

Blocks for Simatic S7-PLC – Step7 and Profibus Connection

SERVOSTAR® 300/400/600 and S700

Table of contents

| | |
|--|----|
| Notes for the S7 project „Sv14_v3c“ for Servostar® S300/400/600 and S700 | 2 |
| Step7-Hardware configuration | 6 |
| Block overview | 9 |
| Notes | 9 |
| Input and output interface of FB14 - Axis_01_FB | 10 |
| FB14 - Axis_01_FB | 10 |
| DB70 – IF_DB | 10 |
| GenLow, GenHigh, GenStartUp und GenSysTime | 11 |
| Addressing from the Hardware-Interface to the FB14 - Axis_01_FB | 11 |
| Program examples for converting the Servostar internal units to user / SI-units | 17 |
| OB1 - Network 17: "IF_DB".ActualPosition (see Pnu 1800: SI-Unit) | 17 |
| OB1 – Netzwerk 18: "IF_DB".ActualSpeed (see Pnu 1815: SI-Unit) | 17 |
| OB1 – Netzwerk 19: "IF_DB".ActualCurrent (see Pnu 1688: SI-Unit) | 17 |
| OB1 – Netzwerk 20: "IF_DB".ActualPressure | 18 |
| OB1 – Netzwerk 21: "IF_DB".ActualFlow | 18 |
| Commissioning FB14 Axis_01_FB | 19 |
| S7-PLC – Switching ON/OFF | 19 |
| S7-PLC – Switching Run/Stop | 19 |
| iAck =1 (Acknowledge) | 19 |
| Initialization | 19 |
| Set operating mode | 19 |
| Enable operation | 19 |
| Jogging mode | 20 |
| Referencing run (Homing run) | 20 |
| Start of an stored EEPROM or RAM MotionTask | 20 |
| Start of the DirectMotionTask (RAM and has the number 0) | 20 |
| Digital speed operating mode | 22 |
| Digital torque operating mode | 22 |
| Electronic gear operating mode | 22 |
| Servopump digital – velocity - operating mode | 23 |
| Macro-Programming | 24 |
| Writing parameter | 27 |
| Reading parameter | 27 |
| Additional Profibus functions | 29 |
| VAT14 variable table | 29 |
| Example for a sequence „Initialisation and Enable Servostar“ | 30 |
| Example for a sequence „Start MotionTasks in the Servostar“ | 31 |
| Example for the “RAM MotionTask 201 in the Servostar“ | 33 |
| Blocks for Servostar parameters | 35 |
| DB 20 - Axis_01_WriteDataDB | 35 |
| DB 21 - Axis_01_ReadDataDB | 35 |
| FB16 - Axis_01_Write | 36 |
| FB17 - Axis_01_Read | 37 |
| FB18 - Axis_01_Compare | 38 |
| Tips and information: | 39 |
| Error messages and warnings from the blocks FB14, FB16, FB17 and FB18 | 39 |
| Override via Profibus | 39 |
| Activating the sin^2 motion profile for DirectMotionTask No. 0 | 40 |
| Bitcoding of error – PNU 1001 - ERRCODE | 43 |
| Bit coding of warnings and manufacturer specific state register – PNU 1002 - DRVSTAT | 44 |
| Switch ON and Start MotionTask timing diagram | 45 |
| Kill MotionTask timing diagram | 46 |

Notes for the S7 project „Sv14_v3c“ for Servostar[®] S300/400/600 and S700

Record of Document Revisions:

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

Trademarks

- Servostar is a registered trademark of Kollmorgen Corporation
- Simatic, Step7, TIA-Portal are registered trademarks of Siemens AG
- Profibus and ProfiNet are registered trademarks of PROFIBUS and PROFINET International(PI)

This document applies to the Profibus-DP connection of the Kollmorgen servo amplifiers Servostar[®] S300/400/600 and S700 to a Simatic S7-PLC controller.

In the following text, the servo amplifiers Servostar[®] S300/400/600 and S700 are simply referred to as „Servostar“.



**Never use the S7 project „Sv14_v3c“ without modifications in an application.
The S7 project „Sv14_v3c“ 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 not liability for damages and precludes all claims arising from the use of the S7 project „Sv14_v3c“ or program components from it.



Take care and follow the security advice of the supplier of the different components.

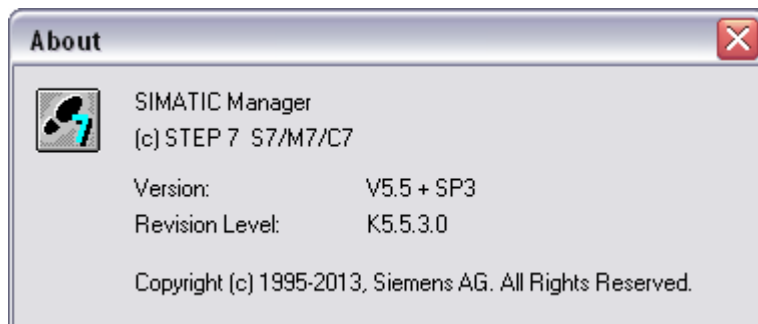
This manual assumes a good knowledge of Simatic Step7, controller and drive technology.

For properly behaviour from the complete Servoaxis consist of S7-PLC and Servostar and motor the commissioning is acceptably realized !!!

All references to the manual refer to the technical description „srprof_e.pdf“.

The terms and abbreviations used correspond to the Servostar setup software DriveGui or the manual in the English language.

The S7 project „Sv14_v3c“ was created in English with German Mnemonics and the version:



The S7 project „Sv14_v3c“ consists of the following components:

- Hardware configuration (system data) for a Simatic S7-CPU317-2DP – 6ES7 317-2AJ10-0AB0 / V2.1
- Hardware configuration (system data) for a VIPA – CPU 315SB – 315-2 AG12
- S7 program – contains the blocks and symbols

| Object name | Symbolic name | Created in language | Size in the work me... | Type | Version (Header) | Name (Header) |
|-------------|---------------------|---------------------|------------------------|-------------------------|------------------|---------------|
| OB1 | GenCycleOB | FBD | 2828 | Organization Block | 0.1 | |
| OB82 | GenIOFbOB | FBD | 38 | Organization Block | 0.1 | |
| OB85 | GenIOFbOB | FBD | 38 | Organization Block | 0.1 | |
| OB86 | GenRackFaultOB | FBD | 38 | Organization Block | 0.1 | |
| OB100 | GenStartupOB | STL | 48 | Organization Block | 0.1 | |
| OB121 | GenProgErrOB | FBD | 38 | Organization Block | 0.1 | |
| OB122 | GenModErrOB | FBD | 38 | Organization Block | 0.1 | |
| FB14 | Axis_01_FB | FBD | 6590 | Function Block | 0.5 | Axis_01 |
| FB16 | Axis_01_Write | STL | 748 | Function Block | 0.2 | Axis_01 |
| FB17 | Axis_01_Read | STL | 930 | Function Block | 0.2 | Axis_01 |
| FB18 | Axis_01_Compare | STL | 802 | Function Block | 0.1 | Axis_01 |
| DB14 | Axis_01_DB | DB | 374 | Instance data block ... | 0.0 | |
| DB16 | Axis_01_WriteDB | DB | 84 | Instance data block ... | 0.0 | |
| DB17 | Axis_01_ReadDB | DB | 92 | Instance data block ... | 0.0 | |
| DB18 | Axis_01_CompareDB | DB | 86 | Instance data block ... | 0.0 | |
| DB20 | Axis_01_WriteDataDB | DB | 844 | Data Block | 0.1 | Axis_01 |
| DB21 | Axis_01_ReadDataDB | DB | 844 | Data Block | 0.1 | Axis_01 |
| DB70 | IF_DB | DB | 168 | Data Block | 0.1 | Axis_01 |
| UDT10 | GenTimerUDT | STL | --- | Data Type | 0.1 | GEN_TIME |
| UDT27 | IFGroupToUnitUDT | STL | --- | Data Type | 0.1 | IF_G_U |
| UDT28 | IFUnitToGroupUDT | STL | --- | Data Type | 0.1 | IF_U_G |
| UDT141 | M_Axis_01_Request | STL | --- | Data Type | 0.2 | Axis_01 |
| UDT143 | M_Axis_01_MaMsg | STL | --- | Data Type | 0.1 | Axis_01 |
| UDT145 | M_Axis_01_State | STL | --- | Data Type | 0.1 | Axis_01 |
| UDT146 | M_Axis_01_InData | STL | --- | Data Type | 0.2 | Axis_01 |
| UDT147 | M_Axis_01_OutData | STL | --- | Data Type | 0.1 | Axis_01 |
| UDT149 | M_Axis_01_RcvSend | STL | --- | Data Type | 0.1 | Axis_01 |
| VAT14 | VAT14 | --- | --- | Variable Table | 0.1 | |
| SFC12 | D_ACT_DP | STL | --- | System function | 1.0 | D_ACT_DP |
| SFC14 | DPRD_DAT | STL | --- | System function | 1.0 | DPRD_DAT |
| SFC15 | DPWR_DAT | STL | --- | System function | 1.0 | DPWR_DAT |
| SFC20 | BLKMOV | STL | --- | System function | 1.0 | BLKMOV |
| SFC21 | FILL | STL | --- | System function | 1.0 | FILL |
| SFC64 | TIME_TCK | STL | --- | System function | 1.0 | TIME_TCK |

Properties - Block Folder Offline

General | Blocks | Checksums | Address priority: | Fill Level

Behavior as in STEP7 < V5.2

Absolute value has priority: ☐ Symbols are applied from the symbol table and the DB for all accesses (I, Q, M, T, C and DB)

Symbol has priority: ☒ Exception: for accesses in structurally unchanged data types, the current symbols will be applied

Recommended for symbolic programming

☐ Exception: symbol accesses on the DB remain as they were programmed in the code block

