goes TRUE as soon as an error occurs. This error is described via the variable eError.
- eError: In the event of an error the output issues an error code (see E_MBUS_ERROR). bError goes TRUE at the same time.
- dwIdNumber: Serial number of the meter (secondary address).
- byStatus: Device status.
- byGEN: Software version of the device.
- byCounter: Number of times the master has accessed data of the respective slave.
- usiRecivedAdr: Received primary address (0-250).
- eMedium: Medium (see E_MBUS_Medium).
- sMan: Manufacturer code.
- stEnergy: Current heat energy (see ST_MBus_Info).
- stVolumeInput1: Volume input 1 (see ST_MBus_Info).
- stVolumeInput2: Volume input 2 (see ST_MBus_Info).
- stDeviceError: Error status MBus output (see ST_MBus_Info).
- stTimepoint: Cutoff date (date and time of the next cutoff date).
- stEnergyTimepoint: Heat energy on the cutoff date.
stCounterInput1: Count value input 1 on the cutoff date (see ST_MBus_Info [204]).
stCounterInput2: Count value input 2 on the cutoff date (see ST_MBus_Info [204]).

stVolume: Volume (see ST_MBus_Info [204]).
stFlow: Instantaneous flow rate (see ST_MBus_Info [204]).
stPower: Power (see ST_MBus_Info [204]).
stForwardTemp: Flow temperature (see ST_MBus_Info [204]).
stReturnTemp: Return temperature (see ST_MBus_Info [204]).
stDeviceClock: Current meter time (see ST_MBus_Info [204]).
stBatteryEndDate: Expected battery shelf life (see ST_MBus_Info [204]).
stFlowMaxTimepoint: Timing of maximum flow rate reading (absolute) (see ST_MBus_Info [204]).
stFlowMax: Maximum flow rate (absolute) (see ST_MBus_Info [204]).
stPowerMaxTimepoint: Timing of maximum power reading (absolute) (see ST_MBus_Info [204]).
stPowerMax: Maximum power (absolute) (see ST_MBus_Info [204]).

VAR_IN_OUT

stCom : ST_MBUS_Communication;

stCom: This structure is used to link the block FB_MBUSKL6781([202]) with the meter blocks (see ST_MBUS_Communication [202]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.1.34.2 FB_MBUS_ZRM_zelsiusZR

This block is used to read heat meters from Zenner.
The block can only be executed together with the block `FB_MBUSKL6781` [18].

Functionality of the block [9]

VAR_INPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>usiAddress</td>
<td>Primary address [10] of the meter to be read with this block.</td>
</tr>
<tr>
<td>stSecAdr</td>
<td>Secondary address [11] of the meter to be read with this block (see <code>ST_MBUS_SecAdr</code> [205]).</td>
</tr>
<tr>
<td>eBaudrate</td>
<td>300, 2400, 9600 baud (see <code>E_MBUS_Baudrate</code> [197]).</td>
</tr>
<tr>
<td>bStart</td>
<td>A positive edge of this input triggers one reading of the meter.</td>
</tr>
<tr>
<td>bSND_NKE</td>
<td>TRUE initializes the meter for each read operation and sets the meter to the first telegram (<code>SND_NKE</code>).</td>
</tr>
<tr>
<td>bReadInit</td>
<td>If the PLC is restarted, the meter is read once.</td>
</tr>
<tr>
<td>tMinSendTime</td>
<td>Standard t#2s. The meter is read again, once the time set here has elapsed.</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 <code>bBusy</code> output is TRUE while the meter is being read.</td>
</tr>
<tr>
<td>bReady</td>
<td>The <code>bReady</code> output is TRUE for one cycle, once meter reading is completed.</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>In the event of an error the output issues an error code (see <code>E_MBUS_ERROR</code> [197]). <code>bError</code> goes TRUE at the same time.</td>
</tr>
<tr>
<td>dwIdNumber</td>
<td>Serial number of the meter (secondary address).</td>
</tr>
<tr>
<td>byStatus</td>
<td>Device status.</td>
</tr>
<tr>
<td>byGEN</td>
<td>Software version of the device.</td>
</tr>
<tr>
<td>byCounter</td>
<td>Number of times the master has accessed data of the respective slave.</td>
</tr>
<tr>
<td>usiRecivedAdr</td>
<td>Received primary address (0-250).</td>
</tr>
</tbody>
</table>

usiAddress: Primary address [10] of the meter to be read with this block.

stSecAdr: Secondary address [11] of the meter to be read with this block (see `ST_MBUS_SecAdr` [205]).

eBaudrate: 300, 2400, 9600 baud (see `E_MBUS_Baudrate` [197]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (`SND_NKE`).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with `bStart`.

bDisabled: TRUE = deselection of the block.
eMedium: Medium (see E_MBUS_Medium [201]).

sMan: Manufacturer code.

stEnergy: Current heat energy (see ST_MBus_Info [204]).

stVolumeInput1: Volume input 1 (see ST_MBus_Info [204]).

stVolumeInput2: Volume input 2 (see ST_MBus_Info [204]).

stDeviceError: Error status MBus output (see ST_MBus_Info [204]).

stTimepoint: Cutoff date (date and time of the next cutoff date) (see ST_MBus_Info [204]).

stEnergyTimepoint: Heat energy on the cutoff date (see ST_MBus_Info [204]).

stCounterInput1: Count value input 1 on the cutoff date (see ST_MBus_Info [204]).

stCounterInput2: Count value input 2 on the cutoff date (see ST_MBus_Info [204]).

