

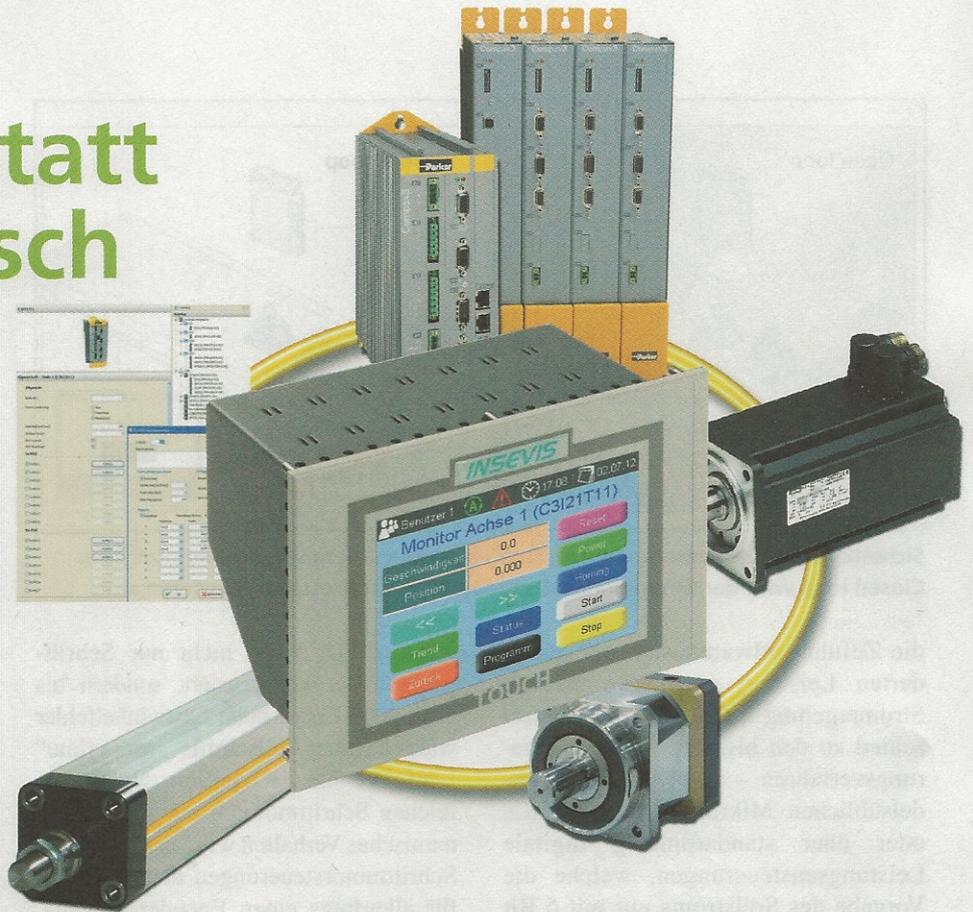
Jörg Peters, Stephan Claußnitzer

# Modular statt monolithisch

Die Einbindung von Antrieben in die Simatic-Steuerungswelt erfolgt in der Regel mit monolithischen Funktionsbausteinen des jeweiligen Herstellers. Ein Anbieterwechsel bringt damit de facto eine Neuprogrammierung der Antriebsfunktionalität mit sich. Modulare Funktionsbausteine lösen dieses Problem.

Die Firmen der Antriebstechnikbranche stellen zur leichteren Einbindung ihrer Komponenten in die Steuerungswelt der Simatic- und dazu kompatiblen SPSen seit geraumer Zeit herstellereigenspezifische S7-Bausteine zur Verfügung. Dabei handelt es sich oft um einen, den Möglichkeiten des Antriebs angepassten beziehungsweise monolithischen Funktionsbaustein. Dieser stellt zwar eine optimierte, jedoch willkürliche Schnittstelle auf. Zudem ist dieser Funktionsbaustein meist auf ein Bussystem zugeschnitten – bei einer S7-CPU normalerweise Profibus-DP/Profinet oder auch – mittels Feldbus-Master-Baugruppen anderer Hersteller – Interbus und CANopen.

