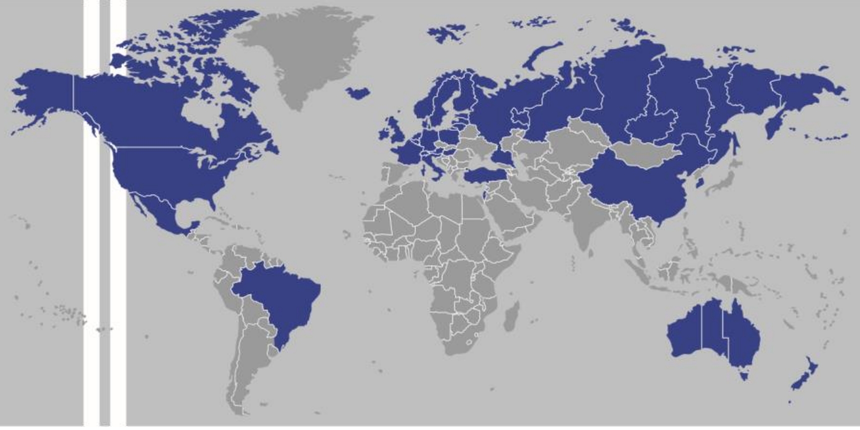


dunkermotoren
advanced motion solutions

DME 230 / 400

Functions and Parameters

for servo drives series
- BN6773
- BN6783

Publication Ref: 160616

Typ:	Part No:
DME 230x4-CO	81703.00100
DME 230x4-I/O	81703.00101
DME 230x4-EC	81703.00102
DME 230x4-PN	81703.00103
DME 400x8-CO	81703.00110
DME 400x8-EC	81703.00111
DME 400x8-PN	81703.00112

Dunkermotoren GmbH | Allmendstraße 11 | D-79848 Bonndorf/ Schwarzwald
Phone +49 (0) 7703 930-0 | Fax +49 (0) 7703 930-210/ 212 | info@dunkermotoren.com

AMETEK®
PRECISION MOTION CONTROL

TrioDrive D/xS / MidiDrive D/xS
TrioDrive D / MidiDrive D
MaxiDrive

Digital Servo Drives for Direct Mains Connection

Functions and Parameters

Operating Instructions 6710.201, V 8.5

These operating instructions apply to
- TrioDrive D/xS servo drives, compact design, BN 6755 to BN 6758
- MidiDrive D/xS servo drives, compact design, BN 6745 to BN 6749
- TrioDrive D servo drives, compact design, BN 6751 to BN 6753
- MidiDrive D servo drives, compact design, BN 6741 to BN 6743
- MaxiDrive servo drives, compact design, BN 6721 to BN 6725
- Operation via personal computer with SPP Windows software
- Access to device functions via communication interfaces

These operating instructions apply together with
- Operating Instructions 6755.202, 6745.202, 6750.202, 6740.202, or 6710.202 (Connection and Commissioning)
- and other operating instructions depending on the equipment

ESR Pollmeier GmbH
Lindenstr. 20

64372 Ober-Ramstadt

Federal Republic of Germany

Phone +49 6167 9306-0
Fax +49 6167 9306-77

E-Mail info@esr-pollmeier.de
www.esr-pollmeier.de

These operating instructions have been prepared with care. However, ESR Pollmeier GmbH can accept no liability for any errors in these operating instructions or possible consequences. Neither can any liability be accepted for direct or indirect damage resulting from abuse of the device.

The relevant regulations concerning safety technology and electromagnetic compatibility must be complied with when using the device.

Subject to alteration.

Contents

Please, also refer to the index at the end of this document.

1 Preliminary Remarks

1.1 About this Description

These Operating Instructions 6710.201 explain functions and parameters of the TrioDrive D/xS, MidiDrive D/xS, TrioDrive D, MidiDrive D, and MaxiDrive digital servo drives with built-in power supply unit for direct connection to 230 V AC (BN 6751 to BN 6758) or 3 × 400/480 V AC (BN 6741 to BN 6749, BN 6721 to BN 6725).

They are applicable together with

- Operating Instructions “Connection and Commissioning” of the servo drive (included in the scope of delivery)
  - Operating Instructions 6755.202 (TrioDrive D/xS),
  - Operating Instructions 6745.202 (MidiDrive D/xS),
  - Operating Instructions 6750.202 (TrioDrive D),
  - Operating Instructions 6740.202 (MidiDrive D) or
  - Operating Instructions 6710.202 (MaxiDrive)
- Operating Instructions “SPP Windows Command and Commissioning Software” (delivered with the optional command and commissioning software SPP Windows)
  - 6710.207

as well as, according to the equipment

- Operating Instructions “Part Program” (running motion sequences independent of a higher-level controller; delivered with the optional part program)
  - Operating Instructions 6710.231
- Operating Instructions “Communication Functions” (access to device functions via communication interface COM2 and/or Interbus; delivered with the optional Interbus interface)
  - Operating Instructions 6710.204
- Operating Instructions “CANopen® Interface” (delivered with the optional CANopen® interface)
  - Operating Instructions 6710.205 or
  - Operating Instructions 6745.205
- Operating Instructions “EtherCAT Interface” (delivered with the optional EtherCAT interface)
  - Operating Instructions 6745.232
- Operating Instructions “Ethernet Interface” (delivered with the optional Ethernet interface)
  - Operating Instructions 6745.236

- Operating Instructions “Profibus DP Interface” (delivered with the optional Profibus DP interface)
  - Operating Instructions 6730.208

For the commissioning of the functions described in these operating instructions, a PC with command and commissioning software SPP Windows is required. Please, make sure that this requirement is met and the above-mentioned operating instructions are available.

1.2 Storing Data

After the control supply voltage has been switched on, the machine data, the program variables, and, as far as included in the scope of delivery, the part program are loaded from the flash EPROM of the digital servo drive into its volatile memory (RAM). There, the data can be changed by command and commissioning software SPP Windows or by variable access via the communication interfaces.

These data can be saved on a data carrier (memory stick, hard disk of a PC, etc.) or in the flash EPROM of the servo drive using SPP Windows.

If machine data, part program, or program variables are not saved after having been changed, these changes will get lost when the control supply voltage is switched off.

For further information on storage media and storage, please see operating instructions 6710.207 “Command and Commissioning Software SPP Windows”.

1.3 Servo Drive System Packages

1.3.1 Technical Specifications of the Servo Drive System Packages

The TrioDrive D/xS, MidiDrive D/xS, TrioDrive D, MidiDrive D, and MaxiDrive servo drives have different technical specifications. This also applies to type, number, and assignment of inputs and outputs.

For details on the technical specifications of the individual servo drive, please see the corresponding operating instructions “Connection and Commissioning” 6755.202 (TrioDrive D/xS), 6745.202 (MidiDrive D/xS), 6750.202 (TrioDrive D), 6740.202 (MidiDrive D), or 6710.202 (MaxiDrive).

1.3.2 TrioDrive D/xS Servo Drive System Packages

TrioDrive D/xS servo drive system packages consist of

- the AC servo motor with coupled resolver, high-resolution incremental encoder, Sincos (Hiperface) encoder, or EnDat encoder as motor position sensor and

- the digital servo drive with built-in power supply unit, CANopen®, EtherCAT, Ethernet, Profibus, or analog interface and integrated safety system.

The series includes

- drives for 4 different currents (0.8 A_{rms} to 6 A_{rms}) for connection to 230 V AC,
- drives with varying equipment (options).

1.3.3 MidiDrive D/xS Servo Drive System Packages

MidiDrive D/xS servo drive system packages consist of

- the AC servo motor with coupled resolver, high-resolution incremental encoder, Sincos (Hiperface) encoder, or EnDat encoder as motor position sensor and
- the digital servo drive with built-in power supply unit, CANopen®, EtherCAT, Ethernet, Profibus, or analog interface and integrated safety system.

The series includes

- drives for 5 different currents (2 A_{rms} to 32 A_{rms}) for connection to 3 × 400/480 V AC,
- drives with varying equipment (options).

1.3.4 TrioDrive D Servo Drive System Packages

TrioDrive D servo drive system packages consist of

- the AC servo motor with coupled resolver, high-resolution incremental encoder, Sincos (Hiperface) encoder, or EnDat encoder as motor position sensor and
- the digital servo drive with built-in power supply unit.

The series includes

- drives for 3 different currents (2 $A_{rms\ to}$ 6 A_{rms}) for connection to 230 V AC,
- drives with varying equipment (options).

1.3.5 MidiDrive D Servo Drive System Packages

MidiDrive D servo drive system packages consist of

- the AC servo motor with coupled resolver, high-resolution incremental encoder, Sincos (Hiperface) encoder, or EnDat encoder as motor position sensor and
- the digital servo drive with built-in power supply unit.

The series includes

- drives for 3 different currents (2 A_{rms} to 8 A_{rms}) for connection to 3 × 400/480 V AC,
- drives with varying equipment (options).

1.3.6 MaxiDrive Servo Drive System Packages

MaxiDrive servo drive system packages consist of

- the AC servo motor with coupled resolver, high-resolution incremental encoder, Sincos (Hiperface) encoder, or EnDat encoder as motor position sensor and
- the digital servo drive with built-in power supply unit.

The series includes

- drives for 5 different currents (2 A_{rms} to 20 A_{rms}) for connection to 3 × 400 V AC,
- drives with varying equipment (options).

2 Safety Instructions

In any case, observe the safety instructions as well as the warnings and hints in the margins of the corresponding operating instructions “Connection and Commissioning” (6755.202, 6745.202, 6750.202, 6740.202, or 6710.202) and all other operating instructions.

Commissioning and parameterization of the servo drives may trigger drive movements. If drive system and/or machine have not been set up and secured properly, health and life of persons may be endangered.

Therefore, working with the drive system is prohibited until the requirements of the machine directive have been met.

In bus systems (CANopen®, EtherCAT, Ethernet, Interbus, Profibus, etc.), a bus participant can be influenced invisibly from outside. This can lead to an unexpected (uncontrollable) system behavior. Do not put the bus into operation unless you have made sure that all participants are properly connected and configured.

2.1 Type of Instructions

The warnings and hints in the margin must be observed under any circumstances:

- **Danger** to health and life due to electrical shock or motion of the drive. When disconnecting the device from the mains, wait for at least 2 minutes until the DC-bus capacitors have discharged before carrying out the measure described.
- **Caution**: Noncompliance violates the safety regulations or statutory provisions and can lead to personal injury or material damage.
- The CE marking requires compliance with the **EMC limits** for the first and second environment according to EN 61800-3 regarding emission and immunity. The instructions marked with this symbol must be observed by all means. Otherwise, the facility in which the drive is operated has to be checked for compliance with the EMC limits at the customer's own responsibility.
- **Check**: Prior to commissioning and in case of failures or problems, check these items first.
- **Tip**, useful hint.

3 Functions: Overview and Structure

The following figure shows an overview of the functional structure of the digital servo drives. Please, note that, depending on the equipment, not all axis operating modes are available for the servo drives.

Drive System Functions
Setpoint generator
Current control loop
Speed control loop
Position control loop
Axis state machine
Monitoring

Other Functions
Input/output
Switching points
Trigger/measuring
Trace
Safety

Part Program Functions
Part program
Part program state machine

Axis Operating Modes

- Torque mode / force mode
- Velocity mode direct
- Velocity mode
- Spindle positioning (sub mode)
- Profile position mode
- Homing mode
- Electronic gearing
- Flying shear
- Velocity profile
- Timed positioning mode
- Cyclic synchronous torque / force mode
- Cyclic synchronous velocity mode
- Cyclic synchronous position mode
- Interpolated position mode

Device Operating Modes

- Command mode
- Program mode

Figure 1: Structure of functions of the digital servo drives

Section Drive System Functions [14] describes the drive system functions. The axis operating modes are described in section Axis Operating Modes [29]. In section Device Operating Modes: Command Mode and Program Mode [60], you can find a description of the device operating modes. The other functions are described in section Input, Output, Trigger, Measuring, and Trace Functions [61]. Separate operating instructions (6710.231) with a description of the part program functions are delivered together with the optional part program.

The user can access the servo drive functions via variables which may consist of only one word (e. g. *position control loop Kp*) or many elements (e. g. *part program*).

In these operating instructions, the variables are printed in italics, described in detail, and – as far as reasonable and possible – assigned to the following variable types for a better clarity:

- Control and status information

  Variables for basic device functions, e.g. *axis control word* or *axis status word*

- Parameters and actual values

  Input and output values for functions, e.g. *target position* or *target velocity*

- Machine data

  Settings for the drive system, e. g. ramp and control loop settings, motor data, etc. These variables can be stored in the non-volatile memory of the servo drive.

The variables are summed up according to function groups and described in detail in section Variable Descriptions 78. Access by name is possible via the index.

To find out whether or not a certain variable is valid in a certain operating mode, see the corresponding information in line “Valid:...” of the individual variable description.

4 Drive System Functions

This section describes all functions of the controller cascade as well as the axis state machine and the monitoring functions.

The controller system includes

- setpoint generator
- position controller
- speed controller
- current controller (torque controller)

The axis control with *axis operating mode* and axis state machine determines which drive system functions are active at a certain point of time.

The following figure shows the basic structure of the controller cascade.

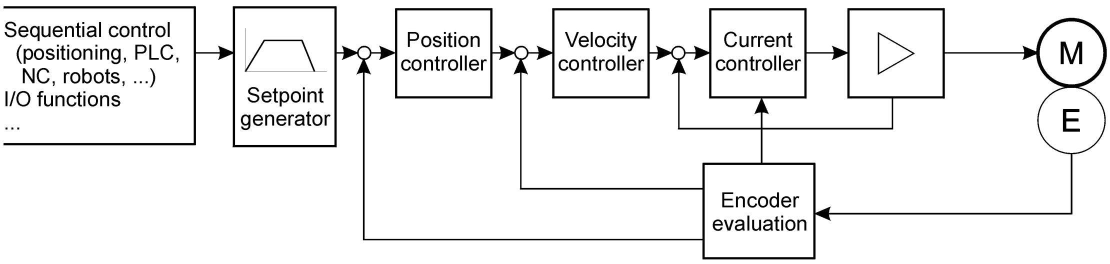


Figure 2: Principles of the controller cascade

To achieve high dynamics with the drive system, the controller setting must be optimized during first commissioning. For detailed information, see section Setting the Motor Data 67.

4.1 Setpoint Generator

The setpoint generator is responsible for the selection of the setpoint sources (e. g. for the speed the digital setting via variable or analog setting via analog input “setpoint”) and the calculation of acceleration and deceleration ramps.

The setpoints set by the setpoint generator are processed by the position, speed, and current controller which are arranged as controller cascade: the position controller generates a velocity setpoint which is processed by the speed controller which generates a current setpoint which is processed by the current controller. In addition to the setpoints, the drives also process feedforward values.

4.2 Position Controller

The position controller is a PI controller. During commissioning, the user must adjust the controller parameters to his machine. It is active in axis operating modes **spindle positioning**, **profile position mode**, **electronic gearing**, **flying shear**, and **interpolated position mode**. The position controller receives setpoints from the setpoint generator. In the controller cascade, the position controller is on a higher level than the speed controller.

4.3 Speed Controller

The speed controller is a PI controller. During commissioning, the user must adjust the controller parameters to his machine. In axis operating modes **velocity mode** and **velocity mode direct**, it receives its setpoints from the setpoint generator. In axis operating modes **profile position mode**, **electronic gearing**, **flying shear**, and **interpolated position mode**, it receives the setpoints from the position controller. In the controller cascade, the speed controller is on a lower level than the position controller and on a higher level than the current controller.

4.4 Current Controller (Torque Controller)

The current controller is a PI controller. The controller parameters depend on the used motor, they are set by selecting the motor during parameterization. In axis operating mode **torque mode**, the current controller receives its setpoints from the setpoint generator. In all other operating modes, it receives its setpoints from the speed controller which is on a higher level in the controller cascade.

4.5 Autocommutation

The current angular position of the rotor in relation to the stator or the position of the secondary part of the linear motor in relation to the primary part must be known so that the servo drive can supply the appropriate current depending on the desired torque or the desired force to the motor. For acquiring this commutation reference after switch-on, various methods are applicable depending on motor and position sensor type. These can be selected via parameter *commutation reference*:

- Absolute information

  When the motor position sensor supplies an absolute information on the position (within one electrical period), it can be read out by the servo drive and accepted as commutation reference. Usually, this applies to

  – rotatory motors with resolver, Sincos (Hiperface) encoder, or EnDat encoder as motor position sensor the mounting offset (i. e. the angle between zero point of the position sensor and phases U-V-W of the motor) of which is known and

– rotatory motors with high-resolution incremental encoders with Z track as motor position sensor the mounting offset of which is known.

For

– rotatory motors with incremental encoders without Z track and

– linear motors without hall sensor

an absolute information on the position is usually not available. For these motors, another method of commutation finding is required.

- Autocommutation (firmware V 8.5.94 and higher)

  When autocommutation is used, reactive current is supplied to the motor and the commutation reference is set in a way that no movement occurs. This control algorithm causes a minimal movement of the axis depending on the resolution of the position sensor and other factors (e. g. friction).

  The autocommutation is carried out automatically when after switch-on the axis state machine is set to state “switched on” or “operation enabled” for the first time. For test purposes, the commutation reference can be set in a way that the autocommutation is carried out with each change to “switched on” or “operation enabled”.

  The autocommutation requires a motor axis freely movable in both directions. For

  – motors with holding brake and

  – linear motors that are not arranged horizontally

  this is usually not true, another method of commutation finding is required.

  For motor position sensors with a low resolution (e. g. an incremental encoder with square-wave signals), execution and result of the autocommutation can be unsatisfactory. In this case, it should be checked whether or not another method of commutation finding would be more suitable.

- Zero point search using rotating field generator (firmware V 8.5.94 and higher)

  When the motor position sensor does not supply an absolute information (e. g. incremental encoder without Z track), however, an autocommutation is not possible or desired, the commutation reference can also be acquired via zero point search using rotating field generator.

  For the zero point search using rotating field generator, a rotating field defined via parameters *rotating field generator frequency* and *rotating field generator voltage* is supplied to the motor to make it move as soon as after switch-on the axis state machine is in state “switched on” or “operation enabled” for the first time. After the index pulse has been reached, the commutation reference is set and the drive system is stopped.

  Prerequisite for zero point search using rotating field generator is a motor position sensor supplying an index pulse.

- Mounting offset at current position (firmware V 8.5.94 and higher)

  If it is ensured that the motor is always at a certain position after switch-on, e. g.

  - a vertically mounted linear motor without holding brake being in the lower final position without current or
  - a motor without holding brake put in a defined position in current-free state by a spring,

  the commutation reference can be set to this position via the *position sensor mounting offset*.

  Caution: A plausibility check or the like is not carried out! If the motor is in another position during switch-on or *position sensor mounting offset* contains a wrong value, the commutation reference is not correct and the motor can start running uncontrolled at maximum velocity and torque or force in operation!

4.6 Axis State Machine

The control behavior of the axis is described using an axis state machine. The control behavior includes e. g. the steps for switching on the drive system and the different possibilities of shutting down the drive system (with braking ramp, quick-stop ramp, or voltage switch-off).

The drive system is switched on and off via the *axis control word* which influences the axis state machine.

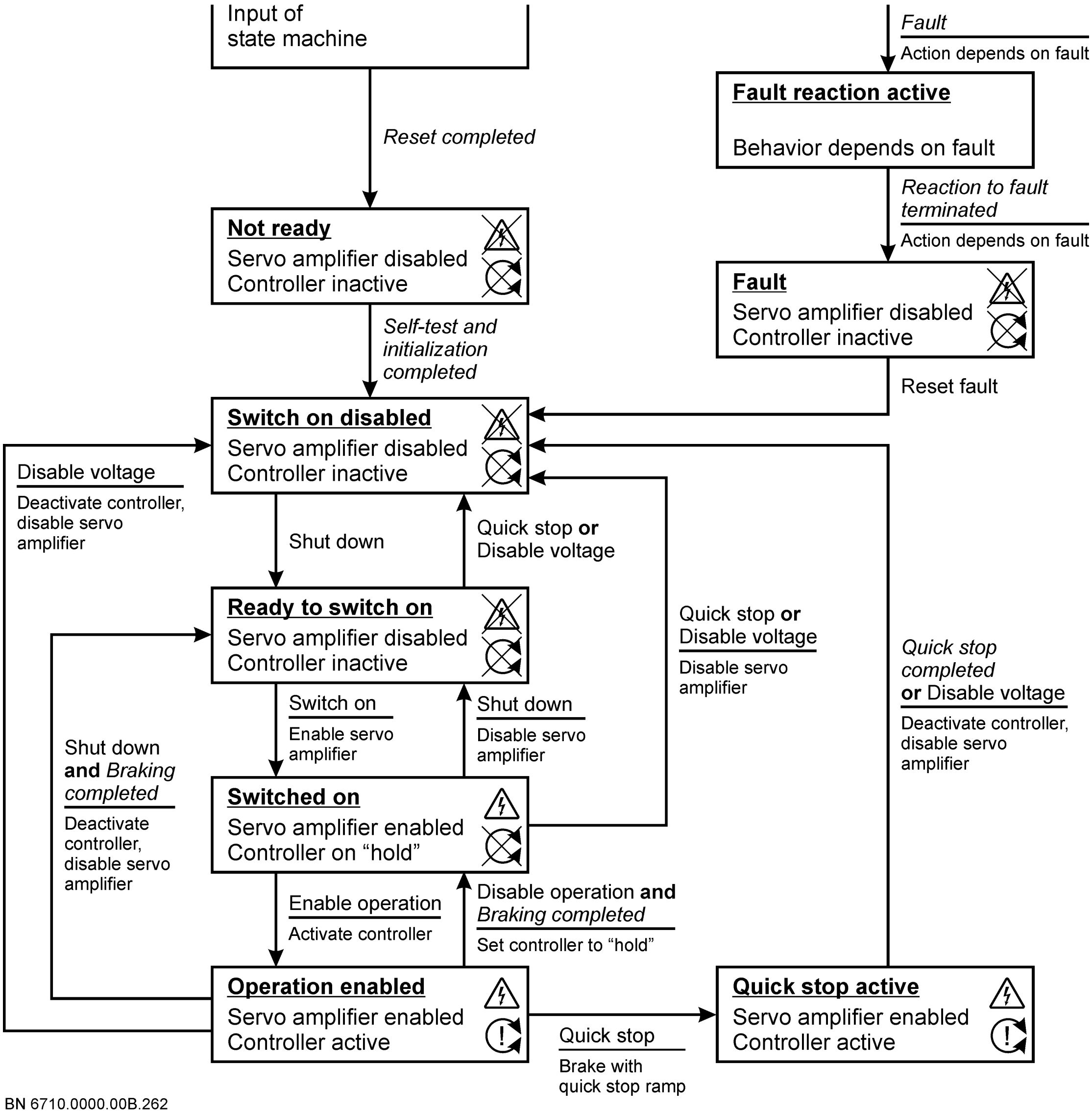


Figure 3: Axis State Machine

The status of the axis state machine can be queried via the *axis status word*.

Axis control includes monitoring functions. These functions switch off the drive system in case of a fault, report that using a fault bit in the *axis status word*, and explain the cause in the *axis error code*.

The variables for axis control and status information are described in section Axis Control and Status Information [83].

Basic information on state machine diagrams is included in Appendix A [190].

The state machine described here corresponds to the one defined by the DRIVECOM user group for networked drive systems. It is used by many manufacturers of electrical servo drive systems.

The axis is always in a defined state displayed via a group of bits in the *axis status word*. A complete overview of the individual bits in the *axis status word* can be found in section Axis Status Word 22.

The message that the axis state machine is in “operation enabled” state and the DC-bus voltage is above the switch-off threshold is communicated by a lit LED “Bereit / Ready” or “RDY” on the front panel of the servo drive (if at least one of these conditions is not fulfilled, LED “Bereit/Ready” or “RDY” flashes). This is a prerequisite for the start of an axis movement. For further details, see section Axis Control Word 20.

The states primarily determine the behavior of drive and motor as well as the enabling of the axis. This is indicated by additional explanations regarding the state. The following applies:

- Not ready

  The power circuit of the servo drive is blocked, the drive is not active.
- Switch on disabled

  The power circuit of the servo drive is blocked, the drive is not active.
- Ready to switch on

  The power circuit of the servo drive is blocked, the drive is not active.
- Switched on

  The power circuit is enabled, the drive is actively on hold.
- Operation enabled

  The power circuit is enabled, the drive is actively on setpoint.

The drive system functions work according to the selected axis operating mode. Position, speed, and torque or force of the motor depend on setpoints and parameters as well as the internal states of the drive system function.

A change between two states (a state transition) occurs when an event happens. During some changes, an action is carried out.

Possible events are

- a control command is given
  - via the *axis control word* or
  - input “enable”
- internal event in the servo drive
- external event at one of the servo drive interfaces

Control commands are usually given via the corresponding bit patterns in the *axis control word*. A complete overview of the individual bits in the *axis control word* can be found in section Axis Control Word 20.

For giving control commands via input “Enable” (“Ena”), the effect of the enable input must be configured using machine date *enable selection code*.

Event “fault” (e. g. limit switch) may occur in each state, it leads to a transition into state “fault reaction active”. To keep the figure clear, the corresponding arrow was not drawn at each state but only once on top without own previous state.

The internal and external events are described as follows:

- Reset completed

  The reset of the servo drive after switch-on of the supply voltages is completed, the communication is ready.

- Self-test and initialization completed

  These processes are completed, the axis state machine is ready for receiving control commands from the *axis control word*.

- Braking completed

  After braking with the braking ramp (*motion profile type*, *deceleration time*, and *ramps reference velocity* parameters), the drive system has reached zero velocity (*velocity threshold*, *velocity threshold time* parameters). In some axis operating modes, this is indicated via bit “Velocity = 0” in the *axis status word*.

- Quick stop completed

  After braking with the quick stop ramp (*motion profile type*, *quick stop time*, and *ramps reference velocity* parameters), the drive system has reached zero velocity (*velocity threshold*, *velocity threshold time* parameters). In some axis operating modes, this is indicated via bit “Velocity = 0” in the *axis status word*.

- Fault

  A fault has occurred (e. g. limit switch). Parameter *axis error code* provides more detailed information on the cause of the fault. Additionally, the fault is indicated by lighting or flashing of LED “Störung / Fault” or “FLT”.

- Fault reaction terminated

  Each fault leads to a particular reaction of the servo drive. The end of this reaction is indicated by this event.

4.6.1 Axis Control Word

The *axis control word* can be used for

- influencing the axis state machine (so that the drive system is switched on or off),
- influencing the part program state machine (so that the part program running can be controlled),
- resetting a fault
- some axis operating modes to give certain commands to the drive system (e. g. *start going to home position* in operating mode **homing mode**).

In the *axis control word*, the individual bits have the following meaning:

Bit	**Axis Control Word** (Bit Assignment)
0	Switch on (axis state machine)
1	Disable voltage (axis state machine)
2	Quick stop (axis state machine)
3	Enable operation (axis state machine)
4 .. 6	Meaning depends on the axis operating mode, see below
7	Reset fault (axis state machine)
8 .. 13	–
14	Part program start (short: PP start) (part program state machine)
15	Part program reset (short: PP reset) (part program state machine)

In the *axis control word*, bits 4 to 6 have the following meaning, depending on the axis operating mode:

- Profile position mode
  - Bit 4: New setpoint
  - Bit 5: Change block immediately
  - Bit 6: Absolute/relative
- Homing mode
  - Bit 4: Start homing mode
- Electronic gearing
  - Bit 4: Synchronize
- Interpolated Position Mode
  - Bit 4: Enable interpolation

In all other operating modes, bits 4 to 6 in the *axis control word* do not have any meaning.

The following table lists all control commands in the *axis control word*.

Axis Control Word (least significant byte)									
Bit	7	6	5	4	3	2	1	0	Control command
	–	1	0	↑	–	–	–	–	Accept target position (absolute) (in axis operating mode ***profile position mode/block mode***)
	–	0	0	↑	–	–	–	–	Accept target position (relative) (in axis operating mode ***profile position mode/block mode***)
	–	1	1	1	–	–	–	–	Accept target position permanently (absolute) (in operating mode ***profile position mode/setpoint mode***)
	–	–	–	1	–	–	–	–	Start homing mode (in axis operating mode ***homing mode***)
	–	–	–	1	–	–	–	–	Synchronize (in axis operating mode ***electronic gearing***)
	–	–	–	↑	–	–	–	–	Enable interpolation (in axis operating mode ***interpolated position mode***)

Axis Control Word (least significant byte)									
Bit	7	6	5	4	3	2	1	0	Control command
	x	–	–	–	x	1	1	0	Shut down (axis state machine)
	x	–	–	–	x	1	1	1	Switch on (axis state machine)
	x	–	–	–	1	1	1	1	Enable operation (axis state machine)
	x	–	–	–	0	1	1	1	Disable operation (axis state machine)
	x	–	–	–	x	0	1	x	Quick stop (axis state machine)
	x	–	–	–	x	x	0	x	Disable voltage (axis state machine)
	⎍	–	–	–	x	x	x	x	Reset fault (axis state machine)

x = bit may be 1 or 0
– = bit assigned to other functions

⎍ = transition 0 → 1

Axis Control Word (most significant byte)									
Bit	15	14	13	12	11	10	9	8	Control command
	x	1	–	–	–	–	–	–	Start (part program state machine)
	0	0	–	–	–	–	–	–	Stop (part program state machine)
	0	1	–	–	–	–	–	–	Resume (program state machine)
	1	0	–	–	–	–	–	–	Reset (program state machine)

x = bit may be 1 or 0
– = bit assigned to other functions

In the part program, the *axis control word* can be accessed via digital outputs O 10 (bit 0 .. 7) and O 11 (bit 8 .. 15). The part program functions are described in operating instructions 6710.231 “Part Program”.

4.6.2 Axis Status Word

In the *axis status word*,

- the device status can be read (state of the axis state machine),
- the status of the part program state machine can be read,
- certain events are displayed in some axis operating modes (e. g. setpoint reached or velocity = 0).

The individual bits in the *axis status word* have the following meaning:

Bit	**Axis Status Word** (Bit Assignment)
0	Ready to switch on (axis state machine)
1	Switched on (axis state machine)
2	Operation enabled (axis state machine)
3	Fault (axis state machine)
4	Voltage disabled (axis state machine)
5	Quick stop (axis state machine)
6	Switch on disabled (axis state machine)
7	–
8	–
9	–
10	Setpoint reached (SR), depending on the axis operating mode, see below
11	–
12 .. 13	Meaning depends on the axis operating mode, see below
14	Part program running (short PP running) (part program state machine)
15	Part program idle (short PP idle) (part program state machine)

Depending on the axis operating mode, bit 10 (setpoint reached) in the *axis status word* stands for:

- Torque mode: torque reached
- Velocity mode direct: velocity reached
- Velocity mode: velocity reached
- Profile position mode: in position
- Electronic gearing: synchronized
- Flying shear: in position (return position)

Depending on the axis operating mode, bits 12 and 13 in the *axis status word* have the following meaning:

- Velocity mode direct
  - Bit 12: velocity = 0
- Velocity mode
  - Bit 12: velocity = 0
- Profile position mode
  - Bit 12: setpoint acknowledgement
- Homing mode
  - Bit 12: home position reached
  - Bit 13: homing fault

- Electronic gearing
  - Bit 12: velocity = 0
- Cyclic synchronous torque mode
  - Bit 12: setpoints are accepted
- Cyclic synchronous velocity mode
  - Bit 12: setpoints are accepted
- Cyclic synchronous position mode
  - Bit 12: setpoints are accepted
- Interpolated Position Mode
  - Bit 12: setpoints are accepted (firmware V 8.5.9 and higher)
  - Bit 13: homing mode

In all other operating modes, bits 10, 12, and 13 in the *axis status word* do not have any meaning.

The following two tables show a list of all state codings for program state machine and axis state machine.

Axis Status Word (least significant byte)										
Bit	7	6	5	4	3	2	1	0		Status of the axis state machine
	–	0	x	–	0	0	0	0		Not ready
	–	1	x	–	0	0	0	0		Switch on disabled
	–	0	1	–	0	0	0	1		Ready to switch on
	–	0	1	–	0	0	1	1		Switched on
	–	0	1	–	0	1	1	1		Operation enabled
	–	0	0	–	0	1	1	1		Quick stop active
	–	0	x	–	1	1	1	1		Fault reaction active
	–	0	x	–	1	0	0	0		Fault

x = bit may be 1 or 0
– = bit assigned to other functions

Axis Status Word (most significant byte)										
Bit	15	14	13	12	11	10	9	8		Status of the part program state machine
	1	0	–	–	–	–	–	–		Idle (out of operation)
	0	1	–	–	–	–	–	–		Running
	0	0	–	–	–	–	–	–		Stopped

– = bit assigned to other functions

In the part program, the *axis status word* can be accessed via digital inputs I 10 (bit 0 .. 7) and I 11 (bit 8 .. 15). The part program functions are described in operating instructions 6710.231 “Part Program”.

4.6.3 Running up the Axis State Machine (Example)

To run up the axis state machine, proceed as follows:

- **Fault query and management**

  Before running up the axis state machine, query the faults and, if required, react correspondingly:

  – Condition: axis status word AND $004F_{hex}$ = 0008_{hex}

  In all following steps in which a certain status is waited for, the fault query and management must be carried out, as well.

- **Start with state “switch on disabled”**

  After the controller start, make sure that the axis state machine changes to state “switch on disabled”, no matter in which state it was before:

  – control command “quick stop” (axis control word: $xxx2_{hex}$) or

  – control command “disable voltage” (axis control word: $xxx0_{hex}$)

- **Wait until state “switch on disabled” has been reached**

  Before the next control command is given, make sure that state “switch on disabled” has been reached.

  – Condition: axis status word AND $004F_{hex}$ = 0040_{hex}

  In “switch on disabled” state, the power circuit is not enabled (no voltage at the motor) and the control loops are not active (motor shaft can be moved manually).

- **Switch to state “ready to switch on”**

  – control command “hold” (axis control word: $xxx6_{hex}$)

- **Wait until “ready to switch on” state has been reached**

  Before the next control command is given, make sure that the “ready to switch on” state has been reached.

  – Condition: axis status word AND $006F_{hex}$ = 0021_{hex}

  In the “ready to switch on” state, the power circuit is not enabled (no voltage at the motor) and the control loops are not active (motor shaft can be moved manually).

  The function of state “ready to switch on” does not differ from state “switch on disabled”. This additional step ensures that the drive system cannot be switched to “switched on” state during a new start e. g. because control command “switch on” still exists in the process data.

- **Switch further to “operation enabled” state**

  The “switch on” and “enable operation” control commands can be given simultaneously (see table "Coding of Control Commands for the Axis State Machine"):

– Control command "enable operation" (axis control word: $xxxF_{hex}$)

- **Wait until "operation enabled" state has been reached**

– Condition: axis status word AND $006F_{hex} = 0027_{hex}$

In state "operation enabled", the power circuit is enabled (voltage exists at the motor). The control loops are active, the motor shaft is moved according to the defined setpoints depending on the selected axis operating mode.

4.6.4 Switching Between Operating Modes

In certain cases, switching between the operating modes is required.

Example: After homing (axis operating mode **homing mode**), the operating mode has to be changed to **profile position mode**. This axis operating mode change is not permitted in "operation enabled" state, therefore, it is necessary to change to another state of the axis state machine to change the operating mode.

The following sections describe the individual steps of this example in detail.

- **Run up the axis state machine, homing mode**

In axis operating mode **going to home position**, the state machine is run up as described in section Running up the Axis State Machine (Example) 25 and the homing mode is carried out.

- **Switch to "switched on" state**

– Control command "disable operation" (axis control word: $xxx7_{hex}$)

- **Wait until "switched on" state has been reached**

– Condition: axis status word AND $006F_{hex} = 0023_{hex}$

In "switched on" state, the power circuit is enabled (voltage exists at the motor). The velocity controller (speed controller) is active and regulates to setpoint 0, thus, the motor shaft is stopped. Exception: In axis operating mode **torque mode**, only the current controller is active. It also regulates to a 0 setpoint so that the axis can be moved manually.

- **Change operating mode**

In "switched on" state, the operating mode can be changed, e. g.

– axis operating mode **profile position mode** (axis operating mode: 01_{hex})

– sub mode **setpoint mode** (axis control word $xx7x_{hex}$)

- **Switch back to state "operation enabled"**

– Control command "operation enabled" (axis control word: $xxxF_{hex}$)

- **Wait until "operation enabled" state has been reached**
  - Condition: axis status word AND $006F_{hex} = 0027_{hex}$

  The motor shaft moves according to setpoint target position and parameters such as target velocity.

4.7 Monitoring

In case of a fault, monitoring functions switch off device and drive system and report a fault in bit 3 of the *axis status word*. Possible faults include excessive current in the motor circuit, overtemperature, or blocking of the motor, wrong connection of the motor position sensor, errors when going to home position or related to limit switches, over- or undervoltage in the DC bus, overtemperature in power circuit or shunt resistor, or errors in part program operation.

After the cause has been removed, the fault can be reset via the *axis control word*.

In addition to axis operating modes and axis state machine, the axis control includes functions providing information on the axis other than the state of the axis state machine or controlling outputs. These are used for monitoring purposes.

- Output "fault"

  When state "fault" has been reached, output "fault" will be activated. State "fault" (and thus the output) can be reset via a control command in the axis control word or input "reset fault".

- Axis error code

  In case of a fault (axis state machine in "fault" state), the *axis error code* displays the cause of the fault. It is displayed until the error has been removed and state "fault" has been left using control command "reset fault". An overview of the axis error codes can be found in Appendix B 191.

4.8 Safety System (TrioDrive D/xS and MidiDrive D/xS, only)

In the TrioDrive D/xS and MidiDrive D/xS servo drives, the wear-free electronic conception for **"safe standstill"** (corresponds to stop category 0 according to EN 954-1 or STO according to EN 61800-5-2) developed by ESR is used. For that, the control energy for the upper and lower power circuit bridge leg is supplied separately via safety inputs SI1 and SI2. If at least one of the two voltages is switched off, current cannot flow into the motor winding any longer.

In the hardware (power circuit), safety inputs SI1 and SI2 trigger the "**safe standstill**" function by disabling the power circuit as soon as at least one of inputs SI1 or SI2 is active (= open = 0 V). This is stored and displayed by signal "safe standstill".

The axis state machine changes to "switch on disabled" state.

Due to the two-channel design (safety category 4 according to EN 954-1), actions relevant for safety are not required on the controller side. The current status of the safety system is displayed in the device control window of SPP Windows (see operating instructions 6710.205). It can be output on digital outputs using signals “input 1 safe standstill”, “input 2 safe standstill”, and “safe standstill triggered”. For the evaluation by the controller, the state can also be read out at digital input 3 via bit 4-7:

Input	Signal
I 3.4	Input 1 safe standstill
I 3.5	Input 2 safe standstill
I 3.6	Logical interconnection 3.4 OR 3.5
I 3.7	Safe standstill triggered

The “safe standstill triggered” signal is reset automatically as soon as axis state “switch on disabled” has been left via the axis state machine.

When the values of input SI1 and SI2 differ for more than one minute, a fault in the external two-channel switching is assumed and the power circuit is additionally switched off by the software. The axis state machine changes to “fault” state, a corresponding fault code is displayed.

The functions relevant for safety are realized in hardware. The firmware serves for display and diagnostics of the safety functions as well as for switching on again by corresponding control commands after the hardware switch-off. It is carried out in the part of the device that is not relevant for safety.

5 Axis Operating Modes

The axis operating modes can be used for determining whether e. g. velocities or positions will be preset. Depending on the servo drive equipment, not all axis operating modes are available. Possible axis operating modes are:

- **Torque mode / force mode,**
- **Velocity mode direct,**
- **Velocity mode,**
- **Spindle positioning** (sub mode)**,**
- **Profile position mode,**
- **Homing mode,**
- **Electronic gearing,**
- **Flying shear,**
- **Velocity profile,**
- **Timed positioning mode**
- **Cyclic synchronous torque/force mode**
- **Cyclic synchronous velocity mode**
- **Cyclic synchronous position mode**, and
- **Interpolated Position Mode**

The axis operating mode is selected via variable *axis operating mode*. It may be changed in states “switch on disabled”, “ready to switch on”, and “switched on”, only.

All axis operating modes use the drive functions responsible for the actual control and processing of setpoint and actual values. Different controllers are active, depending on the operating mode. You will find further details in section Drive System Functions 14.

One condition for the drive to operate properly in these axis operating modes is the setting of basic machine data during commissioning.

Actual values can be read in all axis operating modes via the following variables:

- actual position: *actual position* (position units, to be used preferably), *position sensor actual position* (position sensor steps)
- actual velocity: *actual velocity*
- actual current: *actual current*

The units for position and velocity can be converted into practical units:

- The coordinate system can be adapted to the application via a factor and a zero offset. Due to that, the user is not confined to the steps of the position measuring system, but can define units of his own.

  By default, the units are position sensor steps (PSS) preset by the evaluation of the position measuring system. Using the *position factor* and *home offset* variables, the position sensor steps can be converted into more useful units (e. g. mm or angular units). In the following, these units, which can be selected by the user, are called position units (PU).

The zero offset is carried out via the *home offset* variable.

- The velocity units can be adapted to physical units (e.g. r.p.m. or mm/s) using the *velocity factor setpoint* variable. In the following, these units, which can be selected by the user, are called velocity units (VU).

For further details on how to set the conversion factors, see section Factors and Units Machine Data 158.

In the variable descriptions of these operating instructions the units are stated for all variables containing position and velocity values.

The axis can be operated as

- linear axis or as
- circular axis.

The corresponding setting is carried out via the *axis type* variable. For further information, see section Positioning Range Machine Data 161.

