

Bausteine für die Simatic S7 und die Profibusanschaltung SERVOSTAR® 300/400/600 und S700

Inhaltsverzeichnis

Hinweise für das S7-Projekt „Sv14_v3c“ für Servostar® 300/400/600 & S700	2
Step7-Hardwarekonfiguration	6
Bausteinübersicht	9
Anmerkungen	9
Eingang- und Ausgangsschnittstelle vom FB14 - Axis_01_FB	10
FB14 - Axis_01_FB	10
DB 70 - IF_DB	10
GenLow, GenHigh, GenStartUp und GenSysTime	11
Adressierung von dem Hardware-Interface an den FB14 - Axis_01_FB	11
Gesamtübersicht der Schnittstelle vom FB14 - Axis_01_FB	13
Programmbeispiele zur Umrechnung der reglerinternen Einheiten in Benutzer/SI-Einheiten	17
OB1 - Netzwerk 17: "IF_DB".ActualPosition (see Pnu 1800: SI-Unit)	17
OB1 - Netzwerk 18: "IF_DB".ActualSpeed (see Pnu 1815: SI-Unit)	17
OB1 - Netzwerk 19: "IF_DB".ActualCurrent (see Pnu 1688: SI-Unit)	17
OB1 - Netzwerk 20: "IF_DB".ActualPressure	18
OB1 - Netzwerk 21: "IF_DB".ActualFlow	18
Inbetriebnahme FB14 - Axis_01_FB	19
S7-SPS - AUS/EIN-Schalten	19
S7-SPS - Run/Stop-Schalter	19
iAck =1	19
Initialisierung	19
Betriebsart setzen	19
Betrieb freigeben	19
Anforderungen für die Betriebsart Positionieren	20
Tippbetrieb	20
Referenzfahrt (Homing)	20
Start eines EEPROM oder RAM Fahrauftrags (MotionTask)	20
Start von dem Direktfahrauftrag (RAM und hat die Nummer 0)	20
Betriebsart Drehzahl digital	22
Betriebsart Drehmoment digital	22
Betriebsart Elektrisches Getriebe	22
Betriebsart Servopumpe digital – Drehzahl	23
Makro-Programmierung	24
Parameter schreiben	27
Parameter lesen	27
Zusätzliche Profibus-Funktionen	29
VAT14 Variablentabelle	29
Beispiel für eine Schrittkette „Initialisierung und Enablen Servostar“	30
Beispiel für eine Schrittkette „Starte Fahrsätze im Servostar“	31
Example for the “RAM MotionTask 201 in the Servostar“	33
Bausteine für Servostar-Parameter	35
DB 20 - Axis_01_WriteDataDB	35
DB 21 - Axis_01_ReadDataDB	35
FB16 Axis_01_Write	36
FB17 Axis_01_Read	37
FB18 Axis_01_Compare	38
Tipps und Infos:	39
Fehlermeldungen von den Bausteinen FB14, FB16, FB17 und FB18	39
Override über Profibus	39
Aktivieren der Trajektorie/Profil Beschleunigung Sin^2 bei Direktfahrsatz Nr. 0	40
Bitcodierung der Fehler – PNU 1001 - ERRCODE	43
Bitcodierung der Warnungen und Herstellerspezifisches Statusregister – PNU 1002 - DRVSTAT	44
Switch ON and Start MotionTask timing diagram	45
Kill MotionTask timing diagram	46

Hinweise für das S7-Projekt „Sv14_v3c“ für Servostar[®] 300/400/600 & S700

Bisher erschienene Ausgaben:

Revision/Ausgabe	Comment/Bemerkungen	Author/Autor
A, 04/2018	first edition / Erstausgabe	KeH

Warenzeichen

- Servostar ist ein eingetragenes Warenzeichen der Kollmorgen Corporation
- Simatic, Step7, TIA-Portal sind geschützte Marken der Siemens AG
- Profibus und ProfiNet sind geschützte Marken der PROFIBUS and PROFINET International(PI)

Dieses Dokument ist gültig für die PROFIBUS-DP Anbindung der Kollmorgen Servoverstärker Servostar[®] 300/400/600 & S700 an eine Simatic S7 -300 / 400 Steuerung.

Im folgenden Text werden die Servoverstärker Servostar[®] 300/400/600 & S700 kurz als „Servostar“ bezeichnet.



**Verwenden Sie das S7-Projekt „Sv14_v3c“ niemals unverändert in einer Anwendung.
Das S7-Projekt „Sv14_v3c“ ist ein Beispiel wie der Servostar in ein S7-Projekt integriert werden kann. Dieses Projektbeispiel muss immer an die vorhandene Anwendung angepasst werden.**



KOLLMORGEN Europe GmbH haftet nicht für Schäden und schliesst alle Ansprüche aus, die sich durch den Einsatz des S7-Projekts „Sv14_v3c“ oder Programmteilen daraus ergeben könnten.



Beachten und befolgen Sie unbedingt die Sicherheits- und Warnhinweise der Hersteller zu den jeweiligen Komponenten.

Gute Kenntnisse in Simatic Step7, der Steuerungs- und Antriebstechnik werden vorausgesetzt.

Ein einwandfreies Verhalten der kompletten Servoachse bestehend aus S7-SPS und Servostar und Motor erfordert eine korrekt durchgeführte Inbetriebnahme.

Alle Verweise auf das Handbuch beziehen sich auf die technische Beschreibung „srprof_e.pdf / srprof_d.pdf“

Die verwendeten Begriffe und Abkürzungen entsprechen der Servostar Setup Software DriveGui bzw. dem Handbuch in der Sprache „English“.

Das S7-Projekt „Sv14_v3c“ ist erstellt in der Sprache English mit Mnemonik deutsch und der Version:



Das S7-Projekt „Sv14_v3c“ besteht aus den folgenden Komponenten:

- Hardwarekonfiguration (system data) für eine Simatic S7-CPU317-2DP – 6ES7 317-2AJ10-0AB0 / V2.1
- Hardwarekonfiguration (system data) für eine VIPA-CPU315-SB – 315-2 AG12
- S7-Program – enthält die Bausteine und Symbolik

SIMATIC Manager - [Sv14_V3c -- C:\Siemens\proj\Sv14_V3c]

Objektname Symbolischer Name Erstsprache Größe im Arbeitsspei... Typ Version (Header) Name (Header)

DB1	GenCycleOB	FUP	2726	Organisationsbaustein	0.1	
DB82	GenIOFIOB	FUP	38	Organisationsbaustein	0.1	
DB85	GenFIIOB	FUP	38	Organisationsbaustein	0.1	
DB86	GenRackFaultDB	FUP	38	Organisationsbaustein	0.1	
DB100	GenStartupOB	AWL	48	Organisationsbaustein	0.1	
DB121	GenProgErrOB	FUP	38	Organisationsbaustein	0.1	
DB122	GenModErrOB	FUP	38	Organisationsbaustein	0.1	
FB14	Axis_01_FB	FUP	8054	Funktionsbaustein	0.5	Axis_01
FB16	Axis_01_Write	AWL	748	Funktionsbaustein	0.2	Axis_01
FB17	Axis_01_Read	AWL	938	Funktionsbaustein	0.2	Axis_01
FB18	Axis_01_Compare	AWL	802	Funktionsbaustein	0.1	Axis_01
DB14	Axis_01_DB	DB	392	Instanzdatenbaustei...	0.0	
DB16	Axis_01_WriteDB	DB	84	Instanzdatenbaustei...	0.0	
DB17	Axis_01_ReadDB	DB	100	Instanzdatenbaustei...	0.0	
DB18	Axis_01_CompareDB	DB	86	Instanzdatenbaustei...	0.0	
DB20	Axis_01_WriteDataDB	DB	844	Datenbaustein	0.1	Axis_01
DB21	Axis_01_ReadDataDB	DB	844	Datenbaustein	0.1	Axis_01
DB70	IF_DB	DB	164	Datenbaustein	0.1	Axis_01
UDT10	GenTimerUDT	AWL	---	Datentyp	0.1	GEN_TIME
UDT27	IFGroupToUnitUDT	AWL	---	Datentyp	0.1	IF_G_U
UDT28	IFUnitToGroupUDT	AWL	---	Datentyp	0.1	IF_U_G
UDT141	M_Axis_01_Request	AWL	---	Datentyp	0.2	Axis_01
UDT143	M_Axis_01_MaMsg	AWL	---	Datentyp	0.1	Axis_01
UDT145	M_Axis_01_State	AWL	---	Datentyp	0.1	Axis_01
UDT146	M_Axis_01_InData	AWL	---	Datentyp	0.2	Axis_01
UDT147	M_Axis_01_OutData	AWL	---	Datentyp	0.1	Axis_01
UDT148	M_Axis_01_PnuStruct	AWL	---	Datentyp	0.2	Axis_01
UDT149	M_Axis_01_RcvSend	AWL	---	Datentyp	0.1	Axis_01
VAT14	VAT14	---	---	Variablen-tabelle	0.1	
SFC12	D_ACT_DP	AWL	---	Systemfunktion	1.0	D_ACT_DP
SFC14	DPRD_DAT	AWL	---	Systemfunktion	1.0	DPRD_DAT
SFC15	DPWR_DAT	AWL	---	Systemfunktion	1.0	DPWR_DAT
SFC20	BLKMOV	AWL	---	Systemfunktion	1.0	BLKMOV
SFC21	FILL	AWL	---	Systemfunktion	1.0	FILL
SFC64	TIME_TCK	AWL	---	Systemfunktion	1.0	TIME_TCK

Eigenschaften - Bausteinordner offline

Allgemein Bausteine Prüfsummen Operandenvorrang Füllstand

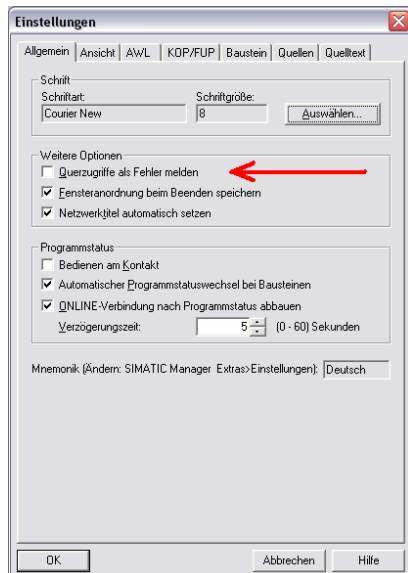
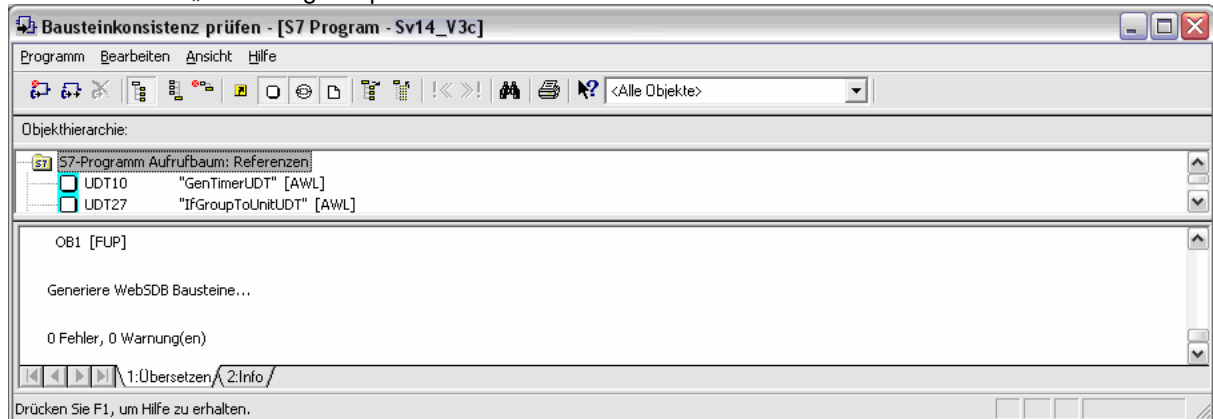
Verhalten wie in STEP7 < V5.2 Empfohlen für symbolische Programmierung

Absolutwert hat Vorrang	<input type="radio"/> Symbole werden bei allen Zugriffen (E,A,M,T,Z und DB) aus der Symboltabelle und den DB aktuell übernommen	<input type="radio"/> Ausgenommen: Symbol-Zugriffe auf DB bleiben wie im Codebaustein programmiert
Symbol hat Vorrang	<input type="radio"/> Ausgenommen: Bei Zugriffen in strukturell unveränderte Datentypen werden Symbole aktuell übernommen	<input checked="" type="radio"/> Bei allen Zugriffen (E,A,M,T,Z und DB)

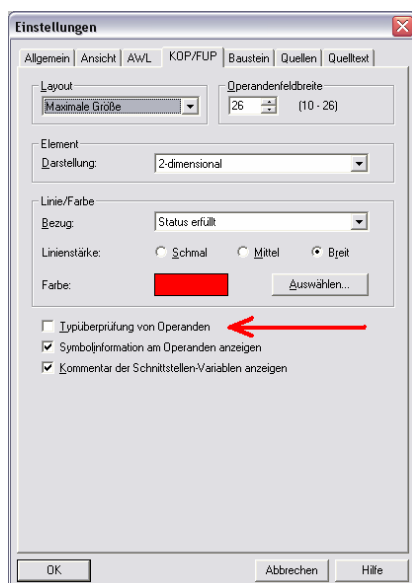
OK Abbrechen Hilfe

Die Eigenschaften des Baustein-Ordnerns Offline – Operandenvorrang sind: „Symbol hat Vorrang bei allen Zugriffen (E,A,M,T,Z und DB)“. Damit kann problemlos die Datenstruktur eines FBs, UDTs oder DBs geändert und erweitert werden. Step7 behält dabei im S7-Projekt die symbolische Adresse bei und ändert automatisch dazu passende die absolute Adresse.

Die gesamte Datenstruktur des Projektes wird mit „Bausteinkonsistenz prüfen“ und „Alles übersetzen“ automatisch aktualisiert. Wird anschliessend die Meldung: „0 Fehler, 0 Warnungen“ nicht angezeigt, enthält das S7-Programm in den Bausteinen noch manuell zu behebende Adresskonflikte. Dies ist manchmal bei Multi-Instanzen der Fall. Diese Adresskonflikte werden dann im KOP/AWL/FUP-Editor manuell in den Bausteinen mit „Datei-Zugriffe prüfen und aktualisieren“ behoben.



Werden Zugriffe auf die Instanzdaten eines DBs weiterhin **ROT** bzw. als Konflikt dargestellt, muss im KOP/AWL/FUP-Editor: EXTRAS\Einstellungen\Allgemein „Querzugriffe als Fehler melden“ deaktiviert werden. Das ist auch die Voreinstellung vom S7-Manager. Anschliessend muss das Programm nochmals übersetzt werden.

