



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

## Inhaltsverzeichnis

Hinweise für das S7-Projekt „Sv14_Aw2“ für SERVOSTAR® 300/400/600 & S700 .....	2
Bausteinübersicht .....	5
Anmerkungen .....	5
Eingang- und Ausgangsschnittstelle des FB14 - Axis_01_FB .....	6
FB14 - Axis_01_FB .....	6
DB 70 - IF_TP170 - Interface DB Touch Panel TP170 .....	6
Gesamtübersicht der Schnittstelle vom FB14 .....	7
Programmbeispiele zur Umrechnung der reglerinternen Einheiten in SI-Einheiten .....	11
OB1 - Netzwerk 16: "IF_TP170".ActualPosition ( see Pnu 1800: SI-Unit ) .....	11
OB1 - Netzwerk 17: "IF_TP170".ActualSpeed ( see Pnu 1815: SI-Unit ) .....	11
OB1 - Netzwerk 18: "IF_TP170".ActualCurrent ( see Pnu 1688: SI-Unit ) .....	11
Inbetriebnahme FB14 Axis_01_FB .....	12
S7-SPS - AUS/EIN-Schalten .....	12
S7-SPS - Start/Stop-Schalten .....	12
iAck =1 .....	12
Initialisierung .....	12
Betriebsart setzen ( programmierte Betriebsarten Positionieren, Drehzahl digital, Drehmoment digital ) .....	12
Betrieb freigeben .....	12
Betriebsart Positionieren .....	13
Tippbetrieb .....	13
Referenzfahrt .....	13
Start eines EEPROM oder RAM Fahrauftrags .....	13
Start eines Direktfahrauftrags ( hat die Nummer 0 ) .....	13
Fahrauftragsarten (PNU 1785 - häufige Werte vgl. Handbuch Kap. IV 2.5.3 ) .....	14
Betriebsart Drehzahl digital .....	15
Betriebsart Drehmoment digital .....	15
Parameter schreiben .....	16
Parameter lesen .....	16
Zusätzliche Profibus-Funktionen .....	16
Beispiel für eine Schrittkette „Initialisierung und Enablen SERVOSTAR“ .....	17
Beispiel für eine Schrittkette „Starte Fahrsätze im SERVOSTAR“ .....	18
Bausteine für SERVOSTAR-Parameter .....	20
DB 20 - Axis_01_WriteDataDB .....	20
DB 21 - Axis_01_ReadDataDB .....	20
FB16 Axis_01_Write .....	21
FB17 Axis_01_Read .....	22
FB18 Axis_01_CompareDB .....	23
Tipps und Infos: .....	24
Fehlermeldungen von den Bausteinen FB14, FB16 und FB17 .....	24
Override über Profibus .....	24
Aktivieren der Trajektorie/Profil Beschleunigung Sin^2 bei Direktfahrsatz Nr. 0 .....	25
Bitcodierung der Fehler – PNU 1001 - ERRCODE .....	27
Bitcodierung der Warnungen und Herstellerspezifisches Statusregister – PNU 1002 - DRVSTAT .....	28
Timing diagram EINSchalten und Start MotionTask .....	29
Timing diagram Kill MotionTask .....	30

## Hinweise für das S7-Projekt „Sv14\_Aw2“ für SERVOSTAR® 300/400/600 & S700

Dieses Dokument ist gültig für die PROFIBUS Anbindung der Kollmorgen Servoverstärker SERVOSTAR® 300/400/600 & S700 an eine Simatic S7 Steuerung. Im folgenden Text werden die Servoverstärker SERVOSTAR® 300/400/600 & S700 kurz als „SERVOSTAR“ bezeichnet.

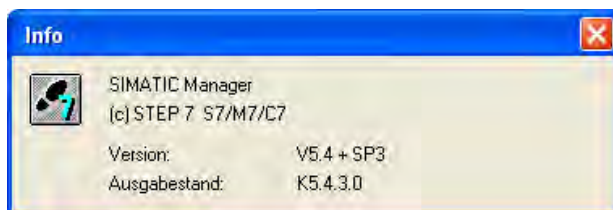
	<p><b>Verwenden Sie das S7-Projekt „Sv14_Aw2“ niemals unverändert in einer Anwendung.</b>  <b>Das S7-Projekt „Sv14_Aw2“ ist ein Beispiel wie der SERVOSTAR in ein S7-Projekt integriert werden kann. Dieses Projektbeispiel muss immer an die vorhandene Anwendung angepasst werden.</b></p>
	<p><b>KOLLMORGEN Europe GmbH haftet nicht für Schäden und schliesst alle Ansprüche aus, die sich durch den Einsatz des S7-Projekts „Sv14_Aw2“ oder Programmteilen daraus ergeben könnten.</b></p>

Gute Kenntnisse in S7, der Steuerungs- und Antriebstechnik werden vorausgesetzt.

Alle Verweise auf das Handbuch beziehen sich auf die technische Beschreibung „PROFIBUS S300/S400/S600/S700 Communication Profile“.

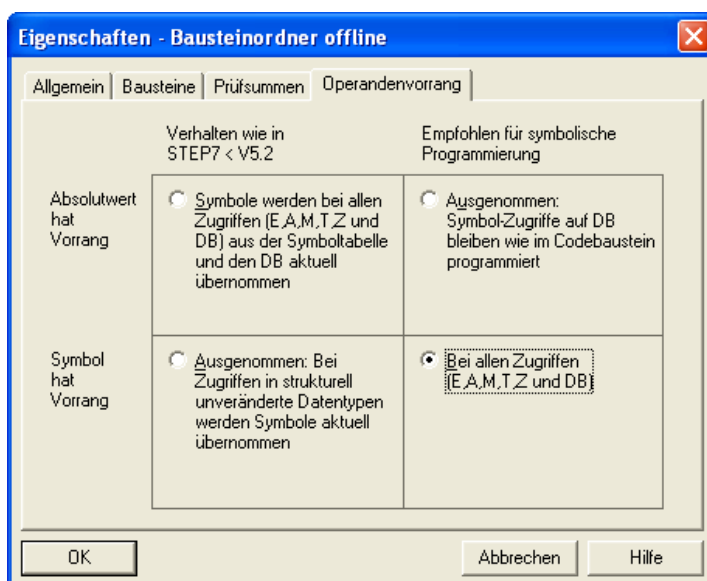
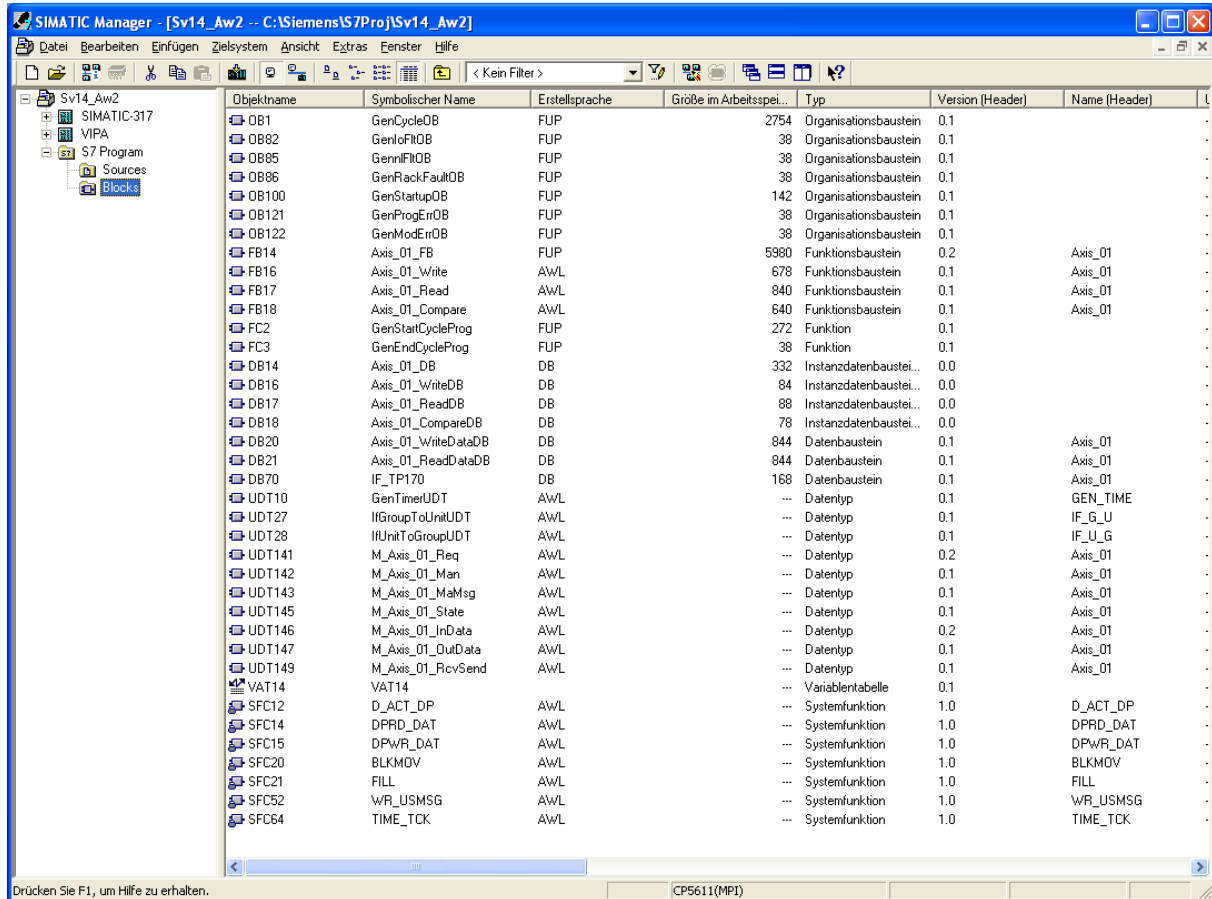
Die verwendeten Begriffe und Abkürzungen entsprechen der Setup Software bzw. dem Handbuch in der Sprache „English“.

