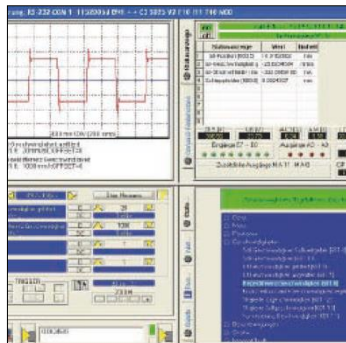




Application Note

Electromechanical Automation Europe
Application Team Offenburg

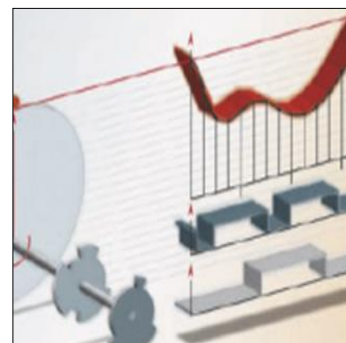


aerospace
climate control
electromechanical
filtration
fluid & gas handling
hydraulics
pneumatics
process control
sealing & shielding



PSD - Communication

ControlManagerAdvanced FB51 for TIA-Portal



Author: K. Woloschin
Application: PSD_Comm_AN0036
Version: V1.1
Last change: 16 July 2018



ENGINEERING YOUR SUCCESS.



PSD - Communication

Warranty Disclaimer

While efforts were made to verify the accuracy of the information contained in this documentation, Parker expressly disclaims all warranties with regard to this application note, including, but not limited to, the implied warranties of merchantability and fitness of a particular purpose. Parker does not warrant, guarantee, or make any representation regarding the use or the results of the use of this application note in terms of correctness, accuracy, or reliability. The contents of this application note are subject to change without notice. Parker will publish updates and revisions of this document as needed. The documents supersedes all previous versions.

Limitation of Liability

You agree that Parker shall not be liable to you under this agreement for any damages, including without limitation any lost profits, or any consequential, incidental, or punitive damages arising out of the use or inability to use this application note and related documents, or for any claim by another party. You agree and hold Parker harmless for all claims and damages from any third party as a result of their use or inability to use any product that you develop based on this application note and the products and/or services documented herein.

Parker Hannifin Manufacturing Germany GmbH&Co.KG

Electromechanical Automation Europe

Business Development & Applications

Copyright © 2010 - 2018

All rights reserved.



PSD - Communication

Table of Contents

1.	DEVICE SUPPLEMENT	4
1.1.	With the option.....	4
1.2.	And the master plc	4
2.	PURPOSE OF THE BLOCK	4
2.1.	Overview	4
2.2.	Restrictions and application.....	4
2.3.	History of modifications	4
3.	SETTING UP	5
3.1.	PSD configuration.....	5
3.2.	SIMATIC - HW Config	5
3.3.	Application interface of "PSDControlManagerAdv"	7
4.	APPLICATION EXAMPLE	12
4.1.	Overview of the connection:.....	12
4.2.	Parameter channel	13

PSD - Communication

1. Device supplement

1.1. With the option

- PSD1S....B2
- PSD1M1....B2

1.2. And the master plc

- SIMATIC S7-1500
- SIMATIC S7-1200

2. Purpose of the Block

2.1. Overview

Absolute	Symbol	Comment	Vers	Date	Device	Application
FB51	PSDControlManager Adanced For TIA -Portal	PSD1..B2	1.1	2017-10-12	PSD	MoveAbs/MoveRel MoveVelocity Gearing

2.2. Restrictions and application

This block simplifies the control of a PSD DRIVE (with PROFIdrive profile) with the S7. The block needs the commands and set values. The block distributes messages and actual values form the drive. The channels PZD and PKW are used in both directions.