Da ein Wechsel des Antriebsherstellers für den S7-Programmierer somit gleichbedeutend mit einer kompletten Neuprogrammierung der Antriebsfunktionalität ist, hat sich die PLCopen zum Ziel gesetzt, den Engineering-Aufwand durch einheitliche Software-Schnittstellen zu reduzieren. Im Antriebsbereich wurden Standards definiert, mittels derer eine Zertifizierung von Antrieben und implementierten Schnittstellen möglich ist. Bei Verwendung von Bussystemen wie CANopen mit Antriebschnittstellen (DS402-Antriebsprofil) ist zudem der Aufwand für die Anpassung an ein konkretes Busprotokoll



gering. Dieser Idee folgend, müssten die großen monolithischen Funktionsbausteine in Einzelfunktionsbausteine aufgeteilt werden, die flexibel und ohne großen Programmieraufwand den Einsatz verschiedener Antriebe ermöglichen.

Obwohl also der Weg der Integration von Antrieben in Anwendungen mit Bewegungsaufgaben vorgezeichnet erscheint, gilt es einige Hürden zu überwinden: Preisgünstige Versionen von Servo- und Schrittmotorantrieben lassen sich meist nicht direkt am Profinet in taktischer Ausführung betreiben, so dass die modularen PLCopen-Bausteine der Siemens-Technologie-SPSen (zum Beispiel S7-300 CPU 315T oder 317T) nicht verwendbar sind. Vielmehr sind oft CANopen- oder Ethercat die Standard-Schnittstelle. Zwar bieten auch viele kostengünstige, zur Simatic-Welt kompatible Steuerungen als Feldbus zum Beispiel CANopen an, meist verfügen diese jedoch nicht über die Ressourcen für eine Bahngenerierung.

Diesem Umstand Rechnung tragend, haben der Steuerungsanbieter Insevis und die auf Motion-Control-Lösungen spezialisierte Firma Inmotec Automation jetzt eine erste Reihe modularer S7-

Funktionsbausteine für CANopen entwickelt, die den S7-Anwendern eine neue Unabhängigkeit und Flexibilität bei der Wahl des Antriebskonzeptes ermöglichen. Mit anderen Worten: Die S7-Seite der Funktionsbausteine ist für die unterstützten Antriebe nahezu identisch. Für den S7-Programmierer vergrößert sich damit die Auswahl bei den verfügbaren intelligenten Antrieben im Low-cost-Bereich, ohne sich mit der für ihn ungewohnten CANopen-Problematik befassen zu müssen.

Alle Antriebsfunktionen auf der Insevis-SPS kommunizieren dabei über asynchrone CANopen-PDOs nach DS301, so dass der Kommunikationsaufwand (Busauslastung) reduziert ist. Beim Antriebsprofil DS402 werden ausschließlich Betriebsarten verwendet, die keine äquidistante Übertragung von Sollwerten (R-PDOs) erfordern. Der so genannte „Interpolated mode“ wird daher nicht verwendet. T-PDOs mit kontinuierlichen Istwerten (Ist-Position, Ist-Geschwindigkeit) sind mit einer Sperrzeit parametrisiert. In diese Lösung integrierbar sind grundsätzlich alle Antriebe mit CANopen-Schnittstelle, die den einfachen „Position mode“ unterstützen, welcher eine antriebsgerechte Bahngenerierung vo-

(Bilder: Insevis)

raussetzt. Pro CANopen-Slave werden mindestens drei R-PDOs, zwei T-PDOs und ein SDO benötigt.

Wie bereits angesprochen, werden bei dem neuen Ansatz alle Antriebsfunktionen als einzelne Funktionsbausteine implementiert. „MC-Power“ ist zum Beispiel ein Funktionsbaustein, der zum Bestromen des Motors dient (weitere Funktionsbausteine: siehe *Kasten*). Kennzeichnend für alle MC-Bausteine ist, dass diese keine M-Merker, T-Zeiten oder Z-Zähler verwenden, sondern deren instanziiierbare IEC-Varianten. Dem Anwendungsprogrammierer stehen somit weiterhin alle Systemressourcen zur Verfügung. Da sich die erstellten Bausteine nicht gegenseitig referenzieren und nur über die Achsreferenz kommunizieren, lassen sich Baustein-Adressen nach Bedarf verschieben.

