Modules for Simatic S7 and Profibus Connection SERVOSTAR® 300/400/600 and S700

Table of contents

Notes for the S7 project "Sv14_Aw2" for SERVOSTAR® 300/400/600 & S700

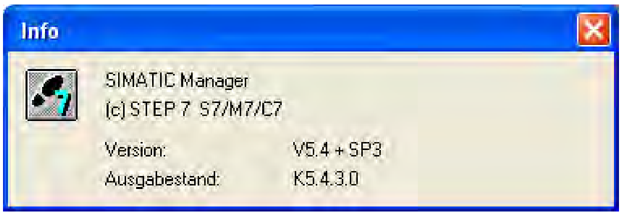
This document applies to the PROFIBUS connection of the Kollmorgen servo amplifiers SERVOSTAR® 300/400/600 & S700 to a Simatic S7 control. In the following text, the servo amplifiers SERVOSTAR® 300/400/600 & S700 are simply referred to as "SERVOSTAR".

⚠	**Never use the S7 project "Sv14_Aw2" without modifications in an application. The S7 project "Sv14_Aw2" is an example how the SERVOSTAR can be integrated in an S7 project. This project example must always be adapted to the existing application.**

⚠	**KOLLMORGEN Europe GmbH assumes no liability for damages and precludes all claims areising from the use of the S7 project "Sv14_Aw2" or program components from it.**

This manual assumes a good knowledge of S7, controller and drive technology.
All references to the manual refer to the technical description "PROFIBUS S300/S400/S600/S700 Communication Profile".
The terms and abbreviations used correspond to the setup software or the manual in the English language.

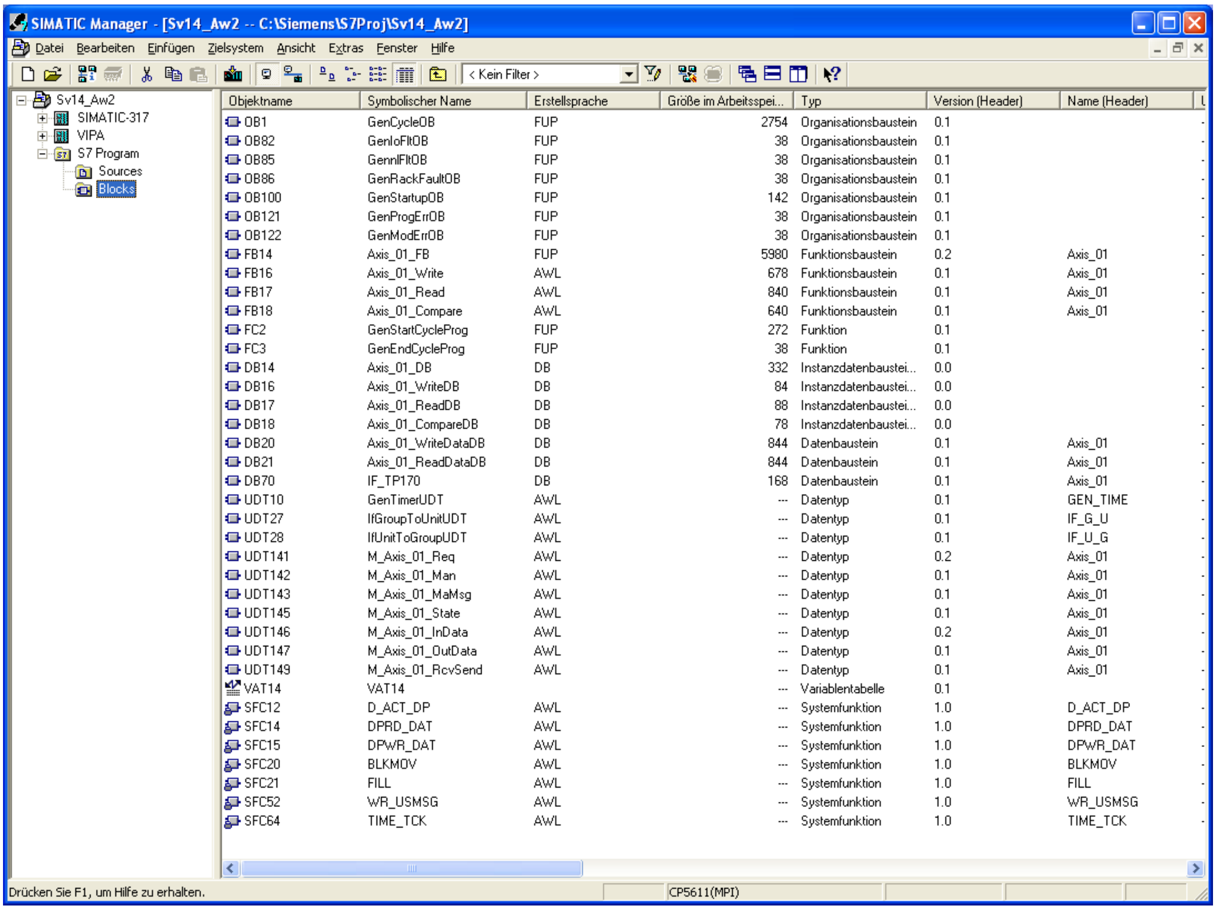
The S7 project "Sv14_Aw2" was created in English with German mnemonics and the version:



SIMATIC Manager
(c) STEP 7 S7/M7/C7
Version: V5.4 + SP3
Ausgabestand: K5.4.3.0

The S7 project “Sv14_Aw2” consists of the following components:

- Simatic-317 – contains only the hardware configuration (system data) for an S7-CPU317-2DP
- VIPA – contains only the hardware configuration (system data) for an VIPA-CPU315-SB
- S7 program – contains the modules and symbols

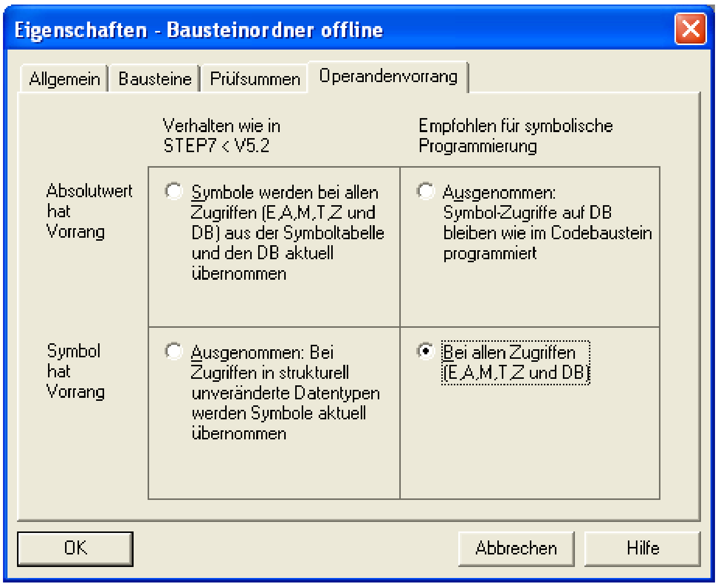


SIMATIC Manager - [Sv14_Aw2 -- C:\Siemens\S7Proj\Sv14_Aw2]

Objektname	Symbolischer Name	Erstellsprache	Größe im Arbeitsspei...	Typ	Version (Header)	Name (Header)
OB1	GenCycleOB	FUP	2754	Organisationsbaustein	0.1	
OB82	GenIoFltOB	FUP	38	Organisationsbaustein	0.1	
OB85	GenIFltOB	FUP	38	Organisationsbaustein	0.1	
OB86	GenRackFaultOB	FUP	38	Organisationsbaustein	0.1	
OB100	GenStartupOB	FUP	142	Organisationsbaustein	0.1	
OB121	GenProgErrOB	FUP	38	Organisationsbaustein	0.1	
OB122	GenModErrOB	FUP	38	Organisationsbaustein	0.1	
FB14	Axis_01_FB	FUP	5980	Funktionsbaustein	0.2	Axis_01
FB16	Axis_01_Write	AWL	678	Funktionsbaustein	0.1	Axis_01
FB17	Axis_01_Read	AWL	840	Funktionsbaustein	0.1	Axis_01
FB18	Axis_01_Compare	AWL	640	Funktionsbaustein	0.1	Axis_01
FC2	GenStartCycleProg	FUP	272	Funktion	0.1	
FC3	GenEndCycleProg	FUP	38	Funktion	0.1	
DB14	Axis_01_DB	DB	332	Instanzdatenbaustei...	0.0	
DB16	Axis_01_WriteDB	DB	84	Instanzdatenbaustei...	0.0	
DB17	Axis_01_ReadDB	DB	88	Instanzdatenbaustei...	0.0	
DB18	Axis_01_CompareDB	DB	78	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_TP170	DB	168	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_Req	AWL	---	Datentyp	0.2	Axis_01
UDT142	M_Axis_01_Man	AWL	---	Datentyp	0.1	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
UDT149	M_Axis_01_RcvSend	AWL	---	Datentyp	0.1	Axis_01
VAT14	VAT14		---	Variablentabelle	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
SFC52	WR_USMSG	AWL	---	Systemfunktion	1.0	WR_USMSG
SFC64	TIME_TCK	AWL	---	Systemfunktion	1.0	TIME_TCK

Drücken Sie F1, um Hilfe zu erhalten.

