

Siemens S7-1200 Beispielprogramm in Tia Portal: Support für Stöber-Geräte der Generation 6

Inhalt

TIA Portal.....	2
Funktionsumfang	2
Applikationen	3
Applikation STÖBER Drive Based im Kommandomodus.....	3
Einstellungen im Regler (hier SI6):.....	5
Erweitern des zyklischen Mappings	5
Bausteine.....	7
FB G6_CmdMode	7
Fehlermeldungen.....	11
Azyklisches Lesen und Schreiben von Parametern	12
G6_GetParamAdress_PN	14
SAT_FaultMessage.....	14
Datentypen.....	14
Von der SPS an den Regler:.....	14
Zugehörige UDT "To_Drive" in der SPS:.....	15
Vom Regler an die SPS:	16
Zugehörige UDT "From_Drive" in der SPS:.....	16
Achsstruktur in der SPS: UDT „AxStruct“	16
UDT „Drive_xx“	17
UDT „AcyclReqHead“, UDT „AcyclRespHead“	17

TIA Portal

Vorbemerkung:

Kenntnisse der Programmierung mit dem TIA-Portal werden vorausgesetzt. Das Erstellen und Bearbeiten der Hardwarekonfiguration sowie die grundlegende Arbeitsweise mit dem Programm müssen bekannt sein.

Das Projekt SAT_DriveBased_V14 wurde in der aktuellen Version V14 SP1 des TIA Portals erstellt. Es enthält einige Bausteine um Achsen mit Stöber-Reglern der Generation 6 (SD6, SI8, SC6) in der Betriebsart Drivebased (Stöber) anzusteuern. Die Bausteine sind nicht geschützt und können vom Anwender angepasst werden. Das Projekt mit den darin enthaltenen Softwareteilen soll als Beispiel dienen. Es wird seitens Stöber keinerlei Gewährleistung übernommen für die Funktion oder Anwendbarkeit in einer konkreten Maschine oder Anwendung.

Es gibt zwei Beispielprojekte, SAT_DriveBased_V14_1500 für eine CPU 1500_V1, und SAT_DriveBased_V14_1200_V1 für eine CPU 1200. Diese Beschreibung trifft auf beide Projekte zu. Zurzeit gibt es keine Softwareteile, die nur auf einer CPU-Familie laufen würden.

Funktionsumfang

Das Beispielprojekt enthält je einen STÖBER SD6, SI6-Einzelachs- und SI6-Doppelachs-regler. Alle drei Geräte können mit demselben FB bedient werden. Die zyklischen Daten vom und zum Antriebsregler werden per UDT an den FB übergeben, der Anwender kann das Mapping auch mit eigenen Daten erweitern.

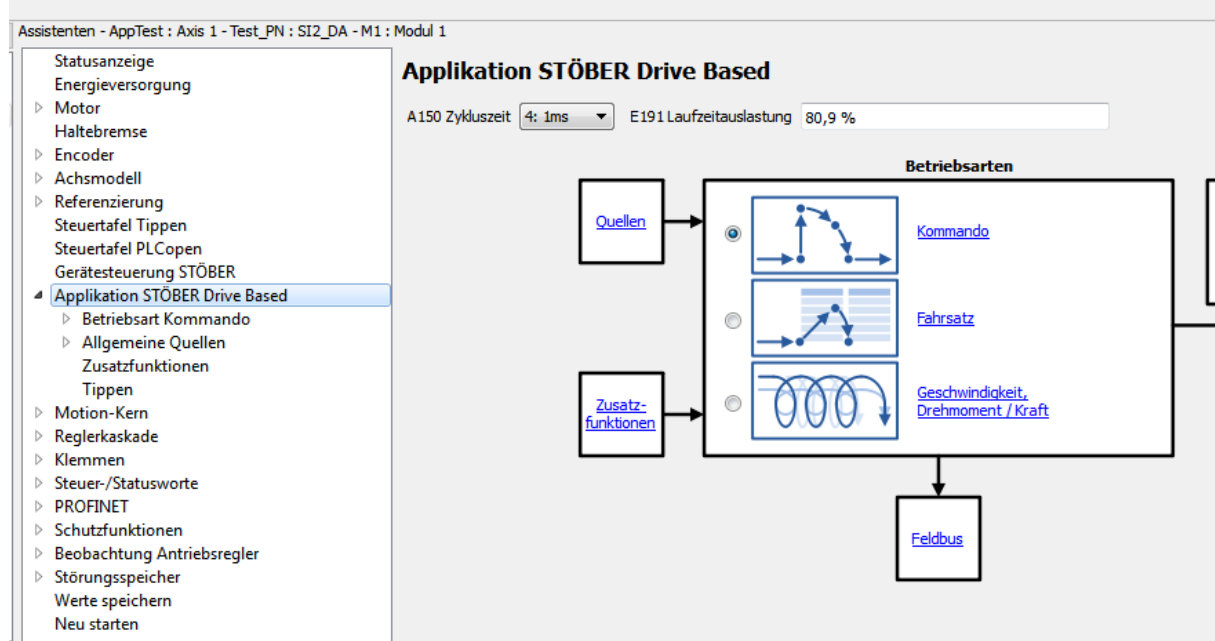
Folgende Funktionen können angesteuert werden:

- STÖBER Drive Based in der Betriebsart „Kommando“ mit Gerätesteuerung STÖBER über Profinet für SD6
- STÖBER Drive Based in der Betriebsart „Kommando“ mit Gerätesteuerung STÖBER über Profinet für SI6 Single Axis
- STÖBER Drive Based in der Betriebsart „Kommando“ mit Gerätesteuerung STÖBER über Profinet für SI6 Double Axis

Bei der Verwendung eines Doppelachsreglers muss für jede Achse ein eigener FB benutzt werden.