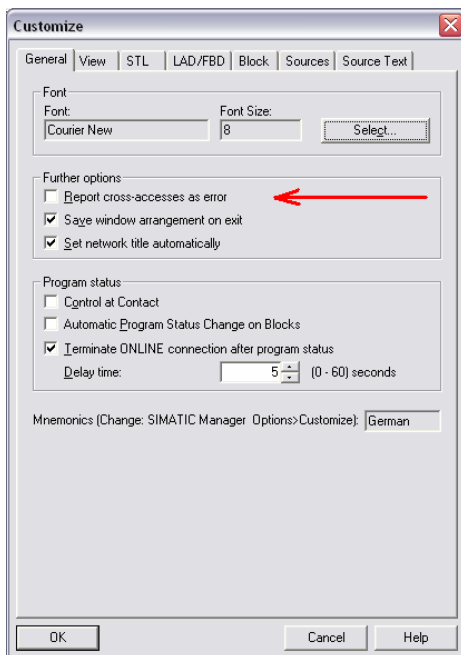
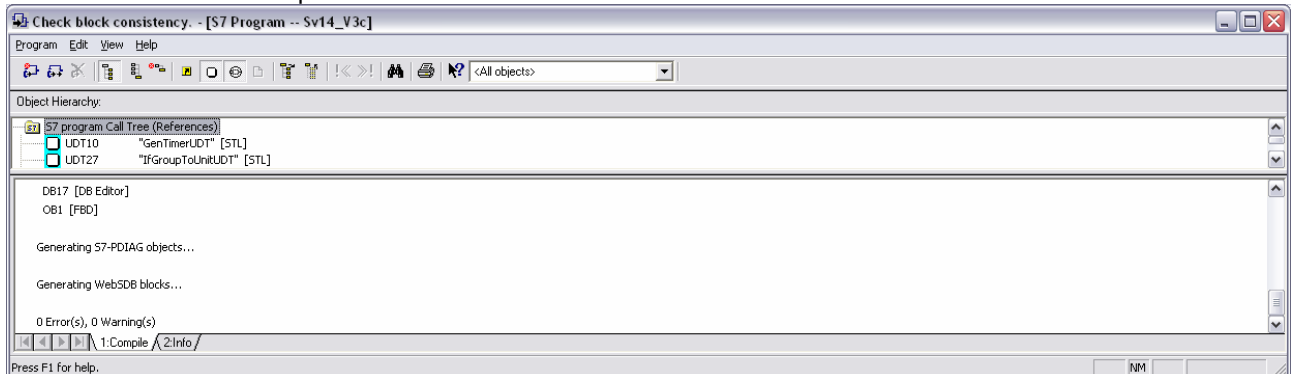
☒ For all accesses (I, Q, M, T, C and DB)

OK Cancel Help

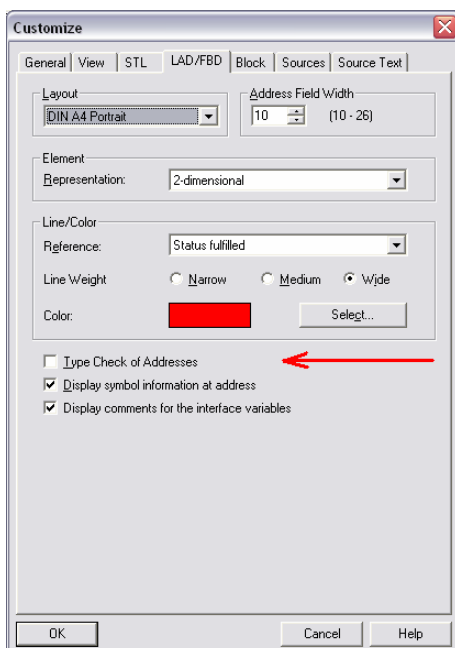
The properties of the block folder Offline – Operand precedence are:
„Symbol has precedence for all accesses (I, Q, M, T, C und DB)“.

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 “Check block consistency” and “Compile all”. If the message “0 errors, 0 warnings” is subsequently not displayed, the S7 program still contains address conflicts in the blocks that need to be resolved manually. This is sometimes the case for multi instances. These address conflicts are then manually removed in the blocks using the LAD/STL/FBD editor with FILE \ “Check and update accesses”.



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 LAD/STL/FBD editor:
 OPTIONS\CUSTOMIZE\General “Report cross accesses as error”. This is also the default setting of the S7 Manager. Afterwards, the program must be compiled again.



For the complete FBD representation of the networks, “Type Check of Addresses” must be deactivated in the LAD/STL/FBD editor:
 OPTIONS\CUSTOMIZE\LAD/FBD.

Step7-Hardware configuration

Among other things please note this settings:

- * Diagnostic address
- * Input Address / Output Address
- * GSD-file-version
- * The Servostar adjust itself automatically to the transmission speed (baud rate)
- * Watchdog

The screenshot shows the 'HW Config - [SIMATIC-317 (Configuration) -- Sv14_V3c]' window. On the left, a rack configuration table is visible:

| Slot | Module |
|------|-----------------|
| 1 | CPU317-2 |
| X1 | MPI/DP |
| X2 | DP |
| 3 | |
| 4 | DI16xDC24V |
| 5 | DO16xDC24V/0.5A |
| 6 | |
| 7 | |
| 8 | |
| 9 | |
| 10 | |

To the right, a network diagram shows 'PROFIBUS(1): DP master system (1)' connected to a slave node labeled '(12) Axis'.

Below the rack configuration, a table for 'PROFIBUS(1): DP master system (1)' is shown:

| PROFIBUS address | Module | Order number | Firmware | Diagnostic address | Comment |
|------------------|--------|--------------|----------|--------------------|---------|
| 12 | Axis | | | 1012 | |

| [12] Axis | | | | | |
|-----------|-------|--------------------------------|-----------|-----------|---------|
| Slot | DP ID | Order Number / Designation | I Address | Q Address | Comment |
| 1 | 4AX | 4 Wort AE/AA/Konsistenz gesamt | 256...263 | 256...263 | |
| 2 | 6AX | 6 Wort AE/AA/Konsistenz gesamt | 264...275 | 264...275 | |

Properties - DP slave

General

Module

Order number: GSD file (type file): KOLL045D.GSD

Family: General

DP slave type: Servostar

Designation: Axis

Addresses

Diagnostic address: 1012

Node/Master System

PROFIBUS... 12

DP master system (1)

SYNC/FREEZE Capabilities

☒ SYNC ☒ FREEZE

☒ Watchdog

Comment:

OK Cancel Help

Servostar settings with DriveGui

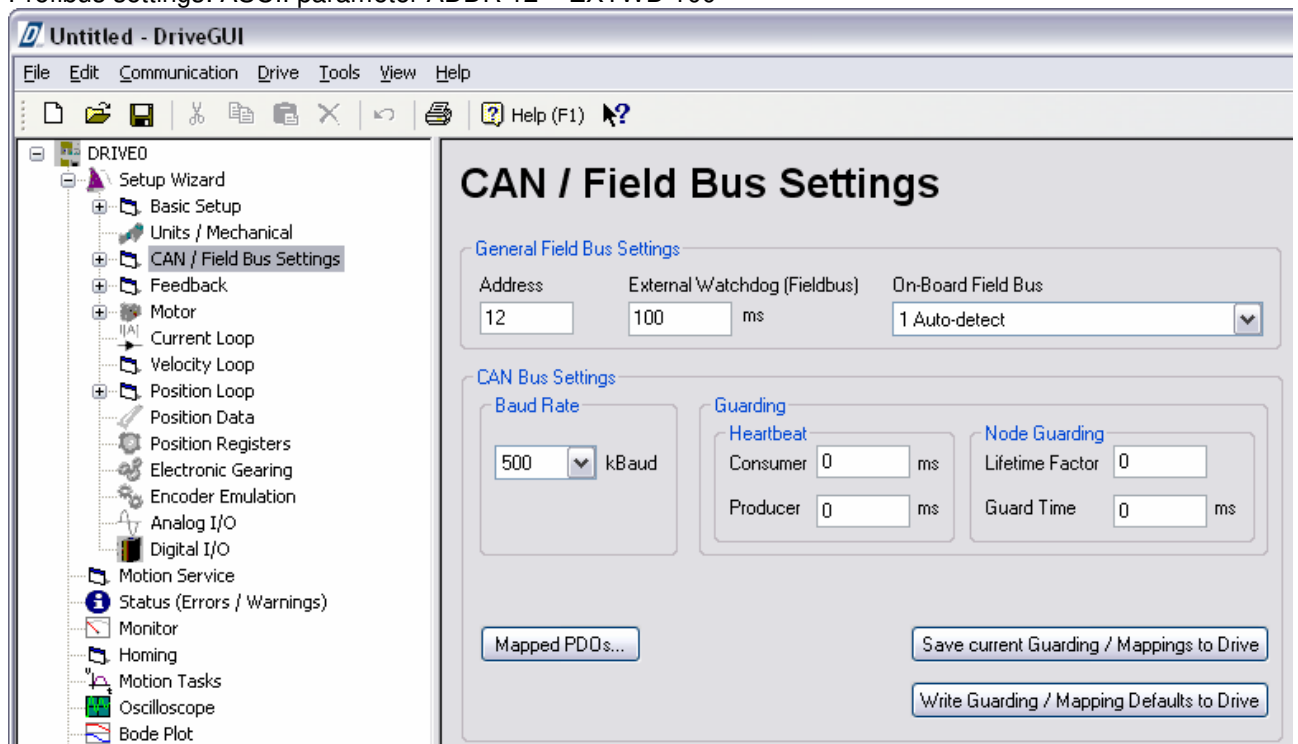
For commissioning the Servostar S400 / S600 the Software Drive is used.
It is not described here more in detail.