CP5611(MPI)

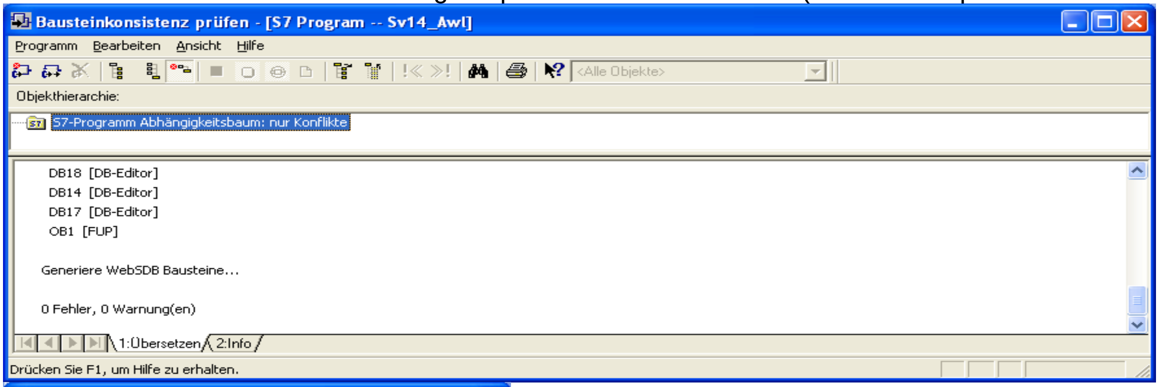


Eigenschaften - Bausteinordner offline

	Verhalten wie in STEP7 < V5.2	Empfohlen für symbolische Programmierung
Absolutwert hat Vorrang	○ Symbole werden bei allen Zugriffen (E,A,M,T,Z und DB) aus der Symboltabelle und den DB aktuell übernommen	○ Ausgenommen: Symbol-Zugriffe auf DB bleiben wie im Codebaustein programmiert
Symbol hat Vorrang	○ Ausgenommen: Bei Zugriffen in strukturell unveränderte Datentypen werden Symbole aktuell übernommen	● Bei allen Zugriffen (E,A,M,T,Z und DB)

The properties of the module folder Offline – Operand precedence are: “Symbol has precedence for all accesses (E,A,M,T,Z and DB)”. This allows changing and expanding the data structure of an FB, UDT or DB without any problems. Step7 retains the symbolic address in the S7 project and automatically changes the absolute address to match it.

The entire data structure of the project is automatically updated with “Bausteinkonsistenz prüfen” (Check module consistency) and “Alles übersetzen” (Compile all). If the message “0 Fehler, 0 Warnungen” (0 faults, 0 warnings) is subsequently not displayed, the S7 program still contains address conflicts in the modules that need to be resolved manually. This is sometimes the case for multi instances. These address conflicts are then manually removed in the modules using the KOP/AWL/FUP editor with “Datei-Zugriffe prüfen und aktualisieren” (Check and update file accesses).



Bausteinkonsistenz prüfen - [S7 Program -- Sv14_Awl]

Objekthierarchie:

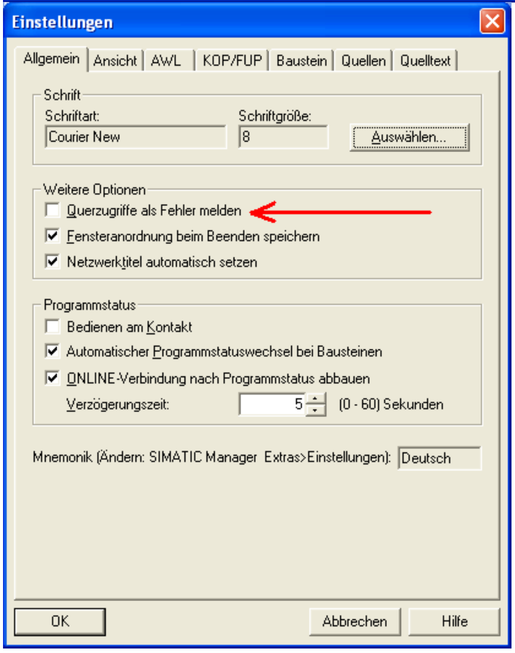
S7-Programm Abhängigkeitsbaum: nur Konflikte

```
  DB18 [DB-Editor]
  DB14 [DB-Editor]
  DB17 [DB-Editor]
  OB1 [FUP]

 Generiere WebSDB Bausteine...

 0 Fehler, 0 Warnung(en)
```

Drücken Sie F1, um Hilfe zu erhalten.



Schrift

Schriftart: Courier New

Schriftgröße: 8

Weitere Optionen

☐ Querzugriffe als Fehler melden

☑ Fensteranordnung beim Beenden speichern

☑ Netzwerktitel automatisch setzen

Programmstatus

☐ Bedienen am Kontakt

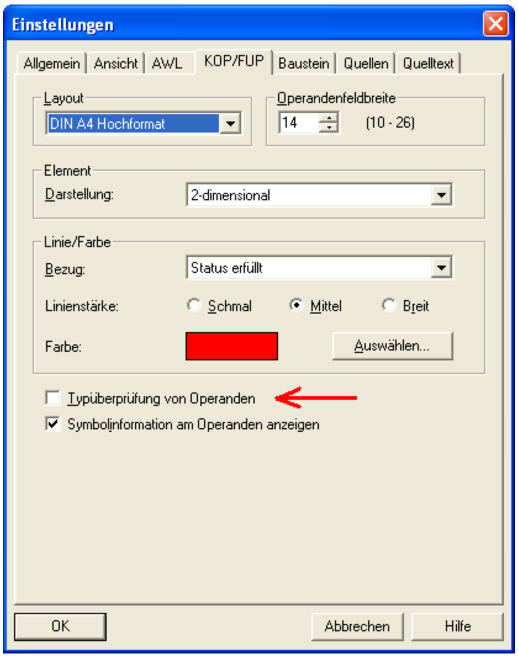
☑ Automatischer Programmstatuswechsel bei Bausteinen

☑ ONLINE-Verbindung nach Programmstatus abbauen

Verzögerungszeit: 5 (0 - 60) Sekunden

Mnemonik (Ändern: SIMATIC Manager Extras>Einstellungen): Deutsch

If accesses to the instance data of a DB continue to be displayed in RED or as a conflict, the following must be deactivated in the KOP/AWL/FUP editor: EXTRAS\Settings\General “Querzugriffe als Fehler melden” (Report cross access as errors). This is also the default setting of the S7 manager. Afterwards, the program must be compiled again.



Layout: DIN A4 Hochformat

Operandenfeldbreite: 14 (10 - 26)

Element

Darstellung: 2-dimensional

Linie/Farbe

Bezug: Status erfüllt

Linienstärke: ○ Schmal ● Mittel ○ Breit

Farbe: 🟥

☐ Typüberprüfung von Operanden

☑ Symbolinformation am Operanden anzeigen

For the complete FUP representation of the networks, “Typüberprüfung von Operanden” (Type check of operands) must be deactivated in the KOP/AWL/FUP editor: EXTRAS\Settings\Kop/FUP.

Module overview

FB14/DB14	Handling module between S7 PLC and SERVOSTAR
FB16/DB16	Module for the FB14 to write data to SERVOSTAR
DB20	Contains the data that are written to SERVOSTAR with FB16
FB17/DB17	Module for the FB14 to read data from SERVOSTAR
DB21	Contains the data that are read from SERVOSTAR with FB17
FB18/DB18	Module for data comparison, e.g. DB20 and DB21
DB70	Interface data module for an operator panel / touch panel

All modules can be renamed or renumbered upon demand.
Afterwards, “Bausteinkonsistenz prüfen” (Check module consistency) must be performed with “Alles übersetzen” (Compile all).
The resulting conflicts may have to be corrected manually in the modules.

The FB14 uses the following S7 system modules:
SFC12 – D_ACT_DP – Deactivating and activating a DP standard slave
SFC14 – DPRD_DAT – Reading consistent data of a a DP standard slave
SFC15 – DPWR_DAT – Writing consistent data of a a DP standard slave
SFC20 – BLKMOV – Copy memory area
SFC21 – FILL – Prefill memory area

The following S7 system modules are used in FC2:
SFC52 – WR_USMSG – Writing user-defined diagnostics event to the S7 PLC diagnostics buffer
SFC64 – TIME_TCK – Reading system time (used to implement the timers in the program)

Notes

- FB14 is programmed as multi instance and, therefore, can be called multiple times in an FB as subprogram module without separate instance data module.

- If FB14 is not implemented as multi instance, then FB14 must be called up for every drive with a separate instance data module.

- FB14 is programmed in AWL and FUP (function chart) with GERMAN mnemonics and symbols with comments in ENGLISH.

- The UDT data type (UserDefinedDatatype) enables a uniform database and an object-oriented programming in Step7.

- The STRUCT data type enables simple addressing of entire data areas via pointer (P#DB.DBX byte).

- The S7 project “Sv14_Aw2” contains a completely functional PLC program for SERVOSTAR.

With a S7 CPU and a SERVOSTAR with Profibus card and motor, the complete servo drive can be easily and quickly commissioned using the VAT14 (table of variables).
The DB70 can also be used simply as interface data module for an operator/touch panel/visualization.

Input and output interface of FB14 - Axis_01_FB

The interface of FB14 uses UDT (user-defined data types).

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

This UDT is also used in the DB 70 - IF_TP170 - Interface DB Touch Panel TP170. This ensures that the data structure in the S7 project has a uniform common database. A modificatyion to the data structure is done once in the UDT. The entire data structure of the S7 project is automatically updated with “Bausteinkonsistenz prüfen” (Check module consistency) and “Alles übersetzen” (Compile all). Time stamp conflicts are also removed.

