SIEB & MEYER

Drive Controller SD4x

PROFINET IO Connection

A getting started guide for users

D-00000394.3
2024-09-16

Trademarks

All product, font and company names mentioned in this manual may be trademarks or registered trademarks of their respective companies.

SIEB & MEYER Worldwide

For questions regarding our products and technical problems please contact us.

SIEB & MEYER AG
Auf dem Schmaarkamp 21
21339 Lueneburg
Germany

Phone: +49 4131 203 0
Fax: +49 4131 203 2000
info@sieb-meyer.de
http://www.sieb-meyer.com

SIEB & MEYER Shenzhen Trading Co. Ltd.
Room A208 2/F,
Internet Innovation and Creation Services Base Building (2),
No.126, Wanxia road, Shekou, Nanshan district,
Shenzhen City, 518067
P.R. China

Phone: +86 755 2681 1417 / +86 755 2681 2487
Fax: +86 755 2681 2967
info@sieb-meyer.cn
http://www.sieb-meyer.cn

SIEB & MEYER Asia Co. Ltd.
5 Fl, No. 578, Sec. 1
Min-Sheng N. Road
Kwei-Shan Hsiang
Guishan Dist., Taoyuan City 33393
Taiwan

Phone: +886 3 311 5560
Fax: +886 3 322 1224
info@sieb-meyer.tw

1 About this Manual

This manual describes how to connect a drive controller of the SD4x device family to a higher-ranking control (controller) via PROFINET IO. For safety instructions and more information on your SD4x drive refer to the according hardware and software documentation.

A basic description of the PROFINET IO network can be found on the website of the PROFIBUS Nutzerorganisation e.V. (PNO) (www.profibus.com/).

The implemented protocol is based on the CANopen profiles. The CANopen profiles describe the data exchange between the bus participants. The CANopen standard specifies the basic communication profile and the device profiles of the connected bus devices.

The PROFINET IO connection as described in this document provides the following communication mechanisms:

- object dictionary
- mapping of process data objects (PDO)
- service data objects (SDO)
- network management (NMT)

In addition, users can make use of the CANopen device profiles, for example the drive profile CiA 402. For this purpose, refer to the standards of the user organization CiA, see section “Other documents”, page 56.

1.1 Abbreviations

CAN	**C**ontroller **A**rea **N**etwork
CiA	**C**AN **i**n **A**utomation
GSDML	**g**eneral **s**tation **d**escription X**ML** file
NMT	**n**etwork **m**anagement
PDO	**p**rocess **d**ata **o**bject
PLC	**p**rogrammable **l**ogic **c**ontrol
PNS	**P**ROFI**N**ET **s**tation
PV	**p**rofile **v**elocity mode
RO	**r**ead **o**nly (object access)
RW	**r**ead **w**rite (object access)
Rx	receive
SDO	**S**ervice **d**ata **o**bject
Tx	transmit
VL	**v**e**l**ocity mode

2 Parameterization in *drivemaster4*

Before you can operate the SD4x drive in the PROFINET IO network, you must set the PROFINET IO communication via the configuration software *drivemaster4*.

Drive Control

1. Open the *drivemaster4* software and select the page “Drive control” in the device tree under “Parameters”.
2. Select the control channel “PROFINET IO”. The reference value channel (main ref) is automatically set the same.

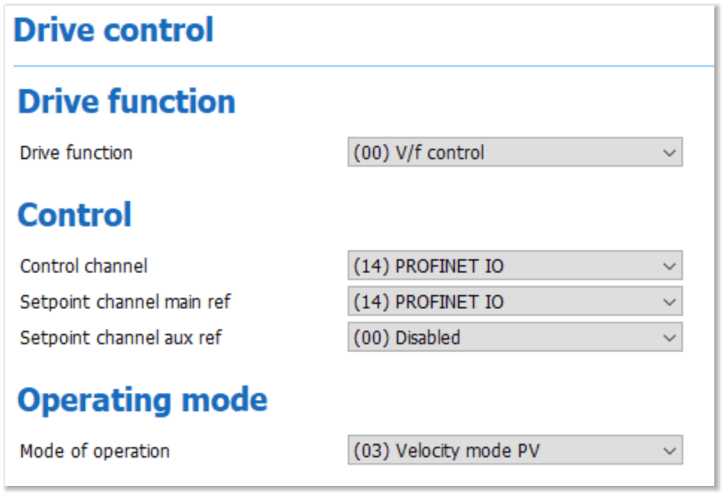


Fig. 1: PROFINET IO parameterization in drivemaster4

The set mode of operation “(03) Velocity mode PV” corresponds to profile velocity mode. For velocity mode you must set the mode of operation to “(02) Velocity mode VL”.

Fieldbus

3. Open the parameter page “Fieldbus” in the device tree of the *drivemaster4* software.
4. Open the tab “PDO mapping” und create the mapping according to the selected operating mode.

   → The following figure shows the mapping for the profile velocity mode:

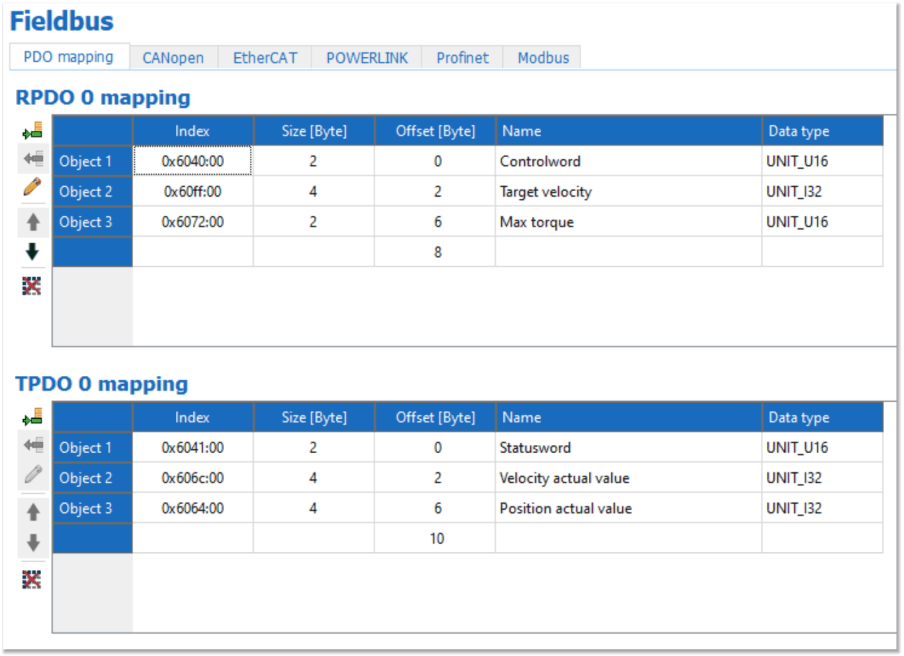


Fig. 2: Example mapping for profile velocity mode

The following figure shows the mapping for the velocity mode:

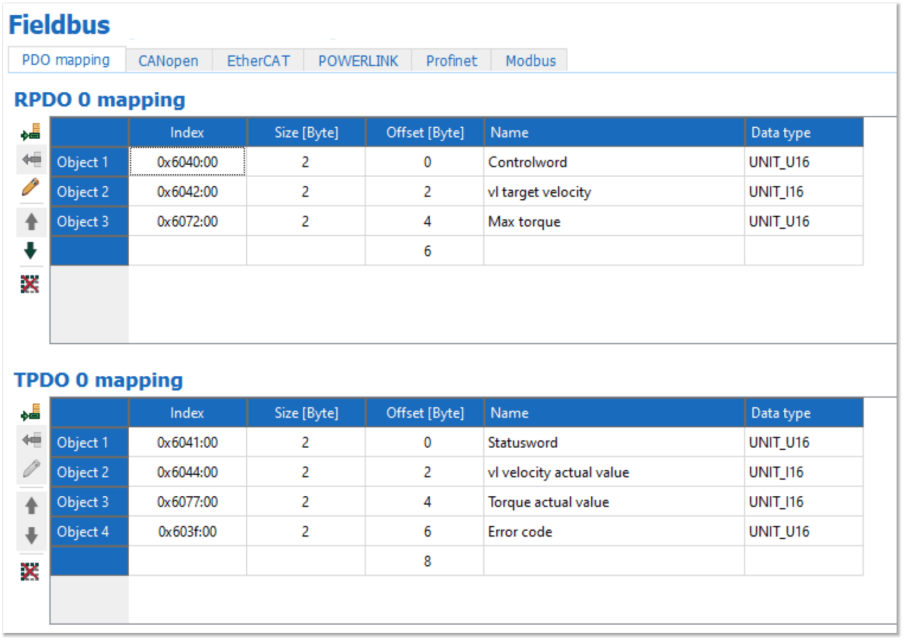


Fig. 3: Example mapping for velocity mode

5. Save the parameters in the drive via the button in the tool bar and restart the drive.

✓ Now, the SD4x drive expects to be supplied cyclically with reference values (PDOs – process data objects) via PROFINET IO.

3 Control via PROFINET IO

The following chapters describe the addressing of the drives in the PROFINET IO network and the connection to the PLC using the Siemens software TIA PORTAL.

3.1 PROFINET IO Network

The PROFINET IO network is ring-shaped. Hereby, the bus is based on the Ethernet protocol and connected via standard Cat 5 network cables.

Within the PROFINET IO network, addressing the SD4x drives is done implicitly by assigning station names in the PROFINET IO controller.

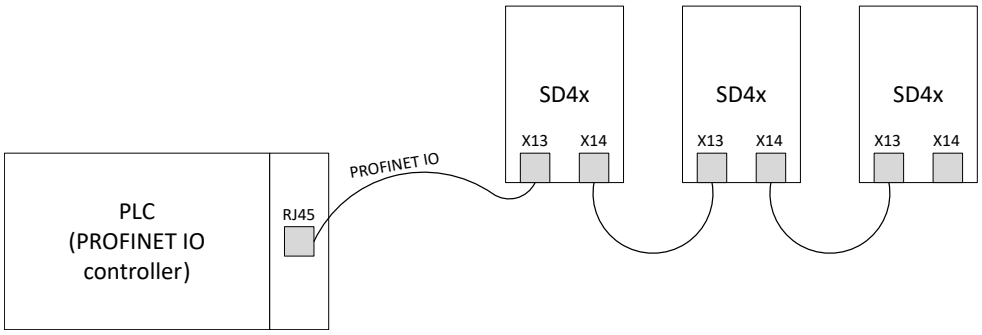


Fig. 4: PROFINET IO network

For information on the connector pin assignment of the PROFINET IO interfaces and the according description of the status LEDs refer to the hardware documentation of the used SD4x drive.

3.2 Software Connection to the PLC

The following example shows the connection to the PLC using the application development system TIA PORTAL by SIEMENS.

1. Copy the GSDML file as well as the BMP file provided by SIEB & MEYER to the project directory
   {drive}:\TIA\workspace\{projectname}\AdditionalFiles\GSD\ .
   - Example GSDML: gsdml-v2.35-sieb_meyer-0400-sd4x-20230510.xml
   - Example BMP: gdsml-0400-0101-sm0sd4x.bmp
2. Start the application development system TIA PORTAL and load the project.

3. To install the GSDML file, select the menu “Options” and then the point “Manage general station description files (GSD)”.

   The following selection window appears.

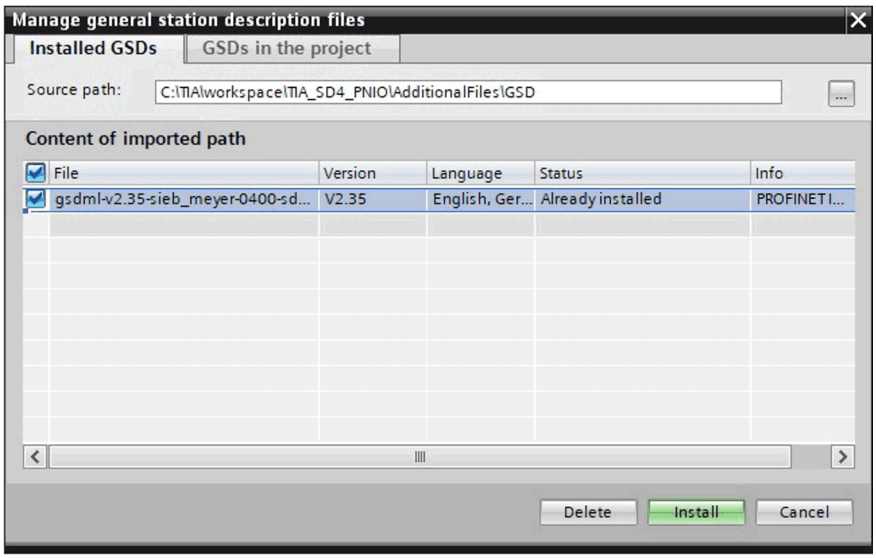


Fig. 5: Manage GSD files

   a. In the top, select the source path, in which the GSDML files are saved.

   b. Select the required file via the check box on the left side and click “Install”.

      → The GSDML file is included in the project, now.

   c. Close the window after successful installation.

      → The hardware catalog is updated now.

4. Switch to the network view.

5. Select the SD4x drive in the hardware (Other field devices → PROFINET IO → I/O → SIEB & MEYER AG → pns → sd4x) und drag it to the network window. .

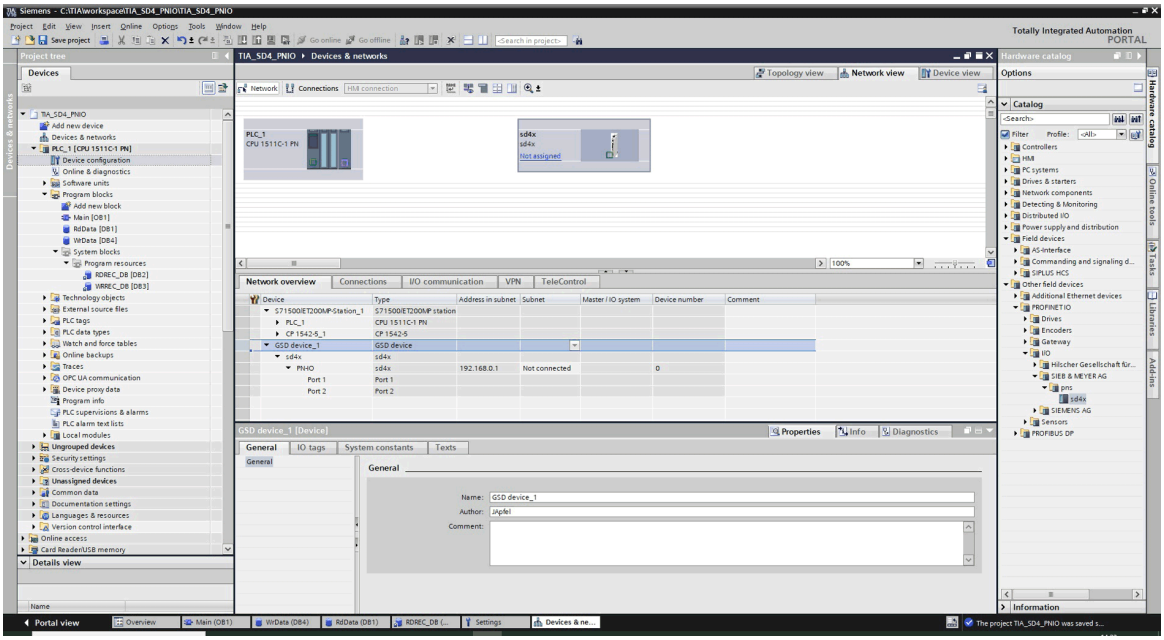


Fig. 6: Network view

6. Right-click on the text “Not assigned” and select the entry “Assign to new IO controller” from the context menu.

   The following window appears:

Fig. 7: Select IO controller

   a. Select the PROFINET interface in the window and click “OK”.

      → The drive is connected with the IO controller now.