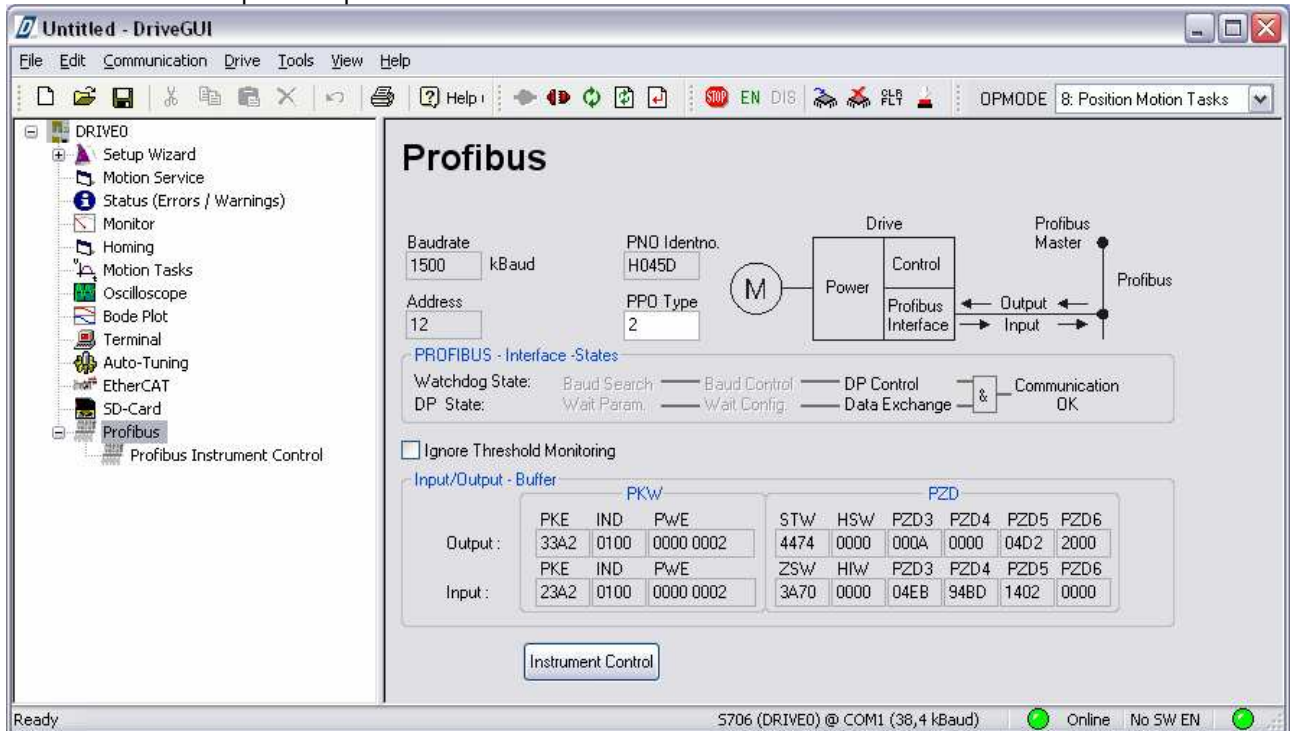
For commissioning the Servostar S300 / S700 the Software DriveGui is used.
DriveGui-Version:



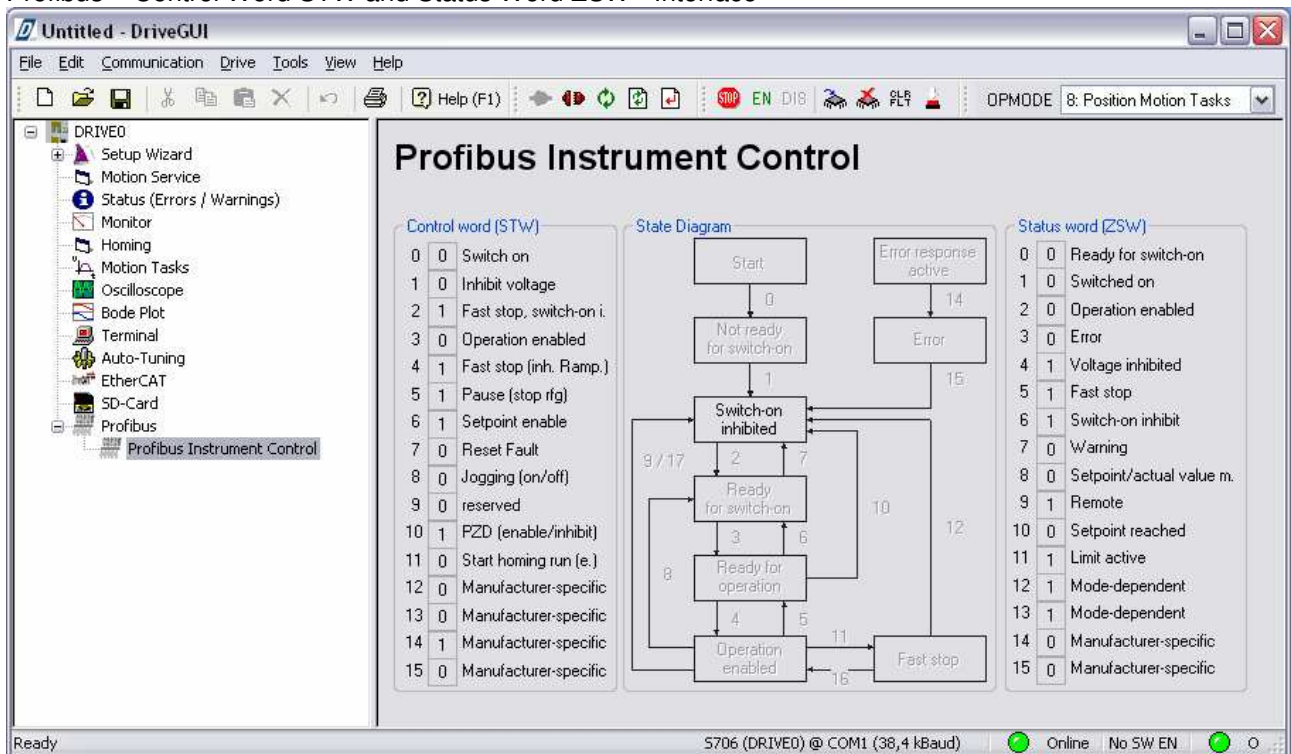
Profibus settings: ASCII parameter ADDR 12 – EXTWD 100



Profibus – Data – Input / Output or Send / Receive – Interface PKW and PZD



Profibus – Control Word STW and Status Word ZSW - Interface



Block overview

| | | |
|-------------------|-----------|---|
| Axis_01_FB | FB14/DB14 | Handling block between S7-PLC and Servostar |
| Axis_01_Write | FB16/DB16 | Block for the FB14 to write PNU data to Servostar |
| Axis_01_WriteData | DB20 | Contains the PNU data that are written to Servostar with FB16 |
| Axis_01_Read | FB17/DB17 | Block for the FB14 to read PNU data from Servostar |
| Axis_01_ReadData | DB21 | Contains the PNU data that are read from Servostar with FB17 |
| Axis_01_Compare | FB18/DB18 | Module for PNU data comparison, e.g. DB20 and DB21 |
| IF_DB | DB70 | Interface data block e.g. for an operator panel / touch panel |

All blocks can be renamed or renumbered upon demand.

Afterwards, "Check block consistency" must be performed with "Compile all".

The resulting conflicts may have to be corrected with "Check and update accesses" or manually in the blocks.

The FB14 uses the following S7-PLC system blocks:

SFC12 – D_ACT_DP – Deactivating and Activating DP Slaves / ProfiNet IO devices

SFC14 – DPRD_DAT – Reading consistent data of a DP Slave / ProfiNet device

SFC15 – DPWR_DAT – Writing consistent data of a DP Slave / ProfiNet device

SFC20 – BLKMOV – Copy memory area

SFC21 – FILL – Fill memory area

The OB1 uses the following S7-PLC system blocks:

SFC64 – TIME_TCK – Reading system time (used to implement the timers in the program)

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

Notes

- The FB14 is programmed as multi instance and, therefore, can be called multiple times in FBs as subprogram block without separate instance data block.
- If FB14 is not implemented as multi instance, then FB14 must be called up for every Servostar amplifier with a separate instance data block DB.
- FB14 is programmed in STL und FBD (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_v3c“ contains a completely functional S7-PLC program for Servostar - Profibus.
- The S7 project „Sv14_v3c“ can easily migrate to the Simatic TIA-Portal.
For TIA13 – SP1 and TIA14 - SP1 are completely functional PLC projects in FBD for S7-1200 PLC and S7-1500 PLC available.

With a S7-300 / 400 - PLC and a Servostar with Profibus option card and motor, the complete Servoaxis can be easily and quickly commissioned using the VAT14 (table of variables).

Input and output interface of FB14 - Axis_01_FB

The interface of FB14, FB16, FB17, FB18 and DB20, DB21, DB70 uses UDT (UserDefinedDataType).

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

The UDTs ensures that the data structure in the S7 project has an uniform common database. A modification to the data structure is done once in the UDT. The entire data structure of the S7 project is automatically updated with “Check block consistency” and “Compile all”. Time stamp conflicts are also removed.

Please check afterwards that the Initial values from the UDT are transferred to the Actual values from the DB. If not then for this DB a separate initialization is required:

Open the DB with the LAD/STL/FBD-Editor – change to the DATA VIEW and EDIT \ Initialize Data Block
 Afterwards, “Check block consistency” must be performed with “Compile all”.

FB14 - Axis_01_FB

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

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

DB70 – IF_DB

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

GenLow, GenHigh, GenStartUp und GenSysTime

This flags will generated see OB100 – NW1 and see OB1 – NW1

OB100 : "Complete Restart"

Comment:

Network 1: reset

```

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

```

OB1 : Program Cycle Organization Block

Comment:

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

```

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

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

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

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

```

Addressing from the Hardware-Interface to the FB14 - Axis_01_FB

The Input address / Output address and Diagnostic address is adjusted in the hardware configuration. See chapter Step7-Hardware configuration. Every Servostar need the mapping from his device configuration to the Axis_01_FB (FB14).