Input variables are preceded by an “i” (input) and output variables by an “o” (output).
This makes the program code easirer to read.

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	

Complete overview of interface from 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 * 2^15 / 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 command ERRCODE
VoltageInhibit	BOOL	
FastStop	BOOL	
SwitchOnInhibit	BOOL	
Warning	BOOL	see ASCII command 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

1*) The current positon is transferred in the process data only in units internal to the controller (2^{20} incr. per motor revolution). The conversion to user units can be performed in the PLC, depending on the position resolution defined in the controller bia the operating software.
Example: Resolution = 5000 incr / 3 revolutions
=> Position in user units = Actual_Position x 5000 / (3 x 2^{20})

Program examples for converting the controller-internal units to SI units

OB1 - network 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 - network 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 - network 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
```

Commissioning FB14 Axis_01_FB

S7 PLC - Switching ON/OFF
If the S7 PLC Profibus master fails, the watchdog timeout starts in the SERVOSTAR.
After startup, the S7 PLC performs a software reset (GenStartUp - M1.2) and deletes all error messages and states in the PLC program.

S7 PLC - Switching Start/Stop
performs a software reset (GenStartUp - M1.2) and deletes all error messages and states in the PLC program. The SERVOSTAR remains enabled and an active travel order continues to be executed.

iAck =1
resets warnings and errors in the SERVOSTAR and in the modules FB14, FB16, FB17, FB18.

Initialization
During initialization, a zero telegram is send in the PIV and PZD. The axis is disabled.

- Set iInit =1 (a pulse is generated internally)
- Wait until oState.InitOk =1
  If oState.InitOk is not =1 or oState.InitError =1, then the initialization was not successful. For error cause, see SERVOSTAR receive telegram.
- The initialization is monitored with a timeout of 1 sec.

Set operating mode (programmed operating modes Positioning, Digital speed, Digital torque)
After switch-on, the SERVOSTAR is always in the safe operating mode -126

- Write the desired operating mode in iData.OpMode
  (see manual: Positioning: 2 ; Digital speed: 1 ; Digital torque: -2)
- Set iSetOpModePb =1 (a pulse is generated internally)
- Wait until oState.OpModeOk = 1
  The activated operating mode is displayed in oOpmodePb
  If oState.OpModePbOk is not =1 or oState.OpModePbError =1, then setting the operating mode was not successful. For error cause, see SERVOSTAR receive telegram.
- At thispoint, oActualPosition, oActualSpeed and oManufactState are also displayed with values.
- Setting the operating mode is monitored with a timeout of 1 sec.

Enable operation
After switch-on, the SERVOSTAR is in the state “Switch-On inhibited”.
iFastStop and iFastStopDisableAxis must be =0 and iSwEnable must be =1.
At the SERVOSTAR, HardwareEnable must be =1 and, if present, AS-Enable must be =1.
The DC link voltage is applied and the display of SERVOSTAR shows “Pxx”

- Set iSetOperationEnabled =1 (a pulse is generated internally)
- Wait until oState.P4_OperationEnabled =1
  If oState.P4_OperationEnabled is not =1, the setting the operating mode was not successful. For error cause, see SERVOSTAR receive telegram.

Positioning operating mode

The Positioning operating mode (oState.OpModeOk = 1 and oOpmodePb =2 dec) is enabled.

Jogging mode

Requirement: The drive is in the state “Operation enabled” and hardware enable is present and fault-free and the display of SERVOSTAR shows “Exx”.

- Write the desired jogging speed to “iData.Config.JogSpeed”
- Set iRequest.JogPlus =1 -> The drive moves in the positive direction
- Set iRequest.JogMinus =1 -> The drive moves in the negative direction oState.JoggingActive is =1 if the drive moves.

Referencing run

The type of referencing must be set with the DRIVEGUI / DRIVE operating software.
Prerequisite: The drive is in the state “Operation enabled” and hardware enable is present and fault-free.

- Write the desired referencing run speed to “iData.Config.RefSpeed”
- Set iRequest.StartRef =1 -> The drive moves and oState.ReferencingActive =1
- Wait until oState.ReferenceOk =1 and oState.ReferencingActive =0 again
- Set iStartRef = 0 – The drive has been referenced

With iEnTimeout =1 and iData.Config.TO_Reference xxx ms [TIME], the referencing run is monitored with a timeout.

Start of an EEPROM or RAM travel order

The DRIVEGUI / DRIVE operating software is used to create travel orders.
Only the parameters of the RAM travel orders can be changed in the drive oin the state “Operation enabled” (see ASCII command MTMUX)
Requirement: The drive is in the state “Operation enabled” and referenced.

- Write the number of the drive order to be started to “iData.MotionTask.Number"
- Set iStartMotionTask =1 (a pulse is generated internally)
- Wait until oState.InPosition =0 and oStateMotionTaskActive =1
- Wait until oState.InPosition =1 and oStateMotionTaskActive =0

The drive executed the travel order.

With iEnTimeout =1 and iData.Config.TO_Position xxx ms [TIME], the travel order is monitored with a timeout.

Start of a direct travel order (has the number 0)

Requirement: The drive is in the state “Operation enabled” and referenced.

- Write 0 to iData.MotionTask.Number
- Write target speed to iData.MotionTask.DirectSpeed
- Write target position to iData.MotionTask.DirectPosition
- Write travel order type to iData.MotionTask.DirectType
- Set iRequest.StartMotionTask =1 (a pulse is generated internally)
- Wait until oState.InPosition =0 and oStateMotionTaskActive =1
- Wait until oState.InPosition =1 and oStateMotionTaskActive =0

The drive executed the travel order.
Note: With a pointer (P#DB.DBX byte), the iData.MotionTask can be written to easily with an SFC20-BLKMOV.

With iEnTimeout =1 and iData.Config.TO_Position xxx ms [TIME], the travel order is monitored with a timeout.

See Switch ON Timing Diagram and Start MotionTask

While the MotionTask is running, the next MotionTask can already be copied to iData.MotionTask with its record and then started immediately with Set iRequest.StartMotionTask =1 (pulse).

A started MotionTask can be paused with iPauseMotionTask =1.

A started MotionTask can be stopped with iRequest.StopMotionTask =1.

A stopped MotionTask remains active in the controller, i.e. if the controller is switched back to oState.P4_OperationEnabled, the MotionTask is continued.
With iSwEnable =0, the stopped MotionTask is killed, i.e. if the controller is switched back to oState.P4_OperationEnabled, the MotionTask is no longer continued.
See: Kill MotionTask timing diagram

Notice: The INPT parameter can be used to set the time in ms in the terminal window of the operating software (or with the PNU1904) for which the signal oManufactState.InPositionWindow is reset at a minimum after the motion task start (see manual ch. VII.1).

Note about positions and speeds: Target position and target speed can be predefined in the units preset in the position controller if bit 13 is set in the travel order type PNU1785 (see manual ch. IV.2.5.3).

Travel order types (PNU 1785 - for frequent values see manual ch. IV 2.5.3)
0x2000 (bit 13 set) Absolute positioning with preset of speeds and positions in user units and trapezoidal travel profile.
0x2003 (bits 0, 1, 13 set) Positioning “Relative target” with preset of speeds and positons in user units.
0x2000 (bit 13 and bit 16 set) Absolute positioning with preset of speeds and positions in user units and sine^2 travel profile.
Bit 16 cannot be set directly in the PZD channel.
See: Activating the trajectory/profile acceleration Sin^2 for direct motion task no. 0

Digital speed operating mode

The digital speed operating mode (oState.OpModeOk = 1 and oOpmodePb =1 dec) is activated and operation is enabled.

Write target speed to iData.DigitalSpeed.Ncmd (16 bit)
Set iRequest.StartNcmd =1
The drive moves with the preset target speed.
The target speed can be changed at any time.
Set iRequest.StartNcmd =0
The drive decelerates via the set speed ramps until standstill.

Actual position, actual speed (16 bit) and the manufacturer-specific status register HSW are cyclically transmitted in the PZD channel.

The speed value (16 bit) is calculated using the following formula:

```
PNU1886 – Ncommand ( Ncmd16 = Ncmd * 2^15 / VOSPD )
```

Digital torque operating mode

The digital torque operating mode (oState.OpModeOk = 1 and oOpmodePb =-2 dec) is activated and operation is enabled.

Write target current to iData.DigitalTorque.Icmd
Set iRequest.StartIcmd =1
The drive injects the preset target current.
The target current can be changed at any time.
Set iRequest.StartIcmd =0 and the drive outputs the target current 0 ampere.

Actual position, actual current and the manufacturer-specific status register HSW are cyclically transmitted in the PZD channel.

The current value is calculated using the following formula:

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

The ICMDVLIM parameter (PNU 1989) can be used to limit the speed of the motor to a maximum value. This prevents the motor from overspeeding of the load is too low.

These 3 operating modes allow implementing most of the requirements in a machine.
Additional operating modes are currently not implemented in FB14, but they can easily be programmed at a later time.

Writing and reading parameters:

All parameters and commands of SERVOSTAR® can be addressed via the Profibus PNU number. In the ASCII list of commands, the numbers can be listed and displayed sequentially. In addition, the number is located in the "Profibus PNU" field for the description of the commands/parameters. Furthermore, the manual features a list of selected parameter numbers.

Writing parameters

Requirement: PIV parameter channel is not yet in use

- Write number to iData.PnuWrite.Number
- Write index to iData.PnuWrite.Index
- Write value to iData.PnuWrite.Value
- Set iPnuWriteStart =1 (pulse)
- Wait until oState.PnuWriteOK
  If oState.PnuWriteOk is not =1 or oState.PnuWriteError =1, the writing the parameter was not successful. For error cause, see SERVOSTAR receive telegram.

Reading parameters

Requirement: PIV parameter channel is not yet in use

- Write number to iData.PnuRead.Number
- Write index to iData.PnuRead.Index
- Write value to iData.PnuWrite.Value
- Set iPnuReadStart =1
- Wait until oState.PnuReadOK
  If oState.PnuReadOk is not =1 or oState.PnuReadError =1, the reading the parameter was not successful. For error cause, see SERVOSTAR receive telegram.
- The data read are output in
  oData.PnuRead.Number
  oData.PnuRead.Index
  oData.PnuRead.Value
  This allows comparing whether the data read are actually the requested data.

Reading parameters may be permanently activated, e.g. for an actual value to be monitored (e.g. PNU 1688 r.m.s. current for determining the torque).

Additional Profibus functions

With iActSlave =1, the DP slave SERVOSTAR is activated in FB14 with SFC12 – D_ACT_DP with running PLC or deactivated with iActSlave =0.
In addition, the Profibus communication is monitored for

- configuration errors
- slave failure
- and telegram errors

The entire data telegrams are output in oAxis [STRUCT]
The SERVOSTAR state machine is output in oState [STRUCT]

With relative positioning, the data can be stored in oData.Canceled [STRUCT] if a travel order is canceled. At a later time (e.g. after EMERGENCY OFF – manual access - door OPEN / door CLOSED), the S7 PLC can use it to correct motion task data and finish the relative travel order with corrected data.

With the VAT14 table of variables, the modules in SERVOSTAR can be controlled and monitored.

Example for a sequencer “Initializing and enabling 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
```

