



Drive Controller SD4x

EtherCAT Connection

A getting started guide for users



Copyright

Original instructions, Copyright © 2026 SIEB & MEYER AG

All rights reserved.

This manual or extracts thereof may only be copied with the explicit authorization by SIEB & MEYER AG.

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
info@sieb-meyer.de
<http://www.sieb-meyer.com>

SIEB & MEYER Shenzhen Trading Co. Ltd.
Room 316, Haixiang Plaza,
No. 1048 Nanhai Avenue,
Zhaoshang Subdistrict, Nanshan District,
Shenzhen City
China

Phone: +86 755 2681 1417 / +86 755 2681 2487
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
info@sieb-meyer.tw

1	About this Manual.....	5
1.1	Abbreviations.....	5
2	Drive Control (CiA 402).....	7
2.1	Service Data Objects (SDO).....	7
2.2	Process Data Objects (PDO).....	8
2.3	State Machine.....	9
2.3.1	Commands.....	10
	Controlword (0x6040).....	10
2.3.2	States.....	11
	Statusword (0x6041).....	11
2.3.3	Option Codes.....	12
	Abort Connection Option Code (0x6007).....	12
	Quick Stop Option Code (0x605A).....	13
	Shutdown Option Code (0x605B).....	13
	Disable Operation Option Code (0x605C).....	13
	Halt Option Code (0x605D).....	13
	Fault Reaction Option Code (0x605E).....	14
2.4	Operating Modes.....	14
2.4.1	Cyclic Synchronous Velocity Mode (CSV).....	14
2.4.2	Profile Velocity Mode (PV).....	16
2.4.3	Velocity Mode (VL).....	16
2.4.3.1	Target and Actual Values in Velocity Mode.....	18
	vl Target Velocity (0x6042).....	18
	vl Velocity Demand (0x6043).....	18
	vl Velocity Actual Value (0x6044).....	18
	vl Velocity Min Max Amount (0x6046).....	18
	vl Velocity Acceleration (0x6048).....	19
	vl Velocity Deceleration (0x6049).....	20
	vl Velocity Quick Stop (0x604A).....	20
	vl Set-point Factor (0x604B).....	21
	vl Dimension Factor (0x604C).....	22
2.4.4	Torque Mode.....	23
2.4.5	Change Operating Mode.....	23
	Supported Drive Modes (0x6502).....	24
	Mode of Operation (0x6060).....	24
	Mode of Operation Display (0x6061).....	25
2.5	Target and Actual Values.....	25
2.5.1	General Target Values.....	25
	Target Velocity (0x60FF).....	25
	Target Torque (0x6071).....	26
	Interpolation Time Period (0x60C2).....	26
	Motion Profile Type (0x6086).....	27
2.5.2	Actual Values.....	27
	Position Actual Value (0x6064).....	27
	Velocity demand value (0x606B).....	27
	Velocity Actual Value (0x606C).....	27
	Torque Demand (0x6074).....	28
	Torque Actual Value (0x6077).....	28
	Current Actual Value (0x6078).....	28
	DC-Link Circuit Voltage (0x6079).....	28
2.5.3	Motor Parameters.....	28
	Motor Rated Current (0x6075).....	28
	Motor Rated Torque (0x6076).....	28
	Motor Type (0x6402).....	29
2.5.4	SI Units.....	29
	Position (0x60A8).....	29
	Velocity (0x60A9).....	30
	Acceleration (0x60AA).....	30
	Jerk (0x60AB).....	30
2.5.5	Scaling.....	31
	Polarity (0x607E).....	31
2.5.6	Offsets.....	31
	Velocity Offset (0x60B1).....	31
	Torque Offset (0x60B2).....	31



2.5.7	Limitations.....	32
	Max Torque (0x6072).....	32
	Max Current (0x6073).....	32
	Max Profile Velocity (0x607F).....	32
	Max Motor Speed (0x6080).....	32
	Positive Torque Limit (0x60E0).....	33
	Negative Torque Limit (0x60E1).....	33
2.5.7.7	Acceleration and Deceleration Limits.....	33
	Max Acceleration (0x60C5).....	33
	Max Deceleration (0x60C6).....	33
	Profile Acceleration (0x6083).....	33
	Profile Deceleration (0x6084).....	34
	Quick Stop Deceleration (0x6085).....	34
	Torque Slope (0x6087).....	34
2.5.8	Others.....	34
	Digital Inputs (0x60FD).....	34
	Digital Outputs (0x60FE).....	35
	Error Code (0x603F).....	36
2.6	Parameter Set Change.....	36
2.6.1	Parameter Set Change via Fieldbus.....	37
3	Control via EtherCAT.....	39
3.1	EtherCAT Network.....	39
3.2	PLC Connection (TwinCAT3).....	39
3.2.1	EtherCAT State Machine (ESM).....	45
3.2.2	SDO Communication.....	46
3.2.3	Addressing and Identification.....	50
3.2.3.1	Auto Increment Address.....	51
3.2.3.2	EtherCAT Address.....	51
	Saving the EtherCAT Address in the SubDevice.....	52
3.2.3.3	Hot Connect.....	53
	Configured Station Alias (ADO 0x0012).....	53
	Explicit Device Identification (ADO 0x0134).....	54
3.2.4	General Objects.....	56
	Object 0x1000 – Device Type.....	56
	Object 0x1600 to 0x1603 – RPDO Mapping Parameter.....	57
	Object 1A00h to 1A03h – TPDO Mapping Parameter.....	58
	Object 0x1C00 – Sync Manager Communication Type.....	59
	Object 0x1C12 – RPDO Assignment.....	60
	Object 0x1C13 – TPDO Assignment.....	60
3.3	Supported Synchronization Types.....	61
3.3.1	Free Run.....	62
3.3.2	DC Synchronous (Distributed Clocks).....	62
4	Drive Parameterization and Diagnostics.....	64
4.1	Parameterization in drivemaster4.....	64
4.2	Diagnosis in drivemaster4.....	64
4.2.1	Telegrams.....	65
4.3	Troubleshooting: State Machine Ignores Control Commands.....	66
5	Reference.....	67
	Appendix.....	68
A	Prefixes (from CiA 303-2).....	68
B	Units (from CiA 303-2 and CiA 402).....	69
	Index.....	71

1 About this Manual

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

A basic description of the EtherCAT network can be found on the website of the EtherCAT Technology Group (www.ethercat.org).

The implemented protocol EtherCAT CoE 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.

With the CoE protocol, EtherCAT provides the same communication mechanisms as the CANopen standard:

- ▶ 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 and of the EtherCAT Technology Group ETG, see [section "Other documents", page 67](#).

1.1 Abbreviations

CAN	C ontroller A rea N etwork
CiA	C AN in A utomation
CoE	C ANopen o ver E therCAT
CSV	c yclic s ynchronous v elocity mode
DC	D istributed C locks
EDS	E lectronic D ata S heet (device description files in ASCII format)
ESI	E therCAT S ubDevice I nformation
ESM	E therCAT S tate M achine
ETG	E therCAT T echnology G roup
EtherCAT	E thernet for C ontrol A utomation T echnology
MDevice	M ain D evice (formerly master)
PDO	p rocess d ata o bject
PLC	p rogrammable l ogic c ontrol
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
SM	S ync M anager
SoE	S ERCOS o ver E therCAT



SU Sync Unit
SubDevice SubordinateDevice (formerly slave)
Tx transmit
VL velocecity mode

2 Drive Control (CiA 402)

The drive controllers of the device family SD4x support the drive profile CiA 402. That means, the drive control described in this chapter is implemented according to the standards DS301 and DS402. The associated drive objects have the prefix “DS301” or “DS402”.

The drive profile CiA 402 defines objects, which are classified as [process data object \(PDO\)](#) 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 ([cyclic synchronous velocity 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

2.1 Service Data Objects (SDO)

Service data objects are exchanged acyclically. They are used for data that is exchanged infrequently and not in real time. During the boot phase of the fieldbus, for example, the fieldbus control (MDevice) and the fieldbus participant (SubDevice) negotiate the PDO mapping via SDOs.

The MDevice triggers each SDO access by sending a request telegram to the SubDevice. Since the request is acyclic, it may take several fieldbus cycles until the SubDevice responds.

The fieldbus control transmits SDO telegrams with subordinate priority. That means, if the total fieldbus capacity is used up by PDO telegrams, SDO communication is not possible anymore.

An SDO request includes 2 addresses:

- ▶ SubDevice address (e.g. ADS address or ID)
- ▶ object address (e.g. object index and object subindex)

One request can address individual objects or entire object blocks. The exact structure and sequence of an SDO request depends on the used fieldbus system and can vary.

2.2 Process Data Objects (PDO)

Process data objects transmit the user data that are exchanged cyclically and in real time.

During the boot phase of the fieldbus, the fieldbus control (MDevice) checks and negotiates, which process data are to be exchanged. This depends on whether a fieldbus participant (SubDevice) supports predefined or configurable PDOs. SD4x drives support configurable PDOs.

Not all data objects of a SubDevice can be processed in real time. Thus, not all data can be exchanged via PDOs. Usually, the fieldbus-specific description files (e.g. ESI, XML or EDS file) specify which data is to be exchanged via PDOs.

Note

Modern fieldbus systems come with sufficient bandwidth to transmit a lot of user data with high speeds. However, we still recommend to exchange data without real-time requirements via service data objects.

2.3 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.

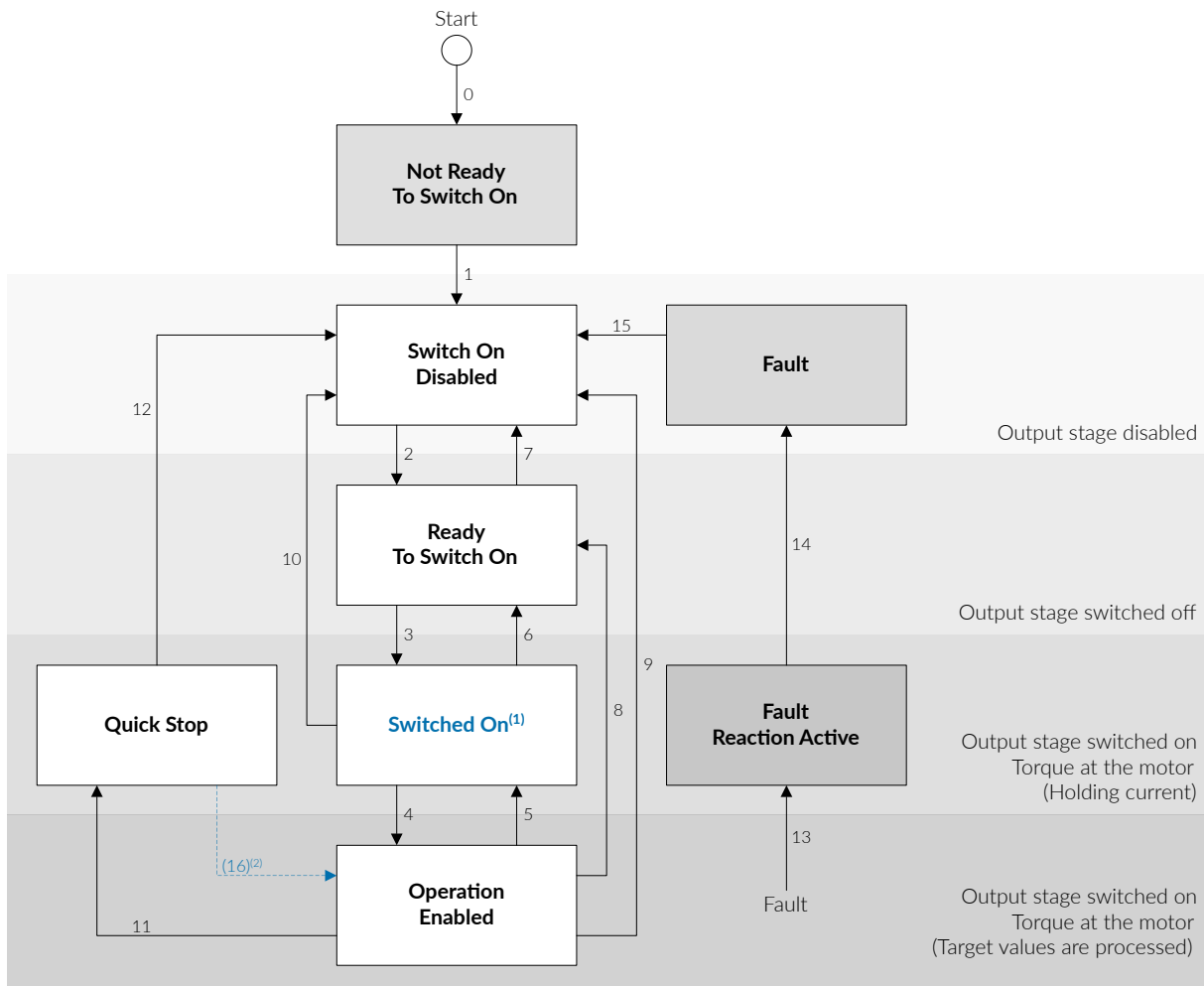


Fig. 1: Drive states and state transitions

- (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 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 only automatically and not via commands. The numbers at the arrows are transitions triggered by commands. Which command triggers which transition is described in [chapter 2.3.1 “Commands”, page 10](#).

Note

In addition, an external event can change the status of the state machine and operation enable via fieldbus is not possible anymore. For further information refer to [chapter 4.3 “Troubleshooting: State Machine Ignores Control Commands”, page 66](#).

2.3.1 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.

2.3.2 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: <i>Drive setup tool (DRS) active*</i> Bit 9: Remote Bit 10: Target reached Bit 11: <i>Current limit reached*</i> Bit 12: Mode-specific 1 Bit 13: Mode-specific 2 Bit 14: <i>Safe Torque Off (STO)*</i> Bit 15: <i>Initialization finished*</i> <i>*Bit assignments shown in italics are SIEB & MEYER specific and may differ with other drive manufacturers.</i>	
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) and 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, the halt bit in the controlword and the current status:

- ▶ Operating modes: bit 10 is set as soon as the target value for speed, position or torque is reached.
Example for velocity modes (CSV/PV/VL): The target value is reached 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]".
- ▶ 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.
- ▶ Halt bit in controlword (bit 8) is active: bit 10 is set as soon as the drive comes to standstill.
- ▶ Changing the operating mode: bit 10 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.

2.3.3 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: –	l16

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: –	l16

The ramps are defined in the operating mode.

2.4 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.

2.4.1 Cyclic Synchronous Velocity Mode (CSV)

The cyclic synchronous velocity mode (CSV mode) enables an external trajectory calculation or position control by a higher-ranking control. Then, the drive follows the target values of the control. For smooth interpolation of the target values, the drive and control must be synchronized. In addition, the **interpolation time period** should match the fieldbus cycle time.

In order to control the drive in the CSV 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.