VAR_IN_OUT

stCom : ST_MBUS_Communication;

stCom: This structure is used to link the block FB_MBUSKL6781() [202] with the meter blocks (see ST_MBUS_Communication [202]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>
4.1.35 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>0x000</td>
<td>0</td>
<td>eMBUS_no_error</td>
<td>No error is present at the block. The block is currently not querying a counter.</td>
</tr>
<tr>
<td>0x000</td>
<td>1</td>
<td>eMBUS_busy</td>
<td>The block is querying a meter.</td>
</tr>
<tr>
<td>0x000</td>
<td>3</td>
<td>eMBUS_Disabled</td>
<td>The block is deselected.</td>
</tr>
<tr>
<td>0x000</td>
<td>4</td>
<td>eMBUS_FBKL6781_Disabled</td>
<td>The function block FB_MBUSKL6781() is deselected.</td>
</tr>
<tr>
<td>0x006</td>
<td>101</td>
<td>eMBUSERROR_CIField_wrong_72hex_expected</td>
<td>The 7th byte in the response telegram contains the CI field. In this byte the hexadecimal number 72 is expected. It stands for variable data structure, low byte is sent first. Only this data structure is supported.</td>
</tr>
<tr>
<td>0x006</td>
<td>102</td>
<td>eMBUSERROR_no_data_received</td>
<td>No data was received.</td>
</tr>
<tr>
<td>0x006</td>
<td>103</td>
<td>eMBUSERROR_error_checksum</td>
<td>The response telegram includes a checksum (sum of all bytes from byte 5). The received checksum does not match the calculated checksum. This happens if the protocol was not received cleanly (e.g. in the event of interference on the cable or if the cable is too long).</td>
</tr>
<tr>
<td>0x006</td>
<td>104</td>
<td>eMBUSERROR_error_in_head_data</td>
<td>The first 4 bytes are not included in the checksum. These 4 bytes are monitored separately.</td>
</tr>
<tr>
<td>0x006</td>
<td>105</td>
<td>eMBUSERROR_usiAddress_over_250</td>
<td>Addresses higher than 250 are not permitted. The input usiAddress of the meter block was assigned a value higher than 250 (exception: Address 254. This address can be used if only one meter is connected).</td>
</tr>
<tr>
<td>0x006</td>
<td>106</td>
<td>eMBUSERROR_send_error</td>
<td>Error message for error during sending.</td>
</tr>
<tr>
<td>0x006</td>
<td>108</td>
<td>eMBUSERROR_received_address_wrong</td>
<td>Received address does not match the sent address.</td>
</tr>
<tr>
<td>0x006</td>
<td>109</td>
<td>eMBUSERROR_cMBUS_MaxCom_below_1</td>
<td>Reserve.</td>
</tr>
<tr>
<td>0x006</td>
<td>110</td>
<td>eMBUSERROR_iComId_over_cMBUS_MaxCom</td>
<td>Reserve.</td>
</tr>
<tr>
<td>0x006</td>
<td>111</td>
<td>eMBUSERROR_manufacturer_sign_wrong</td>
<td>The response telegram includes a manufacturer code. This code is allocated to the counter blocks. This message appears if the received manufacturer code does not match the block used.</td>
</tr>
<tr>
<td>0x007</td>
<td>112</td>
<td>eMBUSERROR_baudrate_wrong</td>
<td>Input eBaudrate of the block was assigned invalid values. Only E_MBUS_Baudrate[197] are allowed.</td>
</tr>
<tr>
<td>0x007</td>
<td>113</td>
<td>eMBUSERROR_ReceiveBufferFull</td>
<td>The receive buffer of the serial interface is full. This may happen with long telegrams and/or long cycle times. The PLC is unable to read the data quick enough from the receive buffer, resulting in data loss. The situation may be resolved by reducing the cycle time.</td>
</tr>
<tr>
<td>0x007</td>
<td>114</td>
<td>eMBUSERROR_E5hex_no_received</td>
<td>No single character E5 hexadecimal was received after initialization of the meter.</td>