Applikationen

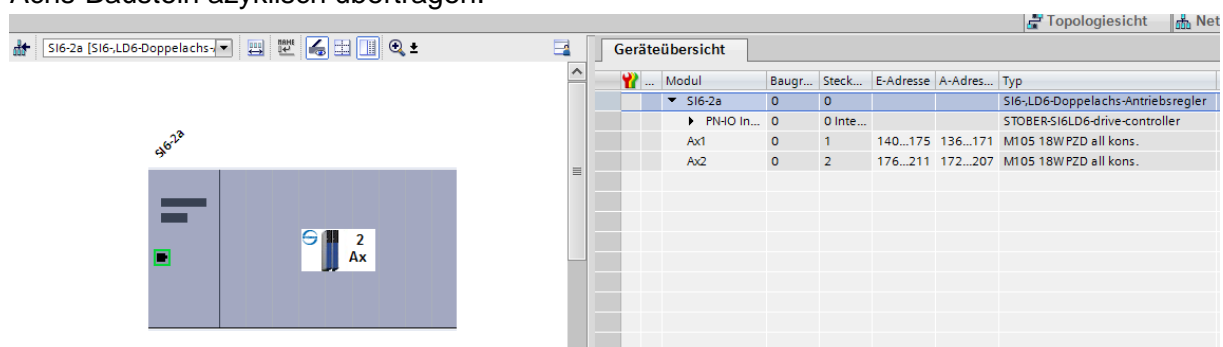
In der aktuellen Ausbaustufe unterstützen die Beispiele die Applikation Kommandomodus. Diese Applikation muß im Antriebsregler über die DriveControlSuite angewählt werden:



Durch diese Einstellung erfolgt eine Vorbelegung der Prozeßdatenschnittstelle mit Standardwerten. Diese Schnittstelle für die zyklische Kommunikation zwischen Steuerung und Regler ist schon vorbereitet für die Betriebsart Fahrsatzmodus, ein Beispielprogramm dafür folgt noch.

Applikation STÖBER Drive Based im Kommandomodus

Für die beiden Applikationen Drivebased Kommando- und Fahrsatzmodus soll das folgende Mapping vorausgesetzt werden. Dafür wird in der Hardwarekonfiguration ein 18-Wort-Ein-Ausgangsmodul verwendet. Damit können alle Kommandos angesteuert und mit Sollwerten versorgt werden. Nicht-zyklische Parameter wie Beschleunigung, Ruck usw. werden vom Achs-Baustein azyklisch übertragen.



Grundsätzlich können auch größere Module verwendet werden, nicht belegte Worte werden generell mit Wert 0 übertragen.

In der Siemens-CPU werden die Daten über den Baustein „Copy_AxStruct_to_Output“ von der SPS auf das Profinet-Modul des Reglers kopiert, durch den Baustein „Copy_Input_to_AxStruct“ werden die Daten vom Regler auf die SPS kopiert. Diese

Bausteine sollten nur einmal aufgerufen werden, sie sind in den Achs-Baustein „G6_CmdMode“ integriert. Der Anwender sollte sie deshalb nicht noch einmal aufrufen.

SD6 / SI6 Einzelachse zyklisches Mapping zum Umrichter:

Koordinate	Name	Datentyp	Einheit	Länge [Byte]
1.A180	Steuer Byte Gerät	BYTE		1
1.J37	Steuerbyte Kommando	BYTE		1
1.I210	Steuerwort	WORD		2
1.J40	Kommando	SINT		1
1.J41	Motion ID	BYTE		1
1.J42	Sollposition	DINT	1/100 unit	4
1.J43	Sollgeschwindigkeit	REAL	Unit	4
1.J56	Soll-Geschwindigkeitsoverride	REAL	1/100 %	4
1.G469	Soll-Drehmoment	REAL	1/100 %	4
1.J01	Seuer-Byte Fahrsatz	BYTE		1
1.J02	Soll-Fahrsatz	INT		2
Reserve				11

Einstellungen im Regler (hier SI6):

Empfangs-Prozessdaten RxPZD				
	Koordinate	Name	Datentyp	Länge
A90[0] PZD Setpoint Mapping Rx	1.A180	Steuer-Byte Gerät: 0000 0100 bin	BYTE	1
A90[1] PZD Setpoint Mapping Rx	1.J37	Steuer-Byte Kommando: 0000 0000 bin	BYTE	1
A90[2] PZD Setpoint Mapping Rx	1.I210	Steuerwort Applikation: 0000 0000 0000 0000 bin	WORD	2
A90[3] PZD Setpoint Mapping Rx	1.J40	Kommando: 0	SINT	1
A90[4] PZD Setpoint Mapping Rx	1.J41	Motion-ID: 0	SINT	1
A90[5] PZD Setpoint Mapping Rx	1.J42	Position: 0,00 °	DINT	4
A90[6] PZD Setpoint Mapping Rx	1.J43	Geschwindigkeit 1: 0 °/s	REAL	4
A90[7] PZD Setpoint Mapping Rx	1.J56	Geschwindigkeits-Override: 100,00 %	REAL	4
A90[8] PZD Setpoint Mapping Rx	1.G469	Soll-Drehmoment/-Kraft: 0,00 %	REAL	4
A90[9] PZD Setpoint Mapping Rx	1.J01	Steuer-Byte Fahrsatz: 0000 0000 bin	BYTE	1
A90[10] PZD Setpoint Mapping Rx	1.J02	Sollfahrsatz: 0	INT	2

Diese Daten werden zyklisch aus dem benutzerdefinierten Datentyp „To_Drive“ kopiert.

SD6 / SI6 Einzelachse zyklisches Mapping zur Steuerung:

Koordinate	Name	Datentyp	Einheit	Länge [Byte]
1.E200(0)	Status Byte 1 Gerät	BYTE		1
1.E200(1)	Status Byte 2 Gerät	BYTE		1
1.E201	Statuswort 2	WORD		2
1.I212	Status Byte Applikation	BYTE		1
1.J39	Status Byte Kommando	BYTE		1
1.I200	Statuswort Applikation	WORD		4
1.I80	Istposition	DINT	1/100 unit	4
1.I88	Istgeschwindigkeit	REAL	Unit	4
1.E90	Istmoment	REAL	1/100 %	4
1.A67	Statuswort anwenderdefiniert	WORD		2
1.E80	Betriebszustand	SINT		1
1.J302	Status-Byte Fahrsatz	BYTE		1
1.J300	Ist-Fahrsatz	INT		2
1.E48	Gerätestatus	SINT		1