Das S7-Projekt „Sv14\_Aw2“ ist erstellt in der Sprache English mit Mnemonik deutsch und der Version:



Das S7-Projekt „Sv14\_Aw2“ besteht aus den folgenden Komponenten:

- Simatic-317 – enthält nur die Hardwarekonfiguration ( system data ) für eine S7-CPU317-2DP
- VIPA – enthält nur die Hardwarekonfiguration ( system data ) für eine VIPA-CPU315-SB
- S7-Program – enthält die Bausteine und Symbolik

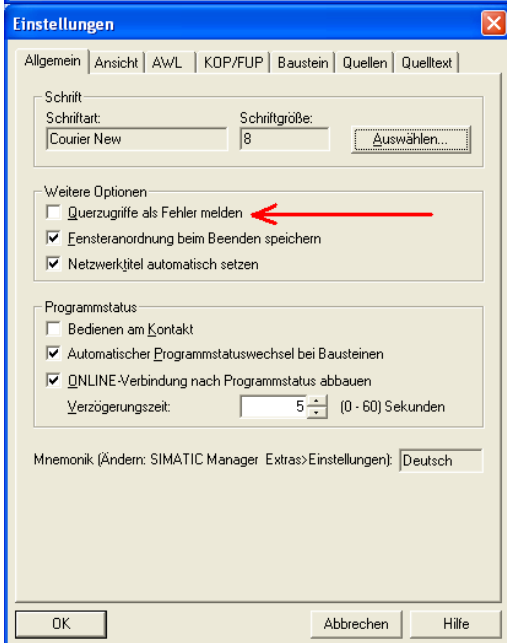
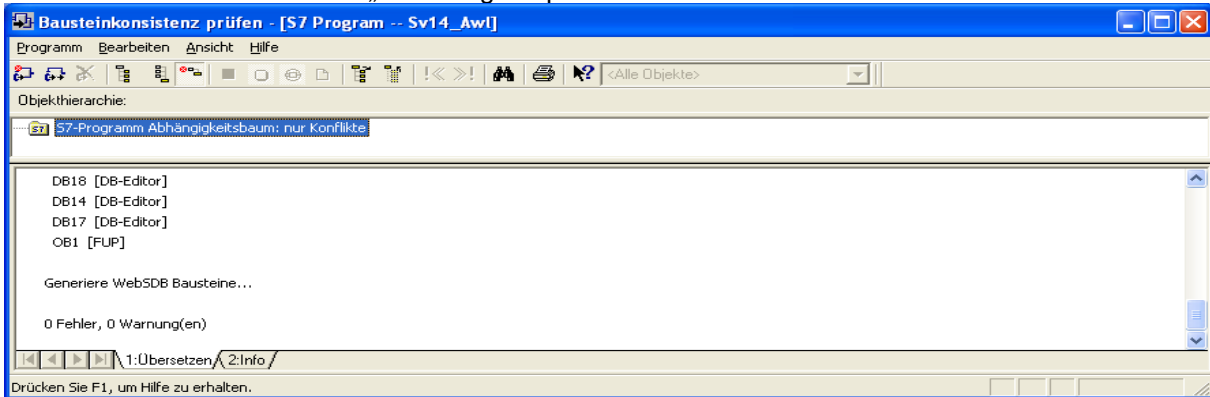


Die Eigenschaften des Baustein-Ordners Offline – Operandenvorrang sind:

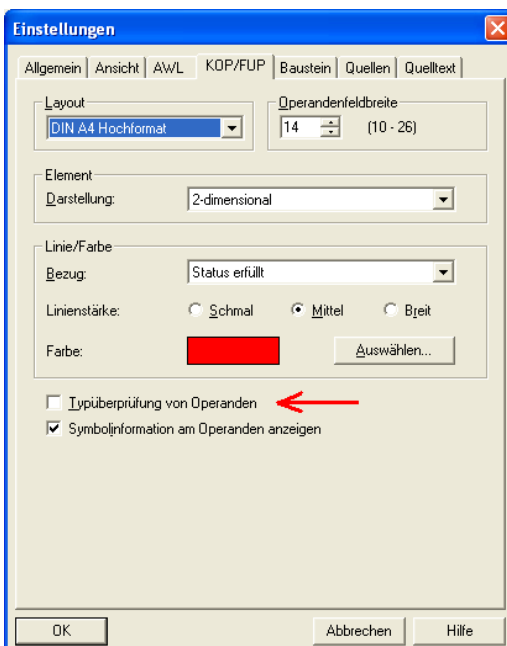
„Symbol hat Vorrang bei allen Zugriffen (E,A,M,T,Z und DB)“.

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 passend 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.

## Bausteinübersicht

FB14/DB14	Hantierungsbaustein zwischen S7-SPS und SERVOSTAR
FB16/DB16	Baustein für den FB14 zum Daten schreiben in den SERVOSTAR
DB20	Enthält die Daten, die mit dem FB16 in den SERVOSTAR geschrieben werden
FB17/DB17	Baustein für den FB14 zum Daten lesen aus dem SERVOSTAR
DB21	Enthält die Daten, die mit dem FB17 aus dem SERVOSTAR gelesen werden
FB18/DB18	Baustein zum Datenvergleich vom z. Bsp. DB20 und DB21
DB70	Interface-Datenbaustein 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 gegebenenfalls von Hand in den Bausteinen korrigiert werden.