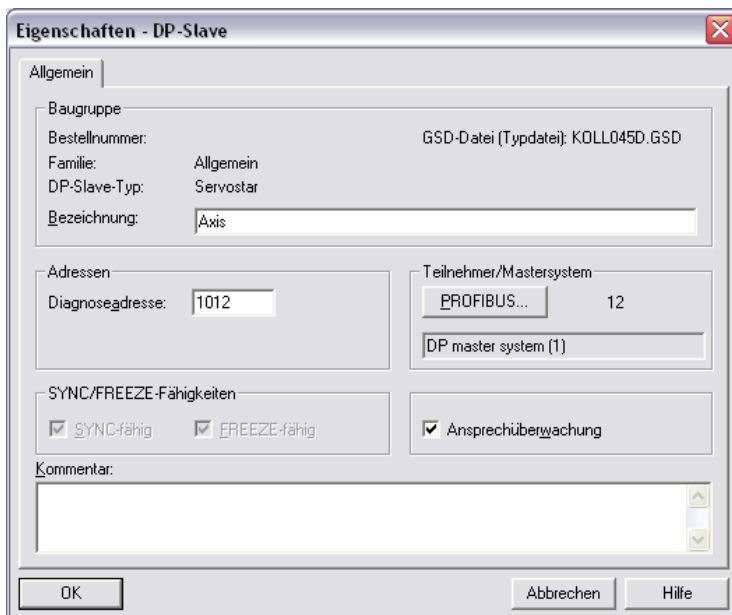
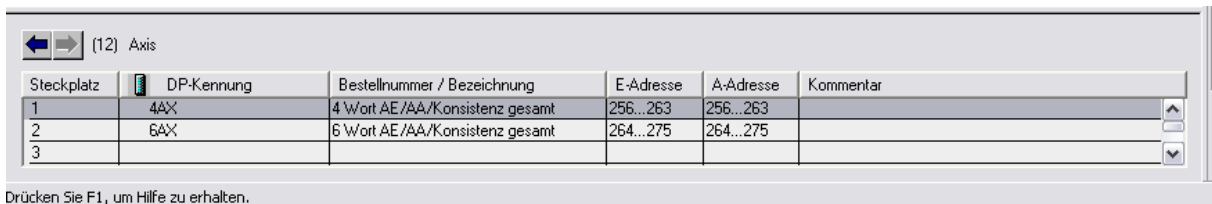
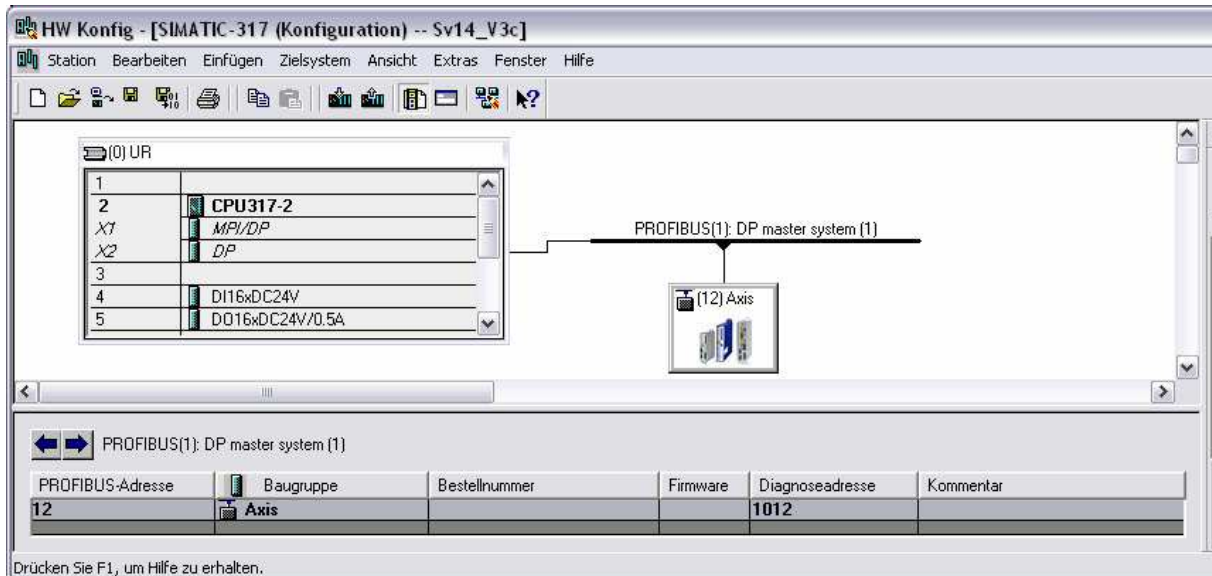


Zur vollständigen FUP-Darstellung der Netzwerke muss im KOP/AWL/FUP-Editor: EXTRAS\Einstellungen\Kop/FUP die „Typüberprüfung von Operanden“ deaktiviert sein.

Step7-Hardwarekonfiguration

unter anderem bitte beachten:

- * Diagnoseadresse
- * Eingang-Adresse / Ausgang-Adresse
- * GSD-file-Version
- * Der Servostar stellt sich automatisch auf Übertragungsgeschwindigkeit (Baudrate) ein.
- * Ansprechüberwachung Watchdog



Servostar Einstellungen mit der DriveGui

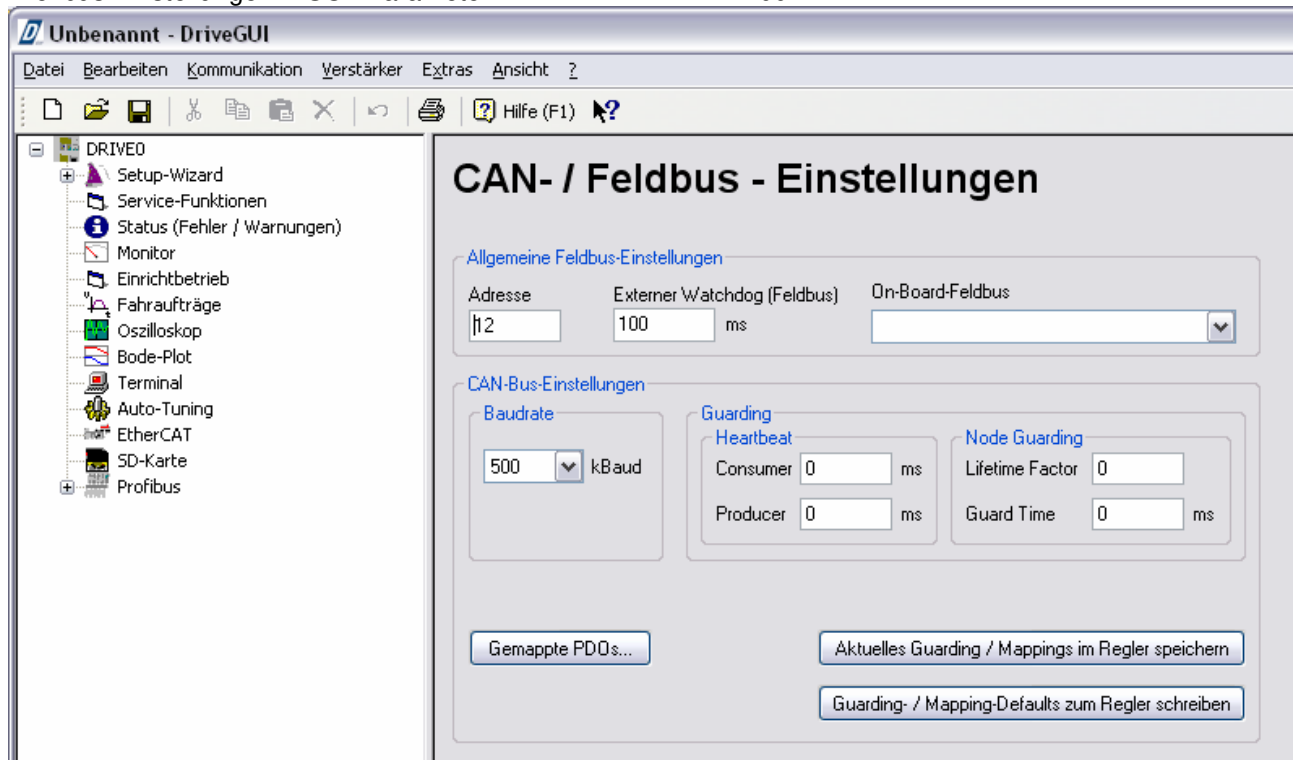
Für die Inbetriebnahme für den Servostar S400 / S600 wird die Software Drive benutzt.
Dies wird hier im Detail nicht näher beschrieben.



Für die Inbetriebnahme für den Servostar S300 / S700 wird die Software DriveGui benutzt.
DriveGui-Version:



Profibus Einstellungen: ASCII Parameter ADDR 12 – EXTWD 100



Profibus – Daten – Eingang / Ausgang oder Send / Receive – Interface PKW und PZD

Unbenannt - DriveGUI

OPMODE: 8: Position Fahrsätze

Profibus

Baudrate: 1500 kBaud
Adresse: 12
PNO-Identnr.: H045D
PPO-Typ: 2

Verstärker: Leistungsteil, Regelung, Profibus-Interface

Profibus-Master: Profibus

Profibus-Interface-Zustände:
Watchdog-Status: Baudr.-Suche, Prüft. d. Baudr., DP Mode, Kommunikation OK
DP-Status: Warten, Param., Warten, Konfig., Datenaustausch

☐ Ansprechüberwachung ignorieren

Ein- / Ausgangsbuffer:

PKW			PZD						
	PKE	IND	PWE	STW	HSW	PZD3	PZD4	PZD5	PZD6
Ausgang:	0000	0000	0000 0000	0034	0000	0000	0000	0000	0000
Eingang:	0000	0000	0000 0000	ZSW	HIW	PZD3	PZD4	PZD5	PZD6
				0270	0000	0000	0000	0000	0000

Zustandsmaschine

Bereit

S706 (DRIVE0) @ COM1 (38,4 kBaud) Online No SW

Profibus – Steuerwort STW and Zustandswort ZSW - Interface

Unbenannt - DriveGUI

OPMODE: 8: Position Fahrsätze

Profibus - Zustandsmaschine

Steuerwort (STW)

0	0	Einschalten
1	0	Spannung sperren
2	1	Schnellh. in Einsch.
3	0	Betrieb freigegeben
4	1	Schnellhalt (HLG sperr.)
5	1	Zwisch. (HLG stop)
6	0	Sollwertfreigabe
7	0	Reset Fehler
8	0	Tippen (Ein/Aus)
9	0	reserviert
10	0	PZD (Freig./Sperren)
11	0	Start Refer. (Flanke)
12	0	herstellerspezifisch
13	0	herstellerspezifisch
14	0	herstellerspezifisch
15	0	herstellerspezifisch

Zustandsmaschine

```

graph TD
    Start -- 0 --> NichtEinschaltbereit[Nicht einschaltbereit]
    NichtEinschaltbereit -- 1 --> Einschaltsperr[Einschaltsperr]
    Einschaltsperr -- 2 --> Einschaltbereit[Einschaltbereit]
    Einschaltbereit -- 3 --> Betriebsbereit[Betriebsbereit]
    Betriebsbereit -- 4 --> BetriebFreigegeben[Betrieb freigegeben]
    BetriebFreigegeben -- 5 --> Schnellhalt[Schnellhalt]
    Schnellhalt -- 6 --> Einschaltbereit
    BetriebFreigegeben -- 7 --> Einschaltsperr
    Einschaltsperr -- 8 --> Einschaltbereit
    Einschaltbereit -- 9 --> Einschaltsperr
    Einschaltsperr -- 10 --> Einschaltbereit
    Einschaltsperr -- 11 --> Schnellhalt
    Schnellhalt -- 12 --> Einschaltsperr
    Schnellhalt -- 13 --> Einschaltsperr
    Schnellhalt -- 14 --> Einschaltsperr
    Schnellhalt -- 15 --> Einschaltsperr
    Schnellhalt -- 16 --> Einschaltsperr
    Fehlerreaktionaktiv[Fehlerreaktion aktiv] -- 14 --> Fehler[Fehler]
    Fehler -- 15 --> Einschaltsperr
    Fehler -- 16 --> Einschaltsperr
    
```

Zustandswort (ZSW)

0	0	Einschaltbereit
1	0	Eingeschaltet
2	0	Betrieb freigegeben
3	0	Fehler
4	1	Spannung gesperrt
5	1	Schnellhalt
6	1	Einschaltsperr
7	0	Warnung
8	0	SollHstwertüberwach.
9	1	Remote
10	0	Sollwert erreicht
11	0	Begrenzung aktiv
12	0	Abh. v. d. Betriebsart
13	0	Abh. v. d. Betriebsart
14	0	herstellerspezifisch
15	0	herstellerspezifisch

Bereit

S706 (DRIVE0) @ COM1 (38,4 kBaud) Online No SW

Bausteinübersicht

Axis_01_FB	FB14/DB14	Hantierungsbaustein zwischen S7-PLC und Servostar
Axis_01_Write	FB16/DB16	Baustein für den FB14 zum Daten schreiben in den Servostar
Axis_01_WriteData	DB20	Enthält die Daten, die vom FB16 in den Servostar geschrieben werden
Axis_01_Read	FB17/DB17	Baustein für den FB14 zum Daten lesen aus dem Servostar
Axis_01_ReadData	DB21	Enthält die Daten, die vom FB17 aus dem Servostar gelesen werden
Axis_01_Compare	FB18/DB18	Baustein zum Datenvergleich vom z. Bsp. DB20 und DB21
IF_DB	DB70	Interface-Datenbaustein z. Bsp. für ein Operator Panel /Touch Panel

Alle Bausteine können bei Bedarf umbenannt bzw. umnummeriert werden.

Anschließend muss „Bausteinkonsistenz prüfen“ mit „Alles übersetzen“ durchgeführt werden.

Daraus resultierende Konflikte müssen in den Bausteinen mit „Zugriffe prüfen und aktualisieren“ oder gegebenenfalls von Hand in den Bausteinen korrigiert werden.

Der FB14 nutzt die folgenden S7-SPS Systembausteine:

SFC12 – D_ACT_DP – Deaktivieren und Aktivieren von DP Slaves / ProfiNet IO devices

SFC14 – DPRD_DAT – Konsistente Daten lesen eines DP Slaves / ProfiNet IO devices

SFC15 – DPWR_DAT – Konsistente Daten schreiben eines DP Slaves / ProfiNet IO devices

SFC20 – BLKMOV – Speicherbereich kopieren

SFC21 – FILL – Speicherbereich vorbesetzen

Im OB1 werden folgende S7 Systembausteine benutzt:

SFC64 – TIME_TCK – Systemzeit lesen (damit werden im Programm die Timer realisiert)

und im OB100 NW1 und OB1 NW1: High, Low, Cycle, Zero, GenSysTime - MW0 and MD10

Anmerkungen

- Der FB14 ist als Multi-Instanz programmiert und kann somit in einem FB mehrmals als Unterprogramm-Baustein ohne eigenen Instanz-Datenbaustein aufgerufen werden.
- Wenn der FB14 nicht als Multi-Instanz implementiert wird, dann muss der FB14 für jeden Antrieb mit einem eigenen Instanz-DatenBaustein aufgerufen werden.
- Der FB14 ist in AWL und FUP (Funktionsplan) mit der Mnemonik DEUTSCH und der Symbolik mit Kommentaren in ENGLISCH programmiert.
- Der Datentyp UDT (UserDefinedDatatype) ermöglicht eine einheitliche Datenbasis und eine objektorientierte Programmierung in Step7.
- Der Datentyp STRUCT ermöglicht ganze Datenbereiche einfach mittels Pointer (P#DB.DBX Byte) zu adressieren.
- Das S7-Projekt „Sv14_v3c“ enthält ein komplett funktionsfähiges S7-SPS Programm für den Servostar - Profibus.
- Das S7-Projekt „Sv14_v3c“ kann in das Simatic TIA- Portal migriert werden.
Es gibt für TIA13 - SP1 – und TIA14 – SP1 komplett funktionsfähige Projekte in FUP für die S7-1200 PLC und die S7-1500 PLC.

Mit einer S7-300 / 400 - SPS und einem Servostar mit Profibus - Optionskarte und Motor kann mit der VAT14 (Variablentabelle) die komplette Servoachse schnell und einfach in Betrieb genommen werden.

Eingang- und Ausgangschnittstelle vom FB14 - Axis 01 FB

Die Schnittstelle vom FB14, FB16, FB17, FB18 und DB20, DB21, DB70 verwendet UDT – (UserDefinedDataType - Anwenderdefinierte Datentypen).

UDT 141 - M_Axis_01_Request – Request
 UDT 143 - M_Axis_01_MaMsg – Machine messages / Error messages
 UDT 145 - M_Axis_01_State
 UDT 146 - M_Axis_01_InData
 UDT 147 - M_Axis_01_OutData
 UDT 148 - M_Axis_01_Pnu
 UDT 149 - M_Axis_01_RcvSend - Receive and Send Profibus

Die Verwendung der UDT stellt sicher dass die Datenstruktur in dem S7-Projekt eine einheitliche gemeinsame Datenbasis hat. Eine Änderung an der gesamten Datenstruktur erfolgt einmalig im UDT. Die gesamte Datenstruktur vom S7-Projekt wird dann mit „Bausteinkonsistenz prüfen“ und „Alles übersetzen“ automatisch aktualisiert. Zeitstempelkonflikte werden ebenfalls beseitigt.

Bitte danach prüfen dass die Anfangswerte von den PLC-Datentypen in die Aktualwerte von dem DB transferiert wurden. Wenn nicht dann ist für diesen DB eine separate Initialisierung erforderlich:
 Den DB mit dem KOP/AWL/FUP-Editor öffnen und anschliessend wechseln zur Datensicht
 - BEARBEITEN \ Datenbaustein initialisieren.

FB14 - Axis 01 FB

Eingangsvariablen haben ein „i“ (Input) und Ausgangsvariablen ein „o“ (Output) vorangestellt. Damit ist der Programmcode leichter lesbar.

VAR_INPUT		
iData	UDT146	
iRequest	UDT141	
iPnu	Struct	
Write	UDT148	
Read	UDT148	
...		
VAR_OUTPUT		
oMaMsg	UDT143	
oState	UDT145	
oData	UDT147	
oAxis	UDT149	
...		

DB 70 - IF DB

ToMachine	UDT27	
FromMachine	UDT28	
Request	UDT141	
State	UDT145	
...		
iData	UDT146	
oData	UDT147	
oMaMsg	UDT143	
...		