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

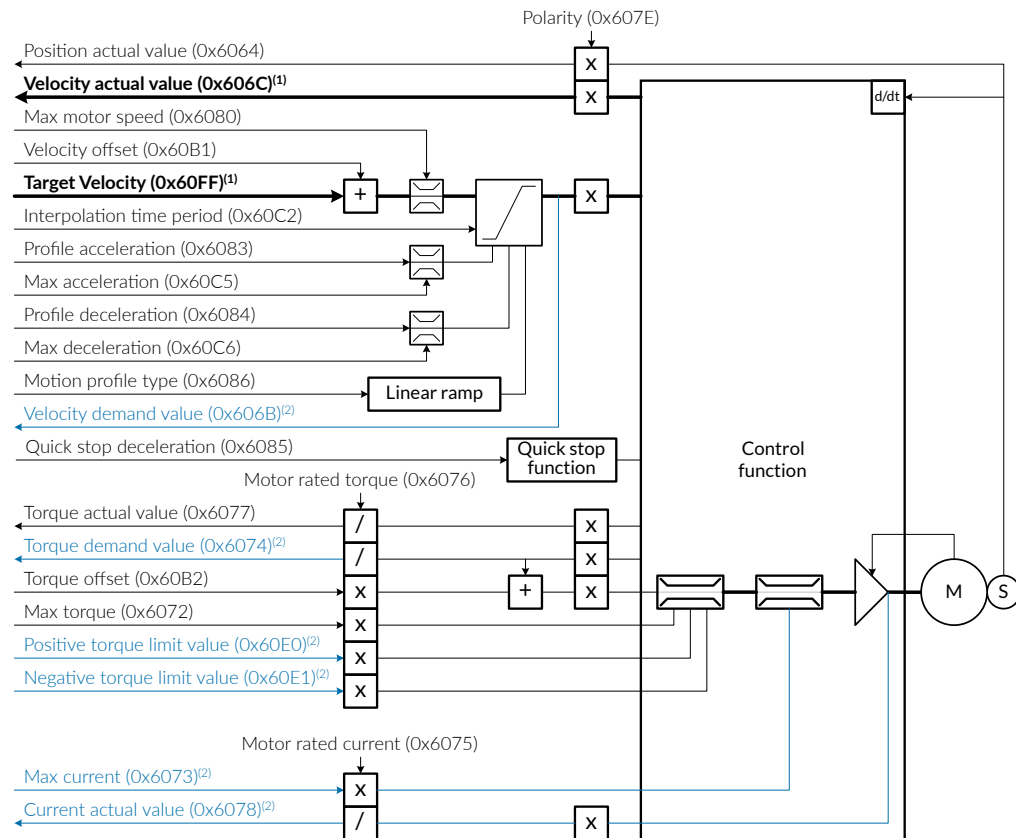


Fig. 2: Block diagram for CSV mode

- (1) The objects marked in bold are required and recommended process data (PDO) for the CSV mode.
- (2) SD4x drives support these objects (marked in blue) additionally to the standard objects in the DS402 drive profile.

The objects **quick stop deceleration** or **profile deceleration** are used when an option code defines a stop via quick stop ramp or deceleration ramp.

The object **torque offset** can be used for the torque feed forward control. In addition, the set torque, current and acceleration limits remain active.

The objects **interpolation time period** and **target velocity** automatically determine the change in speed until the next interpolated position of a trajectory calculation is reached.

CSV-specific bits in the statusword

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

- Bit 12 (drive follows target value) is active as long as the drive follows the trajectory from the control.

2.4.2 Profile Velocity Mode (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:

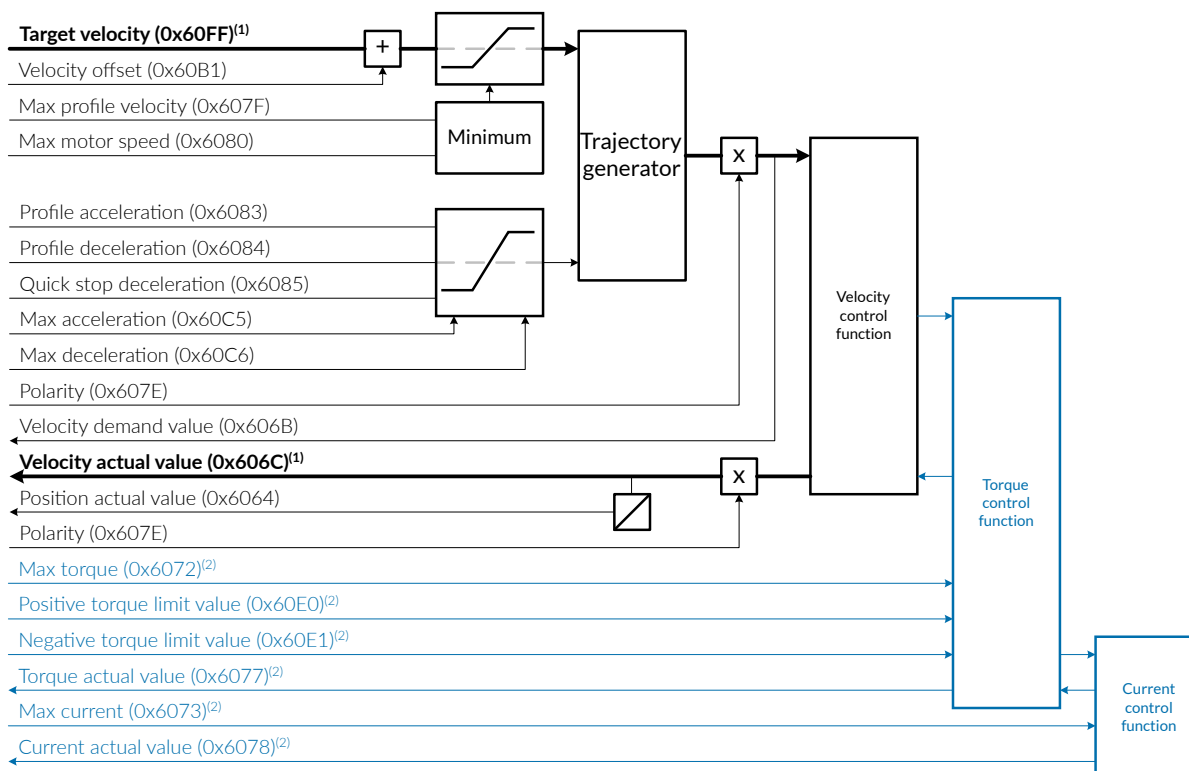


Fig. 3: Block diagram for profile velocity mode (PV)

- (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 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.

2.4.3 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 [Profile Velocity Mode \(PV\)](#), [page 16](#), 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:

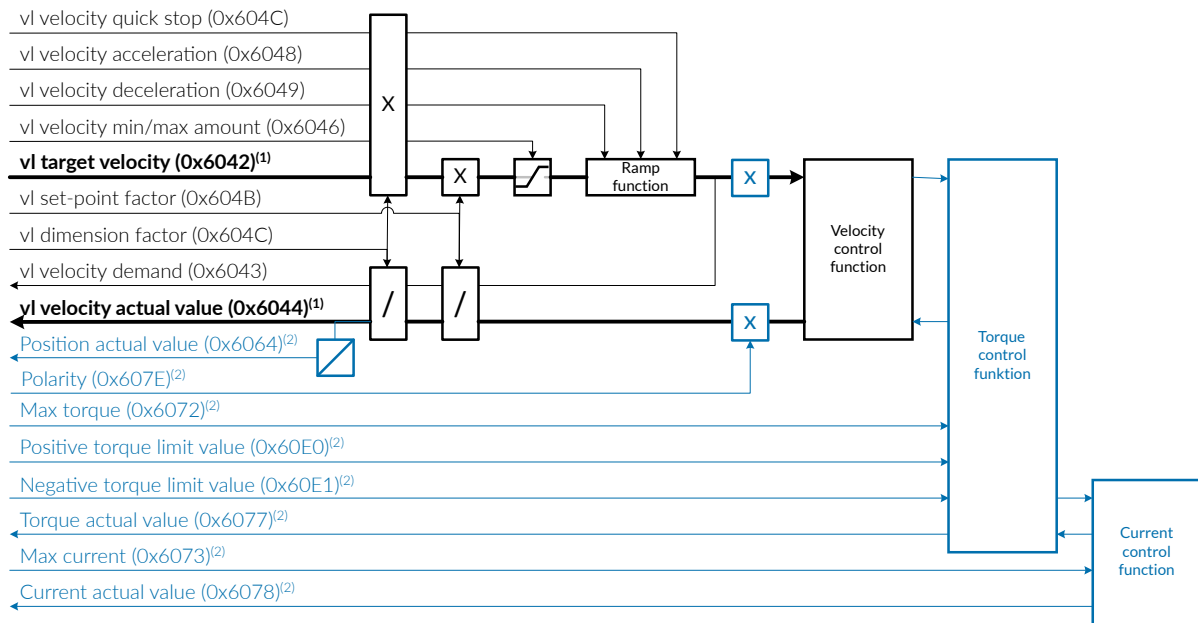


Fig. 4: Block diagram for velocity mode VL

- (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 bit 12 as follows:

- 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]".

2.4.3.1 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 reference value 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.

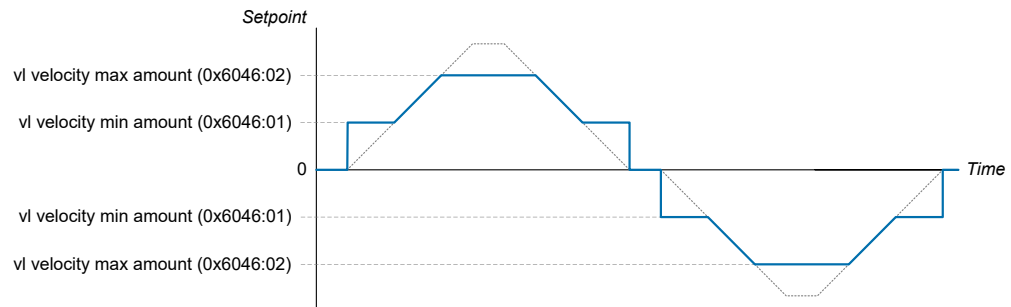


Fig. 5: Speed limitation in velocity mode

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_ACCELERATION_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.

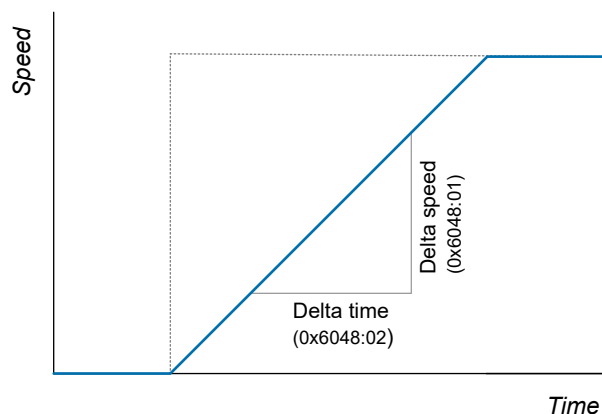


Fig. 6: Limitation of acceleration in velocity mode

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.

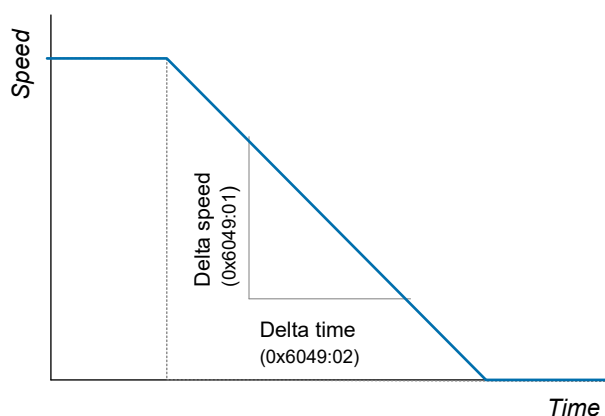


Fig. 7: Limitation of deceleration in velocity mode

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.

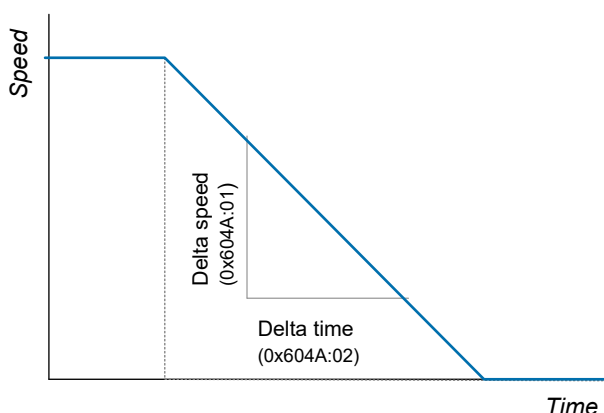


Fig. 8: Limitation of deceleration in the event of a quick stop in velocity mode

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.

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

Application examples

- 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).

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

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.

2.4.4 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:

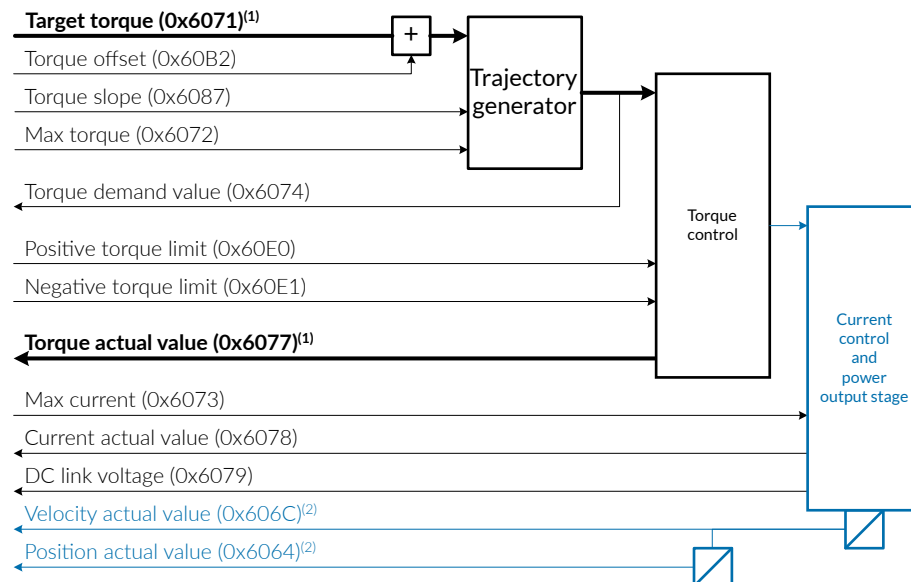


Fig. 9: Block diagram for torque mode

- (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.

2.4.5 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 2.6 "Parameter Set Change", page 36](#)). 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

2.5 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 (vI) uses mostly its own target, actual, scaling and limiting values. These are directly described below the operating mode, see [page 18](#).

2.5.1 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).

Interpolation Time Period (0x60C2)

This object defines the time period between two updates of the target value.