7. Switch to the device view.

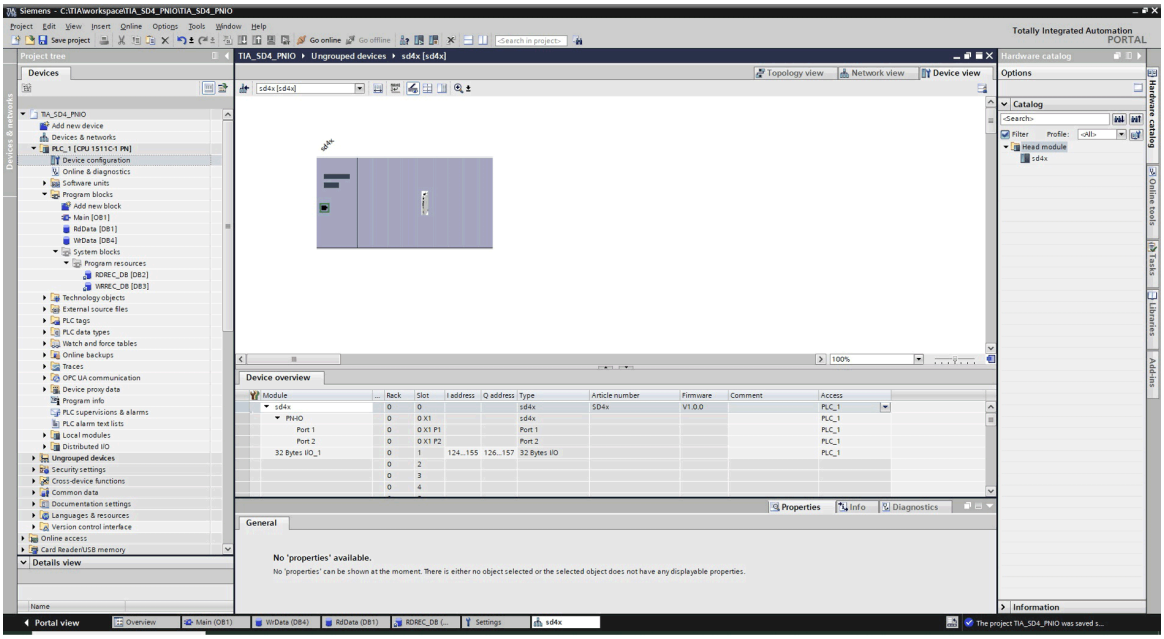


Fig. 8: Device view

8. To rename the drive, proceed as follows:

   a. Click the module name “sd4x” and enter the desired name.

b. Then, assign this device name to the drive via the icon “Name”.

The window “Assign PROFINET device name” appears.

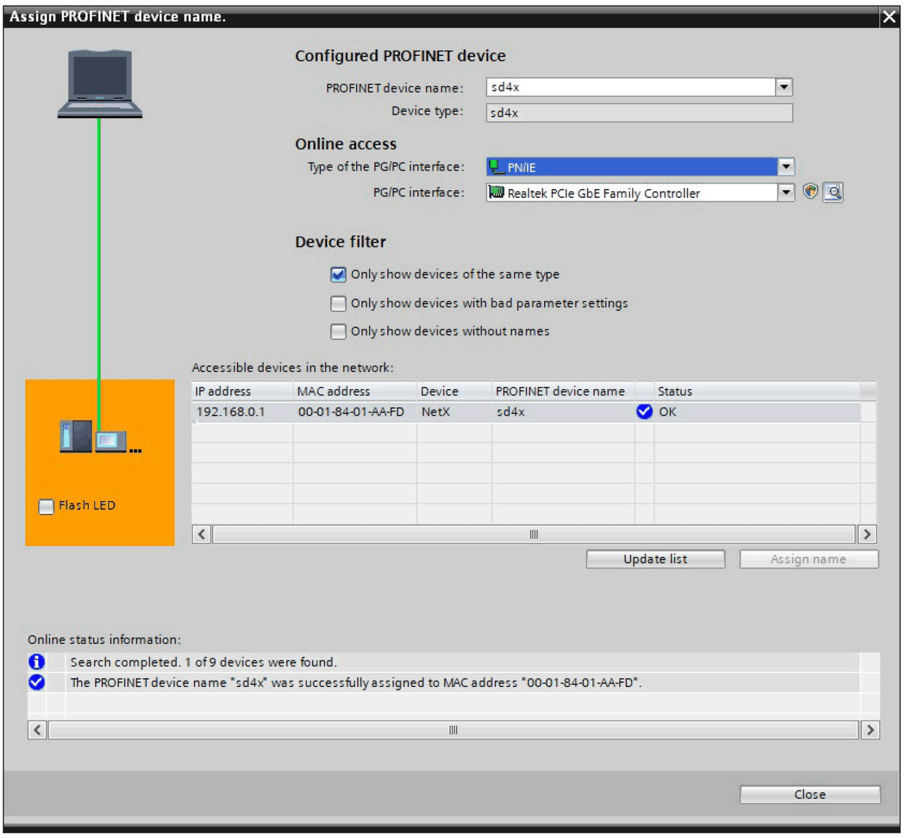


Fig. 9: Assign device name

c. Click the button “Update list”.

→ A list with accessible network devices appears.

d. Select the drive in the list and click the button “Assign name”.

9. Switch to the topology view.

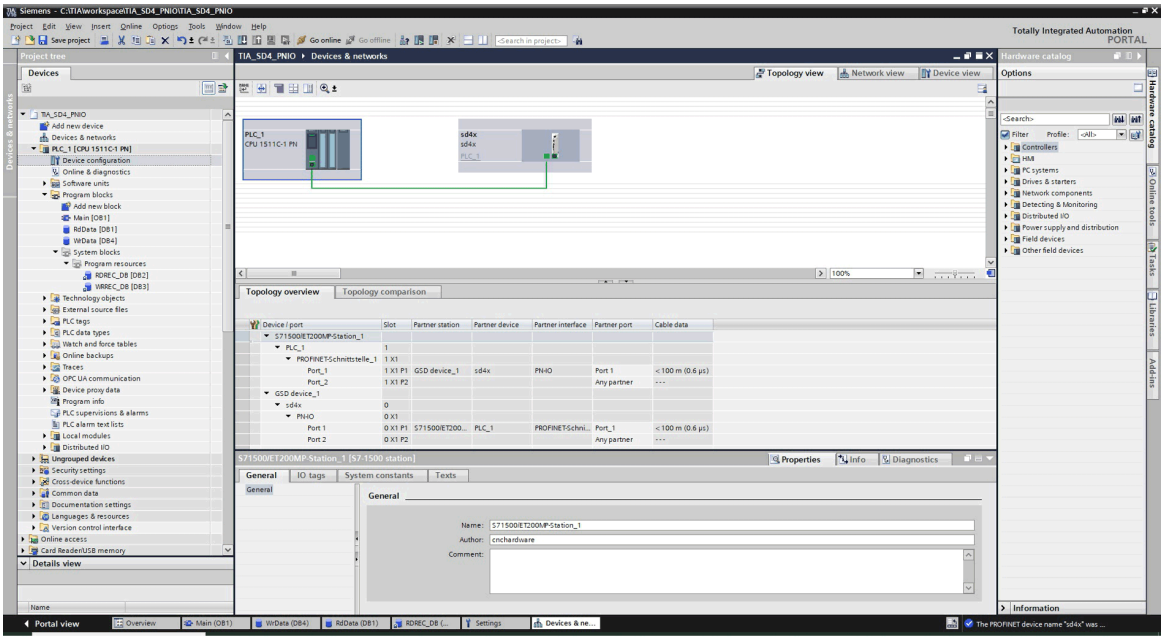


Fig. 10: Topology view

10. Connect the controller ports and the drive port with each other.

    → This gives you additional options for diagnosis.

11. Create new PLC data types for the input ad output data with an “Array[0..31] of Byte”, e.g. DataInStruct and DataOutStruct.

DataInStruct:

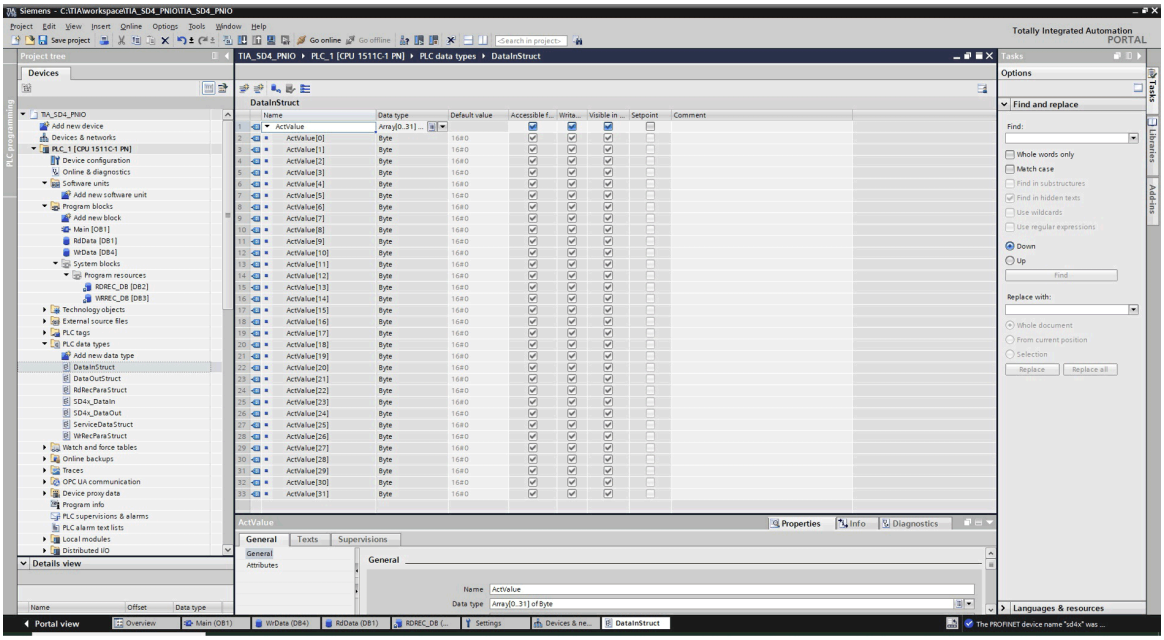


Fig. 11: PLC data types for the input data

DataOutStruct:

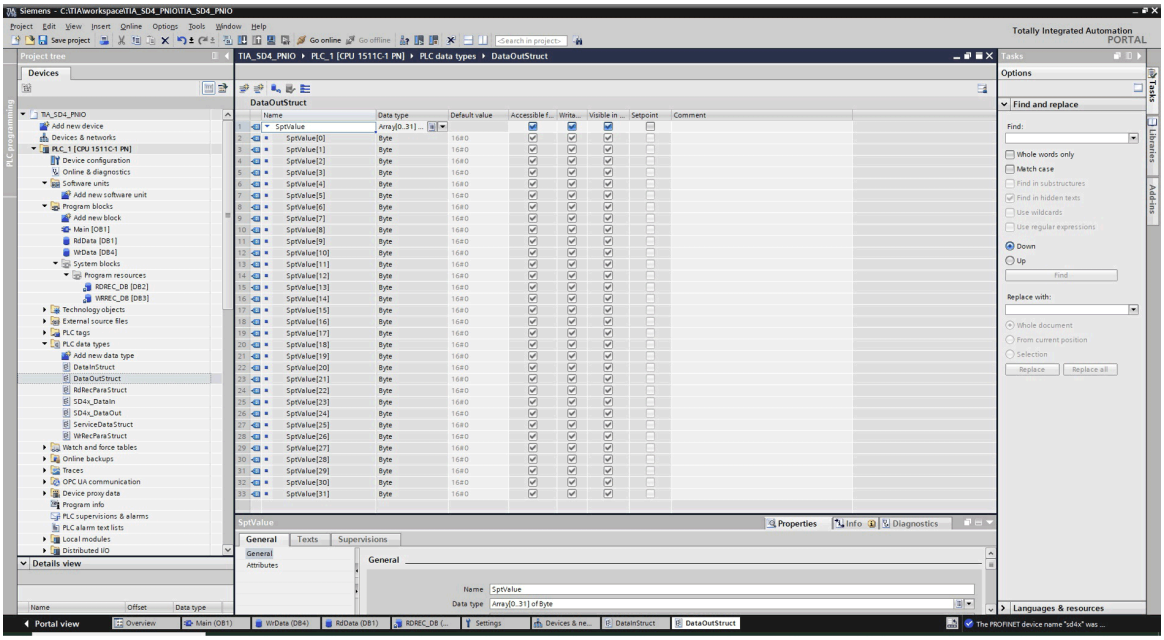


Fig. 12: PLC data types for the output data

12. Add new PLC variables in the standard variable table and link these to the input and output data structures.

    Example:

InData_0	DataInStruct	%I124.0
OutData_0	DataOutStruct	%Q126.0

13. Create your **Main** program now.

    How to assign the program variables to the process data of the drive is described in chapter 3.4 “Cyclic Process Data (PDO)”, page 15.