</tr>
<tr>
<td>0x007</td>
<td>115</td>
<td>eMBUSERROR_no_stop_character</td>
<td>No end character in the data array.</td>
</tr>
<tr>
<td>0x007</td>
<td>116</td>
<td>eMBUSERROR_length_wrong</td>
<td>Number of received characters &lt;&gt; the length field.</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>0x007</td>
<td>5</td>
<td>eMBUSERROR_wrong_terminal</td>
<td>Incorrect terminal connected</td>
</tr>
<tr>
<td>0x007</td>
<td>6</td>
<td>eMBUSERROR_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>0x007</td>
<td>7</td>
<td>eMBUSERROR_stSecAdr_udildNumber_wrong</td>
<td>The input variable stSecAdr.udildNumber is not assigned.</td>
</tr>
<tr>
<td>0x007</td>
<td>8</td>
<td>eMBUSERROR_missing_parts_telem</td>
<td>Not all telegram values were received.</td>
</tr>
<tr>
<td>0x007</td>
<td>9</td>
<td>eMBUSERROR_no_stop_character_received</td>
<td>No stop characters were received (16hex).</td>
</tr>
<tr>
<td>0x007</td>
<td>A</td>
<td>eMBUSERROR_too_many_characters</td>
<td>Too many characters were received.</td>
</tr>
<tr>
<td>0x007</td>
<td>B</td>
<td>eMBUSERROR_TimeOut_FB_KL6781</td>
<td>Timeout FB_KL6781.</td>
</tr>
<tr>
<td>0x007</td>
<td>C</td>
<td>eMBUSERROR_TimeOut_MeterFB</td>
<td>Meter block timeout.</td>
</tr>
<tr>
<td>0x00C</td>
<td>9</td>
<td>eMBUSERROR_COM_PARAMETERCHANGED</td>
<td>Input parameters have changed during reception.</td>
</tr>
<tr>
<td>0x00C</td>
<td>A</td>
<td>eMBUSERROR_COM_TXBUFFOVERRUN</td>
<td>String &gt; transfer buffer.</td>
</tr>
<tr>
<td>0x00D</td>
<td>2</td>
<td>eMBUSERROR_COM_STRINGOVERRUN</td>
<td>End of the string.</td>
</tr>
<tr>
<td>0x00D</td>
<td>3</td>
<td>eMBUSERROR_COM_ZEROCHARINVALID</td>
<td>String may not contain any zero characters.</td>
</tr>
<tr>
<td>0x00D</td>
<td>C</td>
<td>eMBUSERROR_COM_INVALIDPOINTER</td>
<td>Invalid data pointer, e.g. zero.</td>
</tr>
<tr>
<td>0x00D</td>
<td>D</td>
<td>eMBUSERROR_COM_INVALIDRXPOINTER</td>
<td>Invalid data pointer for ReceiveData.</td>
</tr>
<tr>
<td>0x00D</td>
<td>E</td>
<td>eMBUSERROR_COM_INVALIDRXLENGTH</td>
<td>Invalid length for ReceiveData e.g. zero.</td>
</tr>
<tr>
<td>0x00D</td>
<td>F</td>
<td>eMBUSERROR_COM_DATASIZEOVERRUN</td>
<td>End of the data block.</td>
</tr>
<tr>
<td>0x100</td>
<td>1</td>
<td>eMBUSERROR_COM_INVALIDBAUDRATE</td>
<td>Invalid baud rate.</td>
</tr>
<tr>
<td>0x100</td>
<td>2</td>
<td>eMBUSERROR_COM_INVALIDNUMDATABITS</td>
<td>Invalid data bits.</td>
</tr>
<tr>
<td>0x100</td>
<td>3</td>
<td>eMBUSERROR_COM_INVALIDNUMSTOPBITS</td>
<td>Invalid stop bits.</td>
</tr>
<tr>
<td>0x100</td>
<td>4</td>
<td>eMBUSERROR_COM_INVALIDPARITY</td>
<td>Invalid parity.</td>
</tr>
<tr>
<td>0x100</td>
<td>5</td>
<td>eMBUSERROR_COM_INVALIDHANDSHAKE</td>
<td>Invalid handshake.</td>
</tr>
<tr>
<td>0x100</td>
<td>6</td>
<td>eMBUSERROR_COM_INVALIDNUMREGISTERS</td>
<td>Invalid num register.</td>
</tr>
<tr>
<td>0x100</td>
<td>7</td>
<td>eMBUSERROR_COM_INVALIDREGISTER</td>
<td>Invalid register.</td>
</tr>
<tr>
<td>0x100</td>
<td>8</td>
<td>eMBUSERROR_COM_TIMEOUT</td>
<td>COM timeout.</td>
</tr>
</tbody>
</table>
4.2 DUTs

4.2.1 Enums

4.2.1.1 E_MBUS_Baudrate

Configurable baud rates

```plaintext
TYPE E_MBUS_Baudrate :
{
  eMBUS_NoBaudrate := 0,
  eMBUS_Baud300    := 30,
  eMBUS_Baud600    := 60,
  eMBUS_Baud1200   := 120,
  eMBUS_Baud2400   := 240,
  eMBUS_Baud4800   := 480,
  eMBUS_Baud9600   := 960
}
END_TYPE
```