Erstellt werden die Funktionsbausteine im Original mit SCL (Structured Control Language), einer Engineering-Option zu Step7 der Firma Siemens. Die Verwendung der Funktionsbausteine erfordert jedoch kein installiertes SCL-Paket auf dem Entwicklungsrechner des Anwenders. Um Diversitäten bei Antrieben abzufangen und Namenskonflikte mit bereits vorhandenen Bausteinen aus Bibliotheken zu vermeiden – zum Beispiel bei Technologie-Steuern von Siemens, erhalten die MC-Bausteine in Abhängigkeit vom jeweiligen Antriebstyp einen Postfix wie „\_C3“ für die Compax-C3-Familie von Parker oder „\_E2“ für Servo Drive EPOS2 von Maxon Motor.

**Integration am Beispiel Parker-Antrieb**

Folgende Ausführungen beschreiben die Verwendung der modularen Funktionsbausteine am Beispiel eines intelligenten Servoantriebs von Parker. Der Ansatz ist jedoch problemlos auf Antriebe anderer Anbieter – zum Beispiel den erwähnten Servo Drive EPOS2 von Maxon – übertragbar. Die S7-Software ist an den PLCopen-Standard angelehnt und wurde für die S7-Panel-Steuern von Insevis erstellt. Diese haben bereits in der Grundausstattung die Schnittstellen Ethernet, Modbus-TCP und -RTU sowie CANopen an Bord und verfügen über ein integriertes TFT-Touchpanel mit Diagonalen von 3,5 bis 10,2 Zoll.

Im konkreten Beispiel kommt die Panel-SPS PC570V-DPM zum Einsatz. Diese ist befehlskompatibel zur S7-315-2PNDP und verfügt über ein 5,7-Zoll-Panel, welches dem TP177M von Siemens entspricht. Das nur 50 mm flache Gerät übernimmt S7-Programm, Ethernet-Kommunikation, Profibus-DP Master und Visualisierung. Über die integrierte CANopen-Schnittstelle wird konkret ein Compax C3I21T11 (Positionierantrieb für 1 bis 4 Achsen) angesteuert.

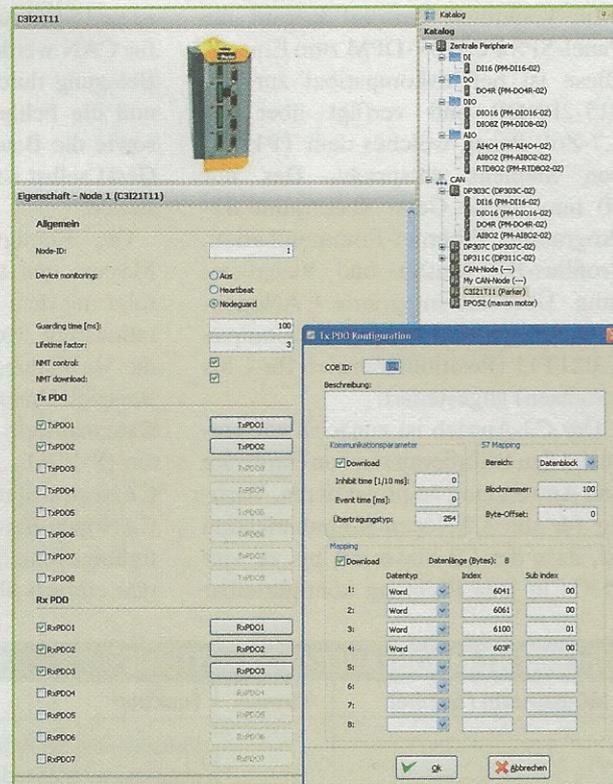
Der C3-Antrieb ist zunächst entsprechend den Erfordernissen von Hardware und Anwendung zu projektieren. Wichtig für die CANopen-Kommunikation ist, dass der C3-Slave mit bis zu vier PDOs in jede Richtung konfigurierbar

ist und SDOs unterstützt. Im C3-Wizard für CAN werden die dynamische PDO-Belegung durch den CANopen-Master und die Fehlerreaktion bei Busausfall sowie die Baudrate festgelegt, am C3-Gerät selbst ist nur die Node-ID einzustellen.