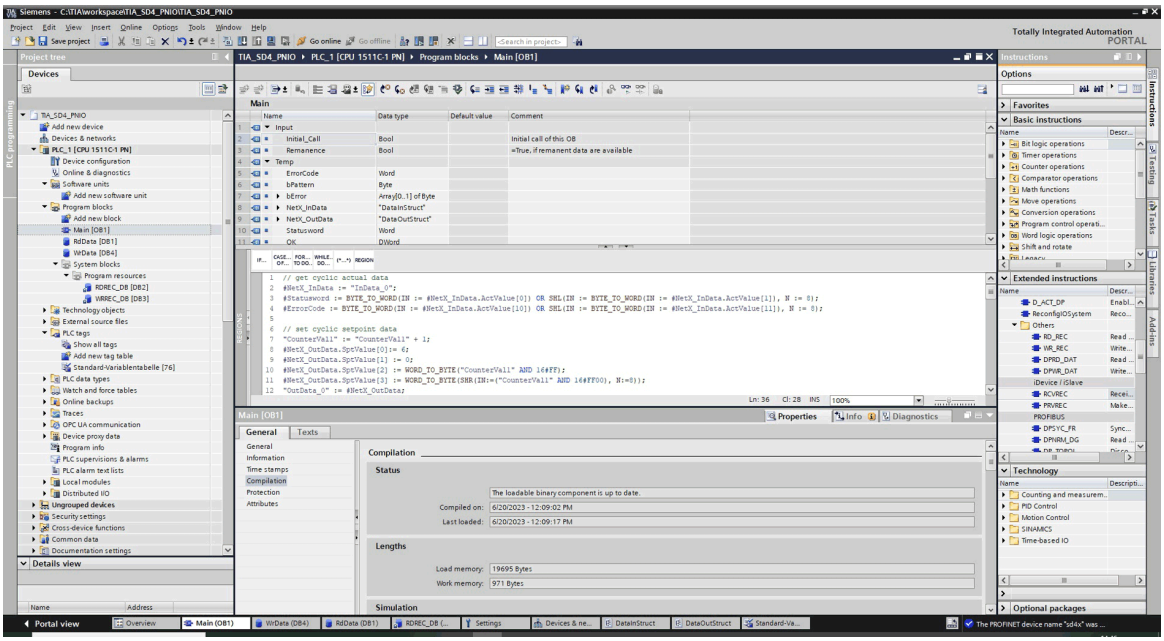


Fig. 13: Create Main program

14. Compile the hardware and the software.

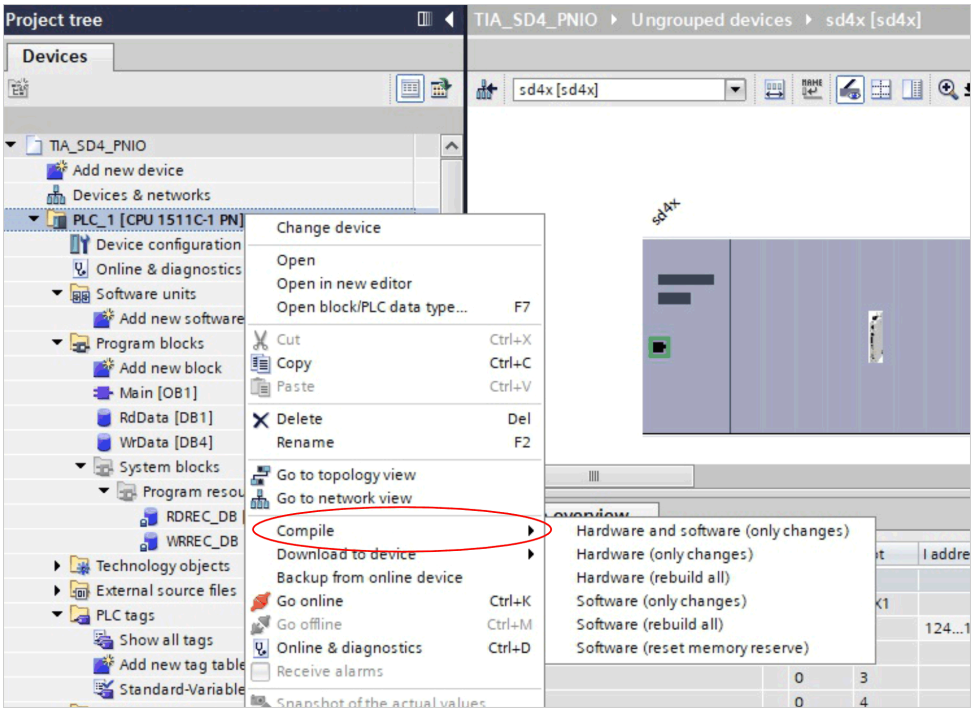


Fig. 14: Compile hardware and software

15. Download the hardware and the software to the PLC.

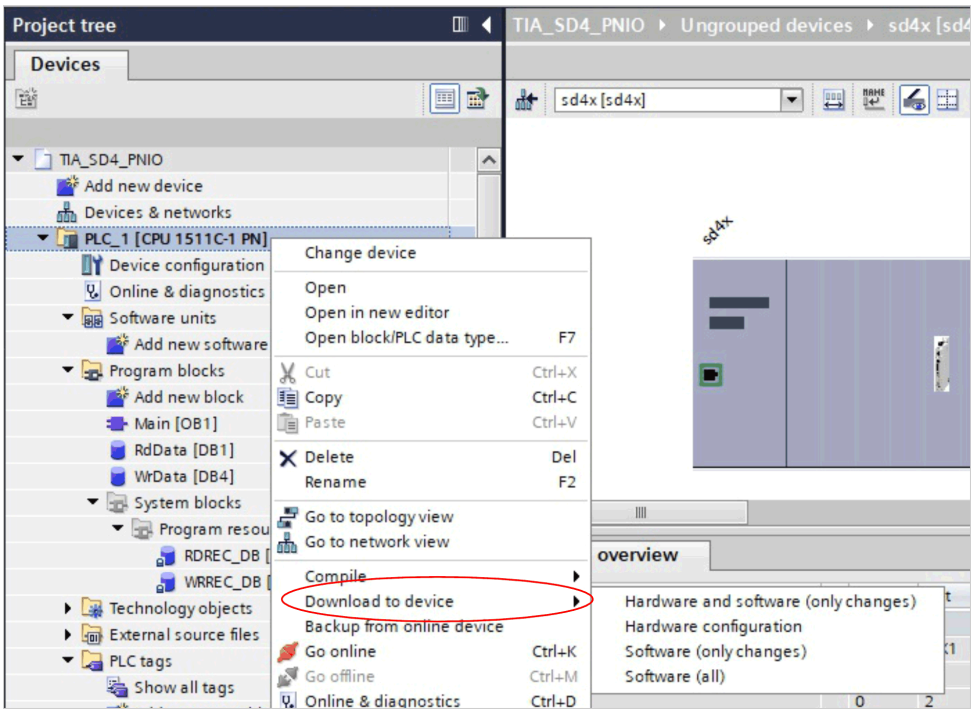


Fig. 15: Load hardware and software

16. Start your PLC in RUN mode.
17. Use the *drivemaster4* software to check that your data are correctly transmitted to the drive.
18. With the PLC in the online mode, check that the drive receives the data correctly.

3.3 PROFINET IO Network Management (NMT)

The drive communication is controlled by the PLC via the PROFINET IO network management. The PROFINET IO network management coordinates the controller and device applications during startup and operation.

The PROFINET IO network management can have the following states:

Status	Value	Description
UNKNOWN	0	Unknown status
OFFLINE	1	The drive configuration is not valid.
STOP	2	The drive cannot communicate with the controller. No connection is established. The bus status of the drive can be set to **on** or **off**.
IDLE	3	The communication is being established.
OPERATE	4	The I/O connection is established. The controller and the drive exchange valid I/O data.

3.4 Cyclic Process Data (PDO)

The process data channel enables cyclic data exchange between the PLC as *controller* and drive as *device*.

The meaning of the drive data is defined in the *drivemaster4* software via the parameter page “Fieldbus → PDO mapping”.

3.4.1 Setpoint Telegram

The setpoint telegram for controlling the drive consists of 32 byte user data. The setpoint of the PLC program must be mapped to these bytes.

Example of the setpoints in profile velocity mode:

Byte	Name	Bit	Description
0	Controlword (low byte) Bits coded acc. to DS402 standard (object 6040h)	0	Switch on
		1	Enable voltage
		2	Quick stop
		3	Enable operation
		4	Operating mode specific
		5	
		6	
		7	Fault reset
1	Controlword (high byte) Bits coded acc. to DS402 standard (object 6040h)	8	Hold
		9	Operating mode specific
		10	Reserved
		11	Manufacturer specific
		12	
		13	
		14	
		15	
2 3 4 5	Target velocity (object 60FFh)		Target velocity byte 0 Target velocity byte 1 Target velocity byte 2 Target velocity byte 3
6 7	Max torque (object 6072h)		Max torque byte 0 Max torque byte 1
8 … 31	Not used		= 0

If a 2-byte object is to be sent, this object must be split into 2 bytes of the setpoint telegram.

Example: DS402 max torque (object 6072h)

```
MaxTorqueByte0 = WORD_TO_BYTE (MaxTorque AND 16#00FF)
MaxTorqueByte1 = WORD_TO_BYTE (SHR (IN := (MaxTorque AND 16#FF00), N := 8))
```

If a 4-byte object is to be sent, this object must be split into 4 bytes of the setpoint telegram.

Example: DS402 target velocity (object 60FFh)

```
TargetVelocityByte0 = DWORD_TO_BYTE (TargetVelocity AND 16#000000FF);
TargetVelocityByte1 = DWORD_TO_BYTE (SHR (IN := (TargetVelocity AND 16#0000FF00), N := 8));
TargetVelocityByte2 = DWORD_TO_BYTE (SHR (IN := (TargetVelocity AND 16#00FF0000), N := 16));
TargetVelocityByte3 = DWORD_TO_BYTE (SHR (IN := (TargetVelocity AND 16#FF000000), N := 24));
```

3.4.2 Actual Value Telegram

The actual value telegram for controlling the drive consists of 32 byte user data. The actual values of the PLC program must be mapped to these bytes.

Example of the actual values in profile velocity mode:

Byte	Name	Bit	Description
0	Statusword (low byte) Bits coded acc. to DS402 standard (object 6041h)	0	Ready to switch on
		1	Switched on
		2	Operation enabled
		3	Fault
		4	Voltage enabled
		5	Quick stop
		6	Switch on disabled
		7	Warning
1	Statusword (high byte) Bits coded acc. to DS402 standard (object 6041h)	8	Manufacturer specific: setup mode active
		9	Remote operation
		10	Target reached
		11	Internal limit reached
		12	Operating mode specific: reserved
		13	
		14	Manufacturer specific: status Safe Torque Off (STO)
		15	Manufacturer specific: initialization finished
2 3 4 5	Velocity actual value (object 606Ch)		Velocity actual value byte 0 Velocity actual value byte 1 Velocity actual value byte 2 Velocity actual value byte 3
6 7 8 9	Position actual value (object 6064h)		Position actual value byte 0 Position actual value byte 1 Position actual value byte 2 Position actual value byte 3
10 11	Error code (object 603Fh)		Error code byte 0 Error code byte 1

If a 2-byte object is to be read, this object must be split into 2 bytes of the actual value telegram.

Example: DS402 error code (object 603Fh)

```
ErrorCode = BYTE_TO_DWORD (ErrorCodeByte0) OR
SHL (IN := BYTE_TO_DWORD (ErrorCodeByte1), N := 8);
```

If a 4-byte object is to be read, this object must be split into 4 bytes of the actual value telegram.

Example: DS402 position actual value (object 6064h)

```
PositionActualValue = BYTE_TO_DWORD (PositionActualValueByte0) OR
SHL (IN := BYTE_TO_DWORD (PositionActualValueByte1), N := 8) OR
SHL (IN := BYTE_TO_DWORD (PositionActualValueByte2), N := 16) OR
SHL (IN := BYTE_TO_DWORD (PositionActualValueByte3), N := 24);
```

3.5 Acyclic Service Data (SDO)

The service data channel enables acyclic data exchange between the PLC as *controller* and drive as *device*. Via this channel the PLC can acyclically read individual objects from the drive or write them to the drive.

The transmission is executed via the communication function blocks RDREC and WREC of the PLC. For information on the functionality, refer to the document “PROFIBUS and PROFINET Guideline, Communication Function Blocks on PROFIBUS DP and PROFINET IO”.

Complying with the standard, the index is limited to 16 bit. If the PLC manufacturer uses a 32-bit index value, only the lower 16 bit can be used.

A maximum length of 1024 byte can be read or written with one call.

Using the 16-bit index, the PLC can address each object in the object dictionary of the drive. Simple objects with a maximum length of 4 byte are read or written directly. If the PLC must address an object with subindex or an array object, the subindex or array offset must be transmitted in advance. For this purpose, the object 158 (0x009E) “PROFINET IO – Subindex and offset” is available. The lowest byte of this object is used for the subindex and the upper 3 bytes are used for the offset:

Byte 3	Byte 2	Byte 1	Byte 0
Offset			Subindex

The following data apply to the offset and subindex of object 158 “PROFINET IO – Subindex and offset”:

Object 158	Offset	Subindex
Value range	0 to 16777215 (0x00 to 0xFFFFFF)	0 to 255 (0x00 to 0xFF)
Default value	0	0

After each write or read operation, except for object 158, the subindex and the offset are internally reset to zero.

3.5.1 Write and Read Operations

The following figure show the principle operations for writing and reading acyclic service data.

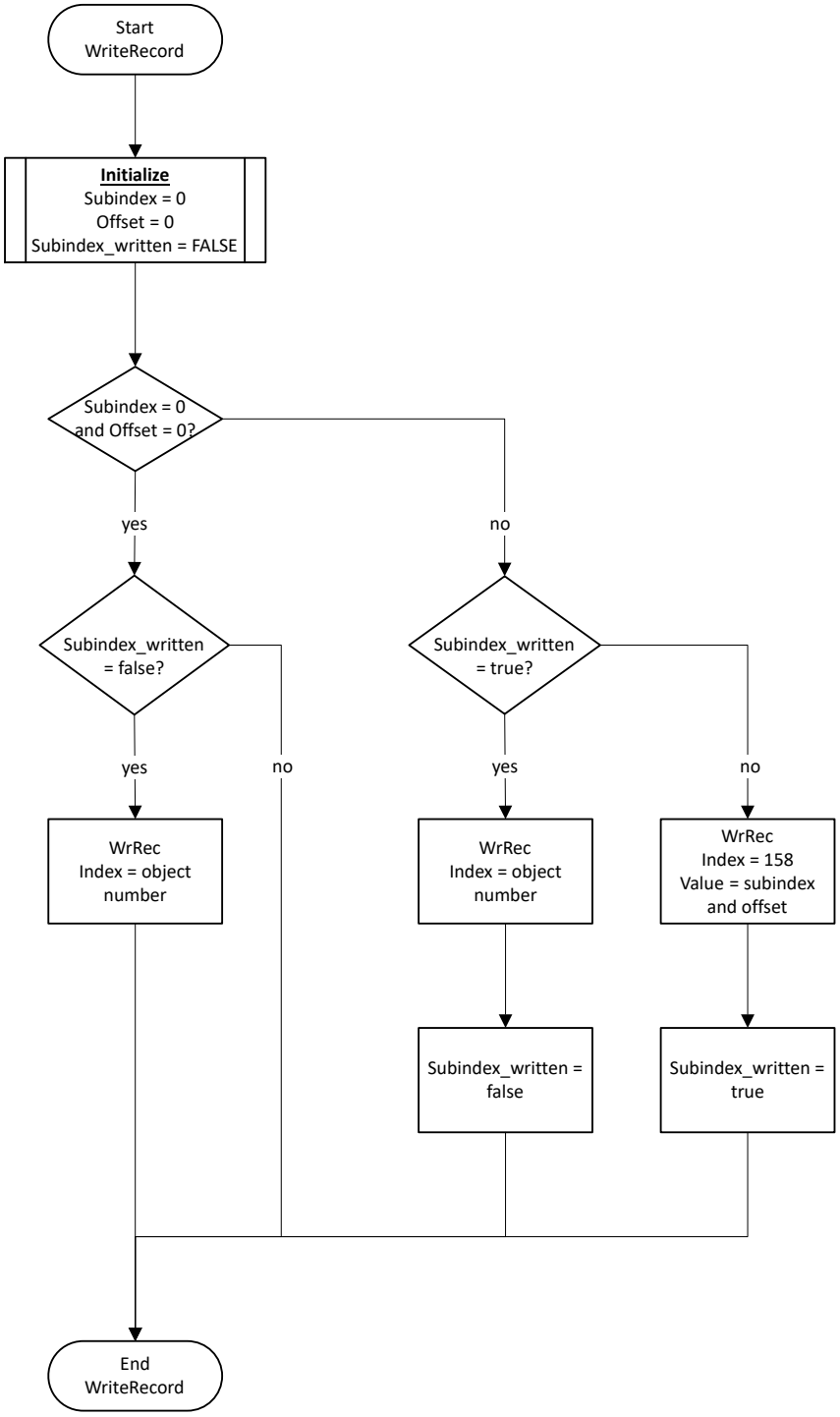


Fig. 16: Writing operation in principle

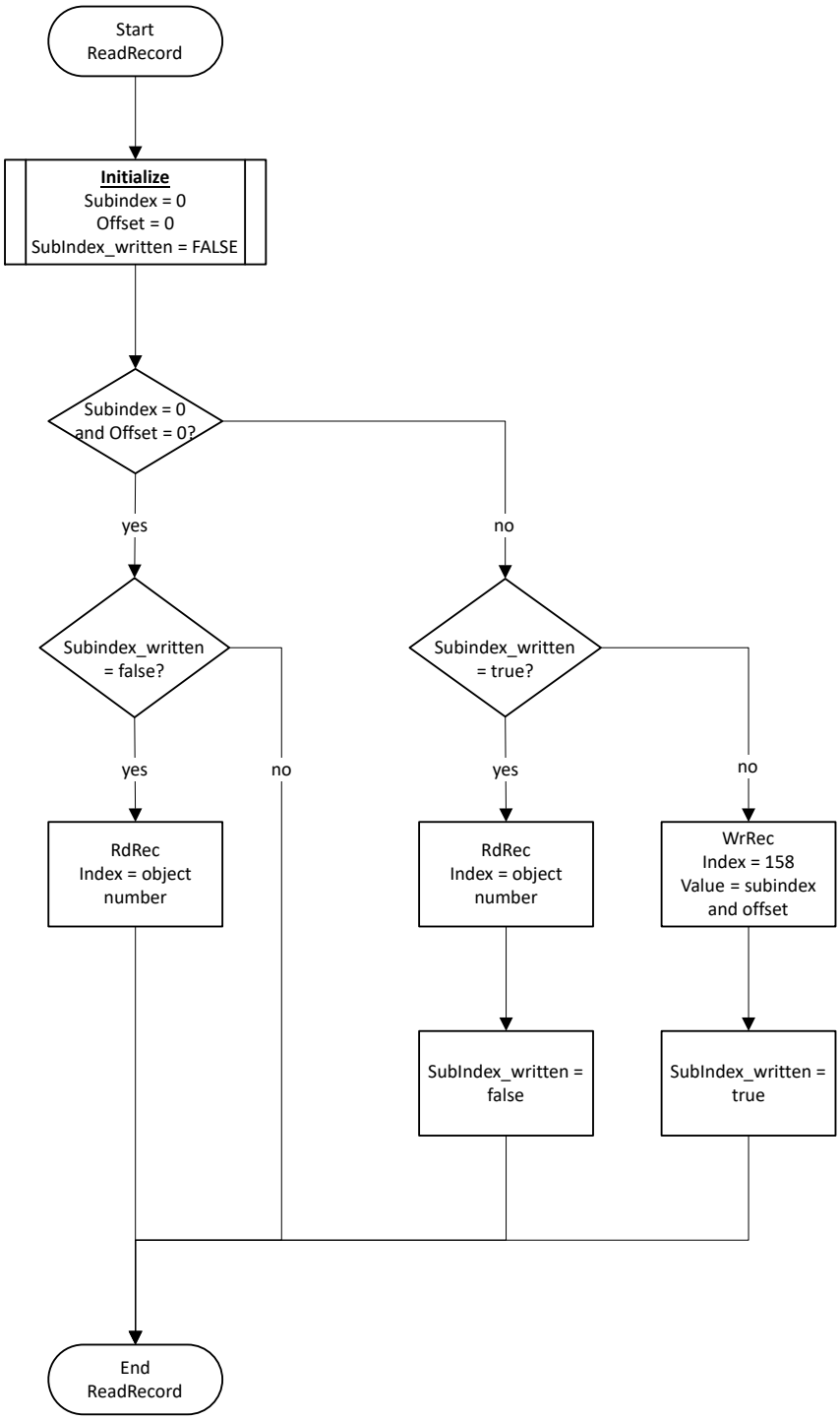


Fig. 17: Reading operation in principle

3.5.2 Program Example for Siemens PLC

The following program example was created with TIA PORTAL for a Siemens PLC S7-1500.