See Switch ON timing diagram and Start MotionTask

Example for a sequencer “Start travel sets in 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
```

See Switch ON Timing Diagram and Start MotionTask

Modules for SERVOSTAR parameters

The S7 project "Sv14_Aw2" contains even more function locks and data blocks that write complete data areas to the SERVOSTAR, read them from the SERVOSTAR amnd compare them in the S7 PLC.

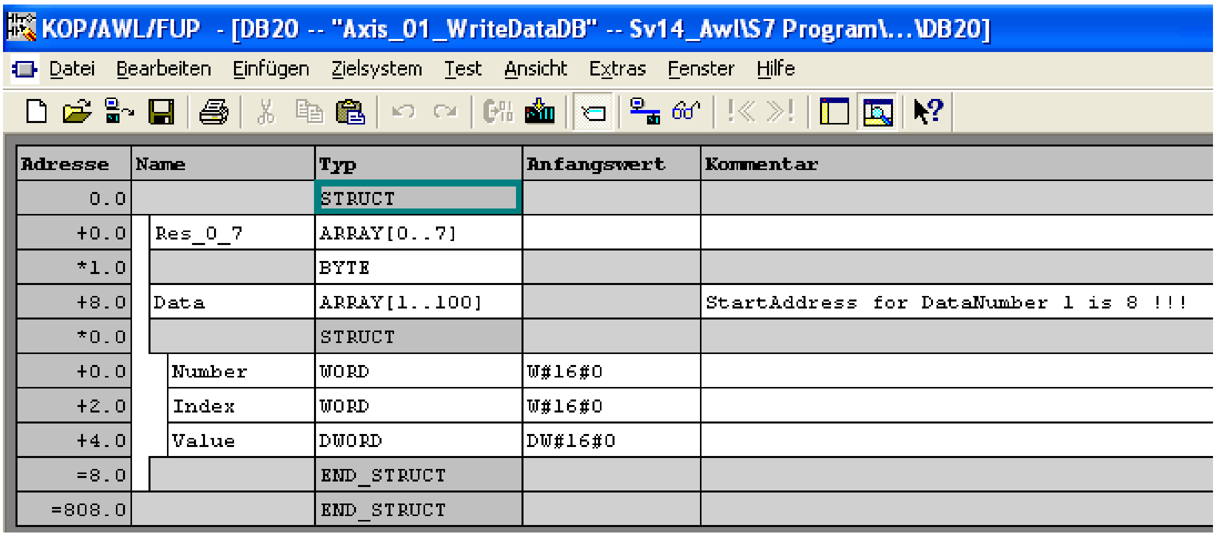
A record consists of 3 parameters:

- PNU - parameter number – indicates the respective parameter
  e.g PNU 1783 startup time O_ACC1.
- Index indicates what the transmitted value is about
  e.g Index=1 - actual value or Index=3 upper limit.
- Value – contains the transmitted value.

DB 20 - Axis_01_WriteDataDB

Contains the data for writing with FB16 Axis_01_Write and FB14 Axis_01 from the S7 PLC over the non-real-time-capable PIV parameter channel in Profibus to the SERVOSTAR.

Declaration from DB20 – for 100 values

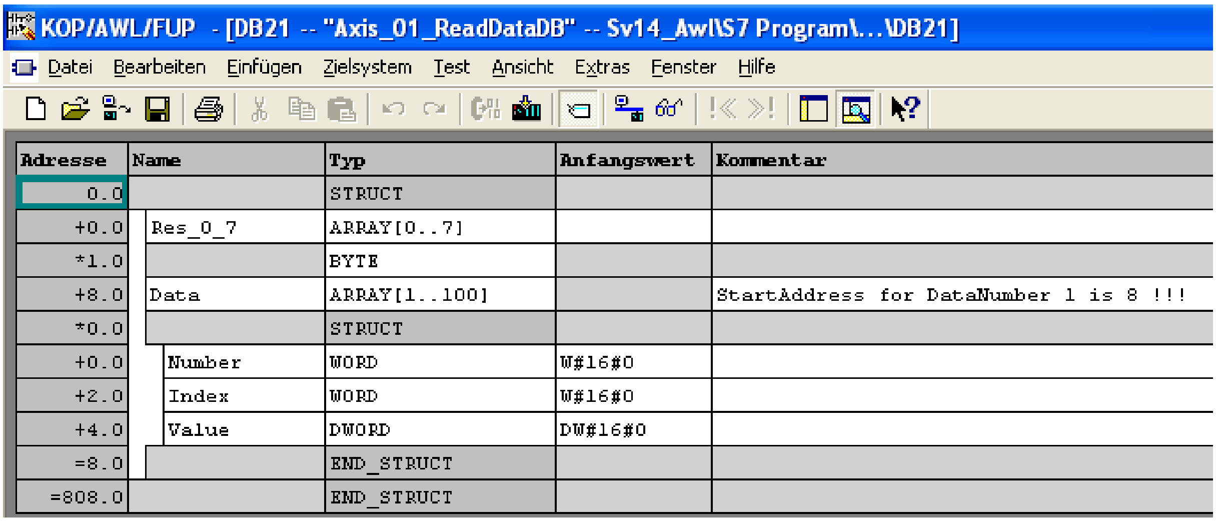

KOP/AWL/FUP - [DB20 -- "Axis_01_WriteDataDB" -- Sv14_Awl\S7 Program\...\DB20]
Datei Bearbeiten Einfügen Zielsystem Test Ansicht Extras Fenster Hilfe

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

Contains the data for reading with FB17 Axis_01_Read and FB14 Axis_01 from the SERVOSTAR over the non-real-time-capable PIV parameter channel in Profibus to the S7 PLC.

Declaration from DB21 – for 100 values


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

Controls FB14 to write datafrom the PLC to the SERVOSTAR

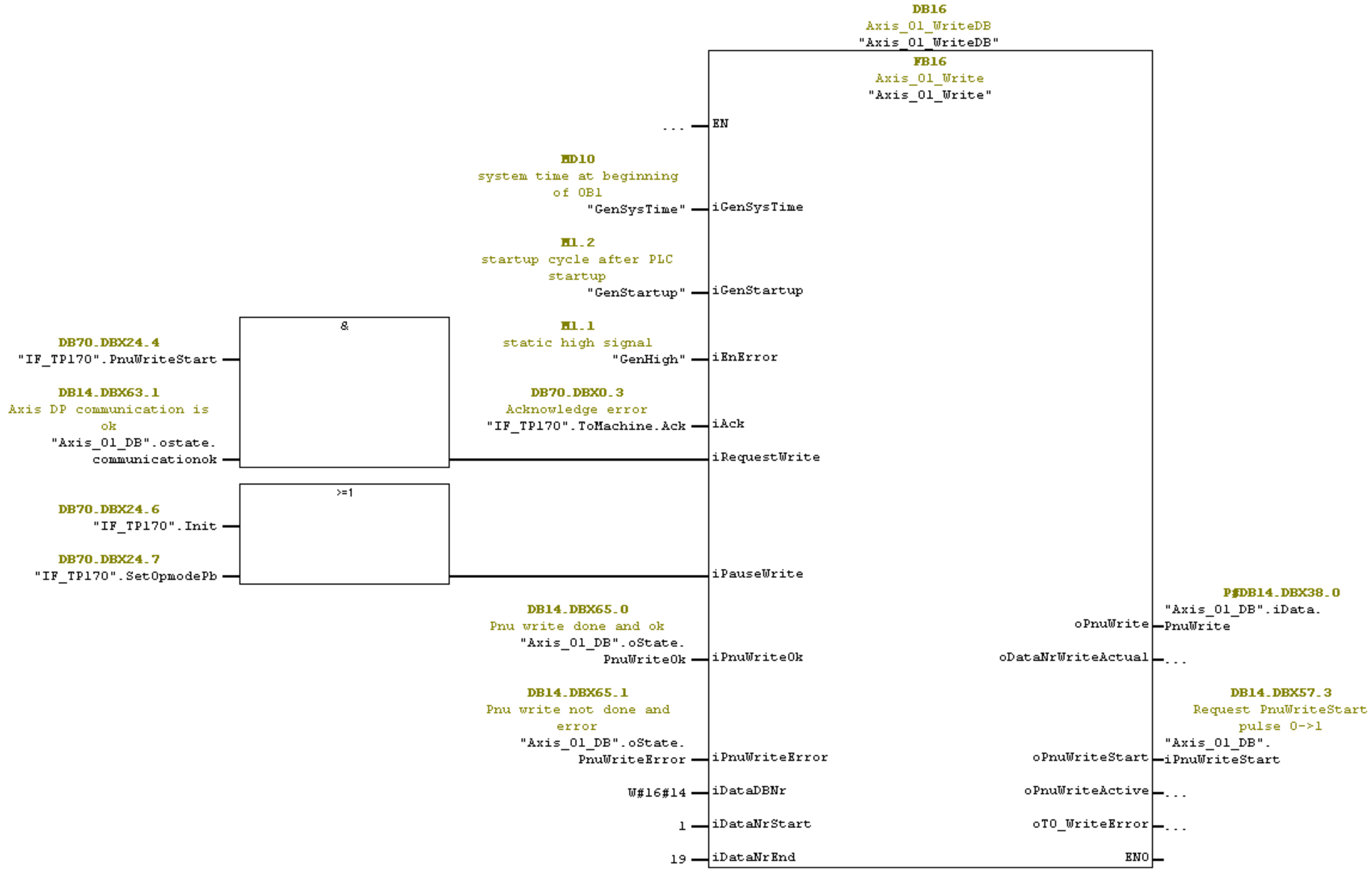
Input and output interface of FB16 - Axis_01_Write

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]



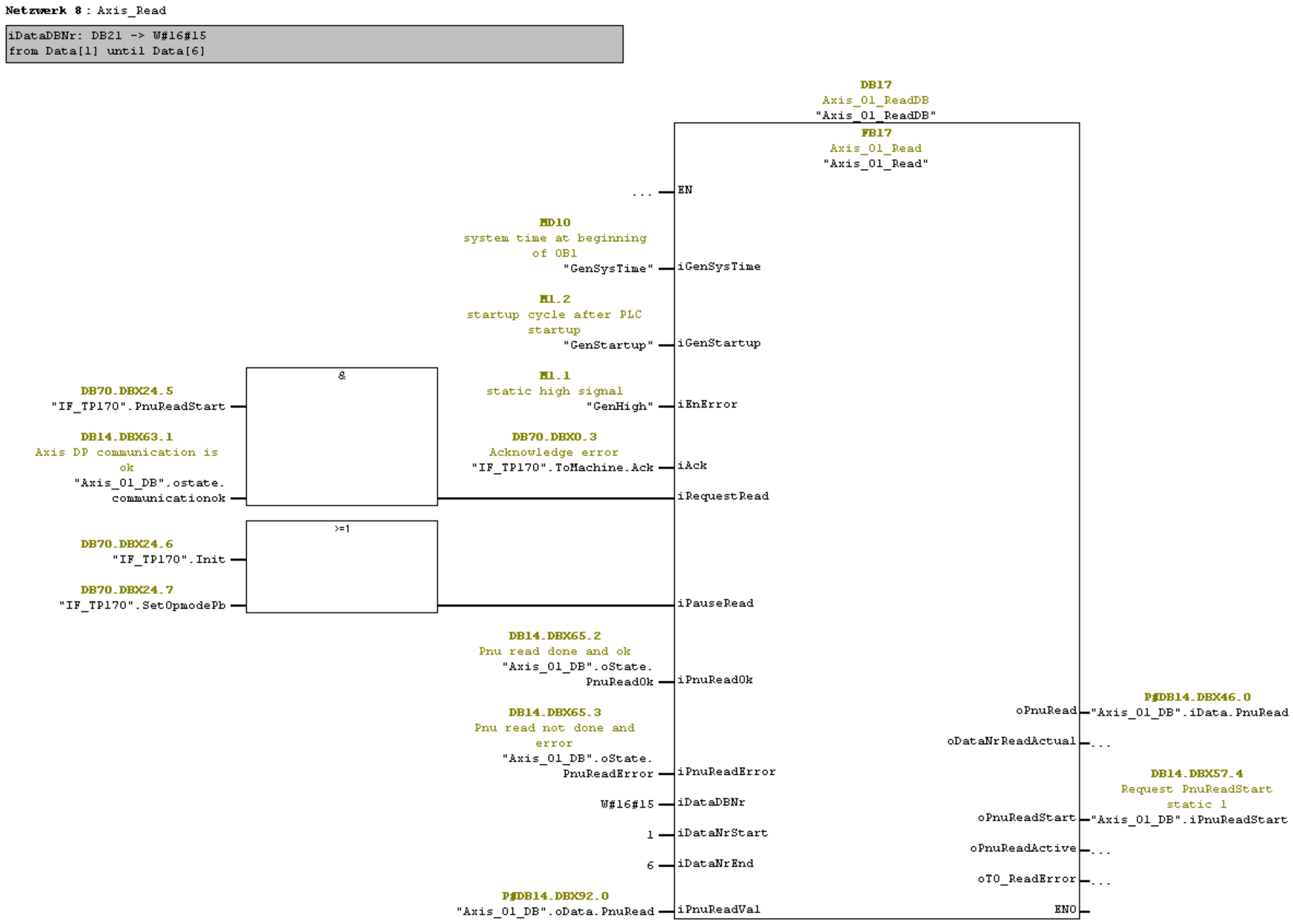
FB17 Axis_01_Read

Controls FB14 to read data from the SERVOSTAR to the PLC

Input and output interface of 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]

Notice:

With simultaneous Read and Write request at FB14, FB14 generates the error `oMaMsg.ErrReadWrite`

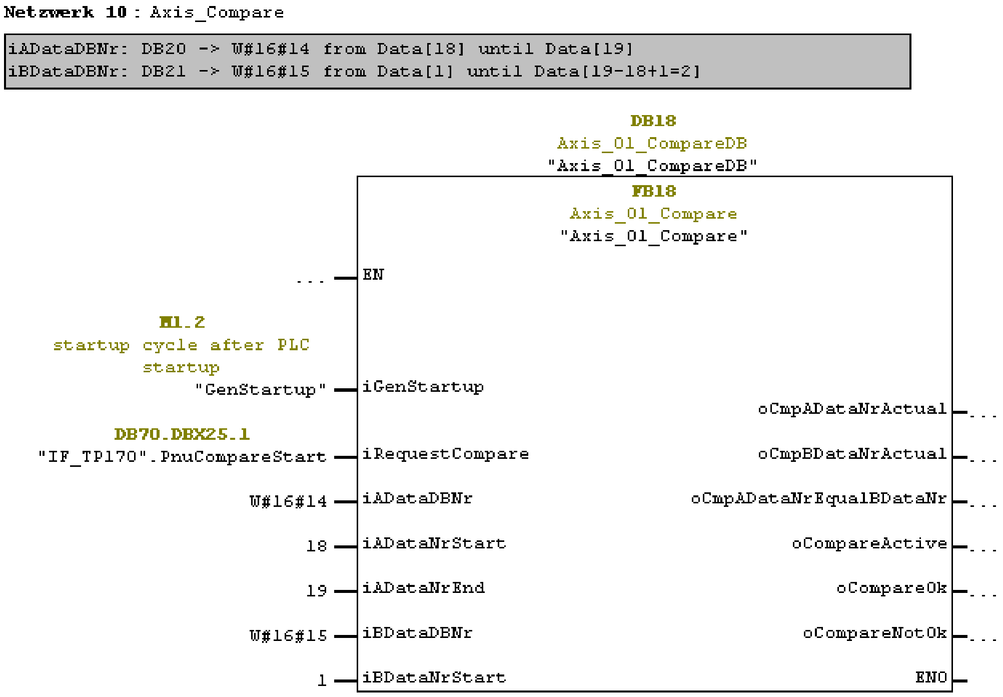
FB18 Axis_01_CompareDB

Compares a data area from the written data from DB20 Axis_01_WriteDataDB with the read data from DB21 Axis_01_ReadDataDB.