Reserve				7
---------	--	--	--	---

Einstellungen im Regler (hier SI6):

Sende-Prozessdaten TxPZD

	Koordinate	Name	Datentyp	Länge
A94[0] PZD ActValue Mapping Tx	1.E200[0]	Status-Byte Gerät: 0000 0000 bin	BYTE	1
A94[1] PZD ActValue Mapping Tx	1.E200[1]	Status-Byte Gerät: 0000 0000 bin	BYTE	1
A94[2] PZD ActValue Mapping Tx	1.E201	Statuswort 2: 0000 0010 0000 1111 bin	WORD	2
A94[3] PZD ActValue Mapping Tx	1.I212	Status-Byte Applikation: 0000 0000 bin	BYTE	1
A94[4] PZD ActValue Mapping Tx	1.J39	Status-Byte Kommando: 0000 0000 bin	BYTE	1
A94[5] PZD ActValue Mapping Tx	1.I200	Statuswort Applikation: 1000 0000 0100 0000 bin	WORD	2
A94[6] PZD ActValue Mapping Tx	1.I80	Istposition: 95593,64 °	DINT	4
A94[7] PZD ActValue Mapping Tx	1.I88	Istgeschwindigkeit: -0,01379694 °/s	REAL	4
A94[8] PZD ActValue Mapping Tx	1.E90	Istmoment/-kraft: -0,17 %	REAL	4
A94[9] PZD ActValue Mapping Tx	1.A67	User Statuswort: 0000 hex	WORD	2
A94[10] PZD ActValue Mapping Tx	1.E80	Betriebszustand: 6: Einschaltbereit	SINT	1
A94[11] PZD ActValue Mapping Tx	1.J302	Status-Byte Fahrsatz: 0000 0000 bin	BYTE	1
A94[12] PZD ActValue Mapping Tx	1.J300	Istfahrsatz: 0	INT	2
A94[13] PZD ActValue Mapping Tx	1.E48	Gerätezustand: 2: Einschaltbereit	SINT	1

Diese Daten werden zyklisch auf den benutzerdefinierten Datentyp „From_Drive“ kopiert.

Erweitern des zyklischen Mappings

Es wird nicht empfohlen, das zyklische Mapping zu verkürzen, da sonst auch die Datentypen in der SPS angepaßt werden müssen.

Falls zusätzliche Daten zyklisch übertragen werden sollen, können die erforderlichen Parameter mittels der DriveControlSuite im Regler eingetragen werden. Wenn in einer Richtung mehr als 18 Worte insgesamt nötig sind, muß in der Hardwarekonfiguration der S7 ein größeres Modul eingesetzt werden. (Je Achse ist nur ein Modul möglich).

Nach einer Änderung der Prozeßdatenzuordnung im Regler muß dieser neu gestartet werden, damit die Änderungen wirksam werden. In der SPS können die ergänzten Daten dann durch direkten Zugriff auf die entsprechenden Ein-/Ausgangsbytes gelesen werden. Es ist auch möglich, die UDTs From_Axis und To_Axis um die zusätzlichen Daten zu erweitern, dann werden die Daten zyklisch im Achs-FB kopiert und landen automatisch in den Strukturen bzw. im Regler.

Azyklische Kommunikation

Verschiedene Daten werden nicht zyklisch übertragen, um den Bus nicht unnötig zu belasten. Dafür sind die Bausteine „Read_Acyclic“ und „Write_Acyclic“ im SPS-Programm enthalten, die den azyklischen Datenverkehr abwickeln. Diese Bausteine werden im Achs-FB aufgerufen, wenn nötig. Der Anwender kann weitere Daten mit Hilfe dieser Bausteine azyklisch übertragen, siehe unten die Beschreibung der Bausteine.

Im Achs-FB werden jetzt schon bei Bedarf übertragen:

SD6 / SI6 Einzelachse azyklische Kommunikation zum Umrichter:

Koordinate	Name	Datentyp	Einheit	Länge [Byte]
1.J47	Maximales Positives Moment	REAL	1/100 %	4
1.J48	Maximales negatives Moment	REAL	1/100 %	4

1.J44	Sollbeschleunigung	REAL	Units / s ²	4
1.J45	Sollverzögerung	REAL	Units / s ²	4
1.J46	Soll-Ruck	REAL	Units / s ³	4
1.I17	Schnellhaltverzögerung	REAL	Units / s ²	4

SD6 / SI6 Einzelachse azyklische Kommunikation vom Umrichter:

Koordinate	Name	Datentyp	Einheit	Länge [Byte]
1.I06	Anzahl Nachkommastellen Position	SINT		1

Bausteine

FB G6_CmdMode

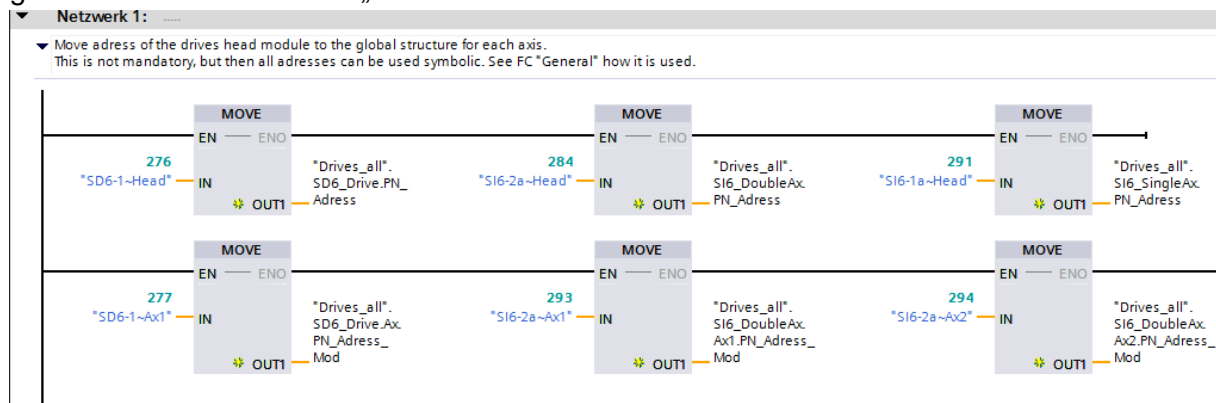
Der Baustein steuert eine Achse mit der Anwendung STÖBER Drive Based in der Betriebsart Kommandomodus.