- eMBUS_NoBaudrate: Standard baud rate = 2400 baud
- eMBUS_Baud300: 300 baud
- eMBUS_Baud600: 600 baud
- eMBUS_Baud1200: 1200 baud
- eMBUS_Baud2400: 2400 baud
- eMBUS_Baud4800: 4800 baud
- eMBUS_Baud9600: 9600 baud

M-Bus counters are generally supplied with 2400 baud.
The KL6781 supports 300, 600, 1200, 2400, 4800, 9600 baud.
The KL6781 interface is set to eBaudrate when the PLC starts or when the input eBaudrate changes.
Not all M-Bus devices support baud rates above 2400.

### Baudraten der M-Bus-Masterklemme KL6781 von Beckhoff

<table>
<thead>
<tr>
<th>KL6781</th>
<th>300</th>
<th>600</th>
<th>1.200</th>
<th>2.400</th>
<th>4.800</th>
<th>9.600</th>
</tr>
</thead>
</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_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.1.2 E_MBUS_Error

Error messages.

```plaintext
TYPE E_MBUS_Error :
{
  eMBUS_no_error          := 0,
  eMBUS_busy              := 1,
  eMBUS_Disabled          := 3,
  eMBUS_FBKL6781_Disabled := 4,
}
```
eMBUSERROR_CIField_wrong_72hex_expected := 101,
eMBUSERROR_no_data_received := 102,
eMBUSERROR_error_checksum := 103,
eMBUSERROR_error_in_head_data := 104,
eMBUSERROR_usiAddress_over_250 := 105,
eMBUSERROR_send_error := 106,
eMBUSERROR_received_address_wrong := 108,
eMBUSERROR_cMBUS_MaxCom_below_1 := 109,
eMBUSERROR_iComId_over_cMBUS_MaxCom := 110,
eMBUSERROR_manufacturer_sign_wrong := 111,
eMBUSERROR_baudrate_wrong := 112,
eMBUSERROR_ReceiveBufferFull := 113,
eMBUSERROR_E5hex_no_received := 114,
eMBUSERROR_no_stop_character := 115,
eMBUSERROR_length_wrong := 116,
eMBUSERROR_wrong_terminal := 117,
eMBUSERROR_Terminal_is_not_initialized := 118,
eMBUSERROR_stSecAdr_udIdNumber_wrong := 119,
eMBUSERROR_missing_parts_telegram := 120,
eMBUSERROR_no_stop_character_received := 121,
eMBUSERROR_too_many_characters := 122,
eMBUSERROR_TimeOut_FB_KL6781 := 123,
eMBUSERROR_TimeOut_MeterFB := 124,
eMBUSERROR_COM_PARAMETERCHANGED := 201,
eMBUSERROR_COM_TXBUFFOVERRUN := 202,
eMBUSERROR_COM_STRINGOVERRUN := 210,
eMBUSERROR_COM_ZEROCHARINVALID := 211,
eMBUSERROR_COM_INVALIDPOINTER := 220,
eMBUSERROR_COM_INVALIDRXPOINTER := 221,
eMBUSERROR_COM_INVALIDRXLENGTH := 222,
eMBUSERROR_COM_DATASENDERROR := 223,
eMBUSERROR_COM_INVALIDBAUDRATE := 16#1001,
eMBUSERROR_COM_INVALIDNUMDATABITS := 16#1002,
eMBUSERROR_COM_INVALIDNUMSTOPBITS := 16#1003,
eMBUSERROR_COM_INVALIDPARITY := 16#1004,
eMBUSERROR_COM_INVALIDHANDSHAKE := 16#1005,
eMBUSERROR_COM_INVALIDNUMREGISTERS := 16#1006,
eMBUSERROR_COM_INVALIDREGISTER := 16#1007,
eMBUSERROR_COM_TIMEOUT := 16#1008

end_type

eMBUS_no_error: No error at the block. The block is currently not querying a meter.

eMBUS_busy: The block is querying a meter.

eMBUS_Disabled: The block is deselected.

eMBUS_FBLK6781_Disabled: The block FB_MBUS_KL6781() [18] is deselected.

eMBUSERROR_CIField_wrong_72hex_expected: The 7th byte in the response telegram contains the CI field. In this byte the hexadecimal number 72 is expected. It stands for variable data structure, low byte is sent first. Only this data structure is supported.

eMBUSERROR_no_data_received: No data was received.

eMBUSERROR_error_checksum: The response telegram includes a checksum (sum of all bytes from byte 5). The received checksum does not match the calculated checksum. This happens if the protocol was not received cleanly (e.g. in the event of interference on the cable or if the cable is too long).