GenLow, GenHigh, GenStartUp und GenSysTime

Diese Merker werden generiert siehe OB100 – NW1 und siehe OB1 – NW1

OB100 : "Complete Restart"

Comment:

Network 1: reset

```

L      L#0
T      MW      0
T      "GenSysTime"          MD10          -- system time at beginning of OB1

```

OB1 : Program Cycle Organization Block

Comment:

Network 1: High, Low, Cycle, Zero, GenSysTime

```

U      "GenLow"              M1.0          -- static low signal
R      "GenLow"              M1.0          -- static low signal

UN     "GenHigh"             M1.1          -- static high signal
S      "GenHigh"             M1.1          -- static high signal

U      "GenHigh"             M1.1          -- static high signal
FP     "GenHfStartUp"         M1.5          -- helpflag for startup cycle after PLC startup
=      "GenStartUp"          M1.2          -- startup cycle after PLC startup

CALL   "TIME_TCK"            SFC64         -- Read the System Time
RET_VAL:="GenSysTime"        MD10         -- system time at beginning of OB1

```

Adressierung von dem Hardware-Interface an den FB14 - Axis_01_FB

Die Eingangs- / Ausgangsadresse und Diagnoseadresse wird in der Hardwarekonfiguration eingestellt. Siehe Kapitel Step7-Hardwarekonfiguration. Jeder Servostar benötigt die Zuordnung von seiner Gerätekonfiguration an den FB14 - Axis_01_FB

Ist die Zuordnung korrekt durchgeführt kommunizieren die S7-SPS und der Servostar über den Profibus und der FB14 - Axis_01_FB meldet oState.CommunicationOk =1

OB1 - GenCycleOB

Network 12: Init parameters

LAddrIn - StartAddressIn in HW-Config periphery 256dez
 LAddrOut - StartAddressOut in HW-Config periphery 256dez
 DiagAddr - DiagnosticAddress in HW-Configuration 1012dez = 3F4hex

// overwrite !!!

```

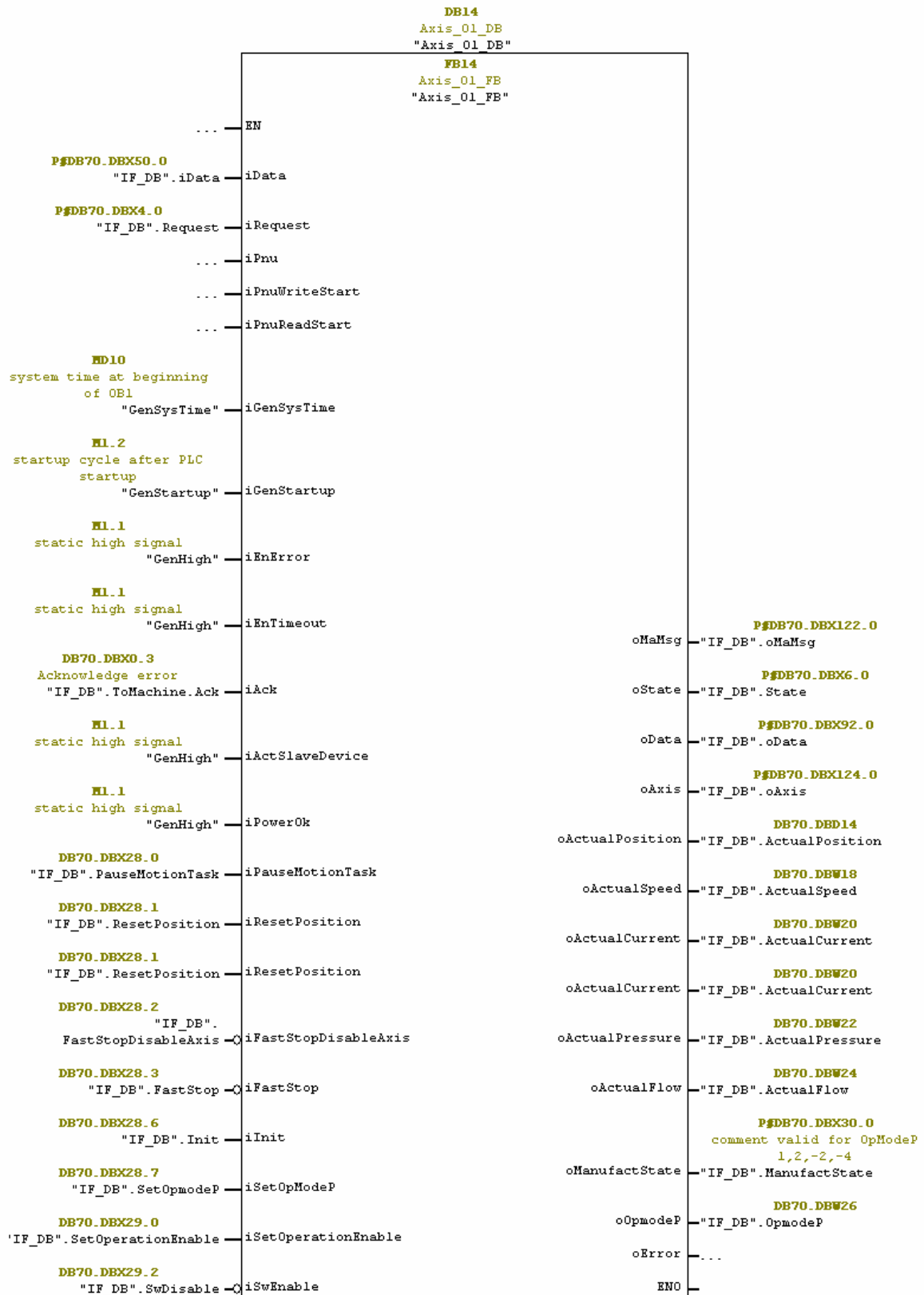
L      256
T      "IF_DE".iData.Config.LAddrIn    DE70.DEW50          -- StartAddressIn in HW-Config periphery dez
T      "IF_DE".iData.Config.LAddrOut    DE70.DEW52          -- StartAddressOut in HW-Config periphery dez

L      W#16#3F4
T      "IF_DE".iData.Config.DiagAddr    DE70.DEW54          -- DiagnosticAddress in HW-Config periphery hex

```

OB1 - GenCycleOB

□ Network 13 : Axis_FB



Gesamtübersicht der Schnittstelle vom FB14 - Axis 01 FB

VAR_INPUT		
iData	STRUCT	Data: HWConfig, OpMode, RefJogSpeed, MotionTask, PNU
Config	STRUCT	
LAddrIn	WORD	StartAddressIn in HW-Config periphery dez
LAddrOut	WORD	StartAddressOut in HW-Config periphery dez
DiagAddr	WORD	DiagnosticAddress in HW-Config periphery hex
TO_Reference	Time	Timeout referencing
TO_Position	Time	Timeout positioning
OpModeP	DWORD	PNU 930 (2 =MotionTask, 1 =VelocityDigital, -2 =TorqueDigital ...)
JogSpeed	WORD	PNU 1889
RefSpeed	WORD	PNU 1896
MotionTask	STRUCT	
Number	WORD	0 =DirectMotionTask
DirectSpeed	DWORD	PNU 1791
DirectPosition	DWORD	PNU 1790
DirectType	WORD	PNU 1785
DigitalSpeed	STRUCT	
Ncmd	WORD	PNU1886 - Ncommand (Ncmd16 = Ncmd * 2 ¹⁵ / VOSPD)
DigitalTorque	STRUCT	
Icmd	WORD	PNU1870 - Icommand (I[A] = Icmd * IpeakAmplifier[A] / 3280)
DigitalPump	STRUCT	
QPRcmd	WORD	QPR command (pressure 1 <-> 10 mbar)
QFcmd	WORD	QF command (flow 1 <-> 0,1 l/min)
iRequest	STRUCT	Requests: Ref Pos StartStopCancelMotionTask Jog
StartRef	BOOL	Start reference movement, static 1
StartIcmd	BOOL	Start I command digital torque, P4, static 1
StartMotionTask	BOOL	Start motion task (direct motion task =0), P4
StopMotionTask	BOOL	Stop motion task, P4->P3
CancelMotionTask	BOOL	Cancel motion task, P3
StartNcmd	BOOL	Start N command digital speed, P4, static 1
JogPlus	BOOL	Jog positive, static 1
JogMinus	BOOL	Jog negative, static 1
MacroInput	BOOL	MacroProgramInput, static 1, PROSTW & 0x200
StartQcmd	BOOL	Start Q command digital speed, P4, static 1
Res_1_2	BOOL	
Res_1_3	BOOL	
Res_1_4	BOOL	
Res_1_5	BOOL	
Res_1_6	BOOL	
Res_1_7	BOOL	
iPnu	STRUCT	
Write	STRUCT	
Number	WORD	
Index	WORD	
Value	DWORD	
Read	STRUCT	
Number	WORD	
Index	WORD	
Value	DWORD	
iPnuWriteStart	BOOL	Request PnuWriteStart pulse 0->1
iPnuReadStart	BOOL	Request PnuReadStart static 1
iGenSysTime	TIME	System time at beginning of OB1
iGenStartup	BOOL	Startup cycle after PLC startup
iEnError	BOOL	Enable error messages
iEnTimeout	BOOL	Enable timeout movement referencing and positioning
iAck	BOOL	Acknowledge WarningsErrors
iActSlaveDevice	BOOL	Activate Slave or Device
iPowerOk	BOOL	All powersupplies are ok
iPauseMotionTask	BOOL	Pause for Motion Task, P4
iResetPosition	BOOL	Reset position, set ActualPosition to RefPosition (ROFFS)
iFastStopDisableAxis	BOOL	FastStop with disable axis, P4->P1
iFastStop	BOOL	FastStop without disable axis, P4->P11
iInit	BOOL	Initialize axis with disable axis
iSetOpModeP	BOOL	Set operating mode Profibus
iSetOperationEnable	BOOL	Set axis state machine to P4_OperationEnabled
iSwEnable	BOOL	Software enable axis

VAR_OUTPUT		
oMaMsg	STRUCT	Error messages
ErrTO_Ref	BOOL	Error timeout reference
ErrTO_Pos	BOOL	Error timeout positioning
ErrNoReferenceSet	BOOL	Error if reference is not set and request
ErrWrongOpMode	BOOL	Error wrong operation mode selected and request
ErrActSlaveDevice	BOOL	Error activating slave or device
ErrCfgInput	BOOL	Error configuration input
ErrRcv	BOOL	Error receiving data
ErrCfgOutput	BOOL	Error configuration output
ErrSend	BOOL	Error sending data
ErrAxis	BOOL	Error from axis
ErrReadWrite	BOOL	Error request Read and Write together
ErrNotEnabled	BOOL	Error if not enabled and request
ErrRes_1_4	BOOL	
ErrRes_1_5	BOOL	
ErrRes_1_6	BOOL	
ErrCmd	BOOL	Error more than one request command active
oState	STRUCT	AxisState
ReferencingActive	BOOL	Axis is referencing
JoggingActive	BOOL	Axis is jogging
VelocityIsZero	BOOL	Axis velocity is zero
MotionTaskActive	BOOL	Axis motion task is active
InPosition	BOOL	Axis is in position
Res_0_5	BOOL	
Res_0_6	BOOL	
Res_0_7	BOOL	
ReferenceOk	BOOL	Axis is referenced
CommunicationOk	BOOL	Axis communication Profibus is ok
InitOK	BOOL	Axis initialisation is ok
InitError	BOOL	Axis initialisation error
OpModeP_Ok	BOOL	Axis opmode Profibus is ok
OpModeP_Error	BOOL	Axis opmode Profibus error
WarningActive	BOOL	Axis warning active
WarnPositionError	BOOL	Axis warning position error
P0_NotReadySwitchOn	BOOL	state diagram
P1_SwitchOnInhibited	BOOL	state diagram
P2_ReadyForSwitchOn	BOOL	state diagram
P3_ReadyForOperation	BOOL	state diagram
P4_OperationEnabled	BOOL	state diagram
P11_FastStopActive	BOOL	state diagram
P13_ErrorReaction	BOOL	state diagram
P14_ErrorActive	BOOL	state diagram
PnuWriteOk	BOOL	Pnu write done and ok
PnuWriteError	BOOL	Pnu write not done and error
PnuReadOk	BOOL	Pnu read done and ok
PnuReadError	BOOL	Pnu read not done and error
Res_3_4	BOOL	
Res_3_5	BOOL	
Res_3_6	BOOL	
Res_3_7	BOOL	
ResponseTelegram_PKW_PWE	DWORD	Axis response telegram after Pnu Rcv or Send
oData	STRUCT	AxisData
Canceled	STRUCT	
DirectMotionTask	STRUCT	
STW	WORD	
Speed	DWORD	
Position	DWORD	
TaskType	WORD	
ActualSpeed	INT	
ActualPosition	DINT	
StartPosition	DINT	
PnuRead	STRUCT	
Number	WORD	
Index	WORD	
Value	DWORD	

oAxis	STRUCT	
Rcv	STRUCT	
PKW		
PKE	WORD	
IND	WORD	
PWE1	WORD	
PWE2	WORD	
PZD	STRUCT	
ZSW	STRUCT	
SetpointActualValMonitor	BOOL	only in OpMode POSITION: Following error
Remote	BOOL	not working, set to 1
SetpointReached	BOOL	only in OpMode POSITION: At Position
LimitActive	BOOL	at the moment not working
ModeDependentx	BOOL	used in ASCII-Modus
ModeDependenty	BOOL	used in ASCII-Modus
ModeDependentz	BOOL	used in ASCII-Modus
Reserved	BOOL	reserved
ReadyForSwitchOn	BOOL	
SwitchedOn	BOOL	
OperationEnabled	BOOL	
Error	BOOL	see ASCII-Kommando ERRCODE
VoltageInhibit	BOOL	
FastStop	BOOL	
SwitchOnInhibit	BOOL	
Warning	BOOL	see ASCII-Kommando STATCODE
HIW	WORD	
PZD3	WORD	
PZD4	WORD	
PZD5	WORD	
PZD6	WORD	
Send	STRUCT	
PKW	STRUCT	
PKE	WORD	
IND	WORD	
PWE1	WORD	
PWE2	WORD	
PZD	STRUCT	
STW	STRUCT	
JoggingOnOff	BOOL	OpMode dependent
Reserved	BOOL	
PZDenableInhibit	BOOL	
StartHomingRun	BOOL	OpMode dependent
ResetPosition	BOOL	
AckWarning	BOOL	Acknowledge warnings
MoTaskDirectOrMoTaskNr	BOOL	Only in OpModes Position: 0=MotionTaskNumber 1=DirectMotionTask
DigitalRevolutionSpeed	BOOL	OpMode dependent, digital velocity
SwitchOn	BOOL	
InhibitVoltage	BOOL	
FastStopSwitchOn	BOOL	1>0Axis FastStopWithEmgyRamp, AxisWillDisabled-STOPMODEECDIS
OperationEnabled	BOOL	
FastStopWithEmgyRamp	BOOL	1>0 Axis fast stop with emergency ramp
PauseStopRfg	BOOL	OpMode dependent, 1>0 Axis stop
SetpointEnable	BOOL	OpMode dependent
ResetFault	BOOL	Reset errors
HSW	WORD	
PZD3	WORD	
PZD4	WORD	
PZD5	WORD	
PZD6	WORD	

oActualPosition	DINT	Actual position, valid if PZD channel is active 1*)
oActualSpeed	INT	Actual speed, valid if PZD channel is active, OpmodePb 1,2
oActualCurrent	INT	Actual current, valid if PZD channel is active, OpmodePb -2
oActualPressure	INT	Actual pressure, valid if PZD channel is active, OpmodeP -7
oActualFlow	INT	Actual flow, valid if PZD channel is active, OpmodeP -7
oManufactState	STRUCT	comment valid for OpMode positioning
Pos3reached	BOOL	
Pos4reached	BOOL	
AxisInternalInitReady	BOOL	
x0_3	BOOL	
VelocityIsZero	BOOL	
SafetyRelayOpen	BOOL	
AxisEnabled	BOOL	
AxisErrorExist	BOOL	
MotionTaskActive	BOOL	
ReferenceDoneAndOK	BOOL	
ReferenceSwitchOn	BOOL	
InPositionWindow	BOOL	
LatchPositionDone	BOOL	
x1_5	BOOL	
Pos1reached	BOOL	
Pos2reached	BOOL	
oOpmodeP	WORD	Actual OpmodeP ProfibusDP (Pnu 930)
oError	BOOL	Error is active

1*) Die aktuelle Positon wird in den Prozessdaten nur in reglerinternen Einheiten (2^{20} Inkr. pro Motorumdrehung) zur S7-SPS übertragen. Die Umrechnung von den reglerinternen Einheiten in Benutzereinheiten kann in der S7-SPS erfolgen und ist abhängig von der mit der Bediensoftware im Regler eingestellten Positonsauflösung PRBASE.