Interpolation time period		Time period for interpolation Sub-IDs: ▶ 0x60C2:00 [U8] = Highest subindex supported (2) ▶ 0x60C2:01 [U8] = Interpolation time period value (1) ▶ 0x60C2:02 [I8] = Interpolation time index (-3)	
DS402_0X60C2_INTERPOLATION_TIME_PERIOD			ID: 0x60C2
RW	Volatile	Unit: –	BYTEARRAY

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

The value for the interpolation time period <subindex1> is indicated in the unit 10 to the power of <subindex2> seconds (default = $1 \times 10^{-3} \text{ s} = 1 \text{ ms}$). If the resulting time T_{itp} is shorter than the bus cycle time, the target value is reached at the time of T_{itp} (if not otherwise limited). Operating modes with target value buffer: after the interpolation time has elapsed, the next value from the buffer is used.

For cyclic operating modes, you should set this object to the current fieldbus cycle time. Otherwise, the interpolation time deviates from the fieldbus cycle time, which causes target value jumps. If interpolation is finished before a new target value is available via fieldbus, the final value of the interpolation is maintained until the next update (see figure to the right).

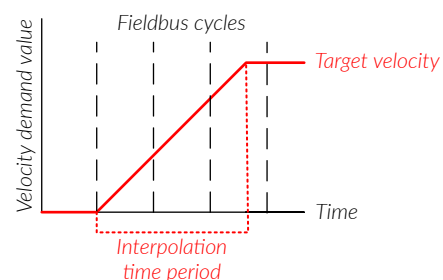


Fig. 10: Interpolation ≠ fieldbus cycle time

The following figures show examples of the interpolation (velocity demand value in blue) of a target value (in red):

Interpolation time period is set correctly:

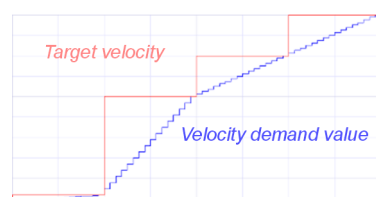


Fig. 11: No target value jumps

Interpolation time period is set incorrectly:

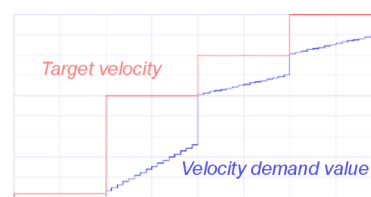


Fig. 12: Target value jumps

Motion Profile Type (0x6086)

This object specifies the shapes of the acceleration and deceleration ramps.

Motion profile type		Motion profile types: 0: Linear ramp 1: Sin ² ramp 2: Jerk-free ramp 3: Jerk-limited ramp	
DS402_0X6086_MOTION_PROFILE_TYPE			ID: 0x6086
RW	Volatile	Unit: –	I16

At present, SD4x drives support the linear ramp shape only. Therefore, this object has the fix value 0.

2.5.2 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 reference value 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		Current torque reference value after ramp and limiting functions The value is indicated in parts per thousand (1000 = 100 %) of the rated torque, object motor rated torque (0x6076).	
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

2.5.3 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

2.5.4 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 [Target and Actual Values in Velocity Mode \(p. 18\)](#)).

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 size 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 standards can be found in the [Appendix \(p. 68\)](#).

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 $10^{-3} \times \text{rpm}$ 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 $10^{-3} \times 1/\text{s}^2$ (*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 $1/\text{s}^2$.

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 $10^{-3} \times 1/\text{s}^3$ (*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 $2 \frac{1}{s^3}$.

2.5.5 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.

2.5.6 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 [Velocity Mode \(VL\) \(p. 16\)](#)) 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**.

2.5.7 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 regenerative 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

2.5.7.7 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/s ² Unit linear motor: mm/s ²	U32

Max Deceleration (0x60C6)

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/s ² Unit linear motor: mm/s ²	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/s ² Unit linear motor: mm/s ²	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/s ² Unit linear motor: mm/s ²	U32

Quick Stop Deceleration (0x6085)

Via this object you can set the deceleration in the event of a quick stop. In the [Option Codes \(p. 12\)](#) 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/s ² Unit linear motor: mm/s ²	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

2.5.8 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 it 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 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-bit 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 according 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.

2.6 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.

Note

If the PDO mapping is set by fieldbus control, this must be repeated after a parameter set change. Usually, the device must complete the fieldbus initialization phase again (e.g. NMT status change from Pre-Operational to Operational).

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 is 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 is 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 is started with parameter set 0. You can switch the parameter set via the fieldbus system only.

Timing diagram

The following diagram shows the timing for a parameter set change. Switching from one parameter set to another is actually done by deactivating and activating the control input “switch on”:

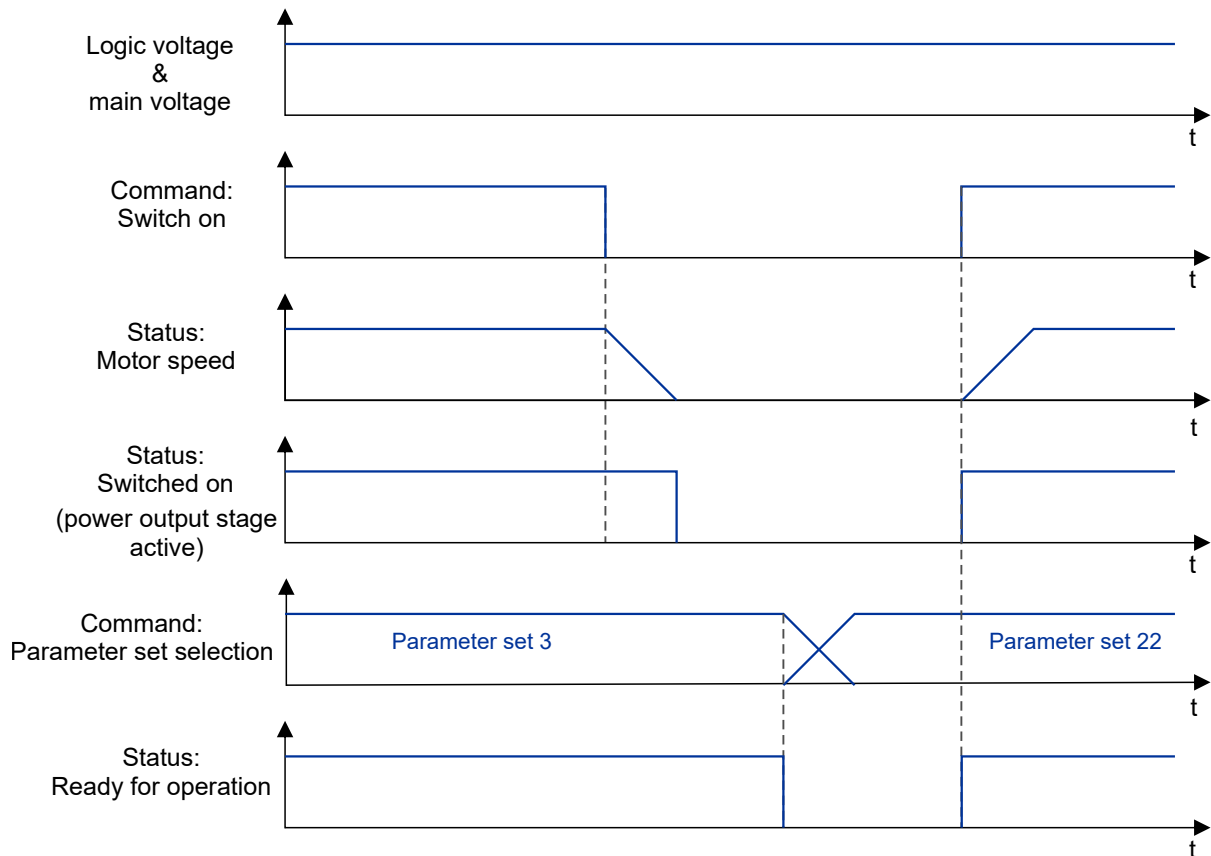


Fig. 13: Timing of parameter set change

The parameter set change via fieldbus is described 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”.

2.6.1 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.

3 Control via EtherCAT

The following chapters describe the addressing of the drives in the EtherCAT network and the connection to the PLC using the Beckhoff software TwinCAT3 as an example.

You must set the parameters for fieldbus communication in the SD4x drive beforehand so that the drive is correctly detected in the EtherCAT network. For further information refer to [chapter 4.1 “Parameterization in drivemaster4”, page 64](#).

3.1 EtherCAT Network

EtherCAT is a high-performance fieldbus especially developed for automation technology. The network is based on the Ethernet protocol and can be connected via standard Cat 5 network cables. However, topology and addressing (OSI layer 3) and telegram transport (OSI layer 4) deviate from standard Ethernet (IP-based or TCP/UDP-based). Therefore, you cannot connect a standard IP-based Ethernet network directly to EtherCAT but must tunnel the communication with Ethernet over EtherCAT (EoE) via the EtherCAT MDevice (e.g. PLC).

EtherCAT is set up in a daisy chain and the addresses are assigned by the MDevice, see [Fig. 14, page 39](#). The daisy chain topology automatically determines the device addresses based on the position of the participant (SubDevice) in the chain. However, the addresses can also be assigned manually by the operator. For more information on addressing refer to [chapter 3.2.3 “Addressing and Identification”, page 50](#).

Only the EtherCAT MDevice can generate telegrams, which are forwarded from one SubDevice to the next in the chain. In this process, each SubDevice reads the target values from the telegram and simultaneously writes its actual values into it. The last participant in the chain sends the telegram back to the MDevice.

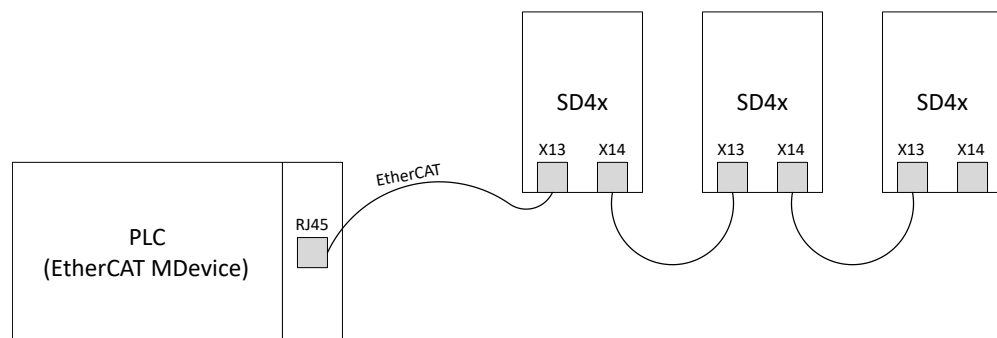


Fig. 14: EtherCAT network (daisy chain)

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

3.2 PLC Connection (TwinCAT3)

The following example shows the connection to the PLC using TwinCAT3 by Beckhoff.

1. Copy the XML file provided by SIEB & MEYER to the TwinCAT3 directory ...\\TwinCAT\\3.1\\Config\\Io\\EtherCAT.

– e.g.: SIEB_MEYER_SD4x.xml

This XML file contains the EtherCAT SubDevice Information (ESI), i.e. it describes the communication parameters and the object dictionary of the drive.

2. Start the TC3 user interface and establish the connection to the target system.
3. Switch the target system into the config-mode and confirm the inquiries “Load I/O Devices?” and “Activate Free Run?” with “Yes”.
4. Right-click on “Devices” and select “Scan” from the context menu.

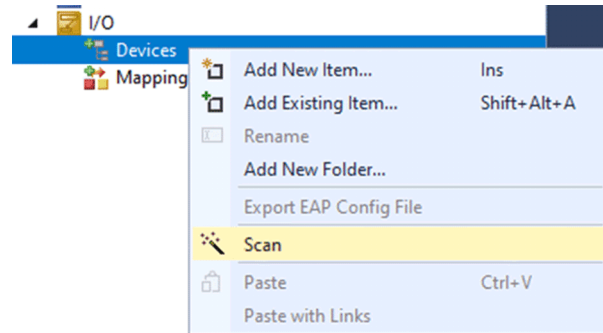


Fig. 15: Execute scan

5. In the following dialog, select the EtherCAT protocol.

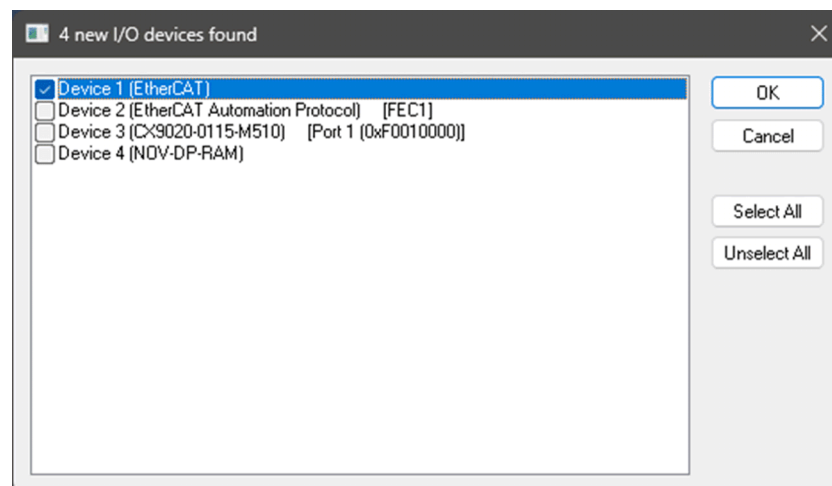


Fig. 16: Select EtherCAT

6. Confirm the inquiry “Scan for boxes” with “Yes”.

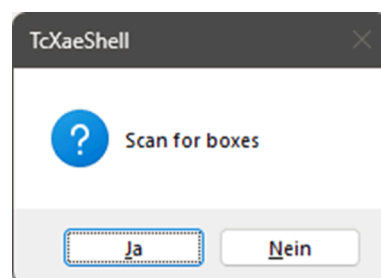


Fig. 17: Scan for boxes

7. Cancel the dialog “EtherCAT drive(s) added”. Otherwise, Beckhoff will automatically create the motion control framework and use it.

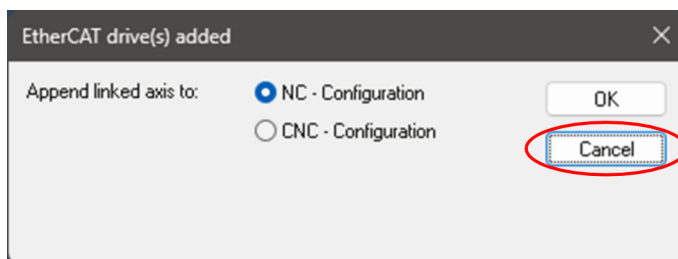


Fig. 18: Cancel EtherCAT drive(s) added

- Now, TwinCAT3 automatically loads all detected EtherCAT devices into the project as shown in the following screenshot:

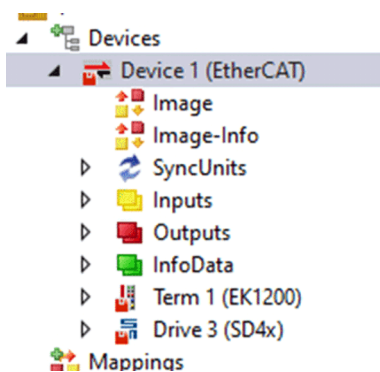


Fig. 19: SD4x in project structure

If the SD4x drive could not or not correctly be detected, the XML file (ESI) is probably not properly installed (see step 1) or TwinCAT3 was not restarted yet.