eMBUSERROR_error_in_head_data: The first 4 bytes are not included in the checksum. These 4 bytes are monitored separately.

eMBUSERROR_usiAddress_over_250: Addresses higher than 250 are not permitted. The input usiAddress of the meter block was assigned a value higher than 250 (exception: Address 254. This address can be used if only one meter is connected).

eMBUSERROR_send_error: Error message for error during sending.

eMBUSERROR_received_address_wrong: Received address does not match the sent address.

eMBUSERROR_cMBUS_MaxCom_below_1: Reserve.
eMBUSERROREQ_iComId_over_cMbus_MaxCom: Reserve.

eMBUSERROREQ_manufacturer_sign_wrong: The response telegram includes a manufacturer code. This code is allocated to the counter blocks. This message appears if the received manufacturer code does not match the block used.

eMBUSERROREQ_baudrate_wrong: Input eBaudrate of the block was assigned invalid values. Only E_MBUS_Baudrate [1-197] are allowed.

eMBUSERROREQ_ReceiveBufferFull: The receive buffer of the serial interface is full. This may happen with long telegrams and/or long cycle times. The PLC is unable to read the data quick enough from the receive buffer, resulting in data loss. The situation may be resolved by reducing the cycle time.

eMBUSERROREQ_E5hex_no_received: No single character E5 hexadecimal was received after initialization of the meter.

eMBUSERROREQ_no_stop_character: No end character in the data array.

eMBUSERROREQ_length_wrong: Number of received characters <> the length field.

eMBUSERROREQ_wrong_terminal: Incorrect terminal connected

eMBUSERROREQ_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?

eMBUSERROREQ_stSecAdr_udildIdNumber_wrong: The input variable stSecAdr_udildIdNumber is not assigned

eMBUSERROREQ_missing_parts_telegram: Not all telegram values were received.

eMBUSERROREQ_no_stop_character_received: No stop characters were received (16hex).

eMBUSERROREQ_too_many_characters: Too many characters were received.

eMBUSERROREQ_Timeout_FB_KL6781: Timeout FB_KL6781.

eMBUSERROREQ_Timeout_MeterFB: Meter block timeout.

eMBUSERROREQ_COM_PARAMETERCHANGED: Input parameters have changed during reception.

eMBUSERROREQ_COM_TXBUFSIZEOVERRUN: String > transfer buffer.

eMBUSERROREQ_COM_STRINGOVERRUN: End of the string.

eMBUSERROREQ_COM_ZEROCARRININVALID: String may not contain any zero characters.

eMBUSERROREQ_COM_INVALIDPOINTER: Invalid data pointer, e.g. zero.

eMBUSERROREQ_COM_INVALIDRXPOINTER:_invalid data pointer for ReceiveData.

eMBUSERROREQ_COM_INVALIDRXLENGTH: Invalid length for ReceiveData e.g. zero.

eMBUSERROREQ_COM_DATASIZEOVERRUN: End of the data block.

eMBUSERROREQ_COM_INVALIDBAUDRATE: Invalid baud rate.

eMBUSERROREQ_COM_INVALIDNUMDATABITS: Invalid data bits.

eMBUSERROREQ_COM_INVALIDNUMSTOBITS: Invalid stop bits.

eMBUSERROREQ_COM_INVALIDPARITY: Invalid parity.

eMBUSERROREQ_COM_INVALIDHANDSHAKE: Invalid handshake.

eMBUSERROREQ_COM_INVALIDNUMREGISTERS: Invalid num register.

eMBUSERROREQ_COM_INVALIDREGISTER: Invalid register.

eMBUSERROREQ_COM_TIMEOUT: COM timeout.
Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.2.1.3 E_MBus_Fct

Value function.

```plaintext
TYPE E_MBus_Fct :
{
  eMBUS_ValueNull          := -1,
  eMBUS_InstantaneousValue := 0,
  eMBUS_Max                := 1,
  eMBUS_Min                := 2,
  eMBUS_ValueDuringErrorState := 3,
  eMBUS_ManufacturerSpecific := 256
}
END_TYPE
```

- **eMBUS_ValueNull**: Not assigned.
- **eMBUS_InstantaneousValue**: Instantaneous value.
- **eMBUS_Max**: Maximum value.
- **eMBUS_Min**: Minimum value.
- **eMBUS_ValueDuringErrorState**: Faulty value.
- **eMBUS_ManufacturerSpecific**: Manufacturer-specific.