If the mapping is executed correct the S7 PLC and the Servostar communicate over the Profibus and the FB14 - Axis_01_FB answer oState.CommunicationOk =1

OB1 - GenCycleOB

Network 12: Init parameters

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

// overwrite !!!

```

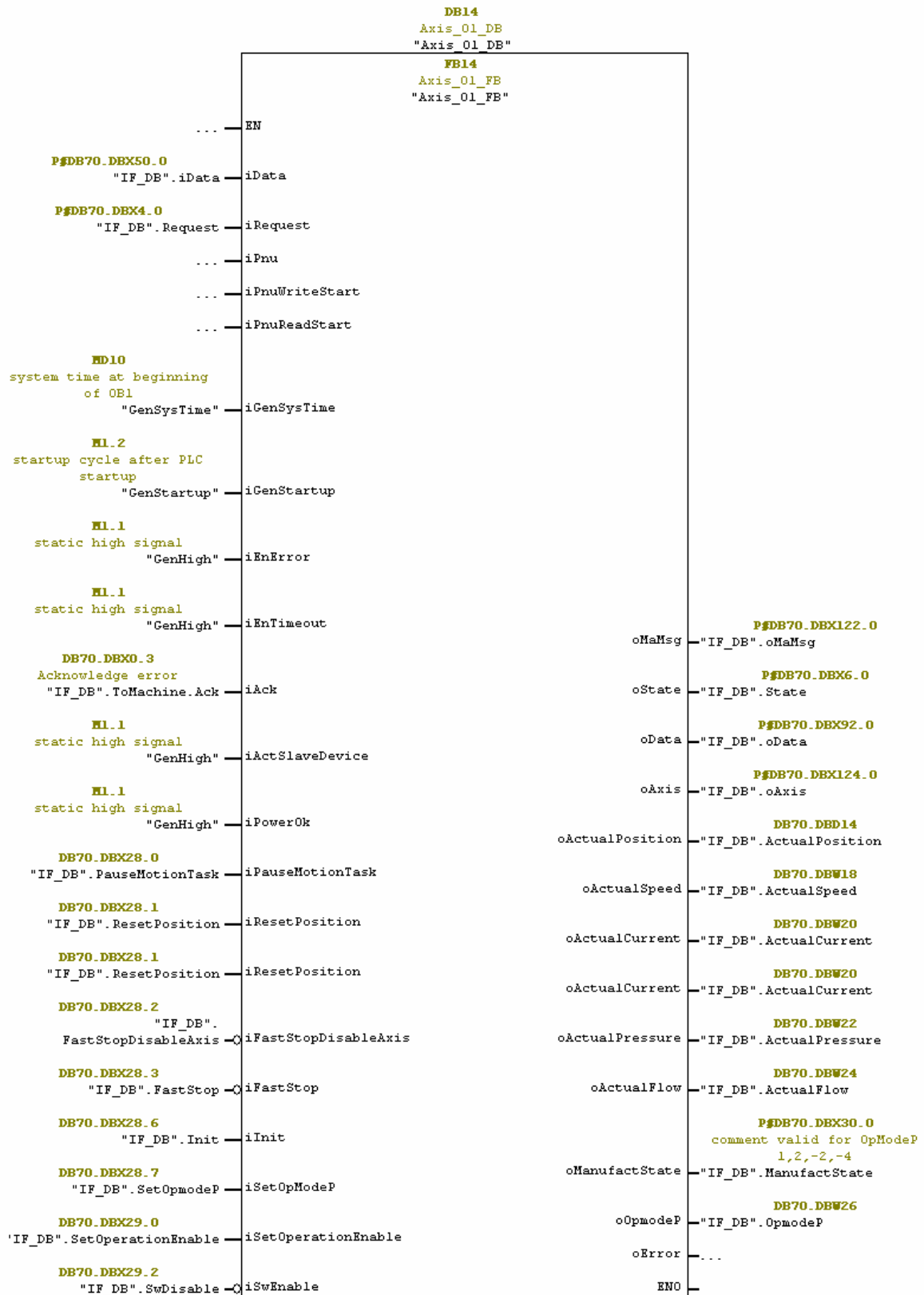
L      256
T      "IF_DE".iData.Config.LAddrIn    DE70.DEW50          -- StartAddressIn in HW-Config periphery dez
T      "IF_DE".iData.Config.LAddrOut    DE70.DEW52          -- StartAddressOut in HW-Config periphery dez

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

```

OB1 - GenCycleOB

□ Network 13 : Axis_FB



Complete overview of interface from FB14

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

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

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

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

1*) The actual position in the Servostar is transferred in the process data only in internal units to the S7-PLC (2^{20} incr. per motor revolution). The conversion from internal units to user units can be performed in the S7-PLC, depending on the position resolution defined in the Servostar by the DriveGui software.

Example with PRBASE 20:

Resolution = 5000 incr / 3 revolutions

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

Program examples for converting the Servostar internal units to user / SI-unitsOB1 - Network 17: "IF_DB".ActualPosition (see Pnu 1800: SI-Unit)

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

```
// "IF_DB".ActualPosition [µm, SI-Unit - DWORD]
// = "Axis_01_DB".oActualPosition [Counts] * (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_DB".ActualPosition
```

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

// example: VOSPD 3600 rpm

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

// use datatype 32-bit integer

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

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

// example for Servo Amplifier - with Ipeak 9 Ampere

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

// use datatype 32-bit integer

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

OB1 – Network 20: "IF_DB".ActualPressure

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

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

OB1 – Network 21: "IF_DB".ActualFlow

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

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

Commissioning FB14 Axis 01 FB

S7-PLC – Switching ON/OFF

If the S7-PLC Profibus-Master fails, the watchdog timeout starts in the Servostar (EXTWD) and generate the warning „n04 – node guarding (watch dog)“.

After startup, the S7-PLC performs a software reset (GenStartUp - M1.2) and deletes all warnings and error messages and states in the PLC program, but not in the Servostar.

If the Servostar boot up faster than the S7-PLC the Servostar generate the n04 warning.

With the ASCII-Parameter S1DLY it is possible to increase the boot up time e.g. to avoid the n04 warning.

After acknowledge the n04 warning the Servostar is able for operation.

S7-PLC – Switching Run/Stop

If the state change from STOP -> RUN the S7-PLC performs a software reset (GenStartUp - M1.2) and deletes all warnings and error messages and states in the PLC program, but not in the Servostar.

iAck =1 (Acknowledge)

Reset warnings and errors in the Servostar and in the S7-PLC program in the blocks FB14, FB16, FB17, FB18.

Initialization

During initialization, a zero telegram (Send.PKW and Send.PZD) is transmitted from S7-PLC to Servostar.

The Servostar disable immediately and the state machine change to oState.P1_SwitchOnInhibited =1

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

Set operating mode

Programmed operating mode: Positioning, Digital velocity, Digital torque, Electronic gear, Digital ServoPump
After 24VDC switch-on, the Servostar is always in the safe internal operation mode -126 and “locked”.

Via Profibus there is a communication established between the S7-PLC and Servostar only from STW and ZSW and the RCV - PZD2, PZD3, PZD4, PZD5, PZD6 (receive telegram) contains all the value “0”.

The S7-PLC must first change and recheck the operating mode.

- Write the desired operating mode in iData.Config.OpModeP
(see manual: Positioning: 2 ; Digital velocity: 1 ; Digital torque: -2 ;
Electronic gear: -4 , Digital Servopump: -7)
- Set iSetOpModeP =1 (a pulse is generated internally)
- Wait until oState.OpModeP_Ok = 1
- The activated operating mode will be visible in oOpModeP
- Now make the recheck for the desired OpmodeP
- If oState.OpModeP_Ok is not =1 or oState.OpModeP_Error =1
or oOpModeP is not the desired OpmodeP then setting the operating mode was not successful.
- For error cause see Servostar receive telegram.
- Now the RCV - PZD2, PZD3, PZD4, PZD5, PZD6 is displayed depending on the operating mode
with values from e.g. oActualPosition, oActualSpeed, oManufactState.
- Setting the operating mode is monitored with a timeout of 1 sec.

Enable operation

After 24VDC switch-on, the Servostar is in oState.P1_SwitchOnInhibited =1.

iFastStop, iFastStopDisableAxis must be =0 and iSwEnable must be =1.

At the Servostar, HardwareEnable must be =1 and if present AS/STO-Enable must be =1.

Only S700: A build in safety card must be in state „RUN“.

The DC link voltage is applied and the display of Servostar shows „Pxx“ without warnings or errors.

- Set iSetOperationEnable =1 (a pulse is generated internally)
- Wait until oState.P4_OperationEnabled =1
- If oState.P4_OperationEnabled is not =1, the enable operation was not successful.

The Servostar is now enabled and able to generate force / torque and move a motor.

Requirement for positioning operation mode

- The operation is enabled (oState.P4_OperationEnabled =1)
- AS/STO- and Hardware-Enable is present
- No warning and error-free and the display from Servostar shows „Exx“
- The operation mode positioning (oState.OpModeP_Ok = 1 und oOpModeP = 2 dec) is activated

Jogging mode

- Write the desired jogging speed to "iData.Config.JogSpeed"
- Set iRequest.JogPlus =1 -> The motor moves positive and oState.JoggingActive =1
- Set iRequest.JogMinus =1 -> The motor moves negative and oState.JoggingActive =1

Referencing run (Homing run)

The referencing type is normally adjusted with DriveGui and stored in the Servostar.

- Write the desired referencing run speed to "iData.Config.RefSpeed"
- Set iRequest.StartRef =1 -> The motor move and oState.ReferencingActive =1
- Wait until oState.ReferenceOk =1 and oState.ReferencingActive again =0
- Set iRequest.StartRef =0 – The Servostar is referenced (homed).

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

Start of an stored EEPROM or RAM MotionTask

Additional requirement: The Servostar is referenced.

The DriveGui is use to create MotionTasks and store them in the Servostar EEPROM.

Only the parameters of the RAM MotionTask can be changed with the S7-PLC in the Servostar even in the state "P4_OperationEnabled" and while the motor is moving (see ASCII - MTMUX)

- Write the number of the MotionTask to iData.MotionTask.Number
- Set iRequest.StartMotionTask =1 (a pulse is generated internally)
- Wait until oState.InPosition =0 and oState.MotionTaskActive =1
- Wait until oState.InPosition =1 and oState.MotionTaskActive =0

The Servostar executed the MotionTask. With iEnTimeout =1 and iData.Config.TO_Position xxx ms [TIME], the MotionTask run is monitored with a timeout.

Actual position, actual speed and the manufacturer state are cyclically transmitted in the RCV.PZD channel.

Start of the DirectMotionTask (RAM and has the number 0)

Additional requirement: The Servostar is referenced.