The control loops (and thus the drive functions) are only active when the axis state machine is in "operation enabled" state. Furthermore, a motion can only take place if the bus voltage is high enough (i. e. in addition to the control supply voltage, the mains must be switched on).

The message that the state machine is in "operation enabled" state and the bus voltage is above the shutdown threshold is indicated by the lighting up of the "Bereit / Ready" or "RDY" LED on the front panel of the servo drive (a flashing "Bereit / Ready" or "RDY" LED indicates that one of these conditions is not met).

5.1 Torque Mode / Force Mode

In axis operating mode **torque mode** (or **force mode** for a linear motor), the torque of the axis can be set via the *torque setpoint* parameter or analog input "setpoint". The direction in which the torque is effective is specified by the sign. Whether the torque is to be preset as a digital or analog value can be selected via machine data *torque setpoint source*. Condition for the start of the motion is that the drive is "Ready".

If the torque is preset via analog input "setpoint", the scaling of the analog value (assignment of the applied setpoint voltage to a specific speed) and the correction of the offset can be achieved via the *torque analog factor* and *torque analog offset* variables.

In torque mode, the *axis control word* does not contain any operating mode-dependent bits.

In the *axis status word*, bit 10 has the following special meaning:

- torque setpoint reached (bit 10 in the *axis status word*)

This message can also be output on a digital output, see *signal selection digital* *output...* in section Input/Output Function Machine Data [131].

In this operating mode, setpoint generator and current control loop work together.

The speed in axis operating mode **torque mode** is currently not limited, thus, the maximum speed specified in the *maximum motor speed* variable may be exceeded.

To find out whether or not a certain variable is valid in axis operating mode **torque mode**, see the corresponding information of the individual variable descriptions in line “Valid: ...”.

Operating mode **torque mode** is not suitable for realizing an (external) speed control because the input values (analog input) are filtered which leads to delays.

5.2 Velocity Mode Direct

In axis operating mode **velocity mode direct**, the velocity of the axis can be set via analog input “setpoint”. This axis operating mode is only possible for servo drives equipped with an analog input. The analog setpoint is read directly from the digital signal processor every 62.5 µs via the microcontroller and processed directly by the speed controller. The direction of motion is specified by the sign. Condition for the start of the motion is that the drive is “Ready”.

Scaling of the analog value at input “setpoint” (assignment of the applied setpoint voltage to a specific speed) and correction of the offset can be achieved via the *target velocity analog factor* and *target velocity analog offset* variables.

Axis operating mode **velocity mode direct** differs from the **velocity mode** operating mode as follows:

- The velocity of the axis is always set via analog input “setpoint” (the *target velocity source* parameter has no effect), the *override factor* is not taken into account.
- The set velocity is immediately accepted as setpoint, ramps are not taken into account.

In **velocity mode direct**, the *axis control word* does not contain any operating mode-dependent bits.

In **velocity mode direct**, bits 10 and 12 in the *axis status word* have the following special meaning:

- target velocity reached (bit 10 in the *axis status word*)
- velocity = 0 (bit 12 in the *axis status word*)

If bit target velocity reached (or more generally setpoint reached, SR for short) has value 1, this shows that, after acceleration, the drive system has reached the set target velocity. For that, the *velocity window* and *velocity window time* parameters are taken into account. This message can also be output on a digital output, see *digital output ... signal selection* in section Input/Output Function Machine Data [131].

If bit velocity = 0 is set, this shows that, after deceleration, the drive has reached velocity zero. For that, the *velocity threshold* and *velocity threshold time* parameters are taken into account.

To safely determine that velocity zero has been reached after deceleration, the corresponding *velocity threshold* and *velocity threshold time* variables must be set so that, even if the axis is subject to motion as a result of control oscillations or an external force, this movement never exceeds the speed specified in the *velocity threshold*.

In operating mode **velocity mode direct**, the motions are influenced by the following parameters:

- Speed control loop machine data with *speed filter cut-off frequency*, *speed control loop Kp*, *speed control loop Ki*, *speed control loop total amplification*
  - determine the controller behavior: smoothing of the actual speed, gain, and optional I component (to compensate for any remaining control offset).

In operating mode **velocity mode direct**, setpoint generator and speed controller work together, the current controller is subordinate to the speed controller.

To find out whether or not a certain variable is valid in axis operating mode **velocity mode direct**, see the corresponding information of the individual variable descriptions in line "Valid: ...".

5.3 Velocity Mode

In axis operating **mode velocity mode**, the velocity of the axis can be set via the *target velocity (digital)* parameter or analog input "setpoint". In both cases, the value is multiplied with the *override factor*. The direction of motion is specified by the sign. Whether the velocity is to be set as a digital or analog value can be selected via machine data *target velocity source*. Condition for the start of the motion is that the drive is "Ready".

If the target velocity is set via analog input "setpoint", the scaling of the analog value (assignment of the applied setpoint voltage to a specific speed) and a correction of the offset can be achieved via the *target velocity analog factor* and *target velocity analog offset* variables.

Axis operating mode **velocity mode** differs from the **velocity mode direct** operating mode as follows:

- The velocity of the axis can be preset either via analog input "setpoint" or via the *target velocity (digital)* parameter, the *override factor* is taken into account.
- Acceleration and deceleration ramps are taken into account during internal setpoint calculation.

In **velocity mode**, the *axis control word* does not contain any operating mode-dependent bits.

In **velocity mode**, bits 10 and 12 in the *axis status word* have the following special meaning:

- target velocity reached (bit 10 in the *axis status word*)
- velocity = 0 (bit 12 in the *axis status word*)

If bit “target velocity reached” (or more generally, “setpoint reached”, SR for short) has value 1, this shows that, after acceleration, the drive has reached the set target velocity. For that, the *velocity window* and *velocity window time* parameters are taken into account. This message can also be output on a digital output, see *signal selection digital output ...* in section Input/Output Function Machine Data [131].

If bit “velocity = 0” is set, this shows that, after deceleration, the drive has reached velocity zero. For that, the *velocity threshold* and *velocity threshold time* parameters are taken into account.

To safely determine that velocity zero has been reached after deceleration while braking, the corresponding *velocity threshold* and *velocity threshold time* variables must be set in a way that, even if the axis is subject to motion as a result of control oscillations or an external force, this movement never exceeds the speed specified in the *velocity threshold*.

In operating mode **velocity mode**, the motions are influenced by the following parameters:

- Ramps machine data (section Ramps Machine Data [164]) with *motion profile type*, *acceleration time*, *deceleration time*, *quick stop time*, and *ramps reference velocity*
  - determine how to reach the velocity and how to decelerate again.
- Speed control loop machine data (section Speed Control Loop Machine Data [167]) with *speed filter cut-off frequency, speed control loop Kp, speed control loop Ki, speed control loop total amplification*
  - determine the controller behavior: smoothing of the actual speed, gain, and optional I component (to compensate for any remaining control offset)

In operating mode **velocity mode**, setpoint generator and speed controller work together, the current controller is subordinate to the speed controller.

To find out whether or not a certain variable is valid in axis operating mode **velocity mode**, see the corresponding information of the individual variable descriptions in line “Valid: ...”.

5.4 Spindle Positioning

The **spindle positioning** function is actually an axis sub mode. It serves for preparing the tool or the tool change and can be called up in axis operating modes **velocity mode** and **velocity mode direct** (from firmware V 5.4a on).

With the axis rotating, a positioning process is started via the “spindle position” command. This positioning turns the axis into a defined spindle position. For that, the target velocity is increased or decreased to the spindle positioning velocity. In a state of an active spindle positioning (input = 1), a target velocity must not be set from outside.

Signal “spindle position reached” indicates that the target position has been reached. For that, the *spindle positioning window* and the *spindle positioning window time* parameters are taken into account. When the drive has reached zero velocity, signal “velocity = 0” is output. The *velocity threshold* and *velocity threshold time* parameters are taken into account. Both messages can also be output on digital outputs.

The spindle positioning process is ended and the drive system returns to axis operating mode **velocity mode** or **velocity mode direct** as soon as input spindle positioning is reset to zero.

For spindle positioning, the axis must be operated as a circular axis. The corresponding setting is made via the *axis type* variable.

In **spindle positioning**, the bit assignment in *axis control word* and *axis status word* corresponds to the bit assignment of axis operating modes **velocity mode** or **velocity mode direct**.

The motions during **spindle positioning** are influenced by the following parameters:

- Ramps machine data (Ramps Machine Data 164) with *motion profile type*, *acceleration time*, *deceleration time*, *quick stop time*, and *ramps reference velocity*
  - determine how to achieve the spindle positioning velocity and how to decelerate to velocity 0 on approaching target position.
- Speed control loop machine data (section Speed Control Loop Machine Data 167) with *velocity threshold* and *velocity threshold time parameters.*
  - For activating the spindle positioning, the current velocity must be higher than zero; the *velocity threshold* and *velocity threshold time* parameters define this state.
- Positioning range machine data (section Positioning Range Machine Data 161) with *axis type*, *range limit min* and *range limit max*
  - axis type must be set to “circular axis”; the *range limit min* and *range limit max* variables specify the valid range for the spindle position.
- Spindle positioning machine data (section Spindle Positioning Machine Data 181) with *spindle positioning direction*
  - determine the direction of rotation during spindle positioning.

In the **spindle positioning** function, setpoint generator and position controller work together, speed controller and current controller are subordinate to the position controller.

5.5 Profile Position Mode

Axis operating mode **profile position mode** can be used for moving to a specific position. This position is preset by the user via the *target position*. Bits 4 to 6 in the *axis control word* ensure that, under certain conditions, the target position is transferred to the *internal target position* variable. The position in *internal target position* is then approached. Condition for the start of the motion is that the drive is “Ready”.

The velocity the target position is approached with is specified via the *target velocity (digital)* parameter or via analog input “setpoint”, depending on the setting in the machine data *target velocity source*. In both cases, the value is multiplied by the *override factor*.

Each time the axis operating mode is switched to **profile position mode**, the actual position is transferred to the *internal target position*. This keeps the axis at its current position until a new target position is set.

In **profile position mode**, bits 4 to 6 in the *axis control word* have the following special meaning:

- new setpoint (bit 4 in the *axis control word*)
- change block immediately (bit 5 in the *axis control word*)
- absolute/relative (bit 6 in the *axis control word*)

Bit “change block immediately” selects one of the two sub-modes

- block mode (bit “change block immediately” = 0) or
- setpoint mode (bit “change block immediately” = 1).

The effect of the two bits “new setpoint” and “absolute/relative” depends on the selected sub-mode; these sub-modes are described in the next sections. An overview of the effect of these control bits is given in the following table.

Profile position mode – operating mode-dependent control bits											
Transfer of *target position* to *internal target position...*	**Axis control word** (least significant byte)										
	Bit	7	6	5	4	3	2	1	0		sub-mode of the profile position mode
...as absolute position		–	1	0	ʃ	–	–	–	–		block mode
...as relative position		–	0	0	ʃ	–	–	–	–		
permanent acceptance (as absolute position)		–	1	1	1	–	–	–	–		setpoint mode

x = bit may be 1 or 0
– = bit assigned to other functions
ʃ = transition 0 → 1

Bit names: Bit 6 = “absolute/relative”
Bit 5 = “change block immediately”
Bit 4 = “new setpoint”

In **profile position mode**, bits 10 and 12 in the *axis status word* have the following special meaning:

- setpoint reached (bit 10 in the *axis status word*)
- setpoint acknowledgment (bit 12 in the *axis status word*)

With value 1, bit “position setpoint reached” (also “in position”, IP, or more generally “setpoint reached”, SR) shows that the axis has reached the *internal target position*. For this decision, the *positioning window* and *positioning window time* parameters are taken into account. This message can also be output on a digital output, see *signal selection digital output ...* in section Input/Output Function Machine Data [131].

Bit “setpoint acknowledgment” shows the transfer of the *target position* to the *internal target position* and is described in more detail under sub-modes **block mode** and **setpoint mode**.

The sequence of motions when approaching a *target position* is influenced by the following parameters:

- *Target velocity* (section Parameters and Actual Values Velocity Mode [87])
  - determines the velocity at which the position specified in the *internal target position* is approached (if selected via *target velocity source*).
- Ramps machine data (section Ramps Machine Data [164]) with *motion profile type*, *acceleration time*, *deceleration time*, *quick stop time*, and *ramps reference velocity*
  - determine how to achieve the target velocity and how to decelerate to velocity 0 on approaching target position.
- Position control loop machine data (section Position Control Loop Machine Data [171]) with *position control loop Kp*
  - determine the controller action (gain).

In axis operating mode **profile position mode**, setpoint generator and position controller work together, the speed and current controllers are subordinate to the position controller.

Prerequisite for the proper functioning of axis operating mode **profile position mode** is the correct setting of the speed controller. This is best carried out in axis operating mode **velocity mode**.

Target position, *internal target position*, and the associated control bits as well as *target velocity* can also be changed during an active positioning process; the axis reacts immediately to these changes.

To find out whether or not a certain variable is valid in axis operating mode **profile position mode**, see the corresponding information of the individual variable descriptions in line “Valid: ...”.

5.5.1 Block Mode

In block mode, the *target position* is transferred to the *internal target position* (and thus approached) when a rising edge is recognized in bit “new setpoint” of the *axis control word*.

The block mode is selected via bit “change block immediately” = 0 in the *axis control word*.

Bit “absolute/relative” in the *axis control word* must be set prior to changing the target position. The state of the bit determines if

- the *target position* is accepted as an absolute value replacing the previous *internal target position* (bit “absolute/relative” = 1) or
- the *target position* is to be interpreted as a relative value, in this case it is added to the previous *internal target position* (bit “absolute/relative” = 0).

With value 1, bit “setpoint acknowledgment” in the *axis status word* indicates that the *target position* has been accepted, which allows the removal of the bit “new setpoint”. Bit “setpoint acknowledgment” = 0 then indicates that a new *target position* can be transferred.

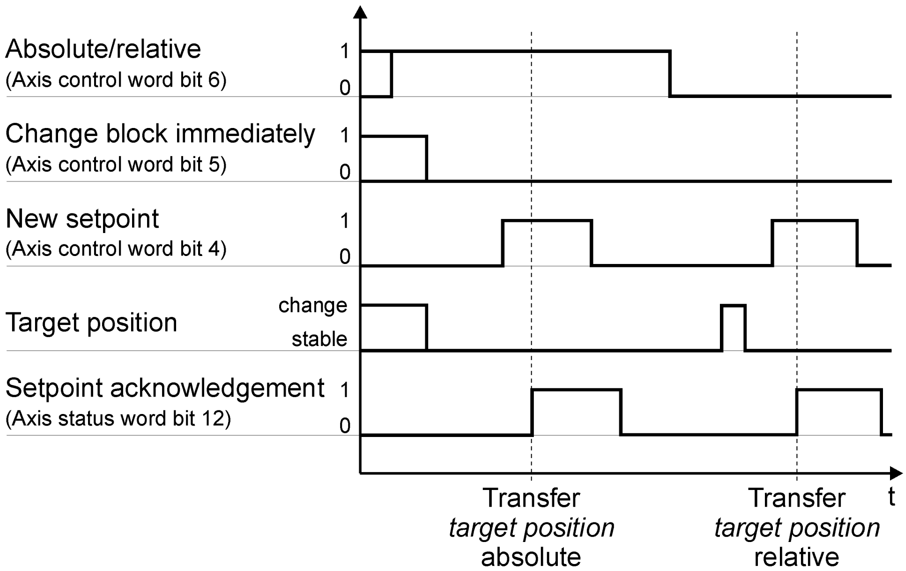


Figure 4: Transfer of *target position* in block mode

With command and commissioning software SPP Windows, the control bits are set automatically when a new *target position* (absolute or relative) is specified via the “set” button.

5.5.2 Setpoint Mode

In setpoint mode the *target position* parameter is constantly queried by the setpoint generator and accepted in the *internal target position*.

The setpoint mode is selected via bits “change block immediately” = 1 and “new setpoint” = 1 in the *axis control word*.

The setpoint mode can only be used with absolute positions, thus bit “absolute/ relative” in the *axis control* word in setpoint mode must always equal value 1.

In setpoint mode, bit “setpoint acknowledgment” in the *axis status word* always has value 1.

This sub-mode is particularly useful for higher-level controllers, as the bits in the *axis control word* do not need to be changed for acceptance of the *new target position*. The device control window of command and commissioning software SPP Windows does not support this sub-mode.

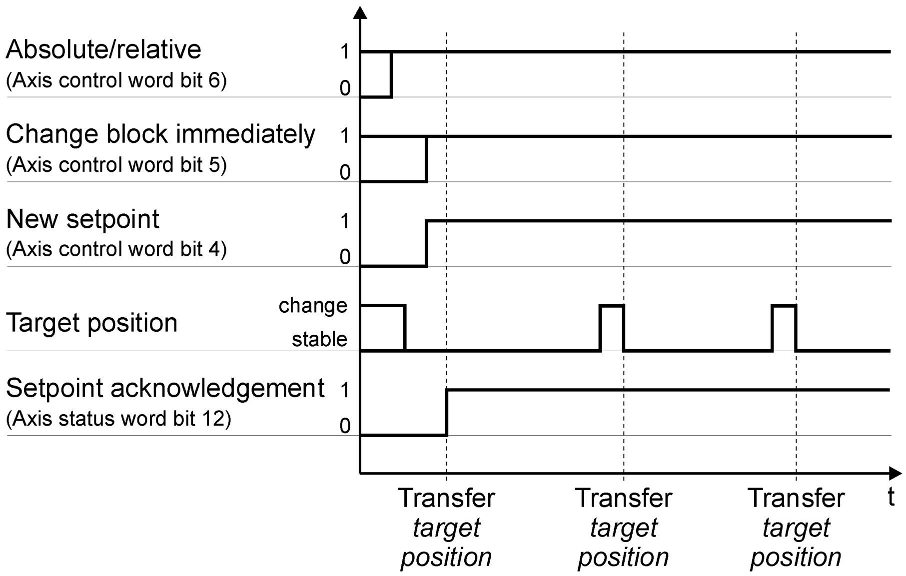


Figure 5: Transfer of the *target position* in setpoint mode

5.6 Homing Mode

In axis operating mode **homing mode**, the home position of the axis can be approached after each switch-on if a motor without multi-turn encoder is used. At this point, the *actual position* of the relevant axis is set to the value specified in the *home offset* variable (standard: 0). This establishes a fixed reference between the *actual position* and the coordinate system of the machine.

The machine home position does not have to correspond to the zero point defined by the application (application zero point for short). The *home offset* specifies the distance between the machine home position and the application zero point. For example, this may be the machine zero point or the workpiece zero point:

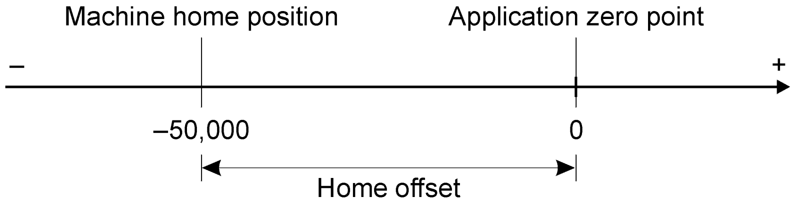


Figure 6: Machine home position and application zero point (example)

If the machine home position is -50,000 (in position units related to the application zero point), as shown in the figure above, -50,000 must be entered for the *home offset* variable. If machine home position and application zero point are located at the same point, 0 is to be entered in *home offset*.

Because the application zero point can be shifted by the *home offset*, this is also referred to as a zero offset.

If the homing switch is exactly on the index pulse to be approached or the next reference signal of the motor position sensor or if the rotor position at the point of the machine home switch has a counter increment (+180 degrees to –180 degrees), different homings can lead to different results. In this case, it is recommended to shift the homing switch.

The *homing selection code* parameter is used to specify how the machine home position is approached (from a positive or negative direction) and how the “pre-pulse” and “limit switch” inputs are evaluated. The tables in the following section show the *homing selection codes* available and the corresponding processes.

The velocity of the **homing mode** is set via the *homing velocity* parameter, the value is multiplied by the *override factor*. Condition for the start of the motion is that the drive is “Ready”.

In **homing mode**, bit 4 in the *axis control word* has the following special meaning:

- start homing mode (bit 4 in the *axis control word*)

Setting the bit “start homing mode” to 1 initiates the homing mode. Resetting the bit to 0 immediately stops homing, i. e. as a rule, this bit must not be reset until the machine home position has been reached.

In SPP Windows, you can start the **homing mode** by selecting axis operating mode **homing mode**, changing to “operation enabled” and operating button “start homing”.

In **homing mode**, bits 12, and 13 in the *axis status word* have the following special meaning:

- home position reached (bit 12 in the *axis status word*)
- homing fault (bit 13 in the *axis status word*)

If bit “home position reached” is set, this indicates that the machine home position has been reached and that the actual position has been set to the value specified in *home offset*. The axis is near the reference signal, the exact position depends on *homing velocity* and ramps.

If bit “homing fault” has value 1, this indicates that a fault has occurred while going to home position. This means that the conditions for the sequence selected via the *homing selection code* have not been fulfilled.

In operating mode **homing mode**, the motions are influenced by the following parameters:

- Ramps machine data (section Ramps Machine Data 164) with *motion profile type*, *acceleration time*, *deceleration time*, *quick stop time*, and *ramps reference velocity*

– determine how to reach the velocity and how to decelerate again.

In this operating mode, setpoint generator and speed controller work together, the current controller is subordinate to the speed controller.

To find out whether or not a certain variable is valid in axis operating mode **homing mode**, see the corresponding information of the individual variable descriptions in line "Valid: ...".

5.6.1 Homing Mode Sequences

If a motor position sensor in multi-turn design is used, going to home position is not required because the position is read from the encoder after each switch-on of the control supply voltage.

In all other cases, a homing must be carried out before the drive can be used for absolute positioning. While going to home position, the axis usually approaches a switch at homing velocity.

The *homing selection code* is used to define if this switch is a limit switch (opener) or a pre-pulse switch (closer) at input "pre-pulse", in which direction going to home position is started (start direction), and in which direction the final position is approached (approach direction). Going to home position is started with bit "start homing mode" in the *axis control word*.

After the machine home switch has been reached, the axis is braked and stopped. It comes to a standstill near, but not exactly on the machine home switch. To approach the application zero after homing, a positioning command must be appended.

- **Selection codes 1 and 2 as well as 17 and 18: approach limit switch**

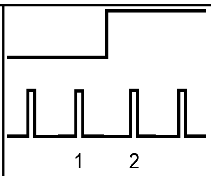
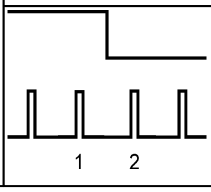
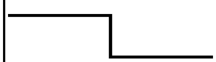
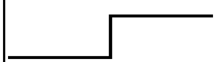
<table>
<tr><th>Homing selection code</th><th>Signal used as switch</th><th>Index pulse used*</th><th>Start direc-tion</th><th>Approach di-rection of fi-nal position</th></tr>
<tr><td>1</td><td>Input "limit switch –"</td><td rowspan="2">the one within the travel range</td><td>–</td><td>+</td></tr>
<tr><td>2</td><td>Input "limit switch +"</td><td>+</td><td>–</td></tr>
<tr><td>17</td><td>Input "limit switch –"</td><td rowspan="2">none</td><td>–</td><td>+</td></tr>
<tr><td>18</td><td>Input "limit switch +"</td><td>+</td><td>–</td></tr>
</table>

* Index pulse (if available) of the motor position sensor used; for absolute encoders final position of homing

A limit switch is approached in positive or negative direction. There, the drive system reverses and leaves the limit switch, again.

With a motor with absolute encoder (resolver, Sincos (Hiperface) encoder, or EnDat encoder), the axis is stopped and the actual position of the axis is set to the absolute position of the encoder under consideration of the home offset. To determine the exact position of a motor with incremental encoder, with selection codes 1 and 2 the next index pulse or the next reference signal of the motor position sensor is approached and the actual position value is set to the home offset. With selection codes 17 and 18, the reference point at the switch edge is determined with a time inaccuracy of 1 ms, to minimize the error, the homing velocity should be low.

- **Selection codes 3 to 6 as well as 19 to 22: approach pre-pulse switch (partly closed, partly opened)**

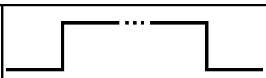
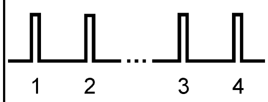
Homing selection code	Signal used as switch			Index pulse used*	Start direction	Approach direction of final position
3	Input “pre-pulse”		pre-pulse switch	1	depending on the position of the axis	–
4			zero pulses*	2		+
5			pre-pulse switch	2	depending on the position of the axis	+
6			zero pulses*	1		–
19	Input “pre-pulse”		pre-pulse switch	none	depending on the position of the axis	–
20						+
21			pre-pulse switch		depending on the position of the axis	+
22						–

* Index pulse (if available) of the motor position sensor used; for absolute encoders final position of homing

A pre-pulse switch is approached in positive or negative direction. The pre-pulse switch is closed on one part of the way, on the other part, it is opened so that the start direction can be derived from the position of the axis. If required, the drive system reverses at the pre-pulse switch in order to reach the desired final position, the index pulse, or the reference signal in front of or behind the switching edge.

With a drive system with absolute encoder (resolver, Sincos (Hiperface) encoder, or EnDat encoder), the axis is stopped and the actual position of the axis is set to the absolute position of the encoder under consideration of the home offset. To determine the exact position of a drive system with incremental encoder, with selection codes 3 to 6 the next index pulse or the next reference signal of the motor position sensor used (if available) is approached and the actual position value is set to the home offset. With selection codes 19 to 22, the reference point at the switch edge is determined with a time inaccuracy of 1 ms, to minimize the error, the homing velocity should be low.

- **Selection codes 7 to 14 as well as 23 to 30: approach pre-pulse switch (closed on one part of the way, only)**

Homing selection code	Signal used as switch			Index pulse used*	Start direction	Approach direction of final position
7	input "pre-pulse"		pre-pulse switch	1	+**	–
8			index pulses*	2		+
10				4		+
11		Between index pulses 2 and 3, further index pulses may occur. These are not considered.		4	–**	+
12				3		–
14				1		–

23	input "pre-pulse"		pre-pulse switch	keiner	+**	–
24						+
26						+
27					–**	+
28						–
30						–

* Index pulse (if available) of the motor position sensor used; for absolute encoders final position of homing

** If the pre-pulse input is active during the start of homing, the drive system will start into the other direction, if required.

A pre-pulse switch is approached in positive or negative direction. The pre-pulse switch is closed on a part of the way, otherwise, it is opened. The limit switches are used as reversing switches if the sequence requires that.

If required, the drive system reverses at the pre-pulse switch in order to reach the desired final position, the index pulse, or the reference signal in front of or behind the switching edge. With a drive system with absolute encoder (resolver, Sincos (Hiperface) encoder, or EnDat encoder), the axis is stopped and the actual position of the axis is set to the absolute position of the encoder under consideration of the home offset. To determine the exact position of a drive system with incremental encoder, with selection codes 7 to 14 the next index pulse or the next reference signal of the motor position sensor used (if available) is approached. There, the actual position value is set to the home offset. With selection codes 23 to 30, the reference point at the switch edge is determined with a time inaccuracy of 1 ms, to minimize the error, the homing velocity should be low.

Homing speeds and brake ramps must be selected in a way that the axis comes to a standstill within the pre-pulse edges. Otherwise, the drive may not reverse and take the wrong index pulse or the wrong pre-pulse edge as reference.

- **Selection codes –1 and –2 as well as 33 and 34: accept absolute position (within one revolution) or evaluate the first index pulse in start direction**

Homing selection code	Signal used as switch	Index pulse used*	Start direction	Approach direction of final position
–1	none	the first one in start direction	+	+
–2			–	–
33			+	+
34			–	–

* Index pulse (if available) of the position sensor used

With these selection codes, the axis does not approach a pre-pulse or limit switch. With a drive system with absolute encoder (resolver, Sincos (Hiperface) encoder, or EnDat encoder), the actual position of the axis is set to the absolute position of the encoder under consideration of the home offset, i. e. the axis is not moved. With a drive system with incremental encoder, the first index pulse in start direction is evaluated and there, the actual position is set to the home offset.

- **Selection codes –3 and –4: evaluate the first positive pre-pulse edge in start direction**

Homing selection code	Signal used as switch	Index pulse used*	Start direction	Approach direction of the final position
–3	Input “pre-pulse”	none	+	+
–4	Input “pre-pulse”		–	–

With these selection codes, the first positive pre-pulse edge in start direction is evaluated. At the pre-pulse switch, the drive system does not reverse, the axis is simply stopped. Index pulses are not considered, the machine home position is determined with a time inaccuracy of 1 ms. For minimizing the fault, a low homing velocity should be selected.

- **Selection codes –5 and –6: approach limit switch**

Homing selection code	Signal used as switch	Index pulse used	Start direction	Approach direction of the final position
–5	Input “limit switch –“	none	–	–
–6	Input “limit switch +“		+	+

A limit switch is approached in positive or negative direction. There, the drive system reverses and leaves the limit switch. After that, the axis is stopped. Index pulses are not considered, the machine home position is determined with a time inaccuracy of 1 ms. For minimizing the fault, a low homing velocity should be selected.

5.7 Electronic Gearing

In axis operating mode **electronic gearing**, the axis follows an externally set position. The position is set via an external position sensor (input encoder signals). This is only true for servo drive systems equipped with an encoder signal input.

In axis operating mode **electronic gearing**, position and velocity of the axis follow the settings of this interface which is called master axis. The axis which is in axis operating mode **electronic gearing** is called the slave axis (or gear axis).

The master axis can be a different servo drive system or just a position measuring system (position sensor), moved either by a drive system in the machine or manually.

During the change from another axis operating mode into the **electronic gearing** operating mode, input “synchronize” or the control bit of the same name in the *axis control word* are used to align the coordinate systems of master and slave axis. As soon as the drive is “ready”, the slave axis moves according to the settings of the master axis.

- With TrioDrive D/xS, MidiDrive D/xS, TrioDrive D, and MidiDrive D, the position information is determined from the external position sensor directly after synchronization. This achieves an exact incremental synchronization.
- With MaxiDrive, the position information of the master axis is processed via the software, in this case, processing times of up to 1 ms may occur. Instead of using input “synchronize”, an exact incremental synchronization can also be initiated via input “index pulse +/–“ (index pulse input of the incremental encoder module, option L1/L2).

The acceleration of the slave axis takes into account the ramp time of the electronic gearing, the lead of the master axis is made up for by increasing the velocity.

In axis operating mode **electronic gearing**, bit 4 of the *axis control word* has the following meaning:

- synchronize (bit 4 in the *axis control word*).

This bit synchronizes the actual position of master and slave axis: during transition from 0 to 1 for this bit, the position deviation of the slave axis is set to 0. The slave axis then follows all subsequent position changes of the master axis, taking into account the gear ratio of the electronic gearing.

In axis operating mode **electronic gearing**, bits 10 and 12 in the *axis status word* have the following special meaning:

- synchronized (bit 10 in the *axis status word*)
- velocity = 0 (bit 12 in the *axis status word*)

With value 1, bit “synchronized” indicates that the drive of the slave axis has reached the position specified by the master axis. For that, the *positioning window* and *positioning window time* parameters are taken into account. This message can also be output on a digital output, see *signal selection digital output ...* in section Input/Output Function Machine Data 131.

- If a synchronized state is not or rarely reached, check the following:
  - Is the *positioning window* large enough?
  - Is the resolution of the external position sensor set correctly? (*Position sensor resolution*)
  - Is speed feedforward switched on? (*speed control loop Kv electronic gearing* = 1)
  - Can the slave axis reach the necessary speed at all?

If bit “velocity = 0” is set, this shows that, after deceleration, the drive system has reached velocity zero, for that, the *velocity threshold* and *velocity threshold time* parameters are taken into account.

In axis operating mode **electronic gearing**, the following machine data are effective for the slave axis:

- Electronic gearing machine data (section Electronic Gearing Machine Data 176) with *external position sensor resolution*, *gear ratio*, *speed control loop Kv electronic gearing*, *ramp time*
  - determine the position sensor resolution of the master axis, the gear ratio between the master and the slave axis, the force of the velocity feedforward (to compensate for any remaining control offset), and the slope during the synchronization process.
- Position control loop machine data (section Position Control Loop Machine Data 171) with *position control loop Kp*
  - determine the controller behavior (gain).
- Speed control loop machine data (section Speed Control Loop Machine Data 167) with *speed filter cut-off frequency*, *speed control loop Kp*, *speed control loop Ki*, *speed control loop total amplification*
  - determine the controller action: smoothing of the actual speed, gain, and optional I component (to compensate for any remaining control offset).

In axis operating mode **electronic gearing**, setpoint generator and position controller work together, the speed and current controllers are subordinate to the position controller.

To find out whether or not a certain variable is valid in axis operating mode **electronic gearing**, see the corresponding information of the individual variable descriptions in line "Valid: ...".

5.8 Flying Shear

Axis operating mode **flying shear** is a special kind of axis synchronization (based on operating mode **electronic gearing**). It is used when a continuous material flow, which is primarily transported by a master axis at a constant velocity, is to be processed during movement (sawn, cut, printed etc.). For this purpose, during a specific phase in the overall operation, the slave axis runs synchronously with the master axis before returning to its starting position.

The positions at which the material is processed are called synchronizing points. The first synchronizing point is set when the flying shear mode is started. All other synchronizing points are a *cutting length* distance apart.

The position of the master axis is set via an *external position sensor* (input encoder signals). This is only true for servo drives equipped with an encoder signal input.

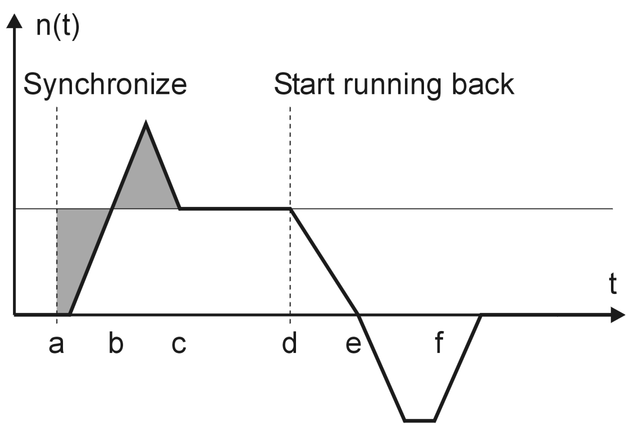


Figure 7: **Flying Shear**, Velocity Time Diagram

The flying shear is started via input "enable saw". Synchronization begins automatically at the event "synchronizing point reached" (time a in figure 7). The acceleration of the slave axis takes into account the ramp time of the electronic gearing, the lead of the master axis is made up for by increasing the velocity (b). As soon as the slave axis is synchronized with the master axis, signal "synchronized / apply saw" is generated (c). This message can also be output on a digital output, see *signal selection digital output ...* in section Input/Output Function Machine Data 131. The material can now be processed.

The return positioning of the **flying shear** is initiated (d) via input “start running back”. The velocity is initially reduced to zero (e) via the deceleration ramp and then accelerated to the return velocity through the usual acceleration ramp. The **flying shear** is braked (f) at the starting position via the deceleration ramp and restarted from there when the next synchronizing point is reached (= old synchronizing point + *cutting length*).

Alternatively, each synchronization can be started via input “flying shear external synchronization”, e. g. for material with different length or distance. Prerequisite for that is the enabling of the external synchronization via the corresponding input.

The return velocity is specified via the *target velocity (digital)* parameter or the “setpoint” analog input, depending on the setting in the machine data *target velocity source*.

Changing the cutting length is not possible during operation.

In axis operating mode **flying shear**, the *axis control word* does not contain any operating mode-dependent bits.

In axis operating mode **flying shear**, bit 10 in the *axis status word* has the following special meaning:

- position setpoint reached (bit 10 in the *axis status word*).

With value 1, this bit indicates that the axis has completed the return positioning. For that, the *positioning window* and *positioning window time* parameters are taken into account. This message can also be output on a digital output, see *signal selection digital output ...* in section Input/Output Function Machine Data 131.

In axis operating mode **flying shear**, the following parameters are effective for the slave axis:

- Flying shear machine data (section Flying Shear Machine Data 180) with *cutting length*, *master velocity threshold*
  – determine the distance between two synchronizing points (in PU) and the minimum velocity of the master axis.
- Electronic gearing machine data (section Electronic Gearing Machine Data 176) with *external position sensor resolution*, *gear ratio*, *speed control loop Kv*, *electronic gearing*, *ramp time*
  – determine the position sensor resolution of the master axis, the gear ratio between the master and the slave axis, the force of the velocity feedforward (to compensate for any remaining control offset), and the slope during the synchronization process.
- Position control loop machine data (section Position Control Loop Machine Data 171) with *position control loop Kp*
  – determine the controller action (gain).
- Speed control loop machine data (section Speed Control Loop Machine Data 167) with *speed filter cut-off frequency*, *speed control loop Kp*, *speed control loop Ki*, *speed control loop total amplification*

– determine the controller action: smoothing of the actual speed, gain, and optional I component (to compensate for any remaining control offset).

- *Target velocity* (section Parameters and Actual Values Velocity Mode 87)

– determines the return velocity (if selected via the *target velocity source*).

- Ramps machine data (section Ramps Machine Data 164) with *motion profile type*, *acceleration time*, *deceleration time*, *quick stop time*, and *ramps reference velocity*

– determine how to reach the return velocity and decelerate again at the starting position.

In axis operating mode **flying shear**, setpoint generator and position controller work together, the speed and current controllers are subordinate to the position controller.

To find out whether or not a certain variable is valid in axis operating mode **electronic gearing**, see the corresponding information of the individual variable descriptions in line "Valid: ...".

5.9 Velocity Profile

With axis operating mode **velocity profile**, a defined velocity time sequence can be run. For each section of the profile, the acceleration or deceleration time as well as the time with constant velocity is stated in milliseconds.

This axis operating mode can only be called up in program mode via the part program. For further information see operating instructions 6710.231 "Part Program".

In axis operating mode **velocity profile**, *axis control word* and *axis status word* do not contain any operating mode-dependent bits.

5.10 Timed Positioning Mode

With axis operating mode **timed positioning mode**, relative target positions can be approached in specified times (point-to-point positioning). For the movement, a variety of curve forms in accordance with VDI manual Technology of Mechanism Design I: Transmission with Varying Velocity Ratio (VDI 2143, Blatt 1, Kapitel 6.1.1 "Normierte symmetrische Bewegungsgesetze für Rast-in-Rast (R-R))" is available.

The relative target position is defined via parameter *position distance single movement*, the duration of the movement via *position time single movement*. The target velocity is not specified, it results for a certain distance and a defined positioning time from the *curve form* by which the driving behavior is specified individually for each positioning:

Curve form	Advantages	Disadvantages	C_v	C_a	C_j	$C_{M\,dyn}$
–1: Linear (VDI 2143: “square parabola”)	lowest C_a value, i. e. lowest inertia forces	not jerk-free (3 jerks), as a result e. g. vibrations, noises, wear and tear	2	**4**	∞	8
–2: Modified acceleration trapezoid	particularly low C_a value, low inertia forces	high C_j and $C_{M\,dyn}$ values	2	**4.89**	61.4	8.09
–3: Modified sine	particularly low C_v and $C_{M\,dyn}$ value, low C_a value, suitable for high velocities	high C_j value	**1.76**	5.53	69.5	**5.4**
–4: Inclined sine	particularly low C_j value, low-vibration, suitable for high velocities	high $C_{M\,dyn}$ value	2	6.28	**39.5**	8.16
–5: 5th-grade polynomial	low C_v, C_a, and $C_{M\,dyn}$ value, suitable for high velocities	high C_j value	**1.88**	5.78	60	6.69

The dynamic velocity characteristic value C_v (maximum value of the standardized velocity) is a measure for the maximum velocity reached during positioning. For a positioning at low velocity, a low C_v value must be reached.

The dynamic acceleration characteristic value C_a (maximum value of the standardized acceleration) describes the maximum acceleration, that means the desired torque or force. A higher value can mean that a more powerful motor is required. (Example: a C_a value of 5.78 results in a torque or force requirement 44.5% higher than for a linear curve form with $C_a = 4$.)

The dynamic jerk characteristic value C_j (maximum value of the standardized jerk function) reflects acceleration changes. Higher values can lead to vibrations, noises, and wear and tear.

The drive torque or drive force due to dynamic inertia forces depends on the dynamic torque characteristic value $C_{M\,dyn}$ (maximum value of the product of velocity(z) × acceleration(z)). The lowest possible value is desirable with regard to low drive system costs.

The positioning is started with digital input “start positioning”, output “in position” reports the terminated positioning.

The operation can be configured as desired using the positioning mode functions at the digital inputs:

- Positioning mode bit 0: positioning direction

  0: clockwise / forward
  1: counterclockwise / backward

- Positioning mode bit 1:

  0: single movement
  1: back and forth movement (total duration = 2 × positioning time)

- Positioning mode bit 2 (reserved)
- Positioning mode bit 3:

  0: single step
  1: continuous movement (the movement will be stopped after input “start positioning” has been set to 0 and the target position has been reached)

The functions and signals can be mapped on physical or logical inputs and outputs.

During **timed positioning mode**, the motion sequence is influenced by the following parameters:

- Position distance single movement, position time single movement, and curve form
  - define the positioning sequence, a change of these values does not have any effect on a running positioning
- Position control loop machine data with *position control loop Kp*
  - define the drive behavior (amplification)

In operating mode **timed positioning mode**, setpoint generator and position controller work together, speed and current controller are subordinate to the position controller. From firmware version V 8.5.9 on, the feedforward velocity can be influenced or switched off using the feedforward velocity factor.