Der FB14 nutzt die folgenden S7 Systembausteine:

SFC12 – D\_ACT\_DP – Deaktivieren und Aktivieren eines DP-Normslaves  
 SFC14 – DPRD\_DAT – Konsistente Daten eines DP-Normslaves lesen  
 SFC15 – DPWR\_DAT – Konsistente Daten eines DP-Normslaves schreiben  
 SFC20 – BLKMOV – Speicherbereich kopieren  
 SFC21 – FILL – Speicherbereich vorbesetzen

Im FC2 werden folgende S7 Systembausteine benutzt:

SFC52 – WR\_USMSG – Anwenderdefiniertes Diagnoseereignis in S7-SPS-Diagnosepuffer schreiben  
 SFC64 – TIME\_TCK – Systemzeit lesen ( damit werden im Programm die Timer realisiert )

### 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\_Aw2“ enthält ein komplett funktionsfähiges SPS-Programm für den SERVOSTAR.

Mit einer S7-CPU und einem SERVOSTAR mit Profibuskarte und Motor kann mit der VAT14 (Variablentabelle) der komplette Servoantrieb schnell und einfach in Betrieb genommen werden.  
 Der DB70 kann auch einfach als Interfacedatenbaustein für ein Operator/TouchPanel/Visualisierung benutzt werden.

## Eingang- und Ausgangschnittstelle des FB14 - Axis\_01\_FB

In der Schnittstelle vom FB14 werden UDT – Anwenderdefinierte Datentypen verwendet.

UDT 141 - M\_Axis\_01\_Request – Request  
 UDT 143 - M\_Axis\_01\_MaMsg – Machine message  
 UDT 145 - M\_Axis\_01\_State  
 UDT 146 - M\_Axis\_01\_InData  
 UDT 147 - M\_Axis\_01\_OutData  
 UDT 149 - M\_Axis\_01\_RcvSend - Receive and Send Profibus

Diese UDT werden auch noch im DB 70 - IF\_TP170 - Interface DB Touch Panel TP170 verwendet. Dadurch ist sichergestellt dass die Datenstruktur in dem S7-Projekt eine einheitliche gemeinsame Datenbasis hat. Eine Änderung an der 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.

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

### FB14 - Axis\_01\_FB

VAR_INPUT		
iData	UDT146	
iRequest	UDT141	
...		
VAR_OUTPUT		
oMaMsg	UDT143	
oState	UDT145	
oData	UDT147	
oAxis	UDT149	
...		

### DB 70 - IF\_TP170 - Interface DB Touch Panel TP170

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

## Gesamtübersicht der Schnittstelle vom FB14

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
OpModePb	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 <sup>15</sup> / VOSPD )
DigitalTorque	STRUCT	
Icmd	WORD	PNU1870 - Icommand ( I[A] = Icmd * IpeakAmplifier[A] / 3280 )
PnuWrite	STRUCT	
Number	WORD	
Index	WORD	
Value	DWORD	
PnuRead	STRUCT	
Number	WORD	
Index	WORD	
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
Res_1_0	BOOL	
Res_1_1	BOOL	
Res_1_2	BOOL	
Res_1_3	BOOL	
Res_1_4	BOOL	
Res_1_5	BOOL	
Res_1_6	BOOL	
Res_1_7	BOOL	
iGenSysTime	TIME	System time at beginning of OB1
iGenStartup	BOOL	Startup cycle after PLC startup
iGenClock2000	BOOL	Periode clock 2000ms
iEnError	BOOL	Enable error messages
iEnTimeout	BOOL	Enable timeout movement referencing and positioning
iAck	BOOL	Acknowledge WarningsErrors
iActSlave	BOOL	Activate DP-slave
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
iPnuWriteStart	BOOL	Request PnuWriteStart pulse 0->1
iPnuReadStart	BOOL	Request PnuReadStart static 1
iInit	BOOL	Initialize axis with disable axis
iSetOpModePb	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
ErrRes_0_2	BOOL	
ErrRes_0_3	BOOL	
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
ErrRes_1_3	BOOL	
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 DP communication is ok
InitOK	BOOL	Axis initialisation is ok
InitError	BOOL	Axis initialisation error
OpModePbOk	BOOL	Axis opmode Profibus is ok
OpModePbError	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- STOPMODEDECDIS
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
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	
oOpmodePb	WORD	Actual OpmodePb Profibus ( Pnu 930)
oError	BOOL	Error is active

<sup>1\*)</sup> Die aktuelle Positon wird in den Prozessdaten nur in reglerinternen Einheiten ( $2^{20}$  Inkr. pro Motorumdrehung) übertragen. Die Umrechnung in Benutzereinheiten kann in der SPS erfolgen, abhängig von der mit der Bediensoftware im Regler eingestellten Positonsauflösung.

Beispiel: Auflösung = 5000 Inkr/ 3 Umdrehungen

$$\Rightarrow \text{Position in Benutzereinheiten} = \text{Actual\_Position} \times 5000 / (3 \times 2^{20})$$

## Programmbeispiele zur Umrechnung der reglerinternen Einheiten in SI-Einheiten

### OB1 - Netzwerk 16: "IF\_TP170".ActualPosition ( see Pnu 1800: SI-Unit )

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

// "IF_TP170".ActualPosition [µm, SI-Unit - DWORD]
// = "Axis_01_DB".oActualPosition [Counts] * (PGEARI / PGEARO) / 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_TP170".ActualPosition
```

### OB1 – Netzwerk 17: "IF\_TP170".ActualSpeed ( see Pnu 1815: SI-Unit )

```
// Example: VOSPD 3600 rpm

// "IF_TP170".ActualSpeed [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_TP170".ActualSpeed
```

### OB1 – Netzwerk 18: "IF\_TP170".ActualCurrent ( see Pnu 1688: SI-Unit )

```
// example for S303 or S703 - with Ipeak 9 Ampere

// "IF_TP170".ActualCurrent in mA [SI-Unit - Word]
// = "Axis_01_DB".oActualCurrent [Counts] * DIPEAK / 3280
// = "Axis_01_DB".oActualSpeed [Counts] * 9000 / 3280