This block is only usable with PPO14 (APA_IO = 8BYTE, PD_IO = 18BYTE)

2.3. History of modifications

V1.1 2017-10-12 First version

PSD - Communication

3. Setting up

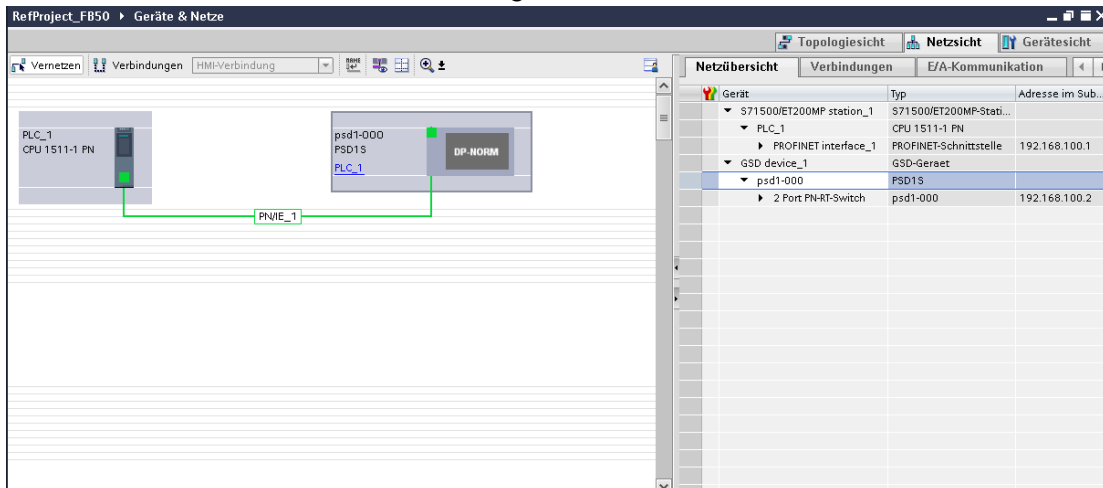
3.1. PSD configuration

With PSDServoManager perform following adjustments:

PROFINET-Settings	
Selected Operation mode	2. Positioning mode with submode MDI (Manual Data Input)
Acyclic Parameter Access (APA)	ParameterChannel_4words
FBI timeout 0x7583	Stop_drive_disabled
PROFINET-Mapping	
APA_IO [16bit words]	4
PD_IO [16bit words]	9
R-PDO: 9 words	
RPDO 1	0x6040.0 Control word 1 (STW1, PNU 967) [16b]
RPDO 2	0x607A.0 MDI target position (MDI_TARPOS) [incr] [32b]
RPDO 3	0x6081.0 MDI velocity (MDI_VELOCITY) [incr/s] [32b]
RPDO 4	0x6083.0 MDI acceleration [incr/s ²] [32b]
RPDO 5	0x6084.0 MDI deceleration [incr/s ²] [32b]
T-PDO: 9 words	
TPDO 1	0x6041.0 Status word 1 (ZSW1, PNU 968) [16b]
TPDO 2	0x6064.0 Position actual value A (XIST_A) [incr] [32b]
TPDO 3	0x606C.0 Speed actual value B (NIST_B) [incr/s] [32b]
TPDO 4	0x0007.0 DummyObj_32Bit [32b]
TPDO 5	0x0007.0 DummyObj_32Bit [32b]

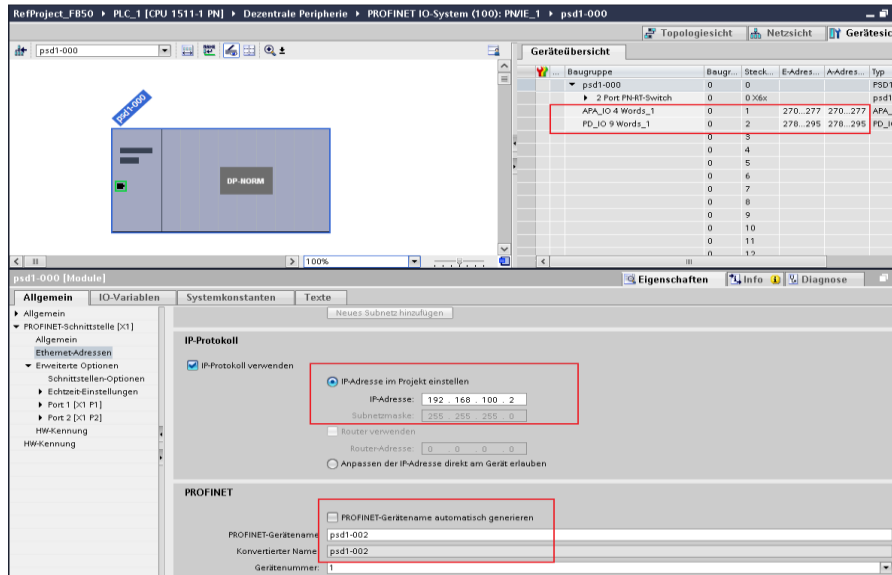
3.2. SIMATIC - HW Config

Add PSD device from Hardware catalog



Assign IP-address and device name. Check device module configuration (APA_IO =4 words, PD_IO = 9 words).