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

The Servostar executed the MotionTask. With iEnTimeout =1 and iData.Config.TO_Position xxx ms [TIME], the DirectMotionTask run is monitored with a timeout.

Actual position, actual speed and the manufacturer state are cyclically transmitted in the RCV.PZD channel.

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

With SFC20-BLKMOV and pointer (P#DB.DBX byte), the Data.MotionTask [STRUCT] can be copied easily.

See "Switch ON and Start MotionTask timing diagram"

A started MotionTask can be paused with `iPauseMotionTask = 1`.
The Servostar remain in state „P4_OperationEnabled“.

A started MotionTask can be stopped with `iRequest.StopMotionTask = 1`.
The Servostar change to state „P3_ReadyForOperation“.
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”

In the OB1 is an example to parameterize parts of a MotionTask and WRITE from S7-PLC to the Servostar with the parameter channel.

Note: The ASCII parameter INPT0 PNU 1904 set the time in ms for which the signal `oManufactState.InPositionWindow` is reset at a defined time after the MotionTask start (see manual ch. VII.1).
So the interface timing behaviour between the S7-PLC and the Servostar is always the same.

Note about positions and speeds - MotionTaskType O C – PNU 1785

0x0000 hex (bit 13 =0)
Absolute positioning with preset of speeds and positions only in incremental internal 32 bit and 16 bit units and trapezoidal motion profile.

0x2000 hex (bit 13 =1)
Absolute positioning with preset of speeds and positions in user units and trapezoidal motion profile.

0x2003 hex (bits 0, 1, 13 =1)
Relative positioning with preset of speeds and positions in user units and trapezoidal motion profile.

0x12000 hex (bit 13 =1 and bit 16 =1)
Absolute positioning with preset of speeds and positions in user units and sin² motion profile.

0x10000 hex (bit 16)
Cannot be set directly in the PZD channel.
See Activating the sin² motion profile for DirectMotionTask

Note about Feedback with Wake&Shake (W&S)

Actual position, actual speed and the manufacturer state are cyclically transmitted in the RCV.PZD channel after Set operating mode is done (`oState.OpModeP_Ok = 1` und `oOpModeP = xxx dec`) and the Wake&Shake procedure is finished.

Remark: `n14 = 1` Scanning for MPHASE (e.g. `FBTYPE=7`)
Is set while start-up of the drive and is cleared after the drive was enabled and MPHASE was determined with Wake&Shake.

If you receive an error from Set operating mode please enable first the Servostar (`iSetOperationEnabled = 1`) and then the Wake&Shake procedure will be execute.
Start afterwards Set operating mode.

Digital speed operating mode

The digital speed operating mode (oState.OpModeOk = 1 and oOpmodeP = 1 dec) is activated and the operation is enabled (oState.P4_OperationEnabled = 1).

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

Actual position, actual speed (16 bit) and the manufacturer state are cyclically transmitted in the RCV.PZD channel.

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

PNU 1886 - Ncommand (Ncmd16 = Ncmd * 2¹⁵ / VOSPD)

Digital torque operating mode

The digital torque operating mode (oState.OpModeOk = 1 and oOpmodeP = -2 dec) is activated and the operation is enabled (oState.P4_OperationEnabled = 1).

- Write target current to iData.DigitalTorque.Icmd
- Set iRequest.StartIcmd = 1
- The Servostar injects the preset target current in the motor.
- The target current can be changed at any time.
- Set iRequest.StartIcmd = 0 and the Servostar outputs the target current 0 Ampere.

Actual position, actual current and the manufacturer state are cyclically transmitted in the RCV.PZD channel.

The current value is calculated using the following formula:

PNU 1870 - Icommand (I[A] = Icmd * IpeakAmplifier[A] / 3280)

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

Electronic gear operating mode

The operation mode Electronic gear (oState.OpModeP_Ok = 1 und oOpmodeP = -4 dec) is activated and the operation is enabled (oState.P4_OperationEnabled = 1).

Actual position, actual speed and the manufacturer state are cyclically transmitted in the RCV.PZD channel.

Servopump digital – velocity - operating mode

only for S300/S700 - see "Applikationsschrift – Servopumpe - s700_servopumpe_d.pdf"

The operation mode Servopump digital (oState.OpModeP_Ok = 1 und oOpmodeP = -7 dez) is activated and the operation is enabled (oState.P4_OperationEnabled =1).

Note: While activating the opmode -7, automatically QENA set to 1.

By switching to another operation mode than -7, the Servopump will be deactivated (QENA=0).

- Write target pressure to iData.DigitalPump.QPFRcmd
OR
- Write target flow to iData.DigitalPump.QPFRcmd
- Set iRequest.StartQcmd =1
- The Servostar injects the preset target pressure / flow.
- The target pressure / flow can be changed at any time.
- Set iRequest.StartQcmd =0 and the Servostar outputs the target pressure / flow 0.

Actual pressure, actual flow, actual current and actual position are cyclically transmitted in the RCV.PZD channel.

With this operation modes the most requirements can solved in a machine.

Further operation modes are not implemented in the FB14, but can very easily additional programmed.

Macro-Programming

A macro program can execute in the Servostar further functions.

For macro programming the Software MacroStar is used.

In the parameter channel could be used for the data exchange between S7 PLC and the Servostar the ASCII parameter DPRVARxx and DPVxx for Macro und PLC programs in the parameter channel.

e.g. DPRVAR1 - PNU 2022 (dec) IND = 1 (dec)

In the real time channel exist additional iRequest.MacroInput

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

For special requirements it could be necessary to change and expand e.g. the Axis_01_FB (FB14) program code.

The complete SEND and RECEIVE data could be used also by the macro program.

Thereby the MacroStar compiling operated error-free it is necessary that the file variables.cfg contain:

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

Example for S7-PLC und Macro-programming in S300/S700:

The motor turns endless and stop followed directly into a defined transfer position without jerk with a smooth \sin^2 brake ramp. The Target position and VJOG velocity will either transmit from the S7-PLC as DirectMotionTask parameter or could be stored as DPRVARxx at the S300/S700-EEPROM.

The complete motion is to be checked at the machine with the DriveGui-Oscilloscope.
(Position, Position error, Actual current, Actual velocity)

S300/S700-Parameter:

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

S7-PLC Program:

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

Activate the Macro programm with M200.0

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

Makro program:

```

PROGRAM PLCINIT

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

END_PROGRAM

//*****
PROGRAM PLCMAIN

END_PROGRAM

//*****

PROGRAM PLC250

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

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

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

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

END_PROGRAM

```

Writing and reading parameters:

All parameters and commands of Servostar can be addressed via the Profibus PNU number. In the ASCII list Object Reference, 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.

ASCII Object Reference

Switch language to [german](#)

sorted by

commands

groups

object n°

ACC

ACC_X

ACCR

ACCUNIT

ACTFAULT

ACTIVE

ACTRS232

ADC0_15

ADDR

AEHA

ALIAS

AIH1HR

AIH1RANGE

AIHTRIG

AIH2TRIG

AIHCFG

ANDB

ACC

Syntax Transmit

Syntax Receive

Type

ASCII Format

DIM

Range

Default

Opmode

Drive State

Start Firmware

Configuration

Function Group

Short Description

ACC

ACC [Data]

ACC <Data>

Variable rw

Integer32

>> ACCUNIT

3 ..126000

31400

0, 1

-

1.0

No

velocity loop

Acceleration ramp

Available in

CAIIBus Object Number

PROFIBUS PNU

DPR

Data Type Bus.DPR

Weighting

Last Change of this Object

EEPROM

Yes

3501 (hex)

1601 (dec) IND = 1 (dec)

1 (dec)

Integer16

2.7

Yes

Description

This variable defines the acceleration ramp for the velocity control loop. The acceleration ramp is only used for command changes resulting in a velocity increase (acceleration). **DEC** is used for braking (deceleration).

See also [ACCUNIT](#)

Writing parameter

Requirement – PKW parameter channel is not yet in use

- Write PNU number to iPnu.Write.Number
 - Write Index to iPnu.Write.Index
 - Write value to iPnu.Write.Value
 - Set iPnuWriteStart =1 (pulse)
 - Wait until oState.PnuWriteOK =1
- 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 parameter

Requirement – PKW parameter channel is not yet in use

- Write PNU number to iPnu.Read.Number
 - Write Index to iPnu.Read.Index
 - 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 to
 - 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. current for determining the torque).

In the OB1 are some examples for several PNUs.

Note: Simultaneous Read and Write request at FB14 generates the error oMaMsg.ErrReadWrite

The S7-SPS could parametrize the Servostar and store this data with SAVE and COLDSTART in the EEPROM.

At some parameters is after a change also a SAVE and afterwards COLDSTART necessary.

Please do this modification ever in the safe state from the Servoaxis.

Disable first the Servostar:

Set iSwEnable =0

Check that oState.P1_SwitchOnInhibited =1

Set the Hardware-Enable Input =0

Set AS/STO-Enable =0

Start then a initialisation (Zero-telegram)

Set iInit =1 (a pulse is generated internally)

Wait until oState.InitOk =1

Change now the parameters: execute with WRITE

Start now the SAVE – command – execute with READ

See ASCII parameter SAVE – PNU 1835

Note: The maximum permissible SAVE write cycles in an EEPROM is limited !!!

Wait 5 seconds – The parameters will now stored in the Servostar-EEPROM.

COLDSTART – command – execute with READ.

The Servostar boot now and make a Newstart / Restart.

The S7-PLC reports meanwhile a „Slave-Servostar – communication error“

Wait until the Servostar ist running again in the Profibus: oStateCommunicationOK =1

Reset now the warnings and errors in the Servostar and in the S7-PLC program in the blocks

iAck =1 (Acknowledge)

Now execute READ and COMPARE to check that the Servostar stored correct the values in the EEPROM.

In the Servostar must now execute again set operating mode and execute enable operation.

Additional Profibus functions

The DP slave Servostar is activated or deactivated in FB14 with SFC12 – D_ACT_DP with running S7-PLC.

- iActSlave =1, Servostar is activate and connect with data exchange in the Profibus.
- iActSlave =0, Servostar is deactivated and data exchange is terminated. The watchdog timeout starts in the Servostar (EXTWD) and generate the warning „n04 – node guarding (watch dog)“. The Servostar can now unplug from the Profibus without error message at the S7-PLC.