// use datatype 32-bit integer

L   "Axis_01_DB".oActualCurrent
L   9000
*I
L   3280
/D
T   "IF_TP170".ActualCurrent
```

## Inbetriebnahme FB14 Axis\_01\_FB

### S7-SPS - AUS/EIN-Schalten

Fällt der Profibusmaster S7-SPS aus dann startet im SERVOSTAR der Watchdog-Timeout.  
Die S7-SPS macht nach dem Anlauf einen Software-Reset (GenStartUp - M1.2) und löscht alle Fehlermeldungen und Zustände im SPS-Programm.

### S7-SPS - Start/Stop-Schalten

macht einen Software-Reset (GenStartUp - M1.2) und löscht alle Fehlermeldungen und Zustände im SPS-Programm. Der SERVOSTAR bleibt enabled und ein aktiver Fahrauftrag wird weiterhin ausgeführt.

### iAck =1

setzt Warnungen und Fehler im SERVOSTAR und in den Bausteinen FB14, FB16, FB17, FB18 zurück.

### Initialisierung

Bei der Initialisierung wird ein Null-Telegramm im PKW und PZD gesendet. Die Achse wird disabled.

- 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 )

Nach dem Einschalten ist der SERVOSTAR immer in der sicheren Betriebsart -126

- Schreibe die gewünschte Betriebsart in iData.OpMode  
( siehe Handbuch: Positionieren: 2 ; Drehzahl digital: 1 ; Drehmoment digital: -2 )
- Setze iSetOpModePb =1 ( intern wird ein Impuls generiert )
- Warte bis oState.OpModeOk = 1  
Die aktivierte Betriebsart wird angezeigt in oOpmodePb  
Wenn oState.OpModePbOk nicht =1 oder oState.OpModePbError =1 dann war Betriebsart setzen nicht erfolgreich. Fehlerursache siehe Receive-Telegramm SERVOSTAR.
- Jetzt werden auch oActualPosition, oActualSpeed und oManufactState mit Werten angezeigt.
- Die Betriebsart setzen wird mit einem Timeout von 1 sec überwacht.

### Betrieb freigeben

Nach dem Einschalten ist der SERVOSTAR im Zustand „Switch-On inhibited / Einschaltsperrung“.

iFastStop und iFastStopDisableAxis müssen =0 und iSwEnable muss =1 sein.

Am SERVOSTAR muss HardwareEnable =1 und falls vorhanden muss AS-Enable =1 sein.

Die Zwischenkreisspannung wird angelegt und im Display vom SERVOSTAR wird „Pxx“ angezeigt.

- 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.  
Fehlerursache siehe Receive-Telegramm SERVOSTAR.

### Betriebsart Positionieren

Die Betriebsart Positionieren (oState.OpModeOk = 1 und oOpModePb =2 dez ) ist aktiviert.

#### Tippbetrieb

Voraussetzung: Der Antrieb ist im Zustand „Operation enabled / Betrieb freigegeben“ und Hardware-Enable ist vorhanden und ist fehlerfrei und im Display vom SERVOSTAR wird „Exx“ angezeigt.

- Schreibe die gewünschte Tippgeschwindigkeit in „iData.Config.JogSpeed“
  - Setze iRequest.JogPlus =1 -> Der Antrieb bewegt sich in die positive Richtung
  - Setze iRequest.JogMinus =1 -> Der Antrieb bewegt sich in die negative Richtung
- oState.JoggingActive ist =1 wenn der Antrieb sich bewegt.

#### Referenzfahrt

Die Art der Referenzfahrt ist mit der Bediensoftware DRIVEGUI / DRIVE einzustellen.

Voraussetzung: Der Antrieb ist im Zustand „Operation enabled / Betrieb freigegeben“ und Hardware-Enable ist vorhanden und ist fehlerfrei.

- 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

Mit der Bediensoftware DRIVEGUI / DRIVE werden Fahraufträge angelegt.

Nur die Parameter der RAM-Fahraufträge können im Antrieb im Zustand „Operation enabled / Betrieb freigegeben“ geändert werden ( siehe ASCII-Kommando MTMUX )

Voraussetzung: Der Antrieb ist im Zustand „Operation enabled / Betrieb freigegeben“ und referenziert.

- Schreibe die Nummer des zu startenden 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 Antrieb hat den Fahrauftrag ausgeführt

Mit iEnTimeout =1 und iData.Config.TO\_Position xxx ms [TIME] wird der Fahrauftrag mit einem Timeout überwacht.

#### Start eines Direktfahrauftrags ( hat die Nummer 0 )

Voraussetzung: Der Antrieb ist im Zustand „Operation enabled / Betrieb freigegeben“ und referenziert.

- Schreibe 0 in iData.MotionTask.Number
- Schreibe Sollgeschwindigkeit in iData.MotionTask.DirectSpeed
- Schreibe Zielposition in iData.MotionTask.DirectPosition
- 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 Antrieb hat den Fahrauftrag ausgeführt

Anmerkung: Mittels Pointer ( P#DB.DBX Byte ) kann der iData.MotionTask mit einem SFC20-BLKMOV einfach beschrieben werden.

Mit iEnTimeout =1 und iData.Config.TO\_Position xxx ms [TIME] wird der Fahrauftrag mit einem Timeout überwacht.

Siehe Timing Diagram EINSchalten und Start MotionTask

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.

Ein gestarteter MotionTask kann mit iPauseMotionTask =1 angehalten werden.

Ein gestarteter MotionTask kann mit iRequest.StopMotionTask =1 gestoppt werden.

Ein gestoppter MotionTask bleibt im Regler weiterhin aktiv, d.h. wenn der Regler wieder in den oState.P4\_OperationEnabled geschaltet wird, wird der MotionTask fortgesetzt.

Mit iSwEnable =0 wird der gestoppte MotionTask gekillt, d.h wenn der Regler wieder in den oState.P4\_OperationEnabled geschaltet wird, wird der MotionTask nicht mehr fortgesetzt.

Siehe: Timing diagram Kill MotionTask

Hinweis: Mit dem Parameter INPT kann im Terminalfenster der Bediensoftware ( 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).

Hinweis zu Positionen und Geschwindigkeiten: Zielposition und Sollgeschwindigkeit können in den im Lageregler eingestellten Einheiten vorgegeben werden, wenn in der Fahrauftragsart PNU1785 das Bit 13 gesetzt ist (vgl. Handbuch Kap. IV.2.5.3).

Fahrauftragsarten (PNU 1785 - häufige Werte vgl. Handbuch Kap. IV 2.5.3 )

0x2000 (Bit 13 gesetzt) Absolute Positionierung mit Vorgabe der Geschwindigkeiten und Positionen in Benutzereinheiten und Fahrprofil Trapez.

0x2003 (Bits 0, 1, 13 gesetzt) Positionierung „relativ Soll“ mit Vorgabe von Geschwindigkeiten und Positionen in Benutzereinheiten.

0x12000 ( Bit13 und Bit16 gesetzt ) Absolute Positionierung mit Vorgabe der Geschwindigkeiten und Positionen in Benutzereinheiten und Fahrprofil Sinus<sup>2</sup>

Das Bit 16 kann nicht direkt im PZD-Kanal gesetzt werden.

Siehe: Aktivieren der Trajektorie/Profil Beschleunigung Sin<sup>2</sup> bei Direktfahrsatz Nr. 0

### Betriebsart Drehzahl digital

Die Betriebsart Drehzahl digital (oState.OpModeOk = 1 und oOpmodePb =1 dez ) ist aktiviert und der Betrieb ist freigegeben.

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 bremsst über die eingestellten Drehzahlrampen bis zum Stillstand ab.

Istposition, Istdrehzahl (16bit ) und das herstellerspezifische Statusregister HSW werden dabei zyklisch im PZD-Kanal übertragen.

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

PNU1886 - Ncommand ( Ncmd16 = Ncmd \* 2<sup>15</sup> / VOSPD )

### Betriebsart Drehmoment digital

Die Betriebsart Drehmoment digital (oState.OpModeOk = 1 und oOpmodePb = -2 dez ) ist aktiviert und der Betrieb ist freigegeben.

Schreibe Sollstrom in iData.DigitalTorque.Icmd

Setze iRequest.StartIcmd =1

Der Antrieb prägt den vorgegebenen Sollstrom ein.

Der Sollstrom kann dabei jederzeit verändert werden.

Setze iRequest.StartIcmd =0 und der Antrieb gibt den Sollstrom 0 Ampere aus.

Istposition, Iststrom und herstellerspezifisches Statusregister HSW werden dabei zyklisch im PZD-Kanal übertragen.

Der Stromwert wird dabei nach folgender Formel berechnet:

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

Mit dem 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.

Mit diesen 3 Betriebsarten sind die meisten Anforderungen in einer Maschine realisierbar. Weitere Betriebsarten sind derzeit nicht im FB14 implementiert, diese können jedoch einfach nachprogrammiert werden.

### Parameter lesen und schreiben:

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

### Parameter schreiben

Voraussetzung: Parameterkanal PKW wird nicht bereits benutzt

- Schreibe Nummer in iData.PnuWrite.Number
  - Schreibe Index in iData.PnuWrite.Index
  - Schreibe Wert in iData.PnuWrite.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 iData.PnuRead.Number
  - Schreibe Index in iData.PnuRead.Index
  - Schreibe Wert in iData.PnuWrite.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 ).

### Zusätzliche Profibus-Funktionen

Mit iActSlave =1 wird im FB14 mit dem SFC12 – D\_ACT\_DP der DP-Slave SERVOSTAR bei laufender SPS aktiviert bzw. mit iActSlave =0 deaktiviert.

Zudem wird die Profibuskommunikation auf

- Konfigurationsfehler
- Slaveausfall
- und Telegrammfehler überwacht

Die gesamten 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] abgespeichert 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.

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



### 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_TP170".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_TP170".Init

U      M      100.3
UN     "Axis_01_DB".oError
U      "Axis_01_DB".oState.InitOk
UN     "Axis_01_DB".oState.OpModePbOk
=      "IF_TP170".SetOpmodePb

U      M      100.3
UN     "Axis_01_DB".oError
U      "Axis_01_DB".oState.InitOk
U      "Axis_01_DB".oState.OpModePbOk
UN     "Axis_01_DB".oState.P4_OperationEnabled
=      "IF_TP170".SetOperationEnable

U      M      100.3
UN     "Axis_01_DB".oError
U      "Axis_01_DB".oState.InitOk
U      "Axis_01_DB".oState.OpModePbOk
U      "Axis_01_DB".oState.P4_OperationEnabled
UN     "Axis_01_DB".oState.ReferenceOk
=      "IF_TP170".Request.StartRef

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