Der Baustein ist so aufgebaut, daß zu Beginn die zyklischen Daten vom Regler eingelesen und in eine Datenstruktur kopiert werden. Die UDT dafür heißt „From_Drive“. Danach werden die Kommandos ausgewertet und mit den zyklischen Sollwerten in eine Datenstruktur eingetragen. Die UDT dafür heißt „To_Drive“. Falls nötig werden Sollwerte wie Beschleunigung etc. azyklisch an den Regler übertragen. Abschließend werden die zyklischen Daten an den Regler übertragen.

Zur einfacheren Handhabung sind die UDTs für eine Achse verpackt in eine UDT „AxStruct“, in der auch noch einige globale Daten für die Achse verwaltet werden (Profinet-Adresse des Submoduls, Anzahl Nachkommastellen, Achsnummer). Zur Unterscheidung der verschiedenen Achsregler-Typen wurden UDTs erstellt „Drive_SD6“, „Drive_SI6_single“, „Drive_SI6_double“. Diese beinhalten neben der AxStruct auch noch Daten wie die Profinet-Adresse des Reglers und das Bit „xLockAcyclic“ für diesen Regler. Dieses Bit wird benötigt, weil nicht mehrere azyklische Zugriffe auf einen Regler möglich sind. Deshalb wurde mit Hilfe des Bits xLockAcyclic ein Verriegelungsmechanismus geschaffen. Jeder Baustein, der azyklische Zugriffe auf einen Regler macht, prüft und setzt das zugehörige Bit. Nach Beenden des Datenverkehrs setzt er das Bit zurück.

Die Datenstrukturen und zusätzlichen Variablen können vom Anwender auch ohne „Hülle“ verwendet werden. Für die korrekte Funktion der Bausteine muß der Anwender sorgen.

Die Profinet-Adressen müssen zu Beginn der Programmbearbeitung einmalig in die UDTs eingetragen werden. Dies kann z.B. in einem Anlauf-OB geschehen, im Beispielpogramm geschieht das im Baustein „Main“ im ersten Netzwerk:



Die Schnittstelle des Achs-Bausteins sieht folgendermaßen aus:

G6_CmdMode			
Input		Output	
usiAxisNo	USINT	xAxisReady	BOOL
PNAdr	HW_IO	xAxisEnabled	BOOL
LADDR	HW_SUBMODULE	xBusy	BOOL
xEnableDrive	BOOL	xDone	BOOL
xQuickStop	BOOL	xStopDone	BOOL
xAxisStop	BOOL	xContEnabled	BOOL
xResetError	BOOL	xAborted	BOOL

xExecMoveVelo	BOOL		xHomingOK	BOOL
xExecMoveAbs	BOOL		xError	BOOL
xExecMoveRel	BOOL		xFBError	BOOL
xExecMoveAdd	BOOL		dwErrorID	DWORD
xExecMoveTorque	BOOL		dwFBErrorID	DWORD
xExecHoming	BOOL		lrActPos	LREAL
lrHomePos	LREAL		rActVel	REAL
lrSetPos	LREAL		xConstantVel	BOOL
rSetvel	REAL		xAccelerating	BOOL
rSetAcc	REAL		xDecelerating	BOOL
rSetDec	REAL		xSTOActive	BOOL
rSetJerk	REAL		xHWLimitPosActive	BOOL
siSetDir	SINT		xHWLimitNegActive	BOOL
rOverride	REAL			
rSetTorque	REAL			
rSpeedLimPos	REAL			
rSpeedLimNeg	REAL			
rQuickStopDec	REAL			
xJogPos	BOOL			
xJogNeg	BOOL			
rJogVel	REAL			
rJogAcc	REAL			
rJogDec	REAL			
InOut				
AxRef	AXSTRUCT			
xLockAcyclic	BOOL			

Beschreibung der Parameter:

usiAxisNo	USINT	Nummer der Achse auf dem Regler. Bei Doppelachsgeräten 1 oder 2, sonst immer 1.
PNAdr	HW_IO	Hier muß die Adresse des Profinet-Moduls angegeben werden. Im TiA-Portal stehen dazu die Systemkonstanten in der Standard-Variablen-tabelle zur Verfügung. Es muß das Element <i>Gerätename~HEAD</i> (Kopfmodul des Reglers als Profinet-Modul) eingesetzt werden.
LADDR	HW_SUB-MODULE	Hier muß die Adresse des Achs-Moduls angegeben werden. Im TiA-Portal stehen dazu die Systemkonstanten in der Standard-Variablen-tabelle zur Verfügung. Es muß das Element (Modul für die Achse des Reglers) eingesetzt werden.
xEnableDrive	BOOL	Mit High-Pegel an diesem Eingang wird der Regler freigegeben. Wenn der Ausgang „xDriveReady“ nicht High wird, ist reglerseitig die Freigabe gesperrt.
xQuickStop	BOOL	Positive Flanke an diesem Eingang löst einen Schnellhalt des Motors aus. Dazu wird die Verzögerung „rQuickStopDec“ verwendet.
xAxisStop	BOOL	Positive Flanke an diesem Eingang bremst den Motor mit der Verzögerung „rSetDec“ bis zum Stillstand ab.
xResetError	BOOL	Positive Flanke an diesem Eingang bewirkt einen Fehlerreset auf dem Regler. Mit der negativen Flanke an diesem Eingang verläßt der Baustein den Fehlerzustand und ein neues Kommando kann gestartet werden.

