



(Bilder: Beckhoff)

Dr.-Ing. Josef Papenfort

Bausteine für Motion Control

Die IEC 61131-3 hat die SPS-Programmierung in den letzten Jahren wesentlich beeinflusst und vereinfacht. Der von PLCopen erarbeitete Motion-Control-Standard soll Gleiches auf dem Gebiet der Bewegungssteuerung bewirken. Erste Projekte mit den neuen Bausteinen sind mittlerweile realisiert.



Dr.-Ing. Josef Papenfort

ist als Softwareingenieur in der Grundlagenentwicklung bei Beckhoff, Verl, tätig.

Bevor mit der IEC 61131-3 ein Standard für die Programmierung von Speicherprogrammierbaren Steuerungen (SPS) geschaffen wurde, kochte jeder Anbieter sein eigenes Süppchen, und es existierte eine große Anzahl von verschiede-

nen Programmiersprachen. Mit der weltweit etablierten IEC 61131-3 ist dieses Problem mittlerweile behoben: Jeder SPS-Programmierer weiß, wie ein Timer-Funktionsbaustein TON funktioniert – und er funktioniert bei allen Herstellern, die den Standard unterstützen, gleich! Die wirkliche Portabilität ist dabei

weniger gefragt; der entscheidende Punkt ist die gemeinsame Sprache. Jeder, der IEC 61131-3 beherrscht, findet sich in Programmen der unterschiedlichen Hersteller schnell zurecht.

Ganz anders sah es bisher bei Motion-Control-Systemen aus. Die Bewegungssteuerung war immer eine

sehr proprietäre Angelegenheit der einzelnen Motion-Control-Anbieter. Nicht nur die Hardware, auch die Software war sehr unterschiedlich. Bei Endanwendern, die mehrere Motion-Control-Systeme unterstützen mussten, verursachte dies erhebliche Kosten für Entwicklung und Wartung.

Um diesen Nachteil auszugleichen, hat die PLCopen, die Nutzerorganisation der IEC-61131-3-Anwender, einen Arbeitskreis ins Leben gerufen, der in den letzten Jahren einen Standard für die Programmierung von Motion-Control-Systemen definiert hat. Vertreten sind in der entsprechenden Task Force nicht nur Hersteller von Antriebstechnik, sondern auch Hersteller von Programmiersystemen und Endanwender. Ihr gemeinsames Ziel: Schaffung einer Bibliothek von SPS-Bausteinen zur Ansteuerung von Achsen aus einem IEC-61131-3-Programm, so dass SPS- und NC-Funktionalität in einer Programmierumgebung liegen. Dabei sollte dieses auf den Applikationsprogrammierer zugeschnittene Baustein-Set gleichermaßen leicht zu handhaben wie zu warten und zudem hardware- und herstellerunabhängig sein.

Vom Servo- bis zum Schrittmotor

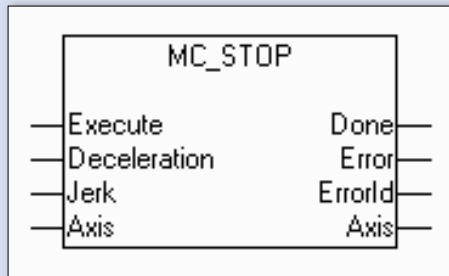
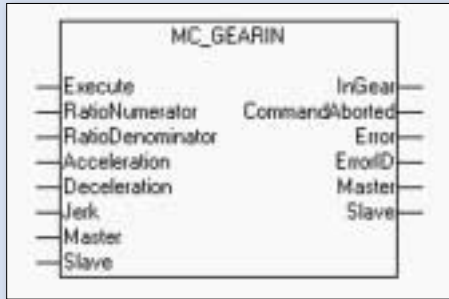
Die Vorgaben, die sich hieraus ableiten ließen: Von der Software-Seite sind alle Bausteine identisch, egal was für eine Hardware sich dahinter befindet. Die Hardware kann ein leistungsfähiger Servo-Antrieb sein oder ein über eine einfache digitale Ausgabe betriebener Schrittmotor. Die Software kann auf einem PC oder direkt auf dem Antrieb laufen, sprich auf einem zentralen oder einem dezentralen System. Die standardisierte Software-Schicht der Motion-Control-Bausteine verdeckt hier die Hersteller- und Hardware-Abhängigkeit.