Beispiel mit PRBASE 20:

Auflösung = 5000 Inkr / 3 Umdrehungen:

=> Position in Benutzereinheiten = Actual_Position x 5000 / (3 x 2^{20})

Programmbeispiele zur Umrechnung der reglerinternen Einheiten in Benutzer/SI-Einheiten

OB1 - Netzwerk 17: "IF_DB".ActualPosition (see Pnu 1800: SI-Unit)

// example: LinearAxis - feed 160000 µm/revolution with gear ratio: i=7

```
// "IF_DB".ActualPosition [µm, SI-Unit - DWORD]
// = "Axis_01_DB".oActualPosition [Counts] * (PGEAR1 / PGEAR0) / 2^PRBASE
// = "Axis_01_DB".oActualPosition [Counts] * (160000/7) / 1048576
// = "Axis_01_DB".oActualPosition [Counts] * 0.021798270089
```

// use datatype 32-bit IEEE floating-point number

```
L  "Axis_01_DB".oActualPosition
DTR
L  2.179827e-002
*R
RND
T  "IF_DB".ActualPosition
```

OB1 – Netzwerk 18: "IF_DB".ActualSpeed (see Pnu 1815: SI-Unit)

// example: VOSPD 3600 rpm

```
// "IF_DB".ActualSpeed [rpm, SI-Unit - Word]
// = "Axis_01_DB".oActualSpeed [Counts] * VOSPD / 2^15
// = "Axis_01_DB".oActualSpeed [Counts] * 3600 / 32768
// = "Axis_01_DB".oActualSpeed [Counts] * 225 / 2048
```

// use datatype 32-bit integer

```
L  "Axis_01_DB".oActualSpeed
L  225
*I
L  2048
/D
T  "IF_DB".ActualSpeed
```

OB1 – Netzwerk 19: "IF_DB".ActualCurrent (see Pnu 1688: SI-Unit)

// example for Servo Amplifier with Ipeak 9 Ampere

```
// "IF_DB".ActualCurrent [ma, SI-Unit - Word]
// = "Axis_01_DB".oActualCurrent [Counts] * DIPEAK / 3280
// = "Axis_01_DB".oActualCurrent [Counts] * 9000 / 3280
```

// use datatype 32-bit integer

```
L  "Axis_01_DB".oActualCurrent
L  9000
*I
L  3280
/D
T  "IF_DB".ActualCurrent
```

OB1 – Network 20: "IF_DB".ActualPressure

// "IF_DB".ActualPressure [xx * 10 mBar, SI-Unit – Word]

L "Axis_01_DB".oActualPressure
T "IF_DB".ActualPressure

OB1 – Network 21: "IF_DB".ActualFlow

// "IF_DB".ActualFlow [xx * 0.1 l/min, SI-Unit - Word]

L "Axis_01_DB".oActualFlow
T "IF_DB".ActualFlow

Inbetriebnahme FB14 - Axis 01 FB

S7-SPS - AUS/EIN-Schalten

Fällt der S7-SPS Profibus-Master aus dann startet im Servostar der Watchdog-Timeout (EXTWD) und generiert die Warnung „n04 – Ansprechüberwachung aktiv (watch dog)“.

Die S7-SPS macht nach dem Anlauf einen Software-Reset (GenStartUp - M1.2) und löscht alle Warnungen und Fehlermeldungen und Zustände im SPS-Programm, aber nicht im Servostar.

Bootet der Servostar schneller als die S7-SPS dann wird im Servostar die Warnung n04 generiert.

Mit ASCII-Parameter S1DLY kann die Boot-Zeit verlängert werden um z.Bsp. die n04 Warnung zu vermeiden. Erst nach Quittieren dieser Warnung kann der Servostar in den Betrieb gehen.

S7-SPS - Run/Stop-Schalter

Im Zustandswechsel von STOP -> RUN macht die S7-SPS einen Software-Reset (GenStartUp - M1.2) und löscht alle Warnungen und Fehlermeldungen und Zustände im SPS-Programm, aber nicht im Servostar.

iAck =1

Setzt die Warnungen und Fehler zurück im Servostar und im S7-SPS Programm in den Bausteinen FB14 - Axis_01_FB, FB16 - Axis_01_Write, FB17 - Axis_01_Read, FB18 - Axis_01_Compare.

Initialisierung

Bei der Initialisierung sendet die S7-SPS ein Null-Telegramm (Send.PKW und Send.PZD) an den Servostar. Der Servostar wird sofort disabled und die Zustandsmaschine wechselt in oState.P1_SwitchOnInhibited =1

- Setze iInit =1 (intern wird ein Impuls generiert)
- Warte bis oState.InitOk =1
Wenn oState.InitOk nicht =1 oder oState.InitError =1 dann war die Initialisierung nicht erfolgreich.
Fehlerursache siehe Receive-Telegramm Servostar.
- Die Initialisierung wird mit einem Timeout von 1 sec überwacht.

Betriebsart setzen

Programmierte Betriebsarten: Positionieren, Drehzahl digital, Drehmoment digital, Elektrisches Getriebe, Servopumpe digital

Nach 24VDC EINschalten ist der Servostar immer in der sicheren internen Betriebsart -126 und „gesperrt“. Über Profibus findet eine Kommunikation zwischen der S7-SPS und dem Servostar nur von dem STW und ZSW statt und die RCV - PZD2, PZD3, PZD4, PZD5, PZD6 (receive telegram) enthalten alle den Wert "0". Die S7-SPS muss zuerst die Betriebsart ändern und anschliessend prüfen.

- Schreibe die gewünschte Betriebsart in iData.OpModeP
(siehe Handbuch: Positionieren: 2 ; Drehzahl digital: 1 ; Drehmoment digital: -2 ;
Elektrisches Getriebe: -4 , Servopumpe digital: -7)
- Setze iSetOpModeP =1 (intern wird ein Impuls generiert)
- Warte bis oState.OpModeP_Ok = 1
Die aktivierte Betriebsart wird angezeigt in oOpmodeP
Wenn oState.OpModeP_Ok nicht =1 oder oState.OpModeP_Error =1 oder oOpModeP ist nicht der angeforderte OpModeP dann war Betriebsart setzen nicht erfolgreich.
Fehlerursache siehe Receive-Telegramm Servostar.
- Jetzt werden auch die RCV - PZD2, PZD3, PZD4, PZD5, PZD6 angezeigt abhängig von der Betriebsart mit Werten für z. Bsp. oActualPosition, oActualSpeed, oManufactState.
- Die Betriebsart setzen wird mit einem Timeout von 1 sec überwacht.

Betrieb freigeben

Nach 24VDC EINschalten ist der Servostar in oState.P1_SwitchOnInhibited =1 (Einschaltsperrung). iFastStop und iFastStopDisableAxis müssen =0 und iSwEnable muss =1 sein.

Am Servostar muss HardwareEnable =1 und falls vorhanden muss AS/STO-Enable =1 sein.

Nur S700: Eine eingebaute Safetycard muss im Zustand „RUN“ sein.

Die Zwischenkreisspannung wird angelegt und im Display vom Servostar wird „Pxx“ angezeigt ohne Warnungen oder Fehler.

- Setze iSetOperationEnabled =1 (intern wird ein Impuls generiert)
- Warte bis oState.P4_OperationEnabled =1
Wenn oState.P4_OperationEnabled nicht =1 dann war Betriebsart setzen nicht erfolgreich.

Der Servostar ist nun enabled und kann eine Kraft / Drehmoment generieren und einen Motor bewegen.

Anforderungen für die Betriebsart Positionieren

- Der Betrieb ist freigegeben (oState.P4_OperationEnabled =1)
- AS/STO- und Hardware-Enable ist vorhanden
- Keine Warnungen und Fehler und das Display vom Servostar zeigt „Exx“
- Die Betriebsart Position (oState.OpModeP_Ok = 1 und oOpModeP = 2 dec) ist aktiviert

Tippbetrieb

- Schreibe die gewünschte Tippgeschwindigkeit in „iData.Config.JogSpeed“
- Setze iRequest.JogPlus =1 -> Der Antrieb bewegt sich positiv und oState.JoggingActive =1
- Setze iRequest.JogMinus =1 -> Der Antrieb bewegt sich negativ und oState.JoggingActive =1

Referenzfahrt (Homing)

Der Referenzfahrttyp wird normalerweise mit der DriveGui eingestellt und im Servostar gespeichert.

- Schreibe die gewünschte Referenzfahrtgeschwindigkeit in „iData.Config.RefSpeed“
- Setze iRequest.StartRef =1 -> Der Antrieb bewegt sich und oState.ReferenceActive =1
- Warte bis oState.ReferenceOk =1 und oState.ReferenceActive wieder =0
- Setze iStartRef = 0 – Der Antrieb ist referenziert

Mit iEnTimeout =1 und iData.Config.TO_Reference xxx ms [TIME] wird die Referenzfahrt mit einem Timeout überwacht.

Start eines EEPROM oder RAM Fahrauftrags (MotionTask)

Zusätzliche Voraussetzung: Der Servostar ist referenziert.

Mit der DriveGui werden Fahraufträge angelegt und im Servostar EEPROM gespeichert.

Nur die Parameter der RAM-Fahraufträge können mit der S7-SPS im Servostar sogar im Zustand “P4_OperationEnabled” und während der Motor sich bewegt geändert werden (siehe ASCII - MTMUX).

- Schreibe die Nummer des Fahrauftrags in „iData.MotionTask.Number
- Setze iStartMotionTask =1 (intern wird ein Impuls generiert)
- Warte bis oState.InPosition =0 und oState.MotionTaskActive =1
- Warte bis oState.InPosition =1 und oState.MotionTaskActive =0

Der Servostar hat den Fahrauftrag ausgeführt. Mit iEnTimeout =1 und iData.Config.TO_Position xxx ms [TIME] wird der Fahrauftrag mit einem Timeout überwacht.

Istposition, Istdrehzahl (16bit) und Hersteller-Status werden dabei zyklisch im RCV.PZD-Kanal übertragen.

Start von dem Direktfahrauftrag (RAM und hat die Nummer 0)

Voraussetzung: Der Servostar ist referenziert.

- Schreibe 0 in iData.MotionTask.Number
- Schreibe Zielposition in iData.MotionTask.DirectPosition
- Schreibe Sollgeschwindigkeit in iData.MotionTask.DirectSpeed
- Schreibe Fahrauftragsart in iData.MotionTask.DirectType
- Setze iRequest.StartMotionTask =1 (intern wird ein Impuls generiert)
- Warte bis oState.InPosition =0 und oState.MotionTaskActive =1
- Warte bis oState.InPosition =1 und oState.MotionTaskActive =0

Der Servostar hat den Fahrauftrag ausgeführt. Mit iEnTimeout =1 und iData.Config.TO_Position xxx ms [TIME] wird der Fahrauftrag mit einem Timeout überwacht.

Istposition, Istdrehzahl (16bit) und Hersteller-Status werden dabei zyklisch im RCV.PZD-Kanal übertragen.

Während der MotionTask läuft kann bereits der nächste MotionTask mit seinem Datensatz an iData.MotionTask kopiert werden und mit Setze iRequest.StartMotionTask =1 (Impuls) dieser dann sofort gestartet werden.

Mit dem SFC20-BLKMOV und Pointer (P#DB.DBX Byte) kann der Data.MotionTask [STRUCT] einfach kopiert werden.

Siehe “Switch ON and Start MotionTask timing diagram”

Ein gestarteter Fahrauftrag kann mit `iPauseMotionTask = 1` angehalten werden.
Der Servostar bleibt in `oState.P4_OperationEnabled = 1`.

Ein gestarteter Fahrauftrag kann mit `iRequest.StopMotionTask = 1` gestoppt werden.
Der Servostar wechselt in `oState.P3_ReadyForOperation = 1`.
Ein gestoppter Fahrauftrag bleibt im Regler weiterhin aktiv, d.h. wenn der Regler wieder in den `oState.P4_OperationEnabled = 1` geschaltet wird, wird der Fahrauftrag fortgesetzt.
Mit `iSwEnable = 0` wird der gestoppte Fahrauftrag gekillt, d.h. wenn der Regler nun wieder in den `oState.P4_OperationEnabled = 1` geschaltet wird, wird der Fahrauftrag nicht mehr fortgesetzt.
Siehe: "Kill MotionTask timing diagram"

In dem OB1 ist ein Beispiel um Teile von einem Fahrauftrag zu parametrieren und mit FB16 WRITE von der S7-SPS in den Servostar mit dem Parameter Kanal zu übertragen.

Hinweis: Mit dem ASCII-Parameter INPT kann im Terminalfenster der DriveGui (oder mit der PNU1904) die Zeit in ms eingestellt werden, für die das Signal `oManufactState.InPositionWindow` nach dem Fahrsatzstart mindestens zurückgesetzt wird (vgl. Handbuch Kap. VII.1).
So ist das Interface Zeitverhalten zwischen der S7-SPS und dem Servostar immer identisch.

Hinweis zu Positionen und Geschwindigkeiten – Fahrauftragsart (MotionTaskType) O C – PNU 1785

0x0000 hex (bit 13 = 0)
Absolute Positionierung mit Vorgabe der Geschwindigkeit und Position in inkrementellen reglerinternen 32 Bit und 16 Bit Einheiten und Fahrprofil Trapez.

0x2000 hex (bit 13 = 1)
Absolute Positionierung mit Vorgabe der Geschwindigkeit und Position in Benutzereinheiten und Fahrprofil Trapez.

0x2003 hex (bits 0, 1, 13 = 1)
Relative Positionierung mit Vorgabe der Geschwindigkeit und Position in Benutzereinheiten und Fahrprofil Trapez.

0x12000 hex (bit 13 = 1 und bit 16 = 1)
Absolute Positionierung mit Vorgabe der Geschwindigkeit und Position in Benutzereinheiten und Fahrprofil Sinus².

0x10000 hex (bit 16)
Das Bit 16 kann nicht direkt im PZD-Kanal gesetzt werden.
Siehe: Aktivieren der Trajektorie/Profil Beschleunigung Sin² bei Direktfahrsatz Nr. 0

Hinweis zu Feedback mit Wake&Shake

Die Istposition, Istgeschwindigkeit und Hersteller-Status werden zyklisch in den RCV.PZD Kanal geschrieben nachdem Betriebsart setzen (`oState.OpModeP_Ok = 1` und `oOpModeP = xxx dec`) und die Wake&Shake Prozedur ausgeführt wurde.

Bemerkung: `n14 = 1` Ermittlung von MPHASE (z.B. FBTYPE=7)
Wird gesetzt beim Einschalten des Verstärkers. Wird gelöscht wenn die Endstufe freigegeben und MPHASE mit Wake&Shake ermittelt wurde.

Falls ein Fehler beim Betriebsart setzen auftritt dann bitte zuerst bei dem Servostar den Betrieb freigegeben (`iSetOperationEnabled = 1`) und damit wird die Wake&Shake Prozedur ausgeführt.
Erst danach die Betriebsart setzen durchführen.

Betriebsart Drehzahl digital

Die Betriebsart Drehzahl digital (oState.OpModeP_Ok = 1 und oOpmodeP = 1 dez) ist aktiviert und der Betrieb ist freigegeben (oState.P4_OperationEnabled =1).

- Schreibe Sollgeschwindigkeit in iData.DigitalSpeed.Ncmd (16bit)
- Setze iRequest.StartNcmd =1
Der Antrieb verfährt mit der vorgegebenen Sollgeschwindigkeit.
Die Sollgeschwindigkeit kann dabei jederzeit verändert werden.
- Setze iRequest.StartNcmd =0
- Der Antrieb brems über die eingestellten Drehzahlrampen bis zum Stillstand ab.

Istposition, Ist Drehzahl (16bit) und Hersteller-Status werden dabei zyklisch im RCV.PZD-Kanal übertragen.

Der Drehzahlwert(16bit) wird dabei nach folgender Formel berechnet:

PNU1886 - Ncommand (Ncmd16 = Ncmd * 2¹⁵ / VOSPD)

Betriebsart Drehmoment digital

Die Betriebsart Drehmoment digital (oState.OpModeP_Ok = 1 und oOpmodeP = -2 dez) ist aktiviert und der Betrieb ist freigegeben (oState.P4_OperationEnabled =1).

- Schreibe Sollstrom in iData.DigitalTorque.lcmd
- Setze iRequest.Startlcmd =1
- Der Antrieb prägt den vorgegebenen Sollstrom ein.
- Der Sollstrom kann dabei jederzeit verändert werden.
- Setze iRequest.Startlcmd =0 und der Antrieb gibt den Sollstrom 0 Ampere aus.

Istposition, Iststrom und Hersteller-Status werden dabei zyklisch im RCV.PZD-Kanal übertragen.

Der Stromwert wird dabei nach folgender Formel berechnet:

PNU1870 - Icommand (I[A] = lcmd * IpeakAmplifier[A] / 3280)

Mit dem ASCII-Parameter ICMDVLIM (PNU 1989) kann die Drehzahl des Motors auf einen Maximalwert begrenzt werden. Damit wird ein Durchgehen des Motors bei zu geringer Belastung verhindert.

Betriebsart Elektrisches Getriebe

Die Betriebsart Elektrisches Getriebe (oState.OpModeP_Ok = 1 und oOpmodeP = -4 dez) ist aktiviert und der Betrieb ist freigegeben (oState.P4_OperationEnabled =1).

Istposition, Iststrom und Hersteller-Status werden dabei zyklisch im RCV.PZD-Kanal übertragen.

Betriebsart Servopumpe digital – Drehzahl

nur für S300/S700 - siehe Applikationsschrift – Servopumpe - s700_servopumpe_d.pdf

Die Betriebsart Servopumpe digital (oState.OpModeP_Ok = 1 und oOpmodeP = -7 dez) ist aktiviert und der Betrieb ist freigegeben.