8. Double-click in the project structure on the SD4x drive to open the settings. Switch to the tab "Process Data":

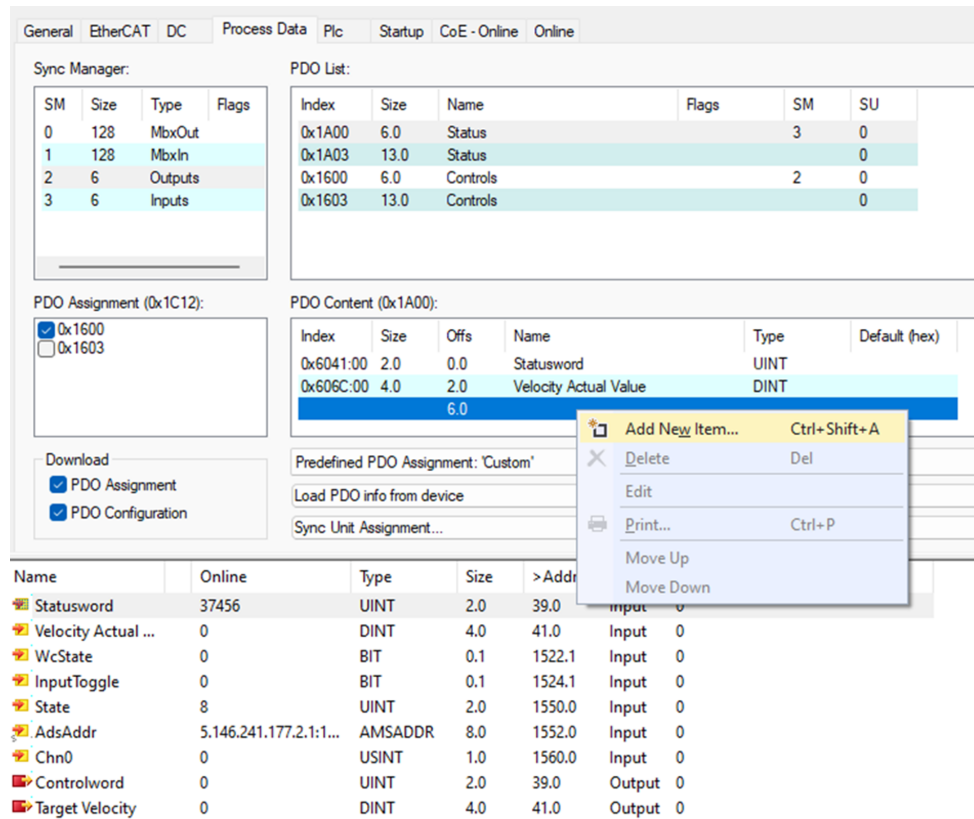


Fig. 20: Process Data

Here you can add or exchange process data.

When you have activated the checkboxes "PDO Assignment" and "PDO Configuration" in the field "Download", this process data mapping is automatically exchanged with the SD4x drive during fieldbus initialization. Only PDOs with a number in the column "SM" (i.e. which are assigned to a SyncManager) are exchanged.

Additionally, you can create your own PDOs: The indexes 0x1A00 to 0x1A03 are supported for TxPDOs (status data) and 0x1600 to 0x1603 for RxPDOs (control data). Via "Predefined PDO Assignment" you can also select ready-made PDO sets.

Note

SD4x drives with a firmware lower than Vx.8.x possibly do not show diagnostic information correctly in the *drivemaster4* software, if other PDO addresses than 0x1A00 and 0x1600 are used.

- In the tab “Startup”, you can add your own parameters to the mapping parameters that will be written during fieldbus initialization. For this purpose, right-click in the list and select “Add New Item” from the context menu. There, you can, for example, preset the object “Mode of operation” as shown in the following figure:

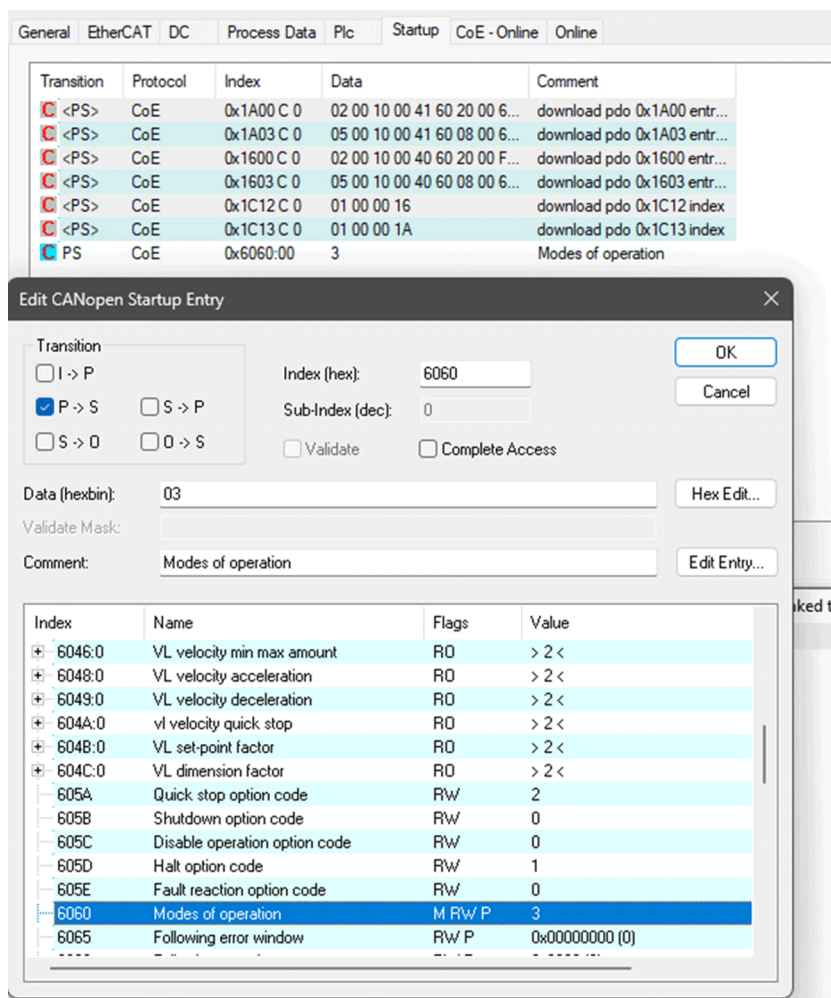


Fig. 21: Set mode of operation

10. If you have already created mappable variables in the PLC program, you can assign these to the process data:

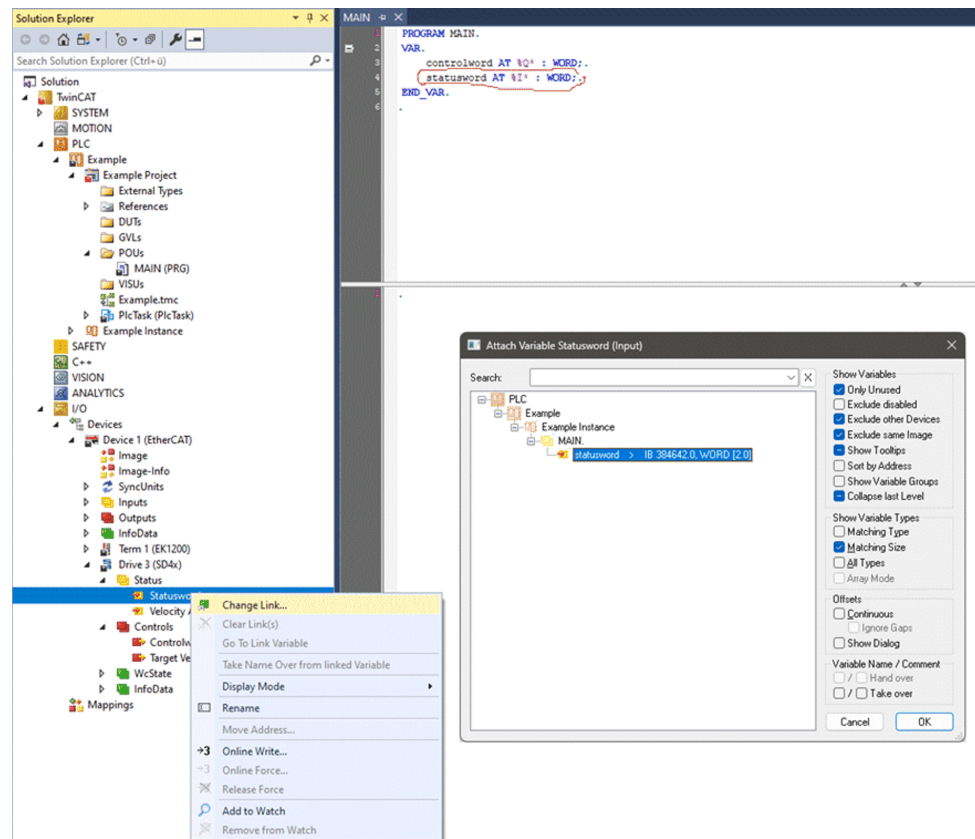


Fig. 22: Linking of variables

11. Load and activate the configuration into the PLC. Confirm the inquiries “Activate Configuration?” and “Restart TwinCAT System in Run Mode?” with “OK”.

→ The configuration and the PLC program are loaded into the PLC and started.

- ✓ Switch to the tab “Online”.

The EtherCAT State Machine switches to the status “Operational” (OP). The linked input and output data is marked and the current values are displayed.

Name	Online	Type	Size	>Addr...	In/Out	User ID	Linked to
Statusword	37456	UINT	2.0	39.0	Input	0	
Velocity Actual ...	0	DINT	4.0	41.0	Input	0	
WcState	0	BIT	0.1	1522.1	Input	0	
InputToggle	0	BIT	0.1	1524.1	Input	0	
State	8	UINT	2.0	1550.0	Input	0	
AdsAddr	5.146.241.177.2.1:1...	AMSADDR	8.0	1552.0	Input	0	
Chn0	0	USINT	1.0	1560.0	Input	0	
Controlword	0	UINT	2.0	39.0	Output	0	
Target Velocity	0	DINT	4.0	41.0	Output	0	

Fig. 23: Online view

3.2.1 EtherCAT State Machine (ESM)

The communication of the drive is controlled by the PLC via the EtherCAT State Machine (ESM). The EtherCAT State Machine coordinates the MDevice and SubDevice applications during boot phase and during operation. Status changes of the SubDevice device are requested by the MDevice.

The EtherCAT state machine can have the following states:

Status	Value
Init	1
Pre-Operational	2
Bootstrap	3
Safe-Operational	4
Operational	8

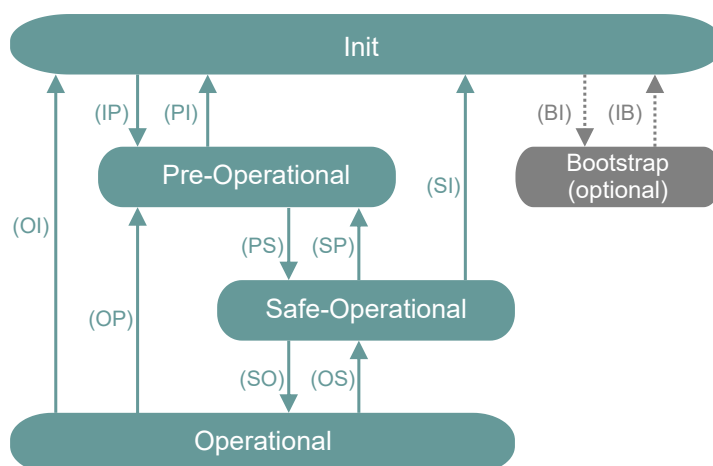


Fig. 24: Status changes of the EtherCAT State Machine

The following status descriptions are taken from the online documentation of the company Beckhoff, see [Beckhoff Information System](https://www.beckhoff.com).

Init	In the Init state, no communication takes place at the application level. The MDevice has access to the registers of the data link layers that inform it.
Pre-Operational (Pre-OP)	In Pre-Operational state, mailbox communication takes place on the application layer, but no process data communication takes place at this stage.
Safe-Operational (Safe-OP)	Process data communication takes place in the Safe-Operational state. Inputs are read, outputs are not yet written.
Operational (OP)	In the Operational state, the outputs have valid values and the inputs continue to be evaluated as in the Safe-Operational state. The receive PDO (RPDO) is sent from MDevice to drive. The transmit PDO (TPDO) is sent from drive to MDevice.
Bootstrap	The Bootstrap state is only reached via the Init state and is only exited again towards the Init state. The Bootstrap state is recommended by the EtherCAT organization for firmware updates via EoE protocol (EoE = Ethernet over EtherCAT).

Note

This update function is not supported by SD4x devices. In the status Bootstrap, no process data communication takes place.

3.2.2 SDO Communication

The SDO communication -Kommunikation runs via the CoE interface (CoE = CANopen over Ethernet) of the EtherCAT MDevice. During the boot phase (startup), the EtherCAT MDevice automatically sends several SDOs. At this point, you can add more SDOs manually (see [chapter 3.2 "PLC Connection \(TwinCAT3\)", page 39](#)).

In addition, TwinCAT3 offers the diagnostics page “CoE - Online”. This page displays the current values of the objects included in the device description file (ESI/XML). In addition, you can change the objects here or add them to the startup SDOs:

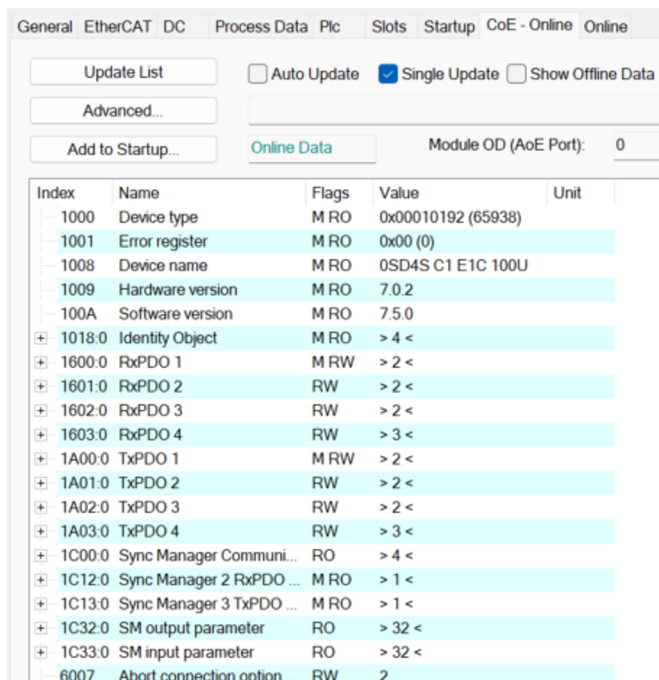


Fig. 25: TwinCAT3 software – CoE - Online

The PLC manufacturers supply ready-made function blocks from the PLC user program. Beckhoff stores these function blocks in the folder “CoE Interface” of the Tc2_EtherCAT library.