Nach der Definition eines Zustandsdiagramms für die Achsen konnten für alle möglichen Übergänge recht einfach Bausteine definiert werden.

Konkret wurden sieben verschiedene Zustände mit einer entsprechenden Anzahl von Übergängen ausfindig gemacht. Für jede Transition von einem Zustand in einen anderen sind ein oder mehrere Bausteine zuständig. Ein Beispiel: Um von „Standstill“ in „Discrete Motion“ zu gelangen, muss der Baustein *MC_MoveAbsolute* aufgerufen werden. Eine Achse wird gestoppt, das heißt in den Zustand Standstill versetzt, indem der Baustein *MC_Stop* aufgerufen wird. Hier durchläuft die Achse während der Bremsphase erst den Zustand „Stopping“ und gelangt dann in den Zustand „Standstill“.

Da die Anforderungen und Ressourcen der unterschiedlichen Antriebe – vom PC bis hin zum 8-bit-Mikrocontroller – sehr unterschiedlich sind, sind nicht alle Bausteine zur Einhaltung des Standards erforderlich. Zwingend notwendig sind die Basisbausteine, die „Extended“-Bausteine hingegen sind den High-end-Motion-Control-Systemen vorbehalten. Gleiches gilt für die Ein- und Ausgänge; auch hier gibt es notwendige und optionale. Damit ist gewährleistet, dass sich dieser Standard bei vielen, auch sehr unterschiedlichen Herstellern durchsetzen kann. Das war auch der Punkt, warum die Versorgung der Bausteine mit Daten über eine nicht näher spezifizierte, per *VAR_IN_OUT* übergebene Struktur durchgeführt wird. Das bedeutet: Bei ansonsten gleichem Interface kann sich hinter der Struktur nur eine einzige Nummer, etwa die Achs-ID, oder eine komplexe Struktur mit vielen Einzelkomponenten verbergen. *VAR_IN_OUT* erlaubt darüber hinaus die Modifikation der Struktur innerhalb der Bausteine.

Das zeitliche Verhalten der Bausteine ist genau spezifiziert. Bei steigender Flanke vom Execute startet die Ausführung des Bausteins, das Zurücksetzen der Ausgänge erfolgt bei fallender Flanke. Wird das Execute gelöscht, während sich die Achse noch bewegt, so



Standardisierte Motion-Control-Bausteine in der Anwendung bei Pressen-Steuerung in Verbindung mit der Software SPS/NC Twincat.



werden die Ausgänge für mindestens einen Zyklus noch gesetzt. Ein weiterer wichtiger Punkt ist die exakt definierte Verhaltensweise der Bausteine beim Eintreten von Fehlern. Es gibt lokale Baustein- und Achsfehlermeldungen direkt am Baustein und zusätzlich noch die Möglichkeit, die Achsfehler über einen speziellen Baustein zu lesen.

Erste Spezifikation im Oktober 2001

Zeitgleich mit dem Erscheinen des ersten Dokuments der PLCopen im Oktober 2001 hat Beckhoff die Bausteine in Form einer Motion-Control-Bibliothek für die Automatisierungssoftware Twincat implementiert. Die ersten Einsätze dieser Bausteine sind erfolgreich abgeschlos-

sen und werden im Folgenden anhand zweier Anwendungen mit sehr unterschiedliche Aufgabenstellungen exemplarisch beschrieben. Der erste Einsatz der Motion-Control-Bausteine erfolgte bei Müller Weingarten, einem international tätigen Anbieter für Anlagen und Systeme der Umformtechnik für metallische Werkstoffe. Das Kerngeschäft ist die Planung, Konzeption und Ausrüstung von Presswerken für die Herstellung von Karosserieteilen in der Automobil- und deren Zulieferindustrie. Konkret handelte es sich bei der realisierten Applikation um die Automatisierung eines Zu- und Abfuhrsystems für eine Pressenstrasse. Der Ablauf im Detail: Rohteile werden mit dem Zuführfeeder unter die Presse transportiert und dort umgeformt. Ein entsprechender Abfuhrfeeder transportiert die gepressten Teile unter der Presse heraus und weiter in eventuell nachfolgende Pressen. Für die Bewegung der Feeder sind zwei links und rechts der Presse befindliche Antriebe zuständig. Zwischen den Antrieben befindet sich ein leichter Carbon-Balken, an dem die Aufnahmevorrichtung für die Teile befestigt ist. Die Synchronität der beiden Antrie-