Prerequisite for a trouble-free functioning of the **timed positioning mode** is the correct setting of the speed control loop.

5.11 Cyclic Synchronous Torque/Force Mode

Axis operating mode **cyclic synchronous torque mode** (according to CiA® 402; corresponds to a force mode for the linear motor) serves for operating one or several axes at a higher-lever controller sending torque or force setpoints to the drive at defined intervals. The control of velocity and position, if applicable, takes place in the higher-level controller.

In **cyclic synchronous torque mode**, the PDOs are transmitted cyclically, with CANopen® synchronously to a SYNC message. In case of a fault or a breakdown of the SYNC transmission, the drive keeps the latest setpoint until it receives new data or a new SYNC message.

In **cyclic synchronous torque mode**, bit 12 in the *axis status word* has the following special meaning:

- Setpoints are accepted

With value 1, this bit indicates that the drive accepts the specified setpoints. (The axis state machine must be in state “operation enabled” so that the setpoints can lead to a movement.)

In **cyclic synchronous torque mode**, the *axis control word* does not contain any operating mode-dependent bits.

In operating mode **cyclic synchronous torque mode**, the fieldbus communication is influenced by the following parameters:

- The mapping PDOs (PDO Mapping)
  - serve for the transmission of axis status word and axis control word as well as the setpoints (torque or force setpoint as well as feedforward acceleration) and actual velocity and position values.
- The machine data PDOs (Comm. Parameter, CANopen® only)
  - control the synchronous transmission of above-mentioned PDOs (value n of *transmission type* = 1).

For variable PDO mapping, the objects

- *Axis control word* (index 6040)
- *Torque setpoint / force setpoint* (index 6071)

must be mapped in an RPDO. Object

- *Torque offset / force offset* (index 60B2)

can be mapped in an RPDO for feedforward torque or force. Objects

- *Axis status word* (index 6041)
- *Actual velocity* (index 606C)
- *Position sensor actual position* (index 6063)

can be mapped in a TPDO.

In axis operating mode **cyclic synchronous torque mode**, only the current control loop is active, see the following diagram.

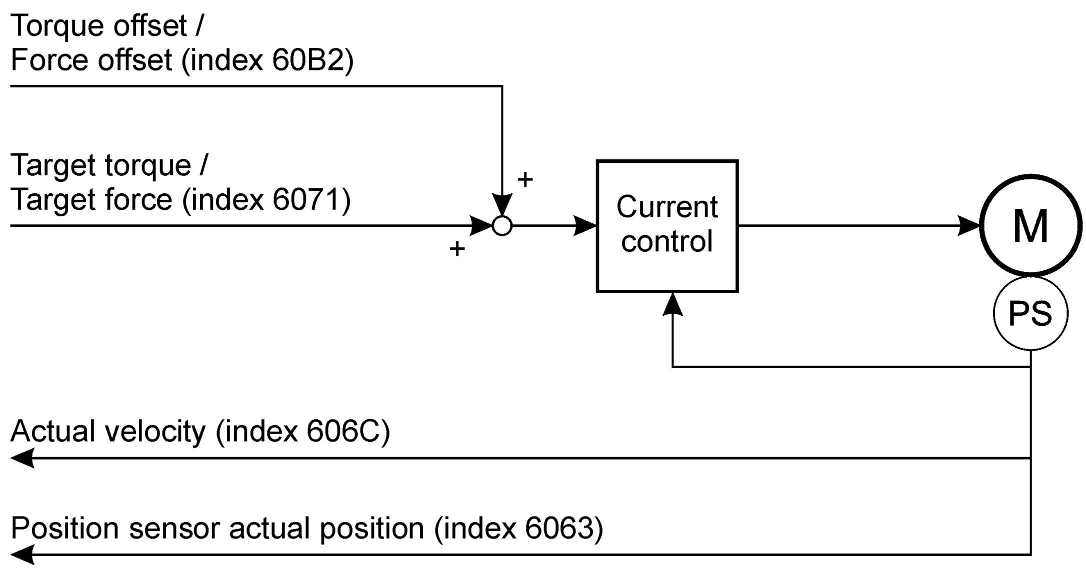


Figure 8: Cyclic synchronous torque mode with all cyclic-synchronously transmitted objects (simplified block diagram)

In axis operating mode **cyclic synchronous torque mode**, the servo drives can be operated with controllers of various manufacturers. If required, please contact ESR.

5.12 Cyclic Synchronous Velocity Mode

Axis operating mode **cyclic synchronous velocity mode** (according to IEC 61800-7-2) serves for operating one or several axes at a higher-level controller sending velocity setpoints to the drive at defined time intervals. The position control takes place in the higher-level controller.

In **cyclic synchronous velocity mode**, the PDOs are transmitted cyclically, with CANopen® synchronously to a SYNC message. In case of a fault or a breakdown of the SYNC transmission, the drive keeps the latest setpoint until it receives new data or a new SYNC message.

In **cyclic synchronous velocity mode**, bit 12 in the *axis status word* has the following special meaning:

- Setpoints are accepted

  With value 1, this bit indicates that the drive accepts the specified setpoints. (The axis state machine must be in state “operation enabled” so that the setpoints can lead to a movement.)

In **cyclic synchronous velocity mode**, the *axis control word* does not contain any operating mode-dependent bits.

In operating mode **cyclic synchronous velocity mode**, the fieldbus communication is influenced by the following parameters:

- The mapping PDOs (PDO Mapping)
  - serve for the transmission of axis status word and axis control word as well as the setpoints (velocity setpoint, feedforward velocity and acceleration) and actual velocity and position values.
- The machine data PDOs (Comm. Parameter, CANopen® only)
  - control the synchronous transmission of above-mentioned PDOs (value n of *transmission type* = 1).

For variable PDO mapping, the objects

- *Axis control word* (index 6040)
- *Velocity setpoint* (index 60FF)

must be mapped in an RPDO. Objects

- *Velocity offset* (index 60B1)
- *Torque offset / force offset* (index 60B2)

can be mapped in an RPDO for feedforward torque or force or feedforward velocity. Objects

- *Axis status word* (index 6041)
- *Actual velocity* (index 606C)
- *Position sensor actual position* (index 6063)

can be mapped in a TPDO. Feedforward velocity can be influenced or switched off with the *feedforward velocity factor*.

In axis operating mode **cyclic synchronous velocity mode**, current control loop and speed control loop are active, the current control loop is subordinate to the speed control loop. See the following diagram.

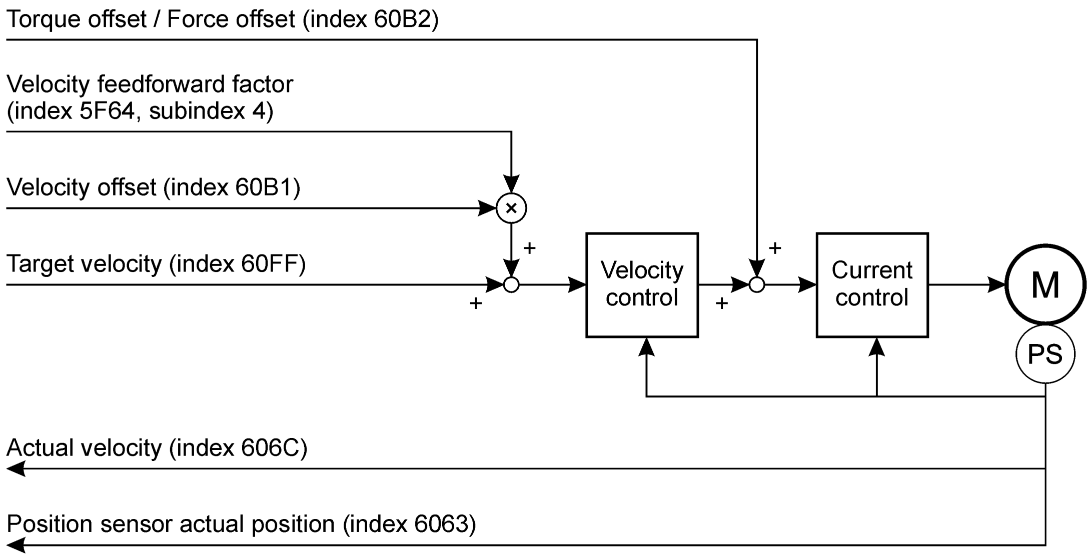


Figure 9: Cyclic synchronous velocity mode with all cyclic-synchronously transmitted objects (simplified block diagram)

In axis operating mode **cyclic synchronous velocity mode**, the servo drives can be operated with controllers of various manufacturers. If required, please contact ESR.

5.13 Cyclic Synchronous Position Mode

Axis operating mode **cyclic synchronous position mode** serves for operating one or several axes for which a time interpolation of the position setpoints is required (e. g. for contouring control). For that, the higher-level controller sends target positions to the drive at defined intervals via an RPDO. In addition to these positions, the fine interpolator of the drive calculates further target positions at the time interval of the position control loop (1 ms). In **cyclic synchronous position mode**, the target positions are given in position sensor steps (PSS), only, not in position units (pU).

In **cyclic synchronous position mode**, the PDOs are transmitted cyclically, with CANopen® synchronously to a SYNC message. In case of a failure or a breakdown of the SYNC transmission, the drive keeps the last target position until it receives new data or a new SYNC message

In **cyclic synchronous position mode**, bit 12 in the *axis status word* has the following special meaning:

- Setpoints are accepted

  With value 1, this bit indicates that the drive accepts the specified setpoints. (The axis state machine must be in state “operation enabled” so that the setpoints can lead to a movement.)

The axis operating mode can only be changed when the state of the axis state machine is “switch on disabled”, “ready to switch on”, or “switched on”.

Only linear axes are supported in the **cyclic synchronous position mode**, an operation as circular axis is not possible. Variables *Position Limit Min* and *Position Limit Max* are not taken into account. For defining the target positions, the complete value range (-2^{32} .. $2^{32}-1$) is available. An operation as endless axis (beyond value range) is possible.

In operating mode **cyclic synchronous position mode**, the fieldbus communication is influenced by the following parameters:

- Mapping PDOs
  – serve for the transmission of axis status word and axis control word as well as setpoints (target position and feedforward) and actual positions and velocities.
- Machine data PDOs (comm. parameters, CANopen®, only)
  – control the synchronous transmission of the above-mentioned mapping PDOs (value n of the *transmission type* = 1).
- Machine data interpolated position mode with interpolation time period
  – determine the time interval for setting the target positions in milliseconds.

For a description of these parameters, see operating instructions 6745.205 “CANopen® Interface” or 6745.232 “EtherCAT Interface”.

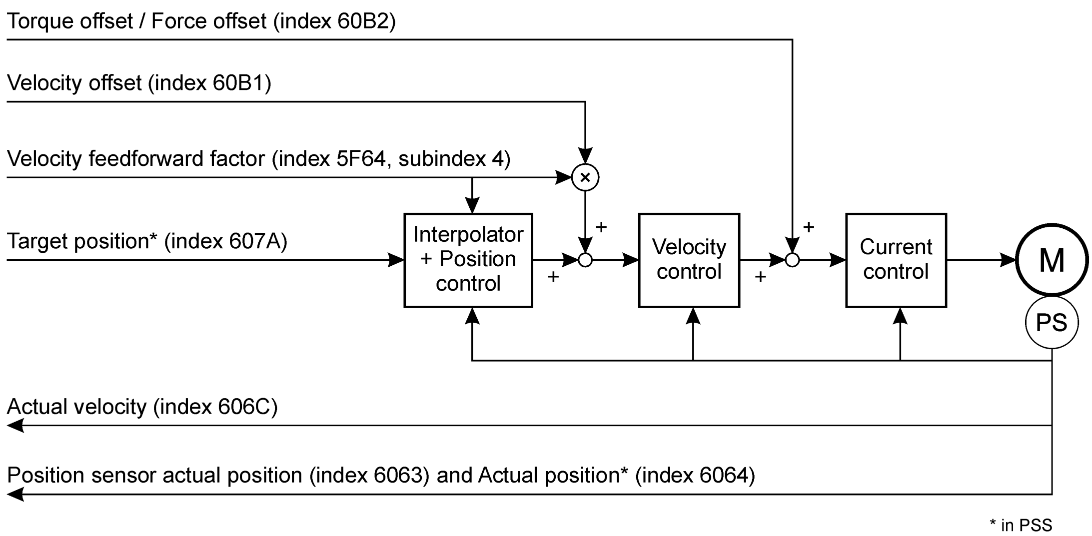


Figure 10: Cyclic synchronous position mode with all cyclic-synchronously transmitted objects (simplified block diagram)

For variable PDO mapping, the objects

- *Axis control word* (index 6040)
- *Target position* (index 607A)

must be mapped in an RPDO. Objects

- *Torque offset / force offset* (index 60B2)
- *Velocity offset* (index 60B1)

can be mapped in an RPDO for feedforward torque or force or feedforward velocity. Objects

- *Axis status word* (index 6041)
- *Position sensor actual position* (index 6063)
- *Actual velocity* (index 606C)

can be mapped in a TPDO. If the *velocity offset* is not received via RPDO, the interpolation calculates the feedforward velocity from the target position of variable *target position*. Feedforward velocity can be influenced or switched off with the *feedforward velocity factor*.

In axis operating mode **cyclic synchronous position mode**, the fine interpolator as setpoint generator acts together with the position control, speed and current control are subordinate to the position control.

In axis operating mode **cyclic synchronous position mode**, the servo drives can be operated with controllers of various manufacturers. If required, please contact ESR.

5.13.1 Activating Cyclic Synchronous Position Mode

As a prerequisite for controlling the axis state machine, the corresponding PDOs must be activated.

1. If not set via machine date *axis operating mode at startup*, select axis operating mode **cyclic synchronous position mode**.
2. By enabling the PDO communication by the master, the state machine of the NMT services or the ESM state machine is switched to “operational”.
3. For the initialization of the **cyclic synchronous position mode**, read out the current actual position and write it into variable *target position*.
4. Switch the axis state machine to “operation enabled”.

5.13.2 Deactivating Cyclic Synchronous Position Mode

For deactivating the **cyclic synchronous position mode**, the axis state machine has to be switched to state “switch on disabled”, “ready to switch on”, or “switched on”. With CANopen®, if required, switch the NMT state machine to “pre-operational”.

After that, the axis operating mode can be changed.

5.14 Interpolated Position Mode

Axis operating mode **interpolated position mode** serves for operating one or several axes for which a time interpolation of the position setpoints is required (e. g. for contouring control). For that, the higher-level controller sends target positions to the drive at defined intervals via an RPDO. In addition to these positions, the fine interpolator of the drive calculates further target positions at the time interval of the position control loop (1 ms). In **interpolated position mode**, target positions are transmitted in position sensor steps (PSS), only, not in length units (LU).

In **interpolated position mode**, the PDOs are transmitted cyclically, with CANopen® synchronously to a SYNC message. In case of a failure or a breakdown of the SYNC transmission, the drive keeps the last target position until it receives new data or a new SYNC message

For devices with firmware versions lower than V 8.5.9, monitoring of hardware limit switches and function “measure position” are not active in operating mode **interpolated position mode**.

For devices with firmware versions lower than V 8.5.9, operating mode-related bits do not exist for the **interpolated position mode** in the *axis status word*. From firmware version V 8.5.9 on, bit 12 in the *axis status word* has the following special meaning in **interpolated position mode**:

- Setpoints are accepted

  With value 1, this bit indicates that the drive accepts the specified setpoints. (The axis state machine must be in state “operation enabled” and the interpolation must be enabled via bit 4 in the *axis control word* so that the setpoints can lead to a movement.)

In the *axis control word*, bit 4 has the following special meaning:

- Enable interpolation

  The interpolation is started by setting bit “enable interpolation”.

The axis operating mode can only be changed when the state of the axis state machine is “switch on disabled”, “ready to switch on”, or “switched on”.

Only linear axes are supported in the **interpolated position mode**, an operation as circular axis is not possible. From firmware version V 8.5.9 on, variables *position limit min* and *position limit max* are not taken into account. In this case, for defining the target positions, the complete value range (-2^{32} .. $2^{32}-1$) is available. An operation as endless axis (beyond value range) is possible.

In operating mode **interpolated position mode**, the fieldbus communication is influenced by the following parameters:

- Mapping PDOs
  - serve for the transmission of axis status word and axis control word as well as target positions (in *interpolation data record*) and actual positions.
- Machine data PDOs (comm. parameters, CANopen®, only)
  - control the synchronous transmission of the above-mentioned mapping PDOs (value n of the *transmission type* = 1).
- Machine data interpolated position mode with *interpolation time period*
  - determine the time interval for setting the target positions in milliseconds.

For a description of these parameters, see operating instructions 6745.205 “CANopen® Interface” or 6745.232 “EtherCAT Interface”.

For variable PDO mapping, the objects

- *Axis control word* (index 6040)
- *Target position* of the *interpolation data record* (index 60C1, subindex 1)

must be mapped in an RPDO. Objects

- *Torque offset / force offset* (index 60B2, in firmware version V 8.5.9 and higher)
- *Velocity offset* (index 60B1)

can be mapped in an RPDO for feedforward torque or force or feedforward velocity. Objects

- *Axis status word* (index 6041)
- *Position sensor actual position* (index 6063)
- *Actual velocity* (index 606C, in firmware version V 8.5.9 and higher updated synchronously)

can be mapped in a TPDO. If the *velocity offset* is not received via RPDO, the interpolation calculates the feedforward velocity from the target position of the *interpolation data record*. Feedforward velocity can be influenced or switched off with the *feedforward velocity factor*.

In axis operating mode **interpolated position mode**, the fine interpolator as setpoint generator acts together with the position control, speed and current control are subordinate to the position control. See the following block diagram.

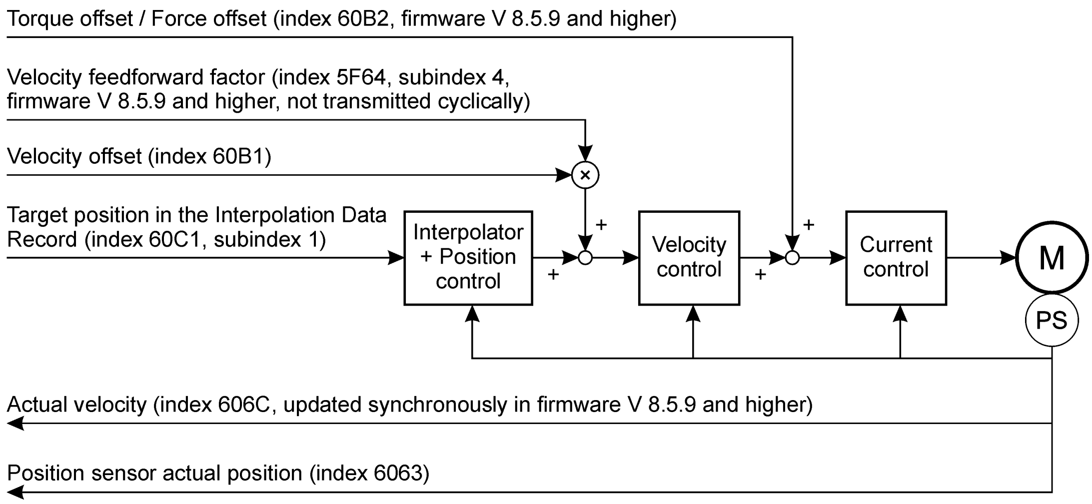


Figure 11: Interpolated position mode with all cyclic-synchronously transmitted objects (simplified block diagram)

In axis operating mode **interpolated position mode**, the servo drives can be operated with controllers of various manufacturers. If required, please contact ESR.

5.14.1 Activating Interpolated Position Mode

As a prerequisite for controlling the axis state machine, the corresponding PDOs must be activated.

1. If not set via machine date *axis operating mode at startup*, select axis operating mode **interpolated position mode**.
2. By enabling the PDO communication by the master, the state machine of the NMT services or the ESM state machine is switched to “operational”.
3. For the initialization of the **interpolated position mode**, read out the current actual position and write it into the *interpolation data record*.
4. Switch the axis state machine to “operation enabled”.
5. Activate the interpolation via “enable interpolation” (bit 4 in the axis control word).

5.14.2 Deactivating Interpolated Position Mode

For deactivating the **interpolated position mode**, the axis state machine has to be switched to state “switch on disabled”, “ready to switch on”, or “switched on”. With CANopen®, if required, switch the NMT state machine to “pre-operational”.

After that, the axis operating mode can be changed.

6 Device Operating Modes: Command Mode and Program Mode

With the device control, the way setpoints for the drive system functions are set can be selected via the device operating mode. There are two device operating modes:

- **Command mode**

  Device operating mode **command mode** permits a direct control of axis movements via single commands by a higher-level controller (PLC, PC). Commands and setpoints can be set via the communication interfaces (serial interface, field bus) or the terminals (e. g. analog input setpoint for velocity and torque setting).

  In **command mode**, *axis operating mode* can be used for selecting whether e. g. velocities or positions are to be preset. The axis operating modes are described in detail in section Axis Operating Modes 29.

  In this case, the *axis control word* defines the way the drive system reacts on setpoint changes. For further information on the *axis control word*, see section Axis Control Word 20.

- **Program mode** (automatic or single step)

  With device operating mode **program mode**, special programs, so-called part programs, can be run on the servo drive. These part programs can be used to run motion sequences independent of a higher-level controller. Via special commands or digital inputs/outputs, programs of a higher-level controller and part programs can communicate and synchronize their sequences. For further information on the part program, see operating instructions 6710.231 “Part Program”.

The device operating mode may only be changed in states “switch on disabled”, “ready to switch on” and “switched on”.

7 Input, Output, Trigger, Measuring, and Trace Functions

This section summarizes all functions which are not necessarily required for drive system movements in normal operation:

The **input/output function** permits communication with other field devices or controllers via inputs/outputs at the front panel and so-called logical inputs/outputs.

Via the **switching points**, the digital outputs can be operated depending on the actual position.

Trigger and measuring functions permit e. g. a measurement of the actual position via a fast input.

The **trace function** can be used for recording setpoints and actual values as well as internal data in the drive system for a subsequent read-out and further processing. These trace functions are used e. g. by the oscilloscope function of command and commissioning software SPP Windows.

7.1 Input/Output Function

7.1.1 Digital Input/Output

The digital input/output function provides access to the I x.x and O x.x inputs and outputs of the servo drive.

These inputs and outputs can be used freely. They can be queried by a higher-level controller via the part program or the communication interfaces.

The following types of inputs and outputs exist:

- **Physical inputs/outputs**

  These are inputs and outputs visibly located at the corresponding connector of the servo drive that can be connected to a signal line. With TrioDrive D/xS, MidiDrive D/xS, TrioDrive D, and MidiDrive D, these inputs are located at connector X7, with MaxiDrive at connectors X4 and X5.

- **Logical inputs/outputs**

  These are invisible internal inputs and outputs that can be set arbitrarily or evaluated by the controller in case of a communication via field bus.

- **Linked inputs and outputs**

  These are invisible internal inputs and outputs connected with each other. If an output is set or deleted, this becomes visible at the correspondingly linked input. Linked inputs/outputs are e. g. used as markers in part programs or for calling up a function automatically via an input using an output signal.

Input	TrioDrive D/xS, MidiDrive D/xS, TrioDrive D, MidiDrive D	MaxiDrive	For free use	Drive-specific assignment
I 1.0 .. I 1.7	physical	physical	×	×
I 2.0 .. I 2.7	logical	physical	×	–
I 3	logical*	logical*	×	I 3.0**
I 4	logical	logical	×	–
I 9.0 .. I 9.7	logical	logical	×	×
I 10, I 11***	logical	logical	×	–
I 15	logical	logical	×	×
I 16 .. I 20	logical	logical	×	–

* I 3.0 = physical = input “enable”

** With TrioDrive D/xS and MidiDrive D/xS, inputs I 3.4 to I 3.7 have a drive-specific assignment, too

*** Inputs I 10 and I 11 are reserved for querying the axis status word.

Output	TrioDrive D/xS, MidiDrive D/xS, TrioDrive D, MidiDrive D	MaxiDrive	For free use	Drive-specific assignment
O 1.0 .. O 1.3	physical	physical	×	×
O 2.0 .. O 2.3	logical	physical	×	–
O 3	logical	logical	×	–
O 4	logical	logical	×	–
O 8.0 .. O 8.7	logical	logical	×	×
O 9.0 .. O 9.7	logical	logical	×	×
O 10, O 11*	logical	logical	×	–
O 15	logical	logical	×	×
O 16 .. O 20	logical	logical	×	–

* Outputs O 10 and O 11 are reserved for changing the axis control word.

Via the machine data, the inputs can be configured individually so that a drive-specific assignment is activated (e. g. “limit switch +”). In this case, they have a drive-specific function and cannot be used freely any longer.

The use of digital inputs for stopping the drive system (e. g. as limit switch) does not meet the safety requirements of the machinery directive. If an emergency-stop or safety switch-off is required for the machine, the safety requirements of the machinery directive must be observed unconditionally.

The digital outputs are no relay outputs. For controlling e. g. a brake using an output, an external relay is required.

The internal inputs I 15 to I 20 are interconnected with the corresponding internal outputs O 15 to O 20. These inputs and outputs are used as markers, e. g. in part programs.

From firmware V 7.5 on, drive system-specific functions can be assigned to the internal inputs and outputs I 15.x or O 15.x so that another function linked with input I 15.x can be called up via output O 15.x.

Example: “trigger status fast position measurement” is to be output at digital output O 15.0. Digital input I 15.0 is linked with function “axis stop”. As soon as fast position measurement has been triggered via the trigger input, output “trigger status fast position measurement” is set to 1. This, in turn, triggers function “axis stop”.

For details on the corresponding variables, see section Parameters and Actual Values of the Input/Output Function [95] (Access to Inputs and Outputs) and section Input/Output Function Machine Data [131] (Configuration of Inputs and Outputs including lists of all selectable functions and signals). For the technical specifications of the digital inputs/outputs, please see operating instructions “Connection and Commissioning” (6755.202, 6745.202, 6750.202, 6740.202, or 6710.202).

7.1.2 Analog Input

Analog inputs are available for devices with -A1 option, only. The analog input function is responsible for the processing of signals from analog input “setpoint”.

Analog input “setpoint” can be used for one of the following tasks:

- Setting the *target velocity* under consideration of ramps etc. (in axis operating modes **profile position mode**, **velocity mode**, and **flying shear**)
- Setting the *target velocity direct* (in axis operating mode **velocity mode direct**)
- Setting the *torque setpoint* (in axis operating mode **torque mode**)
- Setting the *max current amount* (in all axis operating modes)
- Further processing in the part program or via a higher-level controller (via variable *analog inputs*)

Via variables *...-analog factor* and *... -analog offset*, the scaling of the analog value and a correction of the zero point can be carried out.

In section Setpoint Sources Machine Data [144], the variables by which analog input “setpoint” is selected in the various axis operating modes as setpoint source are summarized.

The voltage applied to this input can also be output at an analog output or recorded using the trace function.

For technical specifications of analog input “setpoint”, please refer to operating instructions “Connection and Commissioning” (6755.202, 6745.202, 6750.202, 6740.202, or 6710.202).

7.1.3 Analog Output

Analog outputs are available for devices with -A1 option, only. The analog output function can be used for the output of various signals via analog outputs “Ist1” and “Ist2” of the servo drive.

Machine data *analog output 1 signal selection* or *analog output 2 signal selection* can be used for selecting the signals to be output on the corresponding output. Via parameters *analog output 1 FP factor* or *analog output 2 FP factor*, the output values can be scaled. Further details are described in section Input/Output Machine Data 131.

For the technical specifications of analog outputs “actual values”, please refer to operating instructions “Connection and Commissioning” (6755.202, 6745.202, 6750.202, 6740.202, or 6710.202).

7.2 Switching Points

The switching points can be used for setting the switching outputs depending on the actual position of the axis. 8 switching outputs are available (bits 0 to 7 in *position switching point status*); to each switching output, two switching points *position switching point positive*, *position switching point negative*, and a switch for signal inversion (bits 0 to 7 in *position switching point inverted*) are assigned.

A switching output is set according to the following condition:

switching output = ((*actual position* ≥ *position switching point positive*)

AND (*actual position* < *position switching point negative*))

XOR *position switching point inverted*

This means that the switching output delivers a 1 (if *position switching point inverted* = 0) if the actual position is between the positive (lower limit) and the negative switching point (upper limit). If it is outside this range, the switching output delivers a 0. This behavior is inverted if the switching point is inverted (if *position switching point inverted* = 1), i. e. a 0 is delivered if the actual position is within the range and a 1 if it is outside the range.

The actual position is queried at a sampling rate of 1 kHz which results in a maximum delay time of 1 ms.

For the output of switching outputs on the digital outputs of the servo drive, see section Input/Output Machine Data 131. Further details on the variables can be found in sections Device Control and Status Information 82, *position switching point status,* and Switching Points Machine Data 184.

7.3 Trigger and Measuring Functions

The trigger and measuring functions can be used for measuring and storing the current position of the axis with a signal at trigger input “measure position” or "measure position quickly". The measured value is stated in the *position sensor measured value* in position sensor steps, in the *position sensor measured value PU* in length units defined by the user.

The *trigger mask* releases the measuring mechanism by setting the corresponding bit on 1 (time a in the figure below).

- Position measurement (trigger mask bit 0): The position measurement is triggered via trigger input "position measurement". With the next positive edge at the input, the actual position is stored in *position sensor measured value* and *position sensor measured value 1 PU* (b).

  The trigger input is scanned at the position sensor cycle (1 ms) which determines the time accuracy of triggering.

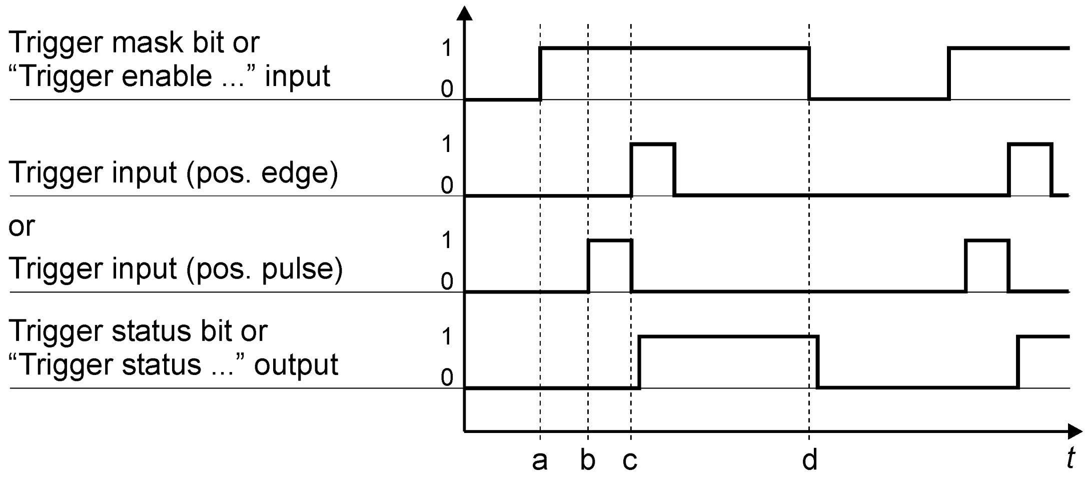


Figure 12: Triggering the Measuring Function

- Fast position measurement (trigger mask bit 2): The position measurement is triggered via trigger input "fast position measurement"

  Two different types of trigger events initiating a triggering are defined:

  – edge: positive or negative

  – pulse: positive or negative

  The selection of the trigger event is defined via variable *trigger mode selection* in which the actual position is stored.

  – In an edge triggering, the actual position is stored with the corresponding edge in variables *position sensor measured value1* and *position sensor measured value1 PU*.

  – In a pulse triggering, the actual position is stored with the first edge of the pulse in variables *position sensor measured value1* and *position sensor measured value1 PU*, the actual position at the point of time of the second edge of the pulse is stored in variables *position sensor measured value2* and *position sensor measured value2 PU*.

In this case, the trigger input is scanned at the signal processor cycle (62.5 µs) which determines the time accuracy of triggering.

The corresponding bit in the *trigger status word* indicates that a measurement has been carried out. This bit is reset automatically by setting the release bit in the *trigger mask*. This also deletes the *position sensor measured value* (c).

If several position measurements are ready for triggering at the same time, the respective bit in the *trigger status word* determines by which trigger input the *actual position* was stored in the *position sensor measured value*. If several bits are active simultaneously in the *trigger status word*, the value in the *position sensor measured value* is undefined.

A fast position measurement is only possible if the motor position sensor is selected as position sensor, otherwise, the value remains undefined.

The position measurement is not supported by SPP Windows.

For further details on the parameters of the trigger and measuring functions, see section Parameters and Actual Values Trigger and Measuring Functions [99].

To make a digital input available for measuring, its drive-specific assignment must be configured to “position measurement” or “fast position measurement”, see *signal selection digital inputs ...* in section Input/Output Machine Data [131].

7.4 Trace Function

The trace function can be used for recording certain actual values and parameters of an axis (e. g. the *actual velocity*) in real-time in a memory area (the so-called *trace buffer*). These data can be read out later on and e. g. displayed graphically to facilitate commissioning of the drive.

Three of these memory areas are available (*trace buffer 1*, *trace buffer 2*, and *trace buffer 3*) so that three signals can be recorded simultaneously.

Command and commissioning software SPP Windows contains oscilloscope functions by which the recording of data and their on-screen display can be controlled in a dialog without having to deal with the following variables. In this case, the trace function can either be started directly in SPP Windows or via an external trigger.

The trace function is released via bit 4 of the *trigger mask* or digital input “trigger release trace recording”. Trace recording is started via *trace start* or with a signal at “trigger input trace”. The signals selected via *trace index 1* to *trace index 3* are stored in the trace buffer. *Trace time* indicates the interval length in controller cycles.

For further details on the parameters of the trace function, see section Parameters of the Trace Function [102].

8 Setting the Machine Data During First Commissioning

One prerequisite for a proper operation of the drive system is the setting of basic machine data during commissioning.

In many cases, a manual setting of machine data by the user is not required if the application data have been set by the manufacturer.

If a setting of machine data by the user is required, he has to consider that the setting of some machine data (e. g. *target position factor*, *target velocity factor*) affects the setting of various other machine data. Therefore, it is important to proceed in the order described below for first commissioning. This order corresponds to the arrangement of the machine data in command and commissioning software SPP Windows.

Please, note that in monitoring mode changes in the machine data have to be transferred to the servo drive by the user, also note that all data transferred get lost when the 24 V are switched off unless they have been stored in the servo drive via “Communication/Save in device”.

This section gives only a short description of the variables, for detailed information on the individual variables see the variable descriptions in section Variable Descriptions 78.

8.1 Starting Command and Commissioning Software SPP Windows

Connect the servo drive with the serial interface of your PC.

Switch on the drive system and start command and commissioning software SPP Windows on your PC. (Information on installation of SPP Windows and update of databases can be found in operating instructions 6710.207.)

Establish a connection between servo drive and PC by confirming prompt “Establish connection” with “OK”.

Now, you can load all data from the drive via “Communication/Load from device”.

8.2 Setting the Motor Data

8.2.1 Selecting the Motor

First of all, select the motor via menu item “Operate/Load motor data...”. There, you will find a list from which you can select the motor used by double-clicking the desired motor name. In case you receive message “No corresponding motor database available” or your motor does not show up in the list, you have to enter the motor data on your own, see next section.

Where required, a window appears with a list of additional data that can be loaded. Select the additional data you would like to load and confirm your selection with “OK” or click on “No” or “Cancel” if you do not want to load additional data. A data set with the machine data of the used motor is loaded, now.

The machine data of the motor were acquired by the drive manufacturer. With devices with firmware versions below V 8.5.9 they cannot be changed individually by the user.

To initialize the servo drive with the new motor data, it is recommended to save them after transmission to the device via “Communication / Save in device” and to switch the servo drive (24 V control voltage) off an on again.

8.2.2 Entering the Motor Data

In case your motor is not included in the motor database or a motor database is not available for your motor, you have to enter the technical data of the motor yourself. To do so, select “Operate / Parameterization...” and enter the data from the data sheet for your motor in “Motor data”.

For devices with firmware versions below V 8.5.9, the motor data cannot be changed individually, instead, a list of the motors included in the database appears. If your motor is not stated in this list, please contact ESR.

When entering the motor data, please make sure that you enter the correct values and observe the defined physical units. By pressing the F1 key, you will get information on the individual parameters.

If the motor data are entered wrongly or incompletely, this may lead to an unexpected (uncontrolled) system behavior and damages at motor and/or servo drive!

To initialize the servo drive with the new motor data, it is recommended to save them after transmission to the device via “Communication / Save in device” and to switch the servo drive (24 V control voltage) off an on again.

8.3 Setting the Basic Axis Data

Then, set the axis data in window “Parameterization/Axis data”. The objects not relevant for the used operating mode can be masked out using the operating mode filter.

First of all, set the desired values for **axis control** using the following variables (variable description section Variable Descriptions Axis Machine Data 141):

- *Action enable inactive/active*

  If you would like to use input “enable” for switching the device on/off or for part program control, you can define the behavior using the enable selection code. By default, the enable input is switched ineffective, enabling is carried out via the axis control word.

- *Brake released at device status* as well as *brake activation time* and *brake release time*

If you use a motor with holding brake, you can use the *release brake selection code* to select in which states of the axis state machine the brake is released, i. e. the corresponding “brake” output is activated. The *brake activation time* indicates the time required by the brake to build up the full holding torque. The *brake release time* indicates the time required for releasing the shaft.

- *Axis operating mode at startup select*

  This variable can be used for defining which axis operating mode is active after the drive has been switched on.

- *Stop axis selection code*

  If you would like to stop the axis via a digital input, this selection code can be used for selecting the way the axis is stopped.

If you use an external position sensor or would like to set target velocity, torque, or maximum current via analog input, make the corresponding settings in “set-point sources”, “position measuring system”, and “drive”.

Setpoint sources (variable description section Setpoint Sources Machine Data 144)

- *Target velocity source*

  If you set the target velocity via an analog input (±10 V interface), select that via the *target velocity source* variable. In this case, set scaling and zero offset in variables *target velocity analog factor* and *target velocity analog offset*.

- *Torque setpoint source*

  If you set the torque via an analog input (±10 V interface), select that via the *torque setpoint source* variable. In this case, set scaling and zero offset in variables *torque analog factor* and *torque analog offset*.

- *Max current amount source*

  If you set the current limitation via an analog input (±10 V interface), select that via the *max current amount source* variable. In this case, set scaling and zero offset in variables *max current amount analog factor* and *max current amount analog offset*.

Position measuring system (variable description section Position Measuring System Machine Data 149)

- *Position sensor source*

  If you have an external position sensor connected at input encoder signals (type code option L1 or L2), you can use this variable to select it as position sensor source. (For devices with firmware versions below V 8.5.9, this parameter can be found in setpoint sources.)

- *Position sensor resolution*

  This variable is used for defining in how many position sensor steps (PSS) a motor revolution is resolved independent of the position sensor used. Information on the possible values can be found in the variable descriptions.

- *Gear ratio*

  In case you use an external position sensor connected with the motor shaft via a gearbox as position sensor source, enter the gear ratio here so that the position information can be calculated for the motor shaft.

- *Position sensor data external sensor*

  In case you use an external position sensor as position sensor source, enter the technical data, here. By pressing the F1 key, you will get information on the individual parameters.

- *Encoder absolute type*

  If you use a multi-turn encoder operated as circular axis, the system will use this encoder like a single-turn encoder.

- *Motorposition sin/cos monitoring*

  This variable can be used for defining that the servo drive monitors the sine/cosine signals (if available) even if otherwise the digital signals of the motor position sensor are evaluated.

Drive system (variable description section Drive System Machine Data 155)

- *Polarities*

  If you use an external position sensor, the position polarity can be changed via the counting direction of the encoder signals.

- With variables *maximum motor speed*, *continuous current motor*, *max current amount*, and *max current amount2*, speed and torque of the motor can be set or limited.

  As standard, the rated speed is accepted as maximum motor speed, and continuous and peak current are limited to 100% or 200% of the motor rated current respectively.

- *Inertia load*

  If you enter the moment of inertia of the load, it will be considered automatically on the control loops.

- *Commutation reference*

  If the motor position sensor does not supply absolute information, you can enter the way the commutation reference will be determined, here.

With the **factors and units**, the indication of rated and actual values can be adjusted to physical units. This must be done before values stated in these units are written into other variables. The following variables are available for that (variable description section Factors and Units Machine Data 158):

- *Position factor numerator / position factor denominator*

  These variables determine the number of position sensor steps (PSS) one position unit (PU) has (standard: 1 PU = 1 PSS).

- *Home offset*

  If homing is required for your application, you can define the home offset, here. The home offset is the distance between the machine home position approached during homing and the application zero.

- *Velocity factor numerator / velocity factor denominator*

  These variables are used for determining how many revolutions per minute one velocity unit (VU) has (standard: 1 VU = 1 r.p.m.).

- *Application coordinate system*

  If the coordinate system of the application differs from the counting direction of the axis, it can be reversed using this variable.

For positioning tasks, the **positioning range** must be set. For that, the following variables are available (variable description section Positioning Range Machine Data 161):

- *Axis type*

  The *axis type* states whether the application is a circular or a linear axis (e. g. via spindle or toothed belt).

- If axis type “circular axis” is selected, the upper and lower limit of the value range (= one “revolution” from the perspective of the application) is set using variables *range limit min* and *range limit max*.
- If axis type “linear axis” is selected, the absolute position limits within which positioning is possible are set using variable *position limit min/max*.