The Profibus communication is monitored with SFC14 - DPRD_DAT and SFC15 - DPWR_DAT for

- configuration errors
- Slave failure
- 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 MotionTask 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 the MotionTask data and finish the relative MotionTask with corrected data.

VAT14 variable table

With the VAT14 table of variables, the blocks can be controlled and monitored.

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

Example for a sequence „Initialisation and Enable Servostar“

```

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

U      M      100.1
=      "IF_DB".ToMachine.Ack

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

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

// -----

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

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

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

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

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

```

See “Switch ON and Start MotionTask timing diagram”

Example for a sequence „Start MotionTasks in the Servostar“

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

```

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

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

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

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

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

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

U      M      110.6
UN     "Axis_01_DB".oState.MotionTaskActive

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

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

```

```
// -----
      U      M      110.3
      SPBN   m201
      L      201
      T      "IF_DB".iData.MotionTask.Number
m201: NOP    0

      U      M      110.5
      SPBN   m210
      L      210
      T      "IF_DB".iData.MotionTask.Number
m210: NOP    0

      U      M      110.7
      SPBN   m211
      L      211
      T      "IF_DB".iData.MotionTask.Number
m211: NOP    0

      CLR
      O      M      110.3
      O      M      110.5
      O      M      110.7
      =      "IF_DB".Request.StartMotionTask
```

See “Switch ON and Start MotionTask timing diagram”

Example for the “RAM MotionTask 201 in the Servostar“

```
// ORDER 201 - RAM MotionTask

// PNU 1947 Index 1 - MTMUX

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


//MTMUX (PNU 1947) address with RamMotionTask 201

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

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

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


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

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

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

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


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

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

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

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


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

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

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

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

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

```

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

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

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

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

```

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

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

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

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

```

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

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

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

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

```

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

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

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

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

```

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

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

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

Note for the MTMUX – sample OB1:

With DriveGui maybe there are no values visible in the MotionTask table No. 201,210,211

Then make a recheck with the DriveGui terminal:

```
--> ORDER 201
```

```
201 -400 1000 -1 -1 -1 -1 -1 -1 400 -1
```

```
--> ORDER 210
```

```
210 -200 1500 -1 100 50 -1 -1 -1 -1 -1
```

```
--> ORDER 211
```

```
211 -500 850 -1 150 250 -1 -1 -1 -1 -1
```

```
-->
```

Blocks for Servostar parameters

The S7 project „Sv14_v3c“ contains more function blocks and data blocks that write complete data areas into the Servostar, read out from the Servostar und compare in the S7-PLC.

A record consists of 3 parameters:

- PNU – parameter number – indicates the respective parameter
z.Bsp PNU 1783 – acceleration 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 the FB14 Axis_01 from the S7-PLC over the non real time PKW parameter channel in Profibus to the Servostar.

Declaration of DB20 – for 100 values

| Address | Name | Type | Initial value | Comment |
|---------|---------|-----------------------|---------------|--|
| 0.0 | | STRUCT | | |
| +0.0 | Res_0_7 | ARRAY[0..7] | | |
| +1.0 | | BYTE | | |
| +8.0 | Data | ARRAY[1..100] | | StartAddress for DataNumber 1 is 8 !!! |
| +8.0 | | "M_Axis_01_PnuStruct" | | |
| =808.0 | | END_STRUCT | | |

DB 21 - Axis_01_ReadDataDB

contains the data for reading with FB17 Axis_01_Read and the FB14 Axis_01 from the Servostar over the non real time PKW parameter channel in Profibus to the S7-PLC.

Declaration of DB21 – for 100 values

| Address | Name | Type | Initial value | Comment |
|---------|---------|-----------------------|---------------|--|
| 0.0 | | STRUCT | | |
| +0.0 | Res_0_7 | ARRAY[0..7] | | |
| +1.0 | | BYTE | | |
| +8.0 | Data | ARRAY[1..100] | | StartAddress for DataNumber 1 is 8 !!! |
| +8.0 | | "M_Axis_01_PnuStruct" | | |
| =808.0 | | END_STRUCT | | |

See OB1 and VAT14

FB16 - Axis_01_Write

controls the FB14 to write data from S7-PLC to the Servostar

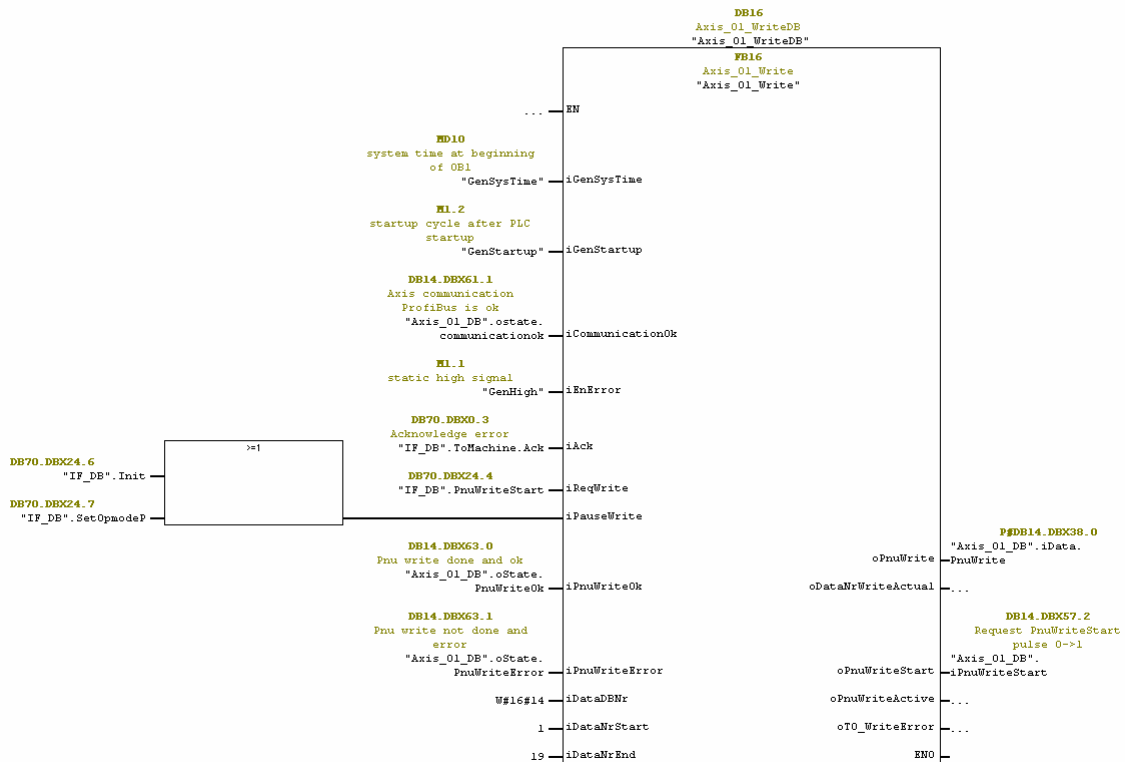
Input and output interface of FB16 - Axis_01_Write

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

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

Network 5: Axis_Write

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



FB17 - Axis_01_Read

controls the FB14 to read data from the Servostar to the S7-PLC

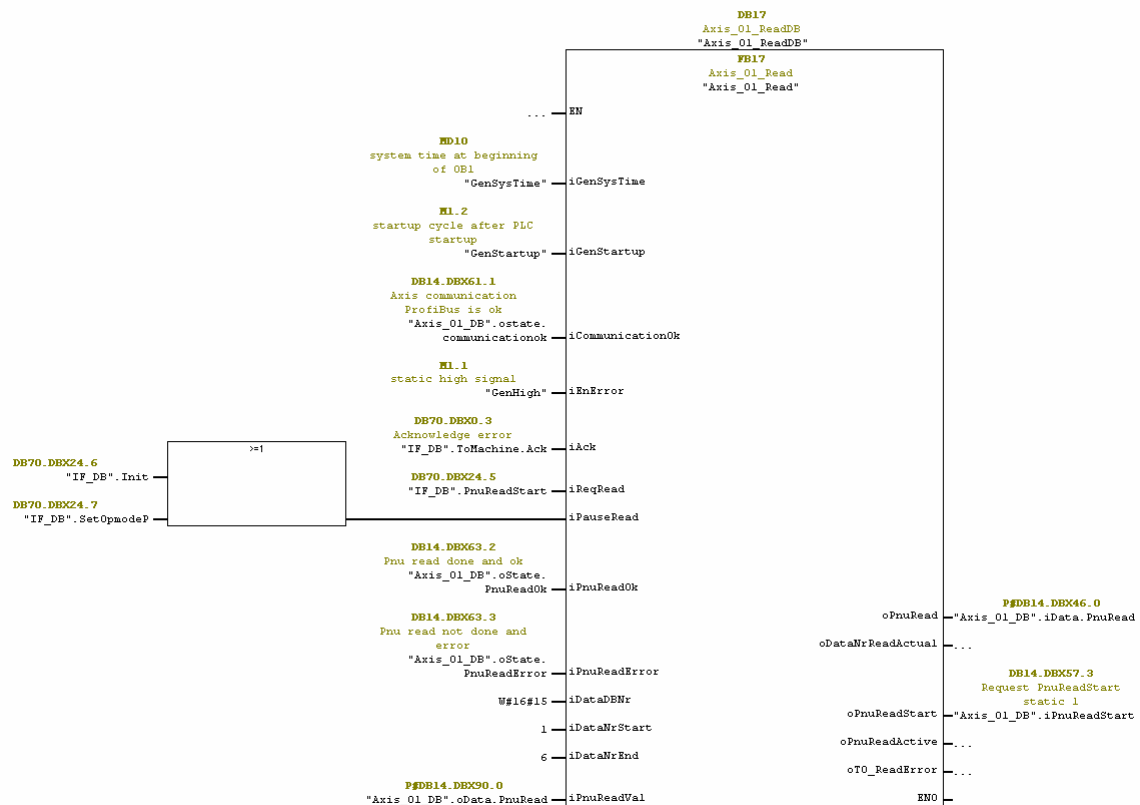
Input and Output interface from the FB17 - Axis_01_Read

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

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

□ Netzwerk 8: Axis_Read

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



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

FB18 - Axis_01_Compare

Compares a data area from the written data from DB20 Axis_01_WriteDataDB with the read data from DB21 Axis_01_ReadDataDB.

