

Drive Based Synchronous – SD6 Manual

en-US
05/2024
ID 443046.03

Table of contents

- Table of contents 2**
- 1 Foreword 5**
- 2 User information 6**
 - 2.1 Storage and transfer 6
 - 2.2 Described product 6
 - 2.3 Directives and standards 6
 - 2.4 Timeliness 6
 - 2.5 Original language 6
 - 2.6 Limitation of liability 6
 - 2.7 Formatting conventions 7
 - 2.7.1 Display of warning messages and information 7
 - 2.7.2 Markup of text elements 8
 - 2.7.3 Mathematics and formulas 8
 - 2.8 Trademarks 9
- 3 Safety notes 10**
- 4 What you should know before commissioning 11**
 - 4.1 DS6 program interface 11
 - 4.1.1 Configuring the view 13
 - 4.1.2 Navigation using sensitive circuit diagrams 14
 - 4.2 Meaning of parameters 15
 - 4.2.1 Parameter groups 15
 - 4.2.2 Parameter types and data types 16
 - 4.2.3 Parameter types 17
 - 4.2.4 Parameter structure 17
 - 4.2.5 Parameter visibility 18
 - 4.3 Signal sources and process data mapping 19
 - 4.4 Non-volatile memory 19
- 5 Commissioning 20**
 - 5.1 Initiating the project 21
 - 5.1.1 Projecting the drive controller and axis 21
 - 5.1.2 Configuring safety technology 22
 - 5.1.3 Creating other drive controllers and modules 22
 - 5.1.4 Projecting the module 23
 - 5.1.5 Projecting the project 23
 - 5.2 Mapping the mechanical axis model 24
 - 5.2.1 Parameterizing the motor 24
 - 5.2.2 Parameterizing the axis model 24

| | | |
|----------|---------------------------------------------------------------------|-----------|
| 5.3 | Referencing absolute position..... | 29 |
| 5.3.1 | Defining the referencing method..... | 29 |
| 5.3.2 | Parameterizing the reference switch..... | 29 |
| 5.3.3 | Setting the reference..... | 30 |
| 5.3.4 | Parameterizing the limit switch..... | 30 |
| 5.4 | Parameterizing Drive Based device control..... | 31 |
| 5.4.1 | Parameterizing transition conditions..... | 31 |
| 5.5 | Parameterizing synchronous operation..... | 32 |
| 5.5.1 | Projecting the IGB motion bus network..... | 32 |
| 5.5.2 | Parameterizing the master/slave assignment..... | 33 |
| 5.5.3 | Monitoring synchronous operation..... | 35 |
| 5.5.4 | Parameterizing the master axis..... | 36 |
| 5.5.5 | Parameterizing slave axes..... | 39 |
| 5.6 | Transmitting and saving a configuration..... | 40 |
| 5.6.1 | Transmitting the configuration..... | 40 |
| 5.6.2 | Saving the configuration..... | 42 |
| 5.7 | Testing the configuration..... | 43 |
| 5.7.1 | Testing jog mode..... | 43 |
| 5.7.2 | Testing the motion commands..... | 45 |
| 5.7.3 | Testing synchronous operation..... | 46 |
| 5.8 | Parameterizing the Drive Based Synchronous application..... | 51 |
| 5.8.1 | General motion variables and signal sources..... | 52 |
| 5.8.2 | Parameterizing motion block operating mode..... | 60 |
| 5.8.3 | Parameterizing command operating mode..... | 78 |
| 6 | More on Drive Based Synchronous? | 83 |
| 6.1 | Drive Based Synchronous – Concept..... | 83 |
| 6.1.1 | Operating modes..... | 84 |
| 6.1.2 | Jog mode..... | 85 |
| 6.1.3 | Control panels..... | 86 |
| 6.1.4 | Motion core..... | 86 |
| 6.1.5 | Data sources..... | 86 |
| 6.1.6 | Additional functions..... | 90 |
| 6.2 | Synchronous operation – Concept..... | 107 |
| 6.2.1 | Model 1: Virtual master or master encoder, multiple slave axes..... | 107 |
| 6.2.2 | Model 2: Master encoder, individual slave axis..... | 108 |
| 6.2.3 | Model 3: Master axis, one or more slave axes..... | 109 |
| 6.3 | IGB motion bus network..... | 110 |
| 6.4 | Axis model..... | 111 |
| 6.5 | Limit switches..... | 114 |
| 6.5.1 | Real axes..... | 114 |
| 6.5.2 | Virtual master..... | 119 |
| 6.5.3 | Special cases..... | 120 |