The following variables are available for defining acceleration and deceleration **ramps** (variable description section Ramps Machine Data 164):

- The *motion profile type* specifies the process of accelerating and decelerating (e. g. linear).
- The *acceleration time* specifies the time for accelerating from 0 to *ramps reference velocity*.
- The *deceleration time* specifies the time for decelerating from *ramps reference velocity* to 0.
- The *quick stop time* specifies the time for decelerating from *ramp reference velocity* to 0. It is only effective if deceleration is executed using the “quick stop” control command in the axis control word or a quick standstill is triggered in case of a fault.
- The *ramps reference velocity* specifies the velocity to which parameters *acceleration time*, *deceleration time*, and *quick stop time* apply. Thus, the quotient from time and reference value defines the slope of the ramp.

8.4 Setting the Current Control Loop

In case the motor data could be loaded from a motor database, the current control loop has already been set and this step can be skipped. In all other cases, the current control loop must be parameterized using commissioning tool current controller in command and commissioning software SPP Windows (menu *Operate / Commissioning / Current control*).

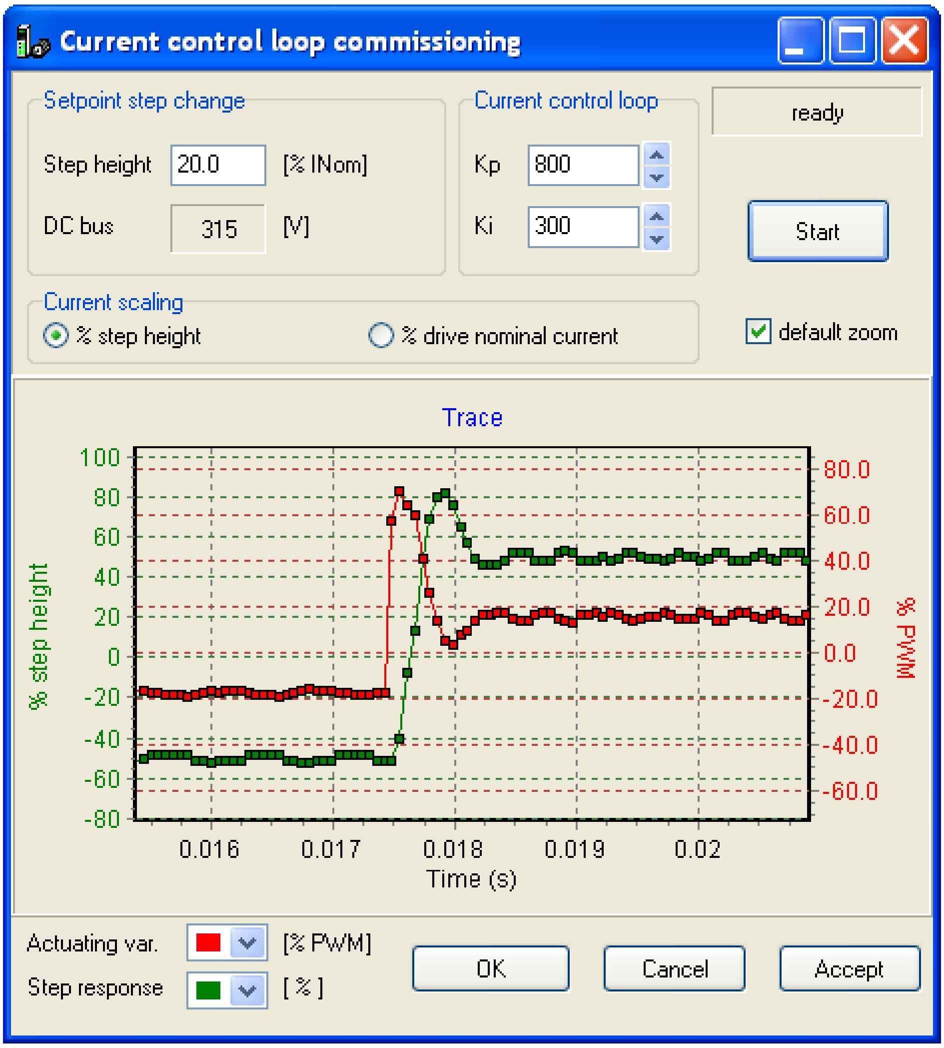


Bild 13: Commissioning tool current control loop

The following must be provided and checked before commissioning tool current control loop can be run:

- Power supply cables and encoder cables must be connected properly (phase order U-V-W, shield, etc.).
- The technical specifications of motor and motor position sensor must be defined in window „Parameterization“ in tab *Motor data*.
- If possible, the motor should be flanged on, its shaft should not be connected to the machine yet.

As the autocommutation function does not work reliably until the current controller parameters have been acquired, the commutation position of the motor is usually unknown when the current control loop commissioning is called up for the first time. In this case, 0 must be set as starting value for current control loop parameters Kp and Ki before the run is triggered for the first time using button “Start”.

Caution: Wrong connection, incorrect data, or inappropriate operation can lead to an undesired behavior and possibly to a motor damage, too! For that reason, the commissioning tool may be used by expert personnel, only! Therefore, this function is accessible for access right specialist, only.

Commissioning tool current controller creates a setpoint step change and the response of the servo drive system is recorded. For that, the idle current is excited in order to avoid a movement of the shaft. Nevertheless, a movement of the shaft must be expected. If possible, the shaft should be locked mechanically. (Please note that an automatic search of the commutation angle is not possible without a parameterized current control loop.)

Optimize the controller parameters by triggering the setpoint step change via button “Start” and evaluating the recorded data for variable und step response in the oscilloscope area of the window. Change the values for Kp and Ki correspondingly and repeat the procedure until you have reached a satisfying result.

Aim of the parameterization is to receive a step change reaching the setpoint after about four to six controller steps (one step corresponds to 62.5 µs). (Tip: Activate function “display points" by right-clicking on the graph in order to see the controller steps.) The step response should not overshoot by more than 25% (75% line) and reach the run-in state after 2 ms at most, see the figure on top. The other diagrams show typical scenarios for wrong controller settings:

- If proportional factor Kp is too low, the current needs too much time to get to the desired setpoint.
- If proportional factor Kp is too high, the drive tends to oscillate.
- If integral factor Ki is too low, the current reaches the desired value too slowly and without overshooting.
- If integral factor Ki is too high, the current oscillates around the desired setpoint several times before finally reaching it.

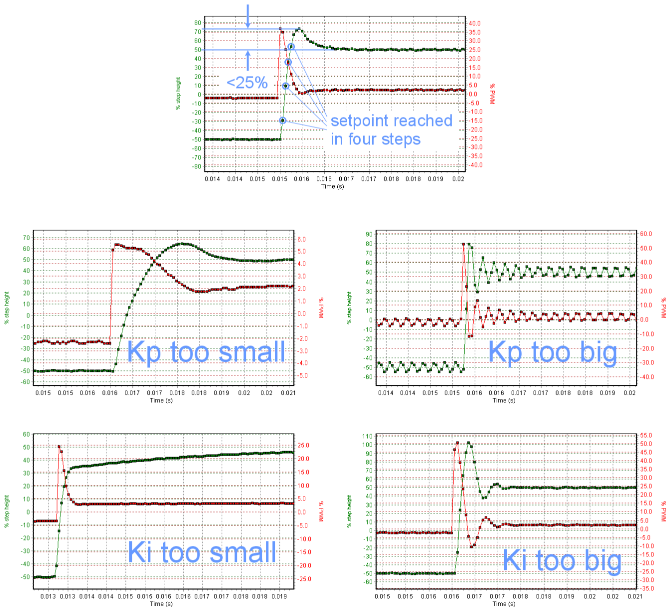


Figure 14: Current control loop optimization

Depending on motor, installed motor position sensor, etc., the diagrams may show a more or less intense “noise”. However, if the recorded step response looks totally different from the examples, the parameters set for the current control loop might be far below or above their optimal values. In this case, it is recommended to cover a wider area of possible values in large steps. Also check whether or not the setpoint step change is set in a way that an appreciable output of the power circuit is reached (approx. 20 to 80% PWM) without limitation by the servo drive.

In case the motor was locked mechanically, the lock can be removed now.

If required, the automatic commutation (autocommutation) can be activated when the correct parameters have been found. For motors with a high number of pole pairs, this is usually desirable because a mechanical adjustment of the encoder can be difficult. After that, the current control loop should be parameterized again with the autocommutation switched on for fine tuning. When the autocommutation works properly, the shaft should not move when commissioning tool current control loop is used, except for higher current values (rated current or higher).

8.5 Setting the Speed Control Loop

The speed control loop optimization is carried out using commissioning tool speed control loop of command and commissioning software SPP Windows (menu *Operate / Commissioning / Speed Control*). Prerequisite is an optimized current control loop (see previous section).

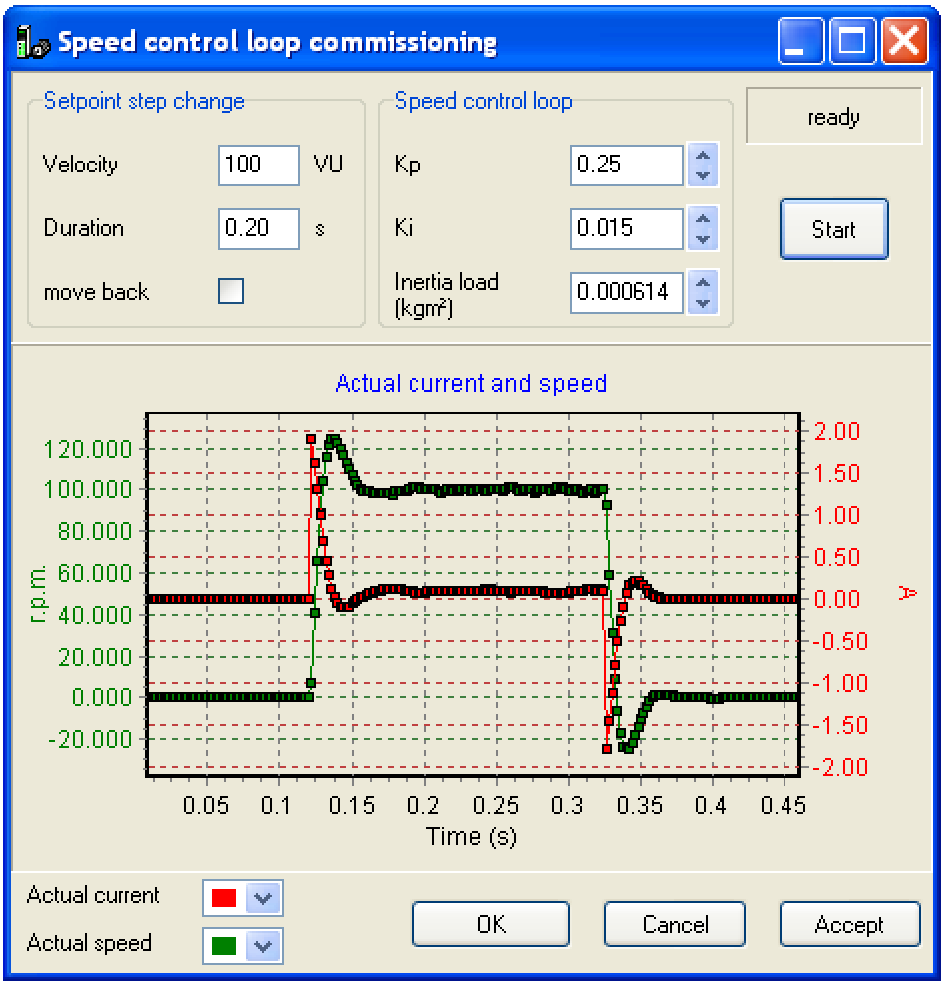


Figure 15: Commissioning tool speed control loop

For the speed control loop optimization, the motor must be flanged on, its shaft should be connected to the machine. It must be able to turn several times without hitting an obstacle. If the travel distance in the machine is limited, use function “return”. If the motor cannot be connected with the machine for the optimization, a mass with a moment of inertia corresponding roughly to the machine part to be moved should be mounted on the shaft.

Caution: If used inappropriately, the setpoint step changes can lead to an unexpected behavior and machine damages. Therefore, the commissioning tool may be used by qualified personnel, only! For that reason, the function is available for access right specialist, only.

With commissioning tool speed control loop, a setpoint step change is created and the response of the servo drive is recorded. The speed should be selected in a way that the required current does not exceed 100%. The duration should be sufficient for the drive to reach a run-in state.

Optimize the controller parameters by triggering the setpoint step change via button “Start” and evaluating the recorded data for variable und step change response in the oscilloscope area of the window. Change the values for Kp and Ki correspondingly and repeat the procedure until you have reached a satisfying result.

Aim of the parameterization is to receive a step change response reaching the setpoint within a short period of time and overshooting by not more than approx. 25%. The other diagrams show typical scenarios for wrong controller settings:

- If proportional factor Kp is too low, the speed needs too much time to get to the desired setpoint.
- If proportional factor Kp is too high, the drive tends to oscillate.
- If integral factor Ki is too low, the speed reaches the desired value too slowly and without overshooting.
- If integral factor Ki is too high, the speed oscillates around the desired setpoint several times before finally reaching it.

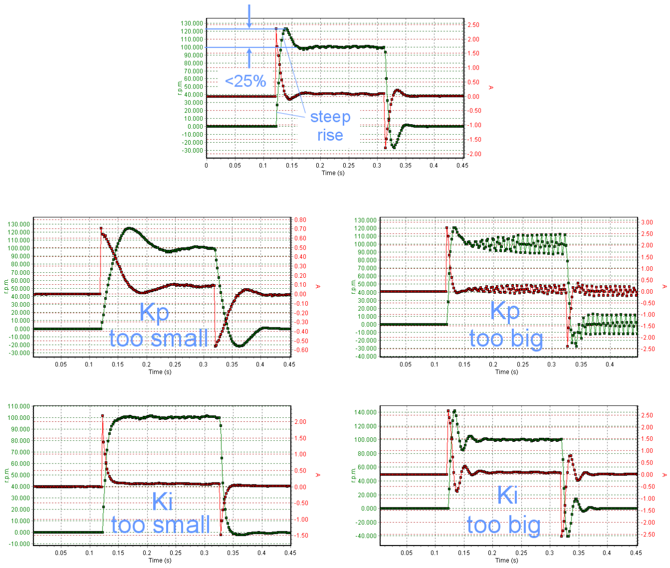


Figure 16: Speed control loop optimization

Depending on motor, installed motor position sensor, load moment of inertia, etc., the diagrams may show distinctly more flat or inclined curves and a more or less intense “noise”. However, if the recorded step change response looks totally different from the examples, the parameters set for the speed control loop might be far below or above their optimal values. In this case, it is recommended to cover a wider area of possible values in large steps. Also check whether or not the setpoint step change is set in a way that an appreciable current is reached without a limitation by the servo drive.

8.6 Setting Other Axis Data and Device Data

Depending on the application, further settings can be carried out in the axis data, subsequently, e.g.

- Position control loop

  For positioning tasks, proportional factor Kp must be set in a way that the position is kept tightly without oscillating at standstill.

- Homing

  If homing is required, select the corresponding selection code (see Homing Mode Sequences 40) and the homing speed, here.

In the device data, the function of the digital and analog inputs and outputs can be set. In case you are using an external shunt resistor, the data must be entered, here.

9 Variable Descriptions

The following sections describe all variables in detail. The variable descriptions are arranged according to control and status information, parameters and actual values, as well as machine data. The structure within the groups corresponds to the structure of the variable lists of command and commissioning software SPP Windows. Access by name is possible via the key index.

In the following, the structure of the variable description is described in detail. Basic information on a variable is summarized in a table in the following form:

Name	Index: `1234`, Short name: `VarName`
SPP Windows	`Variable name` <u>`value`</u> `unit`
Type	...
Array element n	...
Var. type	... access
Unit	...
Standard value	...
Valid	in axis operating modes ...

Name: This contains the detailed variable name (bold type).

Index: The index is a hexadecimal number by which the variables can be accessed during communication with the servo drive.

Short name: This is a short name which can be used instead of the index to access the variable using a special high-level language driver during communication via the serial interface.

SPP Windows In the SPP Windows command and commissioning software, the variable is displayed as shown above. The line contains a sample value for the variable (underlined).

Type: This line includes various information:

- Object code
  - Simple Variable

    The simple variable only contains one element of the specified data type. When accessing a simple variable via index/subindex, subindex 0 must be specified.
  - Array

    The array contains several elements of the specified data type. When accessing an array, the subindex of the required element must be specified (1 or higher). Especially with CANopen® variables, the number of elements of the array can be read via subindex 0.
  - Record

    A record contains several elements of various data types. When accessing a record, the subindex of the required element must be specified (1 or higher). Especially with CANopen® variables, the number of elements of the record can be read via subindex 0.

- Data type
  - Boolean (bool)
  - Integer8 (i8) = byte in two's complement representation
  - Integer16 (i16) = word in two's complement representation
  - Integer32 (i32) = double word in two's complement representation
  - Unsigned8 (u8) = byte, unsigned
  - Unsigned16 (u16) = word, unsigned
  - Unsigned32 (u32) = double word, unsigned
  - Float32 (f32) = floating-point number according to IEEE 754 (bit 31 = sign, bit 30 .. 23 = exponent, bit 22 .. 0 = mantissa)
  - Visible string (VisStr) with length = a sequence of bytes containing text
  - Octet string (OctStr) with length = a sequence of bytes containing binary-coded information

Array element n: If the individual elements of an array have specific names and/ or units, this is specified here.

Var. type: This line includes various information:

- Variable type

  Possible values:
  - data range, mapped to an array
  - part program data range
  - actual value
  - constant (variable value does not change)
  - axis machine data
  - device machine data
  - motor machine data
  - drive machine data
  - machine data
  - setpoint or parameter for a function
  - program variables
  - status information
  - control information
  - adjustable value
- Access

  Possible values:

- read and write
- read only
- write only

Unit: The unit of measurement of the variable value as used internally.

If the variable is accessed via command and commissioning software SPP Windows, the unit is converted automatically by the software, if required. The unit displayed there is valid, e. g. “A” (ampere), “mH” (millihenry), or “°C” (degrees centigrade).

If the variable is accessed directly, e. g. by the higher-level controller, the stated unit must be taken into account and the variable value must be converted, if required. Example: If “1 mA” is stated as unit, a variable value of 500 corresponds to a current of 0.5 A. For unit “1 H”, a variable value of 0.2 corresponds to an inductance of 200 mH. For unit “0.1 °C”, a variable value of 1200 corresponds to a temperature of 120 °C.

Other units used are:

- Position sensor steps (PSS)

  One motor revolution is resolved into the number of steps defined in parameter *position sensor resolution*. That means if a setpoint or actual value has got the value defined in *position sensor resolution*, this corresponds to one motor revolution.

- Position units (PU for short)

  The significance of the position units depends on the setting of the *position factor* variable. These can be divided into two categories:

  - Position factor = 1 PSS / 1 PU (delivery state):

    One position unit corresponds to one position sensor step and thus depends exclusively on the set position sensor resolution.

  - Position factor ≠ 1 PSS / PU (application-specific):

    One position unit corresponds to a specific number of position sensor steps (e. g. 10, 1/10, 27/33), which is set via the *position factor* variable. This can be used for displaying position values in physical units (e. g. mm, µm, º).

  Examples for the application of *position factor* are given in the corresponding section.

- 0.25 r.p.m. / 0.5 r.p.m. / 1.0 r.p.m.

  Certain machine data (such as *max motor speed*) are set in defined fractions of revolutions per minute. Example: A speed of 3,000 r.p.m. corresponds to a parameter value of 6,000 in case it is stated in 0.5 r.p.m. (3,000 / 0.5 = 6,000) or 12,000 in case it is stated in 0.25 r.p.m. In SPP Windows, input is always directly in r.p.m., conversion is automatic.

- Velocity units (VU for short)

  The meaning of the velocity units depends on the setting of the *target velocity factor* variable. These can be divided into two categories:

  – Target velocity factor = 1 r.p.m. / 1 VU

    A velocity unit corresponds to 1 r.p.m., i. e., corresponding parameters are specified directly in revolutions per minute. Caution: because the numerator in *target velocity factor* is specified internally in 0.5 or 0.25 r.p.m., 2 or 4 must be entered respectively in this case (this conversion is not necessary in SPP Windows).

  – Target velocity factor ≠ 1 r.p.m. / 1 VU

    A velocity unit corresponds to a specific number of revolutions per minute (e. g. 4, 400, 48/37), which is set via the *target velocity factor* variable. This can be used for displaying velocity values in physical units (e. g. mm/s, m/s).

  Examples for the application of *target velocity factor* are given in the corresponding section.

- Select

  Only specific numerical values are permitted here. When working with the SPP Windows software, permitted values can be selected from a list; knowledge of number codes is not necessary.

Standard value: With the exception of machine data, the parameter value after switch-on is specified here.

In the case of machine data, after switching on, the value is transferred from the flash memory. The setting of the machine data and their storage in the flash memory is carried out using the SPP Windows command and commissioning software.

Valid: This specifies the validity of the variable or its functionality in the drive system depending on other parameters (e. g. selected axis operating mode) or drives. For variables which did not exist in older software versions, the firmware version with which they were introduced is also specified.

Description: Finally, there is a short text describing the variable.

Example of a variable description:

Position Limit Min Max	Index: 607d, Short name: PosGrenzwert
SPP Windows	position limit min <u>-536,870,912</u> PU position limit max <u>536,870,911</u> PU
Type	array, two type integer32 elements
Array element 1	position limit min in position units (PU)
Array element 2	position limit max in position units (PU)
Var. type	axis machine data read and write
Unit	position units (PU) related to machine home position
Standard value	–
Valid	in axis operating mode ***profile position mode*** (if *axis type* = linear axis)

9.1 Variable Descriptions Control and Status Information

9.1.1 Device Control and Status Information

The control and status information device include variable *device operating mode*.

In command and commissioning software SPP Windows, the control and status information of the device can be found in the “Device control” window.

Device operating mode	Index: 5f42, Short name: GeraetBAAuswahl
SPP Windows	Dev.-md <u>Command mode</u>
Type	simple variable, integer16
Var. type	control information read and write writing permitted in certain axis and program states, only; further information below
Unit	select
Standard value	command mode
Valid	always

The *device operating mode* determines whether the movements are set individually via parameter (device operating mode **command mode**) or control is carried out via a part program in the servo drive (device operating mode **program mode**...).

It is only permitted to switch from command mode to one of the program operating modes, and vice versa, if the axis is in the “Not ready” or “Switch on disabled” state. In addition, the program state machine must be “idle” when switching from one of the program operating modes to command mode.

A change between program mode automatic and program mode single step is only possible if the part program state machine is “idle” or “stopped”.

Permitted values:

Value	Device operating mode
1	Command mode
2	Program mode automatic
3	Program mode single step

For a detailed description of the device operating modes, see section Device Operating Modes: Command and Program Mode 60.

9.1.2 Axis Control and Status Information

The following variables belong to the axis control and status information:

- *Axis operating mode*
- *Axis control word*
- *Axis status word*
- *Position switching point status*
- *Axis error code*
- *Fault detail*

In command and commissioning software SPP Windows, the control and status information of the axis can be found in the “Device control”, “Fault”, or “Device status” windows.

Axis operating mode	Index: 6060, Short name: AchsBAAuswahl
SPP Windows	Axis-md Velocity mode
Type	simple variable, integer8 (V 8.5.9 or higher) simple variable, integer16 (up to V 8.5.9)
Var. type	control information read and write writing permitted in certain axis and program states, only; further information below
Unit	select
Standard value	velocity mode
Valid	in all axis operating modes

Via the *axis operating mode*, an operating mode can be selected for the corresponding axis.

The axis operating modes can only be changed in states “Switch on disabled”, “Ready to switch on”, or “Switched on”.

For a detailed description of the axis operating modes, see section Axis Operating Modes 29.

Permitted values:

Value	Axis operating mode
4	Torque mode
–3	Velocity mode direct (V 3.2 or higher)
3	Velocity mode
1	Profile position mode
6	Homing mode
–1	Electronic gearing
–2	Flying shear (V 4.5 or higher)
–4	Velocity profile (V 5.95 or higher)
–5	Timed positioning mode (V 8.5.9 or higher)
10	Cyclic synchronous torque mode (V 8.5.999.53 or higher)
9	Cyclic synchronous velocity mode (V 8.5.999.54 or higher)
8	Cyclic synchronous position mode (V 8.5.999.62 or higher)
7	Interpolated Position Mode (V 5.95 or higher)

Axis control word	Index: `6040`, Short name: `AchsSteuer`
SPP Windows	`8000 Control word`
Type	simple variable, octet string length 2
Var. type	control information — read and write
Unit	–
Standard value	0000_{hex}
Valid	in all axis operating modes

Using bits and bit groups in the *axis control word*, the following functions are executed for one axis:

- control of axis states (bit 0 .. 3 and 7)
- control of axis movement (bit 4 .. 6)
- control of part program (bit 14, 15)

An overview of the bit assignment of the *axis control word* can be found in section Axis Control Word 20.

Axis status word	Index: `6041`, Short name: `AchsStatus`
SPP Windows	`9408 Status word`
Type	simple variable, octet string length 2
Var. type	status information — read only
Unit	–
Standard value	depends on axis state machines and possible faults
Valid	in all axis operating modes

Bits and bit groups in the *axis status word* report the states of axis and movements:

- indication of the states of the axis state machine set via the *axis control word* (bit 0 .. 3 und 5, 6; additional info in 4)
- indication of the states of movements depends on the individual axis operating modes (bit 10 .. 12)
- indication of the states of the part program (bit 14, 15)

An overview of the bit assignment of the *axis status word* can be found in section Axis Status Word 22.

Position switching point status	Index: 5f37, Short name: PosPSchaltStatus
SPP Windows	–
Type	simple variable, octet string length 2
Var. type	status information read only
Unit	–
Standard value	–
Valid	always; V 5.1 or higher

Bits 0 to 7 of variable *position switching point status* are assigned to switching outputs 1 to 8. The switching points can be used for setting the switching outputs depending on the actual position of the axis.

General information on the switching points can be found in section Switching Points 64. You will find further details on the switching points machine data in section Switching Points Machine Data 184. To output the switching outputs on the digital outputs of the servo drive, see section Input/Output Function Machine Data 131.

Axis error code	Index: 603f, Short name: AchsStoerung
SPP Windows	Code 7303 : Encoder fault
Type	simple variable, octet string length 2
Var. type	status information read only
Unit	–
Standard value	0000_{hex} or a fault that may have occurred during switch-on
Valid	in all axis operating modes

The *axis error code* is set to a specific value by a fault. Once the cause of the fault is eliminated, the *axis error code* can be reset to value 0 (no fault) via bit “Reset fault” in the *axis control word*. The code can also be reset by switching the control supply voltage off and on.

If the axis is in the “Fault” state, the parameter contains a value other than 0. If the axis is not in the “Fault” state, the parameter contains value 0.

A fault is also indicated by the corresponding LED on the front panel of the servo drive. Whether the LED is permanently lit or flashing gives a rough indication of the type of fault:

- “Störung/Fault” or “FLT” LED flashing:

  Programming, parameterization, or software fault

  – fault codes 5520, 9010 as well as all fault codes 6xxx

- “Störung/Fault” or “FLT” LED permanently lit:

  Drive, wiring, or hardware fault

  – all fault codes except for 5520, 9010, and 6xxx

A list containing all possible fault codes and their descriptions can be found in Appendix C 191.

Fault detail	Index: `5f81`, Short name: `StoerDetail`
SPP Windows	`Fault detail` ______________
Type	array, 4 type visible string length 16 elements
Var. type	status information read only
Unit	–
Standard value	–
Valid	in all axis operating modes; V 3.3 or higher

Some fault codes also contain additional information on the cause of the error in clear text.

9.2 Variable Descriptions Parameters and Actual Values

9.2.1 Parameters and Actual Values Torque Mode

The parameters and actual values of the torque mode include the following variables:

- *Torque setpoint / force setpoint*
- *Actual current*

In command and commissioning software SPP Windows, the parameters and actual values of the torque mode can be found in the “Device control” window.

Torque/force setpoint	Index: 6071, Short name: Msoll
SPP Windows	Torque setpoint: 50 % Set
Type	simple variable, integer16
Var. type	setpoint or parameter for a function read and write
Unit	0.1% of the *rated torque* or *force* of the motor
Standard value	–
Valid	in axis operating modes ***torque mode*** and ***cyclic synchronous torque mode***

This variable specifies the torque or the force of the motor.

Actual current	Index: 6078, Short name: IIst
SPP Windows	Actual current 2.5 % 0 [ ] 200%
Type	simple variable, integer16
Var. type	actual value read only
Unit	0.1% of the *rated current* of the motor
Standard value	–
Valid	in all axis operating modes

The actual motor current can be read out here in all axis operating modes.

9.2.2 Parameters and Actual Values Velocity Mode

The parameters and actual values of the velocity mode include the following variables:

- *Target velocity*
- *Actual velocity*
- *Override factor*

In command and commissioning software SPP Windows, the parameters and actual values of the velocity mode can be found in the “Device control” window.

Target velocity	Index: 6081, Short name: VelVerfahr
SPP Windows	Target velocity: 3000 VU Set
Type	simple variable, integer32
Var. type	setpoint or parameter for a function read and write
Unit	velocity units (VU)
Standard value	0
Valid	in axis operating modes ***velocity mode***, ***profile position mode***, and ***flying shear***

The *target velocity* specifies the speed of the axis in axis operating modes **profile position mode** or **velocity mode**, taking into account other parameters such as *acceleration time*.

In axis operating mode **flying shear**, it specifies the velocity for the return movement.

If analog input “Setpoint” is selected as *target velocity source*, the value read out at the input is continuously transferred to the *target velocity* variable in the specified axis operating modes (and can be read out from there as well).

Actual velocity	Index: `606c`, Short name: `VelIst`
SPP Windows	`Actual vel.  2649  VU  0 [    ] 200%`
Type	simple variable, integer32
Var. type	actual value — read only
Unit	velocity units (VU)
Standard value	0; changes when the axis is moved (even manually)
Valid	in all axis operating modes

The *actual velocity* shows the actual velocity of the axis. It is acquired digitally by differentiating the position information of the position measuring system.

Override factor	Index: `5efe`, Short name: `VelOverride`
SPP Windows	`Override factor:  100,0  %  Set`
Type	simple variable, unsigned16
Var. type	setpoint or parameter for a function — read and write
Unit	65535 ($FFFF_{hex}$) corresponds to 400% target velocity
Standard value	16383 (corresponds to 100%)
Valid	in axis operating modes ***velocity mode***, ***profile position mode***, and ***homing mode***

Factor for influencing target velocity and homing velocity, affects the velocity of the drive.

The *override factor* can be used, among other things, to increase or decrease the target velocity during a running part program.

Examples:

- 3276 (corresponds to 20%): The target velocity is reduced to 20% and relayed to the drive; 3,000 r.p.m. target velocity produces 600 r.p.m. at the shaft.
- 16383 (corresponds to 100%): The target velocity is relayed unchanged to the drive; 3,000 r.p.m. target velocity corresponds to 3,000 r.p.m. at the shaft.
- 24575 (corresponds to 150%): The target velocity is increased to 150% and relayed to the drive; 3,000 r.p.m. target velocity produces 4,500 r.p.m. at the shaft.

9.2.3 Parameters and Actual Values Profile Position Mode

The parameters and actual values of the profile position mode include the following variables:

- *Target position*
- *Internal target position*
- *Actual position*
- *Position sensor actual position*
- *Actual position zeroing*

In command and commissioning software SPP Windows, the parameters and actual values of the profile position mode can be found in the “Device control” window.

Target position	Index: `607a`, Short name: `PosZiel`
SPP Windows	`Target position: absolute 0 PU Set`
Type	simple variable, integer32
Var. type	setpoint or parameter for a function read and write
Unit	position units (PU), related to application zero point (in axis operating mode ***profile position mode***), or position sensor steps (PSS) (in axis operating mode ***cyclic synchronous position mode***)
Standard value	0
Valid	in axis operating mode ***profile position mode*** and ***cyclic-synchronous position mode***

Using the *target position*, the user specifies the position the drive is to approach in axis operating mode **profile position mode**. In this case, other parameters such as *target velocity* and *acceleration time* are taken into account.

Information on when the *target position* is accepted in *internal target position* and thus actually approached is included in section Target Position 34.

With command and commissioning software SPP Windows, the control bits for accepting the *target position* into the *internal target position* are set automatically as soon as a new target position (absolute or relative) is defined with the “Set” button.

Internal target position	Index: 5ef8, Short name: PosZielIntern
SPP Windows	`Target position (abs.) 1,000,000 LE`
Type	simple variable, integer32
Var. type	setpoint or parameter for a function read only
Unit	position units (PU), related to application zero point
Standard value	0
Valid	in axis operating mode ***profile position mode***

In axis operating mode **profile position mode**, the *internal target position* contains the position that is being approached by the axis. The *target position* is transferred to the *internal target position* via specific control bit combinations in the *axis control word*. Information on when the *target position* is accepted in *internal target position* and thus actually approached is included in section Profile Position Mode 34.

Depending on whether the profile position mode is absolute or relative, the *internal target position* is set directly or via an addition from *target position*.

Selection is carried out via bit “absolute/relative” in the *axis control word*:

- Absolute profile position mode (bit “absolute/relative” = 1)

  In this case, the *target position* directly specifies the position to be approached by the drive. The *internal target position* parameter is set to *target position* prior to positioning:

  internal target position = target position

  After positioning, the *actual position* corresponds to the *target position*.

- Relative profile position mode (bit “absolute/relative” = 0)

  In this case, the *target position* specifies the distance for the relative approach of the drive to the last absolute *target position*. The absolute target position is calculated by the drive and stored in the *internal target position* parameter. The calculation is as follows:

  internal target position = internal target position + target position

With command and commissioning software SPP Windows, the control bits for accepting the *target position* into the *internal target position* are set automatically as soon as a new target position (absolute or relative) is defined with the “Set” button.

Actual position	Index: 6064, Short name: PosIst
SPP Windows	Position 23,567,385 PU
Type	simple variable, integer32
Var. type	actual value read only
Unit	position units (PU), related to the application zero point
Standard value	0; changes when the axis is moved (even manually)
Valid	in all axis operating modes

The *actual position* shows the current position of the axis in the position units defined by the user.

Position sensor actual position	Index: 6063, Short name: PosIstLg
SPP Windows	–
Type	simple variable, integer32
Var. type	actual value read only
Unit	position sensor steps (PSS), related to the application zero point
Standard value	0; changes when the axis is moved (even manually)
Valid	in all axis operating modes

The *position sensor actual position* shows the current position of the axis in position sensor steps, the *home offset* is taken into account.

In general, the *actual position* should be used for reading the position. The *position sensor actual position* is useful for reading out the current position in position sensor steps and thus independent of the *position factor*.

Actual position zeroing	Index: 5efc, Short name: PosIstReset
SPP Windows	Reset actual pos.
Type	simple variable, octet string length 1
Var. type	setpoint or parameter for a function write only
Unit	–
Standard value	0
Valid	in all axis operating modes

Writing on this parameter (no matter which value) resets the actual position: for the linear axis to home offset, for the circular axis to range limit min + home offset.

9.2.4 Parameters and Setpoints Interpolated Position Mode

The parameters and setpoints of the **interpolated position mode** include the following variables:

- *Interpolation data record*
- *Alternative interpolation data record*
- *Velocity offset*
- *Alternative for velocity offset*

In command and commissioning software SPP Windows, the parameters and setpoints of the **interpolated position mode** can be found in the “Parameterization” window under *Axis data/Interpolated position mode*.

Interpolation data record	Index: `60c1`, Short name: `IpolDataRecord`
SPP Windows	–
Type	record, 1 element (subindex 0 .. 1), integer32
Record element 1	target position absolute
Var. type	setpoint or parameter for a function read and write
Unit	PSS
Value range	$-2^{24}..2^{24}-1$ PSS
Valid	in axis operating mode ***interpolated position mode***

Via this variable, the target positions (in position sensor steps) are transmitted.

Alternative interpolation data record	Index: `5fc1`, Short name: `AltIpolDataRecord`
SPP Windows	–
Type	record, 2 elements (subindex 0 .. 2)
Record element 1 **Record element 2**	interpolation data record bit 0 .. 15, integer16 interpolation data record bit 16 .. 23, integer8
Var. type	setpoint or parameter for a function read and write
Unit	PSS
Value range	$-2^{24}..2^{24}-1$ PSS
Gültig	in axis operating mode ***interpolated position mode***

This variable can be used as an alternative for the 32-bit *interpolation data record* in order to map this record with 24 bit, only.

That way, the values for *interpolation data record*, *velocity offset*, and *control word* can be mapped into an 8-byte PDO together with the *alternative for velocity offset*.

Velocity offset	Index: `60b1`, Short name: `VelocityOffset`	
SPP Windows	–	
Type	simple variable, integer32	
Var. type	setpoint or parameter for a function	read and write
Unit	0.25 r.p.m.	
Standard value	–	
Valid	in axis operating mode ***interpolated position mode***	

If this variable is mapped in **interpolated position mode**, the velocity feedforward is taken from this value and the master has to supply valid values cyclically with the *IpolDataRecord*.

If this variable is not mapped, the velocity feedforward is calculated internally.

Alternative for velocity offset	Index: `6fc2`, Short name: `AltVelocityOffset`	
SPP Windows	–	
Type	record, 2 elements (subindex 0 .. 2)	
Record element 1 **Record element 2**	velocity offset bit 0 .. 15, integer16 velocity offset bit 16 .. 23, integer8	
Type	setpoint or parameter for a function	read and write
Var. type	0.25 r.p.m.	
Standard value	–	
Valid	in axis operating mode ***interpolated position mode***	

This variable can be used as an alternative for the 32-bit *velocity offset* in order to map this record with 24 bits, only.

That way, the values for *interpolation data record*, *velocity offset*, and *control word* can be mapped into an 8-byte PDO together with the *alternative for velocity offset*.

9.2.5 Parameters and Actual Values Timed Positioning Mode

The parameters and actual values of the **timed positioning mode** include the following variable:

- *Velocity feedforward factor*

In command and commissioning software SPP Windows, the parameters and actual values of the **timed positioning mode** can be found in the “Device control” window.

Velocity feedforward factor	Index: `5F64`, Subindex: `4`, Short name: `VelFFKp`
SPP Windows	–
Type	simple variable, float32
Var. type	setpoint or parameter for a function read and write
Unit	1.0
Standard value	1.0
Valid	in axis operating modes ***timed positioning mode***, ***cyclic synchronous velocity mode,*** and ***interpolated position mode***

With this variable, the velocity feedforward is influenced in the stated operating modes. The velocity feedforward can be switched off by setting the *velocity feedforward factor* to 0.

9.2.6 Parameters and Actual Values Cyclic Synchronous Torque/Force Mode

The parameters and actual values of the **cyclic synchronous torque/force mode** include the following variables:

- *Torque offset / force offset*

In command and commissioning software SPP Windows, the parameters and actual values of the **cyclic synchronous torque/force mode** can be found in the “Device control” window.

Torque offset / Force offset	Index: `60B2`, Short name: `TorqueOffset`
SPP Windows	–
Type	simple variable, integer16
Var. type	setpoint or parameter for a function read and write
Unit	0.1% of the *rated torque* or *force* of the motor
Standard value	0
Valid	in axis operating modes ***cyclic synchronous torque mode, cyclic synchronous velocity mode,*** and ***interpolated position mode***

With this variable, an additional torque or an additional force can be preset cyclically for acceleration feedforward.

9.2.7 Parameters and Actual Values Cyclic Synchronous Velocity Mode

The parameters and actual values of the **cyclic synchronous velocity mode** include the following variables:

- *Target velocity*
- *Velocity offset*

In command and commissioning software SPP Windows, the parameters and actual values of the **cyclic synchronous velocity** mode can be found in the “Device control” window.

Target velocity	Index: `60FF`, Short name: `TargetVelocity`
SPP Windows	–
Type	simple variable, integer32
Var. type	setpoint or parameter for a function read and write
Unit	VU
Standard value	–
Valid	in axis operating mode ***cyclic synchronous velocity mode***

With this variable, the target velocity is preset cyclically.

Velocity offset	Index: `60B1`, Short name: `VelocityOffset`
SPP Windows	–
Type	simple variable, integer32
Var. type	setpoint or parameter for a function read and write
Unit	VU
Standard value	0
Valid	in axis operating modes ***cyclic synchronous velocity mode*** and ***interpolated position mode***

With this variable, a velocity feedforward can be preset cyclically. The actually effective value results from the multiplication with the *velocity feedforward factor* (see Parameters and Actual Values Timed Positioning Mode 93).

9.2.8 Parameters and Actual Values Input/Output Function

The parameters and actual values of the input/output function include the following variables:

- *Digital inputs*
- *Digital outputs*
- *Digital inputs acc. to CiA402*
- *Analog inputs*
- *Analog outputs*

In command and commissioning software SPP Windows, the parameters and actual values of the input/output function can be found in the “Device control” window.

For further information on the input and output function see section Input/Output Function 61. The variables for the corresponding machine data are described in section Input/Output Function Machine Data[131].

Digital inputs	Index: 5f56, Short name: DEin
SPP Windows	`Digital inputs 1 00001110`
Type	array, 20 type unsigned8 elements
Array element 1	digital inputs I 1.x
Array element 2	digital inputs I 2.x
Array element 3	digital inputs I 3.x (I 3.0 = enable)
Array element 4	digital inputs I 4.x (software inputs)
Array element 5 .. 8	(reserved for future expansions)
Array element 9	digital inputs I 9.x (software inputs)
Array element 10	status word bit 0 .. 7 (I 10)
Array element 11	status word bit 8 .. 15 (I 11)
Array element 12 .. 14	(reserved for future expansions)
Array element 15 .. 20	linked inputs/outputs (read; write with O 15 .. O 20)
Var. type	actual value read only
Unit	–
Standard value	–
Valid	always

Via this variable, the state of the I x.x digital inputs can be queried. The individual inputs are assigned to bits in the above-mentioned array elements. I x.0 is assigned to bit 0, I x.1 to bit 1 etc. 1 in a bit indicates that +24 V exist at the input.