PSD - Communication



Baugruppe	Steck...	E-Adres...	A-Adres...	Typ
psd1-000	0	0		PSD1S
2 Port PN-RS-Switch	0	0 X6x		psd1-
APA_IO 4 Words_1	0	1	270..277	APA_I...
PD_IO 9 Words_1	0	2	278..295	PD_IO...

IP-Protokoll

☒ IP-Protokoll verwenden

☒ IP-Adresse im Projekt einstellen

IP-Adresse: 192.168.100.2

Subnetzmaske: 255.255.255.0

☐ Router verwenden

Router-Adresse: 0.0.0.0

☐ Anpassen der IP-Adresse direkt am Gerät erlauben

PROFINET

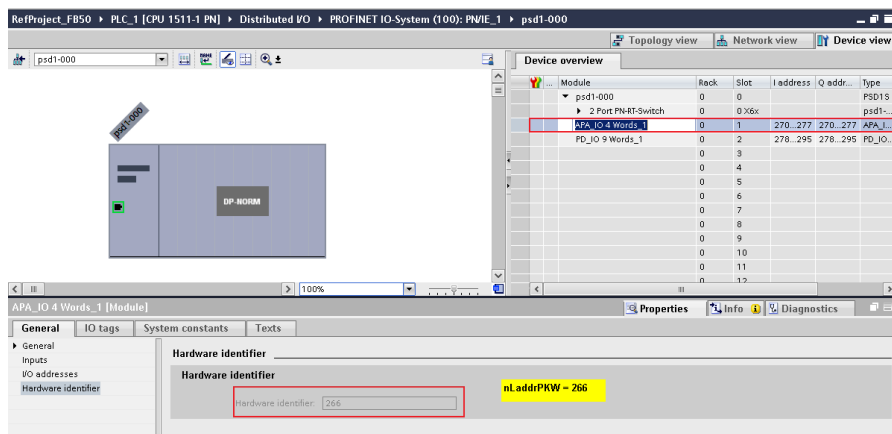
☐ PROFINET-Geräte-Name automatisch generieren

PROFINET-Geräte-Name: psd1-002

Konvertierter Name: psd1-002

Gerätenummer: 1

Adjust hardware identifier for APA_IO in <nLaddrAPA> (DB51.DBW8).



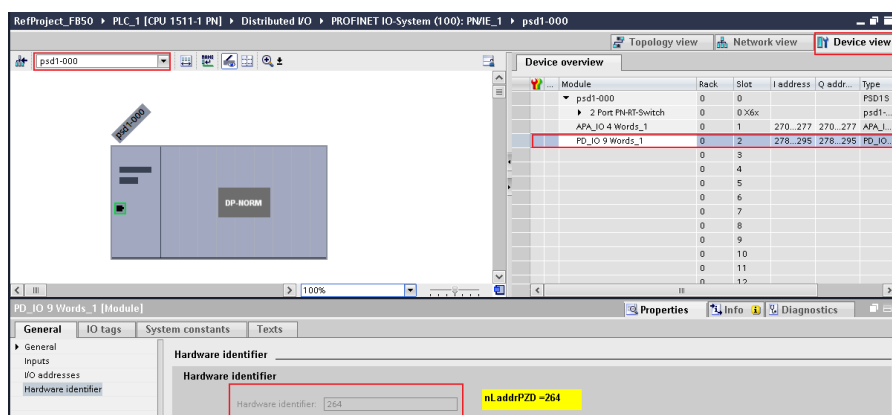
Module	Rack	Slot	I address	Q address	Type
psd1-000	0	0			PSD1S
2 Port PN-RS-Switch	0	0 X6x			psd1-
APA_IO 4 Words_1	0	1	270..277	270..277	APA_I...
PD_IO 9 Words_1	0	2	278..295	278..295	PD_IO...