```

Siehe Timing diagram EINSchalten und Start MotionTask

### 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.OpModePbOk
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_TP170".iData.MotionTask.Number  
m201: NOP    0  
  
      U      M      110.5  
      SPBN   m210  
      L      210  
      T      "IF_TP170".iData.MotionTask.Number  
m210: NOP    0  
  
      U      M      110.7  
      SPBN   m211  
      L      211  
      T      "IF_TP170".iData.MotionTask.Number  
m211: NOP    0  
  
      CLR  
      O      M      110.3  
      O      M      110.5  
      O      M      110.7  
      =      "IF_TP170".Request.StartMotionTask
```

Siehe Timing Diagram EINSchalten und Start MotionTask

## Bausteine für SERVOSTAR-Parameter

Das S7-Projekt „Sv14\_Aw2“ 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 über den nicht echtzeitfähigen PKW-Parameterkanal im Profibus in den SERVOSTAR.

Deklaration vom DB20 – für 100 Werte

KOP/AWL/FUP - [DB20 -- "Axis\_01\_WriteDataDB" -- Sv14\_Awl\S7 Program\...DB20]

Datei Bearbeiten Einfügen Zielsystem Test Ansicht Extras Fenster Hilfe

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 !!!
*0.0		STRUCT		
+0.0	Number	WORD	W#16#0	
+2.0	Index	WORD	W#16#0	
+4.0	Value	DWORD	DW#16#0	
=8.0		END_STRUCT		
=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 über den nicht echtzeitfähigen PKW-Parameterkanal im Profibus in die S7-SPS.

Deklaration vom DB21 – für 100 Werte

KOP/AWL/FUP - [DB21 -- "Axis\_01\_ReadDataDB" -- Sv14\_Awl\S7 Program\...DB21]

Datei Bearbeiten Einfügen Zielsystem Test Ansicht Extras Fenster Hilfe

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 !!!