The function of the digital inputs is defined via variables *signal selection digital inputs ...* (see section Input/Output Function Machine Data[131]).

An input does not need to have a physical connector at the device. The status word, for example, can be queried via inputs I 10 and I 11, inputs I 15 to I 20 can be used as markers in the part program.

The state of the digital inputs is displayed independent of the configuration.

Digital outputs	Index: 5f54, Short name: DAus
SPP Windows	Digital outputs 1 00000110
Type	array, 20 type unsigned8 elements
Array element 1	digital outputs O 1.x
Array element 2	digital outputs O 2.x
Array element 3	digital outputs O 3.x (MaxiDrive: O 3.0 = brake control)
Array element 4 .. 7	(reserved for future expansions)
Array element 8 .. 9	digital outputs O 8.x, O 9.x (software outputs)
Array element 10	control word bit 0 .. 7 (O 10)
Array element 11	control word bit 8 .. 15 (O 11)
Array element 12 .. 14	(reserved for future expansions)
Array element 15 .. 20	linked inputs/outputs (read; write with I 15 .. O 20)
Var. type	setpoint or parameter for a function read and write
Unit	–
Standard value	depending on the configuration
Valid	always

Via this variable, the state of the O x.x digital outputs can be controlled. The individual outputs are assigned to bits in the above-mentioned array elements. O x.0 is assigned to bit 0, O x.1 to bit 1 etc. “1” in a bit switches +24 V to the output.

Only the outputs configured as freely usable in variable *signal selection digital outputs ...* can be controlled via this variable. For the other outputs, the state is defined via the selected signal (see section Input/Output Function Machine Data 131).

An output does not need to have a physical connector at the device. The control word, for example, can be changed via outputs O 10 and O 11, outputs O 15 to O 20 can be used as markers in the part program.

The state of the digital outputs can be read out independent of the configuration.

Digital Inputs Acc. To CiA402	Index: 60FD, Short name: DEin402
SPP Windows	–
Type	simple variable, unsigned32
Var. type	actual value read only
Unit	–
Standard value	–
Valid	always; V 8.5.997 or higher

Irrespective of the digital inputs configured as limit switch and/or pre-pulse, this variable displays the current state at the corresponding input.

Bit	Status
0	Limit switch–
1	Limit switch+
2	Pre-pulse

Value 1 indicates that the respective input function is active, i. e. bit 0 and bit 1 are inverted regarding the physical states at inputs limit switch+ and limit switch– (opener).

Analog inputs	Index: `5f52`, Short name: `AEin`
SPP Windows	`Analog inputs 1 9.9951`
Type	array, 12 type integer16 elements
Array element 1	analog input “Setpoint” or “Setpoint1”, V 4.4 or higher
Array element 2	analog input “Setpoint2”
Array element 3 .. 12	(reserved for future expansions)
Var. type	actual value read only
Unit	–10 V .. +10 V = –32768 .. +32767
Standard value	–
Valid	always, V 4.4 or higher; for devices with -A1 option

In element 1 of variable *analog inputs*, the value of analog input “setpoint” or “setpoint1” can be read. Element 2 contains the value of analog input “setpoint2”.

Please, pay attention to the technical specifications of analog input “setpoint” in operating instructions “Connection and Commissioning” (6755.202, 6745.202, 6750.202, 6740.202, or 6710.202). For options of using the “setpoint” analog input for setting setpoints, please read sections Analog Input 63 and Setpoint Source Machine Data 144.

Analog outputs	Index: `5f50`, Short name: `AAus`
SPP Windows	`Analog outputs 1 0`
Type	array, 4 type integer16 elements
Var. type	setpoint or parameter for a function read and write
Unit	–16384 .. +16383 = –5 V .. +5 V
Standard value	–
Valid	always; V 3.5 or higher; for devices with -A1 option

With the first two elements of variable *analog outputs*, voltages can be output at outputs “Ist1” and “Ist2”.

If an actual value or another signal was mapped to the corresponding analog output using variables *analog output 1 signal selection* or *analog output 2 signal selection*, its value determines the output voltage at output “Ist1” or “Ist2”.

9.2.9 Parameters and Actual Values Trigger and Measuring Functions

The parameters and actual values of the trigger and measuring functions include the following variables:

- *Trigger mask*
- *Trigger status*
- *Trigger mode selection*
- *Position sensor measured value1*
- *Position sensor measured value1 PU*
- *Position sensor measured value2*
- *Position sensor measured value2 PU*

In command and commissioning software SPP Windows, parameter *trigger mode selection* can be found in window “Parameterization/Device data”, the other variables are control and status functions which are not displayed.

Trigger mask	Index: `5f7f`, Short name: `TriggerMaske`	
SPP Windows	–	
Type	simple variable, octet string length 2	
Var. type	setpoint or parameter for a function	read and write
Unit	–	
Standard value	0	
Valid	always; V 3.2 or higher	

If bit 0 or bit 2 of the *trigger mask* is set to 1, it releases digital input “position measurement” or “fast position measurement” to measure the position of the axis in the *position sensor measured value...*. Bit 4 of the trigger mask releases digital input “trigger input trace” for tracing. After the successful position measurement or tracing, the corresponding bit in the trigger status is set.

The function then disables itself until the corresponding bit of the *trigger mask* is set to 0 and then reset to 1.

A description of the function of this variable in conjunction with *trigger status* and *position sensor measured value...* can be found in section Trigger and Measuring Functions 64.

The trace function is described in section Trace Function 66.

Trigger status	Index: `5f80`, Short name: `TriggerStatus`
SPP Windows	–
Type	simple variable, octet string length 2
Var. type	actual value read only
Unit	–
Standard value	0
Valid	always; V 3.2 or higher

If bit 0 or bit 2 in the *trigger status* changes to 1, this indicates that the current position was stored in *position sensor measured value....* With value 1, bit 4 in the *trigger status* indicates that triggering was initiated. The bit in the *trigger status* remains on 1 until a new measurement is released by a change from 1 to 0 and back to 1 in the corresponding bit of the *trigger mask*.

A description of the function of these variables in conjunction with *trigger mask* and *position sensor measured value...* can be found in section Trigger and Measuring Functions 64.

The trace function is described in section Trace Function 66.

Trigger mode selection	Index: `5fb1`, Short name: `TriggerModus`

SPP Windows	`Trigger mode position measurement` <u>`positive edge`</u> `select`
	`Trigger mode fast position measurement` <u>`positive edge`</u> `select`
	`Trigger mode trace function` <u>`positive edge`</u> `select`

Type	array, 16 type integer16 elements
Array element 1	trigger mode position measurement
Array element 3	trigger mode fast position measurement
Array element 5	trigger mode trace function
Var. type	parameter read and write
Unit	–
Standard value	positive edge
Valid	always; V 7.5 or higher

This variable is used for selecting the type of position measurement or the trigger for the trace function.

Permitted values:

Value	Function
0	positive edge
–1	negative edge
–2	positive pulse
–3	negative pulse

Position sensor measured value1	Index: `5f1c`, Short name: `PosAblage`	
SPP Windows	–	
Type	simple variable, integer32	
Var. type	actual value	read only
Unit	position sensor steps (PSS)	
Standard value	0	
Valid	always; V 3.2 or higher	

In case of a position measurement via trigger input “position measurement” or “fast position measurement”, the *actual position* of the axis is stored in this variable in position sensor steps.

In case of pulse triggering (“fast position measurement”, only), the actual position at the point of time of the first edge of the pulse is stored in this variable in position sensor steps.

Position measurement is released via bit 0 or bit 2 in the *trigger mask*, the measurement is displayed in bit 0 or bit 2 of the *trigger status*.

A description of the function of these variables in conjunction with *trigger mask* and *trigger status* can be found in section Trigger and Measuring Functions 64.

Position sensor measured value1 PU	Index: `5e9f`, Short name: `PosAblageLE`	
SPP Windows	–	
Type	simple variable, integer32	
Var. type	actual value	read only
Unit	position units (PU)	
Standard value	0	
Valid	always; V 3.2 or higher	

In case of a position measurement via trigger input “position measurement” or “fast position measurement”, the *actual position* of the axis is stored in this variable in position units.

In case of pulse triggering (“fast position measurement”, only), the actual position at the point of time of the first edge of the pulse is stored in this variable in position units.

Position measurement is released via bit 0 or bit 2 in the *trigger mask*, the measurement is displayed in bit 0 or bit 2 of the *trigger status*.

A description of the function of these variables in conjunction with *trigger mask* and *trigger status* can be found in section Trigger and Measuring Functions 64.

Position sensor measured value2	Index: `5e95`, Short name: `PosAblage2`
SPP Windows	–
Type	simple variable, integer32
Var. type	actual value read only
Unit	position sensor steps (PSS)
Standard value	–
Valid	always; V 7.5 or higher

In case of a position measurement via trigger input “fast position measurement”, the *actual position* of the axis at the point of time of the second edge of the pulse is stored in this variable in position sensor steps.

Position measurement is released via bit 0 or bit 2 in the *trigger mask*, the measurement is displayed in bit 0 or bit 2 of the *trigger status*.

A description of the function of these variables in conjunction with *trigger mask* and *trigger status* can be found in section Trigger and Measuring Functions 64.

Position sensor measured value2 PU	Index: `5e96`, Short name: `PosAblage2LE`
SPP Windows	–
Type	simple variable, integer32
Var. type	actual value read only
Unit	position units (PU)
Standard value	–
Valid	always; V 7.5 or higher

In case of a position measurement via trigger input “fast position measurement”, the *actual position* of the axis at the point of time of the second edge of the pulse is stored in this variable in position units.

Position measurement is released via bit 0 or bit 2 in the *trigger mask*, the measurement is displayed in bit 0 or bit 2 of the *trigger status*.

A description of the function of these variables in conjunction with *trigger mask* and *trigger status* can be found in Trigger and Measuring Functions 64.

9.2.10 Parameters of the Trace Function

The parameters of the trace function include the following variables:

- *Trace buffer 1*
- *Trace buffer 2*
- *Trace buffer 3*
- *Trace index 1*
- *Trace index 2*

- *Trace index 3*
- *Trace time*
- *Trace start*
- *Controller cycle time*

In command and commissioning software SPP Windows, the trace function is operated via the “Trace oscilloscope” window.

For further information on the trace function see section Trace Function 66.

Trace buffer 1	Index: 5f43, Short name: TraceBuffer1
SPP Windows	(used internally)
Type	array, 16 type octet string length 128 elements
Var. type	data range, mapped on an array read only
Unit	–
Standard value	0
Valid	always

Trace buffer 2	Index: 5f45, Short name: TraceBuffer2
SPP Windows	(used internally)
Type	array, 16 type octet string length 128 elements
Var. type	data range, mapped on an array read only
Unit	–
Standard value	0
Valid	always

Trace buffer 3	Index: 5f47, Short name: TraceBuffer3
SPP Windows	(used internally)
Type	array, 16 type octet string length 128 elements
Var. type	data range, mapped on an array read only
Unit	–
Standard value	0
Valid	always

Trace buffer 1 to *trace buffer 3* are memory areas for recording certain signals over time. The recording is started with signal *trace start* or via digital input “trigger input trace”.

Trace index 1	Index: `5f44`, Short name: `TraceIndex1`
SPP Windows	`Channel Actual current`
Type	simple variable, unsigned16
Var. type	setpoint or parameter for a function read and write
Unit	–
Standard value	actual current
Valid	always

Trace index 2	Index: `5f46`, Short name: `TraceIndex2`
SPP Windows	`Channel Actual speed`
Type	simple variable, unsigned16
Var. type	setpoint or parameter for a function read and write
Unit	–
Standard value	actual speed
Valid	always

Trace index 3	Index: `5f48`, Short name: `TraceIndex3`
SPP Windows	`Channel Actual position`
Type	simple variable, unsigned16
Var. type	setpoint or parameter for a function read and write
Unit	–
Standard value	actual position
Valid	always

Trace index 1 to *trace index 3* specify the signals to be recorded in the corresponding variables *trace buffer 1* to *trace buffer 3*. The digital or analog signals are selected via the selection codes as stated in *signal selection digital output ...* and *signal selection analog output ...*, see section Input/Output Function Machine Data 131.

Trace time	Index: `5f49`, Short name: `TraceTime`
SPP Windows	`Sampling rate (ms) 2`
Type	simple variable, unsigned16
Var. type	setpoint or parameter for a function read and write
Unit	controller cycles
Standard value	1
Valid	always

Trace time specifies the time cycle in which signals are recorded in *trace buf-*

fer 1, *trace buffer 2*, and *trace buffer 3* in controller cycle units (see *controller cycle time*). This value can be converted into time units using variable *controller cycle time*.

In SPP Windows, the trace time is stated in milliseconds, conversion is carried out automatically using variable *controller cycle time*.

Trace start	Index: `5f4a`, Short name: `TraceStart`
SPP Windows	“Trigger” button
Type	simple variable, octet string length 1
Var. type	control information — write only
Unit	–
Standard value	–
Valid	always

Writing on parameter *trace start* (no matter which value) starts the recording of signals in *trace buffer 1*, *trace buffer 2*, and *trace buffer 3*.

Alternatively, recording can be started via digital input “trigger input trace”, in this case, the start of the recording must be released via bit 4 in the trigger mask, before.

Controller cycle time	Index: `5f4b`, Short name: `ReglerZyklus`
SPP Windows	(used internally)
Type	simple variable, unsigned16
Var. type	adjustable value — read only
Unit	1 µs
Standard value	1 ms
Valid	always

This parameter permits reading out the *controller cycle time* of the position control loop, it can be used for calculations related to *trace time*.

In SPP Windows, the *controller cycle time* is used for converting the sampling rate in milliseconds.

9.3 Variable Descriptions Basic Data

9.3.1 General Actual and Measured Values

The general actual and measured values include the following variables:

- *DC-bus voltage*
- *Resolver voltage*

- *Heat sink temperature*
- *Motor temperature*
- *Resistance motor temperature sensor*
- *Shunt resistor loading*
- *I^2t loading*

In command and commissioning software SPP Windows, the general actual and measured values can be found in the “Device status” window.

Bus voltage	Index: `6079`, Short name: `UZwKreis`
SPP Windows	`Bus voltage 560 V`
Type	simple variable, unsigned16
Var. type	actual value read only
Unit	1 V
Standard value	–
Valid	always

This variable shows the DC-bus voltage in volts.

Resolver voltage	Index: `5f0a`, Short name: `UResolver`
SPP Windows	`Resolver voltage 2.122 V`
Type	simple variable, unsigned16
Var. type	actual value read only
Unit	1 mV
Standard value	–
Valid	always

The variable shows the resolver voltage in mV. The value should be in the range 2 V ± 10% and not deviate more than 5% for one motor revolution. The resolver voltage is monitored by the servo drive. If the voltage is too high or too low, the device switches off and reports an encoder fault.

The resolver voltage of other motor position sensors, if existing, is converted to a standard value of 2 V and shown in variable *resolver voltage*, too.

Heat sink temperature	Index: `5f0b`, Short name: `TKuehler`
SPP Windows	`Heat sink temperature 23.4 °C`
Type	simple variable, integer16
Var. type	actual value read only
Unit	0.1 °C
Standard value	–
Valid	always

The variable shows the temperature of the heat sink of the servo drive. The heat sink temperature is monitored by the servo drive. If the temperature is too high, the device switches off and reports an “overtemperature” fault.

Motor temperature	Index: `5f0c`, Short name: `TMotor`
SPP Windows	`Motor temperature 28.3 °C`
Type	simple variable, integer16
Var. type	actual value read only
Unit	0.1 °C
Standard value	–
Valid	always; V 8.5 and higher

In case a motor temperature sensor with known resistance temperature characteristic is used (e.g. KTY-83 or KTY-84), the temperature of the motor is displayed in this variable. The motor temperature is monitored by the servo drive. If the temperature is too high, the device switches off and reports an “overtemperature motor” fault.

Resistance motor temperature sensor	Index: `5f03`, Short name: R`Motor`
SPP Windows	`Resistance motor temperature sensor 615 Ohm`
Type	simple variable, unsigned32
Var. type	actual value read only
Unit	1Ω
Standard value	–
Valid	always; V 8.5 and higher

This variable shows the current resistance value used for monitoring the motor temperature. In case the *resistance switching threshold overtemperature* is exceeded, the device switches off and reports an “overtemperature motor” fault.

Shunt resistor loading	Index: `5f98`, Short name: `TBallast`
SPP Windows	`Shunt resistor loading 0.0 %`
Type	simple variable, integer16
Var. type	actual value — read only
Unit	0.1% of possible continuous output
Standard value	–
Valid	always; V 5.1 or higher

The *shunt resistor loading* shows the power currently consumed by the shunt resistor (internally or externally). The specification relates to the possible continuous output of the shunt resistor and is averaged via its thermal time constant. This way, the shunt resistor is also capable of short-term power consumption that is higher than the permissible continuous output.

If the shunt resistor loading exceeds 100%, the software monitoring switches off the power circuit and the device reports a fault.

If an external shunt resistor is used, resistance value, permissible continuous output, and thermal time constant must be entered in the shunt resistor machine data (see section Shunt Resistor Machine Data 129).

I^2t Loading	Index: `5E93`, Short name: `I2tAuslastung`
SPP Windows	`I²t loading 0.0 %`
Type	simple variable, unsigned16
Var. type	actual value — read only
Unit	0.1%
Standard value	–
Valid	always; V 8.5 and higher

This variable shows the “fill level” of the integrator for I^2t monitoring. From a value of 100% on, the servo drive limits the motor current to the motor rated current and displays an “overload” (LED at the front panel) in order to protect the motor.

9.3.2 Motor Machine Data

In command and commissioning software SPP Windows, the motor machine data can be found in the “Parameterization” window in *Motor data*.

During first commissioning with the SPP Windows command and commissioning software, make sure that the motor used is listed in the motor machine data.

Motor data that have been taken from a motor database may possibly not be chanced individually by the user. In this case, they are displayed in SPP Windows for information, only.

SPP Windows supports two procedures for setting the motor data: the selection of the motor via its type designation from a motor database and the individual input of the technical data of the motor in the corresponding machine data. Depending on the servo drive used (firmware version) and the motor, only one procedure may be available. For further information, see section Setting the Motor Data 67.

If the motor does not run as desired, make sure that the right motor has been selected using these variables.

9.3.2.1 General Motor and Current Control Loop Data

The motor machine data include the following variables:

- *Motor name*
- *Motor rated torque / motor rated force*
- *Motor rated current*
- *Motor data*
- *Rated speed*
- *Standstill torque / standstill force*
- *Max. continuous current*
- *Max. peak current*
- *Torque constant / force constant*
- *Resistance (P-P)*
- *Inductance (P-P)*
- *# of pole pairs / pitch*
- *Thermal time constant*
- *Moment of inertia rotor / mass actuator*
- *Electrical sense of rotation*
- *Position sensor mounting offset*
- *Current control loop P component*
- *Current control loop I component*
- *Angle correction controller P component*
- *Angle correction controller I component*
- *Angle correction controller D component*
- *Rotating field generator frequency*
- *Rotating field generator voltage*
- *PWM frequency power circuit*

In command and commissioning software SPP Windows, the motor machine data can be found in the “Parameterization” window in *Motor data*.

Motor name	Index: `5ee6`, Short name: `MotorName`
SPP Windows	`Motor name` `MR 7412U3N60G1` (without motor database) `Motor name` `MR 7412-U3-N060-G01` (with motor database)
Type	simple variable, visible string length 16
Var. type	motor machine data read and write / tool
Unit	–
Standard value	–
Valid	always

The motor designation (short type code) is entered in variable *motor name*. In case motor data from a motor database are used, this short description is evaluated by command and commissioning software SPP Windows. Using this information, the software displays the motor name as listed in the SPP Windows motor database. The motor name displayed in the motor selection and the motor data includes all parts of the motor designation identifying motor and encoder system:

- motor type and size
- DC-bus voltage
- rated speed
- external ventilation (if applicable)
- encoder system

Characters identifying equipment versions of the motor that have nothing to do with power and control characteristics (e. g. keys, foot-mounting, connectors, or terminal boxes) are not displayed.

Motor rated torque / force	Index: `6076`, Short name: `MNenn`
SPP Windows	`Rated torque` `0.700` `Nm` (for rotatory drive systems) `Rated force` `500.000` `N` (for linear motors)
Type	simple variable, unsigned32 (V 8.5.9 or higher) simple variable, unsigned16 (up to V 8.5.9)
Var. type	motor machine data read and write / tool
Unit	0.001 Nm or 0.001 N (V 8.5.9 or higher) 0.1 Nm (up to V 8.5.9)
Standard value	–
Valid	always

The rated torque or force of the motor is entered or displayed, here.

If the motor is operated with a servo drive with lower rated current (see following variable), the rated torque of the motor cannot be reached and a correspondingly lower value has to be entered.

Motor rated current	Index: 6075, Short name: INenn
SPP Windows	Rated current 2.0 A
Type	simple variable, unsigned32 (version V 8.5.9 or higher) simple variable, unsigned16 (up to V 8.5.9)
Var. type	motor machine data — read and write / tool
Unit	1 mA (V 8.5.9 or higher) 0,1 A (up to V 8.5.9)
Standard value	–
Valid	in all axis operating modes

The rated current of the motor is entered or displayed, here.

The rated current of the motor must not exceed the rated current of the servo drive. In case a motor is operated at a servo drive with lower rated current, a correspondingly lower value has to be entered, here. In this case, the rated torque of the motor cannot be reached.

Relative current specifications in other variables refer to the *motor rated current*, that means the value indicated, here.

Motor data	Index: 5E80, Short name: MotorData
Type	record, 19 elements
Var. type	motor machine data — read and write
Valid	in all axis operating modes; V 8.5.9 or higher

From firmware version V 8.5.9 on, the motor machine data are combined in record variable *motor data*. The individual elements are described in the following in the order in which they appear in the motor data of SPP Windows.

Rated speed	Index: 5E80, Subindex: 9, Short name: MotorData9
SPP Windows	Rated speed 6000.0 r.p.m. (for rotatory drive systems) Rated speed 3.50 m/s (for linear motors)
Type	record element, float32
Var. type	motor machine data — read and write
Unit	1 r.p.m. or 1 m/s
Standard value	–

In case the motor data are not accepted from a motor database, enter the rated speed of the motor, here.

Motor rated speed	Index: `5f19`, Short name: `VelNennMotor`
SPP Windows	`Rated speed 6000.0 r.p.m.`
Type	simple variable, unsigned32
Var. type	motor machine data — tool
Unit	0.25 r.p.m. if rated speed ≤ 7,000 r.p.m. 0.5 r.p.m. if 7,000 r.p.m. < rated speed ≤ 14,000 r.p.m. 1.0 r.p.m. if rated speed > 14,000 r.p.m.
Standard value	–
Valid	in all axis operating modes; V 3.2 to V 8.5.9

For devices with firmware versions below V 8.5.9, the rated speed of the motor has to be entered, here.

The machine data of the motor are factory preset. If, in special cases, new motor data has to be loaded, note the following:

- If the previous rated speed of the motor was lower than or equal to 7,000 r.p.m. and the new speed is higher than 7,000 r.p.m.

  or
- if the previous rated speed of the motor was higher than 7,000 r.p.m. but lower than or equal to 14,000 r.p.m. and the new speed is lower than or equal 7000 r.p.m. or higher than 14,000 r.p.m.

  or
- if the previous rated speed of the motor was higher than 14,000 r.p.m. and the new speed is lower than 14,000 r.p.m.
- save the new data in the non-volatile memory of the servo drive and
- switch off the device and switch it on again in order to make the new data valid.

Standstill torque / force	Index: `5E80`, Subindex: `11`, Short name: `MotorData11`
SPP Windows	`Standstill torque 0.90 Nm` (for rotatory drive systems) `Standstill force 600.00 N` (for linear motors)
Type	record element, float32
Var. type	motor machine data — read and write
Unit	1 Nm or 1 N
Standard value	–

Enter the standstill torque or standstill force of the motor, here.

Max. continuous current	Index: 5E80, Subindex: 4, Short name: MotorData4
SPP Windows	`Max. continuous current  2.000  A`
Type	record element, float32
Var. type	motor machine data read and write
Unit	1 A
Standard value	–

The *max. continuous current* corresponds to the current the motor can be operated with permanently under consideration of thermal limits. Usually, this value corresponds to the *motor rated current*.

Max. peak current	Index: 5E80, Subindex: 5, Short name: MotorData5
SPP Windows	`Max. peak current  8.500  A`
Type	record element, float32
Var. type	motor machine data read and write
Unit	1 A
Standard value	–

The *max. peak current* corresponds to the current the motor can be operated with for a short period of time without consideration of the thermal limits.

Torque constant / Force constant	Index: 5E80, Subindex: 12, Short name: MotorData12
SPP Windows	`Torque constant  0.33  Nm/A` (for rotatory drive systems) `Force constant  140.0  N/A` (for linear motors)
Type	record element, float32
Var. type	motor machine data read and write
Unit	1 Nm/A or 1 N/A
Standard value	–

Enter the torque constant or the force constant of the motor, here.

Resistance (P-P)	Index: 5E80, Subindex: 2, Short name: MotorData2
SPP Windows	`Resistance (P-P)  12.500  Ohm`
Type	record element, float32
Var. type	motor machine data read and write
Unit	1 Ω
Standard value	–

Enter the resistance phase-phase, here. (If the resistance of only one phase is specified in the motor data sheet, multiply the value by 2 and enter the result, here.)

Inductance (P-P)	Index: 5E80, Subindex: 3, Short name: MotorData3
SPP Windows	`Inductance (P-P) 45.300 mH`
Type	record element, float32
Var. type	motor machine data — read and write
Unit	1 H
Standard value	–

Enter the inductance phase-phase, here. (If the inductance of only one phase is specified in the motor data sheet, multiply the value by 2 and enter the result, here.)

Number of pole pairs	Index: 5E80, Subindex: 6, Short name: MotorData6
SPP Windows	`# of pole pairs 3`
Type	record element, unsigned16
Var. type	motor machine data — read and write
Unit	–
Standard value	–
Valid	always; for rotatory drive systems, only

Enter the number of pole pairs for one full revolution of the motor. (If only the number of poles is stated in the motor data sheet, divide the value by 2 and enter the result, here.)

Pitch	Index: 5E80, Subindex: 19, Short name: MotorData19
SPP Windows	`pitch 12.0 mm`
Type	record element, float32
Var. type	motor machine data — read and write
Unit	1 m
Standard value	–
Valid	always; for linear motors, only

Enter the magnet pitch (center distance between two adjacent magnet poles N-S of the secondary part of the linear motor) here. (If magnet pitch N-N is defined in the motor data sheet, divide it by 2 and enter the result here.)

Thermal time constant	Index: `5E80`, Subindex: `8`, Short name: `MotorData8`
SPP Windows	`Thermal time constant` <u>`600.0`</u> `s`
Type	record element, float32
Var. type	motor machine data read and write
Unit	1 s
Standard value	–

In case the motor data cannot be taken from a motor database, enter the thermal time constant of the motor here. (If the thermal time constant is entered in minutes in the motor data sheet, multiply the value by 60 and enter the result here.)

The thermal time constant determines the overload capacity of the motor, it is considered for the I^2t monitoring. In case of an overload, the peak current increase is switched back to the motor continuous current. This is indicated by the “Überlast/Overload” or “OVL” LED and can be signaled at output “overload”.

Thermal time constant	Index: `5f05`, Short name: `ITk`
SPP Windows	`Thermal time constant` <u>`10000`</u> `ms`
Typ	simple variable, unsigned16
Art	motor machine data tool
Einheit	1 ms
Standardwert	–
Gültig	in all axis operating modes; up to V 8.5.9

For devices with firmware versions below V 8.5.9, the thermal time constant of the motor is entered, here.

Moment of inertia rotor / mass actuator Index: `5E80`, Subindex: `13`, Short name: `MotorData13`	
SPP Windows	`Moment of inertia rotor` <u>`1.96 E-005`</u> `kgm²` (rotatory drives) `Mass actuator` <u>`2.0`</u> `kg` (linear motors)
Type	record element, float32
Var. type	motor machine data read and write
Unit	1 kgm^2 (rotatory) or 1 kg (linear)
Standard value	–

Enter the moment of inertia of the rotor or the mass of the primary part here. (If the moment of inertia is stated in 10^{-3} kg m^2 in the motor data sheet, divide the value by 1,000 and enter the result here. If the moment of inertia is stated in 10^{-4} kg m^2 or in kg cm^2 in the motor data sheet, divide the value by 10,000 and enter the result here.)

Electrical sense of rotation	Index: `5E80`, Subindex: `20`, Short name: `MotorData20`
SPP Windows	`Electrical sense of rotation positive select`
Type	record element, integer16
Var. type	motor machine data read and write
Unit	–
Standard value	1

The "natural" sense of rotation of the motor results from the current feed of the motor phases U, V, and W (in this order) with three-phase current. If this positive sense of rotation corresponds with your application, select "positive" for the *electrical sense of rotation*, otherwise "negative".

Permissible values are:

Value	Electrical sense of rotation
1	positive
–1	negative

Position sensor mounting offset Index: `5E80`, Subindex: `7`, Short nname: `MotorData7`	
SPP Windows	`Position sensor mounting offset 0.00 °` (rotatory drives) `Position sensor mounting offset 0.0 mm` (linear motors)
Type	record element, float32
Var. type	motor machine data read and write
Unit	1° (rotatory) or 1 m (linear)
Standard value	–

If the zero point of the motor position sensor does not correspond with the zero point of the phase order U-V-W of the motor, the offset can be entered here. The entered value is also used if "mounting offset at current position" was defined as *commutation reference*.

Current control loop P component Index: `5E80`, Subindex: `14`, Short name: `MotorData14`	
SPP Windows	`Current control loop P component 720`
Type	record element, unsigned16
Var. type	motor machine data read and write
Unit	–
Standard value	0

Current control loop I component	
	Index: 5E80, Subindex: 15, Short name: MotorData15
SPP Windows	Current control loop I component 140
Type	record element, unsigned16
Var. type	motor machine data read and write
Unit	–
Standard value	0

These parameters define the proportional amplification and the effect of the integration of the digital current control loop as determined during the setting of the current control loop. For further information, see section Setting the Current Control Loop 71.

Angle correction controller P component	
	Index: 5E80, Subindex: 16, Short name: MotorData16
SPP Windows	Angle correction controller P component 1
Type	record element, unsigned16
Var. type	motor machine data read and write
Unit	–
Standard value	1

Angle correction controller I component	
	Index: 5E80, Subindex: 17, Short name: MotorData17
SPP Windows	Angle correction controller I component 1
Type	record element, unsigned16
Var. type	motor machine data read and write
Unit	–
Standard value	1

Angle correction controller D component	
	Index: 5E80, Subindex: 18, Short name: MotorData18
SPP Windows	Angle correction controller D component 1
Type	record element, unsigned16
Var. type	motor machine data read and write
Unit	–
Standard value	1

(Description in preparation)

Rotating field generator frequency Index: 5E80, Subindex: 21, Short name: MotorData21	
SPP Windows	Rotating field generator frequency 2.0 Hz
Type	record element, integer16
Var. type	motor machine data read and write
Unit	0.1 Hz
Standard value	2.0 Hz
Valid	for rotatory drive systems, only

Rotating field generator voltage Index: 5E80, Subindex: 22, Short name: MotorData22	
SPP Windows	Rotating field generator voltage 4.0 % PWM
Type	record element, unsigned16
Var. type	motor machine data read and write
Unit	0.1 % of the DC-bus voltage
Standard value	4.0 %
Valid	for rotatory drive systems, only

With these two variables, the rotating field is defined if “zero search using rotating field generator” was define as *commutation reference*.

For motors with a very low inductance, a higher value for the frequency and/or a lower value for the voltage might be required.

PWM frequency power circuit	Index: 5F04, Short name: FPwmAuswahl
SPP Windows	PWM frequency power circuit 16 kHz select
Type	simple variable, integer16
Var. type	motor machine data read and write
Unit	–
Standard value	16 kHz

With this variable, you can select the switching frequency of the servo drive power circuit.

Permissible values are:

Value	PWM Frequency Power Circuit
1	16 kHz
0	8 kHz

9.3.2.2 Motor Position Sensor Machine Data

The motor position sensor machine data include the following variables:

- *Encoder data*

- *Measuring steps per revolution (for rotatory drive systems, only)*
- *Sine periods per revolution (for rotatory drive systems, only)*
- *Number of coded revolutions (for rotatory drive systems, only)*
- *Encoder type code*
- *Interface identification*
- *Manufacturer identification*
- *Position transmission motor sensor*
- *Measuring length*
- *Encoder data*
- *Position sensor selection code*
- *Encoder name*

In command and commissioning software SPP Windows, the motor machine data can be found in the “Parameterization” window in *Motor data / Motor position sensor*. Variables *position sensor selection code* and *encoder name* can be found in window “Operate / Status”.

Encoder data	`Index: 5E81`, Short name: `EncoderData`
Type	record, 8 elements
Var. type	motor machine data tool
Valid	always; V 8.5.9 or higher

From firmware version V 8.5.9 on, the machine data of the motor position sensor are grouped in record variable *encoder data*. The individual elements are described in the following.

Measuring steps per revolution	`Index: 5E81`, Subindex: `1`, Short name: `EncoderData1`
SPP Windows	`Measuring steps per revolution 0`
Type	record element, unsigned32
Var. type	motor machine data tool
Unit	1 increment
Standard value	–
Valid	always; for rotatory drive systems, only

For absolute encoders (e. g. Sincos (Hiperface) encoders or EnDat encoders), the number of measuring steps (position values) per revolution of the connected motor position sensor is displayed. Example: 262,144 (18 bit), 8,388,608 (23 bit), or 33,554,432 (25 bit).

For pure incremental encoders, this variable does not have any meaning.

Sine periods per revolution	`Index: 5E81`, Subindex: `2`, Short name: `EncoderData2`
SPP Windows	`Sine periods per revolution 1`
Type	record element, unsigned32
Var. type	motor machine data read and write / tool
Unit	–
Standard value	–
Valid	always; for rotatory drive systems, only

For motor position sensors with sine-cosine signals, the number of sine periods per revolution is entered here. For Sincos (Hiperface) and EnDat encoders, this entry is carried out automatically. For motor position sensors with square-wave signals, the number of pulses per revolution must be entered. Examples: 32, 512, 2048, 5000.

Number of coded revolutions	`Index: 5E81`, Subindex: `3`, Short name: `EncoderData3`
SPP Windows	`Number of coded revolutions 1`
Type	record element, unsigned16
Var. type	motor machine data read and write / tool
Unit	–
Standard value	–
Valid	always; for rotatory drive systems, only

For singleturn encoders, 1 is entered here, for multiturn encoders the number of revolutions that can be acquired. For Sincos (Hiperface) encoders and EnDat encoders, the entry is carried out automatically.

Encoder type code	`Index: 5E81`, Subindex: `4`, Short name: `EncoderData4`
SPP Windows	`Encoder type code 0000`
Type	record element, unsigned16
Var. type	motor machine data tool
Unit	–
Standard value	–

Identifies the used encoder system. Permissible values (others on request):

Value	Encoder type
0000	Encoder type not defined (connected encoder was not recognized)
0001	Unknown encoder type
0002	Resolver, standard, 2-pin
010f	Rotatory Hiperface encoder
110f	Hiperface enocer for linear axes
020f	Rotatory sine-cosine encoder with Z-track

Value	Encoder type
120f	Sine-cosine encoder with Z-track or comparable (e. g. analog hall sensors) for linear axes
030f	Rotatory EnDat encoders
130f	EnDat encoder for linear scales (absolute)
0a0f	Sine-cosine encoder without Z track[1)]
1a0f	Sine-cosine encoder without Z track or hall sensors for linear axes[1)]
0c0f	Rotatory square-wave incremental encoder[2)]
1c0f	Square-wave incremental encoder for linear axes[2)]
320f	Sine-cosine encoder with digital hall sensors for linear axes
3d0f	Square-wave incremental encoders with digital hall sensors for linear axes

[1)] Position sensor without absolute position information (autocommutation or similar required)

[2)] Position sensor with zero pulse, without absolute position information (autocommutation, zero pulse search or similar required)

Interface identification	Index: `5E81`, Subindex: `5`, Short name: `EncoderData5`
SPP Windows	`Interface identification 0`
Type	record element, unsigned16
Var. type	motor machine data Tool
Unit	–
Standard value	depends on the -Rx option of the servo drive

The *interface identification* is set by the manufacturer.

Permissible values:

Value	Interface
01	R1 = Resolver
02	R2 = Sincos (Hiperface) encoder (single- or multiturn)
03	R3 = High-resolution incremental encoder (sine-cosine signals)
04	R4 or R6 = EnDat encoder (single- or multiturn, or linear scale), R7 = BiSS
05	R5 = Incremental encoder with square-wave signals RS 422

Manufacturer identification	Index: `5E81`, Subindex: `6`, Short name: `EncoderData6`
SPP Windows	`Manufacturer identification undefined select`
Type	record element, integer16
Var. type	motor machine data tool
Unit	–
Standard value	depending on the motor position sensor used
Valid	always

(Description in preparation)

Permissible values:

Value	Transmission type
0	undefined
32768	EnDat
32771	BiSS-C unidir.
2	Hiperface SCS70
7	Hiperface SCM70
34	Hiperface SRS50
39	Hiperface SRM50
50	Hiperface SKS36
55	Hiperface SKM36
66	Hiperface SEK52

Position transmission motor sensor Index: 5E81, Subindex: 9, Short name: EncoderData9	
SPP Windows	`Position transmission motor sensor` <u>`Auto (best resolution)`</u>
Type	record element, integer16 read and write
Unit	select
Standard value	-2 = automatic (best resolution)
Valid	always; for option -R6 and -R7, only

This parameter can be used for defining whether the digital or the analog signals of the position sensor should be evaluated for the actual position:

Permissible values:

Value	Transmission type
– 2	Automatic (best resolution)
– 1	Fully digital position evaluation
0	Interpolate position from sine - cosine signals

Measuring length Index: 5E81, Subindex: 10, Short name: EncoderData10	
SPP Windows	`Measuring length` <u>`1000.0`</u> `mm`
Type	record element, float32
Var. type	motor machine data read and write
Unit	1 µm
Standard value	–
Valid	always; for linear motors only

The *measuring length* of the linear scale is the upper limit for the maximally possible travel distance of the linear motor.

Encoder data	Index: `5f3d`, Short name: `EncDat`
SPP Windows	`Resolution per revolution 0`
Type	array, 6 type u32 elements
Array element 1	resolution per revolution (digital) (increments/revolution)
Array element 2	sine periods per revolution (number sines/revolution)
Array element 3	number of coded revolutions
Array element 4	encoder type code
Array element 5	interface identification
Array element 6	encoder manufacturer ID
Var. type	motor machine data tool
Unit	–
Standard value	–
Valid	in all axis operating modes, V 7.0 to V 8.5.9

In devices with firmware version V 7.0 to V 8.5.9, the data of the motor position sensor are entered, here.

Position sensor selection code	Index: `5f09`, Short name: `EncAuswahl`
SPP Windows	`Position sensor selection code Resolver select`
Type	simple variable, integer16
Var. type	axis machine data read only
Unit	select
Standard value	–
Valid	in all axis operating modes

The *position sensor selection code* shows the motor position sensor used. On the basis of the installed module (options R1, R2, R3, or R4), the servo drive recognizes automatically whether a resolver, a Sincos (Hiperface), a high-resolution incremental encoder, or an EnDat encoder is used.

This parameter shows the encoder used for commutation. The position sensor used for position control in axis operating modes **profile position mode** and **homing mode** must be set in the position sensor source.

Permitted values:

Value	Position sensor type
1	Resolver
2	Sincos (Hiperface) encoder
3	High-resolution incremental encoder
4	EnDat encoder

Encoder name	Index: `5f3c`, Short name: `EncName`
SPP Windows	`Designation encoder system` <u>`resolver`</u>
Type	simple variable, visible string length 40
Var. type	axis machine data tool
Unit	–
Standard value	–
Valid	always; V 6.7 or higher

The *encoder name* in the device control window shows a detailed designation of the used motor position sensor, it may contain more detailed information than the *position sensor selection code*.

9.3.2.3 Motor Temperature Sensor Machine Data

The motor temperature sensor machine data include the following variables:

- *Motor temperature sensor data*
- *Temperature sensor type code*
- *Resistance switching threshold overtemp.*
- *Temperature switching threshold overtemp.*

Motor temperature sensor data	Index: `5F78`, Short name: `MTempSensorData`
Type	record, 4 elements
Var. type	motor machine data read and write / tool
Valid	always, V 8.5.3 and higher

The machine data of the motor temperature sensor are grouped in record variable *motor temperature sensor data*. The individual elements are described in the following.

Temperature sensor type code	
	`Index: 5F78`, Subindex: `1`, Short name: `MTempSensorData1`
SPP Windows	`Temperature sensor type code  PTC Sensor`
Type	record element, unsigned16
Var. type	motor machine data read and write / tool
Unit	select
Standard value	–

Permissible values:

Value	Temperature sensor type
1	PTC sensor
2	generical sensor with negative switching threshold
3	generical sensor with positive switching threshold
10	KTY83
11	KTY84

Resistance switching threshold overtemp.	
	`Index: 5F78`, Subindex: `2`, Short name: `MTempSensorData2`
SPP Windows	`Resistance switching threshold overtemp.  2500  Ohm`
Type	record element, unsigned16
Var. type	motor machine data read and write / tool
Unit	1 Ω
Standard value	–

In case the *resistance motor temperature sensor* exceeds the value stated here, the device switches off and reports error "overtemperature motor".