Die hydraulische Tryout-Multicurve-Pressen von Müller Weingarten im Einsatz bei Volkswagen. Im Wolfsburger Automobilkonzerns simulieren insgesamt sechs Pressen die reale Produktionsumgebung, bevor die Werkzeuge in die Serienfertigung gehen.

be stellt ein elektronisches Getriebe sicher. Zur Optimierung der Abläufe gilt es, die Zuführung und Abführung der Teile mit der Presse zu koordinieren – bei einer Aufwärtsbewegung des Stößels werden bereits die Feeder in Gang gesetzt. Die Steuerung für jeweils eine Presse inklusive Hydraulik-Reglern und den zugehörigen Feedern übernimmt ein Industrie-PC, auf dem die Steuerungssoftware Twincat läuft.

„Sonderbausteine“ für spezielle Funktionen

Die Steuerung aller Motion Control spezifischer Aktionen erfolgt bei dieser Applikation über die Motion-Control-Bausteine nach der PLCopen-Spezifikation: Bereits die Freigaben für die einzelnen Achsen werden richtungsabhängig mit dem Baustein *MC_Power* durchgeführt. Nach erfolgter Freigabe werden die Achsen mit *MC_MoveAbsolute* verfahren, während der Fahrt entsprechend der Stößelposition auf neue Zielpositionen verlängert und mit *MC_Stop* im Fehlerfall gestoppt. Die Realisierung des elektronischen Getriebes erfolgte mit den Bausteinen *MC_GearIn* und *MC_GearOut* entsprechend dem Standard. Einige nicht im Standard beschriebene, aber dem Standard nachempfundene Bausteine ergänzen die Implementation, so zum Beispiel ein von Beckhoff speziell geschriebener Baustein, mit dem sich Override-Änderungen durchführen lassen. Die Applikation hätte auch mit den schon vor der Spezifizierung durch die PLCopen bei Beckhoff verfügbaren proprietären Bausteinen geschrieben werden können. Durch den Einsatz der Motion-Control-Bausteine gibt es aber erstmals einen standardisierten einfachen Weg der Programmierung. Dadurch konnte schon bei der Programmierung der Anlage sehr viel Zeit eingespart werden.

Die Einführung der Motion-Control-Bausteine bei Müller Weingarten ist die konsequente Weiterführung der Standardisierungsbestrebungen des Pressenherstellers. Nach dem Einsatz der IEC 61131-3 vor etwa drei Jahren war der Einsatz der Motion-Control-Bausteine entsprechend dem PLCopen-Standard der nächste logische Schritt.

Kurvenscheiben nach IEC 61131-3

Eine weitere Anwendung wurde bei Molex in Singapur realisiert. Molex ist der weltweit zweitgrößte Hersteller von Steckverbindern für die Elektronikindustrie. Die technisch zum Teil sehr hohen Anforderungen können dort nur mit einem eigenen Maschinenbau erfüllt werden. An einigen Prototypmaschinen wurden auch hier bereits erste positive Erfahrungen mit der IEC 61131-3 und den dazu passenden Motion-Control-Bausteinen gesammelt.

Konkret handelt es sich dabei um eine Maschine zum Eintreiben von Stiften in eine Steckerleiste mit variabler Stiftezahl. Die Stifte müssen mit einem Abstand

Function Block Categories			
Administrative		Motion	
Single Axes	Multiple Axes	Single Axes	Multiple Axes
<ul style="list-style-type: none"> - MC_Power - MC_Reset - MC_ReadStatus - MC_ReadAxisError - MC_ReadParameter - MC_WriteParameter - MC_ReadActualPosition 	<ul style="list-style-type: none"> - MC_CamTableSelect 	<ul style="list-style-type: none"> - MC_MoveAbsolute - MC_MoveRelative - MC_MoveAdditive - MC_MoveSuperImposed - MC_MoveVelocity - MC_Home - MC_Stop - MC_PositionProfile - MC_VelocityProfile - MC_AccelerationProfile 	<p>Non-Interpolated</p> <ul style="list-style-type: none"> - MC_CamIn - MC_CamOut - MC_GearIn - MC_GearOut - MC_Phasing <p>Interpolated</p> <ul style="list-style-type: none"> - ...