1. Add another block in “Program blocks”.
   a. Select the type “Global-DB” for the new block.
   b. Add a title and comment as needed.

c. Click “OK” to confirm your input.

→ The new block is created below “Program blocks”.

2. Select the block and add the following variables:

Name	Type
ReadRequest	BOOL
WriteRequest	BOOL
ReadRecord	ARRAY[0..1200] OF BYTE
WriteRecord	ARRAY[0..1200] OF BYTE

Now, you must add the write and read operations to the program example “Main (OB1)” by means of the instances of the functions for WRREC und RDREC.

3. Add **WriteRecord** in the program example “Main (OB1)” as follows:

a. Enter “WRR” in Main.

→ A selection list appears.

b. Select the entry “WRREC” in the selection list.

→ The dialog window “Call options” appears.

c. In the window “Call options”, enter the desired name for the data block, e.g. . “WRREC_DB_9”. Click “OK” to confirm the input.

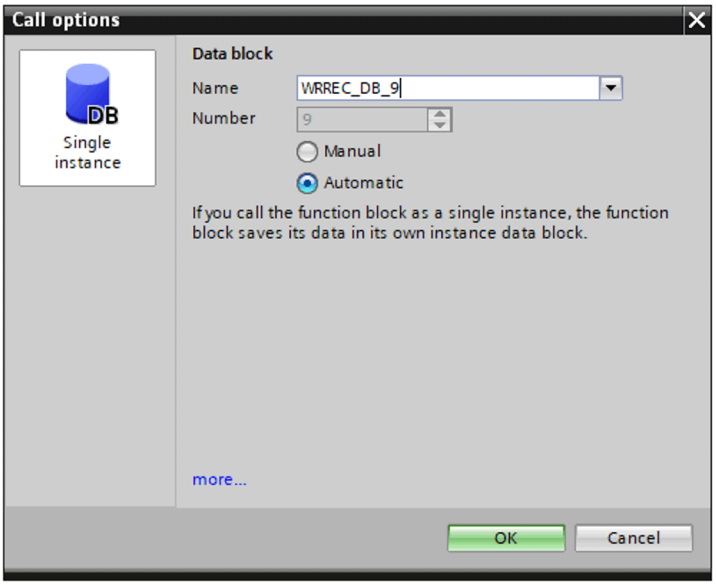


Fig. 18: Call options – WRREC

→ The block is created below “Program blocks → System blocks → Program resources”.

In addition, the call is added in the Main program:

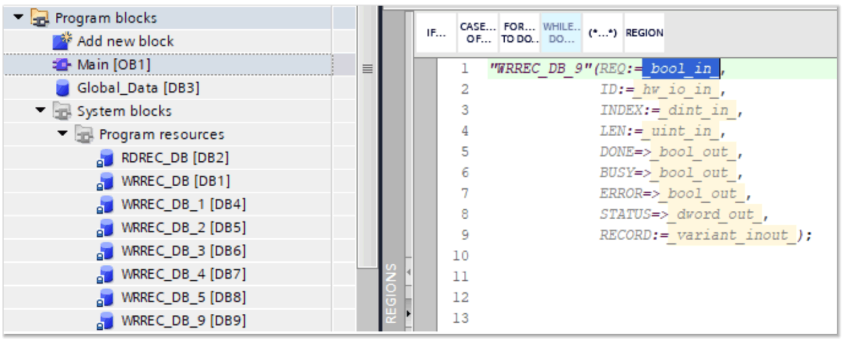


Fig. 19: Main – WRREC

d. You can adapt the call parameters for your application, now.

Example:

```
"WRREC_DB_9"(REQ := "Global_Data".WriteRequest,
ID := "sd4x~Head",
INDEX := 16#60C5,
LEN:=4,
RECORD:="Global_Data".WriteRecord[0]);
```

Program code in principle:

```
// set acyclic data
"Global_Data".WriteSubIndexOffset := 16#02000000;

IF "Global_Data".WriteSubIndexOffset = 0 THEN
IF NOT "Global_Data".ReadSIOwritten THEN
"WRREC_DB_1"(REQ := "Global_Data".WriteRequest,
ID := "sd4x~Head",
INDEX := 16#6073,
LEN := 2,
RECORD := "Global_Data".WriteCurrent);
END_IF;
ELSE
IF "Global_Data".WriteSIOwritten THEN
"WRREC_DB_2"(REQ := "Global_Data".WriteRequest,
ID := "sd4x~Head",
INDEX := 16#604A,
LEN := 2,
RECORD := "Global_Data".WriteRecord);
IF "WRREC_DB_2".DONE THEN
"Global_Data".WriteSIOwritten := FALSE;
END_IF;
ELSE
"WRREC_DB_3"(REQ := "Global_Data".WriteRequest,
ID := "sd4x~Head",
INDEX := 158,
LEN := 4,
RECORD := "Global_Data".WriteSubIndexOffset);
IF "WRREC_DB_3".DONE THEN
"Global_Data".WriteSIOwritten := TRUE;
END_IF;
END_IF;
END_IF;
```

4. Add **ReadRecord** in the program example “Main (OB1)” as follows:

   a. Enter “RDR” in Main.

      → A selection list appears.

   b. Select the entry “RDREC” in the selection list.

      → The dialog window “Call options” appears.

c. In the window “Call options”, enter the desired name for the data block, e.g. . “RDREC_DB_10”. Click “OK” to confirm the input.

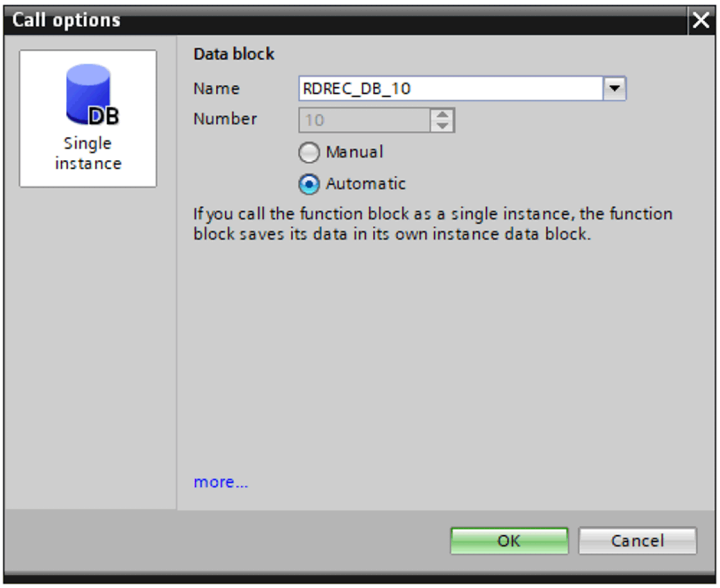


Fig. 20: Call options – RDREC

→ The block is created below “Program blocks → System blocks → Program resources”.

In addition, the call is added in the Main program:

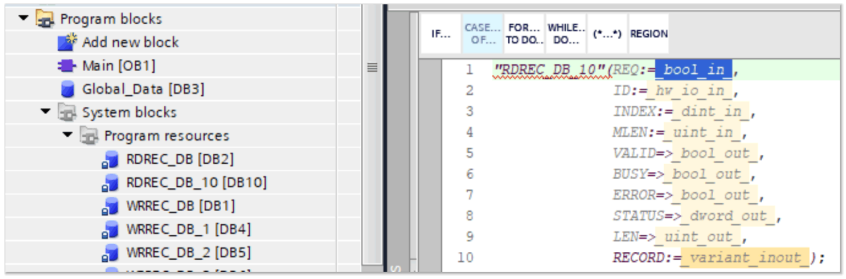


Fig. 21: Main – RDREC

d. You can adapt the call parameters for your application, now.

Example:

```
"RDREC_DB_10"(REQ:="Global_Data".ReadRequest,
ID:="sd4x~Head",
INDEX:=16#604A,
MLEN:=4,
RECORD:="Global_Data".ReadRecord[0]);
```

Program code in principle:

```
// get acyclic data
"Global_Data".ReadSubIndexOffset := 16#01000000;

IF "Global_Data".ReadSubIndexOffset = 0 THEN
IF NOT "Global_Data".ReadSIOwritten THEN
"RDREC_DB"(REQ := "Global_Data".ReadRequest,
ID := "sd4x~Head",
INDEX := 16#6073,
MLEN := 2,
RECORD := "Global_Data".ReadRecord);
END_IF;
ELSE
IF "Global_Data".ReadSIOwritten THEN
"RDREC_DB"(REQ := "Global_Data".ReadRequest,
ID := "sd4x~Head",
INDEX := 16#604A,
```

```
MLEN := 4,
RECORD := "Global_Data".ReadRecord);
IF "RDREC_DB".VALID THEN
"Global_Data".ReadSIOwritten := FALSE;
END_IF;
ELSE
"WRREC_DB"(REQ := "Global_Data".ReadRequest,
ID := "sd4x~Head",
INDEX := 158,
LEN := 4,
RECORD := "Global_Data".ReadSubIndexOffset);
IF "WRREC_DB".DONE THEN
"Global_Data".ReadSIOwritten := TRUE;
END_IF;
END_IF;
END_IF;
```

5. Add the processes for the write and read operation described above in your program. For this purpose, put the data to be written into Global_Data.WriteRecord. Control the reading and writing of the objects via Global_Data.WrRequest and Global_Data.RdReqest.

4 Drive Control

The drive control in devices of the product family SD4x is implemented according to the DS402 standard.

This standard defines objects that are divided into process data objects (PDO) and service data objects (SDO). PDOs are exchanged cyclically (real-time capable) and SDOs are exchanged acyclically (upon request, not real-time capable) via fieldbus.

The drive follows a predefined state machine, which is controlled by the commands of the controlword and returns the current status via the statusword. By means of the option codes you can change the shutdown behavior of the drive for the corresponding shutdown commands. The used target and actual values depend on the selected operating mode (profile velocity mode, velocity mode or torque mode). How to switch the operating mode is described under change operating mode.

The error codes are hardware-dependent. For a detailed description of the error codes refer to the hardware documentation of your device.

Object descriptions

All relevant objects are described in table form with the following information:

Short name of the object		Short description of the object	
Explicit name of the object			Object index
Access[1]	Data memory[2]	Unit	Data type

[1] Possible values:
RW = read/write
RO = read only

[2] Possible Values:
Volatile = data is lost after power cut-off
Non-volatile = data is retained after power cut-off

4.1 State Machine

The state machine (in the DS402 standard: finite state automaton, FSA) controls the power electronics of the drive. The following diagram shows the states with their meaning and the sequence of the state machine.

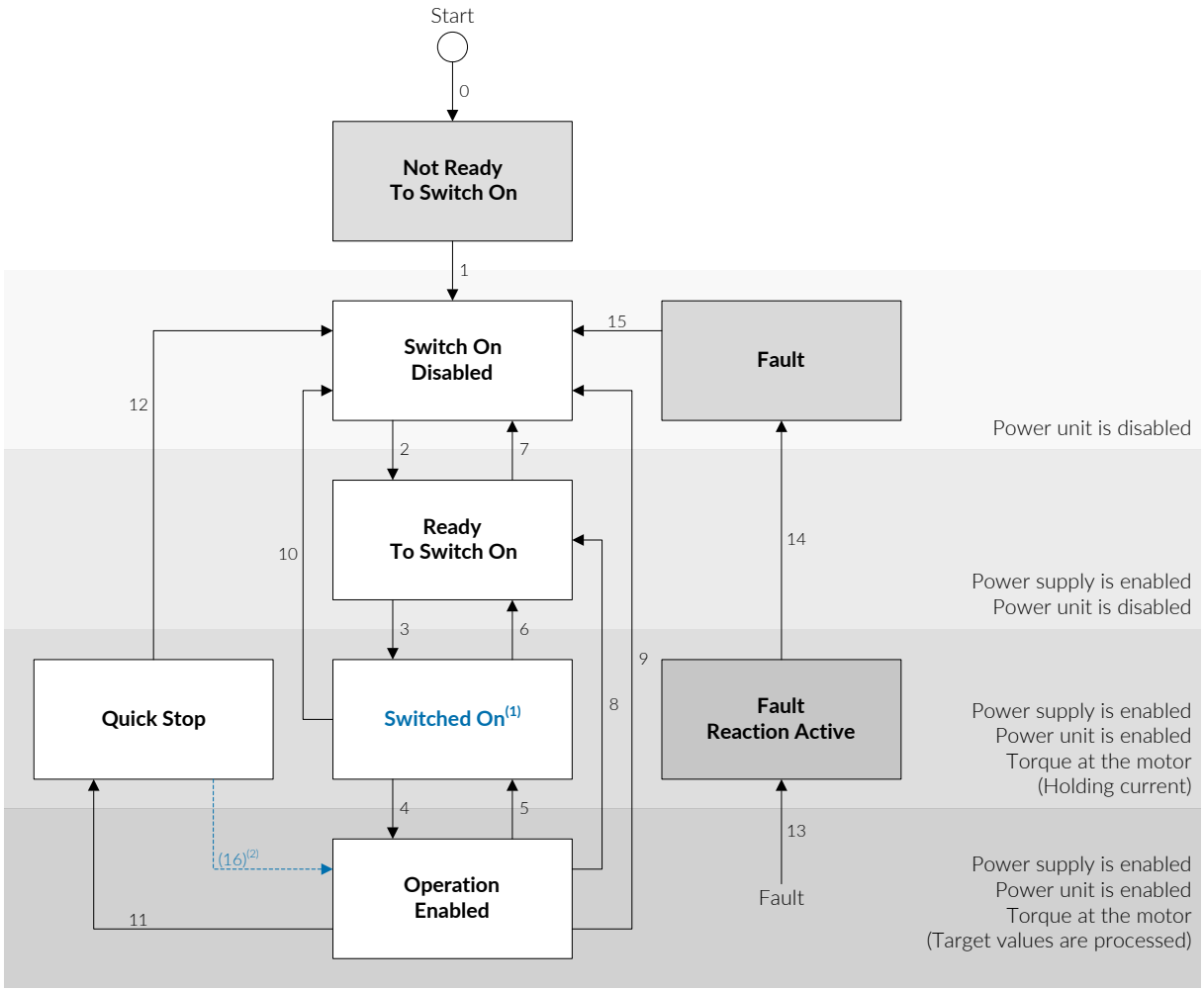


Fig. 22: Drive states and state transitions

(1) Unlike in the DS402 standard, the output stage of SD4x drives is already enabled in the status “Switched On” so that torque is applied to the motor. In this status the target values are not yet enabled and the motor is held.

(2) The transition 16 is not recommended by the DS402 standard anymore. Therefore, SD4x drives do not support transition 16.

The number at the arrows are transitions triggered by commands. Which command triggers which transition is described in chapter 4.1.1 “Commands”, page 26. The boxes represent the states of the state machine. On the right side of the figure, you find the corresponding states of the power electronics. The gray-colored boxes contain states that the state machine reaches automatically and not via commands.

4.1.1 Commands

The control commands depend on the currently active status. They are triggered via setting the following bit patterns in the controlword:

Command	Bits of controlword[1]					Transition
	Bit 7 Fault reset	Bit 3 Enable operation	Bit 2 Quick stop	Bit 1 Enable voltage	Bit 0 Switch on	
Shutdown	0	X	1	1	0	2, 6, 8
Switch on	0	X[2]	1	1	1	3
Disable voltage	0	X	X	0	X	7, 9, 10, 12
Quick stop	0	X	0	1	X	7, 10, 11
Disable operation	0	0	1	1	1	5
Enable operation	0	1	1	1	1	4, (16)[3]
Fault reset	↑[4]	X	X	X	X	15

[1] The bits labeled with “X” have no relevance at the corresponding position in the table.

[2] If the bit is 1, the state machine automatically switches after execution in “Switched on” to the next status “Operation enabled”.

[3] The transition 16 is not recommended by the DS402 standard anymore. Therefore, SD4x drives do not support transition 16.

[4] Positive edge (change from 0 to 1, pulse)

The bits not mentioned in the controlword are irrelevant for the commands. If a command is unknown for a status , it is ignored. The transitions not mentioned happen automatically in the state machine and cannot be triggered via the commands.