Only one record is compared for each PLC cycle.

Input and output interface of FB18 Axis_01_CompareDB

VAR_INPUT		
iGenStartup	BOOL	Startup cycle after PLC startup
iRequestCompare	BOOL	Request compare
iADataDBNr	WORD	ADataDBNumber
iADataNrStart	INT	ADataNumberStart - first number from data to compare
iADataNrEnd	INT	ADataNumberEnd - last number from data to compare
iBDataDBNr	WORD	BDataDBNumber
iBDataNrStart	INT	BDataNumberStart - first number from data to compare

VAR_OUTPUT		
oCmpADataNrActual	WORD	Compare A Data number is actual
oCmpBDataNrActual	WORD	Compare B Data number is actual
oCmpADataNrEqualBDataNr	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

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

Tips and information:

Error messages from modules FB14, FB16 and FB17

Must be implemented in the program so that the PLC program can respond to them.

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 via Profibus

See ASCII list OVERRIDE
The Override function allows controlling the speed of a travel set, the reference travel and the jogging mode.
OVRIDE=0 Override function deactivated
OVRIDE=3 Profibus interface activated for the Override function.

See ASCII list DOVRIDE
When activating a digital OVERRIDE function, this parameter is used to predefine the digital override factor. The following scaling applies:
DOVRIDE=0 motion task speed = 0
DOVRIDE=8192 motion task speed = 100 %

The OVERRIDE function is not possible for sin^2 ramps!

Activating the trajectory/profile acceleration Sin^2 for direct motion task no. 0

The travel profile Sin^2 enables a smooth, jerk-free accelerating and decelerating. This protects the mechanical system (gears, spindles). Pendulum movements etc. are suppressed.

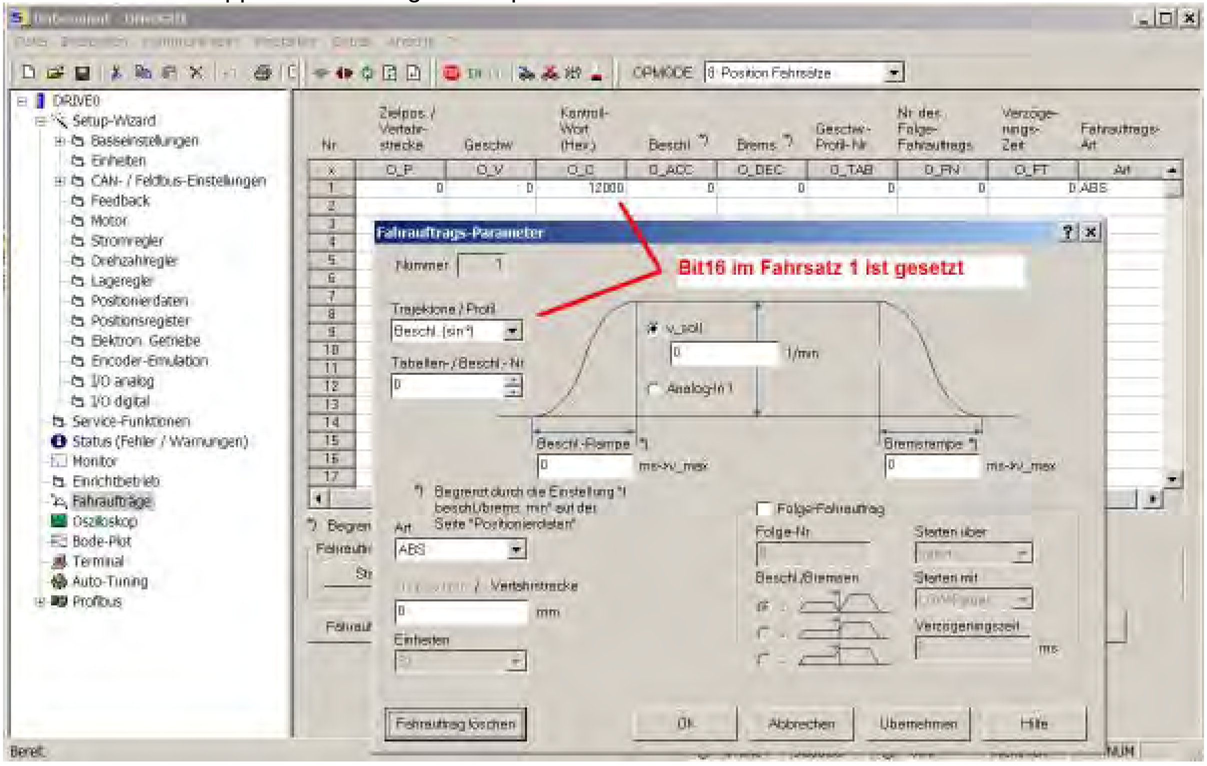
For SERVOSTAR S400/S600, see ASCII list, it is not described here in more detail.

For SERVOSTAR S300/S700 – Profibus see ASCII list O_C
The command O_C (Profibus PNU 1785) defines the travel order type for the local travel set No. 0 (direct travel set).
If bit 16 is set by O_C, it starts a table motion task (sine2 travel). Bit 9 must be set to 0. A bit variable is transmitted as parameter (16 bits).