- 6.6 Referencing..... 122
 - 6.6.1 Referencing methods..... 122
 - 6.6.2 Reference position..... 144
 - 6.6.3 Reference retaining 144
- 6.7 Drive Based device control 145
 - 6.7.1 Drive Based device state machine 145
 - 6.7.2 States, transitions and conditions 146
- 6.8 Motion commands 150
- 6.9 Continue function..... 152
- 6.10 Electronic nameplate..... 153
- 6.11 Lag monitoring..... 154
- 7 Appendix..... 155**
 - 7.1 Standard mapping of Drive Based Synchronous..... 155
 - 7.1.1 SD6: Drive Based Synchronous receive process data 156
 - 7.1.2 SD6: Drive Based Synchronous transmit process data 156
 - 7.2 Detailed information 157
 - 7.3 Abbreviations 158
- 8 Contact 159**
 - 8.1 Consultation, service and address..... 159
 - 8.2 Your opinion is important to us..... 159
 - 8.3 Close to customers around the world 160
- List of figures..... 161**
- List of tables..... 162**
- Glossary 163**

1 Foreword

The DriveControlSuite commissioning software for 6th generation drive controllers offers convenient functions for efficient project configuration and commissioning of drive controllers in multi-axis and single-axis applications.

The Drive Based Synchronous application contained in the software provides universal solutions for drive-based motion control using position, velocity and torque/force control types. The motion commands associated with these control types are categorized into the following operating modes:

- Command
- Motion block

The Drive Based Synchronous application enables you to operate axes in synchronous operation, e.g. to use a virtual master in order to move several axes on a platform equally or to use a master encoder in order to synchronize axes with an upstream machine, such as when using conveyor belts.

Additional functions for monitoring process variables such as positions or velocities offer added convenience in monitoring.

This documentation describes the general functions of the Drive Based Synchronous application and guides you step by step through the setup and project configuration of your drive project in the individual operating modes.

2 User information

This documentation supports you in the setup and project configuration of your drive system with the Drive Based Synchronous application in conjunction with Drive Based device control.

Technical knowledge

To be able to commission one or more drive controllers using the Drive Based Synchronous application, possibly in combination with a controller, you should have basic knowledge of handling 6th generation STOBER drive controllers and of the DriveControlSuite commissioning software.

Technical requirements

Before you begin operating your drive system, you need to have wired the drive controllers involved and initially check that they are functioning correctly. To do this, follow the instructions in the manual for the relevant drive controller.

2.1 Storage and transfer

As this documentation contains important information for handling the product safely and efficiently, it must be stored in the immediate vicinity of the product until product disposal and be accessible to qualified personnel at all times.

Also pass on this documentation if the product is transferred or sold to a third party.

2.2 Described product

This documentation is binding for:

SD6 series drive controllers in conjunction with the DriveControlSuite software (DS6) in V 6.5-L or later and associated firmware in V 6.5-L or later.

2.3 Directives and standards

Refer to the drive controller documentation for the European directives and standards relevant to the drive controller and accessories.

2.4 Timeliness

Check whether this document is the latest version of the documentation. We make the latest document versions for our products available for download on our website:

<http://www.stoeber.de/en/downloads/>.

2.5 Original language

The original language of this documentation is German; all other language versions are derived from the original language.

2.6 Limitation of liability

This documentation was created taking into account the applicable standards and regulations as well as the current state of technology.

No warranty or liability claims for damage shall result from failure to comply with the documentation or from use that deviates from the intended use of the product. This is especially true for damage caused by individual technical modifications to the product or the project configuration and operation of the product by unqualified personnel.

2.7 Formatting conventions

Orientation guides in the form of signal words, symbols and special text markups are used to emphasize specific information so that you are able identify it in this documentation quickly.

2.7.1 Display of warning messages and information

Warning messages are identified with symbols. They indicate special risks when handling the product and are accompanied by relevant signal words that express the extent of the risk. Furthermore, useful tips and recommendations for efficient, error-free operation are specially highlighted.

ATTENTION!

Attention

This indicates that damage to property may occur

- if the stated precautionary measures are not taken.

⚠ CAUTION!

Caution

This word with a warning triangle indicates that minor personal injury may occur

- if the stated precautionary measures are not taken.

⚠ WARNING!