Die Konfiguration des CANopen-Masters und der CANopen-Slaves erfolgt mit dem kostenlosen S7-Konfigurationstool „ConfigStage“. Zudem wird die Verbindung von SPS-Daten – zum Beispiel Datenbaustein und Offset im Datenbaustein – zu den CANopen-Daten (R-PDOs, T-PDOs) projektiert. Ein CANopen-Slave lässt sich als Typ in der CANopen-Slave-Bibliothek der ConfigStage hinterlegen, um typgleiche Geräte einfach über Drag&Drop einzufügen.

ÜBERSICHT MODULARER MC-BAUSTEINE		
MC-Baustein / Symbol	Adresse	Funktion
MC_ReadStatus_C3	FB40	Visualisierung der Antriebszustände (entstromt, stappend, stillstehend, profilbasierte Bewegungsfunktionen aktiv, Endlosbewegung aktiv, synchronisierte Bewegungsfunktionen aktiv, Referenzfahrt aktiv) und zusätzlich Programmcode der Achsreferenz
MC_ReadAxisError_C3	FB41	Visualisierung des Fehlercodes des Antriebs
MC_ReadActualPosition_C3	FB42	Visualisierung der aktuellen Position des Motors
MC_ReadActualVelocity_C3	FB43	Visualisierung der aktuellen Geschwindigkeit des Motors
MC_Reset_C3	FB44	Fehler im Antrieb rücksetzen
MC_Power_C3	FB45	Motor schnellstmöglich bestromen/entstromen
MC_Stop_C3	FB46	Bestromten Motor stoppen
MC_MoveAbsolute_C3	FB47	Absolute Position anfahren
MC_MoveRelative_C3	FB48	Relative Distanz abfahren
MC_MoveAdditive_C3	FB49	Relative Distanz an Bewegung anhängen
MC_MoveVelocity_C3	FB50	Endlosbewegung
MC_GearIn_C3	FB51	Synchronisierte Bewegungsfunktionen ausführen (Elektronisches Getriebe), zum Beispiel einem Leitantrieb via Encoder-Pulsen folgen
MC_Home_C3	FB52	Referenzfahrt ausführen
MC_Jog_C3	FB53	Fahren per Hand +/-, stoppt an den Software-Endgrenzen
C3_Input	FB54	C3-Eingänge auslesen (Standard-Eingänge)
C3_Output	FB55	C3-Ausgänge schreiben (Standard-Ausgänge)
InDataC3Type	UDT100	Datentyp für Eingangsdaten CANopen, pro Achse einmal instanziiieren
OutDataC3Type	UDT101	Datentyp für Ausgangsdaten CANopen, pro Achse einmal instanziiieren
SWPosC3Type	UDT102	Datentyp Statuswort CANopen, interne Verwendung
CWPosC3Type	UDT103	Datentyp Steuerwort CANopen, interne Verwendung
AxisRefC3Type	UDT104	Datentyp Achsreferenz, pro Achse einmal instanziiieren

### Eindeutige Zuweisung von CAN-Einstellungen in die S7-Welt mit der Freeware ConfigStage.



gen. Neben der PDO-Belegung kann im Bibliothekseintrag eine Grafik des CANopen-Slaves abgelegt werden. Für den C3-Antrieb erfolgt die Eintragung einer Nodeguarding-Zeit, über die der CANopen-Master die Integrität des CANopen-Slaves erkennen kann. Zusätzlich wird festgelegt, wie viele PDOs in jeder Richtung nötig sind.