Only one PNU [STRUCT] is compared for each S7-PLC cycle.

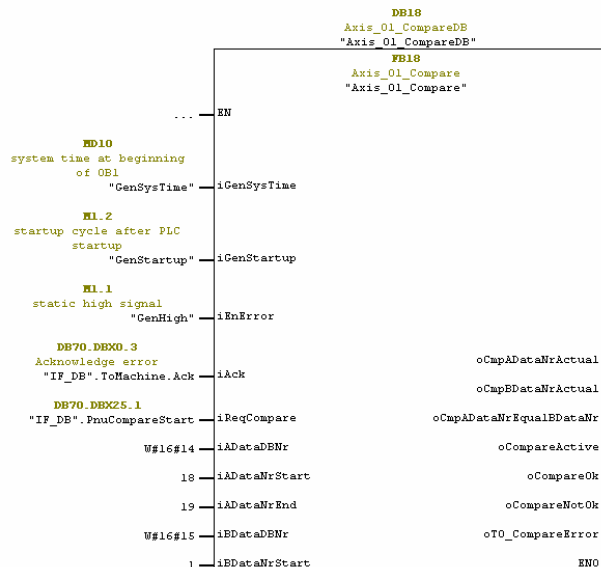
Input and output interface of FB18 Axis_01_CompareDB

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

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

Netzwerk 10: Axis_Compare

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



With the VAT14, the blocks FB14 Axis_01_FB, FB16 Axis_01_Read, FB17 Axis_01_Write and FB18 Axis_01_Compare can be controlled and monitored.

Tips and information:

Error messages and warnings from the blocks FB14, FB16, FB17 and FB18 must be implemented so that the S7-PLC program can respond to them.

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

Override via Profibus

See ASCII list OVERRIDE

The override-function allows controlling the speed of a MotionTask, the reference speed and the jogging speed.

OVRIDE=0 Override function deactivated

OVRIDE=3 Profibus for the digital Override function activated

See ASCII list DOVRIDE:

When activating the digital Override function, this parameter is used to predefine the digital Override factor.

The following scaling applies:

DOVRIDE=0 MotionTask speed = 0 %

DOVRIDE=8192 MotionTask speed = 100 %

The Override function is not possible for sin^2 – ramps !

Activating the sin² motion profile for DirectMotionTask No. 0

The motion profile Sin² 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 more in detail.

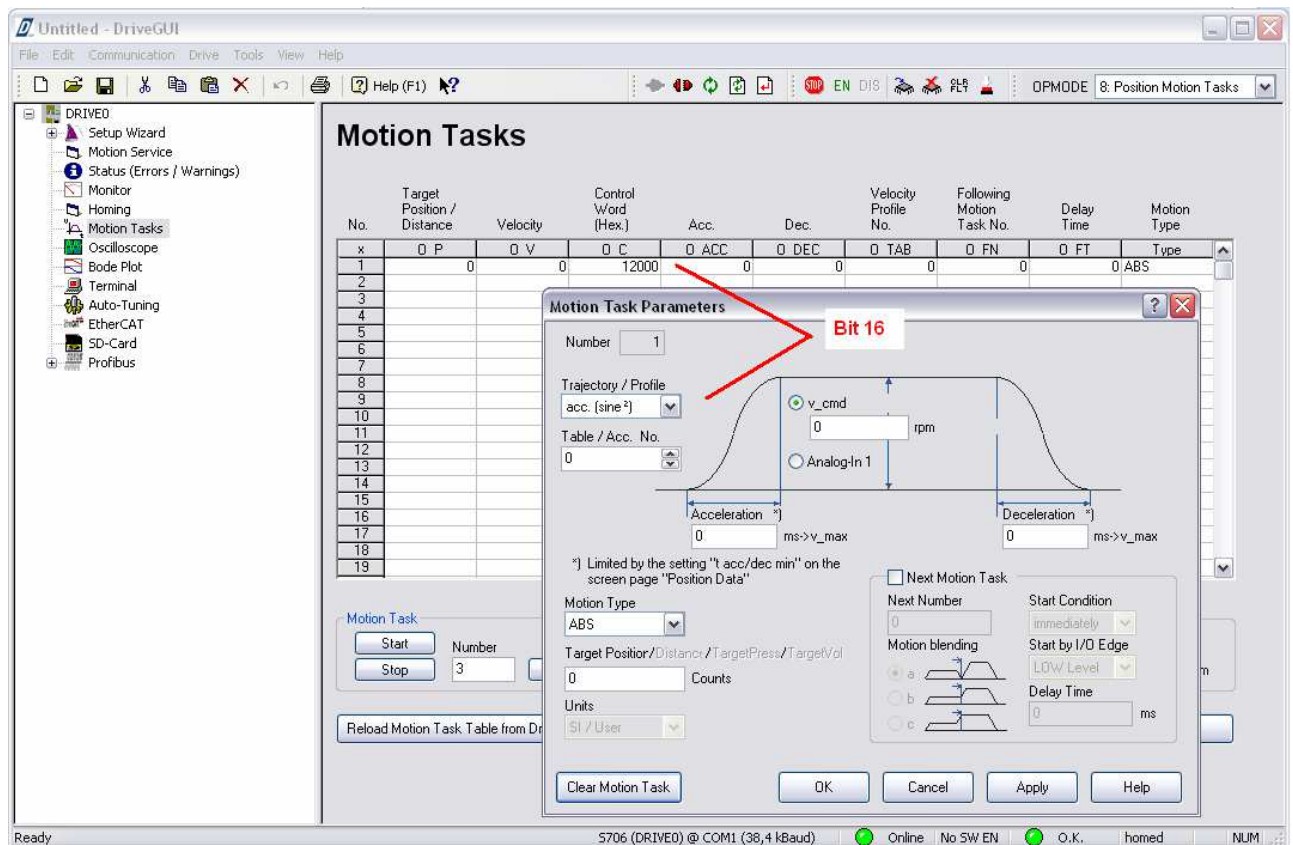
For Servostar S300/S700 – Profibus see ASCII list O_C

The O_C (Profibus PNU 1785) command defines the MotionTaskType for the direct MotionTask No.0. If bit 16 is set by O_C, then the MotionTask has sine² profile. Bit 9 must be set to 0. From the O_C are only the Bits 0 to 15 directly addressed in DirectMotionTaskType (PZD5), thereby the bit 16 must be set separately in S300/S700 by another way.

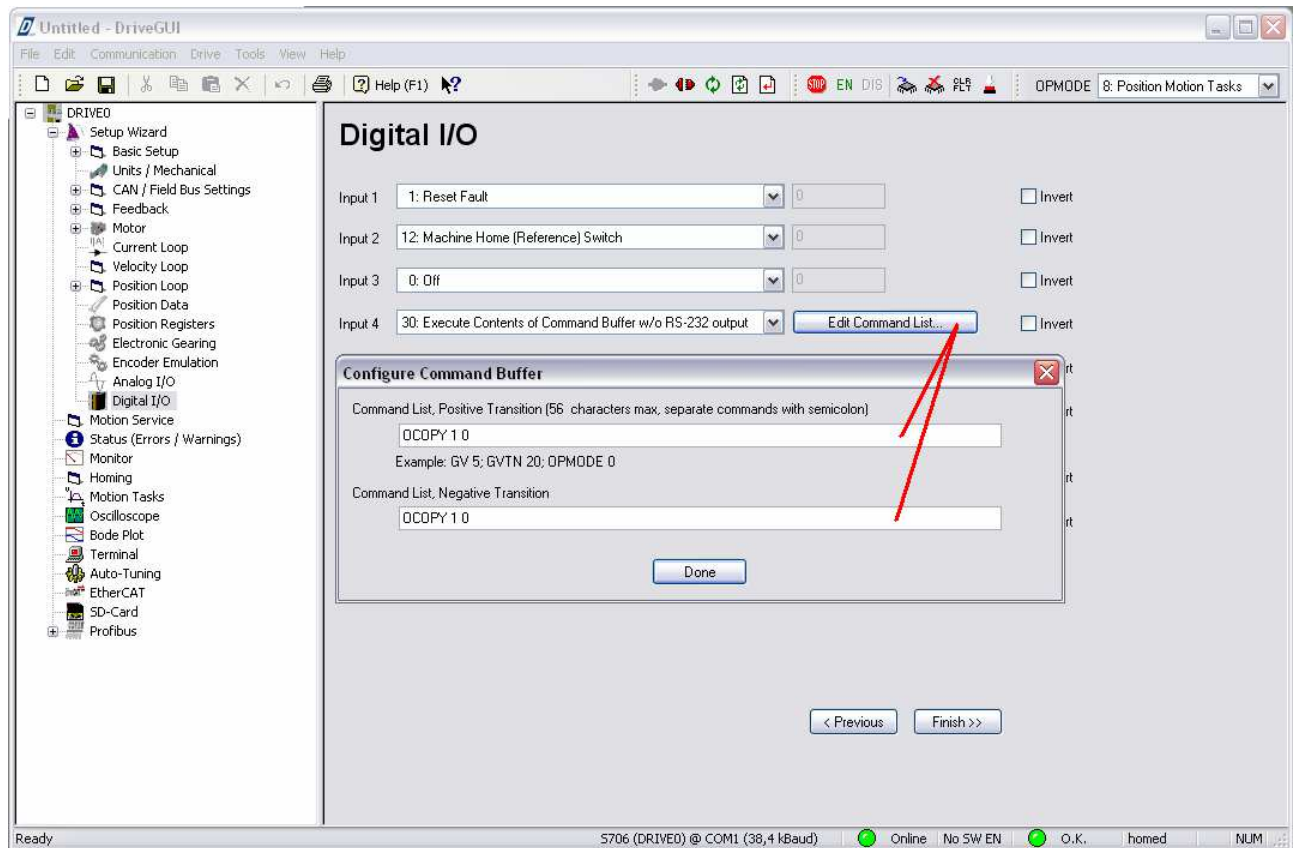
Further the O_C command is not a parameter and is not storable in the Servostar.