Hinweis: Beim Aktivieren der Betriebsart -7 wird automatisch QENA auf 1 gesetzt. Beim Umschalten auf eine andere Betriebsart als -7, wird die Servopumpe abgeschaltet (QENA=0).

- Schreibe SollDruck in iData.DigitalPump.QPFRcmd
- ODER
- Schreibe SollVolumenstrom (SollDurchfluss) in iData.DigitalPump.QFcmd
- Setze iRequest.StartQcmd =1
- Der Antrieb prägt den vorgegebenen Sollwert Druck / Volumenstrom ein.
- Der Sollwert Druck / Volumenstrom kann dabei jederzeit verändert werden.
- Setze iRequest.StartQcmd =0 und der Servostar gibt den Sollwert Druck / Volumenstrom 0 aus.

Istdruck, Istdurchfluss, Iststrom und Istposition werden dabei zyklisch im RCV.PZD-Kanal übertragen.

Mit diesen Betriebsarten sind die meisten Anforderungen in einer Maschine realisierbar.
Weitere Betriebsarten sind derzeit nicht im FB14 implementiert, diese können jedoch einfach nachprogrammiert werden.

Makro-Programmierung

Ein Makroprogramm kann im Servostar weitere Funktionen ausführen.

Zur Programmierung wird die Software MacroStar verwendet.

Im Parameterkanal können für den Datenaustausch zwischen der S7-PLC und dem Servostar die ASCII Parameter DPRVARxx und DPVxx für Macro- und PLC-Programme verwendet werden.

z. Bsp. DPRVAR1 - PNU 2022 (dec) IND = 1 (dec)

Im Echtzeitkanal gibt es zusätzlich iRequest.MacroInput

– siehe Axis_01_FB (FB14) – Network 88: #Axis.Send.PZD.STW.MacroInput

Für besondere Anwendungen kann eine Änderung und Erweiterung z.Bsp. vom Axis_01_FB (FB14) erforderlich sein.

Die gesamten SEND und RECEIVE Daten können ebenfalls vom Makroprogramm verwendet werden.

Damit der Compiler fehlerfrei durchläuft muss im MacroStar in der Datei variables.cfg eingetragen sein:

```
PROSTW,2,""  
PROZSW,2,""  
PROFIN0,2,""  
PROFIN1,2,""  
PROFIN2,2,""  
PROFIN3,2,""  
PROFIN4,2,""  
PROFIN5,2,""  
PROFIN6,2,""  
PROFIN7,2,""  
PROFIN8,2,""  
PROFIN9,2,""  
PROFOUT0,2,""  
PROFOUT1,2,""  
PROFOUT2,2,""  
PROFOUT3,2,""  
PROFOUT4,2,""  
PROFOUT5,2,""  
PROFOUT6,2,""  
PROFOUT7,2,""  
PROFOUT8,2,""  
PROFOUT9,2,""
```

Beispiel zur S7-SPS und Makro-Programmierung im S300/S700:

Der Motor soll endlos drehen und anschliessend direkt in einer definierten Position ruckfrei mit weicher \sin^2 -Bremsrampe anhalten. Die Zielposition und VJOG Drehzahl wird entweder von der S7-SPS als DirectMotionTask-Parameter übergeben oder kann auch als DPRVAR1 und DPRVAR2 im S300/S700-EEPROM abgespeichert werden.

Der komplette Bewegungsablauf ist an der Maschine mit dem DriveGui-Oszilloskop zu kontrollieren.
(Position, Schleppfehler, Iststrom, Istdrehzahl)

S300/S700-Parameter:

```
OPMODE 8 ( oOpmodeP = 2 dez )
PGEARI 3600
PGEARO 1
POSCNFG 1 (Axis type MODULO )
DREF 16
SRND 0
ERND 36000
DPRVAR1 18500
DPRVAR2 3000
DPRVAR3 73728
DPRVAR4 50
DPRVAR5 50
```

S7-PLC Program:

```
// Start iAck
// Start iInit
// Start iSetOpModeP =2
// Start iOpEnable
// Start Reference run
// Start DirektMotionTask to TargetPosition with VJOG-Velocity

U      M      200.0
UN     "Axis_01_DB".oMaMsg.ErrCmd
UN     "Axis_01_DB".iSetOperationEnable
UN     "Axis_01_DB".iSetOpModeP
UN     "Axis_01_DB".iInit
U      "Axis_01_DB".oState.P4_OperationEnabled
U      "Axis_01_DB".oState.ReferenceOk
=      "IF_DB".Request.MacroInput
```

Makro program:

```

PROGRAM PLCINIT

LONG INP1:=0;
LONG INP2:=0;
LONG INP3:=0;
LONG VALUE1:= 1048576;
LONG VALUE2:= 1;
LONG VALUE3:= 35999;
LONG RESULT:= 1;
LONG TEMPVAR1:= 0;
LONG TEMPVAR2:= 0;

END_PROGRAM

//*****
PROGRAM PLCMAIN

END_PROGRAM

//*****

PROGRAM PLC250

// Profi-STW - Bit 9 - MacroInput
// Bit 9 0->1: JOGMOVE with VJOG-Velocity: iData.MotionTask.Direct.Speed or DPRVAR2
// Bit 9 1->0: STOP at TargetPosition: iData.MotionTask.Direct.Position or DPRVAR1 < ERND !!!
// O_ACC with DPRVAR4
// O_DEC with DPRVAR5

// O_P:= DPRVAR1; // TargetPosition
// O_V:= DPRVAR2; // VJOG-Velocity
O_C:= O_C | 0x12000; // DPRVAR3; // 73728dez <-> 0x12000hex: Sin^2 and SI-Units
O_ACC:= DPRVAR4;
O_DEC:= DPRVAR5;

IF O_V <> 0 THEN
    TEMPVAR1:= 7179; // GO
ELSE
    TEMPVAR1:= 8888; // XX
END_IF;

IF TEMPVAR1 = 7179 THEN
    TEMPVAR2:= PROSTW&0x200;
    IF TEMPVAR2 = 0x200 THEN
        INP2:= 0;
        INP3:= 0;
        IF INP1 = 0 THEN
            INP1:= 1;
            VJOGIO:=O_V; // Set VJOG speed
            SETPTR(TRJ,G_MOVEJOGIO); //Start moving
        END_IF;
    ELSE
        IF INP1 = 1 THEN
            INP2:= 1;
            VALUE2:= O_P;
            MULDIV(VALUE1,VALUE2,VALUE3,RESULT);
        END_IF;
        INP1:= 0;
        IF INP2 = 1 THEN
            IF INP3 = 0 THEN
                IF PFB < RESULT THEN
                    INP3:= 1;
                END_IF;
            END_IF;
            IF INP3 = 1 THEN
                IF PFB >= RESULT THEN
                    INP2:= 0;
                    INP3:= 0;
                    MOVEP_NR:=0;
                    SETPTR(TRJ,G_STARTMOVE);
                END_IF;
            END_IF;
        END_IF;
    END_IF;
END_IF;

END_PROGRAM

```


Parameter lesen und schreiben:

Alle Parameter und Kommandos des Servostars sind über die Profibus PNU-Nummer ansprechbar. Diese werden in der ASCII Kommandoliste Objektreferenz der Reihe nach gelistet angezeigt. Außerdem findet sich die Nummer bei der Beschreibung der Kommandos/ Parameter im Feld „Profibus PNU“. Zusätzlich befindet sich im Handbuch eine Liste ausgewählter Parameternummern.

ASCII Objektreferenz		ASCII - Kommando		ACC	
Sprachumschaltung nach englisch		Syntax Senden		ACC [Data]	
sortiert nach:		Syntax Empfangen		ACC <Data>	
Kommandos Gruppen Objekt-Id.		Type		Variable rw	
		ASCII Format		Integer32	
		DIM		>> ACCUNIT	
		Bereich		3 ..126000	
		Default		31400	
		Opmode		0, 1	
		Verstärker Status		-	
		ab Firmware		1.0	
		Konfiguration		Nein	
		Funktionsgruppe		velocity loop	
		Kurzbeschreibung		Beschleunigungsrampe Drehzahlregelung	
		Beschreibung			

Parameter schreiben

Voraussetzung: Parameterkanal PKW wird nicht bereits benutzt

- Schreibe Nummer in iPnu.Write.Number
- Schreibe Index in iPnu.Write.Index
- Schreibe Wert in iPnu.Write.Value
- Setze iPnuWriteStart =1 (Impuls)
- Warte bis oState.PnuWriteOK

Wenn oState.PnuWriteOk nicht =1 oder oState.PnuWriteError =1 dann war Parameter schreiben nicht erfolgreich. Fehlerursache siehe Receive-Telegramm Servostar.

Parameter lesen

Voraussetzung: Parameterkanal PKW wird nicht bereits benutzt

- Schreibe Nummer in iPnu.Read.Number
 - Schreibe Index in iPnu.Read.Index
 - Schreibe Wert in iPnu.Read.Value
 - Setze iPnuReadStart =1
 - Warte bis oState.PnuReadOK
- Wenn oState.PnuReadOk nicht =1 oder oState.PnuReadError =1 dann war Parameter lesen nicht erfolgreich. Fehlerursache siehe Receive-Telegramm Servostar.
- Die gelesenen Daten werden ausgegeben in
oData.PnuRead.Number
oData.PnuRead.Index
oData.PnuRead.Value

Damit kann verglichen werden ob die gelesenen Daten auch die angeforderten Daten sind.

Parameter lesen kann ständig aktiviert sein z.Bsp. für einen zu überwachenden Istwert (z.Bsp PNU 1688 Effektivstrom zur Ermittlung des Drehmoments).

Im OB1 - GenCycleOB sind einige Beispiele für verschiedene PNUs.

Hinweis: Gleichzeitige Read und Write Anforderung am FB14 generiert den Fehler MaMsg.ErrReadWrite

Die S7-PLC kann den Servostar parametrieren und diese Daten mit SAVE und COLDSTART ins EEPROM speichern.

Bei einigen Parametern ist nach einer Änderung ebenfalls ein SAVE und anschliessend COLDSTART erforderlich.

Diese Änderungen bitte immer im sicheren Zustand von der Servoachse durchführen.

Disable zuerst den Servostar:

Setze iSwEnable =0

Prüfe dass oState.P1_SwitchOnInhibited =1 (Einschaltsperrung).

Setze den Hardware-Enable Eingang =0

Setze den Hardware-Eingang AS/STO-Enable =0

Starte dann eine Initialisierung (Null-Telegramm)

Setze iInit =1 (intern wird ein Impuls generiert)

Warte bis oState.InitOk =1

Ändere nun die Parameter: WRITE ausführen

Anschliessend SAVE – Kommando - über einen READ ausführen.

Siehe ASCII-Parameter SAVE - PNU 1835

Hinweis: Die maximal erlaubte Anzahl der SAVE Schreibzyklen in ein EEPROM ist begrenzt !!!

Warte 5 Sekunden – Die Parameter werden nun in das Servostar-EEPROM gespeichert.

COLDSTART – Kommando - über einen READ ausführen.

Der Servostar bootet und macht einen Neustart / Wiederanlauf.

Die S7-PLC meldet dabei einen „Slave-Servostar – Kommunikationsfehler“

Warte bis der Servostar wieder im Profibus läuft: oState.CommunicationOK =1

Setze nun die Warnungen und Fehler zurück im Servostar und im S7-PLC Programm in den Bausteinen

iAck =1 (Acknowledge)

Nun mit READ und COMPARE überprüfen dass der Servostar die Werte in das EEPROM korrekt übernommen hat.

Im Servostar muss nun wieder die Betriebsart setzen und der Betrieb freigegeben durchgeführt werden.

Zusätzliche Profibus-Funktionen

Der DP-Slave Servostar wird im FB14 mit dem SFC12 – D_ACT_DP bei laufender SPS aktiviert bzw. mit deaktiviert.

- iActSlave =1, Servostar ist aktiviert und tauscht Daten über den Profibus aus.
 - iActSlave =0, Servostar ist deaktiviert und der Datenaustausch ist beendet. Die Ansprechüberwachung (watchdog) startet im Servostar (EXTWD) und generiert die Warning „n04 – Ansprechüberwachung aktiv (watch dog)“.
- Der Servostar kann nun vom Profibus getrennt werden ohne eine Fehlermeldung an der S7-SPS.

Zudem wird die Profibuskommunikation mit SFC14 - DPRD_DAT and SFC15 - DPWR_DAT überwacht auf

- Konfigurationsfehler
- Slaveausfall
- Telegrammfehler

Die gesamten SEND und RCV-Datentelegramme werden im oAxis [STRUCT] ausgegeben

Die Servostar-Zustandsmaschine wird im oState [STRUCT] ausgegeben

Bei relativer Positionierung können bei einem Abbruch eines Fahrauftrags die Daten in oData.Canceled [STRUCT] weggespeichert werden. Zu einem späteren Zeitpunkt (z. Bsp. nach NOTAUS – manueller Eingriff - TürAUF / TürZU) kann dann die S7-SPS damit die Fahrsatz-Daten korrigieren und den relativen Fahrauftrag mit korrigierten Daten zu Ende fahren.

VAT14 Variablentabelle

Mit der Variablentabelle VAT14 können die Bausteine im Servostar angesteuert und beobachtet werden.

VAT14 -- Sv14_V3cS7 Program					
	Address	Symbol	Display format	Status value	Modify value
1	DB70.DBX 0.3	"IF_DB".ToMachine.Ack	BOOL		
2	DB70.DBX 28.6	"IF_DB".Init	BOOL		
3	DB70.DBX 28.7	"IF_DB".SetOpModeP	BOOL		
4	DB70.DBX 29.0	"IF_DB".SetOperationEnable	BOOL		
5	DB70.DBX 4.0	"IF_DB".Request.StartRef	BOOL		
6	DB70.DBX 4.1	"IF_DB".Request.StartIcmd	BOOL		
7	DB70.DBX 4.2	"IF_DB".Request.StartMotionTask	BOOL		
8	DB70.DBX 4.3	"IF_DB".Request.StopMotionTask	BOOL		
9	DB70.DBX 4.4	"IF_DB".Request.CancelMotionTask	BOOL		
10	DB70.DBX 4.5	"IF_DB".Request.StartNcmd	BOOL		
11	DB70.DBX 4.6	"IF_DB".Request.JogPlus	BOOL		
12	DB70.DBX 4.7	"IF_DB".Request.JogMinus	BOOL		
13	DB70.DBX 5.0	"IF_DB".Request.MacroInput	BOOL		
14	DB70.DBX 5.1	"IF_DB".Request.StartQcmd	BOOL		
15	DB70.DBX 28.0	"IF_DB".PauseMotionTask	BOOL		
16	DB70.DBX 28.2	"IF_DB".FastStopDisableAxis	BOOL		
17	DB70.DBX 28.3	"IF_DB".FastStop	BOOL		
18	DB70.DBX 29.2	"IF_DB".SwDisable	BOOL		
19					
20	DB70.DBD 64	"IF_DB".iData.Config.OpModeP	DEC		L#2
21	DB70.DBW 68	"IF_DB".iData.Config.JogSpeed	DEC		100
22	DB70.DBW 70	"IF_DB".iData.Config.RefSpeed	DEC		200
23	DB70.DBW 72	"IF_DB".iData.MotionTask.Number	DEC		0
24	DB70.DBD 74	"IF_DB".iData.MotionTask.DirectSpeed	DEC		L#100
25	DB70.DBD 78	"IF_DB".iData.MotionTask.DirectPosition	DEC		L#1234
26	DB70.DBW 82	"IF_DB".iData.MotionTask.DirectType	BIN		2#0010_0000_0000_0000
27	DB70.DBW 84	"IF_DB".iData.DigitalSpeed.Ncmd	DEC		547
28	DB70.DBW 86	"IF_DB".iData.DigitalTorque.Icmd	DEC		200
29	DB70.DBW 88	"IF_DB".iData.DigitalPump.GPRcmd	DEC		43
30	DB70.DBW 90	"IF_DB".iData.DigitalPump.GFcmd	DEC		12
31					
32	DB70.DBX 2.2	"IF_DB".FromMachine.ErrAl	BOOL		
33	DB70.DBX 2.3	"IF_DB".FromMachine.Warning	BOOL		
34					