Hardware identifier

Hardware identifier: 266

nLaddrPKW = 266

Adjust hardware identifier for PD_IO in <nLaddrPD> (DB51.DBW6).



Module	Rack	Slot	I address	Q address	Type
psd1-000	0	0			PSD1S
2 Port PN-RS-Switch	0	0 X6x			psd1-
APA_IO 4 Words_1	0	1	270..277	270..277	APA_I...
PD_IO 9 Words_1	0	2	278..295	278..295	PD_IO...

Hardware identifier

Hardware identifier: 264

nLaddrPZD = 264

PSD - Communication

3.3. Application interface of "PSDControlManagerAdv"

3.3.1. Schematic drawing for in- and output Areas of FB51 / DB51

in input	out output	in_out In- and Output	stat Static Memory
----------	------------	-----------------------	--------------------

DB51

	EN	ENO	
DBX0.0	bEnable	bEnabled	DBX32.0
DBX0.1	bHold	bDriveErr	DBX32.1
DBX0.2	bStop	bModeRunning	DBX32.2
DBX0.3	bFaultReset	bInMode	DBX32.3
DBX0.4	bJogP	bPosErr	DBX32.4
DBX0.5	bJogN	bHomingRunning	DBX32.5
DBX0.6	bExDataTransfer	bHomingAttained	DBX32.6
DBX2.0	nWord	bHomingErr	DBX32.7
DBX4.0	nLaddrPD	bCommErr	DBX33.0
DBX6.0	nLaddrAPA		
DBX8.0	iPosition		
DBX12.0	iVelocity		
DBX16.0	iAcceleration		
DBX20.0	iDeceleration		
DBX24.0	iNumerator		
DBX28.0	iDenominator		
DBX44.0	bStartPositioning		
DBX44.1	bChangeSetImmediate	stRd.iPositionValue	DBD34
DBX44.2	bStartHoming	stRd.iVelocityValue	DBD38
DBD26	iInPosWindowAbs	stRd.nActualError	DBW42
DBW30	nCmd		
DBW32	nObjectIndex	bTransErr	DBX33.1
DBW34	nObjectSubindex		
DBD36	iParameterValue		
DBD40	TonTimer1	iParameterValue	DBD56
DBD44	TonTimer2		
DBX56.0	stPSD_APAInDint.nPKE		
DBX72.0	stPSD_APA_In.nStatus		

PSD - Communication

3.3.2. Declaration of In- and Output

Parameter	Declaration	Data type	description
nMode	IN	BOOL	=1 absolute, =2 relative, =4 velocity, =8 gearing
bEnable	IN	BOOL	=1 energise =0 disenergize with AUS3 - Ramp (Not-Stop)
bExDataTransfer	IN	BOOL	=0 internal DP interface with DPWR_DAT/DPRD_DAT (internal Master in S7 CPU) =1 external DP interface with FC2/FC1 (external Master CP 342-5)
bFaultReset	IN	BOOL	Acknowledge with rising edge, after that it is necessary to activate energise <bEnable> (caused by the edge detection it is needed to be set to 0 first)
bHold	IN	BOOL	=1 Temporary stop (the movement function is still available), =0 continue
bJogN	IN	BOOL	manual negative: JOG – movement within positioning end limits as long as true
bJogP	IN	BOOL	manual positive: JOG – movement within positioning end limits as long as true
bPositionResetMode	IN	BOOL	=0 Normal-, =1 Reset mode selected (in PSDMgr \ drive Configuration \ units... positioning reset distance and positioning reset distance denominator is different from 0 (Not supported with PSD drives yet))
bStop	IN	BOOL	=1 Stop (movement function cancelled)
bChangeSetImmediate	IN_OUT	BOOL	With the Rising edge, a new position profile is activated; <bChangeSetImmediate> is reseted from the block itself. This command is acknowledged from the block with <bModeRunning>. A new command is also accepted if the actual movement is not finished (<bInMode> =1).
bStartHoming	IN_OUT	BOOL	rising edge starts referencing movement, if permitted, neg. edge stops referencing movement, bStartHoming may only be reseted with <bHomingAttained>.
bStartMode	IN	BOOL	With the Rising edge a new profile is activated, <bChangeSetImmediate> is reseted from the block itself. This command is acknowledged from with <bModeRunning>. A new command is not accepted if the actual movement is not finished (<bInMode> =1).