xExecMoveVelo	BOOL	Positive Flanke startet das Kommando „Move Velocity“. Es gelten die Sollwerte „rSetVel“, „rSetAcc“, „rSetDec“ und „rSetJerk“. Die laufende Bewegung kann durch „xAxisStop“ beendet werden oder durch eine andere Bewegung abgelöst werden.
xExecMoveAbs	BOOL	Positive Flanke startet das Kommando „Move Absolute“. Es gelten die Sollwerte „rSetVel“, „rSetAcc“, „rSetDec“ und „rSetJerk“. Die laufende Bewegung kann durch „xAxisStop“ beendet werden oder durch eine andere Bewegung abgelöst werden.
xExecMoveRel	BOOL	Positive Flanke startet das Kommando „Move Relative“. Es gelten die Sollwerte „rSetVel“, „rSetAcc“, „rSetDec“ und „rSetJerk“. Die laufende Bewegung kann durch „xAxisStop“ beendet werden oder durch eine andere Bewegung abgelöst werden.
xExecMoveAdd	BOOL	Positive Flanke startet das Kommando „Move Additive“. Es gelten die Sollwerte „rSetVel“, „rSetAcc“, „rSetDec“ und „rSetJerk“. Die laufende Bewegung kann durch „xAxisStop“ beendet werden oder durch eine andere Bewegung abgelöst werden.
xExecMoveTorque	BOOL	Positive Flanke startet das Kommando „Torque Control“. Der Sollwert für das Drehmoment wird zyklisch an den Regler übertragen (Parameter G469). Um zu vermeiden, daß der Antrieb „durchgeht“, wenn kein Gegenmoment ansteht, wird die Geschwindigkeit begrenzt (siehe unten). Das Kommando kann durch „xAxisStop“ beendet werden oder durch eine andere Bewegung abgelöst werden. Achtung! Bei falscher Kommandierung können gefährliche Situationen entstehen!
xExecHoming	BOOL	Positive Flanke startet das Referenzieren der Achse. Es gelten die Einstellungen für die Referenzierung auf dem Regler (Parameter I30 - I44, I103). Die laufende Bewegung kann durch xAxisStop beendet werden. Wenn I30 auf „5“ gesetzt ist (Referenz setzen), kann während einer laufenden Bewegung die Referenz gesetzt werden (fliegendes Referenzieren). Die Istposition der Achse wird auf den Wert „lrHomePos“ gesetzt.
lrHomePos	LREAL	Sollwert für die Referenzposition. Abhängig von den Einstellungen auf dem Regler (I30) wird dieser Wert sofort oder nach einer Referenzfahrt übernommen.
lrSetPos	LREAL	Sollposition für die aktuell gestartete Bewegung (Zielposition oder Distanz)
rSetvel	REAL	Sollgeschwindigkeit für die aktuell gestartete Bewegung
rSetAcc	REAL	Sollbeschleunigung für die aktuell gestartete Bewegung
rSetDec	REAL	Sollverzögerung für die aktuell gestartete Bewegung
rSetJerk	REAL	Sollruck für die aktuell gestartete Bewegung
siSetDir	SINT	Richtungsangabe für das Kommando Move Relative und Move Additive 0 = richtungsoptimiert (bei modulo-Achsen) 1 = positive Richtung 2 = negative Richtung

		3 = letzte Richtung beibehalten
rOverride	REAL	Geschwindigkeitsoverride für die aktuelle Bewegung (nur bei Positions- und Geschwindigkeitskommandos). Der Override wird zyklisch übertragen, kann also während der Bewegung dynamisch angepaßt werden. Der Sollwert muß zwischen 1.0 (100%) und 0.0 (0%) liegen und wird intern auf diese Werte begrenzt.
rSetTorque	REAL	Zyklischer Drehmomentsollwert im Torque-Modus. Der Sollwert muß zwischen 1.0 (100%) und 0.0 (0%) liegen. Er ist bezogen auf die aktuellen Drehmomentgrenzen des Motors (größerer Betrag von C03 bzw. C05). Siehe auch Beschreibung in der DriveControlSuite.
rSpeedLimPos	REAL	Im Torque-Modus wird die maximale Geschwindigkeit in positiver Richtung auf diesen Wert begrenzt. Der Sollwert muß zwischen 1.00 (100%) und 0.01 (1%) liegen. Er ist bezogen auf die maximale Geschwindigkeit I10. Achtung! Parameter G473 im Regler muss auf „4“ eingestellt sein (Quelle = Parameter)
rSpeedLimNeg	REAL	Im Torque-Modus wird die maximale Geschwindigkeit in negativer Richtung auf diesen Wert begrenzt. Der Sollwert muß zwischen 1.00 (100%) und 0.01 (1%) liegen. Er ist bezogen auf die maximale Geschwindigkeit I10. Achtung! Parameter G476 im Regler muss auf „4“ eingestellt sein (Quelle = Parameter)
rQuickStopDec	REAL	Sollverzögerung für einen Schnellhalt.
xJogPos	BOOL	Wenn kein anderes Kommando gestartet ist, kann die Achse im Tipbetrieb mit diesem Eingang in positiver Richtung bewegt werden. Es gelten die Sollwerte „rJogVel“, „rJogAcc“ und „rJogDec“.
xJogNeg	BOOL	Wenn kein anderes Kommando gestartet ist, kann die Achse im Tipbetrieb mit diesem Eingang in negativer Richtung bewegt werden. Es gelten die Sollwerte „rJogVel“, „rJogAcc“ und „rJogDec“.
rJogVel	REAL	Sollgeschwindigkeit im Tipbetrieb. „rOverride“ ist nicht wirksam.
rJogAcc	REAL	Sollbeschleunigung im Tipbetrieb.
rJogDec	REAL	Sollverzögerung im Tipbetrieb.
AxRef	AXSTRUCT	Datenstruktur für eine Achse. Hier werden die zyklischen Daten für eine Achse verwaltet. Die Anordnung der Daten ist fest verknüpft mit der Reihenfolge der Parameter auf dem Regler (siehe Parameter A90 ff). Der Anwender kann das Mapping erweitern, siehe unten.
xLockBit	BOOL	Es darf immer nur ein azyklischer Zugriff auf einen Regler aktiv sein. Deshalb muss es pro Regler eine globale Variable geben, die immer gesetzt wird, wenn ein Zugriff gestartet wird, und zurückgesetzt, wenn der Zugriff beendet ist. Der Anwender muss dafür sorgen, daß kein zweiter Auftrag an denselben Regler gestartet wird, solange „xLockBit“ aktiv ist.