Bits 0 to 15 are directly addressed in direct motion task 0 (PZD5), bit 16 must be set separately in S300/S700.
The O_C command is not a parameter and, therefore, is not stored non-volatile in the controller. Bit 16 can be set in two ways:

1.) Use the PIV parameterchannel in PNU 1785 to write the value 10000hex to S300/S700

2.) After switchin ON the 24-VDC supply voltage, the S300/S700 automatically copies bit 16 from an EEPROM travel set, e.g. no.1, to the direct motion task no. 0.
This allows the S7 program, to remain unchanged.

Activating bit 16 in the EEPROM travel set, e.g. no.1
In addition, the acceleration ramp O_ACC and deceleration ramp O_DEC can also be set. For value 0, the value PTMIN applies or the largest ramp in each case.



OPMODE 8 Position Fahrsätze

Nr	Zielpos. / Verfahr-strecke	Geschw	Kontroll-Wort (Hex)	Beschl *)	Brems *)	Geschw-Profil-Nr	Nr der Folge-Fahrauftrags	Verzöge-rungs-Zeit	Fahrauftrags-Art
x	O_P	O_V	O_C	O_ACC	O_DEC	O_TAB	O_FN	O_FT	Art
1	0	0	12000	0	0	0	0	0	ABS

Fahrauftrags-Parameter

Nummer 1

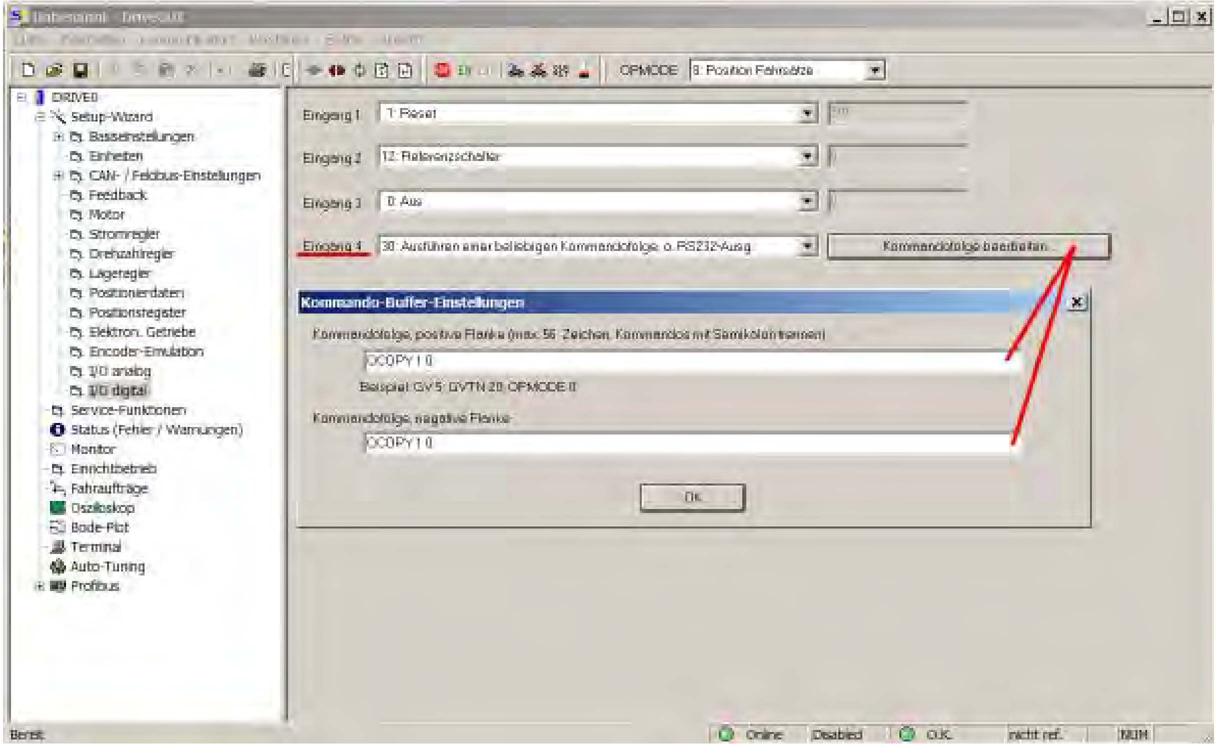
Bit16 im Fahrsatz 1 ist gesetzt

Trajektorie / Profil: Beschl. (sin²)
Tabellen- / Beschl.-Nr: 0
● v_soll: 0 1/min
○ Analog-In 1
Beschl.-Rampe *): 0 ms->v_max
Bremsrampe *): 0 ms->v_max

*) Begrenzt durch die Einstellung "t beschl/brems. min" auf der Seite "Positionierdaten"

Art: ABS
Verfahrstrecke: 0 mm
Einheiten
☐ Folge-Fahrauftrag
Folge-Nr
Beschl./Bremsen
Starten über
Starten mit
Verzögerungszeit ms

An unused input (not wired) is assigned the OCOPY function from EEPROM motion task no.1 to direct motion task no. 0



OPMODE: 8: Position Fahrsätze

Eingang 1: 1: Reset

Eingang 2: 12: Referenzschalter

Eingang 3: 0: Aus

Eingang 4: 30: Ausführen einer beliebigen Kommandofolge, o. RS232-Ausg.

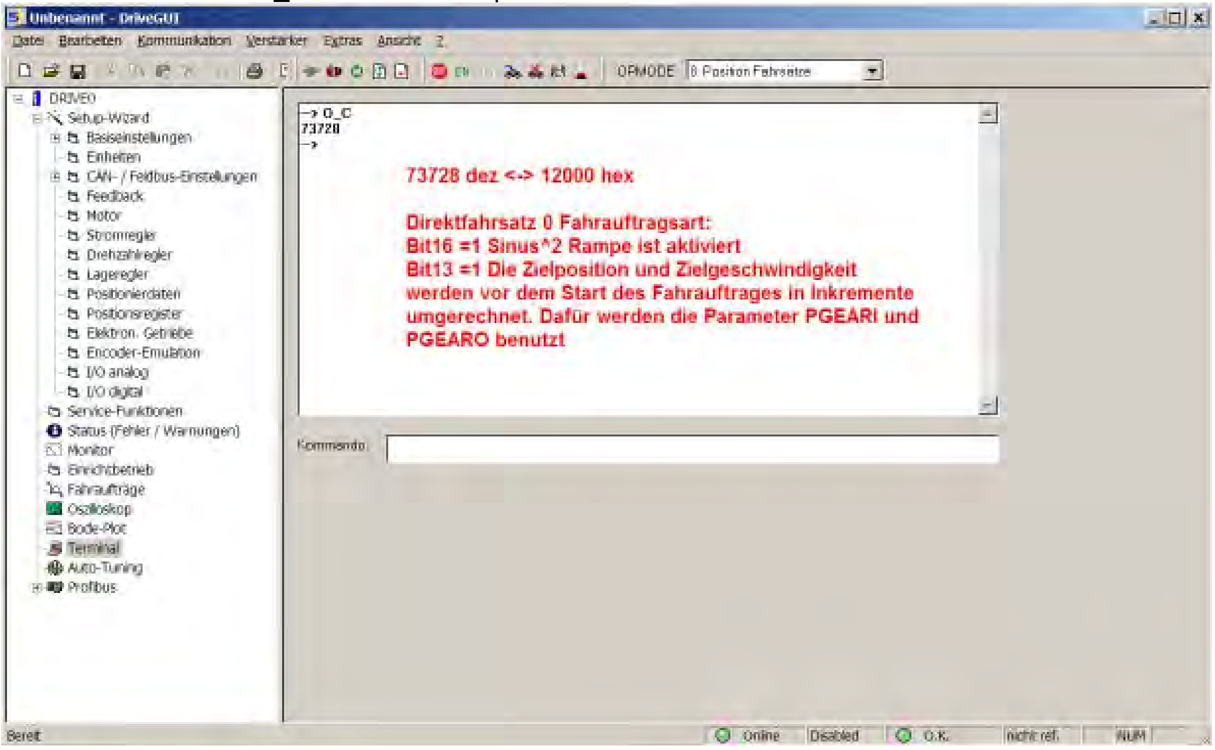
Kommando-Buffer-Einstellungen

Kommandofolge, positive Flanke (max. 56 Zeichen, Kommandos mit Semikolon trennen): OCOPY 1 0

Beispiel: GV 5; GVTN 20; OPMODE 0

Kommandofolge, negative Flanke: OCOPY 1 0

Check the content of O_C with terminal input



OPMODE: 8: Position Fahrsätze