*0.0		STRUCT		
+0.0	Number	WORD	W#16#0	
+2.0	Index	WORD	W#16#0	
+4.0	Value	DWORD	DW#16#0	
=8.0		END_STRUCT		
=808.0		END_STRUCT		

## FB16 Axis\_01\_Write

steuert den FB14 an zum Daten schreiben von der SPS in den SERVOSTAR

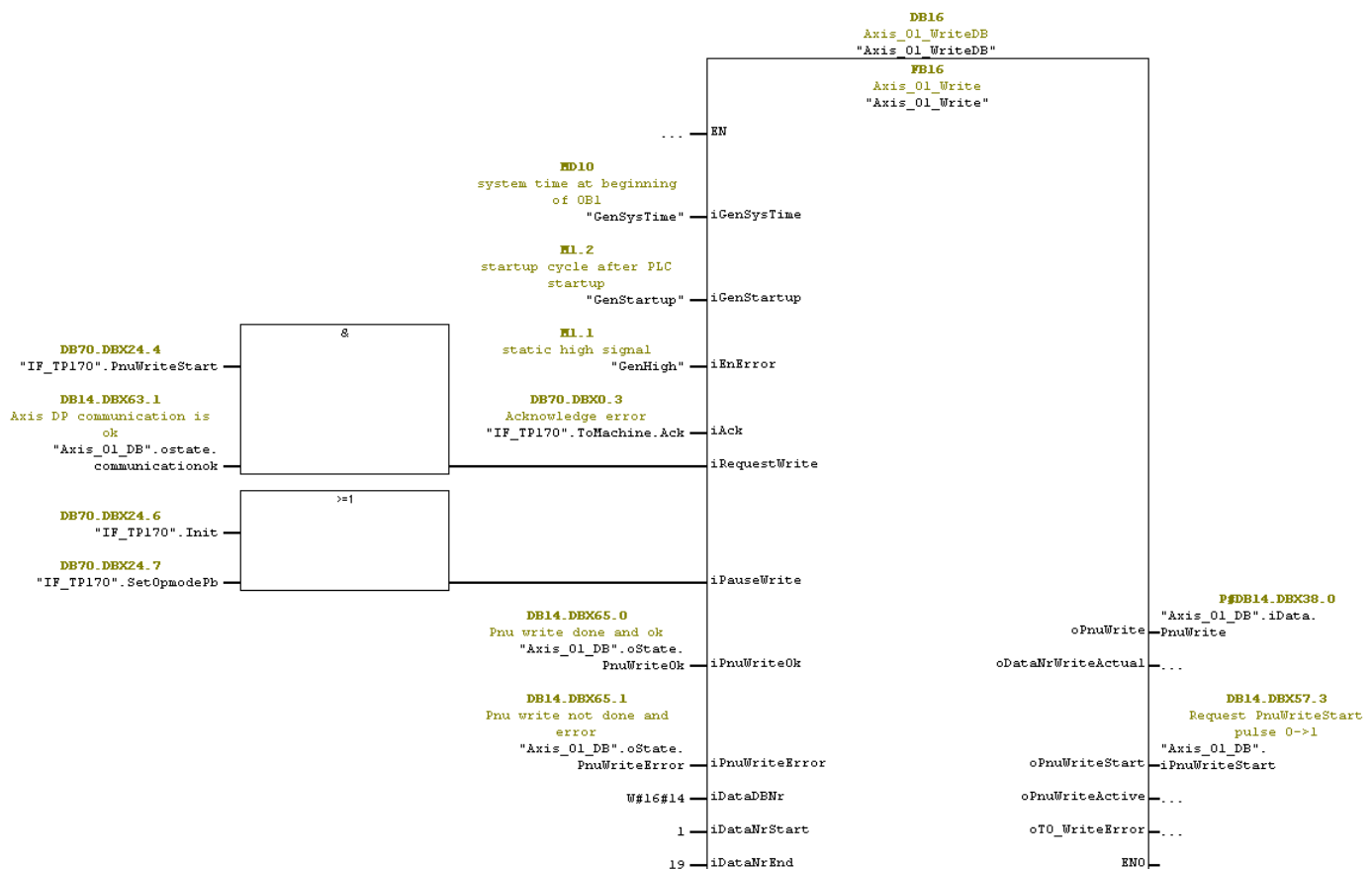
### Eingang- und Ausgangschnittstelle des FB16 - Axis\_01\_Write

VAR_INPUT		
iGenSysTime	TIME	System time at beginning of OB1
iGenStartup	BOOL	Startup cycle after PLC startup
iEnError	BOOL	Enable error messages
iAck	BOOL	Acknowledge WarningsErrors
iRequestWrite	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	BOOL	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 SPS

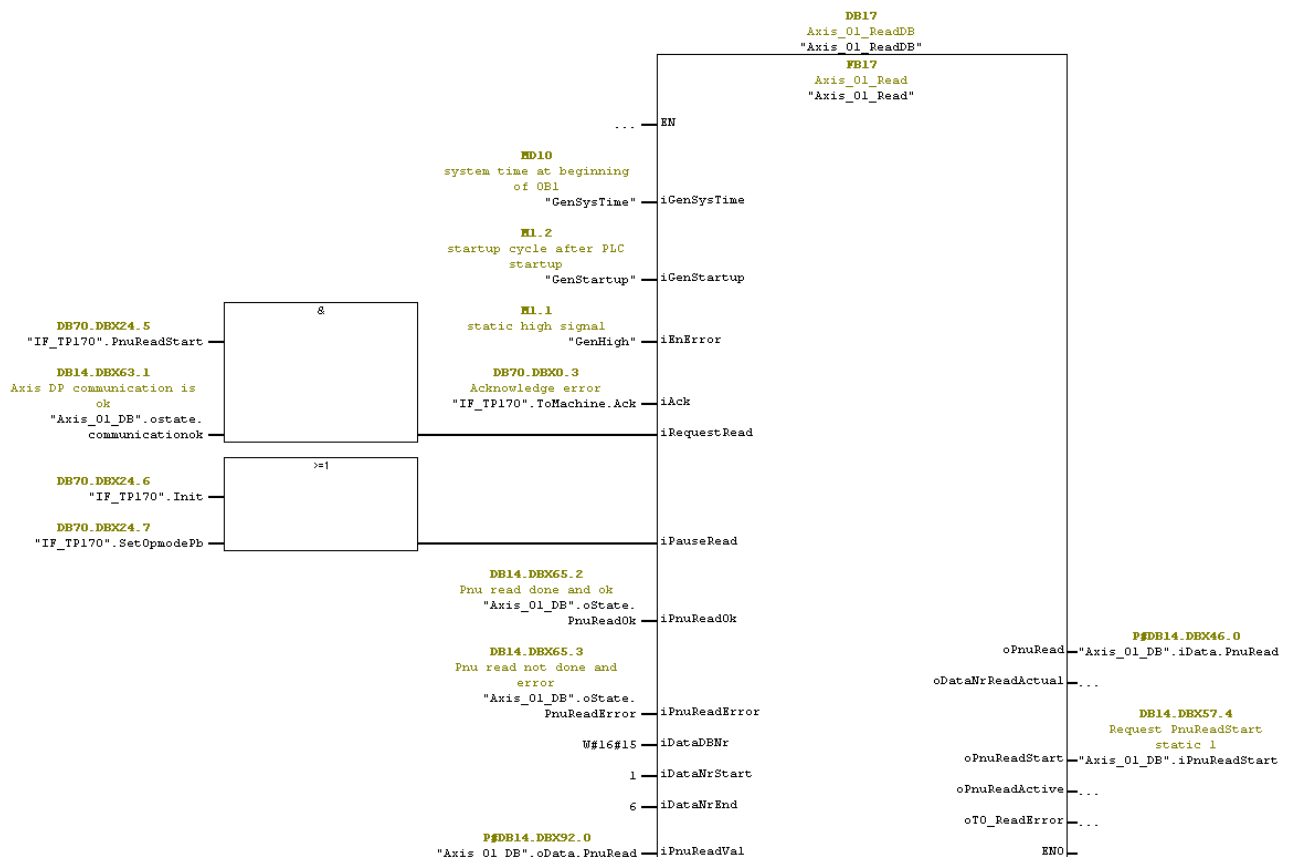
### Eingang- und Ausgangschnittstelle des FB17 - Axis\_01\_Read

VAR_INPUT		
iGenSysTime	TIME	System time at beginning of OB1
iGenStartup	BOOL	Startup cycle after PLC startup
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
iDataDBNr	WORD	ReadDataDBNumber
iDataNrStart	INT	DataNumberStart - first number from data to read
iDataNrEnd	INT	DataNumberEnd - last number from data to read
iPnuReadVal	STRUCT	PnuRead from Axis_FB actual with Number, Index, Value
Number	WORD	
Index	WORD	
Value	DWORD	

VAR_OUTPUT		
oPnuRead	STRUCT	PnuRead from Axis_FB actual with Number, Index
Number	WORD	
Index	WORD	
oDataNrReadActual	WORD	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:

Bei gleichzeitiger Anforderung Read und Write am FB14 generiert der FB14 den Fehler

oMaMsg.ErrReadWrite

### FB18 Axis\_01\_CompareDB

vergleicht einen Datenbereich von den geschriebenen Daten vom DB20 Axis\_01\_WriteDataDB mit den gelesenen Daten vom DB21 Axis\_01\_ReadDataDB.  