xAxisReady	BOOL	Die Achse ist einschaltbereit
xAxisEnabled	BOOL	Die Achse ist freigegeben, der Regler ist aktiv
xBusy	BOOL	Ein Kommando wurde gestartet, die Abarbeitung läuft
xDone	BOOL	Das gestartete Kommando wurde komplett ausgeführt. Falls ein Kommando durch ein anderes abgelöst wurde, bevor es beendet war, gilt xDone für das zuletzt gestartete Kommando. xDone bleibt nur solange High, wie der zugehörige Befehlseingang ansteht. Wird der Befehl vorher Low, steht xDone nur für einen Zyklus an. xDone wird nicht gesetzt bei den Befehlen MoveTorque und MoveVelo sowie den Jogging-Befehlen.
xStopDone	BOOL	Das Kommando xAxisStop wurde komplett ausgeführt, die Achse ist im Stillstand.
xContEnabled	BOOL	Die zuletzt ausgeführte Bewegung kann fortgesetzt werden.
xAborted	BOOL	Die vorher ausgeführte Bewegung wurde abgebrochen.
xHomingOK	BOOL	Die Referenzposition im Regler ist gesetzt und gültig.
xError	BOOL	Im Antriebsregler oder im Ablauf des FB ist ein Fehler aufgetreten.
xFBError	BOOL	In den eingebundenen Siemens-Standardbausteinen ist ein Fehler aufgetreten.
dwErrorID	DWORD	Fehlernummer zu xError (siehe unten)
dwFBErrorID	DWORD	Fehlernummer aus dem Siemens-Baustein (siehe Siemens-Dokumentation zu RDREC, WRREC)
lrActPos	LREAL	Aktuelle Position der Achse in der parametrisierten Einheit.
rActVel	REAL	Aktuelle Geschwindigkeit der Achse in der parametrisierten Einheit.
xConstantVel	BOOL	Die Achse fährt mit konstanter Geschwindigkeit.
xAccelerating	BOOL	Die Achse beschleunigt.
xDecelerating	BOOL	Die Achse bremst.
xSTOActive	BOOL	Die Sicherheitsfunktion STO (Safe torque off) wurde auf dem Regler ausgelöst. Der Motor wird nicht mehr bestromt.
xHWLimitPosActive	BOOL	Der Endschalter in positiver Richtung wurde betätigt.
xHWLimitNegActive	BOOL	Der Endschalter in negativer Richtung wurde betätigt.

Fehlermeldungen

Aus dem Baustein G6_CmdMode werden folgende Fehlernummern gemeldet:

```

Fault[0] := 'No Fault';
Fault[1] := 'Mode not Commandmode';
Fault[2] := 'TimeOut Comm';
Fault[3] := 'Drive fault';
Fault[4] := 'No position command';

```

Im Datenbaustein SAT_G6_FaultDB sind die Texte exemplarisch als Array of String hinterlegt. Dieser DB kann vom Anwender verwendet und erweitert werden.

Azyklisches Lesen und Schreiben von Parametern

Funktionsbaustein zum Lesen von Stöber-Parametern aus dem Regler:

G6_Read_Acyclic			
Input		Output	
usiAxisNo	USINT	xBusy	BOOL
sParam	STRING	xDone	BOOL
xRaedDWord	BOOL	xError	BOOL
xReadReal	BOOL	xTimeOut	BOOL
PNAdr	HW_IO	dwErrorID	DWORD
tTimeOutVal	TIME	dwValue	DWORD
		rValue	REAL
InOut			
xLockBit	BOOL		

Funktionsbaustein zum Senden von Daten an Stöber-Parameter im Regler:

G6_Write_Acyclic			
Input		Output	
usiAxisNo	USINT	xBusy	BOOL
sParam	STRING	xDone	BOOL
xSendDInt	BOOL	xError	BOOL
xSendReal	BOOL	xTimeOut	BOOL
diValue	DINT	dwErrorID	DWORD
rValue	REAL		
PNAdr	HW_IO		
tTimeOutVal	TIME		
InOut			
xLockBit	BOOL		

Beschreibung der Parameter:

usiAxisNo	USINT	Nummer der Achse auf dem Regler. Bei Doppelachsgeräten 1 oder 2, sonst immer 1.
sParam	STRING	Name des Parameters (Stöber-Bezeichnung), der gelesen/geschrieben werden soll. Z.B. „E84“. Der Parameter wird intern auf die korrekte Index/Subindex-Adresse für Profinet umgerechnet. Dabei kann maximal auf die Parameternummer 511 zugegriffen werden.
PNAdr	HW_IO	Hier muß die Adresse des Profinet-Moduls angegeben werden. Im TiA-Portal stehen dazu die Systemkonstanten in der Standard-Variablentabelle zur Verfügung. Es muß das Element <i>Gerätename~HEAD</i> (Kopfmodul des Reglers als Profinet-Modul) eingesetzt werden.
tTimeOutVal	TIME	Hier kann ein Zeitwert für die Timeout-Überwachung vorgegeben werden. Je nach eingestellter Aktualisierungszeit des Profinet kann die azyklische Kommunikation auch Sekunden dauern.
xReadDWord	BOOL	Steigende Flanke an diesem Eingang startet das Lesen des Parameters im Format DWORD. Das Ergebnis steht am Ausgang dwValue zur Verfügung.
xReadReal	BOOL	Steigende Flanke an diesem Eingang startet das Lesen des Parameters im Format REAL. Das Ergebnis steht am Ausgang rValue zur Verfügung.
xSendDInt	BOOL	Steigende Flanke an diesem Eingang startet das Schreiben des Parameters im Format DINT. Der Wert wird am Eingang „diValue“ vorgegeben.