PSD - Communication

Parameter	Declaration	Data type	description
bCommErr	OUT	BOOL	=1 communication failure (Failure from DPWR_DAT / DPRD_DAT) (all other messages are invalid)
bDriveErr	OUT	BOOL	=1 failure from PSD (device / motor)
bEnabled	OUT	BOOL	=1 Axis energised =0 Axis not energised
bHomingAttained	OUT	BOOL	=1 reference ok.
bHomingErr	OUT	BOOL	=1 watchdog/ timeout for Reference -run (occasionally TonTimer2 correct)
bHomingRunning	OUT	BOOL	=1 reference run active
bInMode	OUT	BOOL	=1 axis in target position, in gear, in velocity
bPosErr	OUT	BOOL	=1 watchdog timeout for position -order (occasionally. TonTimer1 correct)
bModeRunning	OUT	BOOL	=1 active
bTransErr	OUT	BOOL	=1 format-, commando failure at transfer from / to PSD
iAcceleration	IN	DINT	acceleration in U32 -Format
iDeceleration	IN	DINT	deceleration in U32 -Format
iInPosWindowAbs	STATIC	DINT	Position window in additional monitoring of <bInPosition> with absolute positioning. The actual position value compared with the position set value. That is only possible with absolute positioning (<bAbsoluteRelative> = false) and position Reset mode (<bPositionResetMode> = false) is not activated.
iParameterValue	STATIC	DINT	PSD object value (source and destination) - write-/read value of the Parameters
iPosition	IN	DINT	target position / distance in increments
iVelocity	IN	DINT	speed in increments
stRd.iPositionValue	OUT	DINT	actual position in increments
stRd.iVelocityValue	OUT	DINT	actual speed in increments
nCmd	STATIC	INT	PSD object transfer: command: 1 read 2 write WORD 3 write DWORD
nObjectIndex	STATIC	INT	PSD object index
nObjectSubindex	STATIC	INT	PSD object subindex
TonTimer1	STATIC	TIME	time value for timeout of Positioning
TonTimer2	STATIC	TIME	time value for timeout Reference run

PSD - Communication

Parameter	Declaration	Data type	description
nLaddrPD	STATIC	WORD	= Hardware identifier of the (PD_IO) Process Data module.
nLaddrAPA	STATIC	WORD	= Hardware identifier of the APA_IO (Acyclic Parameter Access) data module.
stRd.nActualError	STATIC	WORD	Actual Failure (see PSD - Manual) in WORD - Format =1 no Error!
stPSDPKWInDint.nPKE	STATIC	Word 4	Local PKW - input area for external CP
stPSDPKWOutDint.nPKE	STATIC	Word 4	Local PKW - output area for external CP
stPSDPZDIn.nStatus	STATIC	Word 9	Local PZD - input area for external CP
stPSDPZDOut.nControl	STATIC	Word 9	Local PZD - output area for external CP

3.3.3. Sequence of process data

3.3.3.1. Settings of the static Operands at the Block

1. <nLaddrPD>and<nLaddrAPA>

- Parameter from (System constants): HW-ID of the IO-device for PD and APA channel.
- Forward the values to <nLaddrPD> and <nLaddrAPA>.

2. <bExDataTransfer>

- Low: settings for CPU with integrated Master
- High: settings for CPU with external Master

3. <ilnPosWindowAbs>

- additional control window for the message <bInPosition /bInMode>

4. <TonTimer1>

- Time value for watchdog positioning, if this time is too short there is shown the error message <bPosError>.

5. <TonTimer2>

- Time value for watchdog homing, if this time is too short there is shown the error message <bHomingErr>.