Pro SPS-Zyklus wird nur ein Datensatz verglichen.

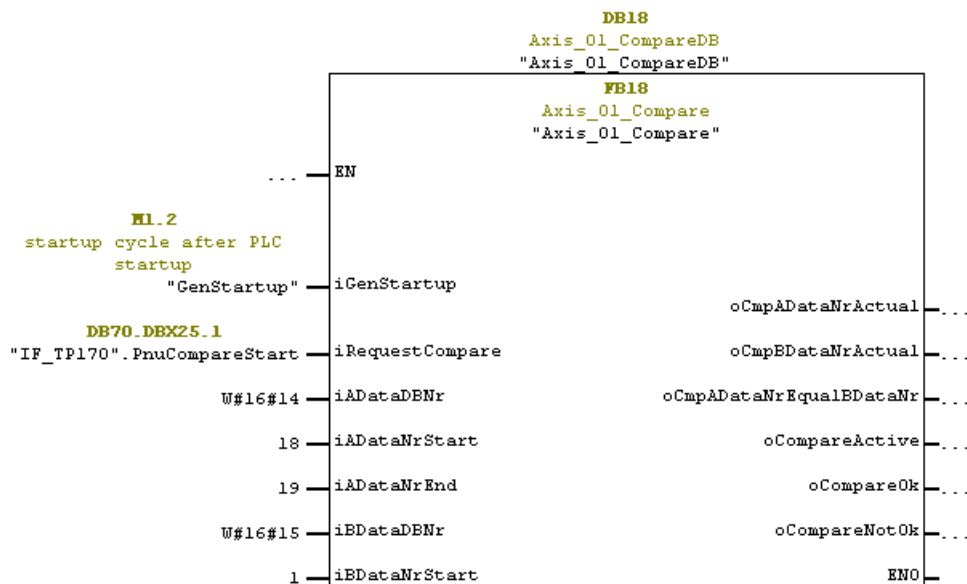
### Eingang- und Ausgangschnittstelle des FB18 Axis\_01\_CompareDB

VAR_INPUT		
iGenStartup	BOOL	Startup cycle after PLC startup
iRequestCompare	BOOL	Request compare
iADaDBNr	WORD	ADaDBNumber
iADaNrStart	INT	ADaNumberStart - first number from data to compare
iADaNrEnd	INT	ADaNumberEnd - last number from data to compare
iBDataDBNr	WORD	BDataDBNumber
iBDataNrStart	INT	BDataNumberStart - first number from data to compare

VAR_OUTPUT		
oCmpADaNrActual	WORD	Compare A Data number is actual
oCmpBDataNrActual	WORD	Compare B Data number is actual
oCmpADaNrEqualBDataNr	BOOL	Compare A Data number is equal B Data number
oCompareActive	WORD	comparing DataBlocks is active
oCompareOk	BOOL	Compare is ok, datas are equal
oCompareNotOk	BOOL	Compare is not ok, datas are not equal

### Netzwerk 10 : Axis\_Compare

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



## Tipps und Infos:

### Fehlermeldungen von den Bausteinen FB14, FB16 und FB17

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
ErrRes_0_2	BOOL	
ErrRes_0_3	BOOL	
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
ErrRes_1_3	BOOL	
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
---------------	------	---

## 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 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<sup>2</sup> bei Direktfahrsatz Nr. 0

Das Fahrprofil Sin<sup>2</sup> 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 lokalen Fahrsatz Nr. 0 (Direktfahrsatz).