O_C - Bit 16 can be set in this ways:

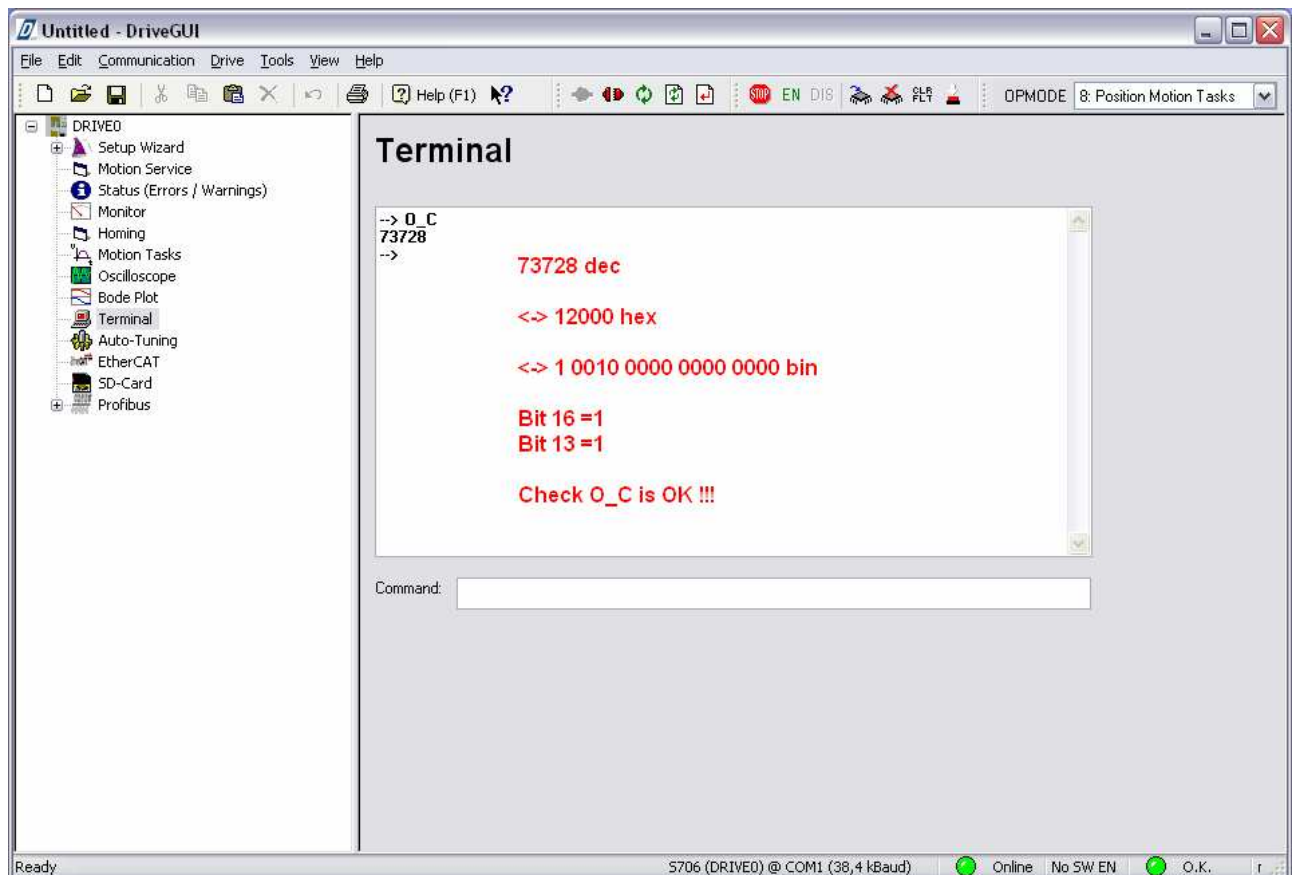
- 1.) Use the PKW parameter channel with PNU 1785 to write the value 10000hex to S300/S700
- 2.) After switching ON the 24-VDC supply voltage, the S300/S700 automatically copies at EEPROM MotionTask with Bit 16 =1 (with motion profile Sin²), e.g. no.1, to the DirectMotionTask No. 0. This allows the S7 program, to remain unchanged and PNU 1785 parameter download is not necessary. In addition, the acceleration ramp O_ACC and deceleration ramp O_DEC can also be set for smooth motion. For value 0, the value PTMIN applies or the largest ramp in each case.



An unused digital Input (not wired) is assigned the OCOPY function from EEPROM MotionTask e.g. No.1 to direct MotionTask No. 0



SAVE and COLDSTART and then check the content of O_C with DriveGui terminal



3.) ASCII command sequence

The command ASCII INxHCMD / INxLCMD is used to define an ASCII command sequence. This command sequence will always be carried out when a rising/falling edge is detected at the input x that has been configured with the function INxMODE=30

A command sequence consists of individual ASCII commands, separated by a semicolon (;)

The maximum length of this command sequence is 56 characters.

With DriveGui-Terminal:

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

Recheck with DriveGui-Terminal:

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

4.) Macro program

Example:

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

Recheck with DriveGui-Terminal:

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

Note: The recheck with DriveGui terminal - there is also possible this result with the same functionality:

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

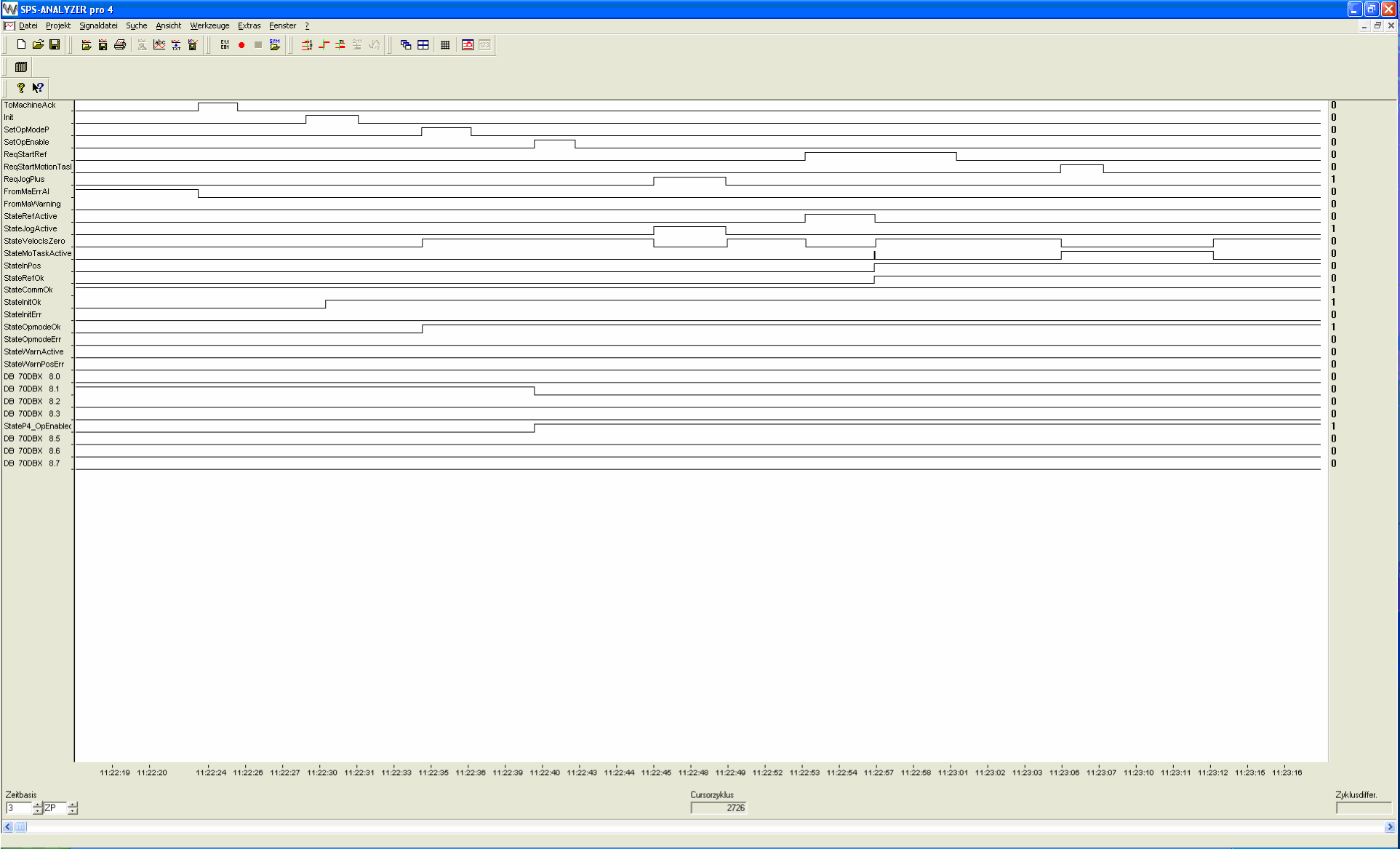
Bitcoding of error – PNU 1001 - ERRCODE

| | | | |
|------------------------|----------|-------|--|
| Axis_PNU1001 | MD 1000 | DWORD | Faults / Error |
| 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 | Error I2tmax exceeded |
| Axis_PNU1001_Bit13_F14 | M 1002.5 | BOOL | Error 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 state register – PNU 1002 - DRVSTAT

| | | | |
|------------------------|----------|-------|---|
| Axis_PNU1002 | MD 1004 | DWORD | Warnings AND manufacturer-specific state 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 | Velocity = 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 (see above) |
| Axis_PNU1002_Bit22 | M 1005.6 | BOOL | Position 1 reached (see above) |
| 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 | Actual position = home position (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 | MotionTask active - motion set, jogging mode, reference move |
| 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 motion task |
| Axis_PNU1002_Bit7_n08 | M 1007.7 | BOOL | Warning 8: A defective motion task 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 and Start MotionTask timing diagram



Kill MotionTask timing diagram