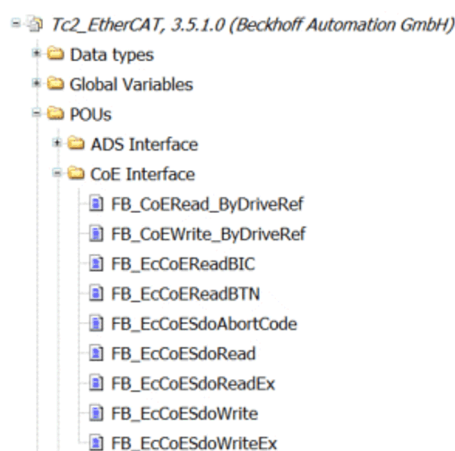


Fig. 26: Tc2_EtherCAT folder

SDO objects are read with FB_EcCoESdoRead and written with FB_EcCoESdoWrite:



Fig. 27: Function block for reading and writing SDOs

For the inputs `sNetId` (`netId`) and `nSlaveAddr` (`port`) you can use the InfoData from the EtherCAT SubDevice. However, the `netId` from InfoData has a different data type than the `netId` inputs of the CoE functions. You can use the function `F_CreateAmsNetId` from the `Tc2_System` library to convert the `netId` to the correct data type.

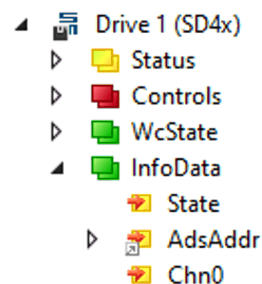


Fig. 28: SubDevice InfoData

The input `tTimeout` is pre-initialized and can remain empty.

`bExecute`, `bBusy` and `bError` are based on the interface of a PLCopen execute function. If an error occurs during processing of the SDO object, the output `nErrId` returns the corresponding error code. This can be interpreted as ADS return code. For this purpose, you can use, for example, `ENUM E_AdsErr` from the `Tc2_System` library or refer to the Beckhoff documentation for help.

The inputs `nIndex`, `nSubIndex`, `cbBufLen` and `pDstBuf` (or `pSrcBuf`) define the SDO object: `nIndex` and `nSubIndex` contain the SDO address, `cbBufLen` contains the object size and `pDstBuf` (or `pSrcBuf`) contain the buffer to be read (or written to). Ideally, the latter points to a memory of the PLC that stores a variable with the correct data type and size. The SDO addresses and the data types are to be found in this document or in the object browser of the *drivemaster4* software:

Name	Object	ID	Sub-ID	Value	Unit	Data type
> Device	DEVICE					
> Device control	DEVICE_CONTROL					
> Communication	COMMUNICATION					
> General	COMMUNICATION_COMMON					
> TCP_IP	TCP_IP					
> DS301	DS301					
> DS402	DS402					
DS402 - Abort connection option code	DS402_0X6007_ABORT_CONNECTION_OPTION_CODE	0x6007	(02)	Disable voltage command		UNIT_I16
DS402 - Error code	DS402_0X603F_ERROR_CODE	0x603f	0			UNIT_U16
DS402 - Controlword	DS402_0X6040_CONTROLWORD	0x6040	0			UNIT_U16
DS402 - Statusword	DS402_0X6041_STATUSWORD	0x6041	36913			UNIT_U16
DS402 - vl target velocity	DS402_0X6042_VL_TARGET_VELOCITY	0x6042	0			UNIT_I16
DS402 - vl velocity demand	DS402_0X6043_VL_VELOCITY_DEMAND	0x6043	0			UNIT_I16
DS402 - vl velocity actual value	DS402_0X6044_VL_VELOCITY_ACTUAL_VALUE	0x6044	0			UNIT_I16
DS402 - vl velocity min max amount	DS402_0X6046_VL_VELOCITY_MIN_MAX_AMOUNT	0x6046	02 00 00 00 00 60 ea 00 ...			UNIT_BYTEARRAY
DS402 - vl velocity acceleration	DS402_0X6048_VL_VELOCITY_ACCELERATION	0x6048	025046000000100			UNIT_DS402_ACCELERATION_REC
DS402 - vl velocity deceleration	DS402_0X6049_VL_VELOCITY_DECELERATION	0x6049	025046000000100			UNIT_DS402_ACCELERATION_REC
DS402 - vl velocity quick stop	DS402_0X604A_VL_VELOCITY_QUICK_STOP	0x604a	025046000000100			UNIT_DS402_ACCELERATION_REC
DS402 - vl set-point factor	DS402_0X604B_VL_SET_POINT_FACTOR	0x604b	02 01 00 01 00			UNIT_BYTEARRAY
DS402 - vl dimension factor	DS402_0X604C_VL_DIMENSION_FACTOR	0x604c	02 01 00 00 01 00 00 ...			UNIT_BYTEARRAY
DS402 - Quick stop option code	DS402_0X605A_QUICK_STOP_OPTION_CODE	0x605a	(02)	Slow down on quick stop ramp and transi...		UNIT_I16
DS402 - Shutdown option code	DS402_0X605B_SHUTDOWN_OPTION_CODE	0x605b	(00)	Disable drive function (switch off the driv...		UNIT_I16
DS402 - Disable operation option code	DS402_0X605C_DISABLE_OPERATION_OPTION_CODE	0x605c	(00)	Disable drive function		UNIT_I16
DS402 - Halt option code	DS402_0X605D_HALT_OPTION_CODE	0x605d	(01)	Slow down on slow down ramp and stay i...		UNIT_I16
DS402 - Fault reaction option code	DS402_0X605E_FAULT_REACTION_OPTION_CODE	0x605e	(01)	Slow down on slow down ramp		UNIT_I16
DS402 - Mode of operation	DS402_0X6060_MODE_OF_OPERATION	0x6060	(03)	Profile velocity mode		UNIT_U8
DS402 - Mode of operation display	DS402_0X6061_MODE_OF_OPERATION_DISPLAY	0x6061	(03)	Profile velocity mode		UNIT_U8
DS402 - Position actual internal value	DS402_0X6063_POSITION_ACTUAL_INTERNAL_VALUE	0x6063	0		counts	UNIT_I32
DS402 - Position actual value	DS402_0X6064_POSITION_ACTUAL_VALUE	0x6064	0			UNIT_I32
DS402 - Following error window	DS402_0X6065_FOLLOWING_ERROR_WINDOW	0x6065	0			UNIT_U32
DS402 - Following error time out	DS402_0X6066_FOLLOWING_ERROR_TIME_OUT	0x6066	0		ms	UNIT_U16
DS402 - Velocity sensor actual value	DS402_0X6069_VELOCITY_SENSOR_ACTUAL_VALUE	0x6069	0		inc/s	UNIT_I32
DS402 - Velocity demand value	DS402_0X606B_VELOCITY_DEMAND_VALUE	0x606b	0			UNIT_I32
DS402 - Velocity actual value	DS402_0X606C_VELOCITY_ACTUAL_VALUE	0x606c	0			UNIT_I32
DS402 - Target torque	DS402_0X6071_TARGET_TORQUE	0x6071	0		%o	UNIT_I16
DS402 - Max torque	DS402_0X6072_MAX_TORQUE	0x6072	10000		%o	UNIT_U16
DS402 - Max current	DS402_0X6073_MAX_CURRENT	0x6073	1188		%o	UNIT_U16
DS402 - Torque demand	DS402_0X6074_TORQUE_DEMAND	0x6074	0		%o	UNIT_U16
DS402 - Motor rated current	DS402_0X6075_MOTOR_RATED_CURRENT	0x6075	10100		mA	UNIT_U32
DS402 - Motor rated torque	DS402_0X6076_MOTOR_RATED_TORQUE	0x6076	1447		mNm	UNIT_U32
DS402 - Torque actual value	DS402_0X6077_TORQUE_ACTUAL_VALUE	0x6077	0		%o	UNIT_I16
DS402 - Current actual value	DS402_0X6078_CURRENT_ACTUAL_VALUE	0x6078	0		%o	UNIT_I16
DS402 - DC link circuit voltage	DS402_0X6079_DC_LINK_CIRCUIT_VOLTAGE	0x6079	325000		mV	UNIT_U32

Fig. 29: drivemaster4 object browser

Example

Note

A TwinCAT3 example project regarding the connection of an SD4x drive via EtherCAT can be found on the SIEB & MEYER website at <https://www.sieb-meyer.com/downloads>.

Task: The object **DS402 - Motor rated torque** is to be read from the SD4x drive via SDO communication.

Procedure:

1. Find the object **DS402 - Motor rated torque** in the object browser of the *drivemas-ter4* software (see [Fig. 29, page 48](#)).
 - a) The displayed ID (0x6076) is the index. The Sub-ID is the subindex. Since the object has no subobjects, the subindex is "0".
 - b) The data type of the object is UNIT_U32, i.e. a 32 bit unsigned integer.
2. In the PLC, create the variable "motorRatedTorque" with the data type UDINT in the VAR section of a block or program.
3. In the input DstBuf, define a pointer to the memory of the variable motorRatedTorque using the function ADR(motorRatedTorque).
4. In the input cbBufLen, you specify the size of the variable in bytes (in this case "4" or read the size with the function SIZEOF(motorRatedTorque)).
5. For the ADS address, create a variable of type AMSADDR and map this variable to InfoData.AdsAddr of the SD4x EtherCAT SubDevice.
Then, the EtherCAT MDevice automatically saves current ADS address of the SD4x drive there.
6. For diagnosis, create a variable adsReturnCode of type Tc2_System.E_AdsErr.

In the PLC application program, the definition might look something like this:

```
VAR.
  adsAddress AT %I* : AMSADDR; // EtherCAT -> InfoData -> AdsAddr.
  sdoReader : Tc2_EtherCAT.FB_EcCoESdoRead;
  motorRatedTorque : UDINT;
  adsReturnCode : Tc2_System.E_AdsErr;
END_VAR.

sdoReader (
  sNetId:= Tc2_System.F_CreateAmsNetId(adsAddress.netId), .
  nSlaveAddr:= adsAddress.port, .
  nSubIndex:= 0, .
  nIndex:= 16#6076, .
  pDstBuf:= ADR(motorRatedTorque), .
  cbBufLen:= SIZEOF(motorRatedTorque), .
  bExecute:= , .
  tTimeout:= , .
  bBusy=> , .
  bError=> , .
  nErrId=> adsReturnCode, .
  cbRead=> );
```

Fig. 30: Read motor rated torque

Result: The function block must be called cyclically. As soon as a rising edge occurs at the input bExecute, the communication starts once. If bBusy is "FALSE", the variable motorRatedTorque contains the read value. If bError is active, adsReturnCode contains an error code.

After a successful communication cycle, online monitoring looks something like this:

Expression	Type	Value	Prepared va...	Address	Comment
* adsAddress	AMSADDR	(netId:=[5.90.167.232.2.1], port...		%I*	EtherCAT -> InfoData -> Ad...
= sdoReader	Tc2_EtherCAT.FB_EcCoES...				
sNetId	T_AmsNetId	'5.90.167.232.2.1'			AmsNetId of the EtherCAT ...
nSlaveAddr	UINT	1002			Address of the slave device.
nSubIndex	BYTE	0			CANopen Sdo subindex.
nIndex	WORD	24694			CANopen Sdo index.
pDstBuf	PVOID	3645393508			Contains the address of the ...
cbBufLen	UDINT	4			Contains the max. number of...
bExecute	BOOL	TRUE			Function block execution is t...
tTimeout	TIME	T#5s			States the time before the fu...
bBusy	BOOL	FALSE			
bError	BOOL	FALSE			
nErrId	UDINT	0			
cbRead	UDINT	4			
nOffset	DWORD	1618345984			
* fbAdsReadEx	ADSREDEX				
* RisingEdge	R_TRIG				
bSet0	BOOL	FALSE			
motorRatedTorque	UDINT	1447			
adsReturnCode	E_ADSERR	NOERR			

```

1 sdoReader(
2   sNetId := '5.90.167.2' := Tc2_System.F_CreateAmsNetId(adsAddress.netId '5.90.167.232.2.1'),
3   nSlaveAddr := 1002 := adsAddress.port 1002,
4   nSubIndex := 0,
5   nIndex := 24694 := 16#6076,
6   pDstBuf := 3645393508 := ADR(motorRatedTorque 1447),
7   cbBufLen := 4 := sizeof(motorRatedTorque 1447),
8   bExecute := ,
9   tTimeout := ,
10  bBusy := ,
11  bError := ,
12  nErrId := 0 => adsReturnCode NOERR,
13  cbRead := );

```

Fig. 31: Return for SDO motorRatedTorque

Thus, the parameterized motor in this example has a rated torque of 1447 mNm.

3.2.3 Addressing and Identification

EtherCAT networks provide different methods for addressing and identification of Sub-Devices. In TwinCAT3, switch to the tab “EtherCAT”. There, the following identification methods for SD4x drives are available:

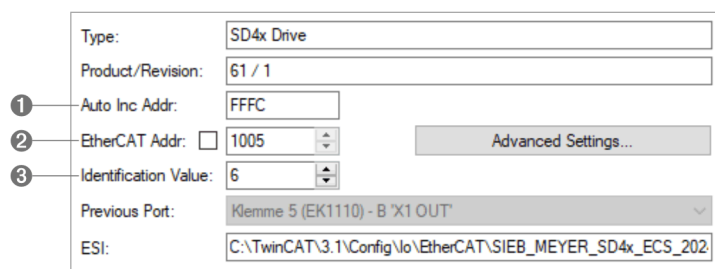


Fig. 32: TwinCAT3 software – EtherCAT addressing

- [1] Auto increment address
- [2] EtherCAT address
- [3] Identification via Hot Connect

Note

The following identification methods are **not** supported by the drive controllers of series SD4x:

- ▶ Data Word
- ▶ Fast Hot Connect

3.2.3.1 Auto Increment Address

The auto increment address corresponds to a topological address of the EtherCAT SubDevice. That means, the SubDevice receives an address according to its physical position in the network.

The EtherCAT MDevice assigns the auto increment address during initialisization of the bus communication. The first SubDevice in the network receives the auto increment address 0 (0x0000). For each subsequent SubDevice the value is decremented, i.e. the second SubDevice has the address -1 (0xFFFF), the third SubDevice has the address -2 (0xFFFE) etc.

3.2.3.2 EtherCAT Address

The EtherCAT address corresponds to the fieldbus address of the SubDevices. You can set a fixed address or the MDevice automatically assigns an address during initialization of the bus communication.