3.3.3.2. Settings of dynamic Operands at the Block

1. Switch on

- Set <bEnable>: the block notifies <bEnabled>

PSD - Communication

- Set *<bStartHoming>*, the block notifies *<bHomingRunning>*. If the homing finished, the block notifies *<bHomingAttained>*.
 - Now reset *<bStartHoming>*.
 - At reaching home position, the message *<bInPosition>* is set.
 - With some home modes (e.g. MN-M 35) and high velocity *<bHomingRunning>* is set so short, that is not visible.
 - If there is a motor with absolute position feedback (SinCos®), it is only one time needed to activate homing. *<bHomingAttained>* remains at True even if the drive is switch off and on again. Attention: With activating, the homing from PSD–Optimisation *<bHomingAttained>* is not set.
2. Select mode
- Set nMode:
1 = MoveAbsolute
2 = MoveRelative
4 = MoveVelocity
8 = Gearing
3. Positioning
- Set Parameters for Positioning
 - *<nMode=1 or 2>*
 - *<iPosition>*
 - *<iVelocity>*
 - *<iAcceleration>*
 - *<iDeceleration>*
 - Start the positioning with activating: *<bStartMode>*
 - *<bStartMode>* is reseted by the block itself
 - The block notifies *<bModeRunning>*
 - With reaching the target position, the message *<bInMode>* is set.
 - A new target position is only possible after *<bInMode>* was set.
4. Velocity mode
- Set Parameters for Velocity mode
 - *<nMode=4>*
 - *<iVelocity>*
 - *<iAcceleration>*
 - Start velocity mode with activating: *<bStartMode>*
 - *<bStartMode>* is reseted by the block itself
 - The block notifies *<bModeRunning>*
 - With reaching the target velocity, the message *<bInMode>* is set.
5. Gearing mode
- Set Parameters for Gearing mode
 - *<nMode=8>*
 - *<iNumerator>*
 - *<iDenominator>*
 - Start gearing mode with activating: *<bStartMode>*

PSD - Communication

- *<bStartMode>* is reseted by the block itself
- The block notifies *<bModeRunning>*
- With reaching the target velocity, die message *<bInMode>* is set.

3.3.3.3. Other Operands at Block

- *<bFaultReset>* acknowledgement of failures of function block (watchdog) or drive.
- *<bStop>* stops a (mode) movement with rising edge
- *<bHold>* interrupt of positioning command, as long as Bit it is *true*; positioning will be finished when the bit is *false* again.
- *<bJogN>* manual mode negative direction, as long as Bit it is *true*.
- *<bJogP>* manual mode positive direction, as long as Bit it is *true*.

3.3.3.4. Messages and display

- *<bCommErr>* communication with PSD not possible
- *<bDriveErr>* PSD is in failure status
- *<stRd.nActualError>* actual Failure number of PSD (see PSD Help)
- *<stRd.iPositionValue>* actual position
- *<stRd.iVelocityValue>* actual velocity

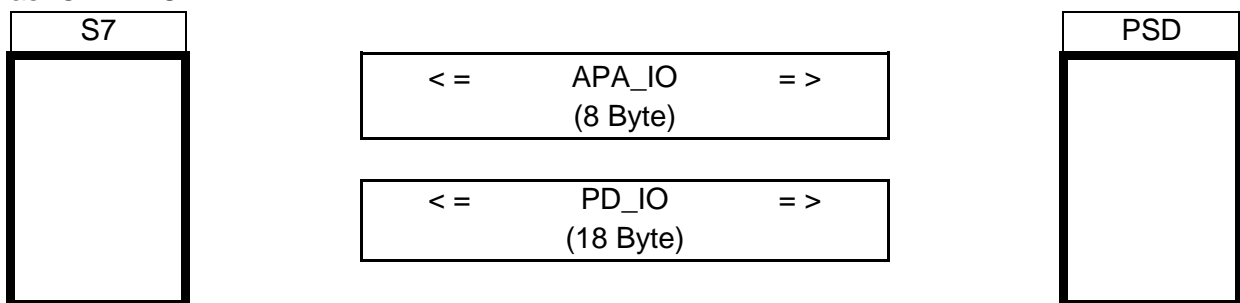
3.3.3.5. Read and write drive objects

- the objects are selected with *<nObjectIndex>* and *<nObjectSubindex>*
- the value is at *<iParameterValue>*
- *nCmd* is the command for the transfer
 - 1 command for read (word and double word)
 - 2 command for write word object (16 Bit)
 - 3 command for write double word object (32 Bit)
- Datatype 16 Bit or 32 Bit is shown in the object info "Datatype".
- *<bTransErr>* shows if there is failure with the data transfer.