Das Mapping der vorbelegten CAN-PDOs ist in der Konfigurationssoftware einfach auf grafischem Weg möglich. Hier werden die Speicherbereiche festgelegt, die der S7-Programmierer für die Daten verwenden möchte. Wie bei Profibus-DP muss sich der Anwender nicht selbst um das „Auffrischen“ der Daten kümmern. Im Beispiel ist die Belegung der ersten T-PDOs dargestellt (siehe *Bild*). Auf das „Statusword“ (Index 0x6041) wird im DB100 ab Offset 0 zugegriffen, auf „Actual operation mode“ (Index 0x6061) ab Offset 2, auf die digitalen Eingänge (Index 0x6100) ab Offset 6 und auf das aktuelle Fehlerwort (0x603F) ab Offset 8. Sobald die MC-Bausteine zum Einsatz kommen, stellen sie die benötigten Daten selbst an den FB-Ausgängen bereit, so dass der direkte Zugriff auf die T-PDO-Kommunikationsdaten im DB100 nicht notwendig ist.

### Zentrale Rolle der Achsreferenz

Die Achsreferenz spielt eine zentrale Rolle im Steuerungsablauf. Als Achse wird das SPS-Abbild eines Antriebsstrangs (Antrieb + Motor) verstanden, welches alle wesentlichen Informationen enthält, die den Steuerungsablauf betreffen. Die Kernaufgaben der Achsreferenz bestehen in der zentralen Koordinierung aller Aktivitäten bezüglich einer einzelnen Achse und der Verwaltung der singulären Ressource CANopen als Schnittstelle zwischen SPS und Antriebsstrang. Die Achsreferenz ist demnach kein „Zeiger auf Daten“, sondern Datenbereich (Datenfeld im Datenbaustein) und aktiver Programmcode (Bestandteil von MC\_ReadStatus). Der Code muss mindestens einmal im SPS-Zyklus aufgerufen werden, wobei der Aufruf indirekt über den Aufruf der Instanz von MC\_ReadStatus erfolgt, die den zusätzlichen Programmcode der Achsreferenz enthält.

Die Achsreferenz übernimmt dabei folgende Aufgaben:

- Zusammenfassen von Anforderungen (Positionieren, Stoppen, Bestromen) einzelner MC-FB-Instanzen und Verteilen von Statusinformationen an diese Instanzen.

- Testen von Bedingungen vor Befehlsausführung daraufhin, ob die Achse bei einem eintreffenden Befehl auch in der Lage ist, den Befehl voraussichtlich auszuführen.
- Zentrales Schreiben der R-PDOs an den CANopen-Slave und Lesen von T-PDOs des CANopen-Slaves. Nur die Achsreferenz hat Zugang zu den CANopen-Daten.
- Abarbeitung von Zustandsautomaten für Achsaktivitäten wie Positionieren, Joggen, Referenzieren, Stoppen oder Bestromen.

Über die Achsreferenz, bestehend aus einer konkreten typenabhängigen Datenstruktur (zum Beispiel vom Typ AxisRefC3Type) und typenabhängigem Programmcode als Bestandteil des MC-Funktionsbausteins MC\_ReadStatus (MC\_ReadStatus\_C3), können Inkompatibilitäten der CANopen-Implementierungen der Antriebshersteller für die Anwendungsprogrammierung abgefangen werden, da der SPS-Programmierer bei der Ansteuerung der Antriebe nur mit den einheitlichen Schnittstellen der MC-Funktionsbausteine hantiert. Bei Tausch eines Antriebs werden die Instanztypen gewechselt – Instanznamen, Schnittstellen und Verhalten der Antriebe bleiben erhalten.

Zusammenfassend lässt sich festhalten: Mit den modularen MC-Bausteinen ist der S7-Anwender in der Lage, seine Programmierung offen zu halten für verschiedene Antriebe. Die bisher aufwendige Änderung der großen monolithischen „Superbausteine“ entfällt, da die Integration weiterer Servo- und Schrittmotorantriebe kostengünstig durch Anpassen der antriebsspezifischen Achsreferenz erfolgen kann. *gh*



**Jörg Peters**

ist Geschäftsführer der Firma Insevis, Erlangen.



**Stephan Claußnitzer**

ist Geschäftsführer und Technischer Leiter bei der Firma Inmotec Automation, Berlin.