The following figure shows the EtherCAT addresses after automatic assignement by the MDevice. The first SubDevice receives the address 1001. For each subsequent SubDevice the value is incremetned by 1:

Number	Box Name	Address
 1	Klemme 6 (EK1200)	
 2	Klemme 2 (EL1859)	1001
 3	Klemme 8 (EL6731)	1002
 4	Klemme 9 (EL6631)	1003
 5	Klemme 5 (EK1110)	1004
 6	Box 11 (SD4x Drive)	1005

Fig. 33: Automatic assignement of EtherCAT addresses

In this case, the EtherCAT address can change when a new SubDevice device is added in the EtherCAT network.

If you want to have the address independent of the topological position in the network, you must activate the checkbox in front of EtherCAT address:



Fig. 34: Set fixed EtherCAT address

This way, the SubDevice always receives the same address. For subsequent SubDevices, the address is incremented, if you do not set a fixed address for them:

Number	Box Name	Address
1	Term 4 (EK1200)	
2	Term 2 (EK1110)	1001
3	Box 3 (SD4x Drive)	1003
4	Box 4 (SD4x Drive)	1004

Fig. 35: Fix assignment of EtherCAT addresses

Saving the EtherCAT Address in the SubDevice

You can save a fix EtherCAT address of a SubDevice in the EEPROM. For this purpose, a connection to the SubDevice must already be established. Proceed as described in the following:

1. In the EtherCAT tab, click the button “Advanced Settings” and switch to the page “Configured Station Alias” in the dialog window.

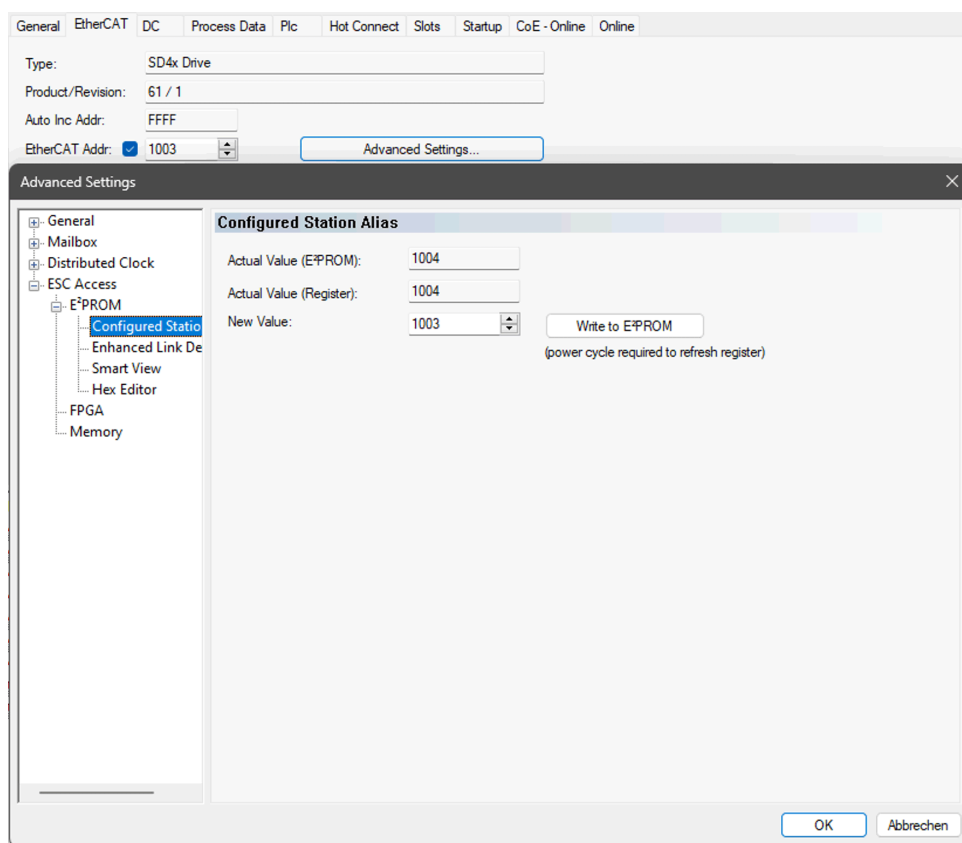


Fig. 36: TwinCAT3 software – advanced settings for EtherCAT

2. Enter the SubDevice address in “New Value” and click the button “Write to EEPROM”.
 3. Click “OK” to confirm the input.
- ✓ The EtherCAT SubDevice will write the entered value to its ESC register 0x0012 with the next restart.

3.2.3.3 Hot Connect

Hot Connect allows assigning EtherCAT SubDevice independent of the topology using fix predefined addresses. Before you can use Hot Connect you must add the SubDevice to a Hot Connect group:

- ⇒ Right-click on the SubDevice and select “Add Hot Connect Group” in the context menu.

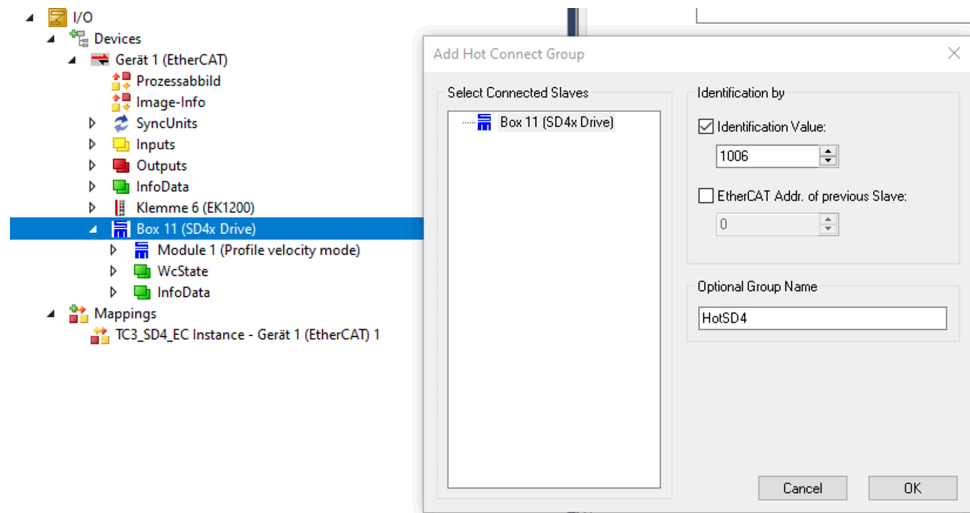


Fig. 37: TwinCAT3 software – Add Hot Connect Group

- a. Enter the address of the Hot Connect group in the “Identification Value”.
 - b. If applicable, select a group name and confirm with “OK”.
- ✓ The device is added to the Hot Connect group. Afterwards, the tab “Hot Connect” is available for the SubDevice.

In order to identify the SubDevices independent of their topological positions, you must assign fixed addresses to them. For this purpose, the following identification methods are available:

- Configured Station Alias
- Explicit Device ID

After the identification method is set, the MDevice only includes the SubDevice in the cyclical communication when it has a valid identification value. If there are identical devices, this ensures that the correct configuration for the respective device is selected. When a device is replaced, the identification value must be set in the replacement device.

Configured Station Alias (ADO 0x0012):

This address is also called “Second Station Address” (SSA).

1. At first, save the desired identification value in the EEPROM of the SubDevice as described in [chapter “Saving the EtherCAT Address in the SubDevice”, page 52](#).

2. Then, switch to the tab “Hot Connect” and click the button “Configure”.

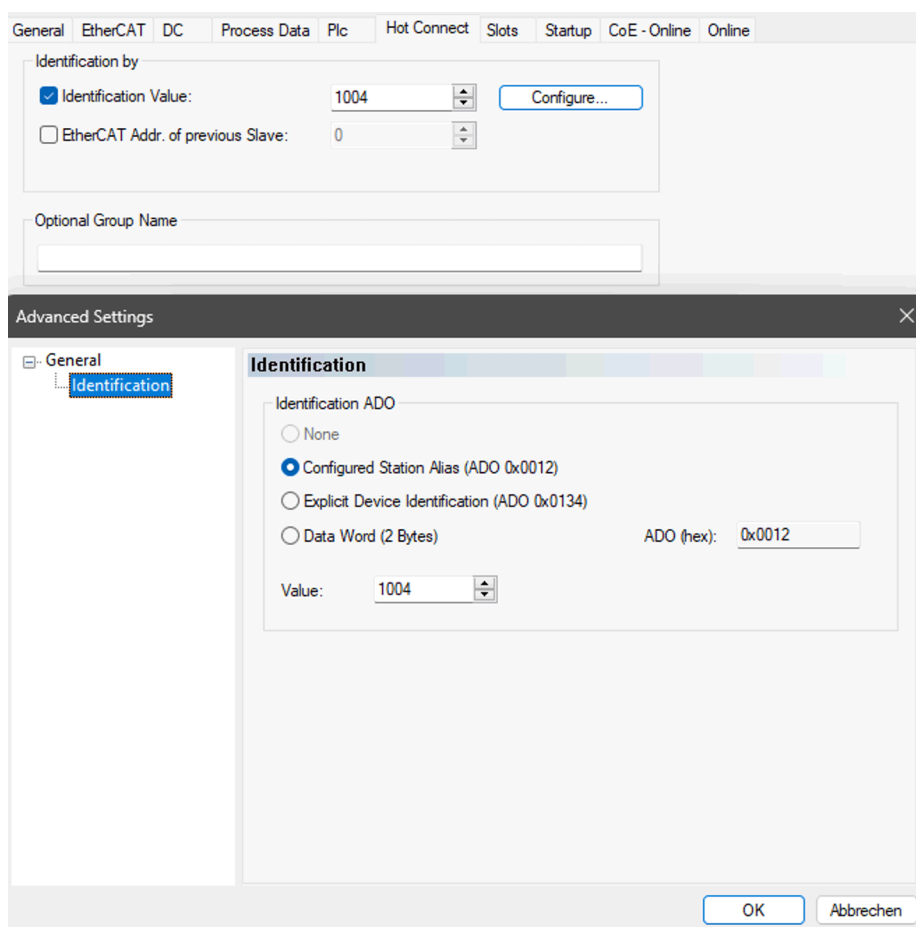


Fig. 38: TwinCAT3 software – Configured Station Alias

- a. Under "Identification", select the option “Configured Station-Alias (ADO 0x0012)” and enter the identification value of the SubDevice.
 - b. Click OK to confirm your input.
- ✓ The SubDevice has now a valid identification value and will be included in the cyclic communication by the EtherCAT MDevice.

Explicit Device Identification (ADO 0x0134)

For this method, the identification value in the SubDevice must match the value in the EtherCAT MDevice.

Note

SD4x drives support this function with parameter set version 1.1.12 and higher (object ID 0x0029).

1. At first, set the identification in the SD4x drive. For this purpose, open the *drivemaster4* software and switch to the parameter page “Fieldbus”.

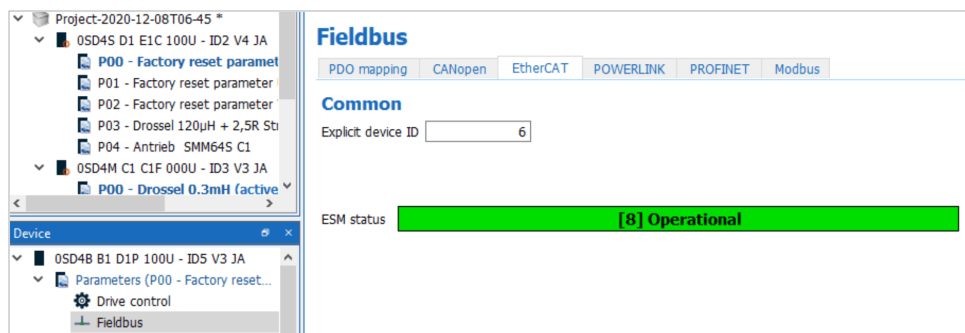


Fig. 39: drivemaster4 software – EtherCAT parameterization

- a. Enter the desired identification value in the field “Explicit device ID”.
The value 0 (default) deactivates that method in the drive. The values 1 to 65535 activate that method in the drive.
 - b. Write the parameters into the drive.
2. Now, set the identification in the EtherCAT MDevice. Switch to the tab “Hot Connect” and click the button “Configure”.

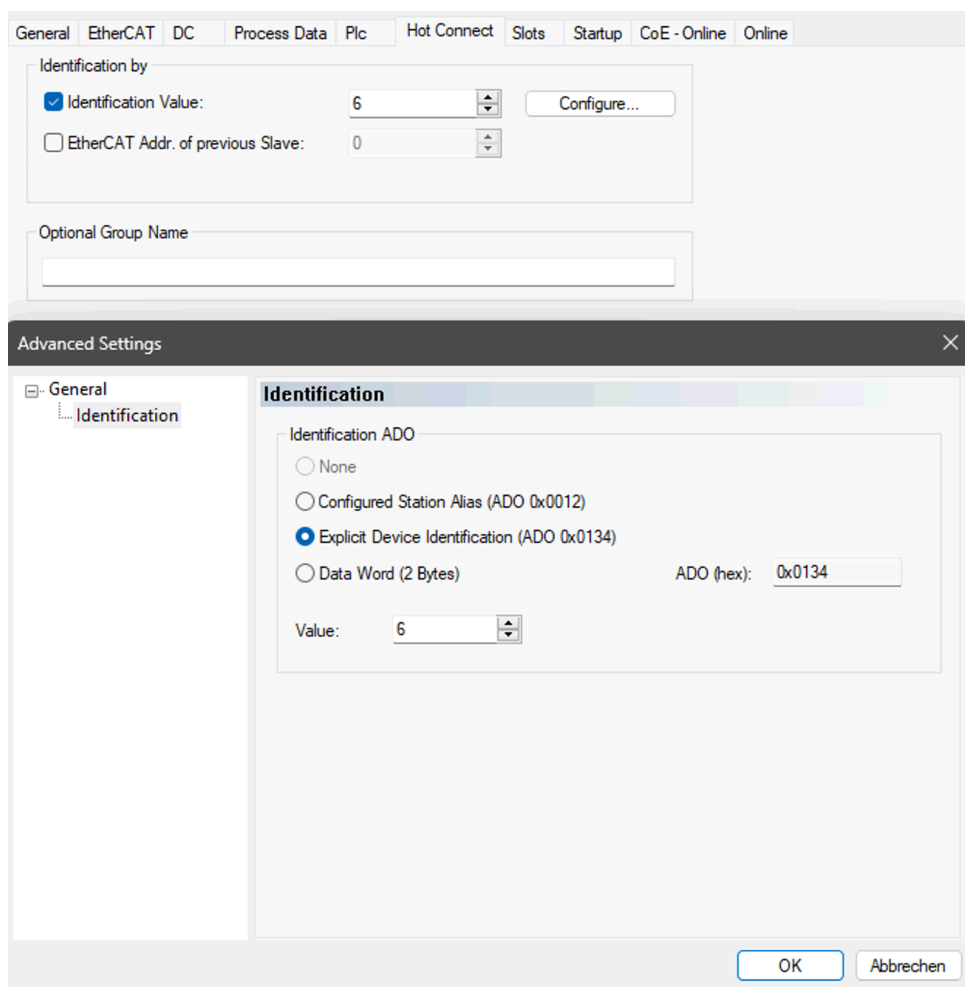


Fig. 40: TwinCAT3 software – Explicit Device Identification

- a. In "Identification", select the option “Explicit Device Identification (ADO 0x0134)”.