Übersicht der Motion-Control-Bausteine nach dem PLCopen-Standard.



von 0,5 mm und einer Geschwindigkeit von 1000 Pins pro Minute einzeln in die entsprechenden Öffnungen eingetrieben werden. Eine rotierende Achse fördert von einem Band kontinuierlich Stifte für die Stecker. Für den Vorschub und das Eintreiben ist eine aufwendige Mechanik verantwortlich.

Der aufnehmende Steckerrohling wird dabei unter die Eintreibereinheit geschoben und dann immer um ein kurzes Stück entsprechend dem Stiftabstand weiterbefördert. Ist der Stecker vollständig bestückt, wird er ausgestoßen. Zu-, Weiter- und Abförderung der Stecker bewerkstelligen zwei Zuführachsen, die parallel angebracht sind und mit fingerartigen Geräten den Stecker in einer Rinne verschieben. Jeweils eine Achse transportiert den Stecker vor die Eintreibereinheit, während sich die zweite Achse gleichzeitig wieder zurückbewegt, um den Transport des nächsten Steckers vorzubereiten.

Die Koordination von zuführenden Achsen und Eintreibachse ist mit zwei Kurvenscheiben realisiert. Die Tabellen werden in der SPS aufbereitet; sie bestehen aus vorgefertigten Teilen für die Beschleunigungsphase und einer Anzahl kurzer Wegsstücke zum Vorschieben des Steckerrohlings unter die Eintreibereinheit. Die Länge der Wegsstücke entspricht dabei dem Stiftabstand. Die Anzahl der Wegsstücke entspricht der Anzahl der Stifte im Stecker. Komplettiert wird die Tabelle mit der Rückfahrphase.

Diese Applikation verwendet neben dem administrativen Motion-Control-Baustein *MC_Power*, der Endlosfahrt *MC_MoveVelocity* für die Bewegung der Eintreibachse und mehrerer *MC_MoveAbsolutes* zur ersten Positionierung der Zuführachsen auch die Tabellenkopplungsbausteine *MC_CAMSelect*, *MC_CAMIn* und *MC_CAMOut*. Mit *MC_CAMSelect* wird eine Tabelle aus gesucht und parametrieret. *MC_CAMIn* dient zum Koppeln von Master und Slave, *MC_CAMOut* zum Abkoppeln. Auch hier konnte die relativ anspruchsvolle Aufgabe mit Hilfe der IEC 61131 und den einfach zu bedienenden standardisierten Motion-Control-Bausteinen innerhalb von zwei Wochen erledigt werden.

zu definieren. Auch für spezielle Aufgaben wie die fliegende Säge oder für spezielle Industriezweige wie die Verpackungsindustrie wären standardisierte Bausteine wünschenswert. Da es eine sehr gute Resonanz auf die erste Version des Standards gegeben hat, wird hier sicherlich bald mit weiteren Bausteinen zu rechnen sein.

Wie auch im Zusammenhang mit der SPS-Programmierung nach IEC 61131-3 ist bei den Motion-Control-Bausteinen nicht unbedingt die Portabilität der Programme das Ziel, sondern ein Verständnis für die Funktionsweise und Beschaltung der Bausteine. Ähnlich wie der Funktionsbaustein bei allen SPS-Systemen gleich funktioniert, ist auch die Funktion der Motion-Control-Bausteine eindeutig definiert. Selbst wenn kleine Unterschiede zulässig sind, ändert das nichts am grundlegenden Verständnis der Bausteine. Für die Hersteller von SPS- und Motion-Control-Systemen sowie für die Anwender bringen die standardisierten Motion-Control-Bausteine eine Reihe von Vorteilen: Den Herstellern stellen sie eine weltweite Akzeptanz ihrer Produkte in Aussicht. Demgegenüber profitiert der Endanwender von weniger Hardware- und Hersteller-Abhängigkeiten und er bekommt einen einfach wiederzuverwendenden Code an die Hand, womit sich insbesondere die Wartungskosten verringern lassen. *gh*

80 Prozent „erschlagen“

Der in der Version 1.0 beschriebene Standard deckt rund 80 % der Motion-Control-Aufgaben ab. Im Rahmen der Fortführung der Task Force Motion Control wäre es sinnvoll, noch Bausteine für interpolierende Bewegungen mit mehreren Achsen