Temperature switching threshold overtemp.	
	`Index: 5F78`, Subindex: `3`, Short name: `MTempSensorData3`
SPP Windows	`Temperature switching threshold overtemp.  0  °C`
Type	record element, unsigned16
Var. type	motor machine data read and write / tool
Unit	0.1 °C
Standard value	–

In case the *motor temperature* exceeds the value stated here, the device switches off and reports error "overtemperature motor".

9.3.3 Servo Drive Machine Data

The servo drive machine data include the following variables:

- *Serial number*
- *Operating time*
- *Drive rated current*
- *Power monitoring*
- *Mains connection selection code*

In command and commissioning software SPP Windows, the servo drive machine data can be found in the “Device status” window.

The servo drive machine data are factory preset. They must not be changed by the user. They are displayed in SPP Windows for the information of specially trained service personnel, only.

If the motor does not run as desired, make sure that the correct *drive rated current* (0.8, 2, 4, 8, 12, 16, 20, 32 A, depending on the device variant) has been selected for your motor and your application.

Serial number	Index: –, Short name: `SerienNummer`
SPP Windows	`Serial number 631234`
Type	simple variable, unsigned32
Var. type	servo drive machine data tool
Unit	–
Standard value	–
Valid	always

The factory-preset serial number of the device can be read out here.

The serial number is also stated on the nameplate.

Operating time	Index: –, Short name: `OperatingTime`
SPP Windows	`Operating time total 20.0 h` `Operating time power circuit 10.0 h`
Type	array, 2 type unsigned32 elements
Array element 1	operating time total
Array element 2	operating time power circuit
Var. type	actual value tool
Unit	1 s
Standard value	–
Valid	always; V 8.5.5 to V 8.5.9 as well as V 8.5.998 or higher

These variables count the total operating time (14 V control supply voltage on) and the operating time of the power circuit (mains voltage and PWM). They are stored in the non-volatile memory and thus still available after switch-off and switch-on.

Drive rated current	Index: –, Short name: `INennVerstaerker`
SPP Windows	`Drive rated current 8.0 A`
Type	simple variable, unsigned16
Var. type	servo drive machine data tool
Unit	0.1 A
Standard value	–
Valid	in all axis operating modes

Here, you can read out the rated current of the servo drive, as it is factory preset according to the device version (0.8, 2, 4, 8, 12, 16, 20, or 32 A).

The *drive rated current* can also be found in the type designation on the name-plate. For that, see the type code of the servo drives in the corresponding operating instructions “Connection and Commissioning”.

Power monitoring	Index: –, Short name: `PwrMonitor`
SPP Windows	–
Type	record, 5 elements
Var. type	actual value tool
Unit	–
Standard value	–
Valid	always; V 8.5.4 or higher

The information on the power monitoring is summarized in record variable *power monitoring*. The individual elements are described in the following:

Active PWM frequency power circuit	Subindex: 1, Short name: `PwrMonitor1`
SPP Windows	`Active PWM frequency power circuit 16 kHz select`
Type	record element, integer16
Var. type	actual value tool
Unit	select
Standard value	–

This variable shows the current switching frequency of the servo drive power circuit as can be set via the motor data.

Possible values:

Value	PWM frequency power circuit
1	16 kHz
0	8 kHz

Power limit	Subindex: 2 to 5, Short name: `PwrMonitor2 to PwrMonitor5`
SPP Windows	`Continuous current limit servo drive` 2.0 A `Peak current limit servo drive` 6.0 A `Active continuous current limit` 1.6 A `Active peak current limit` 3.2 A
Type	4 type unsigned16 record elements
Var. type	actual value tool
Unit	0.1 A
Standard value	–

These variables show the maximally possible and the currently set power limits for continuous current and peak current. The first two values are specified by rated current and peak current of the servo drive. The active power limits result from the rated current of the motor and the settings for *continuous current motor* and *max current amount* in the axis data.

Operating modes option	Index: –, Short name: `BAFreigabeStatus`
SPP Windows	`Operating modes option` B2
Type	simple variable, unsigned32
Var. type	servo drive machine data tool
Unit	–
Standard value	–
Valid	always

This variable contains the value of the operating modes option of the servo drive in binary coding. This value is included in the type code of the servo drive and can also be read from the type code.

Mains connection selection code	Index: –, Short name: `NetzanschlAuswahl`
SPP Windows	`Mains connection selection code` 400V/3-phase
Type	simple variable, integer16
Var. type	servo drive machine data tool
Unit	–
Standard value	–
Valid	always; V 5.2 or higher

The mains connection with which the servo drive is equipped can be read out here.

Permitted values (automatically recognized and displayed):

Value	Mains connection
–5	90 V three-phase
–4	230 V single-phase
–3	230 V three-phase
–2	400 V three-phase
–1	480 V three-phase

The mains connection can also be found in the type designation on the name-plate:

Type:	**Mains connection:**
BN 675x	230 V single-phase (TrioDrive D/xS, TrioDrive D)
BN 674x	400 V three-phase (MidiDrive D/xS, MidiDrive D)
BN 672x	400 V three-phase (MaxiDrive)

9.3.4 Shunt Resistor Machine Data

The shunt resistor machine data include the following variables:

- *Shunt resistor selection code*
- *Resistance external shunt resistor*
- *Continuous power external shunt resistor*
- *Thermal time constant external shunt resistor*

In command and commissioning software SPP Windows, the shunt resistor machine data can be found in the “Parameterization” window under *Device data/ Shunt resistor*.

Shunt resistor selection code	Index: `5f94`, Short name: `BallastAuswahl`
SPP Windows	`Shunt resistor selection internal select`
Type	simple variable, integer16
Var. type	device machine data — read and write
Unit	select
Standard value	internal
Valid	always; V 5.1 or higher

This variable is used for determining whether the internal shunt resistor is used or an external shunt resistor will be connected.

For using the internal shunt resistor, a bridge must be plugged at connector X1 (TrioDrive D/xS, TrioDrive D), X6 (MidiDrive D/xS, MidiDrive D), or X9 (MaxiDrive), see operating instructions “Connection and Commissioning” (6755.202, 6750.202, 6745.202, 6740.202, or 6710.202).

Permitted values:

Value	Used shunt resistor
0	internal
1	external
2	shunt circuit off (for devices with DC-bus coupling, only)

Resistance external shunt resistor	Index: `5f95`, Short name: `ExtBallastR`
SPP Windows	`Resistance ext. shunt resistor  68.0  Ohm`
Type	simple variable, unsigned32
Var. type	device machine data read and write
Unit	0.1 Ω
Standard value	68 Ω
Valid	always; V 5.1 or higher

If an external shunt resistor is used, its resistance must be specified in *resistance external shunt resistor*. This value is required for calculating the shunt resistor loading and thus the software monitoring of the shunt resistor.

Continuous power ext. shunt resistor	Index: `5f96`, Short name: `ExtBallastPd`
SPP Windows	`Continuous power ext. shunt resistor  300  Watt`
Type	simple variable, unsigned16
Var. type	device machine data read and write
Unit	1 W
Standard value	300 W
Valid	always; V 5.1 or higher

If an external shunt resistor is used, its permissible continuous power must be specified in this variable. This value is required for calculating the shunt resistor loading and thus the software monitoring of the shunt resistor.

Thermal time constant ext. shunt resist. Index: `5f97`, Short name: `ExtBallastTk`	
SPP Windows	`Thermal time constant ext. shunt resist.  208  s`
Type	simple variable, unsigned16
Var. type	device machine data read and write
Unit	1 s
Standard value	208 s
Valid	always; V 5.1 or higher

If an external shunt resistor is used, its thermal time constant must be specified in this variable. This value is required for calculating the shunt resistor loading and thus the software monitoring of the shunt resistor.

9.3.5 Input/Output Function Machine Data

The machine data of the input/output function include the following variables:

- *Signal selection digital inputs*
- *Control word user defined bit*
- *Signal selection digital outputs*
- *Status word user defined bit*
- *Linked inputs/outputs*
- *Analog output 1 signal selection*
- *Analog output 2 signal selection*
- *Analog output 1 FP factor*
- *Analog output 2 FP factor*
- *Analog output 1 Int factor*
- *Analog output 2 Int factor*
- *Limit frequency current filter*

In command and commissioning software SPP Windows, the machine data of the input/output function can be found in window “Parameterization” under *Device data*.

Signal selection digital inputs 1.x Index: `5faa`, Short name: `DEin1Auswahl`	
SPP Windows	`Signal selection digital inputs 1.0  reset fault`
Type	array, 8 type integer16 elements
Array element 1 .. 8	configuration digital inputs I 1.0 .. I 1.7
Var. type	device machine data read and write
Unit	select
Standard value	reset fault (I 1.0), pre-pulse (I 1.1), enable saw (I 1.4), start running back (I 1.5), measure position (I 1.6)
Valid	always (effect depending on axis operating mode); V 6.0 or higher

Signal selection digital inputs 9.0	Index: `5fac`, Short name: `DEin9Auswahl`
SPP Windows	`Signal selection digital input 9.0` <u>`for free use`</u>
Type	array, 8 type integer16 elements
Array element 1 .. 8	configuration digital inputs I 9.0 .. I 9.7
Var. type	device machine data — read and write
Unit	select
Standard value	for free use
Valid	always (effect depending on axis operating mode); V 6.0 or higher

With variables *signal selection digital inputs ...*, certain drive functions can be assigned to the corresponding digital inputs (see table below). The value entered in the respective variable determines the drive system function.

Control word user defined bit	Index: `5fa8`, Short name: `DAus11Auswahl`
SPP Windows	`Control word user defined bit 11` <u>`for free use`</u>
Type	array, 8 type integer16 elements
Array element 4 .. 6	control word user defined bit 11 .. 13
Var. type	device machine data — read and write
Unit	select
Standard value	for free use
Valid	always

The *control word user defined bits 11 .. 13* permit an assignment of certain functions to bits 11, 12, and 13 in the *axis control word*.

The following functions can be selected:

Selection code		**Signal designation**	**Remarks**
–1	$ffff_{hex}$	for free use	–
102	66_{hex}	spindle positioning	in axis operating modes ***velocity mode*** and ***velocity mode direct***
110	$6e_{hex}$	limit switch +	(opener)
111	$6f_{hex}$	limit switch –	(opener)
112	70_{hex}	pre-pulse	in ***homing mode***
113	71_{hex}	reset fault	–
119	77_{hex}	synchronize electronic gearing	in axis operating mode ***electronic gearing,*** can be assigned to I 1.4, only
114	72_{hex}	enable saw	in axis operating mode ***flying shear***

Selection code		Signal designation	Remarks
115	73_{hex}	start running back	in axis operating mode ***flying shear***
116	74_{hex}	position measurement	–
135	87_{hex}	fast position measurement	can be assigned to I 1.4, only
117	75_{hex}	mains voltage reduced	mains voltage reduced to 90 V, the internal monitoring thresholds are decreased so that operation can be enabled
140	$8c_{hex}$	positioning mode bit 0 (positioning direction)	for ***timed positioning mode*** 0: clockwise 1: counterclockwise
141	$8d_{hex}$	positioning mode bit 1	for ***timed positioning mode*** 0: simple movement 1: back and forth movement
142	$8e_{hex}$	positioning mode bit 2	(reserved)
143	$8f_{hex}$	positioning mode bit 3	for ***timed positioning mode*** 0: single step 1: continuous running
144	90_{hex}	start positioning	for ***timed positioning mode***
148	94_{hex}	selection max. current amount	switches between *max current amount* and *max current amount 2*
150	96_{hex}	trigger input trace	for the event-depending start of a trace recording
151	97_{hex}	enable pulse acquisition L1	–
170	aa_{hex}	trigger release position measurement	corresponds to bit 0 in the *trigger mask*
172	ac_{hex}	trigger release fast position measurement	corresponds to bit 2 in the *trigger mask*
174	ae_{hex}	trigger release trace	corresponds to bit 4 in the *trigger mask*
169	$a9_{hex}$	stop axis	according to *axis stop selection code*
178	$b2_{hex}$	flying shear external synchronization	in axis operating mode ***flying shear***
179	$b3_{hex}$	enable external synchronization	in axis operating mode ***flying shear***

A drive function must not be assigned to more than one digital input. If a drive function is assigned to two or more inputs, the behavior of this function is undefined if different values exist at the inputs.

Some drive functions cannot be assigned to any input for technical reasons, these functions are assigned to I 1.4.

Signal selection digital outputs 1.x	Index: `5f99`, Short name: `DAus1Auswahl`
SPP Windows	`Signal selection digital output 1.0` <u>`overload`</u> `select`
Type	array, 8 type integer16 elements
Array element 1 .. 8	configuration digital outputs O 1.0 .. O 1.7
Var. type	device machine data read and write
Unit	select
Standard value	overload (O 1.0), setpoint reached (O 1.1), position monitoring (O 1.2), synchronized/apply saw (O 1.3)
Valid	always; V 5.1 or higher

Signal selection digital outputs 8.x	Index: `5fa6`, Short name: `DAus8Auswahl`
SPP Windows	`Signal selection digital output 8.0` <u>`for free use`</u>
Type	array, 8 type integer16 elements
Array element 1 .. 8	configuration digital outputs O 8.0 .. O 8.7
Var. type	device machine data read and write
Unit	select
Standard value	for free use
Valid	always; V 5.1 or higher

Signal selection digital outputs 9.x	Index: `5fa7`, Short name: `DAus9Auswahl`
SPP Windows	`Signal selection digital output 9.0` <u>`for free use`</u>
Type	array, 8 type integer16 elements
Array element 1 .. 8	configuration digital outputs O 9.0 .. O 9.7
Var. type	device machine data read and write
Unit	select
Standard value	for free use
Valid	always; V 5.1 or higher

Certain signals can be output at the corresponding digital outputs via variables *signal selection digital outputs ...* (see table below).

Status word user defined bit	Index: `5fad`, Short name: `DEin11Auswahl`
SPP Windows	`Status word user defined bit 8` <u>`for free use`</u>
Type	array, 8 type integer16 elements
Array element 1 .. 2	status word user defined bit 8 and 9
Var. type	device machine data read and write
Unit	select
Standard value	for free use
Valid	always

Status word user defined bits 8 and 9 can be used for assigning certain functions to bits 8 and 9 in the axis status word.

The following signals can be selected:

Selection code		Signal designation	Remarks
–1	$ffff_{hex}$	for free use	–
45	$2d_{hex}$	fault	bit 3 in the *axis status word*
46	$2e_{hex}$	setpoint reached	bit 10 in the *axis status word*
47	$2f_{hex}$	brake	operates the holding brake of the motor according to *release brake selection code*
48	30_{hex}	synchronized/apply saw	in axis operating modes ***electronic gearing*** and ***flying shear***
49	31_{hex}	overload	=1 if the peak current rise switches back to *continuous current motor*
145	91_{hex}	in position	for ***timed positioning mode***
50	32_{hex}	position monitoring	according to *position monitoring* and *position monitoring time*
51	33_{hex}	waste signal	in axis operating mode ***flying shear***
52	34_{hex}	velocity = 0	in axis operating modes ***velocity mode direct***, ***velocity mode***, and ***electronic gearing***; bit 12 in the *axis status word*
53	35_{hex}	spindle position reached	with function “spindle positioning” in axis operating modes ***velocity mode direct*** and ***velocity mode***
100	64_{hex}	ready (BTB)	= 1 if the DC-bus voltage is available **and** the servo drive is not in state fault
60	$3c_{hex}$	switching point 1	–
61	$3d_{hex}$	switching point 2	–
62	$3e_{hex}$	switching point 3	–
63	$3f_{hex}$	switching point 4	–
64	40_{hex}	switching point 5	–
65	41_{hex}	switching point 6	–
66	42_{hex}	switching point 7	–
67	43_{hex}	switching point 8	–
103	67_{hex}	trace trigger output	–
136	88_{hex}	reference existing	–
160	$A0_{hex}$	trigger status position measurement	corresponds to bit 0 in the *trigger status*
161	$A0_{hex}$	trigger status fast position measurement	corresponds to bit 2 in the *trigger status*
162	$A0_{hex}$	trigger status trace recording	corresponds to bit 4 in the *trigger status*

Selection code		Signal designation	Remarks
167	$A7_{hex}$	synchronous point (flying shear)	in axis operating mode ***flying shear***

The signals of the drive-specific functions can also be output at the internal O 15.x outputs. These outputs are "interconnected" with the internal I 15.x inputs. This way, another function linked to input I 15.x can be called up using output O 15.x.

Signal selection digital inputs 15.x	Index: `5fb3`, Short name: `DEin15Auswahl`
SPP Windows	`Signal selection digital input 15.0  for free use`
Type	array, 8 type integer16 elements
Array element	configuration digital inputs I 15.0 .. I 15.7
Var. type	device machine data read and write
Unit	select
Standard value	for free use
Valid	always; V 7.5 or higher, not for MaxiDrive

Signal selection digital outputs 15.x	Index: `5fb2`, Short name: `DAus15Auswahl`
SPP Windows	`Signal selection digital output 15.0  for free use`
Type	array, 8 type integer16 elements
Array element	configuration digital inputs I 15.0 .. I 15.7
Var. type	device machine data read and write
Unit	select
Standard value	for free use
Valid	always; V 7.5 or higher, not for MaxiDrive

The digital inputs and outputs I 15.x and O 15.x are linked outputs which are connected internally. Their functions correspond to the functions of the *signal selection digital inputs ...* and *signal selection digital outputs ...* (see above).

Analog output 1 signal selection	Index: `5f82`, Short name: `Ist1Auswahl`
SPP Windows	`Analog output 1 signal selection  Actual speed  select`
Type	simple variable, integer16
Var. type	device machine data read and write
Unit	select
Standard value	actual speed
Valid	always; V 3.5 or higher; for devices with -A1 option

Analog output 2 signal selection	Index: `5f83`, Short name: `Ist2Auswahl`
SPP Windows	`Analog output 2 signal selection` <u>`Actual current`</u> `select`
Type	simple variable, integer16
Var. type	device machine data read and write
Unit	select
Standard value	actual current
Valid	always; V 3.5 or higher; for devices with -A1 option

With variables *analog output 1 signal selection* and *analog output 2 signal selection* certain signals can be output at analog output 1 or 2 of the servo drive. The value entered in the respective variable determines which signal is output. The scaling can be changed via the *analog output ... FP-factor* variable (see below).

The following tables list all the signals which can be mapped to the analog outputs and specify the associated selection codes.

Signals generated by the digital signal processor (DSP) in the speed and current controller cycle (62.5 µs):

Selection code		**Signal designation**	**Type**	**Standardization and units**
20	14_{hex}	Actual current	i16	1.5 V = 9676 corresponds to max. continuous current of the servo drive, exception: for devices TrioDrive D/xS 1.0 V = 6451 corresponds to max. continuous current of the servo drive
205	CD_{hex}	Current setpoint		
38	26_{hex}	Current setpoint (DSP internal quantity) – in axis operating mode ***torque mode***, only		
39	27_{hex}	Actual speed (actual velocity)	i16	-5 V .. +5 V = –32768 .. +32767 corresponds to -16384 .. +16383 r.p.m. (V 8.5.9 or higher) or -8192 .. +8192 r.p.m. (up to V 8.5.9)
37	25_{hex}	Speed setpoint (DSP internal quantity) – not available in axis operating mode ***velocity mode direct***		
24	18_{hex}	Rotor position (of motor position sensor) (resolver)	i16	-5 V .. +5 V = –32768 .. +32767 position sensor steps (= 1 revolution)
21	15_{hex}	Analog input “Setpoint” or “Setpoint 1”	i16	–5 V .. +5 V = –32768 .. +32767 corresponds to -10 V .. +10 V at the analog input setpoint
81	51_{hex}	Analog input “Setpoint 2”		

Signals generated by the microcontroller in the position control loop cycle (1 ms). A processing time of approx. 5 ms for variables must be taken into account.

Selection code		**Signal designation**	**Type**	**Standardization and units**
253	FD_{hex}	Current setpoint filtered	i16	1.5 V = 9676 corresponds to

Selection code		Signal designation	Type	Standardization and units
35	23_{hex}	Current output value torque function		max. continuous current of the servo drive
30	$1e_{hex}$	*Torque setpoint* variable (torque setpoint digital)	i16	0.15 V = 1000 corresponds to the motor rated torque
0	00_{hex}	Actual speed of DSP	i16	-5 V .. +5 V = –32768 .. +32767 corresponds to -16384 .. +16383 r.p.m. (V 8.5.9 or higher) or -8192 .. +8192 r.p.m. (up to V 8.5.9)
34	22_{hex}	Actual speed input encoder signals (option Lx), derived from differentiation of position information (related to the master axis)		
43	$2b_{hex}$	Feedforward speed ***electronic gearing*** (related to slave axis)		
31	$1f_{hex}$	*Target velocity* variable (target velocity digital)		
4	04_{hex}	Speed setpoint for DSP (position control loop output)		
2	02_{hex}	Actual position (related to application zero point)	i32	-5 V .. +5 V = -32768 .. +32767 position sensor steps
27	$1b_{hex}$	Position sensor actual position resolver (option R1)		
28	$1c_{hex}$	Position sensor actual position input encoder signals (option Lx)		
6	06_{hex}	*Internal target position* variable (related to application zero point) (position setpoint)		
10	$0a_{hex}$	Control offset of position control loop – not in the axis operating modes ***electronic gearing*** and ***flying shear***		
12	0c hex	*Analog outputs* variable, subindex 1 (analog output value 1)	i16	-5 V .. +5 V = -16384 .. +16383
13	0d hex	*Analog outputs* variable, subindex 2 (analog output value 2)		
14	$0e_{hex}$	*Analog outputs* variable, subindex 3 (analog output value 3		
15	$0f_{hex}$	*Analog outputs* variable, subindex 4 (analog output value 4)		
91	$5B_{hex}$	Encoder voltage	i16	1:1
101	65_{hex}	DC-bus voltage	i16	-5 V .. +5 V correspond to a DC-bus voltage of -32786 .. +32767 volts

If the product of the selected signal and the scaling factor *analog output ... FP factor* (firmware V 5.4 or higher) or *analog output ... Int factor* exceeds the value range of –32768 .. +32767 (integer16), the limit values –32768 or +32767 are output instead.

These types of limit and overflow effects must be taken into account during selection of the factors and interpretation of the output voltages.

Please also note the information on output voltage range, resolution, and bandwidth of the analog outputs in operating instructions “Connection and Commissioning” (6755.202, 6745.202, 6750.202, 6740.202, or 6710.202).

Analog output 1 FP factor	Index: `5fa1`, Short name: `FakAnaIst1`
SPP Windows	`Analog output 1 FP factor` <u>`1.00000`</u>
Type	simple variable, integer32
Var. type	setpoint or parameter for a function — read and write
Unit	65536 (10000_{hex}) corresponds to FP factor = 1.0
Standard value	65536 (corresponds to FP factor = 1.0)
Valid	always; V 5.4 or higher; for devices with -A1 option

Analog output 2 FP factor	Index: `5fa2`, Short name: `FakAnaIst2`
SPP Windows	`Analog output 2 FP factor` <u>`1.00000`</u>
Type	simple variable, integer32
Var. type	setpoint or parameter for a function — read and write
Unit	65536 (10000_{hex}) corresponds to FP factor = 1.0
Standard value	65536 (corresponds to FP factor = 1.0)
Valid	always; V 5.4 or higher; for devices with -A1 option

Analog output 1 FP factor and *analog output 2 FP factor* serve for scaling the voltage output at analog output “Ist1” or “Ist2” . A signal mapped to the output (see *analog output ... signal selection* variable above) is multiplied by this factor before output. In the case of a negative factor, the sign of the signal is inverted.

The output has a value range of –5 V .. +5 V, which corresponds to preset values of –32768 .. +32767 (integer16). This means that the following examples apply:

Value of the signal mapped to “Ist1”	×	**Analog output 1 FP factor**	=	**Signal × factor**	=	**Output “Ist1”**
32767	×	1.0	=	32767	=	+5.0 V
32767	×	0.5	=	16383	=	+2.5 V
3276	×	–10.0	=	–32760	=	–5.0 V

No values higher than +32767 or lower than –32768 are output for the product of signal and factor. If the product exceeds the above value range, the limit values +32767 or –32768 are output instead. These types of limit effects must be taken into account during selection of the factors and interpretation of the output voltages.

Please, also note the information on output voltage range, resolution, and bandwidth of the analog outputs in section “Analog outputs Ist1, Ist2” of operating instructions “Connection and Commissioning” (6755.202, 6745.202, 6750.202, 6740.202, or 6710.202).

Analog output 1 int factor	Index: `5f6b`, Short name: `DSPFakDac1`
SPP Windows	`Analog output 1 Int factor 1`
Type	simple variable, integer16
Var. type	setpoint or parameter for a function read and write
Unit	–
Standard value	1
Valid	always; V 3.5 or higher (cannot be used from V 5.4 on); for devices with -A1 option

Analog output 2 int factor	Index: `5f6c`, Short name: `DSPFakDac2`
SPP Windows	`Analog output 2 Int factor 1`
Type	simple variable, integer16
Var. type	setpoint or parameter for a function read and write
Unit	–
Standard value	1
Valid	always; V 3.5 or higher (cannot be used from V 5.4 on); for devices with -A1 option

Analog output 1 Int factor and *analog output 2 Int factor* serve for scaling the voltage output at analog output “Ist1” or “Ist2” . A signal mapped to the output (see *analog output ... signal selection* variable above) is multiplied by this factor before output. In the case of a negative factor, the sign of the signal is inverted.

The output has a value range of –5 V .. +5 V, which corresponds to preset values of –32768 .. +32767 (integer16). This means that the following examples apply:

Value of the signal mapped to “Ist1”	×	**Analog output 1 FP factor**	=	**Signal × factor**	=	**Output “Ist1”**
32760	×	1	=	32760	=	+5 V
3276	×	10	=	32760	=	+5 V
3276	×	–10	=	–32760	=	–5 V

Please, also note the information on output voltage range, resolution, and bandwidth of the analog outputs in section “Analog outputs Ist1, Ist2” of operating instructions “Connection and Commissioning” (6755.202, 6745.202, 6750.202, 6740.202, or 6710.202).

Limit frequency current filter	Index: 5e94, Short name: FgIsqFilter
SPP Windows	Limit frequency current filter 100.00 Hz
Type	simple variable, float32
Var. type	device machine data read and write
Unit	Hz
Standard value	100 Hz
Valid	always; V 8.5.999.59 or higher

This parameter is used for setting the limit frequency signal "current setpoint filtered" is filtered with (value range approx. 0.01 Hz .. 130 Hz).

9.4 Variable Descriptions Axis Machine Data

9.4.1 Axis Control Machine Data

The axis control machine data include the following variables:

- *Enable selection code*
- *Release brake selection code*
- *Brake delay times*
- *Axis operating mode at startup*
- *Stop axis selection code*

In command and commissioning software SPP Windows, the axis control machine data can be found in the "Parameterization" window under *Axis data/Axis control*.

Enable selection code	Index: 5f15, Short name: FreiAuswahl
SPP Windows	Action enable inactive/active no action/no action select
Type	simple variable, integer16
Var. type	axis machine data read and write
Unit	select
Standard value	no action / no action
Valid	in all axis operating modes

With the *enable selection code*, the effect of the "enable" input can be specified.

Permitted values:

<table>
<tr><th>Value</th><th>Effect if input “enable” = inactive</th><th>Effect if input “enable” = active</th><th>Comment</th></tr>
<tr><td>0</td><td>Ineffective (no action)</td><td>Ineffective (no action)</td><td>–</td></tr>
<tr><td>1</td><td>Axis fault</td><td>Ineffective (no action)</td><td>Axis error code “External fault”</td></tr>
<tr><td>2</td><td>Axis control command “disable voltage”</td><td rowspan="3">Switch axis state machine to “operation enabled”</td><td rowspan="3">Control commands in the axis control word have no effect</td></tr>
<tr><td>3</td><td>Axis control command “quick stop”</td></tr>
<tr><td>4</td><td>Axis control command “stop”</td></tr>
<tr><td>5</td><td>Switches state machine part program control to “idle” state.</td><td>Device control command “start part program” (from label 9999 on)</td><td rowspan="2">On removal of enable, the running motion is interrupted with the axis control command “quick stop”. “Resume part program” is not possible with selection code 6.</td></tr>
<tr><td>6</td><td>Device control command “stop part program”</td><td>Device control command “resume part program” and “start part program” (from label 9999 on)</td></tr>
</table>

Release brake selection code	Index: `5f16`, Short name: `BremseLoesenAuswahl`
SPP Windows	`Brake released at device status` <u>`always released`</u> `select`
Type	simple variable, integer16
Var. type	axis machine data — read and write
Unit	select
Standard value	always released
Valid	in all axis operating modes

The *brake release selection code* determines in which states of the axis state machine the brake is released, i. e. the “brake” output is activated.

Permitted values:

Value	Brake released:
0	Always released
3	In “switched on” state and higher
4	In “operation enabled” state
5	In all states except for “fault”

These are divided into two categories:

- Release brake selection code = 0 (always released)

  The brake is deactivated, i. e. output “brake” is always set, regardless of the device status and the level of bus voltage.

- Release brake selection code ≠ 0

The brake is only released (output “brake” is set), if

– the device status corresponds to the condition in the *release brake selection code* **and**
– the DC-bus voltage is sufficient **and**
– there is no fault.

Brake delay times	Index: `5f18`, Short name: `BremseVerzZeit`
SPP Windows	`Brake activation time 0 ms` `Brake release time 0 ms`
Type	array, 2 type unsigned16 elements, 16, subindex 1 and 2
Element 1	brake release time
Element 2	brake activation time
Var. type	axis machine data read and write
Unit	ms
Standard value	0
Valid	always

This variable can be used for compensating the physical response time (activation of brake) and the release time (release brake) of emergency braking systems.

Via subindex 1 (brake release time), the time between activation of the control and release of the brake is defined, via subindex 2 (brake activation time), the time between activation of the brake and switch-off of the PWM is defined.

Please note: In case of a mains breakdown, the brake is activated immediately. In case of a fault, the brake is activated as soon as state fault reaction active is reached, i. e. brake ramps are not waited for.

Axis operating mode at startup	Index: `5f35`, Short name: `AchsBAVorwahl`
SPP Windows	`Axis operating mode at startup Velocity mode select`
Type	simple variable, integer8 (V 8.5.996 or higher) simple variable, integer16 (up to V 8.5.995)
Var. type	axis machine data read and write
Unit	select
Standard value	velocity mode
Valid	always; V 5.1 or higher

Axis operating mode at startup defines the axis operating mode active after the drive has been switched on.

Permitted values:

Value	Axis operating mode
4	Torque mode
–3	Velocity mode direct (V 3.2 or higher)
3	Velocity mode
1	Profile position mode
6	Homing mode
–1	Electronic gearing
–2	Flying shear (V 4.5 or higher)
–4	Velocity profile (V 5.95 or higher)
–5	Timed positioning mode (V 8.5.9 or higher)
10	Cyclic synchronous torque mode (V 8.5.999.53 or higher)
9	Cyclic synchronous velocity mode (V 8.5.999.54 or higher)
8	Cyclic synchronous position mode (V 8.5.999.62 or higher)
7	Interpolated Position Mode (V 5.95 or higher)

Stop axis selection code	Index: `5e97` Short name: `StopAuswahl`
SPP Windows	`Stop axis selection code` <u>`no reaction`</u> `select`
Type	simple variable, integer16
Var. type	axis machine data read and write
Unit	select
Standard value	no reaction
Valid	in all axis operating modes

The *stop axis selection code* determines the effect of input function “axis stop”, i. e. the way the axis is stopped as soon as the input is set.

Permitted values:

Value	Stop axis
0	no reaction
1	stop without ramp
2	stop with brake ramp
3	quick stop

9.4.2 Setpoint Sources Machine Data

The setpoint sources machine data include the following variables:

- *Target velocity source*
- *Target velocity analog factor*
- *Target velocity analog offset*

- *Torque setpoint source*
- *Torque analog factor*
- *Torque analog offset*
- *Max current amount source*
- *Max current amount analog factor*
- *Max current amount analog offset*

In command and commissioning software SPP Windows, the setpoint sources machine data can be found in the “Parameterization” window under *Axis Data/ Setpoint sources*.

Target velocity source	Index: `5f11`, Short name: `VelVerfahrAuswahl`
SPP Windows	`Target velocity source` `Target velocity digital` `select`
Type	simple variable, integer16
Var. type	axis machine data read and write
Unit	select
Standard value	–
Valid	in axis operating modes ***profile position mode***, ***velocity mode***, and ***flying shear***; only for devices with -A1 option

Variable *target velocity source* is used to determine whether the target velocity is set digitally via the corresponding *target velocity* variable or analog via analog input “setpoint”. Valid values are:

Selection code		**Signal Designation**	**Type**	**Standardization**
21	15_{hex}	Analog input “setpoint” or “setpoint 1” (12 bit input)	I/O	scaleable, offset adjustable
31	$1f_{hex}$	Variable *target velocity* (target velocity digital)	Var	velocity units (VU)

If the setpoint analog input is selected as source, the signal can be scaled via the *target velocity analog factor* variable and zero offset via the *target velocity analog offset* variable. The value read out at the input is continually transferred to the *target velocity* variable in the specified axis operating modes where it can also be read out.

Target velocity analog factor	Index: `5f1b`, Short name: `AnaVelFaktor`
SPP Windows	`Target velocity analog factor` `100.0` `%`
Type	simple variable, integer16
Var. type	axis machine data — read and write
Unit	0.1% of the motor rated speed at 10 V setpoint
Standard value	100%
Valid	in axis operating modes ***profile position mode***, ***velocity mode***, ***velocity mode direct***, and ***flying shear***; V 3.2 or higher; only for devices with -A1 option

The *target velocity analog factor* serves for scaling analog input “setpoint” if this is selected as target velocity source. You can specify how many per thousand of the motor rated speed of the motor corresponds to a setpoint of 10 V.

Examples (for motor rated speed = 3,000 r.p.m.):

- *Target velocity analog factor* = 1000 (100%): 10 V at the analog input correspond to 3,000 r.p.m. setpoint
- *Target velocity analog factor* = 200 (20%): 10 V at the analog input correspond to 600 r.p.m. setpoint

Target velocity analog offset	Index: `5f1e`, Short name: `AnaVelOffset`
SPP Windows	`Target velocity analog offset` `0.0` `r.p.m.`
Type	simple variable, integer16
Var. type	axis machine data — read and write
Unit	0.5 r.p.m. (V 8.5.9 or higher) 0.25 r.p.m. (up to V 8.5.9)
Standard value	0
Valid	axis operating modes ***profile position mode***, ***velocity mode***, ***velocity mode direct***, and ***flying shear***; V 3.2 or higher; only for devices with -A1 option

The *target velocity analog offset* permits a zero offset for analog input “setpoint” if this input is selected as target velocity source. This compensates zero offsets of the analog signals.

When servo drives are exchanged, an adjustment of this value may be required as the internal compensation can vary with the individual devices.

Torque setpoint source	Index: `5f10`, Short name: `MsollAuswahl`
SPP Windows	`Torque setpoint source` `Torque setpoint digital` `select`
Type	simple variable, integer16
Var. type	axis machine data — read and write
Unit	select
Standard value	–
Valid	in axis operating mode ***torque mode***; for devices with -A1 option, only

Variable *torque setpoint source* is used for specifying whether the torque setpoint is set digitally via the corresponding *torque setpoint* or analog via the “setpoint” analog input. Valid values are:

Selection code		Signal designation	Type	Standardization
21	15 hex	Analog input “setpoint” or “setpoint 1” (12 bit input)	I/O	scaleable, offset adjustable
30	1e hex	Variable *torque setpoint* (torque setpoint digital)	Var	per thousand of motor rated current

If the “setpoint” analog input is selected as source, a scaling of the signal can be carried out via variable *torque analog factor*, a zero offset via variable *torque analog offset*.

Torque analog factor	Index: `5f1a`, Short name: `AnaMFaktor`	
SPP Windows	`Torque analog factor 100.0 %`	
Type	simple variable, integer16	
Var. type	axis machine data	read and write
Unit	0.1% of motor rated torque at 10 V setpoint	
Standard value	100%	
Valid	in axis operating mode ***torque mode***; V 3.2 or higher; only for devices with -A1 option	

The *torque analog factor* serves for scaling analog input “setpoint” if this input is selected as the torque setpoint source. You can specify how many per thousand of the motor rated torque correspond to a setpoint of 10 V.

Examples (for motor rated torque = 3.4 Nm):

- *Torque analog factor* = 1000 (100%): 10 V at the analog input correspond to 3.4 Nm setpoint
- *Torque analog factor* = 200 (20%): 10 V at the analog input correspond to 0.68 Nm setpoint

Torque analog offset	Index: `5f1d`, Short name: `AnaMOffset`	
SPP Windows	`Torque analog offset 0.0 %`	
Type	simple variable, integer16	
Var. type	axis machine data	read and write
Unit	0.1% of motor rated torque	
Standard value	0	
Valid	in axis operating mode ***torque mode***; V 3.2 or higher; only for devices with A1 option	

The *torque analog offset* permits a zero offset for analog input “setpoint” if this input is selected as torque setpoint source. This compensates for zero offsets of the analog signals.

When servo drives are exchanged, an adjustment of this value may be required as the compensation can vary with the individual devices.

Max current amount source	Index: 5f28, Short name: IImpulsAuswahl
SPP Windows	`Max current amount source` Variable "max current amount" select
Type	simple variable, integer16
Var. type	axis machine data — read and write
Unit	select
Standard value	–
Valid	in all axis operating modes; V 3.5 or higher; only for devices with -A1 option

Variable *max current amount source* is used for specifying whether the current limit is set digitally via the corresponding *current max amount* variable or analog via analog input “setpoint”. Valid values are:

Selection code		Signal designation	Type	Standardization
21	15_{hex}	Analog input “setpoint” or “setpoint 1” (12 bit input)	I/O	scaleable, offset adjustable
81	51_{hex}	Analog input “setpoint 2” (12 bit input)	I/O	scaleable, offset adjustable
44	$2c_{hex}$	Variable *current max amount*	Var	per thousand of motor rated current

If the “setpoint” analog input is selected as source, the signal can be scaled via the *max current amount analog factor* variable and zero offset via the *max current amount analog offset* variable.

Max current amount analog factor	Index: 5f26, Short name: AnaIImpFaktor
SPP Windows	`Max current amount analog factor` 200.0 %
Type	simple variable, integer16
Var. type	axis machine data — read and write
Unit	0.1% of motor rated current at 10 V
Standard value	100%
Valid	in all axis operating modes; V 3.5 or higher; only for devices with -A1 option

The *max current amount analog factor* serves for scaling analog input “setpoint” if this input is selected as max current amount source. You can specify to how many per thousand of the motor rated current the current is limited at a setpoint of 10 V.

Examples:

- *Max current amount analog factor* = 1000 (100%): 10 V at the analog input correspond to a current limit of 100% of the rated current
- *Max current amount analog factor* = 200 (20%): 10 V at the analog input correspond to a current limit of 20% of the rated current

Max current amount analog offset	Index: `5f27`, Short name: `AnaIImpOffset`	
SPP Windows	`Max current amount analog offset 0.0 %`	
Type	simple variable, integer16	
Var. type	axis machine data	read and write
Unit	0.1% of motor rated current	
Standard value	0	
Valid	in all axis operating modes; V 3.5 or higher; only for devices with -A1 option	

The *max current amount analog offset* permits a zero offset for analog input “setpoint” if this input is selected as max current amount source. This compensates for zero offsets of the analog signals.

When servo drives are exchanged, an adjustment of this value may be required as the compensation can vary with the individual devices.

9.4.3 Position Measuring System Machine Data

The position measuring system machine data include the following variables:

- *Position sensor source*
- *Position sensor resolution*
- *Gear ratio motor shaft revolutions*
- *Gear ratio drive shaft revolutions*
- *Ext. sensor encoder data*
- *Ext. Sensor: Measuring steps per revolution*
- *Ext. Sensor: Sine periods per revolution*
- *Ext. Sensor: Number of coded revolutions*
- *Ext. Sensor: Encoder type code*
- *Ext. Sensor: Interface identification*
- *Ext. Sensor: Manufacturer identification*
- *Encoder absolute type*
- *Motorposition sin/cos monitoring*

In command and commissioning software SPP Windows, these variables can be found in the “Parameterization” window under *Axis data/Position measuring system*.

Position sensor source	Index: `5f12`, Short name: `PosIstLgAuswahl`
SPP Windows	`Position sensor source` `Actual encoder value motor axis` `select`
Type	simple variable, integer16
Var. type	axis machine data — read and write
Unit	select
Standard value	position sensor actual position motor axis
Valid	in all axis operating modes

Variable *position sensor source* is used for specifying the position sensor used for the *actual position*. Valid values are:

Selection code		**Signal designation**
27	$1b_{hex}$	Actual encoder value motor axis (motor position sensor)
28	$1c_{hex}$	Actual encoder value external encoder at input encoder signals (option Lx)

The *position sensor source* must not be mixed up with the *position sensor selection code* indicating the position sensor used for commutation. This is recognized automatically and set to the installed encoder module – resolver (option R1), Sincos (Hiperface) encoder (Option R2), high-resolution incremental encoder (option R3), or EnDat encoder (option R4).

Position sensor resolution	Index: `608f`, Short name: `PosAufloesung`
SPP Windows	`Position sensor resolution` `65536` `PSS/rev.`
Type	array, 2 type unsigned32 elements
Array element 1	position sensor steps (PSS) (with incremental encoder after internal quadruplication)
Array element 2	motor revolutions (always 1)
Var. type	axis machine data — read and write
Unit	position sensor steps (PSS) per revolution
Standard value	65536
Valid	in all axis operating modes

In the *position sensor resolution* you have to specify the number of steps one revolution of the motor is resolved in. With that, the internal resolution is determined by which positions are determined and calculated.