- b. In the field “Value” enter the identification of the SubDevice and confirm with “OK”.
3. Switch to the tab “EtherCAT” and click “Advanced Settings”. In the dialog, switch to the page “Behavior” and activate the checkbox “Check Identification”.

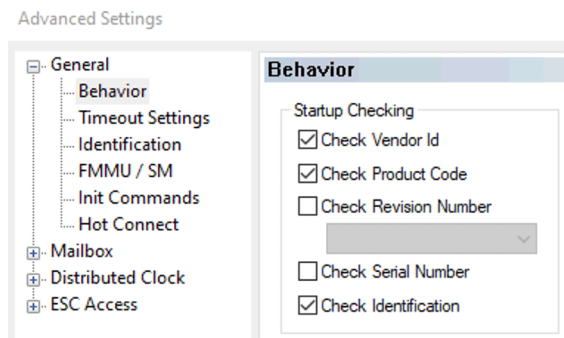


Fig. 41: TwinCAT3 software – set behavior

- ✓ During startup, the EtherCAT MDevice now checks that the identification value set here matches the identification value in the EtherCAT SubDevice. If so, the MDevice includes the SubDevice in the cyclic communication.

3.2.4 General Objects

The following general objects from the DS301 drive profile and from the EtherCAT specification are used for addressing and PDO mapping.

Object 0x1000 – Device Type

This object returns the type and function of the device.

Note

Via the objects 0x1000 (device type) and 0x6502 (supported drive modes) the PLC can determine the device type and the available operating modes. Via the object 0x6060 (mode of operation) the PLC can switch to the desired operating mode, see [chapter 2.4.5 “Change Operating Mode”, page 23](#).

Index	0x1000
Name	Device type
Object code	VAR
Data type	Unsigned 32

Access	RO
PDO mapping	–
Unit	–
Value range	0 ... (2 ³² –1)
Default value	0x00010192

Bit number								
31	...	24	23	...	16	15	...	0
Additional Information						Device profile number		
Mode bits			Type					
Device profile number:		0x0192		device profile DS402				
Additional information:		0x01		Frequency converter				

Object 0x1600 to 0x1603 – RPDO Mapping Parameter

These objects contain the mapping parameters for RPDO 0 to RPDO 3 of the device.

All values must be set via the bus system.

Note

If a PDO mapping for RPDO 0 is entered in the *drivemaster4* software, the PLC will overwrite this mapping.

The number of parameters is set in subindex 0. The subindexes 1 to 8 contain the according mapping parameters.

Index	0x1600 to 0x1603
Name	RPDO mapping parameter
Object code	RECORD
Data type	PDO mapping parameter record

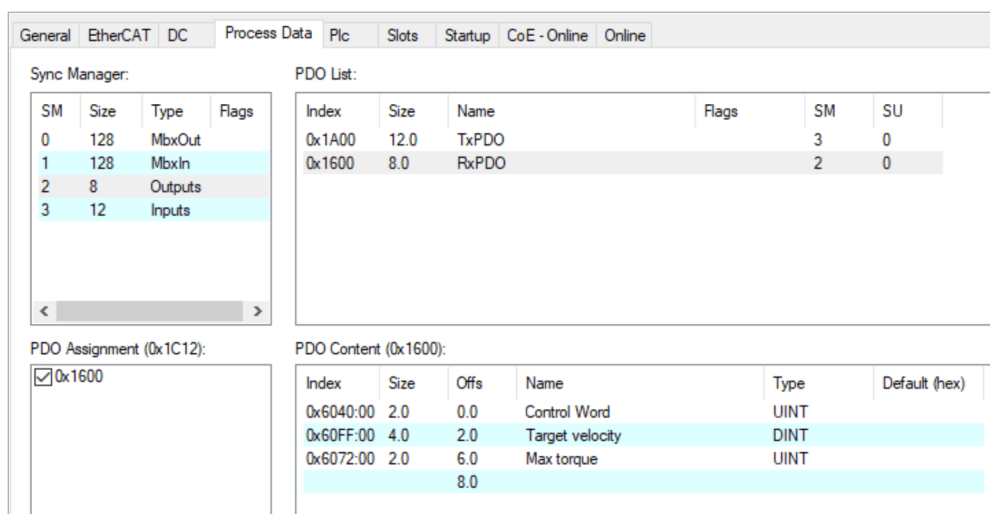
Subindex	0
Name	Number of mapped objects
Data type	Unsigned 8
Access	RW
PDO mapping	–
Unit	–
Value range	0 ... 8
Default value (profile velocity mode)	0x1600 (RPDO 0): 3 0x1601 (RPDO 1): 0 0x1602 (RPDO 2): 0 0x1603 (RPDO 3): 0
Description	Number of mapped objects

Subindex	1 to 8
Name	RPDO mapping parameter x
Data type	Unsigned 32
Access	RW
PDO mapping	–
Unit	–
Value range	See value definition below
Default value (profile velocity mode)	0x1600 (RPDO 0): 0x60400010, 0x60420010, 0x60720010 0x1601 (RPDO 1): 0, 0, 0, 0, 0, 0, 0, 0 0x1602 (RPDO 2): 0, 0, 0, 0, 0, 0, 0, 0 0x1603 (RPDO 3): 0, 0, 0, 0, 0, 0, 0, 0
Description	Description of the PDO (by index, subindex and length)

Value definition of mapping parameter:

Bit number							
31	...	16	15	...	8	7	0
Object index		Object subindex			Length in bits (8, 16, 32)		

The following figure shows the RPDO 0 mapping (object 0x1600) for the profile velocity mode in the TwinCAT3 software:



The screenshot shows the TwinCAT3 software interface with the 'Process Data' tab selected. The 'PDO List' table shows the mapping for RPDO 0 (Index 0x1600, Size 8.0, Name RxPDO, SM 2, SU 0). The 'PDO Assignment (0x1C12)' section shows the checkbox for 0x1600 is checked. The 'PDO Content (0x1600)' table shows the mapping for the RPDO 0 content, including Control Word (Index 0x6040:00, Size 2.0, Offs 0.0, Type UINT), Target velocity (Index 0x60FF:00, Size 4.0, Offs 2.0, Type DINT), and Max torque (Index 0x6072:00, Size 2.0, Offs 6.0, Type UINT).

SM	Size	Type	Flags
0	128	MbxOut	
1	128	MbxIn	
2	8	Outputs	
3	12	Inputs	

Index	Size	Name	Flags	SM	SU
0x1A00	12.0	TxPDO		3	0
0x1600	8.0	RxPDO		2	0

Index	Size	Offs	Name	Type	Default (hex)
0x6040:00	2.0	0.0	Control Word	UINT	
0x60FF:00	4.0	2.0	Target velocity	DINT	
0x6072:00	2.0	6.0	Max torque	UINT	

Fig. 42: TwinCAT3 software – RPDO 0 mapping

Object 1A00h to 1A03h – TPDO Mapping Parameter

These objects contain the mapping parameters for TPDO 0 to TPDO 3 of the device.

All values must be set via the bus system.

Note

If a PDO mapping for TPDO 0 is entered in the *drivemaster4* software, the PLC will overwrite this mapping.

The number of parameters is set in subindex 0. The subindexes 1 to 8 contain the according mapping parameters.

Index	1A00h to 1A03h
Name	TPDO mapping parameter
Object code	RECORD
Data type	PDO mapping parameter record

Subindex	0
Name	Number of mapped objects
Data type	Unsigned 8
Access	RW
PDO mapping	–
Unit	–
Value range	0 ... 8
Default value (profile velocity mode)	1A00h (RPDO 0): 3 1A01h (RPDO 1): 0 1A02h (RPDO 2): 0 1A03h (RPDO 3): 0
Description	Number of mapped objects

Subindex	1 to 8
Name	TPDO mapping parameter x
Data type	Unsigned 8
Access	RW
PDO mapping	–

Unit	–
Value range	See value definition below
Default value (profile velocity mode)	1A00h (RPDO 0): 60400010h, 60420010h, 60720010h 1A01h (RPDO 1): 0, 0, 0, 0, 0, 0, 0, 0, 0 1A02h (RPDO 2): 0, 0, 0, 0, 0, 0, 0, 0, 0 1A03h (RPDO 3): 0, 0, 0, 0, 0, 0, 0, 0, 0
Description	Description of the PDO (by index, subindex and length)

Value definition of mapping parameter:

Bit number						
31	...	16	15	...	8	7
Object index			Object subindex		Length in bits (8, 16, 32)	

The following figure shows the TPDO -0 mapping (object 0x1A00) for the profile velocity mode in the TwinCAT3 software:

The screenshot shows the TwinCAT3 software interface with the 'Process Data' tab selected. The 'PDO List' table shows two entries: 0x1A00 (TxPDO) and 0x1600 (RxPDO). The 'PDO Assignment (0x1C13)' section shows 0x1A00 selected. The 'PDO Content (0x1A00)' table shows four entries: 0x6041:00 (Status Word, UINT), 0x606C:00 (Velocity actual value, DINT), 0x6064:00 (Position actual value, DINT), and 0x603F:00 (Error code, UINT).

Fig. 43: TwinCAT3 software – TPDO 0 mapping

Object 0x1C00 – Sync Manager Communication Type

Index	0x1C00
Name	Sync Manager Communication Type
Object code	REC
Data type	See subindex descriptions

Subindex	0
Name	Subindex 0
Access	RO
PDO mapping	Not possible
Data type	Unsigned 8
Unit	–
Value range	4
Default value	4
Internal object number	–
Description	Highest subindex supported

Subindex	1 – 4
Name	PDO mapping
Access	RO
PDO mapping	Not possible

Data type	Unsigned 8
Unit	–
Value range	1 ... 4
Default value	See below
Internal object number	–
Description	Type of sync manager

Set values for PDO mapping objects

Object	Sync Manager Type (object 0x1C00)	
Set value subindex 0	4	Highest subindex supported
Set value subindex 1	1	Channel 1: mailbox write
Set value subindex 2	2	Channel 2: mailbox read
Set value subindex 3	3	Channel 3: process data write
Set value subindex 4	4	Channel 4: process data read

Object 0x1C12 – RPDO Assignment

This object describes the PDOs that are transmitted as EtherCAT input data.

Index	0x1C12
Name	RPDO assignment
Object code	RECORD
Data type	See subindex descriptions

Subindex	0
Name	Subindex 0
Access	RO
PDO mapping	Not possible
Data type	Unsigned 8
Unit	–
Value range	1 ... 4
Default value	1
Internal object number	–
Description	Highest subindex supported

Subindex	1 to 4
Name	Subindex x
Access	RW
PDO mapping	Not possible
Data type	Unsigned 16
Unit	–
Value range	0x1600 ... 0x1603
Default value	0x1600
Internal object number	–
Description	PDO mapping for the nth mapping object

Object 0x1C13 – TPDO Assignment

This object describes the PDOs that are transmitted as EtherCAT output data.

Index	0x1C13
Name	TPDO assignment
Object code	RECORD
Data type	See subindex descriptions

Subindex	0
Name	Subindex 0
Access	RO
PDO mapping	Not possible
Data type	Unsigned 8
Unit	–
Value range	1 ... 4
Default value	1
Internal object number	–
Description	Highest subindex supported

Subindex	1 to 4
Name	Subindex x
Access	RW
PDO mapping	Not possible
Data type	Unsigned 16
Unit	–
Value range	0x1A00 ... 0x1A03
Default value	0x1A00
Internal object number	–
Description	PDO mapping for the nth mapping object

3.3 Supported Synchronization Types

The SD4x drive supports the following synchronization types:

- ▶ Free Run: the drive is synchronized to an internal clock and therefore not synchronous with EtherCAT.
- ▶ DC synchronous (DC = Distributed Clocks): Based on the DC unit, the drive is synchronized to the Sync0 event. This signal is accurate to a few nanoseconds.

The synchronization type “SM synchronous” (synchronous to SyncManager event) is not supported due to the high real-time requirements for the motor control. With this synchronization type, the drive is synchronized to the received EtherCAT frame. However, the synchronization signal is only accurate in the microsecond range (jitters).

You can find the synchronization types provided by the drive in the SyncManager objects 0x1C32 and 0x1C33, in each subobject “Synchronization types supported”.

The following figure presents a comparison of the synchronization types:

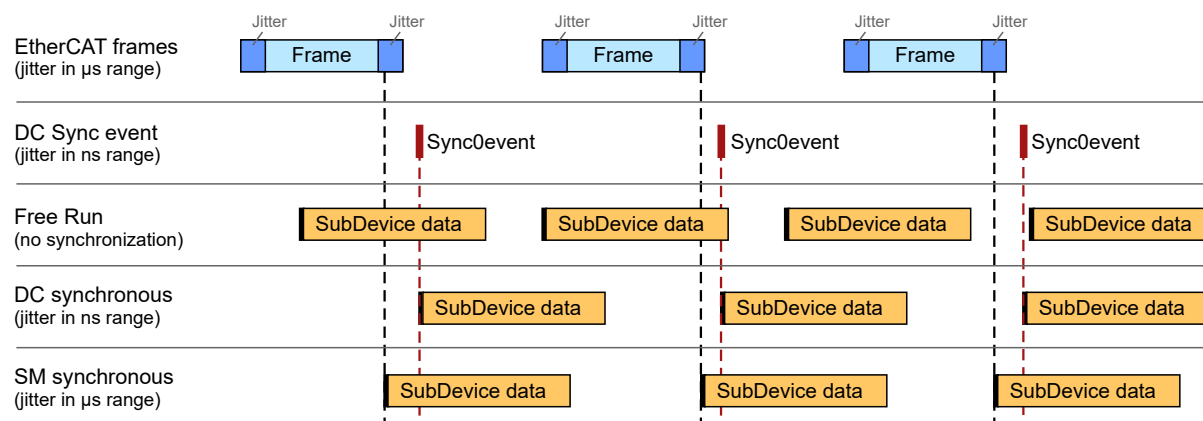


Fig. 44: EtherCAT synchronization types

3.3.1 Free Run

The drive is synchronized to an internal clock. When the drive receives an EtherCAT frame, it saves the EtherCAT process data temporarily. With the next internal interval, the drive applies and processes this data.

Target value processing in the drive has a fix interval (default: 16 kHz) and therefore cannot be set.

The SyncManager objects 0x1C32 and 0x1C33 contain the currently active synchronization type, in each subobject "Synchronization type". In addition, the subobject "Cycle time" specifies the current (internal) interval as cycle time in nanoseconds.

3.3.2 DC Synchronous (Distributed Clocks)

Drive and EtherCAT MDevice are synchronized to the Distributed Clocks (DC). Therefore, their cycles run synchronously. Target value processing in the drive keeps its own interval (default: 16 kHz). However, the target values are only applied with the next Sync0 interval. This way, MDevice and SubDevice process the target values at the same time (deviations only in the nanosecond range). Moreover, processing is synchronized with all other EtherCAT participants that are DC synchronous.