Beispiel für eine Schrittkette „Initialisierung und Enablen Servostar“

```

U      "Axis_01_DB".oState.CommunicationOk
L      S5T#1S
SE     T      100
U      T      100
FP     M      100.0
S      M      100.1
R      M      100.2
R      M      100.3
R      M      100.4
R      M      100.5
R      M      100.6
R      M      100.7

U      M      100.1
=      "IF_DB".ToMachine.Ack

U      M      100.1
L      S5T#1S
SE     T      101
U      T      101
S      M      100.2
R      M      100.1

U      M      100.2
UN     "Axis_01_DB".oError
L      S5T#1S
SE     T      102
U      T      102
S      M      100.3
R      M      100.2

// -----

U      M      100.3
UN     "Axis_01_DB".oError
UN     "Axis_01_DB".oState.InitOk
=      "IF_DB".Init

U      M      100.3
UN     "Axis_01_DB".oError
U      "Axis_01_DB".oState.InitOk
UN     "Axis_01_DB".oState.OpModeP_Ok
=      "IF_DB".SetOpmodeP

U      M      100.3
UN     "Axis_01_DB".oError
U      "Axis_01_DB".oState.InitOk
U      "Axis_01_DB".oState.OpModeP_Ok
UN     "Axis_01_DB".oState.P4_OperationEnabled
=      "IF_DB".SetOperationEnable

U      M      100.3
UN     "Axis_01_DB".oError
U      "Axis_01_DB".oState.InitOk
U      "Axis_01_DB".oState.OpModeP_Ok
U      "Axis_01_DB".oState.P4_OperationEnabled
UN     "Axis_01_DB".oState.ReferenceOk
=      "IF_DB".Request.StartRef

U      M      100.3
UN     "Axis_01_DB".oError
U      "Axis_01_DB".oState.InitOk
U      "Axis_01_DB".oState.OpModeP_Ok
U      "Axis_01_DB".oState.P4_OperationEnabled
U      "Axis_01_DB".oState.ReferenceOk
R      M      100.3

```

Siehe "Switch ON and Start MotionTask timing diagram"

Beispiel für eine Schrittkette „Starte Fahrsätze im Servostar“

```
// IF Ready then Write Data from PLC to Servostar with pulse from M110.1 and FB16
// and then Start RAM_MotionTask 201,210 and 211
// remark: the RAM_MotionTask 201 has a following RAM_MotionTask 202
```

```
U      "Axis_01_DB".oState.CommunicationOk
UN     "Axis_01_DB".oError
U      "Axis_01_DB".oState.InitOk
U      "Axis_01_DB".oState.OpModeP_Ok
U      "Axis_01_DB".oState.P4_OperationEnabled
U      "Axis_01_DB".oState.ReferenceOk
L      S5T#1S
SE     T      110
U      T      110
FP     M      110.0
S      M      110.1
R      M      110.2
R      M      110.3
R      M      110.4
R      M      110.5
R      M      110.6
R      M      110.7

U      M      110.1
L      S5T#2S
SE     T      111
U      T      111
S      M      110.2
R      M      110.1

U      M      110.2
UN     "Axis_01_DB".oState.MotionTaskActive
L      S5T#2S
SE     T      112
U      T      112
S      M      110.3
R      M      110.2

U      M      110.3
L      S5T#2S
SE     T      113
U      T      113
S      M      110.4
R      M      110.3

U      M      110.4
UN     "Axis_01_DB".oState.MotionTaskActive
L      S5T#2S
SE     T      114
U      T      114
S      M      110.5
R      M      110.4

U      M      110.5
L      S5T#2S
SE     T      115
U      T      115
S      M      110.6
R      M      110.5

U      M      110.6
UN     "Axis_01_DB".oState.MotionTaskActive

L      S5T#2S
SE     T      116
U      T      116
S      M      110.7
R      M      110.6

U      M      110.7
L      S5T#2S
SE     T      117
U      T      117
R      M      110.7
```

```
// -----  
  
      U      M      110.3  
      SPBN   m201  
      L      201  
      T      "IF_DB".iData.MotionTask.Number  
m201: NOP   0  
  
      U      M      110.5  
      SPBN   m210  
      L      210  
      T      "IF_DB".iData.MotionTask.Number  
m210: NOP   0  
  
      U      M      110.7  
      SPBN   m211  
      L      211  
      T      "IF_DB".iData.MotionTask.Number  
m211: NOP   0  
  
      CLR  
      O      M      110.3  
      O      M      110.5  
      O      M      110.7  
      =      "IF_DB".Request.StartMotionTask
```

Siehe "Switch ON and Start MotionTask timing diagram"

Example for the “RAM MotionTask 201 in the Servostar“

```
// ORDER 201 - RAM MotionTask

// PNU 1947 Index 1 - MTMUX

// PNU 1790 Index 1 - O_P target position/path for the motion task
// PNU 1791 Index 1 - O_V target speed/velocity
// PNU 1785 Index 1 - O_C type of motion task (control word)
// PNU 1783 Index 1 - O_ACC acceleration ramp /starting acceleration
// PNU 1786 Index 1 - O_DEC braking ramp / deceleration
// PNU 1784 90Index 1 - O_TAB number of the lookup table
// PNU 1788 Index 1 - O_FN number of following motion tasks
// PNU 1789 Index 1 - O_FT delay before starting next motion task


//MTMUX (PNU 1947) address with RamMotionTask 201

    L    1947
    T    "Axis_01_WriteDataDB".Data[50].Number

    L    1
    T    "Axis_01_WriteDataDB".Data[50].Index

    L    201
    T    "Axis_01_WriteDataDB".Data[50].Value


// PNU 1790 Index 1 - O_P target position/path for the motion task

    L    1790
    T    "Axis_01_WriteDataDB".Data[51].Number

    L    1
    T    "Axis_01_WriteDataDB".Data[51].Index

    L    4.234567e+003
    RND
    T    "Axis_01_WriteDataDB".Data[51].Value


// PNU 1791 Index 1 - O_V target speed/velocity

    L    1791
    T    "Axis_01_WriteDataDB".Data[52].Number

    L    1
    T    "Axis_01_WriteDataDB".Data[52].Index

    L    987.654e+003
    RND
    T    "Axis_01_WriteDataDB".Data[52].Value


// PNU 1785 Index 1 - O_C type of motion task (control word)

    L    1785
    T    "Axis_01_WriteDataDB".Data[53].Number

    L    1
    T    "Axis_01_WriteDataDB".Data[53].Index

    L    L#73728
    T    "Axis_01_WriteDataDB".Data[53].Value
```

```
// PNU 1783 Index 1 - O_ACC acceleration ramp /starting acceleration
```

```

L      1783
T      "Axis_01_WriteDataDB".Data[54].Number

L      1
T      "Axis_01_WriteDataDB".Data[54].Index

L      1.000000e+002
RND
T      "Axis_01_WriteDataDB".Data[54].Value
```

```
// PNU 1786 Index 1 - O_DEC braking ramp / deceleration
```

```

L      1786
T      "Axis_01_WriteDataDB".Data[55].Number

L      1
T      "Axis_01_WriteDataDB".Data[55].Index

L      5.000000e+001
RND
T      "Axis_01_WriteDataDB".Data[55].Value
```

```
// PNU 1784 Index 1 - O_TAB number of the lookup table
```

```

L      1784
T      "Axis_01_WriteDataDB".Data[56].Number

L      1
T      "Axis_01_WriteDataDB".Data[56].Index

L      0
T      "Axis_01_WriteDataDB".Data[56].Value
```

```
// PNU 1788 Index 1 - O_FN number of following motion tasks
```

```

L      1788
T      "Axis_01_WriteDataDB".Data[57].Number

L      1
T      "Axis_01_WriteDataDB".Data[57].Index

L      0
T      "Axis_01_WriteDataDB".Data[57].Value
```

```
// PNU 1789 Index 1 - O_FT delay before starting next motion task
```

```

L      1789
T      "Axis_01_WriteDataDB".Data[58].Number

L      1
T      "Axis_01_WriteDataDB".Data[58].Index

L      0
T      "Axis_01_WriteDataDB".Data[58].Value
```

Hinweis für das MTMUX – Beispiel OB1:

With DriveGui maybe there are no values visible in the MotionTask table No. 201, 210,211
Then make a recheck with the DriveGui terminal:

```

--> ORDER 201
201 -400 1000 -1 -1 -1 -1 -1 -1 400 -1
--> ORDER 210
210 -200 1500 -1 100 50 -1 -1 -1 -1 -1
--> ORDER 211
211 -500 850 -1 150 250 -1 -1 -1 -1 -1
-->
```


Bausteine für Servostar-Parameter

Das S7-Projekt „Sv14_v3c“ enthält noch weitere Funktionsbausteine und Datenbausteine die ganze Datenbereiche in den Servostar schreiben, aus dem Servostar lesen und in der S7-SPS vergleichen.

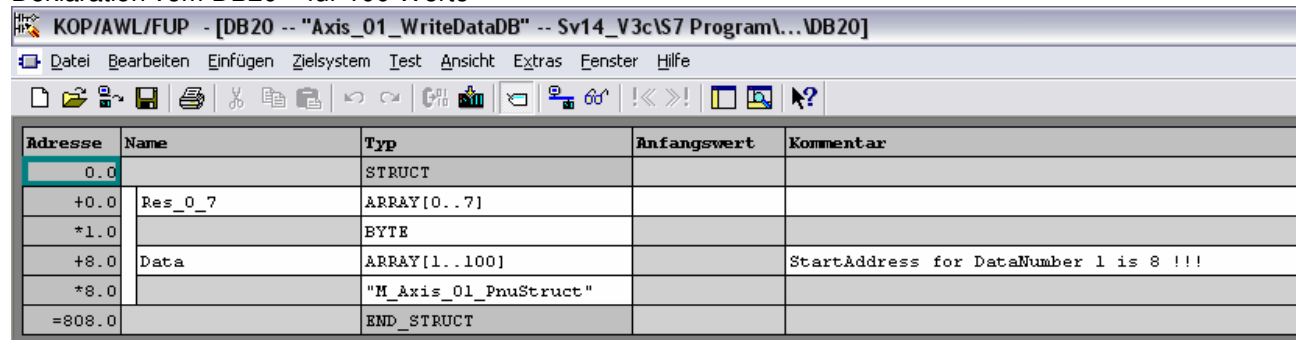
Ein Datensatz besteht aus 3 Parametern:

- PNU - Parameternummer – gibt an um welchen Parameter es sich handelt
z.Bsp PNU 1783 Anfahrtzeit O_ACC1.
- Index gibt an um was es sich bei dem übertragenen Wert handelt
z.Bsp Index=1 - Istwert oder Index=3 oberer Grenzwert.
- Value – enthält den übertragenen Wert.

DB 20 - Axis_01_WriteDataDB

enthält die Daten zum schreiben mit dem FB16 Axis_01_Write und dem FB14 Axis_01 von der S7-SPS über den nicht echtzeitfähigen PKW-Parameterkanal im Profibus in den Servostar.

Deklaration vom DB20 – für 100 Werte

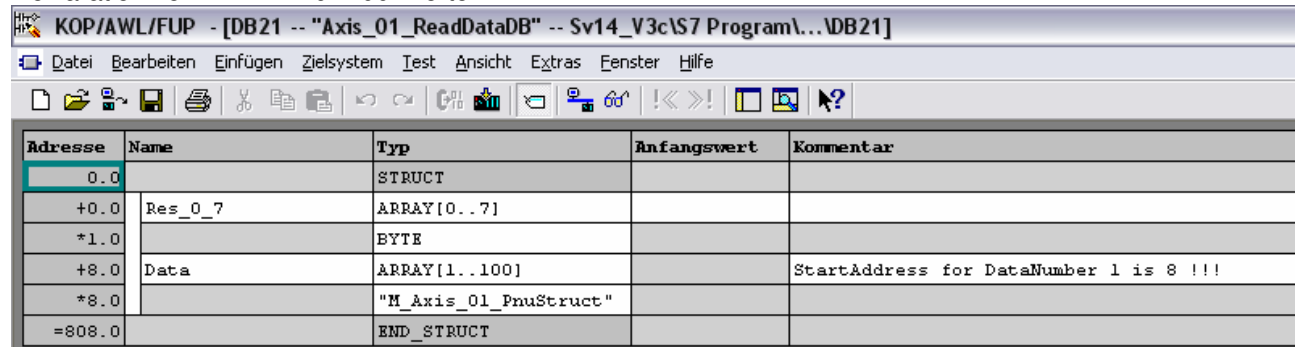


Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	Res_0_7	ARRAY[0..7]		
*1.0		BYTE		
+8.0	Data	ARRAY[1..100]		StartAddress for DataNumber 1 is 8 !!!
*8.0		"M_Axis_01_PnuStruct"		
=808.0		END_STRUCT		

DB 21 - Axis_01_ReadDataDB

enthält die Daten zum lesen mit dem FB17 Axis_01_Read und dem FB14 Axis_01 von dem Servostar über den nicht echtzeitfähigen PKW-Parameterkanal im Profibus in die S7-SPS.

Deklaration vom DB21 – für 100 Werte



Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	Res_0_7	ARRAY[0..7]		
*1.0		BYTE		
+8.0	Data	ARRAY[1..100]		StartAddress for DataNumber 1 is 8 !!!
*8.0		"M_Axis_01_PnuStruct"		
=808.0		END_STRUCT		

Siehe OB1 und VAT14

FB16 Axis_01_Write

steuert den FB14 an zum Daten schreiben von der S7-SPS in den Servostar

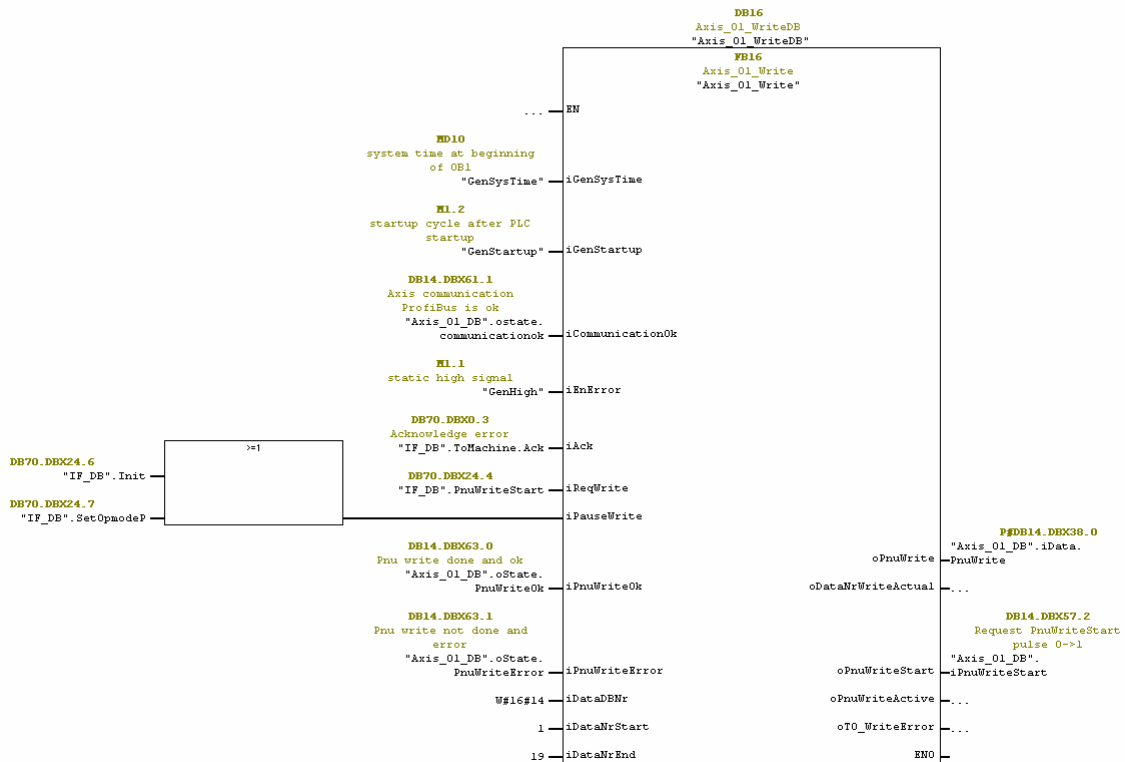
Eingang- und Ausgangschnittstelle des FB16 - Axis_01_Write

VAR_INPUT		
iGenSysTime	TIME	System time at beginning of OB1
iGenStartup	BOOL	Startup cycle after PLC startup
iCommunicationOk	BOOL	Axis communication Profibus is OK
iEnError	BOOL	Enable error messages
iAck	BOOL	Acknowledge WarningsErrors
iReqWrite	BOOL	RequestWrite
iPauseWrite	BOOL	PauseWrite (necessary for SetOpmode or InitAxis)
iPnuWriteOk	BOOL	PnuWriteOk =1 succesful
iPnuWriteError	BOOL	PnuWriteError =1 not succesful
iDataDBNr	WORD	WriteDataDBNumber
iDataNrStart	INT	DataNumberStart - first number from data to write
iDataNrEnd	INT	DataNumberEnd - last number from data to write

VAR_OUTPUT		
oPnuWrite	STRUCT	PnuWrite to Axis_FB actual with Number, Index, Value
Number	WORD	
Index	WORD	
Value	DWORD	
oDataNrWriteActual	INT	Data number is writing actual
oPnuWriteStart	BOOL	Request write start for Axis_FB
oPnuWriteActive	BOOL	Writing DataBlock is active
oTO_WriteError	BOOL	Timeout writing is active but not working

Netzwerk 5: Axis_Write

iDataDBNr: DB20 -> W#16#14
from Data[1] until Data[8]