4. Application example

4.1. Overview of the connection:

Connection between one PLC SIMATIC S7 1500 as IO-CONTROLLER and one drive PSD as IO-DEVICE.



PSD - Communication

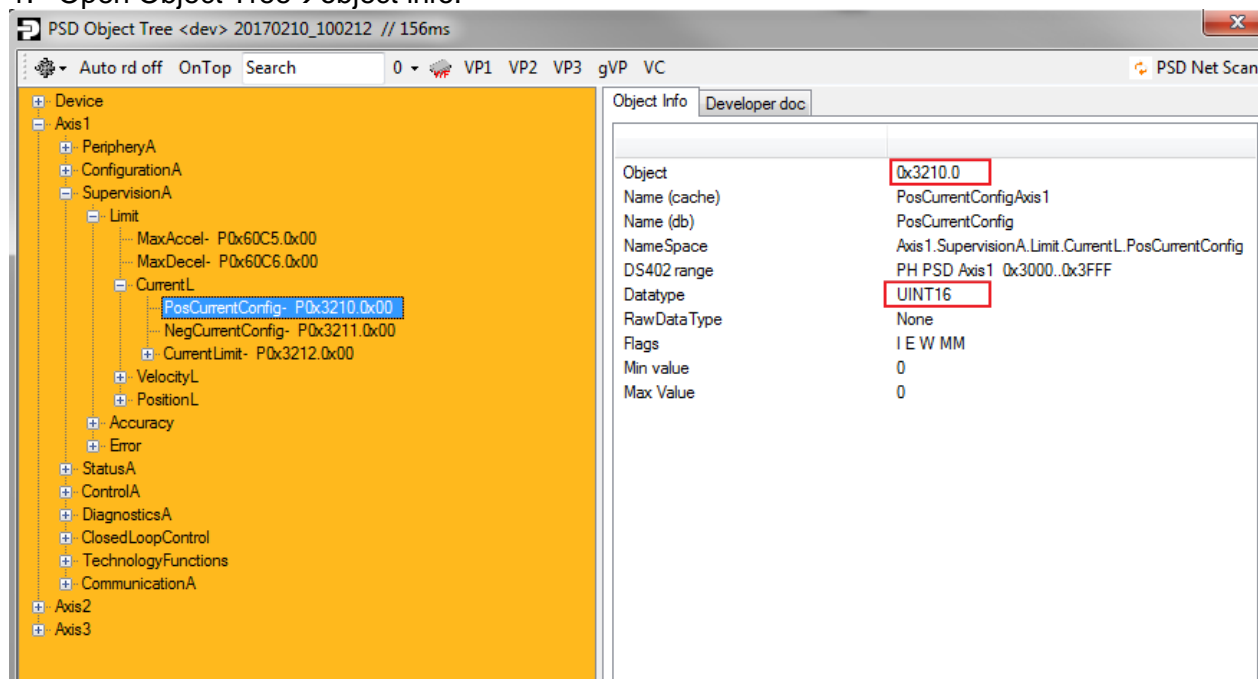
4.2. Parameter channel

4.2.1. Used area

	FB50		
DBW28	nCmd	bTransErr	DBX3.1
DBW30	nPNUIndex		
DBW32	nPNUSubindex		
DBD34	iParameterValue	iParameterValue	DBD34

4.2.2. Procedure for reading and writing of current limit positive object [0x3210.0]

1. Open Object Tree→object info:



2. Set requested object index and subindex:

nObjectIndex <3210>

nObjectSubindex <0>

3. Next edit the commando for "read" (1) !

nCmd <1>

4. Check object value

iObjectValue e.g.<2000>

PSD - Communication

5. Next set new object value
iObjectValue e.g.<1000>
6. Edit new commando for "write 16b" (2) !
nCmd <2>
7. Verify the result in PSD ServoMgr (object value 0x3210.1).



Caution:

VP and WF use system resources, which decrease the effectiveness of internal communication. This may have the effect that there occur error messages like „cycle time overrun“. So you should use these two commands not too often.

Try to use them during the axis is not energized. It would be better to change first a couple of parameters and activate them together by writing VP only one time.