In order to synchronize the drive to the Sync0 interval, the Sync0 cycle time must be a multiple of the drive's minimum cycle time for target value processing: $\text{Sync0 cycle time} = 2^n \times \text{minimum cycle time}$.

At present, the following Sync0 intervals are available for DC synchronization:

Sync0 interval	Resulting Sync0 cycle time
4 kHz	250 μs
2 kHz	500 μs
1 kHz	1 ms
500 Hz	2 ms
250 Hz	4 ms
125 Hz	8 ms

The minimum cycle time can be read from the SyncManager objects 0x1C32 and 0x1C33, from each subobject "Minimum cycle time". In the states Pre-Operational and Operational of the EtherCAT State Machine (ESM), the drive is ready for synchronization. If Sync0 signals are detected with the right interval, the drive is Sync0 synchronous. If not, the drive remains in Free Run mode.

The SyncManager objects 0x1C32 and 0x1C33 contain the currently active synchronization type, in each subobject "Synchronization type". In addition, you can switch to the desired synchronization type via this subobject. If the requested type is possible, this synchronization type will be applied. The new synchronization type remains active until another type is requested or until the next automatic switch (status of ESM switches).

⚠ WARNING



Switching the synchronization type during operation

The drive requires a certain time (up to several seconds) to synchronize with a new Sync0 interval. Therefore, switching the Sync0 interval or the synchronization type during operation can cause serious errors in the drive control. Under certain circumstances, personal injury and property damage may result.

→ Disable operation before you switch the Sync0 interval or the synchronization type.

If the drive cannot be synchronized to the Sync0 signal, the subobject "Sync Error" becomes active in the SyncManager objects. You can map this subobject as PDO.

After successful synchronization, the Sync0 interval is specified as cycle time in the SyncManager objects, in each subobject "Cycle time" in nanoseconds.

4 Drive Parameterization and Diagnostics

The following chapters describe the parameterization and diagnostics of an SD4x drive with the software *drivemaster4*.

4.1 Parameterization in *drivemaster4*

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

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

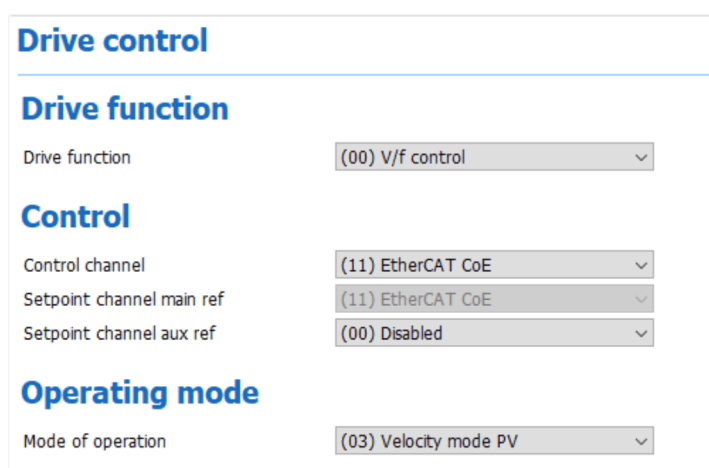


Fig. 45: EtherCAT parameterization in *drivemaster4*

The set mode of operation “(03) Velocity mode PV” corresponds to [profile velocity mode \(p. 16\)](#). For [velocity mode \(p. 16\)](#) you must set the mode of operation to “(02) Velocity mode VL”.

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

Acyclic communication is independent of the reference value channel and works always.

Note

For EtherCAT, you do not need to make any settings on the parameter page “Fieldbus”. The EtherCAT MDevice (PLC) conducts the PDO mapping and sends it to the SD4x drive.

4.2 Diagnosis in *drivemaster4*

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

4.2.1 Telegrams

The following figure shows the logged telegrams on the EtherCAT diagnosis page.

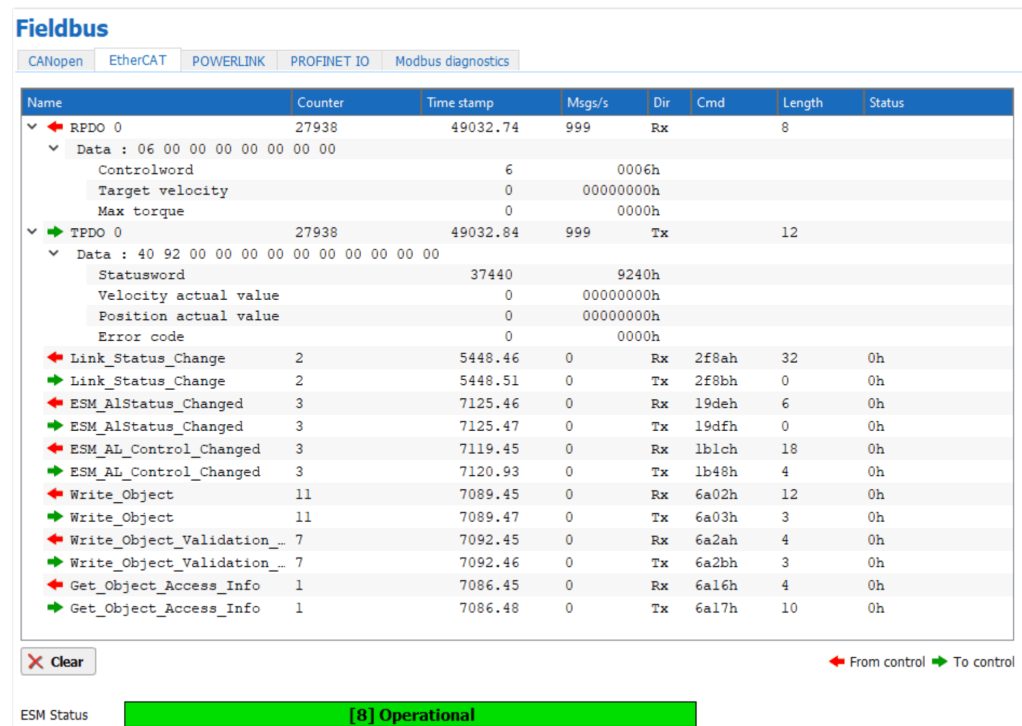


Fig. 46: EtherCAT bus activities

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.

ESM status

This field indicates the status of the EtherCAT state machine. The following states are possible: Init, Pre-Operational, Bootstrap, Safe-Operational, Operational.

4.3 Troubleshooting: State Machine Ignores Control Commands

External events that change the output stage status override the control commands from the controlword of the state machine. This override persists as long as the event is active. The following events are the most common causes of such status changes:

1. Disabling digital input
 - The digital input functions **(01) Switch on** and **(02) Switch on static** are AND-linked with **control bit 0 (switch on)** in the controlword. Therefore, they can prevent switching on via fieldbus.
 - The digital input function **(05) Quick stop** is AND-linked with **control bit 2 (quick stop)** in the controlword. Therefore, it can activate the quick stop function independently of the fieldbus.

To fix the problem, you can deselect the disabling input in the parameterization, see documentation “Drive Controller SD4x – Functions and Parameters”, chapter “Digital Inputs”.

2. Power supply failure

If the power supply is interrupted, the SD4x drive reacts as follows:

 - With output switched on: The drive switches off with error E48.
 - With output stage switched off: The drive automatically switches to the status “Switch on disabled”.
3. Safety function STO (Safe Torque Off)

If the safety function STO is triggered, the SD4x drive reacts as follows:

 - With output stage switched on: The drive switches off with error E18.
 - With output stage switched off: The drive automatically switches to the status “Switch on disabled”.

For more information on the safety function STO refer to the hardware documentation of your SD4x drive, chapter “Functional Safety”.

4. Control via *drivemaster4* setup tool

If device control of the SD4x drive via *drivemaster4* setup tool is active, the device cannot be controlled via fieldbus. In this case, **status bit 8 (Drive Setup Tool (DRS) active)** is set.



Tip

In the *drivemaster4* software under “Tools/Diagnosis → Actual drive values → Enable/Disable”, you can see all hardware and software components that are relevant to enable operation via fieldbus.

5 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
EtherCAT Technology Group	▶ ETG.1000 1 - 6 EtherCAT services and protocols ▶ ETG.2000 EtherCAT SubDevice Information (ESI), specification and implementation
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
EtherCAT Technology Group	www.ethercat.org
CiA e.V.	www.can-cia.org
Beckhoff Automation GmbH & Co. KG	www.beckhoff.com

Appendix

A Prefixes (from CiA 303-2)

Index	Symbol	Factor
0x13 ... 0x7F	Reserved	
0x12	E	10^{18}
0x11	-	10^{17}
0x10	-	10^{16}
0x0F	P	10^{15}
0x0E	-	10^{14}
0x0D	-	10^{13}
0x0C	T	10^{12}
0x0B	-	10^{11}
0x0A	-	10^{10}
0x09	G	10^9
0x08	-	10^8
0x07	-	10^7
0x06	M	10^6
0x05	-	10^5
0x04	-	10^4
0x03	k	10^3
0x02	h	10^2
0x01	da	10^1
0x00	-	10^0
0xFF	d	10^{-1}
0xFE	c	10^{-2}
0xFD	m	10^{-3}
0xFC	-	10^{-4}
0xFB	-	10^{-5}
0xFA	μ	10^{-6}
0xF9	-	10^{-7}
0xF8	-	10^{-8}
0xF7	n	10^{-9}
0xF6	-	10^{-10}
0xF5	-	10^{-11}
0xF4	p	10^{-12}
0xF3	-	10^{-13}
0xF2	-	10^{-14}
0xF1	f	10^{-15}
0xF0	-	10^{-16}
0xEF	-	10^{-17}
0xEE	a	10^{-18}
0xED to 0x80	Reserved	

B Units (from CiA 303-2 and CiA 402)

Index	Symbol	Factor
0x00	Dimensionless	–
0x01	m	Meter
0x02	kg	Kilogram
0x03	s	Second
0x04	A	Ampere
0x05	K	Kelvin
0x06	mol	mole
0x07	cd	Candela
0x10	rad	Radian
0x11	sr	Steradian
0x20	Hz	Hertz
0x21	N	Newton
0x22	Pa	Pascal
0x23	J	Joule
0x24	W	Watt
0x25	C	Coulomb
0x26	V	Volt
0x27	F	Farad
0x28	Ω	Ohm
0x29	S	Siemens
0x2A	Wb	Weber
0x2B	T	Tesla
0x2C	H	Henry
0x2D	°C	Degree celsius
0x2E	lm	Lumen
0x2F	lx	Lux
0x30	Bq	Becquerel
0x31	Gy	Gray
0x32	Sv	Sievert
0x33	kat	Katal
0x40	g	Gon
0x41	°	Grade
0x42	'	Arc minute
0x43	"	Arc second
0x44	l or L	Liter
0x45	a	Are
0x46	ha	Hectare
0x47	min	Minute
0x48	h	Hour
0x49	d	Day
0x4A	a	Year
0x4B	g	Gram
0x4C	t	Ton (metric)
0x4E	bar	Bar
0x4F	P	Poise
0x50	St	Stokes
0x51	eV	Electron volt
0x52	u	Atomic mass unit
0x53	AU	Astronomic unit
0x54	pc	Parsec
0x55	m/s^2	Meter per square second

Index	Symbol	Factor
0x56	Nm	Newton meter
0x57	s ²	Square second
0x58	m ²	Square meter
0x59	m ³	Cubic meter
0x5A	Pa?s	Pascal second
0x5B	J/(kg?K)	Joule per kilogram multiplied by Kelvin
0x5C	W/(m?K)	Watt per meter multiplied by Kelvin
0x5D	J/(mol?K)	Joule per mole multiplied by Kelvin
0x5E	W/(m ² ?sr)	Watts per square meter multiplied by steradian
0x5F	kat/m ³	Katal per cubic meter
0xA0	s ³	Cubic second
0xB4	–	Revolution
0xB5	–	Increment
0xAC	–	Step

Index

A

Abbreviations [5](#)
Addressing [39](#)
Addressing (TC3) [50](#)
Auto increment address (TC3) [51](#)

C

Configured Station-Alias (TC3) [53](#)
Control commands of drive [10](#)
Controlword [10](#)
CSV mode [14](#)
Cyclic synchronous velocity mode [14](#)

D

DC synchronous [62](#)
Diagnosis
 drivemaster4 [64](#)

E

ESM states [45](#)
EtherCAT address (TC3) [51](#)
Explicit device identification [54](#)

F

Free Run [62](#)

H

Hot Connect (TC3) [53](#)

O

Objects
 abort connection option code [12](#)
 acceleration [30](#)
 controlword [10](#)
 current actual value [28](#)
 DC link circuit voltage [28](#)
 device type [56](#)
 digital inputs [34](#)

 digital outputs [35](#)
 disable operation option code [13](#)
 error code [36](#)
 fault reaction option code [14](#)
 halt option code [13](#)
 interpolation time period [26](#)
 jerk [30](#)
 max acceleration [33](#)
 max current [32](#)
 max deceleration [33](#)
 max motor speed [32](#)
 max profile velocity [32](#)
 max torque [32](#)
 mode of operation [24](#)
 mode of operation display [25](#)
 motion profile type [27](#)
 motor rated current [28](#)
 motor rated torque [28](#)
 motor type [29](#)
 negative torque limit [33](#)
 polarity [31](#)
 position [29](#)
 position actual value [27](#)
 positive torque limit [33](#)
 profile acceleration [33](#)
 profile deceleration [34](#)
 quick stop deceleration [34](#)
 quick stop option code [13](#)
 RPDO mapping parameter [57](#)
 shutdown option code [13](#)
 statusword [11](#)
 supported drive modes [24](#)
 target torque [26](#)
 target velocity [25](#)
 torque actual value [28](#)
 torque demand [28](#)
 torque offset [31](#)
 torque slope [34](#)
 TPDO mapping parameter [58](#)

- velocity [30](#)
- velocity actual value [27](#)
- velocity demand value [27](#)
- velocity offset [31](#)
- vl dimension factor [22](#)
- vl set-point factor [21](#)
- vl target velocity [18](#)
- vl velocity acceleration [19](#)
- vl velocity actual value [18](#)
- vl velocity deceleration [20](#)
- vl velocity demand [18](#)
- vl velocity min max amount [18](#)
- vl velocity quick stop [20](#)

Operating mode change [23](#)

Operating modes [14](#)

Option codes [12](#)

P

Parameter set change [36](#)

PDO [8](#)

Profile torque mode [23](#)

Profile velocity mode [16](#)

S

SDO [7](#)

SDO communication (TC3) [46](#)

State machine [9](#)

- troubleshooting [66](#)

State transitions of the drive [9](#)

States [11](#)

Statusword [11](#)

Synchronization types [61](#)

T

Torque mode [23](#)

TwinCAT3 connection [39](#)

V

Velocity mode [16](#)

Velocity mode PV [16](#)

VL [16](#)