If you would like to set and read position values in physical position units (e. g. in mm, µm, or angular degrees), you can do so by specifying a conversion factor *position factor* (see section Factors and Units Machine Data 158). This conversion factor is used for converting the position sensor steps to your units.

From firmware V 8.5.9 on the *position sensor resolution* can be selected freely by the user. It is recommended to select the highest one possible.

- If a resolver is used as position sensor, the upper limit is 65.536.
- If a position sensor with sine-cosine signals is used, the upper limit results from the number of sine periods multiplied by 8,192; for 512 sine periods 4,194,304 and for 2048 sine periods 16,777,216.
- For position sensors with digital transmission of the actual position, the upper limit is the number of transmitted position values per revolution, e.g. 524,288 or 33,554,432.

For multiturn encoders, please make sure that the product of *position sensor resolution* and number of possible revolutions is lower than 536,870,912 (=2^{29}), e.g. for 4,096 possible revolutions, the position sensor resolution must be lower than 131,072.

For devices with firmware versions below V 8.5.9, the *position sensor resolution* cannot be selected freely. If the motor position sensor (resolver, Sincos (Hiperface) encoder, high-resolution incremental encoder, or EnDat encoder) is used as position sensor, 65,536 must be preset, here. This means that each revolution is resolved into 65,536 steps even if the motor position sensor offers a higher resolution.

You can also use an external position sensor (incremental encoder) as position sensor instead of the motor position sensor. This external position sensor is connected to input encoder signals (option L1 or ZL1). In this case, the number of steps of the position sensor (incremental encoder) must be converted to the number of steps per motor revolution for the *position sensor resolution*. Furthermore, it is also necessary to take into account that the steps in the servo drive are quadrupled. Example :

- motor with gearbox i = 5,
- incremental encoder with 1000 steps per revolution at gear output

=>One motor revolution corresponds to 1/5 revolution at the gear output, i. e. 1000 / 5 = 200 steps of the incremental encoder. After quadruplication in the servo drive, this corresponds to 200 × 4 = 800 steps of a motor revolution, so that, for the purposes of this example, 800 “position sensor steps” need to be entered in this variable.

From firmware version V 8.5.999.56 on, the gear ratio can be defined in variable *gear ratio* and does not have to be considered for conversion.

Gear ratio	Index: `6091`, Subindex: `2`, Short name: `GearRatio`	
SPP Windows	`Gear ratio motor shaft rev.` <u>1</u> `Gear ratio driving shaft rev.` <u>1</u>	
Type	array, 2 type unsigned32 elements	
Array element 1 **Array element 2**	number revolutions motor shaft number revolutions drive shaft	
Var. type	axis machine data	read and write
Unit	–	
Standard value	1:1	
Valid	always; V 8.5.999.56 and higher	

For gearboxes with external position sensor at the drive side, the gear ratio can be entered here so that it is considered automatically for the *position sensor resolution*.

Ext. sensor encoder data	`Index: 5E86`, Short name: `Encoder2Data`	
Type	record, 5 elements	
Var. type	axis machine data	read and write
Valid	always; V 8.6 or higher	

The machine data of the optional external position sensor are grouped in record variable *Ext. sensor encoder data*. The structure is based on the machine data of the motor position sensor. The individual elements are described in the following.

Ext. sensor: measuring steps per revol. / meter	`Index: 5E86`, Subindex: `1`, Short name: `Encoder2Data1`	
SPP Windows	`Ext. sensor: measuring steps per revol.` <u>0</u>	
Type	record element, unsigned32	
Var. type	axis machine data	read and write / tool
Unit	1 increment, for linear length scales per meter	
Standard value	–	

For incremental encoders with square-wave signals (e. g. option -L1 or -ZL1), the number of measuring steps resulting from internal quadruplication is entered here. Example: An encoder with 250 pulses per revolution (see below) supplies 1.000 measuring steps per revolution.

For absolute encoders (e. g. EnDat encoder for option -ZL4), the number of measuring steps (position values) per revolution is displayed here. Examples: 262,144 (18 bit), 8,388,608 (23 bit), or 33,554.432 (25 bit).

Ext. sensor: sine periods per revolution / meter Index: 5E86, Subindex: 2, Short name: Encoder2Data2	
SPP Windows	Ext. sensor: sine periods per revolution 1
Type	record element, unsigned32
Var. type	axis machine data read and write / tool
Unit	–
Standard value	–

For position sensors with square-wave signals (e. g. for option -L1 or -ZL1), the number of pulses per revolution is entered here. If an EnDat encoder is used for option -ZL4, the number of sine periods per revolution is displayed here. Example: 16, 1024, 250.

Ext. sensor: number of coded revolutions Index: 5E86, Subindex: 3, Short name: Encoder2Data3	
SPP Windows	Ext. sensor: number of coded revolutions 1
Type	record element, unsigned16
Var. type	axis machine data read and write / tool
Unit	–
Standard value	–

For singleturn encoders, 1 is entered here, for multiturn encoders, the number of revolutions that can be acquired. For EnDat encoders, the entry is carried out automatically. Whether or not the servo drive can evaluate several revolutions depends on the interface.

Ext. sensor: encoder type code Index: 5E86, Subindex: 4, Short name: Encoder2Data4	
SPP Windows	Ext. sensor: encoder type code 0000
Type	record element, unsigned16
Var. type	axis machine data read and write / tool
Unit	–
Standard value	–

Ext. sensor: encoder type code uses the same assignment as *Encoder type code* of the motor position sensor. For EnDat encoders, this entry is carried out automatically.

Currently reasonable values (others on request):

Value	Encoder type
0	enocder type not defined
External encoders for option -L1 or -ZL1:	
0c0f	rotatory square-wave incremental encoders

Value	Encoder type
1c0f	square-wave incremental encoders for linear axes
External encoders for option -ZL4 / ZL7:	
030f	rotatory EnDat encoders
130f	EnDat or BiSS encoders for linear axes (absolute)

Ext. sensor: interface identification Index: 5E86, Subindex: 5, Short name: Encoder2Data5	
SPP Windows	`Ext. sensor: interface identification 0`
Type	record element, unsigned16
Var. type	axis machine data read and write / tool
Unit	–
Standard value	RSS incremental with zero pulse

Ext. sensor: interface identification is set by the manufacturer and uses the same assignment as *interface identification* of the motor position sensor.

Currently reasonable values (others on request):

Value	Interface
2	RS 422/RS 488 digital (e.g. EnDat or BiSS)
5	RS 422 = incremental with zero pulse (encoder with square-wave signals)

Manufacturer ID Index: 5E86, Subindex: 6, Short name: Encoder2Data6	
SPP Windows	`Ext. sensor: manufacturer ID undefined select`
Type	record element, unsigned16
Var. type	axis machine data read and write
Unit	select
Standard value	undefined
Valid	V 8.5.999.63 and higher

In case a digital absolute encoder is used, the type is defined, here.

Permissible values:

Value	Type
0	undefined
32768	EnDat
32771	BiSS-C one-directional

Encoder absolute type	Index: `5e9d`, Short name: `PosAbsTyp`
SPP Windows	`Encoder absolute type` <u>`Auto`</u> `select`
Type	simple variable, integer16
Var. type	axis machine data read and write
Unit	select
Standard value	auto
Valid	in all axis operating modes

If the position sensor actual position of the motor axis is selected as *position sensor source*, this variable may be used for defining a multi-turn encoder as single-turn for going to home position with a circular axis.

Permitted values:

Value	**Position sensor**
0	Auto
1	Single-turn

9.4.4 Drive System Machine Data

The drive system machine data include the following variables:

- *Polarities*
- *Maximum motor speed*
- *Continuous current motor*
- *Max current amount*
- *Max current amount 2*
- *Inertia load / Mass*
- *Commutation reference*

In command and commissioning software SPP Windows, the drive system machine data can be found in the “Parameterization” window under *Axis data/Drive*.

Polarities	Index: `607e`, Short name: `Polaritaeten`
SPP Windows	`Polarities` <u>`negative (ext. encoder)`</u> `select`
Type	simple variable, octet string length 1
Var. type	axis machine data read and write
Unit	select
Standard value	“position negative” (bit 7 = 1)
Valid	in all axis operating modes

The setpoint and actual values are multiplied by 1 or -1, depending on their polarity. This enables users to reverse the direction of orientation.

Bit 7: position polarity

- 0 = positive
- 1 = negative (standard value)

Bit 6: reserved

- intended for future expansions, currently without effect

The position polarity must not and does not need to be changed if the motor position sensor (resolver, Sincos (Hiperface) encoder, high-resolution incremental encoder, or EnDat encoder) is implemented as position measuring system.

If an external position sensor is implemented at input encoder signals (option Lx or ZLx), the counting direction of the encoder signals can be changed via the position polarity in order to close the position control loop, if necessary.

If the polarities are set incorrectly, the control loops do not close and the axis may “run away”, i. e. run at maximum speed.

Maximum motor speed	Index: `6080`, Short name: `Nmax`
SPP Windows	`Maximum motor speed 6000.0 r.p.m.` (for rotatory drive systems) `Maximum speed 5.00 m/s` (for linear motors)
Type	simple variable, unsigned32
Var. type	axis machine data — read and write
Unit	0.5 r.p.m. or 1/4096 m/s (V 8.5.9 or higher) 0.25 r.p.m. (up to V 8.5.9)
Standard value	–
Valid	in all axis operating modes except for ***torque mode / force mode***

The maximum speed is set for both directions of rotation together. It serves as a protection for the motor and can be taken from the motor data sheet.

In axis operating mode **torque mode** and **force mode**, a speed limitation is not carried out at the moment.

Monitoring of the speed via software, as is the case with this variable, does not comply with the safety requirements of the machinery directive.

- If the speed is to be monitored at the machine, compliance with the safety requirements of the machinery directive is essential.

Continuous current motor	Index: `5f02`, Short name: `IDauer`
SPP Windows	`Continuous current motor 100.0 %I rated`
Type	simple variable, unsigned16
Var. type	axis machine data — read and write
Unit	0.1% of motor rated current
Standard value	100%
Valid	in all axis operating modes

The *continuous current motor* can be set here in per thousand of the *motor rated current*. Generally, 100% is set here, lower values can be used as a torque limit.

A limitation of the torque via software, as is the case with *continuous current motor*, does not comply with the safety requirements of the machinery directive.

- If you want to limit the torque at the machine, compliance with the safety requirements of the machinery directive is essential.

Max current amount	Index: `6073`, Short name: `IImpuls`
SPP Windows	`Max current amount 200.0 %I rated`
Type	simple variable, unsigned16
Var. type	axis machine data read and write
Unit	0.1% of motor rated current
Standard value	200%
Valid	in all axis operating modes

The peak current of the servo drive can be set per thousand of the *motor rated current* using *max current amount*. Generally, 200% is set here, lower values can be used as a torque limit (e. g. 100%, if no increase in peak current beyond the *motor rated current* is to occur).

Alternatively, the torque limit can be implemented via analog input “Setpoint”, see section Setpoint Sources Machine Data [144].

A limitation of the torque via software, as is the case with *max current amount*, does not comply with the safety requirements of the machinery directive.

- If you want to limit the torque at the machine, compliance with the safety requirements of the machinery directive is essential.

Max current amount 2	Index: `5e9c`, Short name: `IImpuls2`
SPP Windows	`Max current amount 2 200.0 %I rated`
Type	simple variable, unsigned16
Var. type	axis machine data read and write
Unit	0.1% of motor rated current
Standard value	200%
Valid	in all axis operating modes

If the *max current amount selection* function is used at a digital input, this input can be used for switching between variables *max current amount* (input not active) and *max current amount 2* (input active).

Inertia load or **Mass**	`Index: 5E82`, Short name: `InertiaLoad`
SPP Windows	`Inertia load 0.0 kgm²` (for rotatory drive systems) `Mass 0.0 kg` (for linear motors)
Type	simple variable, float32
Var. type	axis machine data read and write
Unit	1 kgm^2 or 1 kg
Standard value	0
Valid	in all axis operating modes

Enter the moment of inertia or the mass of the carriage incl. load and primary part, here. (If the moment of inertia is stated in 10^{-3} kg m^2, divide the value by 1,000 and enter the result here. If the moment of inertia is stated in 10^{-4} kg m^2 or in kg cm^2, divide the value by 10,000 and enter the result here.)

Commutation reference	`Index: 5E83`, Short name: `KommRefAuswahl`
SPP Windows	`Commutation reference Absolute information`
Type	simple variable, integer16
Var. type	axis machine data read and write
Unit	select
Standard value	absolute information
Valid	in all axis operating modes

The method for commutation finding is selected using this variable. For detailed information, see section Autocommutation [15].

Permissible values:

Value	Commutation finding
0	Absolute information (method determined by the applied encoder system)
–1	Autocommutation
–2	Zero point search using rotating field generator
–3	Autocommutation with each switch-on
–4	Zero point search using rotating field generator with each switch-on
–5	Always evaluate absolute information
–6	Mounting offset at current position

9.4.5 Factors and Units Machine Data

The factors and units machine data include the following variables:

- *Position factor*
- *Home offset*
- *Velocity factor*

In command and commissioning software SPP Windows, the factors and units machine data can be found in the “Parameterization” window under *Axis data/ Factors and units*.

Position factor	Index: 6093, Short name: PosFaktorSoll
SPP Windows	Position factor numerator 1 PSS Position factor denominator 1 PU
Type	array, 2 type unsigned32 elements
Array element 1	numerator in position sensor steps (PSS)
Array element 2	denominator in position units (PU)
Var. type	axis machine data read and write
Unit	(see above)
Standard value	1 PSS / 1 PU
Valid	in all axis operating modes

Variable *position factor* is used for specifying the meaning of the position units (PU). This permits an adaptation to physical units (e. g. mm or µm). The position units determine the representation of the variables *target position* and *actual position* and most of the other variables that represent position values.

Two categories can be distinguished:

- Position factor = 1 PSS / 1 PU (on delivery):

  One position unit corresponds to one position sensor step and thus depends exclusively on the set resolution of the position measuring system.

- Position factor ≠ 1 PSS / 1 PU (application-specific):

  One position unit corresponds to a specific number of position sensor steps (e. g. 10, 1/10, 27/33) which is set via this variable. This permits the display of position values in physical units (e. g. mm, µm).

Example 1: This example assumes a linear drive system where a motor drives a carriage via a spindle with the following values:

- Lead screw pitch = 5 mm (the carriage moves by 5 mm per motor revolution)
- Position sensor resolution = 65536 PSS (resolver)

In this case, a *position factor* of 65536/5000 means that

- 65536 increments (PSS), i. e. one motor revolution (= 5 mm), correspond to
- 5000 position units (PU)

Thus, 1 position unit (PU) corresponds exactly to 1 µm.

Example 2 (external position sensor): This example assumes a linear drive system where a motor drives a carriage via a spindle with the following values:

- Lead screw pitch = 5 mm (the carriage moves by 5 mm per motor revolution)
- Position sensor resolution = 4000 PSS (external position sensor with 1000 steps, connected to the motor; pulse quadruplication in the servo drive produces 4000 steps per motor revolution)

In this case, a *position factor* of 4/5 means that

- 4 increments (PSS) (= 1/1000 revolution = 5 µm) correspond to
- 5 position units

Thus, 1 position unit (PU) corresponds to exactly 1 µm.

The available value range is 1/268,435,456 ($1/2^{28}$) to 268,435,456/1 ($2^{28}/1$).

Any change to the *position factor* influences the settings of numerous other machine data described in the following sections. For this reason, the *position factor* must always be set prior to these other machine data.

Home offset	Index: `607c`, Short name: `PosRefOffset`
SPP Windows	`Home offset 0 PU`
Type	simple variable, integer32
Var. type	axis machine data — read and write
Unit	position units (PU)
Standard value	0
Valid	in all axis operating modes

The *home offset* specifies the distance between the machine home position and a zero point defined by the application (application zero point for short). This may be the machine zero point or the workpiece zero point. If machine home position and application zero point are located at the same point, 0 is to be entered in *home offset*.

Because the application zero point can be shifted by the *home offset*, this is also referred to as a zero offset.

For a detailed description with an example, see section Homing Mode [38]. Additionally, please not the effects of a zero offset on the positioning range, see section Positioning Range Machine Data [161].

With multi-turn encoders, homing is not required. Even with those encoders, the absolute position can be adjusted to the application zero using the *home offset*.

Velocity factor	Index: `6094`, Short name: `VelFaktorSoll`
SPP Windows	`Velocity factor numerator 1.0 r.p.m.` `Velocity factor denominator 1 VU`
Type	array, two type unsigned32 elements
Array element 1	numerator in 0.5 r.p.m. or 1/4096 m/s (V 8.5.9 or higher) numerator in 0.25 r.p.m. (up to V 8.5.9)
Array element 2	denominator in velocity units (VU)
Var. type	axis machine data — read and write
Unit	(see above)
Standard value	1 r.p.m. / 1 VU or 1/4096 m/s = 1 GE
Valid	in all axis operating modes

With variable *velocity factor*, the meaning of the velocity units (VU) is defined. This permits an adaptation to physical units (e. g. r.p.m. or mm/s). The velocity units determine the representation of the variables *target velocity*, *actual velocity,* and most other variables that represent velocities.

On delivery, the *velocity factor* of rotatory drive systems is set to 1 r.p.m. / 1 VU. Thus, a velocity unit always corresponds to 1 r.p.m.

Two categories can be distinguished:

- Target velocity factor = 1 r.p.m. / 1 VU

  One velocity unit corresponds to 1 r.p.m., i. e., corresponding parameters are specified directly in revolutions per minute.

- Target velocity factor ≠ 1 r.p.m. / 1 VU

  One velocity unit corresponds to a specific number of revolutions per minute (e. g. 4, 400, 48/37) which is set via the *velocity factor* variable. This permits the display of velocity values in physical units (e. g. mm/s, m/s).

Example:

- A factor of 1/10 means: 1 r.p.m. corresponds to 10 velocity units (VU). Thus, the corresponding velocity values are represented in 0.1 r.p.m.

Any change to the *velocity factor* influences the settings of numerous other machine data described in the following sections. For this reason the *velocity factor* must always be set prior to these other machine data.

9.4.6 Positioning Range Machine Data

The positioning range machine data include the following variables:

- *Axis type*
- *Range limit*
- *Position limit*

In command and commissioning software SPP Windows, the positioning range machine data can be found in the “Parameterization” window under *Axis data/ Positioning range*.

Axis type	Index: `5ef9`, Short name: `PosStreckentyp`
SPP Windows	`Axis type` <u>`Linear axis`</u> `select`
Type	simple variable, integer16
Var. type	axis machine data — read and write
Unit	select
Standard value	linear axis
Valid	in all axis operating modes

The *axis type* indicates to the profile position mode whether the axis is a linear or a circular axis. Also see *range limit* and *position limit min/max*.

Permitted values:

Value	Axis type
0	Linear axis
1	Circular axis with path optimization: Absolute preset *target positions* are approached via the shortest route (path optimiza-tion).
2	Circular axis without path optimization: Absolute preset *target positions* are approached in a positive direction if the *target position* is greater than the old target position (variable *internal target position*) and in a negative direction if it is smaller.

After changing an *axis type* or the *range limit*, you must first go to home position before you can work in one of the other axis operating modes.

Range limit	Index: `607b`, Short name: `PosBereichsGrenze`
SPP Windows	`Range limit min  0  PU` `Range limit max  65,535  PU`
Type	array, 2 type integer32 elements
Array element 1	range limit min in position units (PU)
Array element 2	range limit max in position units (PU)
Var. type	axis machine data read and write
Unit	position units (PU), range limit min = application zero point
Standard value	0 PU .. 65535 PU
Valid	in all axis operating modes (if axis type = circular axis)

If *axis type* circular axis is selected, this variable is used for setting the upper and lower limit of the value range. If, for example, a circle is to be resolved with 1/1000°, the following settings must be entered:

- *range limit min = 0*
- *range limit max = 359,999*

This example shows that, for the range limit on a circular axis, the values *range limit min (0)* and *range limit max* + 1 (360,000) coincide.

The following limits apply to profile position mode:

Positioning	Min. target position	Max. target position
absolute	*range limit min*	*range limit max*
relative	-1 revolution	+1 revolution

1 revolution is: *range limit max – range limit min* + 1.

Examples:

Range limit		Target position (absolute)		Target position (relative)	
min.	max.	min.	max.	min.	max.
0	359	0	359	-360	+360
1	65,536	1	65,536	-65,536	+65,536
-1,799	1,800	-1,799	1,800	-3,600	+3,600
-50	49	-50	49	-100	+100
0	21,599	0	21,599	-21,600	+21,600

The value range for the upper and lower limit is between -1,073,741,824 and +1,073,741,823, *range limit min* must always be smaller than *range limit max*.

Before specifying the range limits, the position units must be defined via the *position factor*.

In order to prevent rounding errors, *position factor* and *range limit* must be selected so that they can be converted to position sensor steps without remainders.

In case of a zero offset with the *home offset* and *range limit min* ≠ 0 it must be ensured that *range limit min* is mapped to the application zero point.

After changing an *axis type* or the *range limit*, you must first go to home position before you can work in one of the other axis operating modes.

Position limit	Index: `607d`, Short name: `PosGrenzwert`
SPP Windows	`Position limit min -536,870,912 PU` `Position limit max 536,870,911 PU`
Type	array, 2 type integer32 elements
Array element 1	position limit min in position units (PU)
Array element 2	position limit max in position units (PU)
Var. type	axis machine data — read and write
Unit	position units (PU) related to machine home position
Standard value	-536,870.912 PU .. 536,870,911 PU
Valid	in axis operating mode ***profile position mode*** (if axis type = linear axis)

If a linear axis is selected as *axis type*, *position limit min* and *position limit max* are the absolute position limits between which the target positions (in absolute form) must range. Each new target position is checked with regard to these limits.

The position limit values always relate to the machine home position. If the zero point of the coordinate system is shifted by a change of the home offset, the maximally permissible positions related to the application remain unchanged. Therefore, if applicable, the home offset must be taken into account when presetting absolute target positions.

Example: *Position limit min* = -1,000 PU

Position limit max = +1,000 PU

Home offset = 200 PU

=> permitted target positions = -800 .. 1,200 PU

Prior to defining the position limit values, the position units must be defined via the *position factor*.

The value range for the upper and lower limit is -536,870,912 and +536,870,911 position sensor steps (take into account conversion PU to PSS with *position factor*). However, these limits are virtually never reached as, at a resolution of 1 µm, they correspond to a range of more than 1,000 m. *Position limit min* must always be smaller than *position limit max*.

If, when switching from another axis operating mode to operating mode profile position mode, it cannot be ensured that the actual position is within the position limit values, you must first go to home position.

Monitoring of the limit positions via software, as is the case with *position limit*, does not comply with the safety requirements of the machinery directive.

- If the position is to be monitored at the machine, compliance with the safety requirements of the machinery directive is essential.

9.4.7 Ramps Machine Data

The ramps machine data include the following variables:

- *Motion profile type*
- *Acceleration time*
- *Deceleration time*
- *Quick stop time*
- *Ramps reference velocity*

In command and commissioning software SPP Windows, the ramps machine data can be found in the "Parameterization" window under *Axis data/Ramps*.

Motion profile type	Index: `6086`, Short name: `RampFormAuswahl`
SPP Windows	`Motion profile type` <u>`linear`</u> `select`
Type	simple variable, integer16
Var. type	axis machine data — read and write
Unit	select
Standard value	linear
Valid	in axis operating modes ***profile position mode***, ***velocity mode***, and ***homing mode***

The *motion profile type* specifies the behavior of the acceleration and deceleration process.

Permitted values are:

Value	Ramp type
–1	jump (no ramp)
0	linear ramp
1	sin^2 ramp

Acceleration time Index: `6083 (V 8.5.996 + higher)/5ef4 (to V 8.5.995)`, Short name: `RampBeschlZeit`	
SPP Windows	`Acceleration time 250 ms`
Type	simple variable, unsigned32
Var. type	axis machine data read and write
Unit	1 ms
Standard value	250 ms (value range 1 .. 32767)
Valid	in axis operating modes ***profile position mode***, ***velocity mode***, and ***homing mode***

The *acceleration time* specifies the time needed to accelerate to the *ramps reference velocity*. This ensures the same acceleration even with different target velocities.

Deceleration time Index: `6084 (V 8.5.996 + higher)/5ef5 (up to V 8.5.995)`, Short name: `RampVerzZeit`	
SPP Windows	`Deceleration time 250 ms`
Type	simple variable, unsigned32
Var. type	axis machine data read and write
Unit	1 ms
Standard value	250 ms (value range 1 .. 32767)
Valid	in axis operating modes ***profile position mode***, ***velocity mode***, and ***homing mode***

The *deceleration time* specifies the time needed to decelerate from the *ramps reference velocity* to 0. This ensures the same deceleration, even with different target velocities.

If control command “quick stop” in the axis status word is used for decelerating, the *quick stop time* is used instead of the *deceleration time*.

Quick stop time Index: `6085 (V 8.5.996 + higher)/5ef6 (to V 8.5.995)`, Short name: `RampSchnellZeit`	
SPP Windows	`Quick stop time 200 ms`
Type	simple variable, unsigned32
Var. type	axis machine data read and write
Unit	1 ms
Standard value	200 ms (value range 1 .. 32767)
Valid	in axis operating modes ***profile position mode***, ***velocity mode***, and ***homing mode***

The *quick stop time* specifies the time needed for decelerating from the *ramps reference velocity* to 0. It is only effective in case of a deceleration using control command "quick stop" in the axis control word. Otherwise, the *deceleration time* is valid.

Ramps reference velocity	Index: `5ef7`, Short name: `RampVelBezug`
SPP Windows	`Ramps reference velocity 3000 VU`
Type	simple variable, unsigned32
Var. type	axis machine data read and write
Unit	velocity units (VU)
Standard value	3,000 VU (value range 1 .. 32767)
Valid	in axis operating modes ***profile position mode***, ***velocity mode, homing mode***, ***electronic gearing***, and ***flying shear***

In operating modes **profile position mode**, **velocity mode**, and **homing mode**, the ramps reference velocity specifies the velocity to which the *acceleration time*, *deceleration time*, and *quick stop time* parameters apply. In operating modes **electronic gearing** and **flying shear**, it is the reference value for *ramp time el. gearing*.

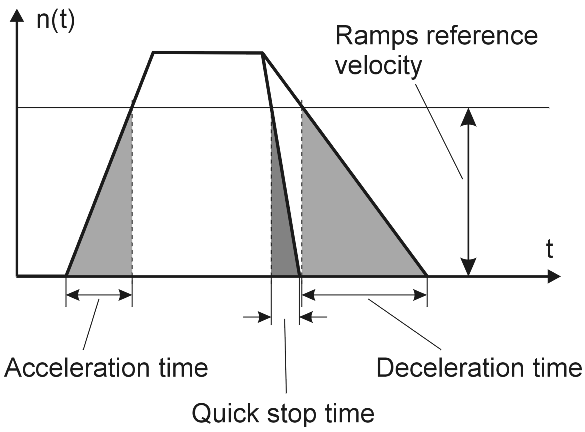


Figure 17: Acceleration and Deceleration Ramps

When the *ramps reference velocity* is changed, the slope of all ramps changes, as well.

The *ramps reference velocity* is converted internally to r.p.m. and must not be set to “0”. Depending on the definition of the velocity units (VU), low values higher than 0 VU can be rounded to 0 r.p.m.

To switch off the ramps, select motion profile type “jump” or set *acceleration time, deceleration time*, and *quick stop time* to zero.

9.4.8 Speed Control Loop Machine Data

The speed control loop machine data include the following variables:

- *Speed filter cut-off frequency*
- *Speed control loop Kp*
- *Speed control loop Ki*
- *Speed control loop total amplification*
- *Order of nonrecursive velocity filter*
- *Velocity threshold*
- *Velocity threshold time*
- *Velocity window*
- *Velocity window time*

In command and commissioning software SPP Windows, the speed control loop machine data can be found in the “Parameterization” window under *Axis data/ Speed control loop*.

For details on the setting of the speed control loop parameters, see section Setting Current, Speed, and Position Control Loop 71.

Speed filter cut-off frequency	Index: `5f22`, Short name: `VelIstFilterF0`
SPP Windows	`Speed filter cut-off frequency 100 Hertz`
Type	simple variable, unsigned16
Var. type	axis machine data read and write
Unit	1 Hz
Standard value	100 Hz
Valid	in all axis operating modes except for ***torque mode***; V 3.4 to V 8.5.9

This machine date is used for setting the cut-off frequency of the low-pass filter which is used for smoothing the actual speed determined from the motor position encoder signals. This is a digital low-pass of the second order; the 3 dB limiting frequency of the low-pass corresponds to double the value specified here.

Speed control loop Kp	Index: `5ef1`, Short name: `VelKp`
SPP Windows	`Speed control loop Kp 0.9800`
Type	simple variable, unsigned16
Var. type	axis machine data read and write
Unit	V 8.5.9 or higher: 2048 (800_{hex}) corresponds to a value of 1.0
Standard value	1.0 = 2048 (V 8.5.9 or higher) or 2000 (permissible value range 1 .. 32767)
Valid	in all axis operating modes except for ***torque mode***

Parameter *speed control loop Kp* sets the proportional gain of the digital speed control loop. Use commissioning aid speed control loop (see section Setting the Speed Control Loop 74).

Speed control loop Ki	Index: `5ef2`, Short name: `VelKi`
SPP Windows	`Speed control loop Ki 0.4900`
Type	simple variable, unsigned16
Var. type	axis machine data read and write
Unit	V 8.5.9 or higher: 4096 (1000_{hex}) corresponds to a value of 1.0
Standard value	1.0 = 4096 (8.5.9 or higher) or 4100 (permissible value range 1 .. 32767)
Valid	in all axis operating modes except for ***torque mode***

Parameter *speed control loop Ki* sets the gain of the integration of the digital speed control loop. Use commissioning aid speed control loop (see section Setting the Speed Control Loop 74).

Speed control loop total amplification	
	Index: `5f21` (axis 1), Short name: `VelKGesamt`
SPP Windows	`Speed control loop total amplification 1.000`
Type	simple variable, unsigned16
Var. type	axis machine data read and write
Unit	2048 (800_{hex}) corresponds to a value of 1.0
Standard value	2048 (corresponds to 1.0)
Valid	in all axis operating modes except for ***torque mode***; V 3.4 to V 8.5.9

The *speed control loop total amplification* can be used for increasing the loop gain of the speed control loop by a factor of 1 to 16. A value of 2048 corresponds to a factor of 1.0, and a value of 32767 to a factor of 16; higher or lower values must not be set. Doubling the *speed control loop total amplification* has the same effect as a simultaneous doubling of the machine data *speed control loop Kp* and *speed control loop Ki*.

It is particularly useful to change this machine date when machine data *speed control loop Kp* or *speed control loop Ki* are set to the maximum value and a further increase in loop gain is required.

Order of nonrecursive velocity filter	
	Index: `5F66`, Subindex: `11`,Short name: `VelFilter11`
SPP Windows	`Order of nonrecursive velocity filter 250 µs`
Type	record element, unsigned16
Var. type	axis machine data read and write
Unit	select
Standard value	250 µs
Valid	in all axis operating modes

The noise of the actual speed value can be influenced with this variable.

Higher times reduce the noise of the actual speed value. Lower times permit better speed dynamics. If applicable, higher amplification factors (Kp, Ki) of the speed control loop are possible with lower times.

Permissible values:

Value	Averaging via
1	1 current control loop cycles (62.5 µs)
2	2 current control loop cycles (125 µs)
4	4 current control loop cycles (250 µs)
8	8 current control loop cycles (500 µs)
16	16 current control loop cycles (1 ms)

Velocity threshold	Index: `606f`, Short name: `VelSchwelle`
SPP Windows	`Velocity threshold 20.0 ± r.p.m.`
Type	simple variable, unsigned32
Var. type	axis machine data read and write
Unit	0.5 r.p.m. (V 8.5.9 or higher) 0.25 r.p.m. (up to V 8.5.9)
Standard value	20 r.p.m.
Valid	in all axis operating modes except for ***torque mode***

The *velocity threshold* specifies the actual velocity range around zero velocity at which “velocity = 0” is reported.

In axis operating mode **velocity mode**, this message is indicated with bit 12 “Velocity = 0” in the axis status word.

This message is evaluated in all axis operating modes in order to generate the internal events “braking completed” and “quick stop completed” (see section Axis State Machine 17).

Message “velocity = 0” is only generated if the actual velocity is within the range of the *velocity threshold* at least for the *velocity threshold time*.

In order to safely determine that the velocity zero has been reached, the corresponding *velocity threshold* and *velocity threshold time* variables must be set in a way that, even if the axis is subject to motion as a result of control oscillations or an external force, this movement never exceeds the speed specified in the *velocity threshold.*

Velocity threshold time	Index: `6070`, Short name: `VelSchwelleZeit`
SPP Windows	`Velocity threshold time 0 ms`
Type	simple variable, unsigned16
Var. type	axis machine data read and write
Unit	1 ms
Standard value	0 ms
Valid	in all axis operating modes except for ***torque mode***

Message “velocity = 0” is only created if the actual velocity is within the range of the *velocity threshold* for at least the *velocity threshold time*.

Velocity window	Index: `606d`, Short name: `VelFenster`
SPP Windows	`Velocity window 20.0 ± r.p.m.`
Type	simple variable, unsigned32
Var. type	axis machine data — read and write
Unit	0.5 r.p.m. (V 8.5.9 or higher) 0.25 r.p.m. (up to V 8.5.9)
Standard value	±20 r.p.m.
Valid	in axis operating modes ***velocity mode***, ***velocity mode direct***, and ***electronic gearing***

The *velocity window* specifies the range of the actual velocity around the target velocity in which “target velocity reached” is reported with bit 10 of the axis status word. This message can also be output at a digital output. For that, see *signal selection digital output ...* in section Input/Output Function Machine Data [131].

Message “target velocity reached” is only generated if the actual speed is within the range of the *velocity window* at least for the *velocity window time*.

Velocity window time	Index: `606e`, Short name: `VelFensterZeit`
SPP Windows	`Velocity window time 0 ms`
Type	simple variable, unsigned16
Var. type	axis machine data — read and write
Unit	1 ms
Standard value	0 ms
Valid	in axis operating modes ***velocity mode***, ***velocity mode direct***, and ***electronic gearing***

Message “target velocity reached” is only generated if the actual speed is within the range of the *velocity window* at least for the *velocity window time*.

9.4.9 Position Control Loop Machine Data

The position control loop machine data include the following variables:

- *Position control loop Kp*
- *Position control loop Kp Divisor*
- *Positioning window*
- *Positioning window time*
- *Position monitoring*
- *Position monitoring time*
- *Kp feedforward speed*
- *Following error window*

- *Following error time out*

In command and commissioning software SPP Windows, the position control loop machine data can be found in the “Parameterization” window under *Axis data/Position control loop*.

Position control loop Kp	Index: `5ef0`, Short name: `PosKpp`	
SPP Windows	`Position control loop Kp` <u>`10`</u>	
Type	simple variable, unsigned16 (by V 2.4: unsigned8)	
Var. type	axis machine data	read and write
Unit	–	
Standard value	10	
Valid	in axis operating modes ***profile position mode*** and ***electronic gearing***	

Together with *position control loop Kp divisor*, parameter *position control loop Kp* sets the gain of the position control loop.

Position control loop Kp divisor	Index: `5f0d`, Short name: `PosKppDiv`	
SPP Windows	`Position control loop Kp divisor` <u>`60`</u>	
Type	simple variable, unsigned16	
Var. type	axis machine data	read and write
Unit	–	
Standard value	60	
Valid	in axis operating modes ***profile position mode*** and ***electronic gearing***; V 8.5.9 and higher	

The amplification of the position control loop results from *position control loop Kp* divided by *position control loop Kp divisor*.

Positioning window	Index: `6067`, Short name: `PosFenster`	
SPP Windows	`Positioning window` <u>`100`</u> `±PSS`	
Type	simple variable, unsigned32	
Var. type	axis machine data	read and write
Unit	position sensor steps (PSS)	
Standard value	±100 LgS	
Valid	in axis operating modes ***profile position mode*** and ***electronic gearing***	

The *positioning window* is placed symmetrically around the target position. As soon as the actual position is within this *positioning window*, bit 10 “position set-point reached” or “synchronized” is set in the axis status word. This message can also be output at a digital output. For that, see *signal selection digital output ...* in section Input/Output Function Machine Data [131].

Message “position setpoint reached” or “synchronized” is only generated if the actual position is within the *positioning window* at least for the *positioning window time*.

Positioning window time	Index: `6068`, Short name: `PosFensterZeit`
SPP Windows	`Positioning window time  0  ms`
Type	simple variable, unsigned16
Var. type	axis machine data read and write
Unit	1 ms
Standard value	0 ms
Valid	in axis operating modes ***profile position mode*** and ***electronic gearing***

The *positioning window time* prevents the setting of bit 10 “position setpoint reached” or “synchronized” in the axis status word if the *positioning window* is only briefly swept over. The axis has to be in the *positioning window* for the preset *positioning window time* before this bit is set.

Position monitoring	Index: `5f1f`, Short name: `PosUeberwach`
SPP Windows	`Position monitoring  100  ±PSS`
Type	simple variable, unsigned32
Var. type	axis machine data read and write
Unit	position sensor steps (PSS)
Standard value	±100 PSS
Valid	in axis operating mode ***profile position mode***; V 3.3b to V 8.5.995

The *position monitoring* is placed symmetrically around the target position. As soon as the actual position is outside the position range specified by the *position monitoring*, output “position monitoring” is set.

Output “position monitoring” is only set when the actual position is outside the range of the *position monitoring* for at least the *position monitoring time*.

Position monitoring time	Index: `5f20`, Short name: `PosUeberwachZeit`
SPP Windows	`Position monitoring time  0  ms`
Type	simple variable, unsigned16
Var. type	axis machine data read and write
Unit	1 ms
Standard value	0 ms
Valid	in axis operating mode ***profile position mode***; V 3.3b to V 8.5.995

The *position monitoring time* prevents the setting of output “position monitoring” if the position area around the target position specified by the *position monitoring* parameter is left for a short period of time. The axis must be outside the position area specified by the *position monitoring* parameter for the *position monitoring time* before this output is set.

Kp feedforward speed	`Index: 5F64`, Subindex: `4`, Short name: `TimedPosition4`
SPP Windows	`Kp feedforward speed 1.000`
Type	simple variable, float32
Var. type	axis machine data — read and write
Unit	–
Standard value	1.0
Valid	V 8.5.999.53 or higher

With this variable, the feedforward speed can be set.

Following Error Window	`Index: 6065`, Short name: `FollowingErrWindow`
SPP Windows	`Following error window 4294967295 PSS`
Type	simple variable, unsigned32
Var. type	axis machine data — read and write
Unit	PSS
Standard value	4294967295 ($FFFF\ FFFF_{hex}$)
Valid	in axis operating modes ***interpolated position mode*** and ***cyclic synchronous position mode;*** V 8.5.997 or higher

The *following error window* is located symmetrically around the target position. As soon as the actual position is outside the position range defined by the *following error window*, the device changes to “fault“ state and reports error “following error position control loop“.

Error “following error position control loop“ is not executed unless the actual position is outside the range *following error window* for at least the *following error timeout*.

Following Error Time Out	`Index: 6066`, Short name: `FollowingErrTimeout`
SPP Windows	`Following error time out 0 ms`
Type	simple variable, unsigned16
Var. type	axis machine data — read and write
Unit	ms
Standard value	M
Valid	in axis operating modes ***interpolated position mode*** and ***cyclic synchronous position mode;*** V 8.5.997 or higher

The *following error time out* prevents that error “following error position control loop“ is triggered when the position range around the target position defined by the *following error window* is left for a short period of time. The axis must be outside the position range defined by the *following error window* for the defined *following error time* before this error is triggered.

9.4.10 Homing Mode Machine Data

The homing mode machine data include the following variables:

- *Homing selection code*
- *Homing mode velocity*

In command and commissioning software SPP Windows, the homing mode machine data can be found in the “Parameterization” window under *Axis data/Homing mode*.

The *home offset* parameter is defined in section Factors and Units Machine Data 158.

Homing selection code	Index: `6098`, Short name: `RefAuswahl`
SPP Windows	`Homing selection code` `Selection code -1` `select`
Type	simple variable, integer8 (V 8.5.996 and higher) simple variable, integer16 (up to V 8.5.995)
Var. type	axis machine data read and write
Unit	select
Standard value	selection code –1
Valid	in axis operating mode ***homing mode*** (if position measuring system = incremental encoder)

The *homing selection code* defines the way the machine home position is approached. Section Homing Mode Sequences 40 describes the possible sequences and lists the permitted values for the *homing selection code*.

Homing velocity	Index: `6099`, Short name: `HomingSpeed`
SPP Windows	`Velocity during search for switch` `500` `VU` `Velocity during search for zero` `500` `VU`
Type	array, 2 type unsigned32 elements (V 8.5.996 or higher) simple variable, unsigned32 (up to V 8.5.9)
Element 1	velocity during search for switch (V 8.5.999 or higher)
Element 2	velocity during search for zero (V 8.5.999 or higher)
Var. type	axis machine data read and write
Unit	velocity units (VU)
Standard value	500 VU
Valid	in axis operating mode ***homing mode***

The *homing velocity* specifies the velocity at which the machine home position is approached. The *velocity during search for switch* defines the velocity at which the reference switch is approached. The *velocity during search for zero* defines the velocity at which the zero point is approached. For devices with firmware versions below V 8.5.9, only one *homing velocity* that can be set exists.