FB17 Axis_01_Read

steuert den FB14 an zum Daten lesen vom Servostar in die S7-SPS

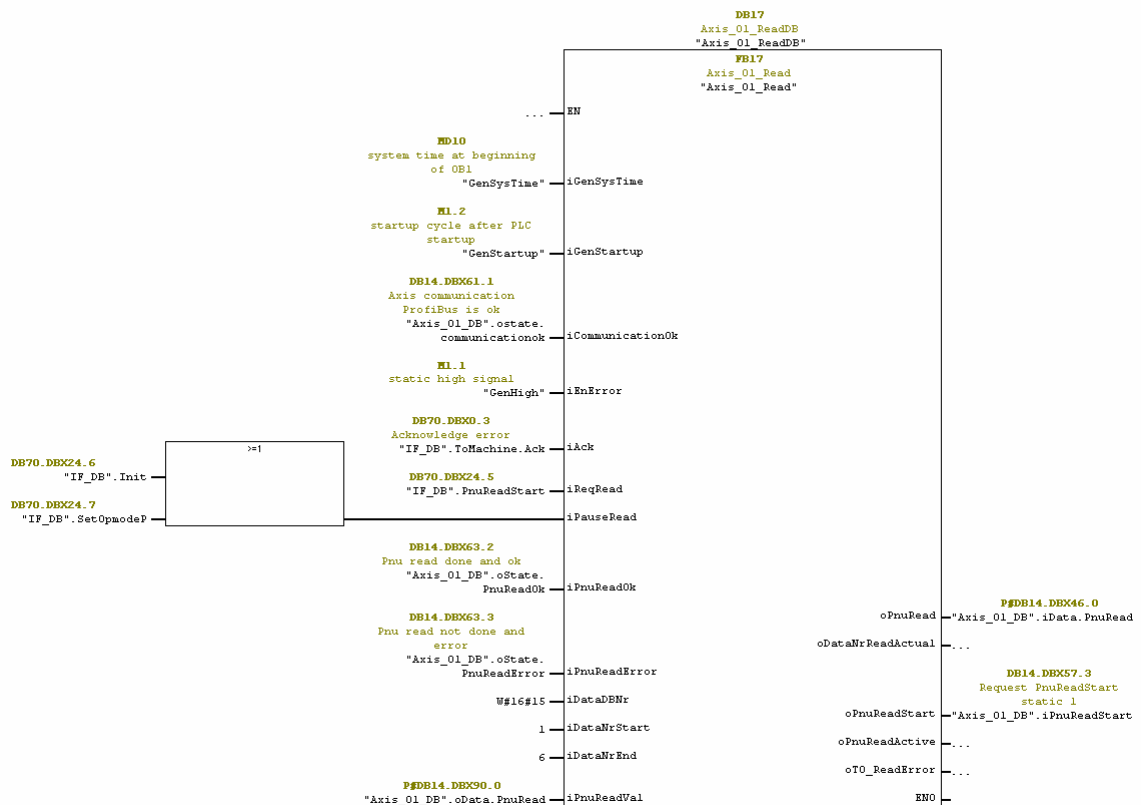
Eingang- und Ausgangschnittstelle des FB17 - Axis_01_Read

VAR_INPUT		
iGenSysTime	TIME	System time at beginning of OB1
iGenStartup	BOOL	Startup cycle after PLC startup
iCommunicationOk	BOOL	Axis communication Profibus is OK
iEnError	BOOL	Enable error messages
iAck	BOOL	Acknowledge WarningsErrors
iRequestRead	BOOL	RequestRead
iPauseWrite	BOOL	PauseRead (necessary for SetOpMode or InitAxis)
iPnuWriteOk	BOOL	PnuReadOk =1 succesful
iPnuWriteError	BOOL	PnuReadError =1 not succesful
iPnuReadVal	STRUCT	PnuRead from Axis_FB actual with Number, Index, Value
Number	WORD	
Index	WORD	
Value	DWORD	
iDataDBNr	WORD	ReadDataDBNumber
iDataNrStart	INT	DataNumberStart - first number from data to read
iDataNrEnd	INT	DataNumberEnd - last number from data to read

VAR_OUTPUT		
oPnuRead	STRUCT	PnuRead from Axis_FB actual with Number, Index
Number	WORD	
Index	WORD	
oDataNrReadActual	INT	Data number is reading actual
oPnuReadStart	BOOL	Request read start for Axis_FB
oPnuReadActive	BOOL	Reading DataBlock is active
oTO_ReadError	BOOL	Timeout reading is active but not working

□ Netzwerk 8: Axis_Read

iDataDBNr: DB21 -> W#16#15
from Data[1] until Data[6]



Hinweis: Gleichzeitige Anforderung Read und Write am Axis_01_FB (FB14) generiert den Fehler oMaMsg.ErrReadWrite

FB18 Axis_01_Compare

vergleicht einen Datenbereich von den geschriebenen Daten vom DB20 Axis_01_WriteDataDB mit den gelesenen Daten vom DB21 Axis_01_ReadDataDB.

Nur eine PNU [STRUCT] wird in einem S7-SPS Zyklus verglichen.

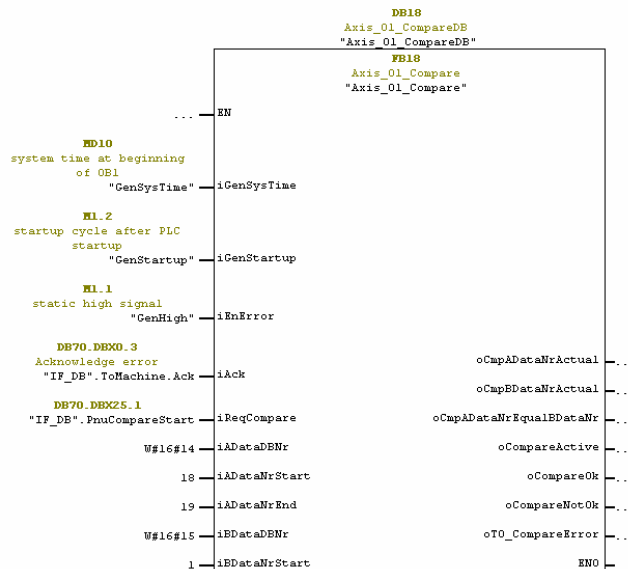
Eingang- und Ausgangschnittstelle des FB18 Axis_01_CompareDB

VAR_INPUT		
iGenSysTime	TIME	System time at beginning of OB1
iGenStartup	BOOL	Startup cycle after PLC startup
iEnError	BOOL	Enable error messages
iAck	BOOL	Acknowledge WarningsErrors
iReqCompare	BOOL	Request compare
iADaDBNr	WORD	ADaDBNumber
iADaNrStart	INT	ADaNumberStart - first number from data to compare
iADaNrEnd	INT	ADaNumberEnd - last number from data to compare
iBdaDBNr	WORD	BdaDBNumber
iBdaNrStart	INT	BdaNumberStart - first number from data to compare

VAR_OUTPUT		
oCmpADaNrActual	INT	Compare A Data number is actual
oCmpBdaNrActual	INT	Compare B Data number is actual
oCmpADaNrEqualBdaNr	BOOL	Compare A Data number is equal B Data number
oCompareActive	BOOL	comparing DataBlocks is active
oCompareOk	BOOL	Compare is ok, datas are equal
oCompareNotOk	BOOL	Compare is not ok, datas are not equal
oTO_CompareError	BOOL	Timeout compare is active but not working

Netzwerk 10: Axis_Compare

```
iADaDBNr: DB20 -> W#16#14 from Data[18] until Data[19]
iBdaDBNr: DB21 -> W#16#15 from Data[1] until Data[19-18+1=2]
```



Mit der Variablen-tabelle VAT14 können die Bausteine READ, WRITE und COMPARE angesteuert und beobachtet werden.

Tipps und Infos:

Fehlermeldungen von den Bausteinen FB14, FB16, FB17 und FB18

sind in das Programm zu implementieren damit das SPS-Programm darauf reagieren kann.

oMaMsg	STRUCT	Error messages
ErrTO_Ref	BOOL	Error timeout reference
ErrTO_Pos	BOOL	Error timeout positioning
ErrNoReferenceSet	BOOL	Error if reference is not set and request
ErrWrongOpMode	BOOL	Error wrong operation mode selected and request
ErrActSlave	BOOL	Error activating slave
ErrCfgInput	BOOL	Error configuration input
ErrRcv	BOOL	Error receiving data
ErrCfgOutput	BOOL	Error configuration output
ErrSend	BOOL	Error sending data
ErrAxis	BOOL	Error from axis
ErrReadWrite	BOOL	Error request Read and Write together
ErrNotEnabled	BOOL	Error if not enabled and request
ErrRes_1_4	BOOL	
ErrRes_1_5	BOOL	
ErrRes_1_6	BOOL	
ErrCmd	BOOL	Error more than one request command active
oTO_WriteError	BOOL	Timeout writing is active but not working
oTO_ReadError	BOOL	Timeout reading is active but not working
oTO_CompareError	BOOL	Timeout reading is active but not working

Override über Profibus

Siehe ASCII-Liste OVERRIDE

Mit Override-Funktion kann die Geschwindigkeit eines Fahrsatzes, der Referenzfahrt und des Tippbetriebes beeinflusst werden.

OVRIDE=0 Override-Funktion abgeschaltet

OVRIDE=3 Profibus Schnittstelle für die digitale Override-Funktion aktiviert.

Siehe ASCII-Liste DOVRIDE:

Beim Aktivieren einer digitalen OVERRIDE-Funktion wird mit Hilfe dieses Parameters der digitale Override-Faktor vorgegeben. Dabei gilt folgende Normierung:

DOVRIDE=0 Fahrsatzgeschwindigkeit = 0 %

DOVRIDE=8192 Fahrsatzgeschwindigkeit = 100 %

Die OVERRIDE – Funktion ist nicht bei \sin^2 – Rampen möglich !

Aktivieren der Trajektorie/Profil Beschleunigung \sin^2 bei Direktfahrsatz Nr. 0

Das Fahrprofil \sin^2 ermöglicht ein weiches, ruckfreies Anfahren und Abbremsen. Damit wird die Mechanik geschont (Zahnräder, Spindeln). Pendelbewegungen etc. werden unterdrückt.

Bei Servostar S400/S600 siehe ASCII-Liste, dies wird hier nicht näher beschrieben.

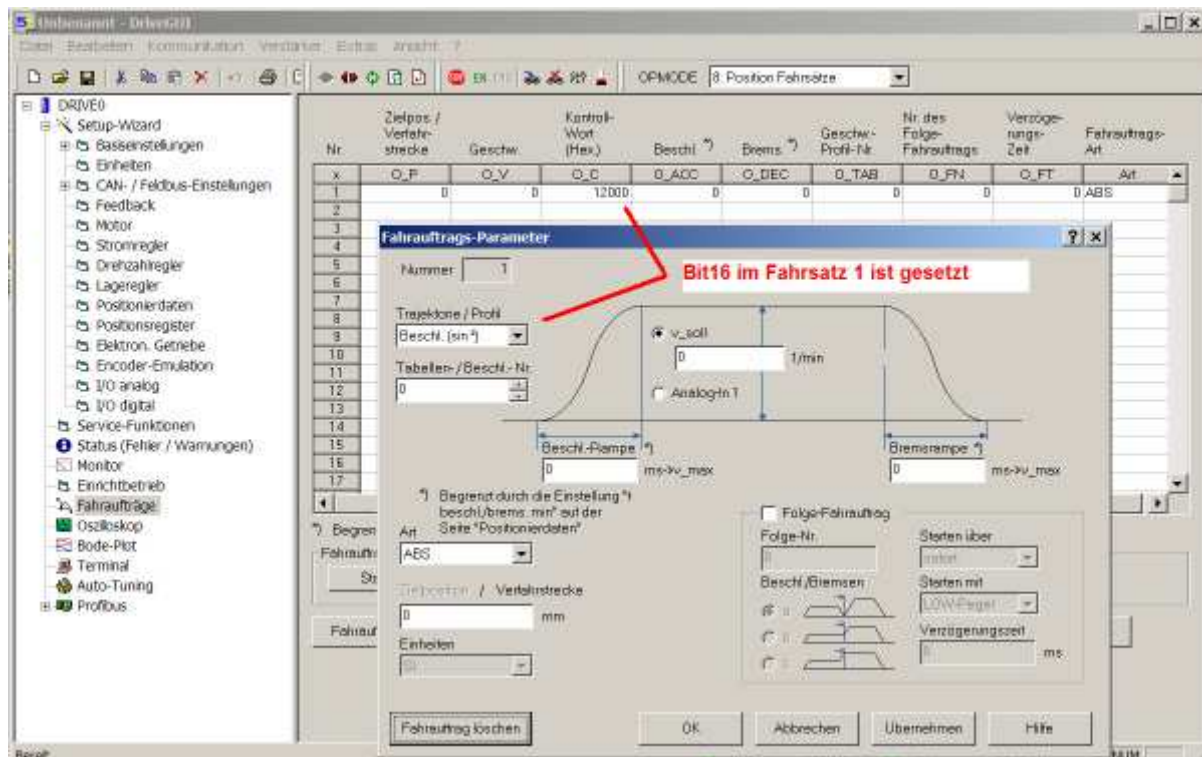
Bei Servostar S300/S700 – Profibus siehe ASCII-Liste O_C

Das Kommando O_C (Profibus PNU 1785) definiert die Fahrauftragsart für den Direktfahrsatz Nr. 0. Ist das Bit 16 von O_C gesetzt, dann hat der Fahrsatz das \sin^2 -Profil. Das Bit 9 muss auf 0 gesetzt werden. Von dem O_C werden nur die Bits 0 bis 15 direkt im DirectMotionTaskType (PZD5) adressiert, deshalb muss das Bit 16 anderweitig im Servostar S300/S700 gesetzt werden.

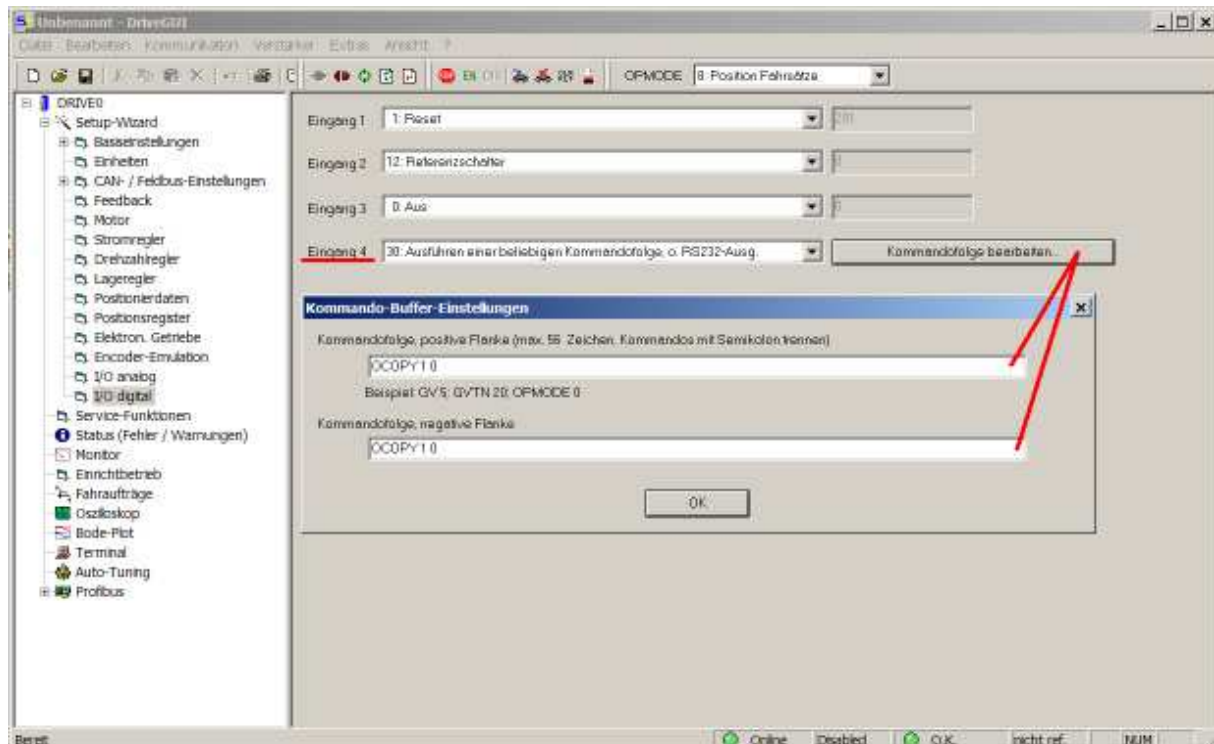
Das Kommando O_C ist kein Parameter und kann deshalb nicht im Servostar gespeichert werden.