Controlword (0x6040)

The controlword is a bit-coded WORD and transmits control demands to the drive.

Controlword	Control commands: Bit 0: Switch on Bit 1: Enable voltage Bit 2: Quick stop Bit 3: Enable operation Bit 4: Mode-specific 1 Bit 5: Mode-specific 2 Bit 6: Mode-specific 3 Bit 7: Fault reset Bit 8: Halt Bits 9 to 15: Reserved		
DS402_0x6040_CONTROLWORD			ID: 0x6040
RW	Volatile	Unit: –	U16

The commands for controlling the state machine are made up of the bits 0 (switch on), 1 (v), 2 (quick stop), 3 (enable operation) and 7 (fault reset).

The function “Halt” (bit 8) is only active in the status “Operation enabled”. This function is used to stop the motor with the halt option code. Then, the motor is held. The status “Operation enabled” is maintained the entire time.

Information on the mode-specific bits 4 to 7 can be found under the respective operating mode.

4.1.2 States

The statusword returns the states of the state machine according to the following bit patterns. For the different states, the following bits are set:

Status	Bits of statusword[1]					
	Bit 6 Switch on disabled	Bit 5 Quick stop	Bit 3 Fault	Bit 2 Operation enabled	Bit 1 Switched on	Bit 0 Ready to switch on
Not ready to switch on (firmware not ready)	0	X	0	0	0	0
Switch on disabled	1	X	0	0	0	0
Ready to switch on	0	1	0	0	0	1
Switched on	0	1	0	0	1	1
Operation enabled	0	1	0	1	1	1
Quick stop active	0	0	0	1	1	1
Fault reaction active	0	X	1	1	1	1
Fault	0	X	1	X	0	X

[1] The bits labeled with “X” have no relevance at the corresponding position in the table.

The bits not mentioned in the statusword are irrelevant for the states.

Statusword (0x6041)

The statusword is a bit-coded WORD and contains status information of the drive.

Statusword	Device states: Bit 0: Ready to switch on Bit 1: Switched on Bit 2: Operation enabled Bit 3: Fault Bit 4: Voltage enabled Bit 5: Quick stop Bit 6: Switch on disabled Bit 7: Warning Bit 8: *Drive setup tool (DRS) active** Bit 9: Remote Bit 10: Target reached Bit 11: *Current limit reached** Bit 12: Mode-specific 1 Bit 13: Mode-specific 2 Bit 14: *Safe Torque Off (STO)** Bit 15: *Initialization finished** **Bit assignments shown in italics are SIEB & MEYER specific and may differ with other drive manufacturers.*		
DS402_0x6041_STATUSWORD			ID: 0x6041
RO	Volatile	Unit: –	U16

The bits 0 (ready to switch on), 1 (switched on), 2 (operation enabled), 3 (fault), 5 (quick stop) und 6 (switch on disabled) return the current status of the state machine via bit patterns. In the event of an error, the object ***error code*** (ID 0x603F) returns the error information.

Bit 4 (enable voltage) is active when the voltage for supplying the DC link is applied.

Bit 7 (warning) is active in the event of a warning message. The object ***error code*** (ID 0x603F) returns information on the current warning message.

Bit 8 (Drive Setup Tool (DRS) active) is SIEB & MEYER-specific and active when the *drivemaster4* software or another setup tool controls the drive.

Bit 9 (remote) is active when the drive is controlled by the control commands of the controlword.

Bit 10 (target reached) is active when the drive has reached the target value. This bit depends on the operating mode and the current status:

- Velocity modes VL/PV: bit 10 is set when the set target speed is reached.
- Status “Quick stop active”: if the drive does not automatically exit the status “Quick stop active”, bit 10 indicates that the quick stop function has been completed and the motor is being held.
- Changing the operating mode: the bit is set after the operating mode has been successfully changed.

Bit 11 (current limit reached) is SIEB & MEYER-specific and active when the drive is operated at the set current limit.

The bits 12 and 13 are mode-specific. For more information refer to the chapters on the operating modes.

Bit 14 (Safe Torque Off (STO)) is SIEB & MEYER-specific and active when the safety function STO keeps the drive switched off. An example would be an emergency stop that is connected with the SAFE inputs of the drive.

Bit 15 (initialization finished) is SIEB & MEYER-specific und becomes active as soon as the drive has been successfully initialized.

4.1.3 Option Codes

The option codes define the behavior of the drive on receipt of a stop command (e.g. disable operation) or in case of a shutdown event (e.g a fault or loss of connection to the control unit).

Abort Connection Option Code (0x6007)

This object defines the reaction of the drive to a loss of connection due to one of the following events: bus off status, heartbeat/node guarding error, NMT stopped status, reset of application or configuration.

Abort connection option code		Selection of the reaction to a fieldbus communication error: 00: No action 01: Fault signal 02: Disable voltage command 03: Quick stop command	
Object: DS402_0X6007_ABORT_CONNECTION_OPTION_CODE			ID: 0x6007
RW	Volatile	Unit: –	I16

Quick Stop Option Code (0x605A)

This object defines the reaction of the drive to a quick stop command.

Quick stop option code		Selection of the reaction to a quick stop command: 00: Disable drive function 01: Slow down on slow down ramp and transit into 'switch on disabled' 02: Slow down on quick stop ramp and transit into 'switch on disabled' 03: Slow down on current limit and transit into 'switch on disabled'	
DS402_0X605A_QUICK_STOP_OPTION_CODE			ID: 0x605A
RW	Volatile	Unit: –	I16

The ramps are defined in the operating mode.

Shutdown Option Code (0x605B)

This object defines the reaction of the drive to a shutdown command, i.e. the drive switches from the status “operation enabled” to the status “ready to switch on”.

Shutdown option code		Selection of the reaction to the command “shutdown”: −1: Slow down via short-circuiting the motor phases 00: Disable drive function (switch off the drive power output stage) 01: Slow down on slow down ramp, disable drive function	
DS402_0X605B_SHUTDOWN_OPTION_CODE			ID: 0x605B
RW	Volatile	Unit: –	I16

The slow down ramp is defined in the operating mode.

Disable Operation Option Code (0x605C)

This object defines the reaction of the drive to a disable operation command, i.e. the drive switches from the status “operation enabled” to the status “switched on”.

Disable operation option code		Selection of the reaction to the command “disable operation”: 01: Slow down on slow down ramp, disable drive function	
DS402_0X605C_DISABLE_OPERATION_OPTION_CODE			ID: 0x605C
RW	Volatile	Unit: –	I16

The slow down ramp is defined in the operating mode.

Note

The value is not intended to be changed.

Halt Option Code (0x605D)

This object defines the reaction of the drive to a halt command.

Halt option code		Selection of the reaction to the command “halt”: 01: Slow down on slow down ramp and stay in 'operation enabled' 02: Slow down on quick stop ramp and stay in 'operation enabled' 03: Slow down on current limit and stay in 'operation enabled'	
DS402_0X605D_HALT_OPTION_CODE			ID: 0x605D
RW	Volatile	Unit: –	I16

The ramps are defined in the operating mode.

Fault Reaction Option Code (0x605E)

This object defines the reaction of the drive to a fault.

Fault reaction option code		Selection of the reaction to a fault: 00: Disable drive function, motor is free to rotate 01: Slow down on slow down ramp 02: Slow down on quick stop ramp 03: Slow down on current limit 04: Slow down on voltage limit	
DS402_0X605D_HALT_OPTION_CODE			ID: 0x605E
RW	Volatile	Unit: –	I16

The ramps are defined in the operating mode.

4.2 Operating Modes

The drive supports several operating modes, which depend on the device variant, the motor, the connected sensors and the parameterized drive function.

The following chapters describe the available operating modes and how to change the operating mode.

4.2.1 PV

In order to control the drive in the operating mode 'profile velocity mode', at least the following objects are required: ***controlword***, ***statusword*** and ***target velocity***. It is recommended to use the ***velocity actual value*** as return value. If the motor is not equipped with a speed sensor, the drive calculates the actual speed from the rotating field that is currently applied to the motor.

The following diagram shows the used objects and examples of their function:

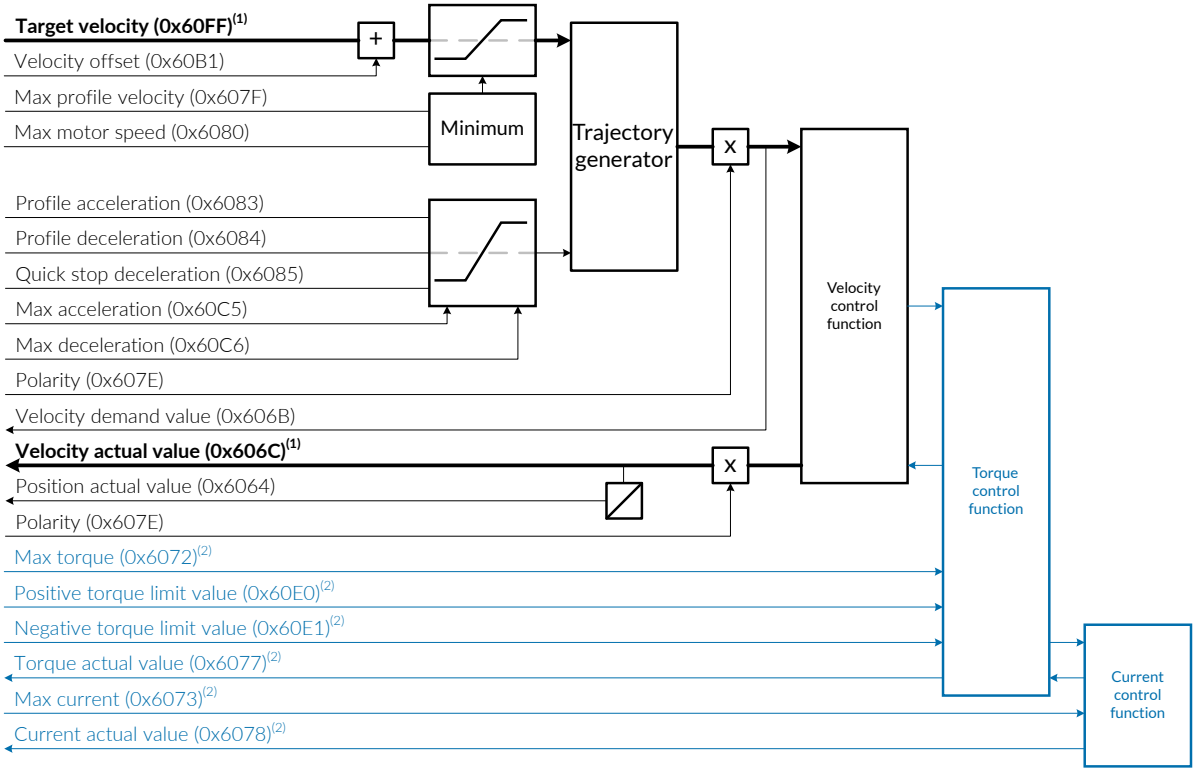


Fig. 23: Block diagram for profile velocity mode (PV)

(1) The objects marked in bold are required and recommended process data (PDO) for the profile velocity mode.

(2) SD4x drives support these objects (marked in blue) additionally to the standard objects in the DS402 drive profile.

PV-specific bits in the controlword and the statusword

The mode-specific bits 4, 5 and 6 in the controlword are optional bits and not supported in this operating mode.

The mode-specific bits in the statusword are assigned as follows:

- Bit 10 (target reached) is active when the difference between actual speed and target speed is longer than the speed window time within the speed window.
  You can set the speed window in the *drivemaster4* software under “Messages → Messages → Reference value reached [M51]”.
- Bit 12 (speed zero) is deactivated when the speed actual value exceeds the speed threshold longer than the speed threshold time. Below this threshold the bit is active and indicates that the axis is at standstill.
  You can set the speed window in the *drivemaster4* software under “Messages → Messages → Speed zero [M15]”.
- Bit 13 (maximum slip) is active when the drive has reached the parameterized slip.

4.2.2 Velocity Mode (VL)

The velocity mode uses its own objects for scaling the target and actual values. By default, the speeds are specified in *revolutions per minute* [rpm]. You can change the unit via the scaling object ***vl dimension factor***. In addition, the object ***vl Set-point Factor*** is used to scale the target and actual speed. Due to the object size, the resolution of the target and actual speed is limited to 16 bits in this operating mode. For applications that require a higher resolution, you can use the operating mode PV, page 31, which offers a 32 bit resolution for target and actual speeds.

In order to control the drive in the operating mode 'velocity mode', at least the following objects are required: ***controlword***, ***statusword*** and ***vl target velocity***. It is recommended to use the ***vl velocity actual value*** as return value.

The following diagram shows the used objects and examples of their function:

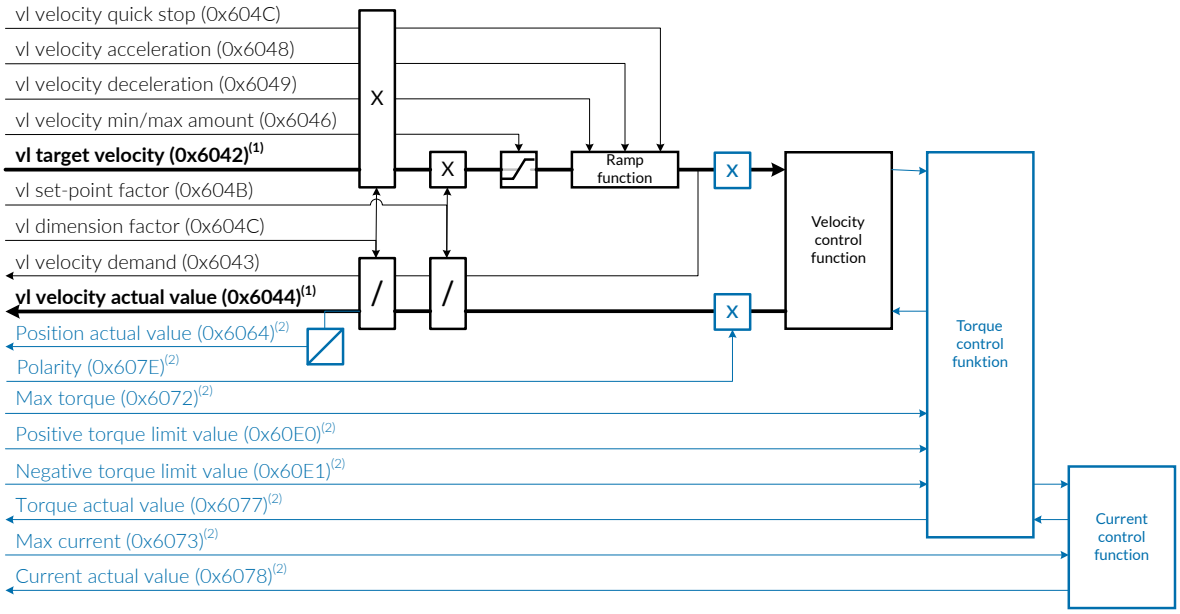


Fig. 24: Block diagram for velocity mode VL

(1) The objects marked in bold are required and recommended process data (PDO) for the velocity mode.

(2) SD4x drives support these objects (marked in blue) additionally to the standard objects in the DS402 drive profile.

VL-specific bits in the controlword and the statusword

The mode-specific bits 4, 5 and 6 in the controlword are optional bits and not supported by SD4x drives.

The mode.specific bits in the statusword are reserved in velocity mode according to the DS402 drive profile. However, SD4x drives use the bits 10 and 12 as follows:

- Bit 10 (target reached) is active when the difference between actual speed and target speed is longer than the speed window time within the speed window.

  You can set the speed window in the *drivemaster4* software under “Messages → Messages → Reference value reached [M51]”.
- Bit 12 (speed zero) is deactivated when the speed actual value exceeds the speed threshold longer than the speed threshold time. Below this threshold the bit is active and indicates that the axis is at standstill.

  You can set the speed window in the *drivemaster4* software under “Messages → Messages → Speed zero [M15]”.