### 4.2.1.4 E_MBUS_Medium

Medium

```plaintext
TYPE E_MBUS_Medium :
{
  eMBUS_MediumNull         := -1,
  eMBUS_MediumOther        := 0,
  eMBUS_MediumOil          := 1,
  eMBUS_MediumElectricity  := 2,
  eMBUS_MediumGas          := 3,
  eMBUS_MediumHeat_Outlet  := 4,
  eMBUS_MediumSteam        := 5,
  eMBUS_MediumHot_Water    := 6,
  eMBUS_MediumNot_Water    := 7,
  eMBUS_MediumHeat_Cost_Allocator := 8,
  eMBUS_MediumCompressed_Air := 9,
  eMBUS_MediumCooling_load_meter_outlet := 10,
  eMBUS_MediumCooling_load_meter_intlet := 11,
  eMBUS_MediumHeat_Inlet   := 12,
  eMBUS_MediumHeat_cooling_load_Meter := 13,
  eMBUS_MediumBusSystem    := 14,
  eMBUS_MediumUnknownMedium := 15,
  eMBUS_MediumReserved16   := 16,
  eMBUS_MediumReserved17   := 17,
  eMBUS_MediumReserved18   := 18,
  eMBUS_MediumReserved19   := 19,
  eMBUS_MediumReserved20   := 20,
  eMBUS_MediumReserved21   := 21,
  eMBUS_MediumColdWater    := 22,
  eMBUS_MediumDualWater    := 23,
  eMBUS_MediumPressure     := 24,
  eMBUS_MediumA_D_Converter := 25,
}```
eMBUS_MediumReserved26 := 26,
eMBUS_MediumReserved27 := 27,
eMBUS_MediumReserved28 := 28,
eMBUS_MediumReserved29 := 29,
eMBUS_MediumReserved30 := 30
)

END_TYPE

eMBUS_MediumNull: Not assigned.
eMBUS_MediumOther: Other.
eMBUS_MediumOil: Oil.
eMBUS_MediumElectricity: Electricity.
eMBUS_MediumGas: Gas.
eMBUS_MediumHeat_Outlet: Heat (return).
eMBUS_MediumSteam: Steam.
eMBUS_MediumHot_Water: Hot water.
eMBUS_MediumWater: Water.
eMBUS_MediumHeat_Cost_Allocator: Heating cost distributor.
eMBUS_MediumCompressed_Air: Compressed air.
eMBUS_MediumCooling_load_meter_outlet: Cooling (return).
eMBUS_MediumCooling_load_meter_inlet: Cooling (supply/flow).
eMBUS_MediumHeat_inlet: Heat (supply/flow).
eMBUS_MediumHeat_cooling_load_Meter: Heating / cooling.
eMBUS_MediumBusSystem: Bus / system.
eMBUS_MediumUnknownMedium: Unknown.
eMBUS_MediumReserved16: Reserved.
eMBUS_MediumReserved17: Reserved.
eMBUS_MediumReserved18: Reserved.
eMBUS_MediumReserved19: Reserved.
eMBUS_MediumReserved20: Reserved.
eMBUS_MediumReserved21: Reserved.
eMBUS_MediumColdWater: Cold water.
eMBUS_MediumDualWater: Mixed water.
eMBUS_MediumPressure: Pressure.
eMBUS_MediumA_D_Converter: A/D converter.
eMBUS_MediumReserved26: Reserved.
eMBUS_MediumReserved27: Reserved.
eMBUS_MediumReserved28: Reserved.
eMBUS_Medium Reserved29: Reserved.
eMBUS_MediumReserved30: 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_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2 Structures

4.2.2.1 ST_KL6781outData22B

Process image of the outputs.
Linked to the terminals in the System Manager.

```
TYPE ST_KL6781outData22B :
  STRUCT
    Ctrl : WORD;
    D   : ARRAY[0..21] OF BYTE;
  END_STRUCT
END_TYPE
```

Ctrl: Control word.
D: 22 bytes for the output data of the M-Bus.

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.2 ST_KL6781inData22B

Process image of the inputs.
Linked to the terminals in the System Manager.

```
TYPE ST_KL6781inData22B :
  STRUCT
    Status : WORD;
    D      : ARRAY[0..21] OF BYTE;
  END_STRUCT
END_TYPE
```

Status: Status word.
D: 22 bytes for the input data of the M-Bus.

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.3 ST_MBUS_Communication

Interne Struktur.

This structure is used to link the block FB_MBUSKL6781() with the meter blocks.