Das Bit16 kann auf diese Weise gesetzt werden:

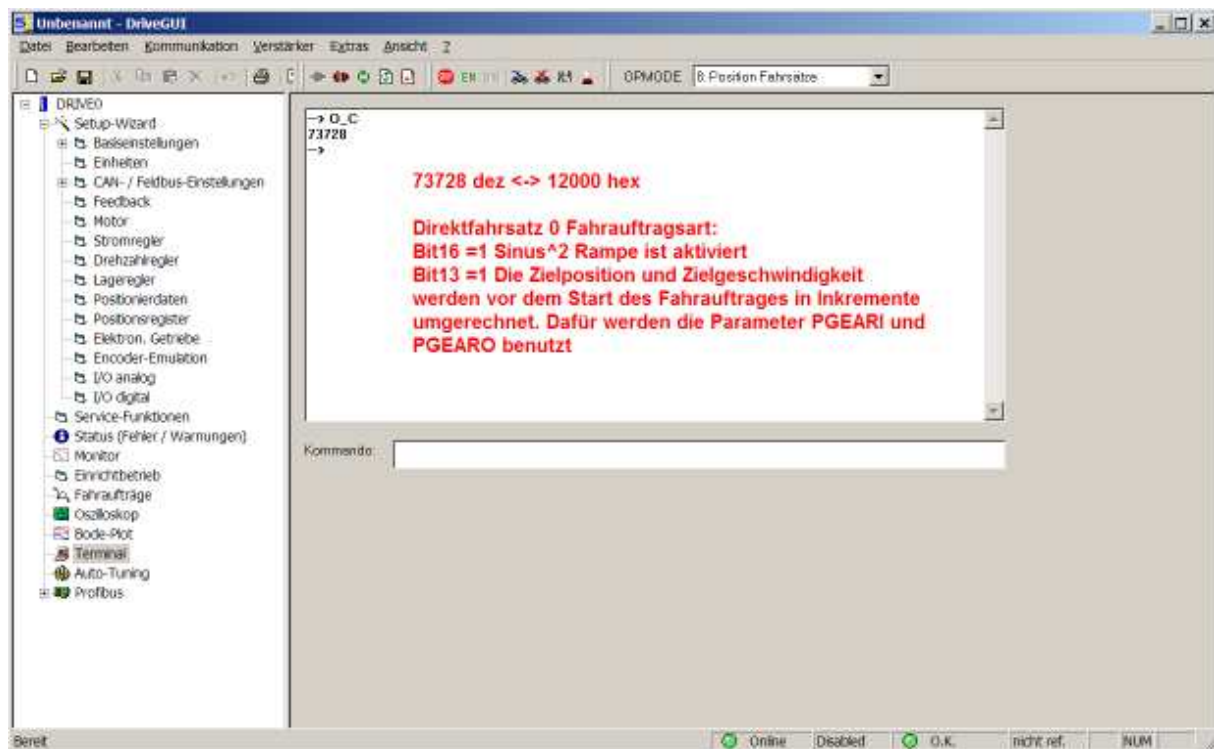
- 1.) Schreibe über den Parameterkanal PKW in die PNU 1785 den Wert 10000hex in den S300/S700
- 2.) Nach dem EINSchalten der 24VDC – Versorgungsspannung kopiert sich der S300/S700 automatisch das Bit16 von einem EEPROM-Fahrsatz z.Bsp. Nr.1 in den Direktfahrsatz Nr. 0. Dadurch bleibt das S7-Programm unverändert und PNU 1785 Parameterdownload ist nicht erforderlich. Zusätzlich können auch die Beschleunigungs- O_ACC und Bremsrampe O_DEC für eine weiche Bewegung eingestellt werden. Bei Wert 0 gilt der Wert PTMIN bzw. jeweils die grösste Rampe.



Einem nicht benötigten Eingang (nicht verdrahtet) wird die OCOPY-Funktion von EEPROM Fahrsatz Nr.1 in den Direktfahrsatz Nr.0 zugewiesen.



SAVE und COLDSTART und Überprüfung mit DriveGui -Terminal den Inhalt von O_C



3.) ASCII Kommandofolge

Mit dem Kommando ASCII INxHCMD / INxLCMD kann eine ASCII-Befehlsfolge definiert werden. Diese Befehlsfolge wird immer dann ausgeführt, wenn eine steigende Flanke auf dem, mit der Funktion INxMODE=30,33 konfigurierten Eingang erkannt wird. Eine Befehlsfolge besteht aus einzelnen ASCII-Kommandos, die mit einem Semicolon (;) getrennt sind. Die maximale Länge dieser Befehlsfolge beträgt 56 Zeichen.

Mit DriveGui-Terminal:

```
IN4MODE 30
IN4HCMD O_C 73728; O_ACC 100; O_DEC 200
IN4LCMD O_C 73728; O_ACC 100; O_DEC 200
SAVE
COLDSTART
```

Überprüfung mit DriveGui-Terminal:

```
ORDER 0
0 0 0 73728 100 200 0 0 0 0
```

4.) Macro program

Siehe macro program Beispiel:

```
O_C:= O_C | 0x12000; // DPRVAR3; // 73728dez <-> 0x12000hex: Sin^2 and SI-Units
```

Überprüfung mit with DriveGui-Terminal:

```
ORDER 0
0 0 0 73728 0 0 0 0 0 0
```

Hinweis: Bei der Überprüfung mit dem DriveGui-Terminal kann auch dieses Ergebnis mit derselben Funktionalität angezeigt werden:

```
ORDER 0
0 -1 -1 73728 -1 -1 -1 -1 -1 -1
```

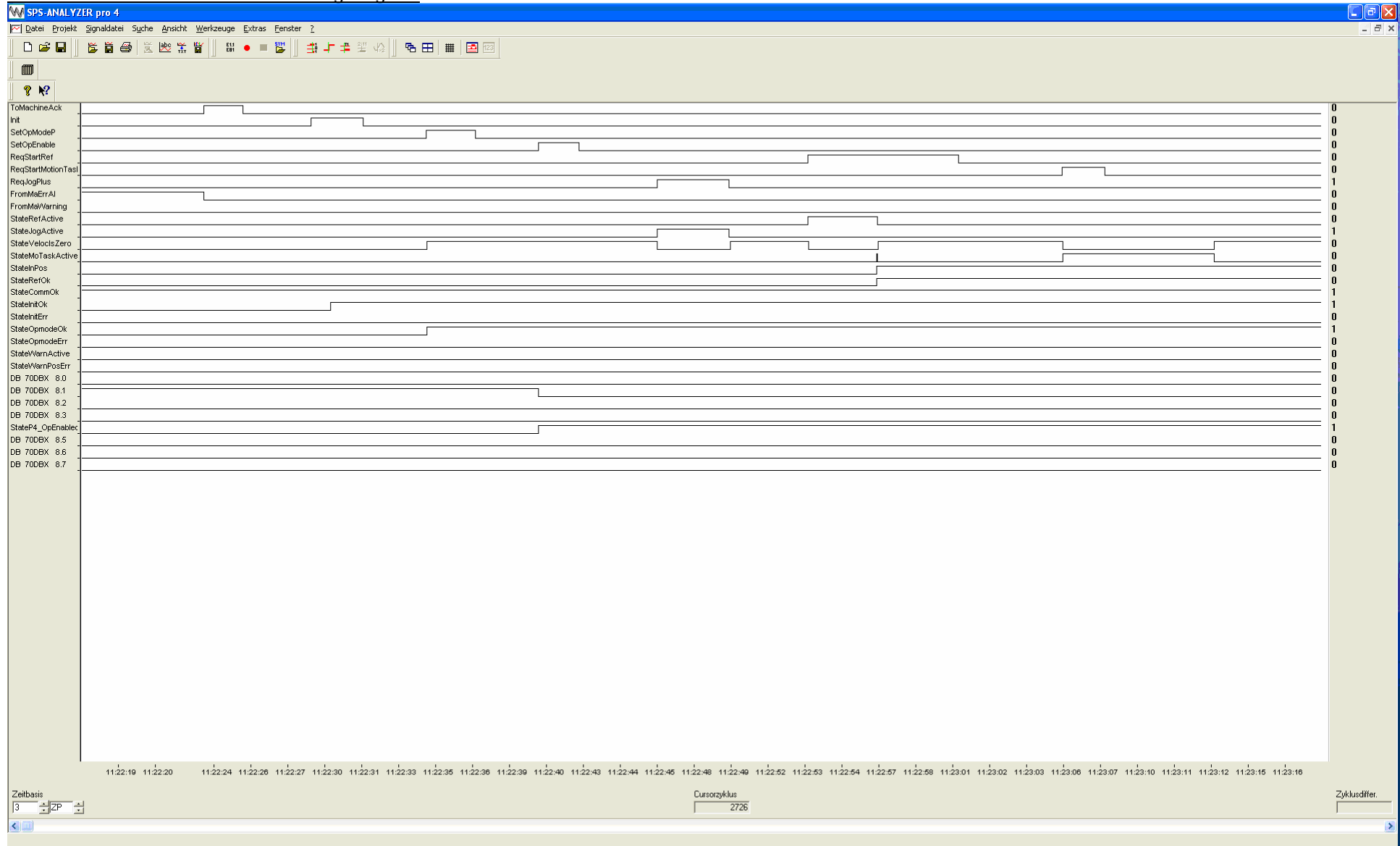

Bitcodierung der Fehler – PNU 1001 - ERRCODE

Axis_PNU1001	MD 1000	DWORD	Fehler
Axis_PNU1001_Bit31_F32	M 1000.7	BOOL	Systemfehler
Axis_PNU1001_Bit30_F31	M 1000.6	BOOL	Fehler SafetyCard
Axis_PNU1001_Bit29_F30	M 1000.5	BOOL	Fehler Notstop Time-out
Axis_PNU1001_Bit28_F29	M 1000.4	BOOL	Slotkartenfehler
Axis_PNU1001_Bit27_F28	M 1000.3	BOOL	Fehler Synchronisation EtherCAT
Axis_PNU1001_Bit26_F27	M 1000.2	BOOL	Fehler STO
Axis_PNU1001_Bit25_F26	M 1000.1	BOOL	Fehler bei Referenzfahrt - Hardware-Endschalter
Axis_PNU1001_Bit24_F25	M 1000.0	BOOL	Kommutierungsfehler
Axis_PNU1001_Bit23_F24	M 1001.7	BOOL	Fehler Warnung in Fehler gewandelt
Axis_PNU1001_Bit22_F23	M 1001.6	BOOL	Fehler in der CAN – Kommunikation
Axis_PNU1001_Bit21_F22	M 1001.5	BOOL	Fehler reserviert
Axis_PNU1001_Bit20_F21	M 1001.4	BOOL	Fehler Handling Error
Axis_PNU1001_Bit19_F20	M 1001.3	BOOL	Fehler Slot-Error
Axis_PNU1001_Bit18_F19	M 1001.2	BOOL	Fehler Einbruch der Zwischenkreisspannung
Axis_PNU1001_Bit17_F18	M 1001.1	BOOL	Fehler Ballast (defekter Ballasttransistor)
Axis_PNU1001_Bit16_F17	M 1001.0	BOOL	Fehler A/D-Wandler
Axis_PNU1001_Bit15_F16	M 1002.7	BOOL	Fehler Netz-BTB
Axis_PNU1001_Bit14_F15	M 1002.6	BOOL	Fehler I2tmax überschritten
Axis_PNU1001_Bit13_F14	M 1002.5	BOOL	Fehler Endstufe: Erdschluss, Motorkurzschluss oder Ballastkurzschluss
Axis_PNU1001_Bit12_F13	M 1002.4	BOOL	Fehler Umgebungstemperatur
Axis_PNU1001_Bit11_F12	M 1002.3	BOOL	Fehler reserviert
Axis_PNU1001_Bit10_F11	M 1002.2	BOOL	Fehler Bremse
Axis_PNU1001_Bit9_F10	M 1002.1	BOOL	Kabelbruch ROD Schnittstelle oder Zeitproblem Reglerbooten Master Slave
Axis_PNU1001_Bit8_F09	M 1002.0	BOOL	Fehler EEPROM
Axis_PNU1001_Bit7_F08	M 1003.7	BOOL	Fehler Überdrehzahl
Axis_PNU1001_Bit6_F07	M 1003.6	BOOL	Fehler interne Versorgungsspannungen
Axis_PNU1001_Bit5_F06	M 1003.5	BOOL	Fehler Motortemperatur
Axis_PNU1001_Bit4_F05	M 1003.4	BOOL	Fehler Unterspannung
Axis_PNU1001_Bit3_F04	M 1003.3	BOOL	Feedback-Fehler
Axis_PNU1001_Bit2_F03	M 1003.2	BOOL	Schleppfehler bei Ausführung der externen Trajektorie
Axis_PNU1001_Bit1_F02	M 1003.1	BOOL	Fehler Überspannung
Axis_PNU1001_Bit0_F01	M 1003.0	BOOL	Fehler Kühlkörpertemperatur

Bitcodierung der Warnungen und Herstellerspezifisches Statusregister – PNU 1002 - DRVSTAT

Axis_PNU1002	MD 1004	DWORD	Warnungen UND herstellerspezifisches Statusregister
Axis_PNU1002_Bit31	M 1004.7	BOOL	Fehler steht an
Axis_PNU1002_Bit30	M 1004.6	BOOL	Endstufe freigegeben
Axis_PNU1002_Bit29	M 1004.5	BOOL	Sicherheitsrelais hat angesprochen (STO)
Axis_PNU1002_Bit28	M 1004.4	BOOL	Drehzahl = 0
Axis_PNU1002_Bit27	M 1004.3	BOOL	-
Axis_PNU1002_Bit26	M 1004.2	BOOL	Initialisierung beendet (interne Initialisierung Verstärkers abgeschlossen)
Axis_PNU1002_Bit25	M 1004.1	BOOL	Position 4 erreicht (s.o.)
Axis_PNU1002_Bit24	M 1004.0	BOOL	Position 3 erreicht (s.o.)
Axis_PNU1002_Bit23	M 1005.7	BOOL	Position 2 erreicht
Axis_PNU1002_Bit22	M 1005.6	BOOL	Position 1 erreicht
Axis_PNU1002_Bit21	M 1005.5	BOOL	-
Axis_PNU1002_Bit20	M 1005.4	BOOL	Positions latch erfolgt
Axis_PNU1002_Bit19	M 1005.3	BOOL	In Position
Axis_PNU1002_Bit18	M 1005.2	BOOL	Aktuelle Position = Home Position (Referenzschalter ist belegt)
Axis_PNU1002_Bit17	M 1005.1	BOOL	Referenzpunkt gesetzt (nach einer Referenzfahrt bzw. Absolutwertgeber)
Axis_PNU1002_Bit16	M 1005.0	BOOL	Fahrauftrag aktiv - Fahrsatz, Tippbetrieb, Referenzfahrt)
Axis_PNU1002_Bit15_n16	M 1006.7	BOOL	Warnung 16: Reserve
Axis_PNU1002_Bit14_n15	M 1006.6	BOOL	Warnung 15: Geschwindigkeits-Strom Tabelle INXMODE 35 Fehler
Axis_PNU1002_Bit13_n14	M 1006.5	BOOL	Warnung 14: SinCos Kommutierung nicht vollzogen
Axis_PNU1002_Bit12_n13	M 1006.4	BOOL	Warnung 13: Erweiterungskarte arbeitet nicht ordnungsgemäß
Axis_PNU1002_Bit11_n12	M 1006.3	BOOL	Warnung 12: HIPERFACE® oder EnDat®: Motordefaultwerte wurden geladen
Axis_PNU1002_Bit10_n11	M 1006.2	BOOL	Warnung 11: Endschalter NSTOP betätigt
Axis_PNU1002_Bit9_n10	M 1006.1	BOOL	Warnung 10: Endschalter PSTOP betätigt
Axis_PNU1002_Bit8_n09	M 1006.0	BOOL	Warnung 9: Beim Fahrauftrag-Start war kein Referenzpunkt gesetzt
Axis_PNU1002_Bit7_n08	M 1007.7	BOOL	Warnung 8: Ein fehlerhafter Fahrauftrag wurde gestartet
Axis_PNU1002_Bit6_n07	M 1007.6	BOOL	Warnung 7: Software-Endschalter 2 überschritten
Axis_PNU1002_Bit5_n06	M 1007.5	BOOL	Warnung 6: Software-Endschalter 1 überschritten
Axis_PNU1002_Bit4_n05	M 1007.4	BOOL	Warnung 5: Netzphase fehlt
Axis_PNU1002_Bit3_n04	M 1007.3	BOOL	Warnung 4: Ansprechüberwachung (Feldbus) aktiv
Axis_PNU1002_Bit2_n03	M 1007.2	BOOL	Warnung 3: eingestelltes Schleppfehler-Fenster überschritten
Axis_PNU1002_Bit1_n02	M 1007.1	BOOL	Warnung 2: eingestellte Bremsleistung erreicht
Axis_PNU1002_Bit0_n01	M 1007.0	BOOL	Warnung 1: I ² t-Meldeschwelle überschritten

Switch ON and Start MotionTask timing diagram



Kill MotionTask timing diagram