4.2.2.1 Target and Actual Values in Velocity Mode

vl Target Velocity (0x6042)

This object contains the target speed in velocity mode.

vl target velocity		Target speed in velocity mode When the objects ***vl set-point factor*** (0x604B) and ***vl dimension factor*** (0x604C) are set to the value 1 (default), this object returns the speed value in rpm. Positive values cause forward motion, negative values cause backward motion.	
DS402_0x6042_VL_TARGET_VELOCITY			ID: 0x6042
RW	Volatile	Unit: rpm (default)	I16

vl Velocity Demand (0x6043)

This object returns the resulting target speed in velocity mode after ramp and limiting functions.

vl velocity demand		Speed setpoint in velocity mode The value indicates the speed setpoint after the ramp and limiting functions. The unit and direction correspond to object ***vl target velocity*** (0x6042).	
DS402_0x6043_VL_VELOCITY_DEMAND			ID: 0x6043
RO	Volatile	Unit: rpm (default)	I16

vl Velocity Actual Value (0x6044)

This object returns the speed actual value in velocity mode.

vl velocity actual value		Actual speed in velocity mode The unit and direction correspond to object ***vl target velocity*** (0x6042).	
DS402_0x6044_VL_VELOCITY_ACTUAL_VALUE			ID: 0x6044
RO	Volatile	Unit: rpm (default)	I16

Depending on the application (with or without a sensor) the speed actual value is either determined by a sensor or derived from the calculated rotating field.

vl Velocity Min Max Amount (0x6046)

Via this object you can limit the speed in velocity mode.

vl velocity min max amount		Speed limitation in velocity mode Sub-IDs: ▸ 0x6046:00 = Highest subindex supported (2) ▸ 0x6046:01 = vl velocity min amount ▸ 0x6046:02 = vl velocity max amount If the object ***vl dimension factor*** (0x604C) is set to the value 1 (default), this object indicates the speed limits in rpm.	
DS402_0X6046_VL_VELOCITY_MIN_MAX_AMOUNT			ID: 0x6046
RW	Volatile	Unit: rpm (default)	BYTEARRAY

In its subindex 0, this object contains the last subindex of the object (in this case 2).

The following figure shows the limitation of the minimum and maximum speed via the object ***vl velocity min max amount***. The dotted line shows the speed setpoint and the blue, continuous line shows the resulting target speed.

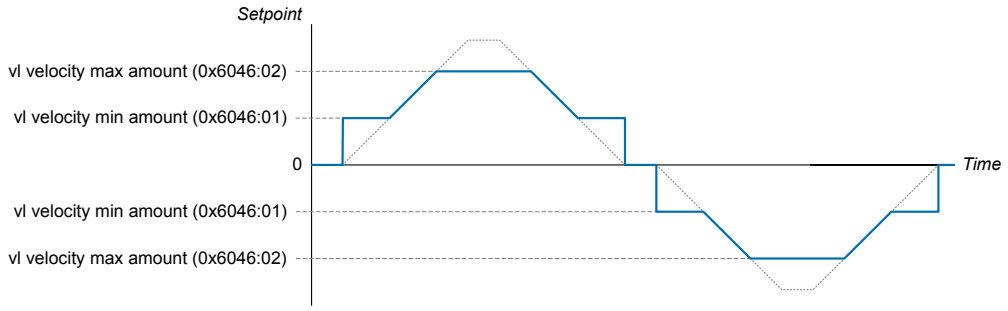


Fig. 25: Speed limitation in velocity mode

vl Velocity Acceleration (0x6048)

Via this object you can limit the acceleration in velocity mode.

vl velocity acceleration		Acceleration ramp in velocity mode Sub-IDs: ▸ 0x6048:00 = Highest subindex supported (2) ▸ 0x6048:01 = Delta speed If the object ***vl dimension factor*** (0x604C) is set to the value 1 (default), this object indicates the speed in rpm. ▸ 0x6048:02 = Delta time	
DS402_0x6048_VL_VELOCITY_ACCELERATION			ID: 0x6048
RW	Volatile	Unit: rpm/s (default)	DS402_ACCERLERATION_REC

In its subindex 0, this object contains the last subindex of the object (in this case 2).

The following figure shows the limitation of the acceleration via the object ***vl velocity acceleration***. The dotted line shows the speed setpoint and the blue, continuous line shows the resulting target speed.

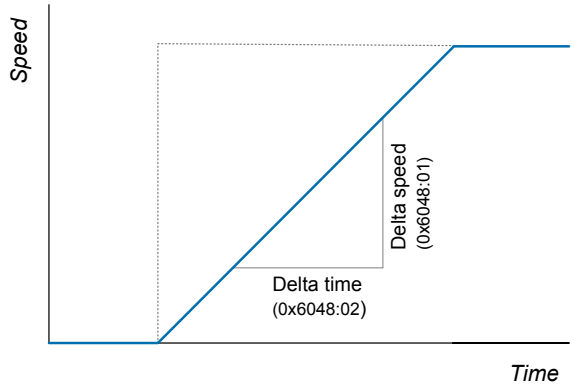


Fig. 26: Limitation of acceleration in velocity mode

Thus, the resulting acceleration ramp with default settings is specified in *revolutions per minute per second*.

vl Velocity Deceleration (0x6049)

Via this object you can limit the deceleration in velocity mode.

vl velocity deceleration	Deceleration ramp in velocity mode Sub-IDs: ▸ 0x6049:00 = Highest subindex supported (2) ▸ 0x6049:01 = Delta speed If the object ***vl dimension factor*** (0x604C) is set to the value 1 (default), this object indicates the speed in rpm. ▸ 0x6049:02 = Delta time The time is indicated in s. The value 0 is not permitted, in this case the minimum value (1 s) is used.		
DS402_0x6049_VL_VELOCITY_DECELERATION			ID: 0x6049
RW	Volatile	Unit: rpm/s (default)	DS402_ACCERLERATION_REC

In its subindex 0, this object contains the last subindex of the object (in this case 2).

The following figure shows the limitation of the deceleration via the object ***vl velocity deceleration***. The dotted line shows the speed setpoint and the blue, continuous line shows the resulting target speed.

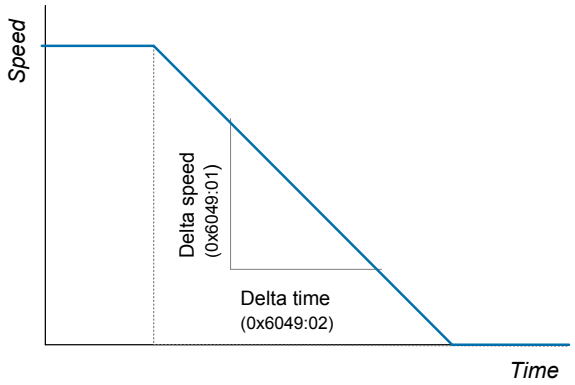


Fig. 27: Limitation of deceleration in velocity mode

Thus, the resulting deceleration ramp with default settings is specified in *revolutions per minute per second*.

vl Velocity Quick Stop (0x604A)

Via this object you can limit the deceleration in the event of a quick stop in velocity mode.

vl velocity quick stop	Quick stop ramp in velocity mode Sub-IDs: ▸ 0x604A:00 = Highest subindex supported (2) ▸ 0x604A:01 = Delta speed If the object ***vl dimension factor*** (0x604C) is set to the value 1 (default), this object indicates the speed in rpm. ▸ 0x604A:02 = Delta time The time is indicated in s. The value 0 is not permitted, in this case the minimum value (1 s) is used.		
DS402_0x604A_VL_VELOCITY_QUICK_STOP			ID: 0x604A
RW	Volatile	Unit: rpm/s (default)	DS402_ACCERLERATION_REC

In its subindex 0, this object contains the last subindex of the object (in this case 2).

The following figure shows the limitation of the deceleration in the event of a quick stop via the object ***vl velocity quick stop***. The dotted line shows the speed setpoint and the blue, continuous line shows the resulting target speed.

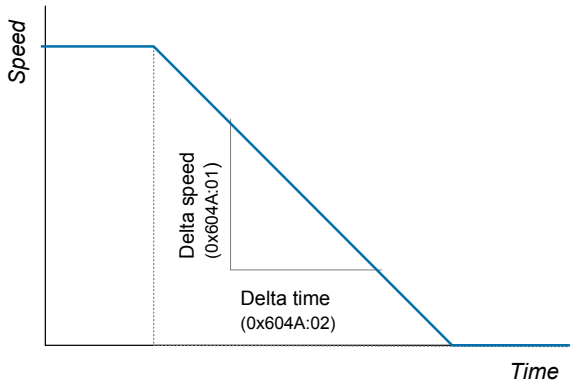


Fig. 28: Limitation of deceleration in the event of a quick stop in velocity mode

Thus, the resulting quick stop ramp with default settings is specified in *revolutions per minute per second*.

vl Set-point Factor (0x604B)

Via this object you can set an additional scaling factor for the speed target and actual values in velocity mode.

vl set-point factor		Scaling factor for the setpoint and the actual value in velocity mode Sub-IDs: ▸ 0x604B:00 = Highest subindex supported (2) ▸ 0x604B:01 = Numerator The numerator has no unit. The value 0 is not permitted, in this case the last valid value is retained. ▸ 0x604B:02 = Denominator The denominator has no unit. The value 0 is not permitted, in this case the last valid value is retained.	
DS402_0X604B_VL_SET_POINT_FACTOR			ID: 0x604B
RW	Volatile	Unit: –	DS402_ACCERLERATION_REC

In its subindex 0, this object contains the last subindex of the object (in this case 2).

This unit scaling only applies to the objects ***vl target velocity*** (0x6042), ***vl velocity demand*** (0x6043) and ***vl velocity actual value*** (0x6044), i.e. not the ramp and limiting functions.

$$Scaling\ factor\ for\ setpoints = \frac{Numerator\ (0x604B:01)}{Denominator\ (0x604B:02)}$$

Application examples

1. The speed is to be specified in *revolutions per minute* with one decimal position:
   – Numerator remains set to 1.
   – Denominator is set to 10.
   The values of the objects ***vl target velocity***, ***vl velocity demand*** and ***vl velocity actual value*** are immediately reinterpreted: the value 101 means 10.1 rpm.

   Note

   This setting only allows speed settings/displays between −3276.8 and 3276.7 rpm.

2. Speeds above 32767 rpm must be set:
   – Numerator is set to 10.

– Denominator remains set to 1.

The values of the objects ***vl target velocity***, ***vl velocity demand*** and ***vl velocity actual value*** are immediately reinterpreted: the value 101 means 1010 rpm.

Note

This setting allows speed settings/displays between −327680 and 327670 rpm. But the target and actual speeds can only be specified/displayed in rpm steps of 10.

vl Dimension Factor (0x604C)

Via this object all speed values in velocity mode are scaled. Beside the target and actual values this also applies to the ramp and limiting functions.

vl dimension factor		Scaling factor for the speed unit in velocity mode Sub-IDs: ▸ 0x604C:00 = Highest subindex supported (2) ▸ 0x604C:01 = Numerator The numerator has no unit. The value 0 is not permitted, in this case the last valid value is retained. ▸ 0x604C:02 = Denominator The denominator has no unit. The value 0 is not permitted, in this case the last valid value is retained.	
DS402_0X604C_VL_DIMENSION_FACTOR			ID: 0x604C
RW	Volatile	Unit: –	DS402_ACCERLERATION_REC

In its subindex 0, this object contains the last subindex of the object (in this case 2).

The purpose of this scaling: Each user-specific speed consists of specific units in relation to time units (e.g. 1/s, cams/min, m/s etc.). The dimension factor converts these speed units to a common basis (e.g. revolutions per minute).

$$Scaling\ factor\ to\ rpm = \frac{Numerator\ (0x604C:01)}{Denominator\ (0x604C:02)}$$

When the scaling factor is changed, the ramp and limiting functions are automatically scaled to the new unit. This applies to the following objects: vl velocity min amount (0x6046:01), vl velocity max amount (0x6046:02), vl velocity acceleration (0x6048:01), vl velocity deceleration (0x6049:01) and vl velocity quick stop (0x604A:1). If a higher-ranking control exchanges one of these objects as PDO, the control must perform the scaling of the corresponding object.

Application example

The speed is to be specified in *revolutions per second* with one decimal position:

- Numerator is set to 60.
- Denominator is set to 10.

The values of the objects ***vl target velocity***, ***vl velocity demand*** and ***vl velocity actual value*** are immediately reinterpreted: the value 32 means 3.2 1/s or 192 rpm.

Note

This setting allows speed settings/displays between −196608 and 196602 rpm. But the target and actual speeds can only be specified/displayed in rpm steps of 6.

4.2.3 Torque Mode

In order to control the drive in the operating mode 'torque mode', at least the following objects are required: ***controlword***, ***statusword*** and ***target torque***. It is recommended to use the ***torque actual value*** as return value.

The following diagram shows the used objects and examples of their function:

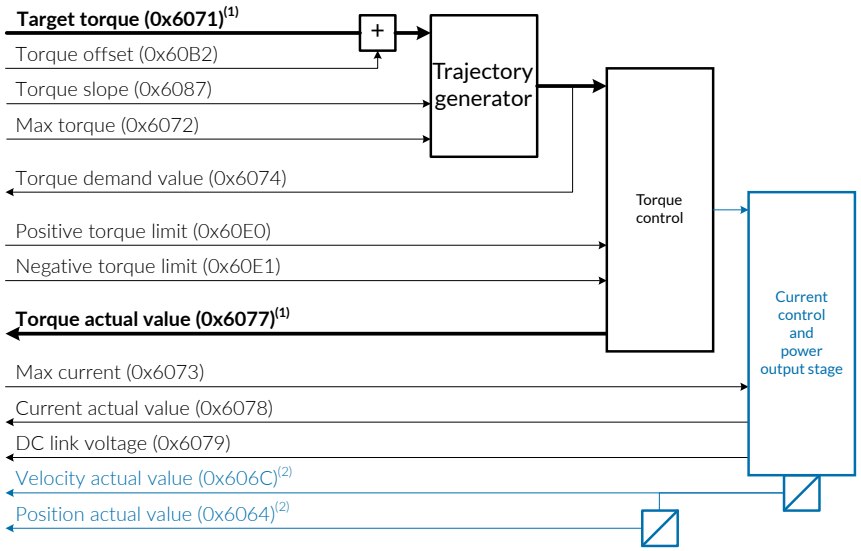


Fig. 29: Block diagram for torque mode

(1) The objects marked in bold are required and recommended process data (PDO) for the profile velocity mode.

(2) SD4x drives support these objects (marked in blue) additionally to the standard objects in the DS402 drive profile.

4.2.4 Change Operating Mode

You can easily switch between the available operating modes of the drive.

Note

As an alternative, SD4x drives allow switching the complete parameter set (see chapter 4.4 “Parameter Set Change”, page 50). You should consider this option when several parameters must be changed in addition to the operating mode.

⚠ WARNING

Risk of injury due to unintentional motor start

→ Before you change the operating mode, set the target values to 0 in order to prevent unintentional starting of the motor.

The available operating modes are contained in the object ***supported drive modes*** (0x6502). You can change the operating mode using the object ***mode of operation*** (0x6060). The currently active operating mode is returned in the object ***mode of operation display*** (0x6061).

After the operating mode was successfully changed, bit 10 (target reached) in the ***statusword*** (0x6041) is set.

The new operating mode is only applied and becomes active in the status “operation enabled”. In this status the operating mode cannot be changed again.

A new operating mode setting is only applied when the status switches to “operation enabled”. If the state machine is already in this status, it must leave the status “operation enabled” at first and then switch back into it.

Supported Drive Modes (0x6502)

This object contains the currently available operating modes for the drive as bit code. The bits set in the 32-bit object indicate that the respective operating modes are available.

Supported drive modes		Available operating modes: Bit 00: pp – profile position mode Bit 01: vl – velocity mode Bit 02: pv – profile velocity mode Bit 03: tq – profile torque mode Bit 04: hmaf – homing mode with additional feedback Bit 05: hm – homing mode Bit 06: ip – interpolated position mode Bit 07: csp – cyclic sync position mode Bit 08: csv – cyclic sync velocity mode Bit 09: cst – cyclic sync torque mode Bit 10: cstca – cyclic sync torque mode with commutation angle Bit 11: ppaf – profile position mode with additional feedback Bit 12: pvaf – profile velocity mode with additional feedback Bit 13: cspaf – cyclic sync position mode with additional feedback Bit 14: csvaf – cyclic sync velocity mode with additional feedback Bit 15: r – reserved	
DS402_0X6502_SUPPORTED_DRIVE_MODES			ID: 0x6502
RO	Volatile	Unit: –	U32

Mode of Operation (0x6060)

Via this object you can change the operating mode.

Mode of operation		Selection of the operating mode: 01: Profile position mode 02: Velocity mode 03: Profile velocity mode 04: Profile torque mode 06: Homing mode 07: Interpolated position mode 08: Cyclic sync position mode 09: Cyclic sync velocity mode 10: Cyclic sync torque mode 11: Cyclic sync torque mode with commutation angle 12: Profile position mode with additional feedback 13: Profile velocity mode with additional feedback 14: Homing mode with additional feedback 15: Cyclic sync position mode with additional feedback 16: Cyclic sync velocity mode with additional feedback	
DS402_0x6060_MODE_OF_OPERATION			ID: 0x6060
RW	Volatile	Unit: –	U8

Not all operating modes are available. As standard, the SD4x drive supports the operating modes 2 (velocity mode), 3 (profile velocity mode) and 4 (profile torque mode).