```
-> O_C
73728
->
```

73728 dez <-> 12000 hex

Direktfahrsatz 0 Fahrauftragsart:
Bit16 =1 Sinus^2 Rampe ist aktiviert
Bit13 =1 Die Zielposition und Zielgeschwindigkeit werden vor dem Start des Fahrauftrages in Inkremente umgerechnet. Dafür werden die Parameter PGEARI und PGEARO benutzt

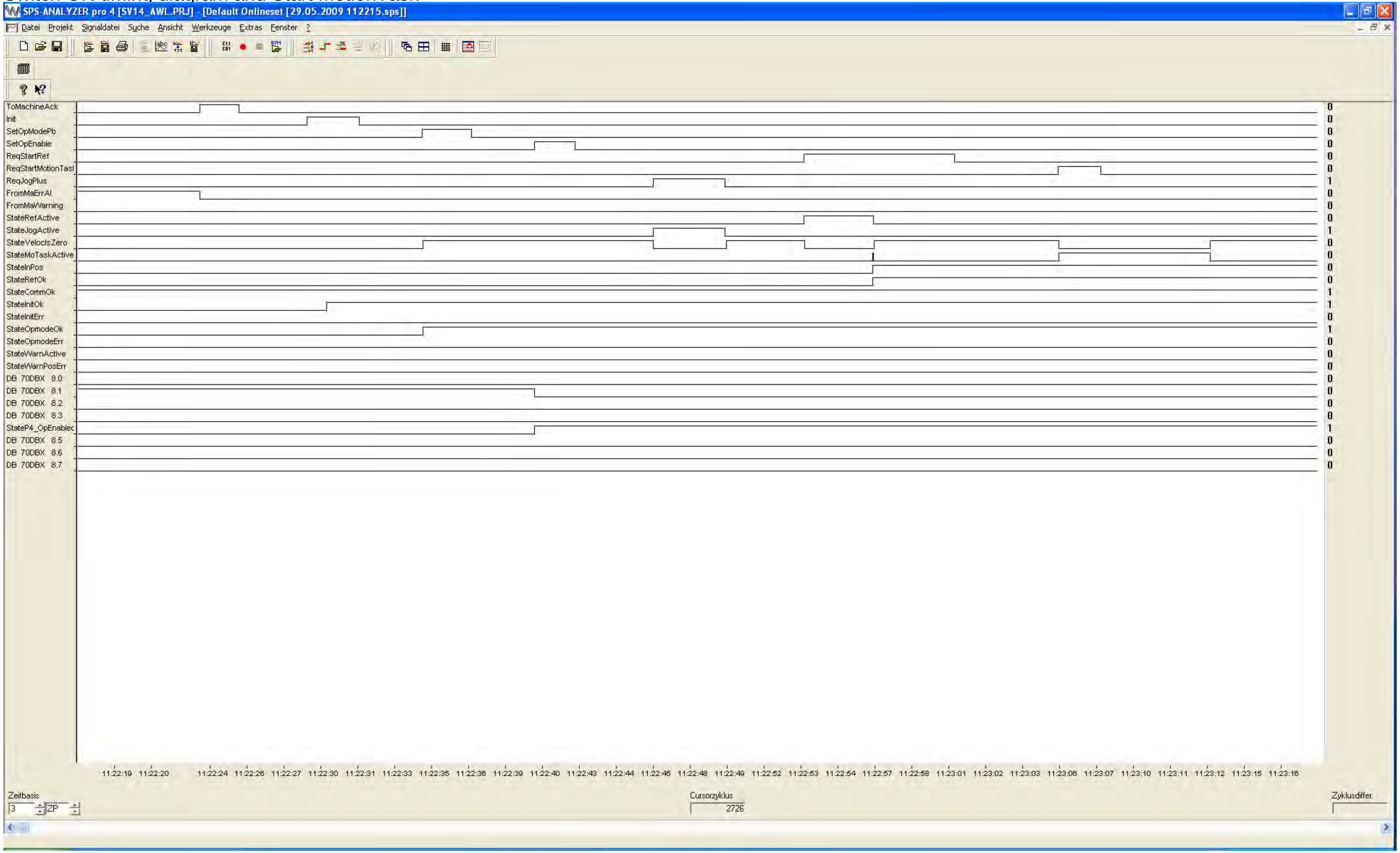
Bit coding of error PNU 1001 - ERRCODE

Axis_PNU1001	MD 1000	DWORD	Faults
Axis_PNU1001_Bit31_F32	M 1000.7	BOOL	System error
Axis_PNU1001_Bit30_F31	M 1000.6	BOOL	Error SafetyCard
Axis_PNU1001_Bit29_F30	M 1000.5	BOOL	Error Emergency stop timeout
Axis_PNU1001_Bit28_F29	M 1000.4	BOOL	Slot card error
Axis_PNU1001_Bit27_F28	M 1000.3	BOOL	EtherCAT synchronization error
Axis_PNU1001_Bit26_F27	M 1000.2	BOOL	STO error
Axis_PNU1001_Bit25_F26	M 1000.1	BOOL	Error during reference travel – hardware limit switch
Axis_PNU1001_Bit24_F25	M 1000.0	BOOL	Commutating error
Axis_PNU1001_Bit23_F24	M 1001.7	BOOL	Warning error changed to error
Axis_PNU1001_Bit22_F23	M 1001.6	BOOL	Error in CAN communication
Axis_PNU1001_Bit21_F22	M 1001.5	BOOL	Error reserved
Axis_PNU1001_Bit20_F21	M 1001.4	BOOL	Error handling error
Axis_PNU1001_Bit19_F20	M 1001.3	BOOL	Error slot error
Axis_PNU1001_Bit18_F19	M 1001.2	BOOL	Error collapse in DC link voltage
Axis_PNU1001_Bit17_F18	M 1001.1	BOOL	Ballast error (defective ballast transistor)
Axis_PNU1001_Bit16_F17	M 1001.0	BOOL	Error A/D converter
Axis_PNU1001_Bit15_F16	M 1002.7	BOOL	Error network BTB
Axis_PNU1001_Bit14_F15	M 1002.6	BOOL	Fehler I2tmax exceeded
Axis_PNU1001_Bit13_F14	M 1002.5	BOOL	Fehler output stage: Ground fault, short circuit or ballast short circuit
Axis_PNU1001_Bit12_F13	M 1002.4	BOOL	Error ambient temperatur
Axis_PNU1001_Bit11_F12	M 1002.3	BOOL	Error reserved
Axis_PNU1001_Bit10_F11	M 1002.2	BOOL	Error brake
Axis_PNU1001_Bit9_F10	M 1002.1	BOOL	Cable break ROD interface or timing problem of master slave controller booting
Axis_PNU1001_Bit8_F09	M 1002.0	BOOL	Error EEPROM
Axis_PNU1001_Bit7_F08	M 1003.7	BOOL	Error overspeed
Axis_PNU1001_Bit6_F07	M 1003.6	BOOL	Error internal supply voltages
Axis_PNU1001_Bit5_F06	M 1003.5	BOOL	Error motor temperature
Axis_PNU1001_Bit4_F05	M 1003.4	BOOL	Error undervoltage
Axis_PNU1001_Bit3_F04	M 1003.3	BOOL	Feedback error
Axis_PNU1001_Bit2_F03	M 1003.2	BOOL	Contouring error when executing external trajectory
Axis_PNU1001_Bit1_F02	M 1003.1	BOOL	Error overvoltage
Axis_PNU1001_Bit0_F01	M 1003.0	BOOL	Error heat sink temperature

Bit coding of warnings and manufacturer-specific status registers – PNU 1002 - DRVSTAT

Axis_PNU1002	MD 1004	DWORD	Warnings AND manufacturer-specific status register
Axis_PNU1002_Bit31	M 1004.7	BOOL	Error present
Axis_PNU1002_Bit30	M 1004.6	BOOL	Output stage enabled
Axis_PNU1002_Bit29	M 1004.5	BOOL	Safety relay tripped (STO)
Axis_PNU1002_Bit28	M 1004.4	BOOL	Speed = 0
Axis_PNU1002_Bit27	M 1004.3	BOOL	-
Axis_PNU1002_Bit26	M 1004.2	BOOL	Initialization complete (internal initialization of amplifier finished)
Axis_PNU1002_Bit25	M 1004.1	BOOL	Position 4 reached (see above)
Axis_PNU1002_Bit24	M 1004.0	BOOL	Position 3 reached (see above)
Axis_PNU1002_Bit23	M 1005.7	BOOL	Position 2 reached
Axis_PNU1002_Bit22	M 1005.6	BOOL	Position 1 reached
Axis_PNU1002_Bit21	M 1005.5	BOOL	-
Axis_PNU1002_Bit20	M 1005.4	BOOL	Position latch made
Axis_PNU1002_Bit19	M 1005.3	BOOL	In Position
Axis_PNU1002_Bit18	M 1005.2	BOOL	Current position = home oosition (reference switch is occupied)
Axis_PNU1002_Bit17	M 1005.1	BOOL	Reference point set (after a reference travel or absolute encoder)
Axis_PNU1002_Bit16	M 1005.0	BOOL	Travel order active - Travel set, jogging mode, reference travel
Axis_PNU1002_Bit15_n16	M 1006.7	BOOL	Warning 16: Reserve
Axis_PNU1002_Bit14_n15	M 1006.6	BOOL	Warning 15: Speed current table INXMODE 35 error
Axis_PNU1002_Bit13_n14	M 1006.5	BOOL	Warning 14: SinCos commutation not completed
Axis_PNU1002_Bit12_n13	M 1006.4	BOOL	Warning 13: Expansion card does not work properly
Axis_PNU1002_Bit11_n12	M 1006.3	BOOL	Warning 12: HIPERFACE® or EnDat®: Motor default values have been loaded
Axis_PNU1002_Bit10_n11	M 1006.2	BOOL	Warning 11: Limit switch NSTOP activated
Axis_PNU1002_Bit9_n10	M 1006.1	BOOL	Warning 10: Limit switch PSTOP activated
Axis_PNU1002_Bit8_n09	M 1006.0	BOOL	Warning 9: No reference point was set for Start travel order
Axis_PNU1002_Bit7_n08	M 1007.7	BOOL	Warning 8: A defective travel order was started
Axis_PNU1002_Bit6_n07	M 1007.6	BOOL	Warning 7: Software limit switch 2 exceeded
Axis_PNU1002_Bit5_n06	M 1007.5	BOOL	Warning 6: Software limit switch 1 exceeded
Axis_PNU1002_Bit4_n05	M 1007.4	BOOL	Warning 5: Power supply phase missing
Axis_PNU1002_Bit3_n04	M 1007.3	BOOL	Warning 4: Watchdog (fieldbus) active
Axis_PNU1002_Bit2_n03	M 1007.2	BOOL	Warning 3: Set contouring error exceeded
Axis_PNU1002_Bit1_n02	M 1007.1	BOOL	Warning 2: Set braking power reached
Axis_PNU1002_Bit0_n01	M 1007.0	BOOL	Warning 1: I²t reporting threshold exceeded

Switch ON timing diagram and Start MotionTask



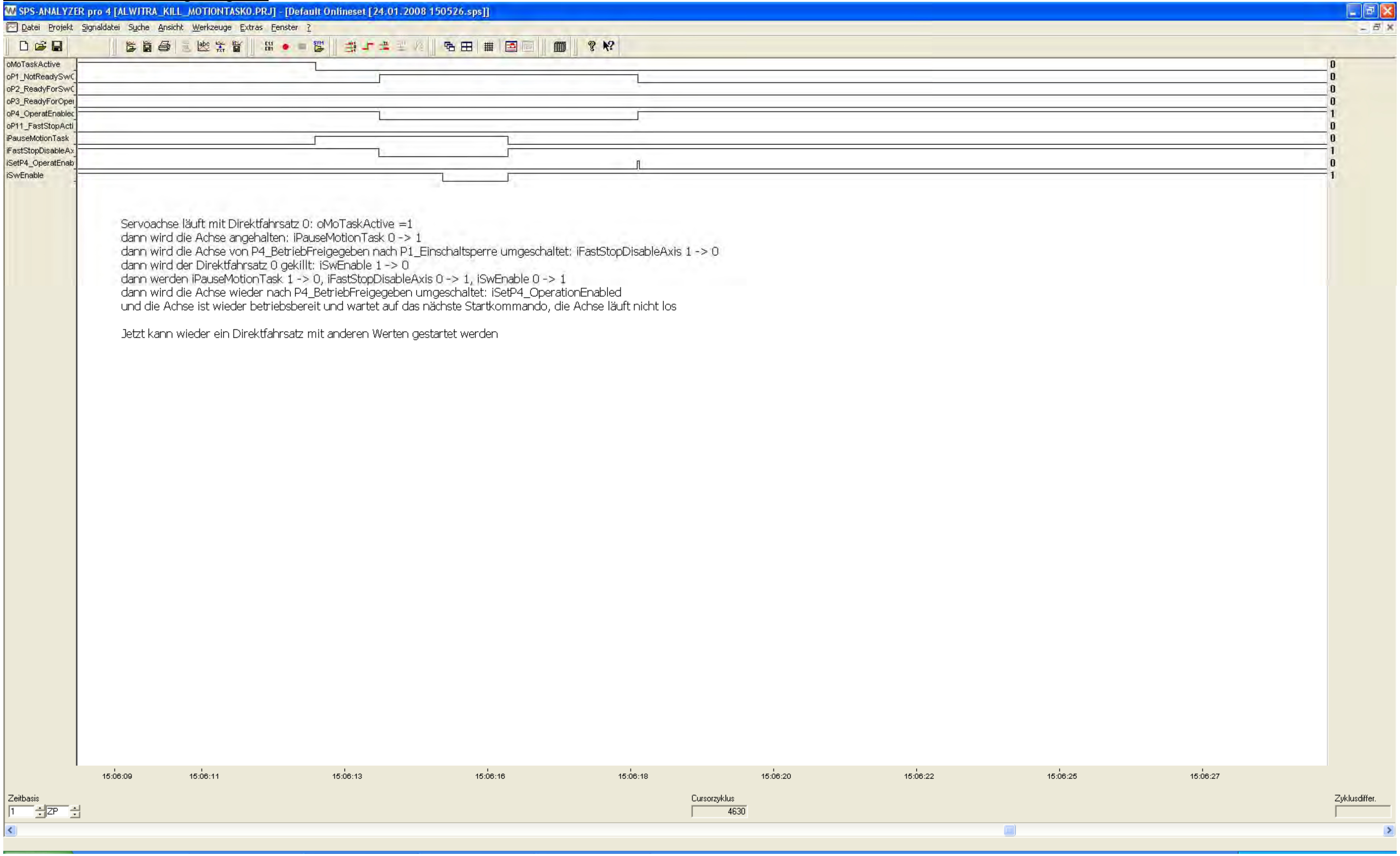
SPS-ANALYZER pro 4 [SV14_AWL.PRJ] - [Default Onlineset [29.05.2009 112215.sps]]

Zeitbasis: 3 ZP

Cursorzyklus: 2726

Zyklusdiffer.:

Kill MotionTask timing diagram



Servoachse läuft mit Direktfahrsatz 0: oMoTaskActive =1
dann wird die Achse angehalten: iPauseMotionTask 0 -> 1
dann wird die Achse von P4_BetriebFreigegeben nach P1_Einschaltsperre umgeschaltet: iFastStopDisableAxis 1 -> 0
dann wird der Direktfahrsatz 0 gekillt: iSwEnable 1 -> 0
dann werden iPauseMotionTask 1 -> 0, iFastStopDisableAxis 0 -> 1, iSwEnable 0 -> 1
dann wird die Achse wieder nach P4_BetriebFreigegeben umgeschaltet: iSetP4_OperationEnabled
und die Achse ist wieder betriebsbereit und wartet auf das nächste Startkommando, die Achse läuft nicht los

Jetzt kann wieder ein Direktfahrsatz mit anderen Werten gestartet werden