Ist das Bit 16 von O\_C gesetzt, so wird ein Tabellenfahrsatz (Sinus2-Fahrt) gestartet. Das Bit 9 muss auf 0 gesetzt werden. Als Parameter wird eine Bit-Variable übergeben (16 Bits).

Die Bits von 0 bis 15 werden im Direktfahrsatz 0 ( PZD5 ) direkt adressiert, das Bit 16 muss deshalb anderweitig im S300/S700 gesetzt werden.

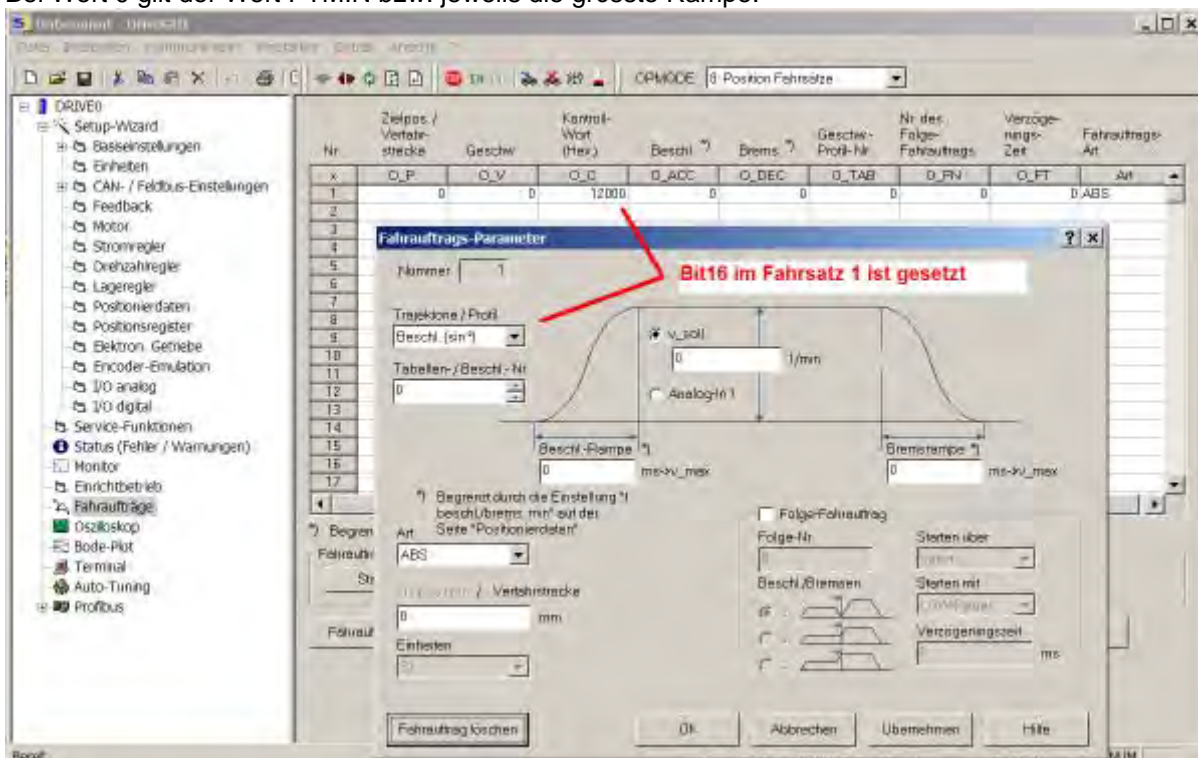
Das Kommando O\_C ist kein Parameter und bleibt deshalb nicht im Regler nullspannungsicher gespeichert. Zum Setzen von dem Bit16 gibt es zwei Möglichkeiten:

- 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.

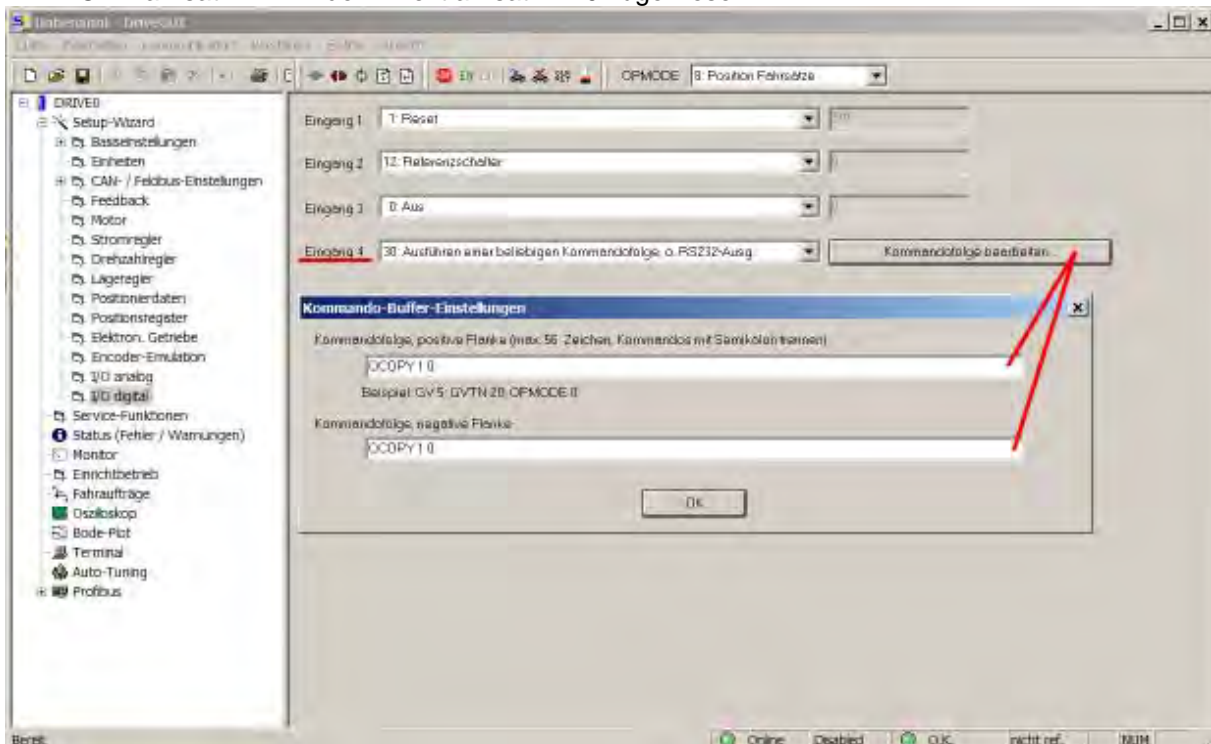
Aktivieren des Bit16 im EEPROM-Fahrsatz z. Bsp. Nr.1

Zusätzlich können auch die Beschleunigungs- O\_ACC und Bremsrampe O\_DEC eingestellt werden.

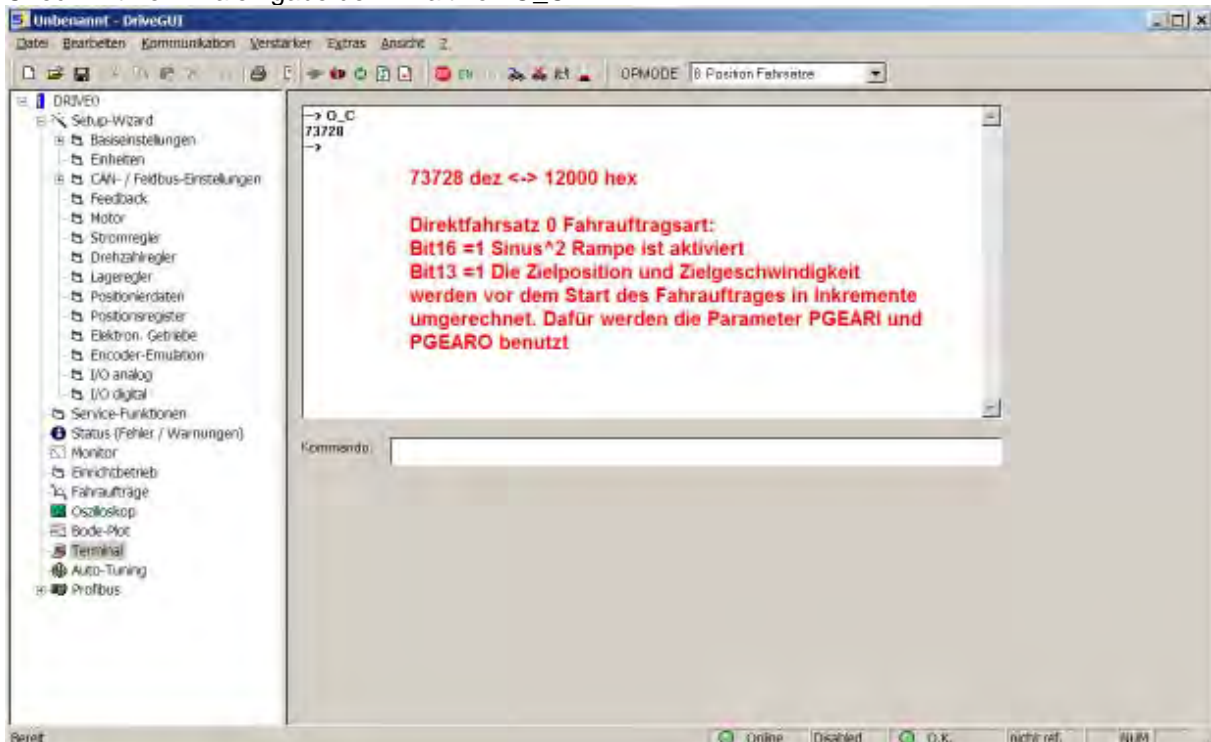
Bei Wert 0 gilt der Wert PTMIN bzw. jeweils die grösste Rampe.



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



Check mit Terminaleingabe den Inhalt von O\_C



# Bitcodierung der Fehler – PNU 1001 - ERRCODE

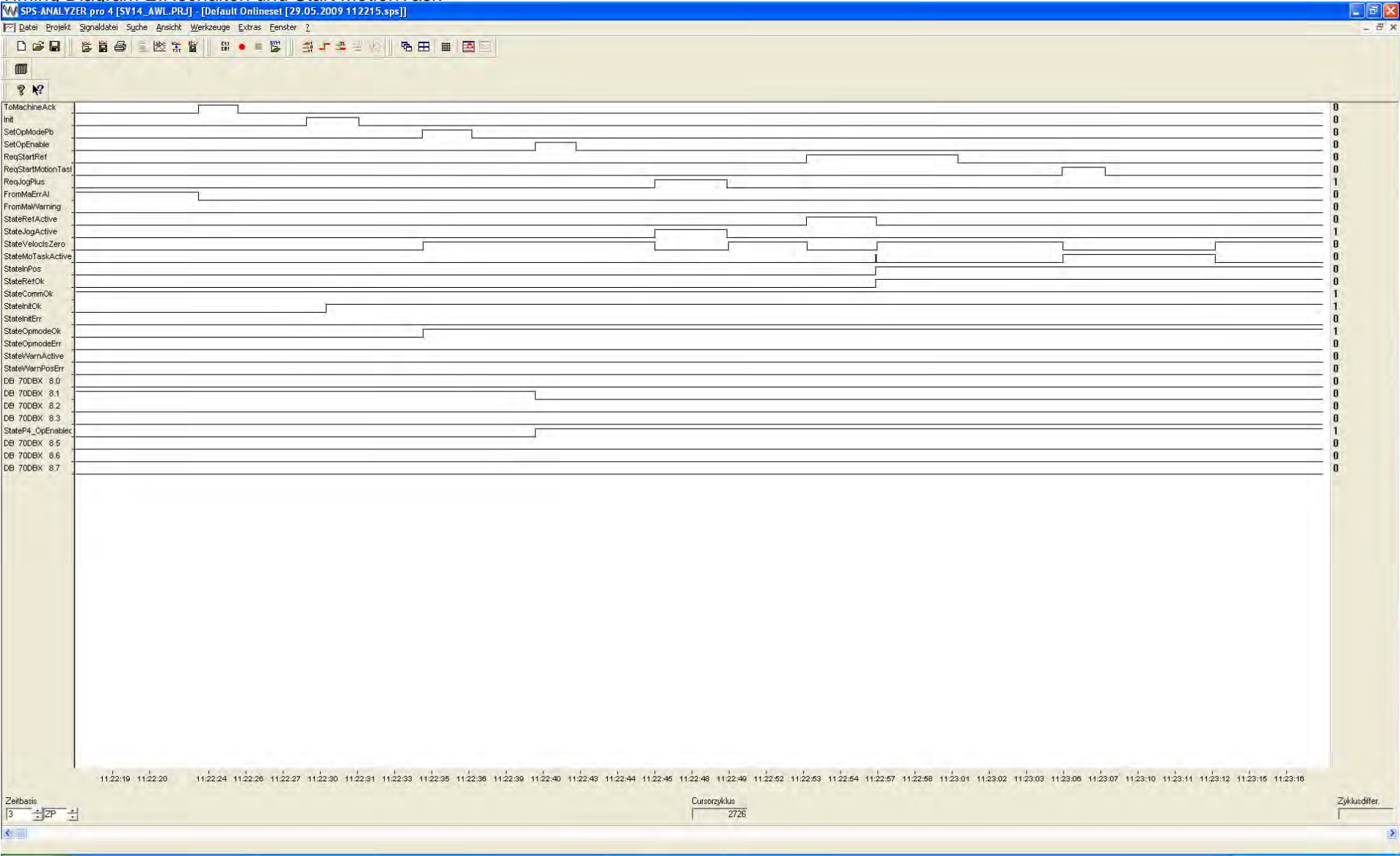
Axis_PNU1001	MD 1000	DWORD	Fehler
Axis_PNU1001_Bit31_F32	M 1000.7	BOOL	Systemfehler
Axis_PNU1001_Bit30_F31	M 1000.6	BOOL	Fehler SafetyCard
Axis_PNU1001_Bit29_F30	M 1000.5	BOOL	Fehler Notstop Time-out
Axis_PNU1001_Bit28_F29	M 1000.4	BOOL	Slotkartenfehler
Axis_PNU1001_Bit27_F28	M 1000.3	BOOL	Fehler Synchronisation 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	Positionslatch 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 <sup>2</sup> t-Meldeschwelle überschritten



Timing Diagram EINSchalten und Start MotionTask



## Timing Diagram Kill MotionTask