Mode of Operation Display (0x6061)

This object returns the currently active operating mode.

Mode of operation display		Active operating mode: 01: Profile position mode 02: Velocity mode 03: Profile velocity mode 04: Profile torque mode 06: Homing mode 07: Interpolated position mode 08: Cyclic sync position mode 09: Cyclic sync velocity mode 10: Cyclic sync torque mode 11: Cyclic sync torque mode with commutation angle 12: Profile position mode with additional feedback 13: Profile velocity mode with additional feedback 14: Homing mode with additional feedback 15: Cyclic sync position mode with additional feedback 16: Cyclic sync velocity mode with additional feedback	
DS402_0X6061_MODE_OF_OPERATION_DISPLAY			ID: 0x6061
RO	Volatile	Unit: –	U8

4.3 Target and Actual Values

The following chapters describe the target and actual values relevant for the drive profile (DS402). How the values are used is defined by the selected operating mode

Note

The velocity mode (vl) uses mostly its own target, actual , scaling and limiting values. These are directly described below the operating mode, see page 33.

4.3.1 General Target Values

Target Velocity (0x60FF)

This object indicates the target speed. The motor reaches this target value usually at the end of the acceleration ramp during a profile movement. With positioning operating modes, the target value applies to both moving directions.

Target velocity		Target speed in user units	
DS402_0x60FF_TARGET_VELOCITY			ID: 0x60FF
RW	Volatile	Unit: –	I32

The value is indicated in user-defined speed units. The default setting for SD4x drives is 10^{-3} *rpm*. The speed unit is scaled with the factor group ***SI unit velocity*** (0x60A9). In addition, the object ***velocity offset*** (0x60B1) can affect the resulting target value. You can define additional speed and acceleration limits.

By default, you should set the drive system so that positive values cause forward motion and negative values cause backward motion.

Target Torque (0x6071)

This object indicates the target torque.

Target torque		Target torque The value is indicated in parts per thousand (1000 = 100 %) of the rated torque, object ***motor rated torque*** (0x6076).	
DS402_0X6071_TARGET_TORQUE			ID: 0x6071
RW	Volatile	Unit: ‰	I16

For the target torque you can set an offset via the object ***torque offset*** (0x60B2). Limitations are set via the objects ***torque slope*** (0x6087), ***positive torque limit*** (0x60E0) and ***negative torque limit*** (0x60E1).

4.3.2 Actual Values

Position Actual Value (0x6064)

This object returns the actual position determined by the position measuring system (e.g. encoder).

Position actual value		Actual position in user units	
DS402_0X6064_POSITION_ACTUAL_VALUE			ID: 0x6064
RO	Volatile	Unit: –	I32

As standard, the value is indicated in (encoder) increments, see object ***SI unit position*** (0x60A8). The number of increments per revolution is defined in the *drivemaster4* software in the feedback settings.

Velocity demand value (0x606B)

This object returns the resulting target speed after ramp and limiting functions.

Velocity demand value		Speed setpoint after ramp and limiting functions The unit and direction correspond to object ***target velocity*** (0x60FF).	
DS402_0x606B_VELOCITY_DEMAND_VALUE			ID: 0x606B
RO	Volatile	Unit: –	I32

Velocity Actual Value (0x606C)

This object returns the actual speed.

Velocity actual value		Actual speed in user units The unit and direction correspond to object ***target velocity*** (0x60FF).	
DS402_0x606C_VELOCITY_ACTUAL_VALUE			ID: 0x606C
RO	Volatile	Unit: –	I32

Depending on the application (with or without a sensor) the speed actual value is either determined by a sensor or derived from the calculated rotating field.

Torque Demand (0x6074)

This object returns the resulting target torque after ramp and limiting functions.

Torque demand	Torque setpoint after ramp and limiting functions The unit and direction correspond to object ***target torque*** (0x6071).	
DS402_0x6074_TORQUE_DEMAND		ID: 0x6074

RO	Volatile	Unit: ‰	I16

Torque Actual Value (0x6077)

This object returns the actual torque.

Torque actual value		Torque actual value The unit and direction correspond to object ***target torque*** (0x6071).	
DS402_0x6077_TORQUE_ACTUAL_VALUE			ID: 0x6077
RO	Volatile	Unit: ‰	I16

Current Actual Value (0x6078)

This object returns the actual current.

Current actual value		Actual current value The value is indicated in parts per thousand (1000 = 100 %) of the rated current, object ***motor rated current*** (0x6075).	
DS402_0x6078_CURRENT_ACTUAL_VALUE			ID: 0x6078
RO	Volatile	Unit: ‰	I16

DC-Link Circuit Voltage (0x6079)

This object returns the currently applied DC link voltage.

DC link circuit voltage		DC link voltage	
DS402_0x6079_DC_LINK_CIRCUIT_VOLTAGE			ID: 0x6079
RO	Volatile	Unit: mV	U32

4.3.3 Motor Parameters

Motor Rated Current (0x6075)

This object contains the parameterized motor rated current. The value should match the specification on the motor type plate.

Motor rated current		Rated current of the motor All relative current values refer to this value.	
DS402_0x6075_MOTOR_RATED_CURRENT			ID: 0x6075
RW	Volatile	Unit: mA	U32

Motor Rated Torque (0x6076)

This object contains the parameterized motor rated torque. The value should match the specification on the motor type plate.

Motor rated torque		Rated torque of the motor All relative torque values refer to this value.	
DS402_0x6076_MOTOR_RATED_TORQUE			ID: 0x6076
RW	Volatile	Unit rotary motor: mNm Unit linear motor: Nm	U32

Motor Type (0x6402)

Via this object you can select the motor type.

Motor type		Selection of the motor type: 0x0000: Non-standard motor 0x0001: Phase modulated DC motor 0x0002: Frequency controlled DC motor 0x0003: PM synchronous motor 0x0004: AC synchronous sinewave wound field 0x0005: AC synchronous reluctance switched 0x0006: AC asynchronous induction polyphase wound rotor 0x0007: AC asynchronous induction squirrel cage 0x0008: AC synchronous step 0x0009: Micro-step stepper motor 0x000A: AC synchronous sinusoidal PM 0x000B: AC synchronous brushless PM trapezoidal 0x000C: AC synchronous reluctance sync 0x000D: DC commutator PM 0x000E: DC commutator wound field series 0x000F: DC commutator wound field shunt 0x0010: DC commutator wound field compound 0x0011 to 0x7FFF: No motor type assigned	
DS402_0X6402_MOTORTYPE			ID: 0x6402
RO	Volatile	Unit: increment	U16

4.3.4 SI Units

SI (French *Système international d'unités*) is the unit system based on the international system of quantities. The following SI unit objects are the basis for the units of the target and actual values in the operating modes (except velocity mode (VL), see chapter "Target and Actual Values in Velocity Mode", page 33).

At present, the SI unit objects for SD4x drives are read-only objects.

Structure of the SI unit objects

The objects for SI units are 32 bits in site and divided into 4 parts:

Bit number			
31 ... 24	**23 ... 16**	**15 ... 8**	**7 ... 0**
Prefix	SI numerator	SI denominator	Reserved(00)

The documents CiA 890 and CiA 402 describe, which variable values correspond to which Si units or prefixes. A summarizing table with the possible prefixes and SI units of both standard can be found in the Appendix.

Position (0x60A8)

This object contains the position unit.

Position		Position unit	
DS402_0X60A8_SI_UNIT_POSITION			ID: 0x60A8
RW	Volatile	Unit: –	U32

With SD4x drives, the position is indicated in increments by default. This corresponds to the following variable content:

Byte	**3**	**2**	**1**	**0**
Value	00	B5	00	00
Content	10^0	Increments	1	–

Therefore, a position setting of 1244 means 1244 increments.

Velocity (0x60A9)

This object contains the speed unit.

Velocity		Velocity unit	
Object: DS402_0X60A9_SI_UNIT_VELOCITY			ID: 0x60A9
RW	Volatile	Unit: –	U32

With SD4x drives, the speed is indicated in 10^{-3} × rpm by default. The unit rpm stands for *revolution per minute*. This corresponds to the following variable content:

Byte	**3**	**2**	**1**	**0**
Value	FD	B4	47	00
Content	10^{-3}	Revolution	Minute	–

Therefore, a speed setting of 2500000 means 2500 rpm.

Acceleration (0x60AA)

This object the unit for acceleration and deceleration.

Acceleration		Acceleration/deceleration unit	
DS402_0X60AA_SI_UNIT_ACCELERATION			ID: 0x60AA
RW	Volatile	Unit: –	U32

With SD4x drives, the acceleration/deceleration is indicated in 10^{-3} × 1/s² (*revolutions per square second*) by default. This corresponds to the following variable content:

Byte	**3**	**2**	**1**	**0**
Value	FD	B4	57	00
Content	10^{-3}	Revolution	Square second	–

Therefore, an acceleration setting of 300000 means 300 1/s².

Jerk (0x60AB)

This object contains the jerk unit.

Jerk		Jerk unit	
DS402_0X60AB_SI_UNIT_JERK			ID: 0x60AB
RW	Volatile	Unit: –	U32

With SD4x drives, the jerk is indicated in 10^{-3} × 1/s³ (*revolutions per cubic second*) by default. This corresponds to the following variable content:

Byte	**3**	**2**	**1**	**0**
Value	FD	B4	A0	00
Content	10^{-3}	Revolution	Cubic second	–

Therefore, a ruck setting of 2000 means 2 1/s³.

4.3.5 Scaling

In SD4x drives, only the scaling object ***polarity*** is implemented.

Polarity (0x607E)

Via this object, you can invert the moving direction of the motor.

Polarity		Moving direction of the motor	
DS402_0X607E_POLARITY			ID: 0x607E
RW	Volatile	Unit: –	U8

If the polarity parameter is not set, a clockwise rotating motor will move in positive direction, an anticlockwise rotating motor will move in negative direction. If the polarity parameter is set, the moving direction is reversed.

The 8-bit object is divided as follows:

Bit number			
7	**6**	**5 … 1**	**0**
Position polarity	Velocity polarity	Reserved	Global polarity

- Bit 7 applies to positioning modes (e.g. profile position mode, interpolated position mode, cyclic sync position mode).
- Bit 6 applies to velocity modes (e.g. profile velocity mode, cyclic sync velocity mode).
- If bit 0 is set too the value 1, the inversion applies to all operating modes. In this case, the values of the bits 6 and 7 are irrelevant and should not be set.

4.3.6 Offsets

Velocity Offset (0x60B1)

This object indicates the offset for target speeds.

Velocity offset		Offset for target speeds The unit is the same as in object ***target velocity*** (0x60FF).	
DS402_0X60B1_VELOCITY_OFFSET			ID: 0x60B1
RW	Volatile	Unit: –	I32

In the positioning operating modes, this object contains the input value for the velocity feed forward control. In the velocity modes, it contains the velocity offset of the drive. That means, in the velocity modes (except chapter “Velocity Mode (VL)”, page 32) the target speed is made up of the ***target velocity*** and the ***velocity offset***.

Torque Offset (0x60B2)

This object indicates the offset for target torque setting.

Torque offset		Offset for target torque The value is indicated in parts per thousand (1000 = 100 %) of the rated torque, object ***motor rated torque*** (0x6076).	
DS402_0X60B2_TORQUE_OFFSET			ID: 0x60B2
RW	Volatile	Unit: ‰	I16

In the positioning and velocity operating modes, this object contains the input value for the torque feed forward control. In the velocity modes, it contains the velocity offset of the drive. That means, in den torque modes the target torque is made up of the ***target torque*** and the ***torque offset***.

4.3.7 Limitations

Max Torque (0x6072)

This object contains the maximum permitted value for the torque.

Max torque		Torque limitation for torque profile The value is indicated in parts per thousand (1000 = 100 %) of the rated torque, object ***motor rated torque*** (0x6076).	
DS402_0x6072_MAX_TORQUE			ID: 0x6072
RW	Volatile	Unit: ‰	U16

Max Current (0x6073)

Via this object you can set a limitation for the torque-generating current in the motor.

Max current		Current limitation of the active current The value is indicated in parts per thousand (1000 = 100 %) of the rated current, object ***motor rated current*** (0x6075).	
DS402_0x6073_MAX_CURRENT			ID: 0x6073
RW	Volatile	Unit: ‰	U16

Max Profile Velocity (0x607F)

Via this object you can set the maximum speed.

Max profile velocity		Speed limit for the motion profile The value applies to both directions. The unit is the same as in object ***target velocity*** (0x60FF).	
DS402_0x607F_MAX_PROFILE_VELOCITY			ID: 0x607F
RW	Volatile	Unit: –	U32

Max Motor Speed (0x6080)

Via this object you can set the maximum speed for the motor. The value should be taken from the motor data sheet and is used to protect the motor.

Max motor speed		Maximum speed specified by the motor The value applies to both directions. The unit is the same as in object ***target velocity*** (0x60FF).	
DS402_0x6080_MAX_MOTOR_SPEED			ID: 0x6080
RW	Volatile	Unit: –	U32

Positive Torque Limit (0x60E0)

Via this object you can set the maximum positive torque in the motor. The positive torque causes a positive velocity with motor operation and a negative velocity with regenerative operation.

Positive torque limit		Upper limit for torque setpoint The value is indicated in parts per thousand (1000 = 100 %) of the rated torque, object ***motor rated torque*** (0x6076).	
DS402_0X60E0_POSITIVE_TORQUE_LIMIT_VALUE			ID: 0x60E0
RW	Volatile	Unit: ‰	I16

Negative Torque Limit (0x60E1)

Via this object you can set the maximum negative torque in the motor. The negative torque causes a negative velocity with motor operation and a positive velocity with re-generative operation.

Negative torque limit		Lower limit for torque setpoint The value is indicated in parts per thousand (1000 = 100 %) of the rated torque, object ***motor rated torque*** (0x6076).	
DS402_0X60E1_NEGATIVE_TORQUE_LIMIT_VALUE			ID: 0x60E1
RW	Volatile	Unit: ‰	I16

4.3.7.1 Acceleration and Deceleration Limits

Max Acceleration (0x60C5)

Via this object you can set the maximum permissible acceleration. This is used to limit the acceleration to an acceptable value in order to prevent damage to the motor and moving parts.

Max acceleration		Parameterized maximum acceleration of the system The unit is set in object ***acceleration*** (0x60AA).	
DS403_0X60C5_MAX_ACCELERATION			ID: 0x60C5
RW	Volatile	Unit rotary motor: 1/s² Unit linear motor: mm/s²	U32

Max Deceleration (0x60C5)

Via this object you can set the maximum permissible deceleration. This is used to limit the deceleration to an acceptable value in order to prevent damage to the motor and moving parts.

Max deceleration		Parameterized maximum deceleration of the system The unit is set in object ***acceleration*** (0x60AA).	
DS403_0X60C6_MAX_DECELERATION			ID: 0x60C6
RW	Volatile	Unit rotary motor: 1/s² Unit linear motor: mm/s²	U32

Profile Acceleration (0x6083)

Via this object you can set the acceleration.

Profile acceleration		Acceleration ramp The unit is set in object ***acceleration*** (0x60AA).	
DS402_0x6083_PROFILE_ACCELERATION			ID: 0x6083
RW	Volatile	Unit rotary motor: 1/s² Unit linear motor: mm/s²	U32

Profile Deceleration (0x6084)

Via this object you can set the deceleration.

Profile deceleration		Deceleration ramp The unit is set in object ***acceleration*** (0x60AA).	
DS402_0x6084_PROFILE_DECELERATION			ID: 0x6084
RW	Volatile	Unit rotary motor: 1/s² Unit linear motor: mm/s²	U32

Quick Stop Deceleration (0x6085)

Via this object you can set the deceleration in the event of a quick stop. In the Option Codes (p. 28) you can configure when a quick stop is triggered.

Quick stop deceleration		Quick stop ramp The unit is set in object ***acceleration*** (0x60AA).	
DS402_0x6085_QUICK_STOP_DECELERATION			ID: 0x6085
RW	Volatile	Unit rotary motor: 1/s² Unit linear motor: mm/s²	U32

Torque Slope (0x6087)

Via this object you can set the rate of change of the torque.