xSendReal	BOOL	Steigende Flanke an diesem Eingang startet das Schreiben des Parameters im Format REAL. Der Wert wird am Eingang „rValue“ vorgegeben
dwValue, diValue	DWORD, DINT	Als Eingang: Sollwert für den Parameter „sParam“ im Format DINT, als Ausgang: Wert des Parameters im Format DWORD
rValue	REAL	Als Eingang: Sollwert für den Parameter „sParam“ im Format REAL, als Ausgang: Wert des Parameters im Format REAL
xBusy	BOOL	High-Pegel an diesem Ausgang steht an, solange die Bearbeitung des Bausteins läuft und kein Ergebnis vorliegt.
xDone	BOOL	High-Pegel an diesem Ausgang steht an, wenn die Bearbeitung des Bausteins fertig ist und ein Ergebnis vorliegt. Beim Senden heißt das, daß der Wert korrekt übertragen wurde, beim Lesen ist der Wert an dem entsprechenden Ausgang gültig.
xError	BOOL	High-Pegel an diesem Ausgang steht an, wenn bei der Bearbeitung des Bausteins ein Fehler auftrat. Am Ausgang „dwErrorID“ kann eine Fehlernummer abgelesen werden.
xTimeOut	BOOL	Die Bearbeitung des Bausteins wurde abgebrochen, weil innerhalb der Zeit „tTimeOutVal“ kein Ergebnis vorlag. Der Wert wurde nicht übertragen, bzw. nicht gelesen.
dwErrorID	DWORD	An diesem Ausgang steht im Fehlerfall der Wert aus dem Siemens-FB zur Ansicht. Siehe Dokumentation der Fa. Siemens zu den Bausteinen RDREC, WRREC, Parameter „STATUS“ bzw. Stöber-Dokumentation zu Profinet bei Schreib-/Lese Fehlern. Außerdem wird 16#0000 03E7 ausgegeben, wenn beide Eingänge xReadDWord und xReadReal gleichzeitig anstehen (bzw. beide xSend....).
xLockBit	BOOL	Es darf immer nur ein azyklischer Zugriff auf einen Regler aktiv sein. Deshalb muß es pro Regler eine globale Variable geben, die immer gesetzt wird, wenn ein Zugriff gestartet wird, und zurückgesetzt, wenn der Zugriff beendet ist. Der Anwender muß dafür sorgen, daß kein zweiter Auftrag an denselben Regler gestartet wird, solange „xLockBit“ aktiv ist.

G6_GetParamAdress_PN

Diese Funktion ermittelt aus der angegebenen Stöber-Parameter-Nummer die zugehörige Index- und Subindex-Adresse für die azyklische Kommunikation via Profinet. Es können maximal die Parameternummern bis 511 übertragen werden. Die Ermittlung der Adresse ist auch in der Stöber-Dokumentation zur Profinet-Anschaltung erläutert.

SAT_FaultMessage

Dies ist eine einfache Funktion, um die Fehlernummer „dwErrorID“, die der Achs-FB ausgibt, in einen Text umzusetzen. Der Text wird aus dem DB „SAT_G6_Fault_DB“ entnommen.

Datentypen

UDT „AxStruct“









Zur einfacheren Verwaltung der Achsen gibt es eine Achsstruktur. Sie enthält alle Daten, die zyklisch zwischen Regler und SPS ausgetauscht werden, sowie einige globale Daten, die grundsätzlich für die Funktion benötigt werden. Zur „AxStruct“ in der vorliegenden Form gehört zwingend ein bestimmtes Mapping der zyklischen Daten, dieses muß auf der SPS und dem Regler gleich ausgeführt sein.

Im SPS-Programm werden die Daten in den Funktionen Copy_Input_to_AxStruct bzw. Copy_AxStruct_to_Output durch einen Zugriff mittels DPRD_DAT bzw. DPWR_DAT konsistent gelesen bzw. geschrieben. Wenn der Anwender das Mapping im Regler und die Achsstruktur in der SPS gleich erweitert, werden die zusätzlichen Daten automatisch mit kopiert. Wenn die Ergänzungen nicht in der Achsstruktur eingetragen werden, muß der Anwender selber dafür sorgen, daß die Daten im SPS-Programm verwendet werden können. Parametrierung in DriveControlSuite (hier für SI6):

Von der SPS an den Regler:

Empfangs-Prozessdaten RxPZD				
	Koordinate	Name	Datentyp	Länge
A90[0] PZD Setpoint Mapping Rx	1.A180	Steuer-Byte Gerät: 0000 0000 bin	BYTE	1
A90[1] PZD Setpoint Mapping Rx	1.J37	Steuer-Byte Kommando: 0000 0000 bin	BYTE	1
A90[2] PZD Setpoint Mapping Rx	1.I210	Steuerwort Applikation: 0000 0000 0000 0000 bin	WORD	2
A90[3] PZD Setpoint Mapping Rx	1.J40	Kommando: 0	SINT	1
A90[4] PZD Setpoint Mapping Rx	1.J41	Motion-ID: 0	SINT	1
A90[5] PZD Setpoint Mapping Rx	1.J42	Position: 0,00 °	DINT	4
A90[6] PZD Setpoint Mapping Rx	1.J43	Geschwindigkeit 1: 0 °/s	REAL	4
A90[7] PZD Setpoint Mapping Rx	1.J56	Geschwindigkeits-Override: 0,00 %	REAL	4
A90[8] PZD Setpoint Mapping Rx	1.G469	Soll-Drehmoment/-Kraft: 0,00 %	REAL	4
A90[9] PZD Setpoint Mapping Rx	1.J01	Steuer-Byte Fahrsatz: 0000 0000 bin	BYTE	1
A90[10] PZD Setpoint Mapping Rx	1.J02	Sollfahrsatz: 0	INT	2

Zugehörige UDT "To_Drive" in der SPS:

To_Drive									
	Name	Datentyp	Defaultwert	Erreichbar a...	Schrei...	Sichtbar i...	Einstellwert	Kommentar	
1	 CtrlByteDev_A180	Byte	 16#0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Controlbyte Device	
2	 CtrlByteCmd_J37	Byte	16#0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Controlbyte Commandmode	
3	 CtrlWord_I210	Word	16#0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Controlword	
4	 Command_J40	SInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Command	
5	 MotionID_J41	SInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Motion-ID	
6	 SetPos_J42	DInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Setposition for Command	
7	 SetSpeed_J43	Real	0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Setspeed for Command	
8	 SetSpeedOver_J56	Real	0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Setspeedoverride	
9	 SetTorque_G469	Real	0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Settorque for command	
10	 CmdByteSetMode_J01	Byte	16#0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Controlbyte Commandset-Mode	
11	 SetCmdSet_J02	Int	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Set Commandndset	