```
TYPE ST_MBUS_Communication :
  STRUCT
    bStart     : BOOL;
    bBusy      : BOOL;
    bSND_NKE   : BOOL;
  END_STRUCT
END_TYPE
```

Requirements
bSend: Data is being sent.
bStartManuell: Manual start.
bBlockadeSecAdr: Secondary addressing results in blocking.
usiAddress: Primary address.
byCField: C field.

stSecAdr: Secondary address (see \texttt{ST\_MBUS\_SecAdr} [\pageref{ST\_MBUS\_SecAdr}]).

eError: Error number (see \texttt{E\_MBUS\_ERROR} [\pageref{E\_MBUS\_ERROR}]).
eBaudrate: Baud rate (see \texttt{E\_MBUS\_Baudrate} [\pageref{E\_MBUS\_Baudrate}]).

arrMBusLongFrame: Sent or received bytes.

bySendByte: Number of sent bytes.

uiMaxCount: Maximum number of read commands.
uiCount: Current read command.
stKomRxBuffer: Receive buffer.
stKomTxBuffer: Send buffer.

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.4 \texttt{ST\_MBus\_Data}

Value information.

\begin{verbatim}
TYPE ST_MBus_Data :
  STRUCT
    sValue : STRING(25);
    sUnit : STRING(20);
    sInfo : STRING;
    eFct : E_MBus_Fct;
    iTariff : INT;
    iStorNo : INT;
    iUnit : INT;
END_STRUCT
END_TYPE
\end{verbatim}
sValue: Value.

sUnit: Unit.

sInfo: Information.

eFct: Function (see E_MBus_Fct [200]).

iTariff: Tariff.

iStorNo: Memory number.

iUnit: Unit (integer).

byVIFE: VIFE.

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.5 ST_MBus_Data2

Structure of the output values in the block FB_MBUS_General_Ext() [79].

```plaintext
TYPE ST_MBus_Data2 : STRUCT
  arrData : ARRAY[1..CMBUS_MaxData] OF ST_MBus_Data;
END_STRUCT
END_TYPE
```

arrData: 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_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.6 ST_MBus_Info

Value information.

```plaintext
TYPE ST_MBus_Info : STRUCT
  sValue : STRING(25);
  sUnit : STRING(20);
  eFct : E_MBus_Fct;
END_STRUCT
END_TYPE
```

sValue: Value as string.

sUnit: Unit of the value as string.

eFct: Function (see E_MBus_Fct [200]).

M-Bus devices may supply very large values, which cannot be displayed or can only be displayed inaccurately as numbers on BC/BX systems. The values are therefore supplied as strings (sValue).
Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

### 4.2.2.7 ST_MBUS_SecAdr

Secondary address of a meter.

```plaintext
TYPE ST_MBUS_SecAdr :
   STRUCT
      udiIdNumber : UDINT := 16#FFFFFFFF;
      uiManufacturer : UINT := 16#FFFF;
      usiVersion : USINT := 16#FF;
      usiMedium : USINT := 16#FF;
   END_STRUCT
   END_TYPE
```

**udiIdNumber**: Serial number of the meter.

**uiManufacturer**: Manufacturer code.

**usiVersion**: Counter software version.

**usiMedium**: Medium.

### 4.2.2.8 ST_MBus_Scan

Scanning information.

```plaintext
TYPE ST_MBus_Scan :
   STRUCT
      usiAddress : USINT;
      dwIdNumber : DWORD;
      byStatus : BYTE;
      eMedium : E_MBUS_Medium;
      sMan : STRING(3);
      byGEN : BYTE;
   END_STRUCT
   END_TYPE
```

**usiAddress**: Primary address of the meter.

**dwIdNumber**: Serial number of the meter (secondary address)

**byStatus**: Status.

**eMedium**: Medium (see E_MBUS_Medium).

**sMan**: Manufacturer code.

**byGEN**: Software version of the device.
4.2.2.9  

**Hydrometer**

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_MBUS_DueDayHYD1</td>
<td>Structure of the cutoff date values in the block FB_MBUS_HYD_Sharpy_00.</td>
</tr>
</tbody>
</table>

4.2.2.9.1  ST_MBUS_DueDayHYD1

Structure of the cutoff date values in the block FB_MBUS_HYD_Sharpy_00.

```plaintext
TYPE ST_MBUS_DueDayHYD1 :
  STRUCT
    stEnergy : ST_MBus_Info;
    stVolume : ST_MBus_Info;
    stTariff1 : ST_MBus_Info;
    stTariff2 : ST_MBus_Info;
    stDate : ST_MBus_Info;
    stDateFutureDueDay : ST_MBus_Info;
  END_STRUCT
END_TYPE
```

- **stEnergy**: Energy meter reading (see ST_MBus_Info).
- **stVolume**: Volume meter reading (see ST_MBus_Info).
- **stTariff1**: Meter reading tariff 1 (see ST_MBus_Info).
- **stTariff2**: Meter reading tariff 2 (see ST_MBus_Info).
- **stDate**: Cutoff date (see ST_MBus_Info).
- **stDateFutureDueDay**: Future cutoff date (see ST_MBus_Info).

**Requirements**

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.2.2.10  Metrima

<table>
<thead>
<tr>
<th>Data types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_MBus_F22</td>
<td>Structure of the monthly values in the block FB_MBUS_SVM_F22_Ext.</td>
</tr>
</tbody>
</table>

4.2.2.10.1  ST_MBus_F22

Structure of the monthly values in the block FB_MBUS_SVM_F22_Ext.