Torque slope		Torque change rate The value is indicated in parts per thousand (1000 = 100 %) of the rated torque, object ***motor rated torque*** (0x6076).	
DS402_0x6087_TORQUE_SLOPE			ID: 0x6087
RW	Volatile	Unit: ‰/s	U32

4.3.8 Others

Digital Inputs (0x60FD)

This object returns the states of the digital inputs of the SD4x drive. The object is 32-bit variable. If a bit is logical 1, the function assigned to this bit is active.

Digital inputs		States of the digital inputs Bit assignment: Bit 0: Negative limit switch Bit 1: Positive limit switch Bit 2: Home switch Bit 3: Interlock active Bits 4 to 15: Reserved Bits 16 to 31: Inputs 1 to 16 - configurable	
DS402_0X60FD_DIGITAL_INPUTS			ID: 0x60FD
RO	Volatile	Unit: –	U32

Bit 0 returns the status of the negative limit switch. With SD4x drives, this bit corresponds to the function “(16) Negative limit switch” for digital inputs.

Bit 1 returns the status of the positive limit switch. With SD4x drives, this bit corresponds to the function “(15) Positive limit switch” for digital inputs.

Bit 3 returns the status of the interlock input. If this input signal is deactivated, the SD4x drive must switch to the status “switch on disabled” or “fault reaction active”. This disables the power output stage of the drive and prevents is from being reactivated. With SD4x drives, the bit corresponds to the inverted function “(01) Switch on” for digital inputs. Example: If the digital input 2 of the SD4x is parameterized with the function “(01) Switch on” and the physical input 2 is logical 0, then bit 3 in the object ***digital inputs*** is logical 1 and bit 17 logical 0. If the physical input 2 is logical 1, then bit 3 is be logical 0 and bit 17 logical 1.

Bit 16 to 31 correspond to the inputs of the device. The used number of bits depends on the actual number of physical inputs provided by the SD4x drive.

Digital Outputs (0x60FE)

Via this object you can retrieve as well as set the states of the digital outputs of the SD4x drive.

It provides the following subobjects: status variable ***physical outputs*** (0x60FE:1) and masking variable ***bit mask*** (0x60FE:2). Both subobjects are 32-bis variables and contain the bit assignment of the digital outputs. The masking variable allows taking over the control of individual digital outputs from the device. If the masking variable for an output is logical 0, the device controls this output and its current status is returned via the status variable. As soon as the bit in the masking variable is set to logic 1, the status variable controls the corresponding output.

Digital outputs		Access to the digital outputs Bit assignment: Bit 0: Motor holding brake active Bits 1 to 15: Reserved Bits 16 to 31: Outputs 1 to 16 - configurable Sub-IDs: ▸ 0x60FE:0 = Highest subindex supported (2) ▸ 0x60FE:1 = Physical outputs ▸ 0x60FE:2: Bit mask	
DS402_0X60FE_DIGITAL_OUTPUTS			ID: 0x60FE
RW	Volatile	Unit: –	BYTEARRAY

Bit 0 returns the status or controls the output of the motor holding brake. With SD4x drives, this bit corresponds to the function “(05) Motor holding brake” for digital outputs.

Bit 16 to 31 correspond to the outputs of the device. The used number of bits depends on the actual number of physical outputs provided by the SD4x drive.

Highest subindex supported (0x60FE:0)

In its subindex 0, this object contains the last subindex of the object (in this case 2).

Physical outputs (0x60FE:1)

Depending on the masking variable ***bit mask*** (0x60FE:2) this object provides either the current states of the digital outputs or controls individual outputs. Active bits are logical 1, non-active bits are logical 0.

Bit mask (0x60FE:2)

This object can transfer the control of individual digital outputs from the device to the object ***physical outputs*** (0x60FE:1). A logical 1 for a bit means that the output is controlled via the object; a logical 0 for a bit means that the status of the output is read.

Example:

- Output 1: parameterized function “(01) Ready for operation”
- Output 2: parameterized function “(03) Switched on”

The status of the physical output 1 is to be observed while the physical output 2 is to return the logical status 1 (active).

To achieve this, set the bits in the object ***bit mask*** as follows: bit 16 to logical 0 and bit 17 to logical 1. Then, the logical status of output 2 is defined by bit 17 in object ***physical outputs***, bit 16 still corresponds to the logical status of the physical output 1. As long as bit 16 in the masking variable is logical 0, the control cannot change the status of output 1.

Error Code (0x603F)

This object returns the error code of the last error that occurred.

Error code		Last saved error code accoring to drive profile DS402. The value 0 means that no error is present in the device.	
DS402_0x603F_ERROR_CODE			ID: 0x603F
RO	Volatile	Unit: –	U16

The error codes hardware-dependent. For a detailed description of the error codes refer to the hardware documentation of your device.

4.4 Parameter Set Change

You can create up to 64 parameter sets for one SD4x drive. Switching to another parameter set is only possible, when the output stage is disabled. When the parameter set change was successful, the device is restarted with the selected parameter set.

You can switch the parameter set via the following control sources: *drivemaster4* software, digital inputs or fieldbus. The control source also determines the parameter set that is activated on device startup.

Selection source (0x2037)

This object determines the control source for selecting and switching the parameter set. Internally (within the *drivemaster4* software) you can find this object under index_0x0037.

The following file options are available:

- **00: Fixed selection (EEPROM)**:
  This option is the factory setting. The device ist started with the last selected parameter set. You can switch the parameter set via the *drivemaster4* software as well as via fieldbus system.
- **(01) Digital inputs:**
  The device ist started with parameter set 0 and switches directly to the parameter set specified by the digital inputs. You can switch the parameter set via the digital inputs only.
- **(02) Bus system:**
  The device ist started with parameter set 0. You can switch the parameter set via the fieldbus system only.

The parameter set change via fieldbus is decribed in more detail below. For a description of the parameter set change via *drivemaster4* software or the digital inputs refer to the document “Drive Controller SD4x – Functions and Parameters”.

4.4.1 Parameter Set Change via Fieldbus

The parameter set is switched via the object 0x202E (selection).

The following requirements must be met for a successful parameter set change via the bus system:

1. The object 0x2037 (selection source) is set to **(02) Bus system** or **(00) Fixed selection (EEPROM)**.
2. The output stage is disabled.

Selection (0x202E)

The object contains the number of the currently selected parameter set. Internally (within the *drivemaster4* software) you can find this object under index_0x002E.

In order to use the object 0x202E for switching between existing parameter sets, only the respective parameter set number must be written into the object.

A parameter set change restarts the device and therefore the fieldbus connection is lost for a short time – i.e. no device control.

5 Diagnosis

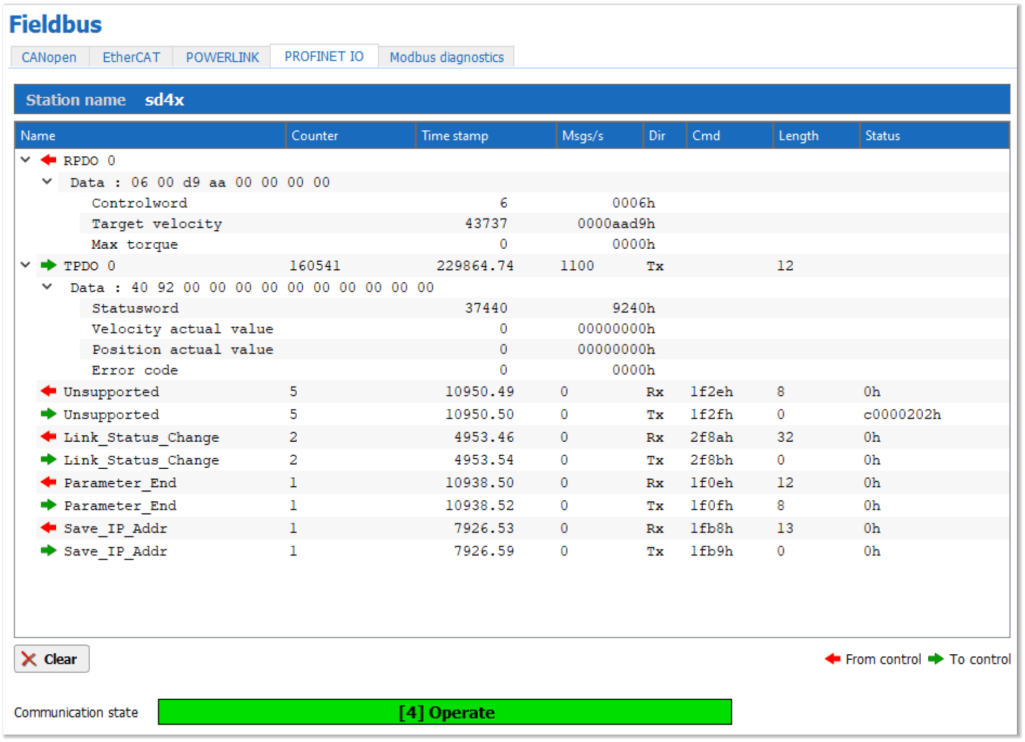
5.1 Diagnosis in *drivemaster4*

In the user interface *drivemaster4*, the page “Diagnosis → Fieldbus” displays the transmitted telegrams between drive and PLC in the tab “PROFINET IO”.

In addition, the IP configuration as well as diagnostic data are displayed on the right side of the page.

5.1.1 Telegrams

The following figure shows the logged telegrams on the PROFINET IO diagnosis page.



Name	Counter	Time stamp	Msgs/s	Dir	Cmd	Length	Status
RPDO 0							
Data : 06 00 d9 aa 00 00 00 00							
Controlword		6	0006h				
Target velocity		43737	0000aad9h				
Max torque		0	0000h				
TPDO 0	160541	229864.74	1100	Tx		12	
Data : 40 92 00 00 00 00 00 00 00 00 00 00							
Statusword		37440	9240h				
Velocity actual value		0	00000000h				
Position actual value		0	00000000h				
Error code		0	0000h				
Unsupported	5	10950.49	0	Rx	1f2eh	8	0h
Unsupported	5	10950.50	0	Tx	1f2fh	0	c0000202h
Link_Status_Change	2	4953.46	0	Rx	2f8ah	32	0h
Link_Status_Change	2	4953.54	0	Tx	2f8bh	0	0h
Parameter_End	1	10938.50	0	Rx	1f0eh	12	0h
Parameter_End	1	10938.52	0	Tx	1f0fh	8	0h
Save_IP_Addr	1	7926.53	0	Rx	1fb8h	13	0h
Save_IP_Addr	1	7926.59	0	Tx	1fb9h	0	0h

Fig. 30: PROFINET IO bus activities

Station name

The top line indicates the PROFINET IO station name. The station name is stored in the following object:

Object 155 (0x009B) Station name

The station name consists of the following bytes:

- length indication of the current station name: 2 byte
- station name in ASCII format: max. 240 byte

Example:

- Data length of station name: **4 byte**
- Station name: **sd4x**

Representation in the object browser of the *drivemaster4* software:

Object **PROFINET IO - Station name**

	00	01	02	03	04	05	06	07	08
0000	04	00	73	64	34	78	00	00	00

Fig. 31: Station name

Output window

You can unfold the lines of the cyclic telegrams RPDO x and TPDO x to see their individual parameterized objects.

The acyclic telegrams are displayed in pairs as request and response telegram. User data are not displayed because more than 1000 bytes could be transmitted here and therefore the readability would be lost.

The following information is displayed in the output window for the telegrams:

Name	telegram name
Counter	counter of telegram frequency
Timestamp	drive-internal time stamp
Msgs/s	number of sent telegrams per second
Dir	telegram direction from the point of view of the drive: ▸ Rx: telegram was received by the drive. ▸ Tx: telegram was sent by the drive.

Additional information for acyclic telegrams:

cmd	command number of acyclic telegram
Length	length of telegram in bytes
Status	error status for the execution of the command

Clear

You can clear the output window by click on the button “Clear”. Afterwards, logging the detected messages is restarted.

Communication state

This field displays the communication status of the bus system. The following states are possible: Unknown, Offline, Stop, Idle and Operate.

5.1.2 Diagnostic data

The following figure shows the IP configuration for the Ethernet communication and the diagnostic data on the right side of the PROFINET IO diagnosis page. The indicated values are typical values for the status OPERATE.

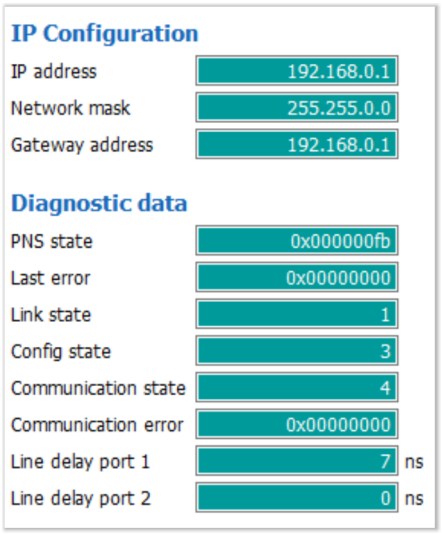


Fig. 32: IP configuration and diagnostic data

The data are stored in the following objects.

Object 154 (0x009A) diagnostic data

The diagnostic data consist of 8 values with 4 bytes each:

Variable	Value	Description	Example value
PNS state	Bit	Status of the protocol stack in NetX90	**0x000000FB**
	0	Device information is set	**1**
	1	PROFINET stack is started	**1**
	2	Reserved	**0**
	3	At least one API is present	**1**
	4	Module 0 is available	**1**
	5	Module 0 and submodule 1 are available	**1**
	6	Network communication is permitted	**1**
	7	Network communication is activated	**1**
	8	Configuration is locked	**0**
	9	Fatal error occurred	**0**
	10	Diagnostic data with fatal error available	**0**
	11	Data set requiring maintenance exists	**0**
	12	Maintenance request data set exists	**0**
Last error		Last error is displayed	**0x00000000**
Link state	0	No information available	**0x00000001**
	1	Physical link works correctly	
	2	Physical link works with low speed	
	3	No physical link available	

Variable	Value	Description	Example value
Config state	0	Not configured	**0x00000003**
	1	Configured via DBM files	
	2	Error during configuration via DBM files	
	3	Configured via application	
	4	Configuration via application in progress	
	5	Error during configuration via application	
	6	Configured via warm boot parameters	
	7	Configuration via warm boot parameters in progress	
	8	Error during configuration via warm boot parameters	
Communication state	0	UNKNOWN (black)	**0x00000004**
	1	OFFLINE (gray)	
	2	STOP (red)	
	3	IDLE (yellow)	
	4	OPERATE (green)	
Communication error		Error status in communication channel is displayed	**0x00000000**
Line delay port 1		Signal delay [ns] for port 1 is displayed	**0x00000007**
Line delay port 2		Signal delay [ns] for port 2 is displayed	**0x00000000**

Representation in the object browser of the *drivemaster4* software:

Object **PROFINET IO - Diagnostic data**

	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
0000	fb	00	00	00	00	00	00	00	01	00	00	00	03	00	00	00
0010	04	00	00	00	00	00	00	00	07	00	00	00	00	00	00	00

Fig. 33: Diagnostic data

Object 156 (0x009C) IP address

The IP address consist of 3 values with 4 bytes each:

Description	Example value
IP address	**C0:A8:00:01**
Network mask	**FF:FF:00:00**
Gateway	**C0:A8:00:01**

Representation in the object browser of the *drivemaster4* software:

Object **PROFINET IO - IP address**

	00	01	02	03	04	05	06	07	08	09	0a	0b
0000	01	00	a8	c0	00	00	ff	ff	01	00	a8	c0

Fig. 34: IP address

5.2 Diagnosis in PLC

No connection with drive

If the connection to a device cannot be established, check that the station name, the IP address and the MAC address match the data in the drive. If not, set the desired data in the *drivemaster4* software and write them to the drive.

6 Reference

Other documents

The following documents provide more information on this topic:

Supplier	Document
SIEB & MEYER AG	▸ drivemaster4 - User Manual ▸ Drive System SD4 – Ethernet Configuration
PROFIBUS Nutzerorganisation e.V. (P-NO)	As registered user you can download documents for PROFINET from the website of the organization.
CiA e.V.	▸ CiA 301, version 4.2.0. CANopen application layer and communication profile ▸ CiA 402, version 5.0.0. CANopen device profile drives and motion control ▸ CiA 890, version 5.0.0. Presentation of SI units and prefixes

Websites

The following websites provide more information on this topic:

Supplier	Document
SIEB & MEYER AG	www.sieb-meyer.com
PROFIBUS Nutzerorganisation e.V. (P-NO)	www.profibus.com
CiA e.V.	www.can-cia.org
Siemens AG	www.siemens.com

7 Index

P

S

T

V