Vom Regler an die SPS:

Sende-Prozessdaten TxPZD

	Koordinate	Name	Datentyp	Länge
A94[0] PZD ActValue Mapping Tx	1.E200[0]	Status-Byte Gerät: <offline> (0000 0010 bin)	BYTE	1
A94[1] PZD ActValue Mapping Tx	1.E200[1]	Status-Byte Gerät: <offline> (0000 0000 bin)	BYTE	1
A94[2] PZD ActValue Mapping Tx	1.E201	Statuswort 2: <offline> (0000 0000 0000 1100 bin)	WORD	2
A94[3] PZD ActValue Mapping Tx	1.I212	Status-Byte Applikation: 0000 0000 bin	BYTE	1
A94[4] PZD ActValue Mapping Tx	1.J39	Status-Byte Kommando: 0000 0000 bin	BYTE	1
A94[5] PZD ActValue Mapping Tx	1.I200	Statuswort Applikation: <offline> (1000 0000 0000 0010 bin)	WORD	2
A94[6] PZD ActValue Mapping Tx	1.I80	Istposition: <offline> (0,00 °)	DINT	4
A94[7] PZD ActValue Mapping Tx	1.I88	Istgeschwindigkeit: <offline> (0 °/s)	REAL	4
A94[8] PZD ActValue Mapping Tx	1.E90	Istmoment/-kraft: <offline> (-1,08 %)	REAL	4
A94[9] PZD ActValue Mapping Tx	1.A67	Statuswort anwenderdefiniert: <offline> (0000 hex)	WORD	2
A94[10] PZD ActValue Mapping Tx	1.E80	Betriebszustand: <offline> (2: Störung)	SINT	1
A94[11] PZD ActValue Mapping Tx	1.J302	Status-Byte Fahrsatz: <offline> (0000 0000 bin)	BYTE	1
A94[12] PZD ActValue Mapping Tx	1.J300	Istfahrsatz: <offline> (0)	INT	2
A94[13] PZD ActValue Mapping Tx	1.E48	Gerätezustand: <offline> (5: Störung)	SINT	1

Zugehörige UDT "From_Drive" in der SPS:

From_Drive									
	Name	Datentyp	Defaultwert	Erreichbar a...	Schrei...	Sichtbar i...	Einstellwert	Kommentar	
1	StatusByte1_Dev_E200_0	Byte	16#0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	first statusbyte from Device	
2	StatusByte2_Dev_E200_1	Byte	16#0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	second statusbyte from Device	
3	StatusWord_E201	Word	16#0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Statusword	
4	StatusByteAppl_I212	Byte	16#0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Statusbyte Application	
5	StatusbyteCmd_J39	Byte	16#0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Statusbyte Command	
6	StatusWordAppl_I200	Word	16#0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Statusword Application	
7	ActPos_I80	Dint	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Actual Position	
8	ActSpeed_I88	Real	0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Actual Speed	
9	ActTorque_E90	Real	0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Actual Torque	
10	StatusWordUser_A67	Word	16#0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	User defined Statusword	
11	OpStatus_E80	Sint	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Operational Status	
12	StatusByteCmdSet_J302	Byte	16#0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Statusbyte Commandset	
13	ActCmdSet_J300	Int	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Actual Commandset	
14	DevStatus_E48	Sint	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Actual status of device	

Achsstruktur in der SPS: UDT „AxStruct“

AxStruct									
	Name	Datentyp	Defaultwert	Erreichbar a...	Schrei...	Sichtbar i...	Einstellwert	Kommentar	
1	Cycl_FromDrive	*From_Drive*		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
2	Cycl_ToDrive	*To_Drive*		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
3	PN_Address_Mod	HW_SUBMODULE	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Profinet adress of module for the axis	
4	iNoOfDigits	Int	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	number of digits in actual position I80	
5	usiAxisNo	USInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Axis number on the drive (1..2)	

Die UDT enthält die zyklischen Daten vom und an den Regler sowie globale Informationen zur Achse: Die Profinet-Adresse des Moduls (~Ax1 bzw. ~Ax2), die Anzahl Nachkommastellen für die Achsposition (I06) sowie die Achsnummer auf dem Regler.

UDT „Drive_xx“

Durch diese reglerspezifische UDT wird ein kompletter Regler beschrieben

Drive_SD6								
	Name	Datentyp	Defaultwert	Erreichbar a...	Schrei...	Sichtbar i...	Einstellwert	Kommentar
1	▶ Ax	*AxStruct		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Data from/to axis
2	LockAcyclic	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	acyclic data exchange is active, waiting for end
3	PN_Address	HW_IO	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Profinet adress of device

Drive_SI6_double								
	Name	Datentyp	Defaultwert	Erreichbar a...	Schrei...	Sichtbar i...	Einstellwert	Kommentar
1	▶ Ax1	*AxStruct		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Data from/to axis 1 of this drive
2	▶ Ax2	*AxStruct		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Data from/to axis 2 of this drive
3	LockAcyclic	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	acyclic data exchange is active, waiting for end
4	PN_Address	HW_IO	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Profinet adress of device

Drive_SI6_single								
	Name	Datentyp	Defaultwert	Erreichbar a...	Schrei...	Sichtbar i...	Einstellwert	Kommentar
1	▶ Ax	*AxStruct		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Data from/to axis
2	LockAcyclic	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	acyclic data exchange is active, waiting for end
3	PN_Address	HW_IO	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Profinet adress of device

Darin ist für jede Achse eine UDT „AxStruct“ angelegt sowie als globale Information die Profinet-Adresse des Reglers (~HEAD) und das Verriegelungsbit für den Azyklischen Datenverkehr.

Im Beispielpogramm sind alle Achsen in einem Datenbaustein „Drive_All“ zusammengefaßt, dadurch vereinfacht sich die (symbolische) Programmierung. Der Anwender kann den Datenbaustein je nach Anforderung entsprechend erweitern.

UDT „AcyclReqHead“, UDT „AcyclRespHead“

Diese beiden UDTs werden für die Abwicklung der azyklischen Kommunikation benötigt. Sie dürfen nicht verändert werden!