```plaintext
TYPE ST_MBus_F22 :
  STRUCT
    stEnergy : ST_MBus_Info;
    stVolume : ST_MBus_Info;
    stVolume2 : ST_MBus_Info;
    stPulsecounter1 : ST_MBus_Info;
    stPulsecounter2 : ST_MBus_Info;
    stDate : ST_MBus_Info;
  END_STRUCT
END_TYPE
```

- **stEnergy**: Energy meter reading (see ST_MBus_Info).
- **stVolume**: Volume meter reading (see ST_MBus_Info).
stVolume2: Volume meter reading (see ST_MBus_Info[204]).

stPulsecounter1: Meter reading pulse counter 1 (see ST_MBus_Info[204]).

stPulsecounter2: Meter reading pulse counter 2 (see ST_MBus_Info[204]).

stDate: Date (see ST_MBus_Info[204]).

Requirements

<table>
<thead>
<tr>
<th>Development environment</th>
<th>required TC3 PLC library</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT from v3.1.4020.14</td>
<td>Tc2_MBus from 3.3.4.0</td>
</tr>
</tbody>
</table>

4.3 GLVs

4.3.1 Globale Variablen_MBUS

If they are declared in the program, a warning message is generated during program compilation, since the constant already exists. This warning can be ignored.

```plaintext
VAR_GLOBAL CONSTANT
   cMBUS_MaxData := 40,
   cMBUS_MaxTelegrams := 2,
   cMBUS_MaxDataParam := 10,
END_VAR
```

- **cMBUS_MaxData**: The constant applies to all instances of the blocks FB_MBUS_General() [76], FB_MBUS_General_Ext() [79] and FB_MBUS_General_Param() [83]. It indicates the maximum data volume expected in a meter telegram.

- **cMBUS_MaxTelegrams**: The constant applies to all instances of the FB_MBUS_General_Ext block() [79]. It indicates the maximum number of telegrams to be expected.

- **cMBUS_MaxDataParam**: The constant applies to all instances of the FB_MBUS_General_Param blocks() [83]. It indicates the maximum number of values to be displayed by the instances of block FB_MBUS_General_Param() [83].

4.4 Integration into TwinCAT

4.4.1 KL6781 with CX5120

This example describes how a simple PLC program for M-Bus can be written in TwinCAT and how it is linked with the hardware. The task is to read a counter with four digital inputs.

Example: https://infosys.beckhoff.com/content/1033/tcplclib_tc2_mbus/Resources/zip/6218378891.zip

Hardware

**Setting up the components**

The following hardware is required:

- 1x CX5120 Embedded PC
- 1x KL6781 M-Bus master terminal
- 1x KL9010 end terminal

Set up the hardware and the M-Bus components as described in the associated documentation.
This example assumes that the counter address is known.

**Software**

**Creation of the PLC program**

Create a new "TwinCAT XAE project" and a "Standard PLC project".

Add the library Tc2_MBus under References in the PLC project.

Generate a Global Variable List with the name GVL_MBus and create the following variables:

```plaintext
VAR_GLOBAL
  stComIn  AT %I*: ST_KL6781inData22B;
  stComOut AT %Q*: ST_KL6781outData22B;
  stCom  : ST_MBUS_Communication;
END_VAR
```

- **stComIn**: Input variable for the M-Bus terminal (see ST_KL6781inData22B [202]).
- **stComOut**: Output variable for the M-Bus terminal (see ST_KL6781outData22B [202]).
- **stCom**: Required for the communication with M-Bus (see ST_MBUS_Communication [202]).

Create a program (CFC) for the background communication with M-Bus. The FB_MBUSKL6781 [18] block is called in this program. Make sure to link the communication block with stComIn, stComOut and stCom.

Create a MAIN program (CFC) in which the block FB_MBUS_REL_PADIn4 [156] is called up. Link the input usiAddress of the counter block with the local variable usiAddress and stCom with the global variable stCom.

Navigate to the task configuration section and configure the PlcTask. By way of example, the task is assigned priority 16 and a cycle time of 6 ms.
Create a further task for the background communication. Assign a higher priority (smaller number) and a lower interval time to this task than the PLCTask.

Add the program for the communication to this task. Further information on task configuration can be found in the description of the function block FB_MBUSKL6781 [18].

**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 corresponding tasks.
Now link the global variables of PLC program with the inputs and outputs of the Bus Terminals. Create the Solution and enable the configuration.

When linking the data array, make sure that you select both the All types and Continuous options. Use the Shift key and the right mouse button to mark all data bytes of the terminal.

After starting the PLC, the current values are regularly read by the counter.
Appendix

5.1 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:
http://www.beckhoff.com

You will also find further documentation for Beckhoff components there.

Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG
Huelshorstweg 20
33415 Verl
Germany
Phone: +49(0)5246/963-0
Fax: +49(0)5246/963-198
e-mail: info@beckhoff.com

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(0)5246/963-157
Fax: +49(0)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(0)5246/963-460
Fax: +49(0)5246/963-479
e-mail: service@beckhoff.com
More Information:
www.beckhoff.com/te1000/