The *homing velocity* must be selected in a way that the frequency of the incremental encoder pulses does not exceed 35 kHz after internal quadruplication, so that the machine home position can be clearly recognized. This corresponds to a speed of 1050 r.p.m. with an incremental encoder resolution of 2000 pulses per revolution after quadruplication (an incremental encoder with 500 pulses per revolution).

9.4.11 Electronic Gearing Machine Data

The electronic gearing machine data include the following variables:

- *External master axis selection code*
- *External position sensor resolution*
- *Gear ratio*
- *Speed control loop Kv electronic gearing*
- *Ramp time el. gearing*
- *Master velocity threshold*
- *Electronic gearing control*

Parameter *speed filter cut-off frequency master* is not documented here as it does not have an effect.

In command and commissioning software SPP Windows, the electronic gearing machine data can be found in the “Parameterization” window under *Axis data/ Electronic gearing*.

External master axis selection code Index: `5f17`, Short name: `ElgExternAuswahl`	
SPP Windows	`External master axis` <u>`external encoder`</u> `select`
Type	simple variable, integer16
Var. type	axis machine data — read and write
Unit	select
Standard value	external encoder
Valid	in axis operating modes ***electronic gearing*** and ***flying shear***; only for devices with -Lx or -ZLx option

Variable *external master axis selection code* is used for setting the interface via which the position setpoints are read in axis operating modes **electronic gearing** and **flying shear**.

Valid values are:

Selection code	Signal designation
1	external encoder quadrature signal at the encoder signals input (option Lx or ZLx)
2	process data field bus (currently not implemented)
3	external encoder pulse/direction (V 8.5.9 or higher)

External position sensor resolution	Index: `5f72`, Short name: `PosAuflExtern`	
SPP Windows	`External position sensor resolution 4096 pulses/rev. or PSS/rev.`	
Type	array, 2 type unsigned32 elements	
Array element 1	pulses (V 8.5.9 or higher) position sensor steps (PSS) (up to 8.5.9)	
Array element 2	motor revolutions (always 1)	
Var. type	axis machine data	read and write
Unit	position sensor steps (PSS) per revolution	
Standard value	4096	
Valid	in axis operating modes ***electronic gearing*** and ***flying shear***; only for devices with -Lx or -ZLx option	

Depending on the firmware version, the *external position sensor resolution* can be used for setting the resolution of the external encoder directly in pulses/revolution or converted into position sensor steps (PSS) per revolution. If the master axis delivers quadrature signals, the number of position sensor steps is four times higher that the number of pulses per revolution. Examples:

- For an encoder with 250 pulses per revolution, the value 250 pulses/rev. or 1000 PSS/rev. is set for the *external position sensor resolution*.
- If a further servo drive with encoder emulation with 1024 pulses per revolution is used as the master axis, value 1024 pulses/rev. or 4096 PSS/rev. must be entered as *external position sensor resolution*.

Gear ratio	
Index: 5efa (V 8.5.999.3 + higher)/`5f06 (up to V 8.5.999.08)`, Short name: `ElgUeInkrem`	
SPP Windows	`Gear ratio numerator (slave axis) 1 PSS` `Gear ratio denominator (master axis) 1 pulses or PSS`
Type	array, 2 type integer16 elements
Array element 1	numerator in position sensor steps (PSS) of the slave axis
Array element 2	denominator in impulses of the master axis (V 8.5.999.3 or higher) denominator in position sensor steps (PSS) of the master axis (up to V 8.5.999.08)
Var. type	axis machine data read and write
Unit	pulses or position sensor steps (PSS)
Standard value	1 : 1
Valid	in axis operating modes ***electronic gearing*** and ***flying shear***; only for devices with -Lx or -ZLx option

The *gear ratio* is a quotient indicating the ratio position and velocity of the slave axis follow the values of the master axis. A value of 1:10 indicates that 1 step of the slave axis corresponds to 10 steps of the master axis. Please, note that the coordinate systems of master and slave axis are usually different. For a gear ratio of 1:1 in revolutions, please enter a ratio of 65,536 PSS to the *external position sensor resolution* which considers the maximum value range (-32768 .. +32767).

Example:

- position sensor resolution = 65,536 PSS/rev.
- external position sensor resolution = 4,000 pulses/rev.
- gear ratio numerator (slave axis) = 16,384 (= 65,536 : 4)
- gear ratio denominator (master axis) = 1,000 (= 4,000 : 4)

has the effect that 1 revolution of the master axis corresponds to 1 revolution of the slave axis.

The direction of rotation of the slave axis can be switched by a negative sign in the numerator or denominator.

Basically, any value can be set for numerator and denominator, however, it must be ensured that design and dynamics of the drive of the slave axis are suitable for following the step ratios.

Speed control loop Kv electronic gearing	Index: `5f07`, Short name: `ElgKv`
SPP Windows	`Speed control loop Kv electronic gearing 1.00000`
Type	simple variable, unsigned16
Var. type	axis machine data read and write
Unit	0 corresponds to Kv = 0; 32767 corresponds to Kv = 1.0
Standard value	32767 (corresponds to Kv = 1.0)
Valid	in axis operating modes ***electronic gearing*** and ***flying shear***; V 4.2 or higher, only for devices with -Lx option

For optimum operation of the slave axis in axis operating modes **electronic gearing** and **flying shear**, a velocity feedforward for the speed control loop is implemented. This sets the velocity of the slave axis to that of the master axis under consideration of the set gear ratio. The influence of this feedforward can be adjusted via the *speed control loop Kv electronic gearing* parameter by means of a gain factor with the value range 0 to 1.

Normally, the value is set to 1. Depending on the application, it can be reduced to 0, which switches off the velocity feedforward. In this case, the electronic gearing is adjusted by the position control loop via the position deviation of the slave axis. Generally, the gain *speed control loop Kv electronic gearing* should be left at the default value 1.

The value range for the gain Kv is 0 .. 1; value 0 of this parameter corresponds to a gain of 0 and value 32767 ($7FFF_{hex}$) to a gain of 1.

Ramp time el. gearing	Index: `5f32`, Short name: `FlsRampZeit`	
SPP Windows	`Ramp time 300 ms`	
Type	simple variable, unsigned32	
Var. type	axis machine data	read and write
Unit	1 ms	
Standard value	300 ms	
Valid	in axis operating modes ***electronic gearing*** and ***flying shear***; V 4.5 or higher, only for devices with -Lx or -ZLxoption	

After the “synchronize” command in operating modes **electronic gearing** and **flying shear**, the slave axis is accelerated by the ramp set in the *ramp time el. gearing*. The reference quantity is (as for the other ramp times) the *ramps reference velocity* from the ramps machine data, see section Ramps Machine Data 164.

To catch up on the lead of the master axis, the slave axis is accelerated above the velocity of the master axis, deceleration to the target velocity is carried out with the same ramp.

The ramp function is switched off by setting the *ramp time el. gearing* to 0.

Master axis velocity threshold	Index: `5f33`, Short name: `VelElgSollSchwelle`	
SPP Windows	`Master velocity threshold 5.0 r.p.m.`	
Type	simple variable, unsigned32	
Var. type	axis machine data	read and write
Unit	0.25 r.p.m.	
Standard value	5 r.p.m.	
Valid	in axis operating modes ***electronic gearing*** and ***flying shear***; V 4.6 to V 8.5.9; only for devices with -Lx or -ZLx option	

The *master axis velocity threshold* specifies the range around zero velocity within which the velocity of the master axis is evaluated as a standstill (compare the *velocity threshold* variable in section Speed Control Loop Machine Data 167).

The value should be at least 30% higher than the velocity, which produces an encoder pulse frequency of 250 Hz. Example

- An encoder with 12000 PSS/rev. resolution generates 3000 pulses per revolution. 250 pulses per second (250 Hz) are achieved as follows

  250 pulses/sec / 3000 pulses/rev. = 0.083 rev./sec = 5 r.p.m.

  Including a safety margin of 30%, the *master axis velocity threshold* must not be set to less than 6.5 r.p.m.

Electronic gearing control	Index: 5F08, Short name: ElgFreqOnly
SPP Windows	Electronic gearing control Electronic gearing select
Type	simple variable, integer16
Var. type	axis machine data read and write
Unit	select
Standard value	electronic gearing
Valid	in axis operating mode ***electronic gearing***; V 8.5.9 or higher

(Description in preparation)

Permissible values:

Value	Electronic gearing control
–1	Electronic gearing
–2	Velocity mode

9.4.12 Flying Shear Machine Data

The flying shear machine data include the following variable:

- *Cutting length*

In command and commissioning software SPP Windows, the flying shear machine data can be found in the “Parameterization” window under *Axis data/Flying shear*.

Cutting length	Index: `5f31`, Short name: `FlsSchnittLaenge`
SPP Windows	`Cutting length 0 LE`
Type	simple variable, unsigned32
Var. type	axis machine data read and write
Unit	position units
Standard value	0 PU
Valid	in axis operating mode ***flying shear***; V 4.5 to V 8.5.9; only for devices with -Lx or -ZLx option

The *cutting length* is used for defining the distance of the synchronizing points and thus the length of the pieces into which the processed material is divided.

9.4.13 Spindle Positioning Machine Data

The spindle positioning machine data include the following variables:

- *Spindle position*
- *Spindle positioning velocity*
- *Spindle positioning window*
- *Spindle positioning windowtime*
- *Spindle positioning direction*

In command and commissioning software SPP Windows, the spindle positioning machine data can be found in the “Parameterization” window under *Axis data/ Spindle positioning*.

Spindle position	Index: `5ea3`, Short name: `SpindelPosZiel`
SPP Windows	`Spindle position 0 PU`
Type	simple variable, integer32
Var. type	axis machine data read and write
Unit	position units (PU)
Standard value	0
Valid	for spindle positioning, in axis operating modes ***velocity mode direct*** and ***velocity mode***; V 5.4 or higher

The *spindle position* defines the target for spindle positioning. It is specified as circular axis position and must be within the positioning range (*range limit min* and *range limit max*).

Spindle positioning velocity	Index: 5ea4, Short name: SpindelVelVerfahr
SPP Windows	Spindle positioning velocity 0 VU
Type	simple variable, integer32
Var. type	axis machine data read and write
Unit	velocity units (VU)
Standard value	0
Valid	for spindle positioning, in axis operating modes ***velocity mode direct*** and ***velocity mode***; V 5.4 or higher

With parameter *spindle positioning velocity*, the target velocity of spindle positioning is set.

Spindle positioning window	Index: 5ea5, Short name: SpindelPosFenster
SPP Windows	Spindle positioning window 100 PSS
Type	simple variable, integer32
Var. type	axis machine data read and write
Unit	position sensor steps (PSS)
Standard value	100
Valid	for spindle positioning, in axis operating modes ***velocity mode direct*** and ***velocity mode***; V 5.4 or higher

The *spindle positioning window* is placed symmetrically around the *spindle position*. As soon as the actual position is within this *spindle positioning window*, signal "in position" is generated. This message can also be output at a digital output, see *signal selection digital output ...* in section Input/Output Function Machine Data 131.

Spindle positioning window time	Index: 5ea6, Short name: SpindelPosFenstZeit
SPP Windows	Spindle positioning window time 0 ms
Type	simple variable, unsigned16
Var. type	axis machine data read and write
Unit	1 ms
Standard value	0
Valid	for spindle positioning, in axis operating modes ***velocity mode direct*** and ***velocity mode***; V 5.4 or higher

The *spindle positioning window time* prevents the generation of signal "in position" if the *spindle positioning window* is only briefly swept over. The axis must be in the *spindle positioning window* for the set *spindle positioning window time* before this signal is generated.

Spindle positioning direction	Index: `5ea7`, Short name: `SpindelPosParameter`
SPP Windows	`Spindle positioning direction` `last direction of rotation` `select`
Type	simple variable, octet string length 2
Var. type	axis machine data read and write
Unit	select
Standard value	last direction of rotation
Valid	for spindle positioning, in axis operating modes ***velocity mode direct*** and ***velocity mode***; V 5.4 or higher

Parameter *spindle positioning direction* is used for setting the direction of rotation in which the spindle position is approached.

Permitted values:

Value	Direction of rotation
0	to the right
1	to the left
3	last direction of rotation

9.4.14 Output Encoder Signals Machine Data

The output encoder signals machine data include the following variables:

- *Pulses encoder signals*
- *Zero offset encoder signals*

These machine data are only of significance for option Gx (output encoder signals).

In command and commissioning software SPP Windows, the output encoder signals machine data can be found in the “Parameterization” window under *Axis data/Encoder emulation*.

Pulses encoder signals	Index: `5ecc`, Short name: `DSPIncPU`
SPP Windows	`Pulses per revolution` `1024`
Type	simple variable, integer16
Var. type	axis machine data read and write
Unit	pulses/motor revolution
Standard value	1024
Valid	for output encoder signals (option Gx) with resolver as motor position sensor (option R1)

This variable can be used for setting the number of pulses per motor revolution output encoder signals (option Gx) outputs on channel A and B. The value range for this parameter is 50 .. 1024 pulses per motor revolution. If a controller is connected, this value can be doubled or quadrupled through evaluation of the pulse edges.

The *pulse encoder signals* variable is only effective if a resolver (option R1) is implemented as motor position sensor. If a Sincos (Hiperface) encoder (option R2), a high-resolution incremental encoder (option R3), or an EnDat encoder is implemented, the pulse number for output encoder signals (option Gx) corresponds to the number of sine periods of the motor position sensor (e.g. 512 or 2048 sine periods per motor revolution).

Zero offset encoder signals	Index: `5ecb`, Short name: `EncOffNull`	
SPP Windows	`Zero offset encoder signals  0  pulses`	
Type	simple variable, unsigned16	
Var. type	axis machine data	read and write
Unit	pulses	
Standard value	0	
Valid	for output encoder signals (option Gx) with resolver as motor position sensor (option R1)	

This variable can be used for offsetting the index pulse of the output encoder signals from the mechanically preset position by a specific number of pulses. The highest permissible value is *pulse encoder signals*–1 (offset by almost one revolution).

Zero offset encoder signals is only effective if a resolver (option R1) is implemented as motor position sensor. If a Sincos (Hiperface) or an EnDat encoder (option R2 or R4) is implemented, no index pulse is available at output encoder signals (option Gx), and thus it is not possible to carry out an index pulse shift. With a high-resolution incremental encoder (option R3), the index pulse cannot be shifted.

9.4.15 Switching Points Machine Data

The switching points machine data include the following variables:

- *Position switching point positive*
- *Position switching point negative*
- *Position switching point inverted*

In command and commissioning software SPP Windows, the switching point machine data can be found in the “Parameterization” window under *Axis data/ Switching points*.

Position switching point positive	Index: 5f36, Short name: PosPSchaltpunkt
SPP Windows	Position switching point positive 1 0 PU
Type	array, 16 type integer32 elements
Var. type	axis machine data read and write
Unit	position units (PU)
Standard value	0 PU
Valid	always, V 5.1 or higher

Position switching point negative	Index: 5f3a, Short name: PosNSchaltpunkt
SPP Windows	Position switching point negative 1 2.147.483.647 PU
Type	array, 16 type integer32 elements
Var. type	axis machine data read and write
Unit	position units (PU)
Standard value	2,147,483,647 PU
Valid	always, V 5.4 or higher

Two positions are specified with *position switching point positive* and *position switching point negative*. If the actual position of the axis is between these two positions, the associated bit in the *position switching point status* is set; if the actual position is lower than *position switching point positive* or higher than or equal to *position switching point negative*, the corresponding bit is 0.

With *position switching point inverted*, this logic can be inverted.

Position switching point inverted	Index: 5f3b, Short name: PosNOTSchaltpunkt
SPP Windows	Position switching point inverted 00000000
Type	simple variable, octet string length 2
Var. type	axis machine data read and write
Unit	–
Standard value	00000000
Valid	always, V 5.4 or higher

The *position switching point inverted* variable can be used for inverting the logic for each switching point:

- If the bit in position *switching point inverted* is 0, the associated bit in *position switching point status* indicates whether or not the actual position is located **within** the limits *position switching point positive* and *position switching point negative*.
- If the bit in position switching point inverted is 1, the associated bit in position switching point status indicates whether or not the actual position is located outside the limits position switching point positive and position switching point negative.

9.4.16 Interpolated Position Mode Machine Data

The interpolated position mode machine data include the following variables:

- *Interpolation Submode Select*
- *Interpolation Time Period*
- *Interpolation Sync Definition*
- *Interpolation Data Configuration*
- *Following error window*
  Description see Position Control Loop Machine Data 171
- *Following error time out*
  Description see Position Control Loop Machine Data 171

In command and commissioning software SPP Windows, the **interpolated position mode** machine data can be found in the “Parameterization” window under *Axis data/Interpolated position mode*.

The individual CANopen® or EtherCAT variables such as e. g. transmit and receive PDO can be found in operating instructions 6710.205 / 6745.205 “CANopen® Interface” or 6745.232 “EtherCAT Interface”. In the SPP Windows software, they can be found in window “Parameterization” under *CANopen/ EtherCAT*. With TrioDrive D, MidiDrive D and MaxiDrive, this window is only displayed if the servo drive is accessed via the fieldbus interface. Display of these variables with access via the serial interface is not possible with these devices.

Interpolation Sub Mode Select	Index: `60c0`, Short name: `IpolSubmodeAuswahl`
SPP Windows	`Interpolation Submode  0`
Type	simple variable, integer16, constant (variable value does not change)
Access	read only, writing possible but without effects
Var. type	axis machine data
Unit	–
Standard value	0 (linear interpolation)
Valid	in axis operating mode ***interpolated position mode***; V 8.5.1 or higher

This variable determines the interpolation mode. Only linear interpolation is supported.

Interpolation Time Period	Index: `60c2`, Short name: `IpolTimePeriod`
SPP Windows	`Interpolation Time Units 4 10^idx s` `Interpolation Time Index –3`
Type	record, 2 elements (subindex 0 .. 2)
Record element 1 **Record element 2**	Interpolation time units (1 .. 10), unsigned8 Interpolation time index (–3), integer8
Var. type	axis machine data
Access	record element 1: read and write record element 2: read only, writing possible but without effects
Unit	–
Standard value	4 ms
Valid	in axis operating mode ***interpolated position mode***; V 8.5.1 or higher

The time interval at which the target positions are transmitted in **interpolation position mode** are determined via the *interpolation time unit*. The *interpolation time index* determines the unit for record element 1 (-3 for 10^{-3} s = 1 ms).

Interpolation Sync Definition	Index: `60c3`, Short name: `IpolSyncDefinition`
SPP Windows	`Synchronize on group 0` `ip_sync every n events 1`
Type	array, 2 elements (subindex 0 .. 2), unsigned8, constant (variable value does not change)
Array element 1 **Array element 2**	synchronize on group = 0 ip sync every n events = 1
Var. type	axis machine data
Access	read only, writing possible but without effects
Unit	–
Valid	in axis operating mode ***interpolated position mode***; V 8.5.1 or higher

Interpolation Data Configuration	Index: `60c4`, Short name: `IpolDataConfig`
SPP Windows	–
Type	record, 6 elements (subindex 0 .. 6); constant (variable value does not change)
Record element 1 **Record element 2** **Record element 3** **Record element 4** **Record element 5** **Record element 6**	max. buffer size, unsigned32; = 1 actual size, unsigned32; = 1 buffer organization, unsigned8; = 0 (FIFO buffer) buffer position, unsigned16; = 0 size of data record, unsigned8; = 1 buffer clear, unsigned8; = 1
Var. type	axis machine data
Access	read only, writing possible but without effects
Unit	–
Valid	in axis operating mode ***interpolated position mode***; V 8.5.1 or higher

These variables determine synchronization and configuration of the **interpolated position mode**. The values cannot be changed.

9.4.17 Timed Positioning Mode Machine Data

The **timed positioning mode** machine data include the following variables:

- *Positioning distance single movement*
- *Positioning time single movement*
- *Curve form*
- *Kp velocity feedforward*

In command and commissioning software SPP Windows, the **timed positioning mode** machine data can be found in the “Parameterization” window under *Axis data/Timed positioning mode*.

Variable *Kp velocity feedforward* can be found under *Axis data/Position control loop*.

Timed positioning mode	Index: `5F64`, Short name: `TimedPosMode`
Type	record, 5 elements
Var. type	axis machine data — read and write
Valid	in axis operating mode ***timed positioning mode***; V 8.5.995 or higher

The parameters and actual values **timed positioning mode** are combined in record variable *timed positioning mode*. The individual elements are described in the following.

Position distance single movement	Index: `5F64`, Subindex: `1`, Short name: `TimedPosMode1`
SPP Windows	`Position distance single movement  0  LgS`
Type	simple variable, unsigned32
Var. type	axis machine data — read and write
Unit	1 PSS
Standard value	–

For function **timed positioning mode**, the (relative) distance (single distance) is preset via this variable.

Position time single movement Index: 5F64, Subindex: 2, Short name: TimedPosMode2	
SPP Windows	`Position time single movement 0 ms`
Type	simple variable, unsigned32
Var. type	axis machine data read and write
Unit	1 ms
Standard value	–

For function **timed positioning mode,** the duration of the positioning (for the single movement) is preset using this variable.

Curve form Index: 5F64, Subindex: 3, Short name: TimedPosMode3	
SPP Windows	`Curve form triangle select`
Type	simple variable, integer16
Var. type	axis machine data read and write
Unit	select
Standard value	-1 (triangle)

For function **timed positioning mode,** the curve form is selected using this variable.

Permissible values are:

Value	Curve form
–1	Triangle (linear VDI 2143: “quadratic parabola”)
–2	Modified acceleration trapezoid (acc. to VDI 2143)
–3	Modified sine curve (acc. to VDI 2143)
–4	Inclined sine curve (acc. to VDI 2143)
–5	5th degree polynomial (acc. to VDI 2143)

10 Appendix

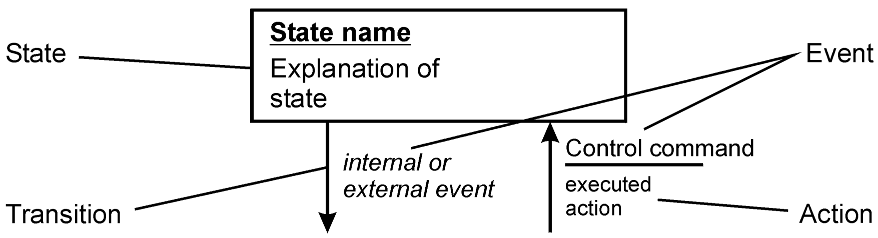
10.1 Appendix A State Machines

State machines describe the behavior of systems. The graphical representation of a state machine is called state diagram.

The elements of a state machine are

- *states*
- *transitions*
- *events*
- *actions*

A representation of the elements of a state machine in the state diagram is shown in the following figure:



BN 2555.0000.00B.250

Figure 18: Elements of state machines

States (shown as a rectangle with the name of the state and further explanations, if required) can be changed by **transitions** (displayed as arrows). A transition is performed when an **event** occurs. In the case of a transition, an **action** is carried out (represented below the event, separated from it by a line). Transmissions for which an action is not carried out are also permitted.

In the application of state machines, the following types of events are differentiated:

- Control commands

  These are events the user can trigger by writing the control command into a control word. Each control command is coded as bit pattern.

- Internal or external events

  Internal or external events (represented in italics) are triggered by the servo drive or a component connected to one of its interfaces. This can be e. g. a fault or the fact that a certain velocity has been reached.

10.2 Appendix B Axis Fault Codes and Part Program Errors

If a fault occurs, the device status (displayed in the axis status word) switches to the “Fault” state and the corresponding axis fault code is displayed. Once the cause of the fault is eliminated, the fault state can be reset by changing the “Reset fault” bit from 0 to 1 in the axis control word.

In the case of faults occurring during startup or while the part program is running (axis error code $62..._{hex}$), the program state machine switches to the “Idle” state from where the part program can be restarted once the fault state is reset.

If a fault occurs while the part program is running, the “Program block number display” shows the corresponding block in the part program.

The following list shows the possible axis error codes (hexadecimal) with fault designation and a short explanation. They are grouped according to their function:

- General fault codes
- Part program fault codes
- Other fault codes
- CANbus fault codes
- EtherCAT fault codes
- Profibus fault codes

General fault codes

0000_{hex} no fault

2320_{hex} Overcurrent motor phase

There is a short circuit in the motor circuit, or the chosen motor data do not match the connected motor.

2330_{hex} Earth fault motor phase

There is an earth fault in the motor circuit, or the chosen motor data do not match the connected motor.

3130_{hex} Phase failure

Failure of one or several phases L1, L2, or L3 at X1. Only for MidiDrive D/xS.

3210_{hex} Overvoltage in the DC-bus

Check whether or not the bridge between +R/R_{int} at the shunt resistor connection is implemented or the external shunt resistor is connected correctly.

3220_{hex} Undervoltage in the DC-bus

Mains voltage is missing, check power supply and connection. If there is no fault: a fuse may have blown in the shunt circuit, please contact the factory.

4210_{hex} Overtemperature power circuit (> 80 ºC)

The ambient temperature is too high or the motor supply cable is too long (remedy: motor choke).

4310_{hex} Overtemperature motor (threshold depends on the temperature sensor in the motor)

Insufficient heat dissipation in the motor or current limit set to an inadmissibly high value for the actual operating mode.

5110_{hex} Hardware fault low voltage supply

Check that the control supply voltage is +24 V_{DC} (see the technical specifications in the operating instructions of the servo drive). If there is no fault: please contact the factory.

5520_{hex} Hardware fault flash memory

Please contact the factory.

6010_{hex} Watchdog reset of the microcontroller

Please contact the factory.

Part program fault codes

6210_{hex} Unknown block type

The block type of a block cannot be executed with the existing firmware version of the servo drive. This may be due to the fact that the loaded part program was created with SPP Windows using a newer version of the object directory (OD), which knows more (newer) block types than the existing firmware.

6220_{hex} Label not found

The label used by a block of type *jump to label*, *repeat*, *jump on input*, or *subroutine call* does not exist anywhere in the part program.

6221_{hex} Start label not found

The label necessary for starting the part program (program start label) does not exist anywhere in the part program.

6231_{hex} Subroutine stack overflow

The maximum nesting depth of 10 subroutine calls (block type *subroutine call*) has been exceeded.

6232_{hex} Subroutine stack underflow

Failed to execute a block of type *return from subroutine* because there was no subroutine call beforehand.

6233_{hex} Repeat list overflow

The maximum number of 10 simultaneously active iterations has been exceeded by one block (block type *repeat*).

6235_{hex} Object access fault

Fault with a block of the type *object access*. This may be due to the following causes:

- The subindex is invalid for the specified index.
- The type of access (read or write) is not permitted for the variable.
- Access is not permitted due to the device status.

6236_{hex} Object does not exist

Fault with a block of type *object access*. The variable with the specified index does not exist in the servo drive.

6237_{hex} Object too long

Fault with a block of type *object access*. The variable is larger than 4 bytes and can therefore not be read or written to with this block type.

6250_{hex} Axis access fault

A fault has occurred while accessing an axis function (block types *position*, *feed*, *go to reference point*). Possible causes: selection of a non-existent axis or the target position is outside the position limit values.

6255_{hex} Fault while going to home position

A fault has occurred while going to home position (block type *go to reference point*).

6257_{hex} Change of operating mode not permitted

The operating mode cannot be changed in the following states:

- operation enabled
- quick stop active
- fault reaction active
- fault

6270_{hex} Internal I/O access fault

A fault has occurred while accessing an I/O function in a block of type *position, M function, jump on input, wait for input, switching point,* or *wait for input value and jump*.

6275_{hex} Division by zero

Fault during execution of a mathematical operation.

6280_{hex} Internal timer fault

An internal fault has occurred in block type *wait*.

Other fault codes

$54A1_{hex}$ Safe standstill SI1-Err inactive

Fault in the two-channel switching of the safety function: input SI2 active, but SI1 inactive.

$54A2_{hex}$ Safe standstill SI2-Err inactive

Fault in the two-channel switching of the safety function: input SI1 active, but SI2 inactive.

6310_{hex} Flash memory parameter loss

Please contact the factory.

6330_{hex} Data fault during initialization

For further details on the cause of this fault: refer to object “fault detail”. Please contact the factory.

6331_{hex} Data fault during initialization, object access

For further details on the cause of this fault: refer to the object “fault detail”. Please contact the factory.

7112_{hex} Overcurrent in the shunt circuit

A short circuit has occurred in the shunt circuit or the value of the external shunt resistor is too low (see technical specifications in the operating instructions of the servo drive).

7115_{hex} Overtemperature shunt resistor

The internal shunt resistor is overloaded, an external shunt resistor is required.

7116_{hex} Overtemperature brake chopper

The external shunt resistor is overloaded, please contact the factory.

7121_{hex} Motor is blocked

Motor is at standstill for longer than 10 s at full rated torque (overload).

$712F_{hex}$ Autocommutation not successful

The commutation position could not be found by the selected autocommutation method. Check the settings and make sure that the axis can move freely and is not blocked or stiff.

7303_{hex} Encoder fault motor position sensor

The motor position sensor is either connected not at all or not correctly, there is a cable break or a short circuit, or the voltage is too low. Another possible cause might be interference signals in the encoder connection cable e. g. due to a not or not correctly connected shield in the motor supply cable (power).

7304_{hex} Transmission error or missing data for digital position transmission (encoder)

The digital position data are wrong. Possible causes: the position sensor is not connected correctly, a cable break or short circuit has occurred. Another possible cause might be interference signals in the encoder connection cable, e.g. due to a not or not correctly connected shield in the motor supply cable (power).

7305_{hex} SinCos voltage monitoring Z track / incorrect encoder voltage Z track

The motor position sensor is either connected not at all or not correctly, there is a cable break or the voltage is too low.

7314_{hex} Transmission error or missing data for digital position transmission (ext. encoder)

The digital position data are wrong. Possible causes: the position sensor is not connected correctly, a cable break or short circuit has occurred. Another possible cause might be interference signals in the encoder connection cable, e.g. due to a not or not correctly connected shield in the motor supply cable (power).

7321_{hex} Sincos encoder parameter channel fault

The motor position sensor is either connected not at all or not correctly, or there is a cable break.

7322_{hex} Motor position sensor not supported

This firmware does not support the connected or parameterized encoder system.

7323_{hex} Fault parameterization motor position sensor

The motor position sensor was parameterized with invalid data which are not supported by this firmware.

7331_{hex} Fault parameter channel EnDat encoder (external encoder)

The position sensor is either connected not at all or not correctly or a cable is broken.

7332_{hex} External encoder not supported

The connected or parameterized external encoder is not supported by this firmware.

7333_{hex} Incorrect parameterization of the external encoder

The external encoder was parameterized with incorrect data not supported by this firmware version.

8611_{hex} Following error position control loop

Taking into account *following error time out*, the actual position is outside the *following error window*.

9000_{hex} External fault

Enable canceled at enable selection code “1” (*enable selection code* variable)

9010_{hex} Signal processor fault

Please contact the factory

$A011_{hex}$ Limit switch configuration fault

Check that the limit switches are assigned correctly.

$A012_{hex}$ “Limit switch +” approached

Input “limit switch +” of the servo drive interface inactive.

$A013_{hex}$ “Limit switch –” approached

Input “limit switch –” of the servo drive interface inactive.

$A020_{hex}$ Fault while going to home position

e. g. limit switch instead of machine home switch found.

Fault codes CAN bus

8130_{hex} CAN bus fault: Life Guarding

See operating instructions 6710.205 or 6745.205 “CANopen® interface”: “Life Guarding”.

8180_{hex} CAN bus fault: Bus-Off

See operating instructions 6710.205 or 6745.205 “CANopen® interface”: “Send and receive monitoring”.

Fault codes EtherCAT

A000 EtherCAT state machine transition error

Incorrect configuration of the sync managers for input or output process data (e. g. wrong length)

Fault codes Profibus

8140_{hex} Profibus master WD timeout

The watchdog time of the master response monitoring was exceeded.

$81B0_{hex}$ Profibus fatal error

Fault in the Profibus communication process.

10.3 Appendix C Firmware Versions Regarding Functions and Parameters

This section summarizes notes on changes in the firmware with references to the corresponding sections in the text. The latest modifications are listed first.

If you have worked with a servo drive with an older firmware, before (e. g. V 8.0) and have obtained a new servo drive with a new firmware (e. g. V 8.5.8), please observe all sections referring to changes between the two version numbers.

This appendix lists all firmware changes regarding the functions and parameters described in these operating instructions. For further firmware modifications, also see the respective appendices of the other operating instructions for the servo drives.

Changes V 8.5.999.65 compared with V 8.5.999.64:

- New *homing selection codes*: 17 to 30 as well as 33 and 34

Changes V 8.5.999.64 compared with V 8.5.999.63:

New variable:

- *Brake delay times* (index 5F18, subindex 1 and 2)

Changes V 8.5.999.63 compared with V 8.5.999.62:

- Values 1 and 2 are no longer available for the *release brake selection code* (index 5F16).

New variable:

- *Manufacturer identification* (index 5E86, subindex 6)

Changes V 8.5.999.62 compared with V 8.5.999.61:

- In operating mode **cyclic synchronous position mode** the unit for *target position* (index 607A) is PSS (else: LE)

Changes V 8.5.999.61 compared with V 8.5.999.55:

New variable:

- *Position transmission motor sensor* (index 5E81, subindex 9)

Changes V 8.5.999.55 compared with V 8.5.999.54:

New variable:

- *Order of nonrecursive velocity filter* (index 5F66, subindex 11)

Changes V 8.5.999.54 compared with V 8.5.999.53:

New variables:

- *Target velocity* (index 60FF)
- *Velocity offset* (index 60B1)
- *Velocity feedforward* (index 5F64, subindex 4)

Changes V 8.5.999.53 compared with V 8.5.999.50:

New variable:

- *Torque offset / force offset* (index 60B2)

Changes V 8.5.999.50 compared with V 8.5.999.3:

New variable:

- *External sensor encoder data* (index 5E86)

Changes V 8.5.999.3 compared with V 8.5.999.08:

- In addition to “external quadrature encoder” (selection code 01) setting “external impulse/direction encoder” (selection code 03) is available for variable *external master axis* (index 5F17).
- *External position sensor resolution* (index 5F72) includes the resolution of the external position sensor in pulses/revolutions (before: PSS/revolution).
- The index for variable *gear ratio* is 6091, the type is integer32 (before: index 5F06, type integer16). The unit for the denominator (master axis) is pulses (before: PSS).

Changes V 8.5.999.08 compared with V 8.5.997:

New variable:

- *Measuring length* (index 5E81, subindex 10)

Changes V 8.5.997 compared with V 8.5.996:

New variables:

- *Following error window* (index 6065)
- *Following error time out* (index 6066)
- *Digital inputs acc. to CiA402* (index 60FD)

Changes V 8.5.996 compared with V 8.5.995:

- The data type of *axis operating mode at startup* (index 5F35) is integer8 (before: integer16).
- The data type of *homing selection code* (Index 6098) is integer8 (before: integer16).
- The index of *acceleration time* is 6083 (before: 5EF4), the function is the same.
- The index of *deceleration time* is 6084 (before: 5EF5), the function is the same.
- The index of *quick stop time* is 6085 (before: 5EF6), the function is the same.
- *Homing velocity* (index 6099) is an array with two elements (before: simple variable).

New variables:

- *Inertia load* or *mass* (index 5E82)

Changes V 8.5.995 compared with V 8.5.96:

- The data type of the *motor rated torque* (index 6076) is unsigned32 (before: unsigned16).
- The data type of the *motor rated current* (index 6075) is unsigned32 (before: unsigned16).
- For variables *signal selection digital inputs*, *control word user defined bit*, *signal selection digital outputs*, and *status word user defined bit* new selection codes for controlling function timed positioning mode are available.

Changes V 8.5.96 compared with V 8.5.94:

New variable:

- *Motor temperature sensor data* (index 5F78)

Changes V 8.5.94 compared with V 8.5.91:

New variable:

- *Commutation reference* (index 5E83)

Changes V 8.5.91 compared with V 8.5.9:

New variable:

- *Encoder data* (index 5E81)

Changes V 8.5.9 compared with V 8.5.8.07:

New axis operating modes:

- Timed positioning mode
- Cyclic synchronous torque/force mode
- Cyclic synchronous velocity mode
- Cyclic synchronous position mode

Changed variables:

- The data type of *axis operating mode* (index 6060) is integer8 (before: integer16).
- The value of *position sensor resolution* (index 608F) can be defined freely in the first variable of the array (before: 65536 for using the motor position sensor or number of steps of the external position sensor after internal quadruplication). The second variable of the array (index 608F, subindex 2) must be 1 (before: number of revolutions).
- The unit for *maximum motor speed* (index 6080), *velocity factor numerator* (index 6094, subindex 1), and *velocity window* (index 606D) is 0.5 r.p.m. (before: 0.25 r.p.m.).

New variables:

- *Motor data* (index 5E80)

- *Electronic gearing control* (index 5F08)

Changes V 8.5.8.0.7 compared with V 8.5.8.04:

- Brake control: consideration of brake delay times
- New variable: *brake delay time*
- Changed selection codes for *release brake selection code*

Changes V 8.5.8.0.4 compared with V 8.5.8.03:

- Support for Ethernet (option F8), ESR, and Modbus TCP protocols
- Permits switching to operating modes profile position mode and electronic gearing in standstill, however, in state operation enabled.

Changes V 8.5.8.03 compared with V 8.5.8.02:

- Modbus protocol via ZF8 module

Changes V 8.5.8.02 compared with V 8.5.8.01:

- New functions for flying shear: input signal 178 (synchronize flying shear externally) and 179 (enable external synchronization)
- Jerk-free synchronization

Changes V 8.5.8.01 compared with V 8.5.8:

- Flying shear now for TrioDrive D/xS / MidiDrive D/xS

Changes V 8.5.8 compared with V 8.5.7:

- New functions: DC-bus coupling (16 A and higher), permanent switch-off of the shunt circuit can be configured, shunt circuit is controlled even if the PWM is off .
- The unit of the *velocity offset* (index 60B1) is VU.

Changes V 8.5.7 compared with V 8.5.5:

- New variable: *position encoder 2 value* (index 5004)

Changes V 8.5.5 compared with V 8.5.3:

- MidiDrive D/xS with rated current 16 A and 32 A is supported
- Operating hours counter, total time and power circuit
- New selection code for input enable: disable operation/operation enabled

Changes V 8.5.3 compared with V 8.5:

- Temperature sensors KTY 83 and KTY 84 for motor temperature measurement are supported
- New variables: *speed control loop Kv electronic gearing* (index 5f07), *external master axis selection code* (index 5f17), *gear ratio* (index 5f06), and *external position sensor resolution* (index 5f72)
- Options G1 and L1 are supported

Changes V 8.5 compared with V 8.0:

- New communication objects: *mains voltage* (index 5f4f) and *resistance motor temperature sensor* (index 5f03)
- Free PDO mapping
- Interpolated position mode available for CANopen
- Remanent variables (index 5f60, subindex 1 to 255)

Changes V 8.0 compared with V 7.7:

- Variables *digital inputs configuration*, *digital outputs configuration*, *position control loop Ki*, and *position control loop Ki range* removed.
- New variable: *I2t control*
- Function safe standstill for TrioDrive D/xS and MidiDrive D/xS

10.4 Appendix D Versions of the Document

2000-10-27	V 5.7, MH	for firmware V 5.7 first English version, based on German version; references to SPP for DOS removed
2000-11-13	V 5.7a, MH	Correction in type code; inputs/outputs I 8.x, O 8.x, O 9.x
2001-02-02	V 5.7b, MH	Corrections in assignment of axis control word and axis status word to digital I/O; OEM corrections
2008-10-28	V 8.0, KS	for firmware V 8.0 TrioDrive D/xS and MidiDrive D/xS added; DriveTerminal removed; completely new structure; text, technical specifications, as well as variables updated
2012-06-26	V 8.0e, KS	Modified as online help document, drive BN

		6755 added; Ethernet interface added; figures new, appendix List of Variables and Old Firmware Versions removed
2013-06-05	V 8.5, KS	for firmware V 8.5.9
		Appendix Firmware Versions Regarding Functions and Parameters added; section Homing Mode Sequences updated; operating mode **flying shear** updated; operating mode **torque mode** updated; variables changed and new variables added; sections Setting the Current Control Loop and Setting the Speed Control Loop new; new operating modes: **timed positioning mode**, **cyclic synchronous torque/ force mode**, **cyclic synchronous velocity mode**, **cyclic synchronous position mode**, and **interpolated position mode**; fault codes updated, BiSS encoder added

11 Keyword Index

In the keyword index, groups were set up to provide a better overview of certain keywords in addition to the direct entry in alphabetical order:

- Outputs

  Using the designation of an output, you can find out which functions are influencing it. Please, also see operating instructions “Connection and Commissioning” (6755.202, 6745.202, 6750.202, 6740.202, or 6710.202).

- Inputs

  Using the designation of an input, you can find out which functions the input acts on. Please, also see operating instructions “Connection and Commissioning” (6755.202, 6745.202, 6750.202, 6740.202, or 6710.202).

- LEDs

  The designation of a LED can be used for finding information on which states are indicated. Please, also see operating instructions “Connection and Commissioning” (6755.202, 6745.202, 6750.202, 6740.202, or 6710.202).

- Variables

  Using the designation of a variable, you can find the variable description of the corresponding variable.

- Variable short names

  Using the short name of a variable, you can find the variable description of the corresponding variable.

Further groups were set up for an illustration of the connections.

Index

- A -

- B -

- C -

- D -

- E -

- F -

- G -

- H -

- I -

- K -

- L -

- M -

- N -

- O -

- P -

- Q -

- R -

- S -

- T -

- U -

- V -

- Z -