Warning

This word with a warning triangle means there may be a considerable risk of fatal injury

- if the stated precautionary measures are not taken.

⚠ DANGER!

Danger

This word with a warning triangle indicates that there is a considerable risk of fatal injury

- if the stated precautionary measures are not taken.

Information

Information indicates important information about the product or serves to emphasize a section in the documentation that deserves special attention from the reader.

2.7.2 Markup of text elements

Certain elements of the continuous text are distinguished as follows.

| | |
|-------------------------------------------------------------------|----------------------------------------------|
| Important information | Words or expressions with a special meaning |
| Interpolated position mode | Optional: File or product name or other name |
| <u>Detailed information</u> | Internal cross-reference |
| http://www.samplelink.com | External cross-reference |

Software and other displays

The following formatting is used to identify the various information content of elements referenced by the software interface or a drive controller display, as well as any user entries.

| | |
|--------------------------------|---------------------------------------------------------------------------------------------------------------------|
| Main menu Settings | Window names, dialog box names, page names or buttons, combined proper nouns, functions referenced by the interface |
| Select Referencing method A | Predefined entry |
| Save your <own IP address> | User-defined entry |
| EVENT 52: COMMUNICATION | Displays (status, messages, warnings, faults) |

Keyboard shortcuts and command sequences or paths are represented as follows.

| | |
|----------------------|---------------------------------------------------|
| [Ctrl], [Ctrl] + [S] | Key, key combination |
| Table > Insert table | Navigation to menus/submenus (path specification) |

2.7.3 Mathematics and formulas

The following signs are used to represent mathematical relationships and formulas.

| | |
|---|----------------|
| – | Subtraction |
| + | Addition |
| × | Multiplication |
| ÷ | Division |
| | Absolute value |

2.8 Trademarks

The following names used in connection with the device, its optional equipment and its accessories are trademarks or registered trademarks of other companies:

| | |
|----------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CANopen [®] , CiA [®] | CANopen [®] and CiA [®] are registered European Union trademarks of CAN in AUTOMATION e.V., Nuremberg, Germany. |
| EnDat [®] | EnDat [®] and the EnDat [®] logo are registered trademarks of Dr. Johannes Heidenhain GmbH, Traunreut, Germany. |
| EtherCAT [®] , Safety over EtherCAT [®] , TwinCAT [®] | EtherCAT [®] , Safety over EtherCAT [®] and TwinCAT [®] are registered trademarks of patented technologies licensed by Beckhoff Automation GmbH, Verl, Germany. |
| PLCopen [®] | PLCopen [®] is a registered trademark of the PLCopen Organisation, Gorinchem, Netherlands. |
| PROFIBUS [®] , PROFINET [®] | PROFIBUS [®] and PROFINET [®] are registered trademarks of PROFIBUS Nutzerorganisation e.V., Karlsruhe, Germany. |

All other trademarks not listed here are the property of their respective owners.

Products that are registered as trademarks are not specially indicated in this documentation. Existing property rights (patents, trademarks, protection of utility models) are to be observed.

3 Safety notes

WARNING!

Risk of fatal injury if safety notes and residual risks are not observed!

Failure to observe the safety notes and residual risks in the drive controller documentation may result in accidents causing serious injury or death.

- Observe the safety notes in the drive controller documentation.
 - Consider the residual risks in the risk assessment for the machine or system.
-

WARNING!

Malfunction of the machine due to incorrect or modified parameterization!

In the event of incorrect or modified parameterization, malfunctions can occur on machines or systems which can lead to serious injuries or death.

- Observe the security notes in the drive controller documentation.
 - Protect the parameterization, e.g. from unauthorized access.
 - Take appropriate measures for possible malfunctions (e.g. emergency off or emergency stop).
-

4 What you should know before commissioning

The following chapters provide a quick introduction to the structure of the program interface and accompanying window designations as well as relevant information about parameters and generally saving your project configuration.

4.1 DS6 program interface

Using the graphical interface of the DriveControlSuite commissioning software (DS6), you can project, parameterize and commission your drive project quickly and efficiently. In case of service, you can evaluate diagnostic information such as operating states, fault memories and fault counters of your drive project using DriveControlSuite.

Information

The program interface of DriveControlSuite is available in German, English and French. To change the language of the program interface, select **Settings > Language**.

Information

The DriveControlSuite help in the menu bar can be reached via **Help > Help for DS6** or via the [F1] key on your keyboard. When you press [F1] in an area of the program, the corresponding help topic opens.

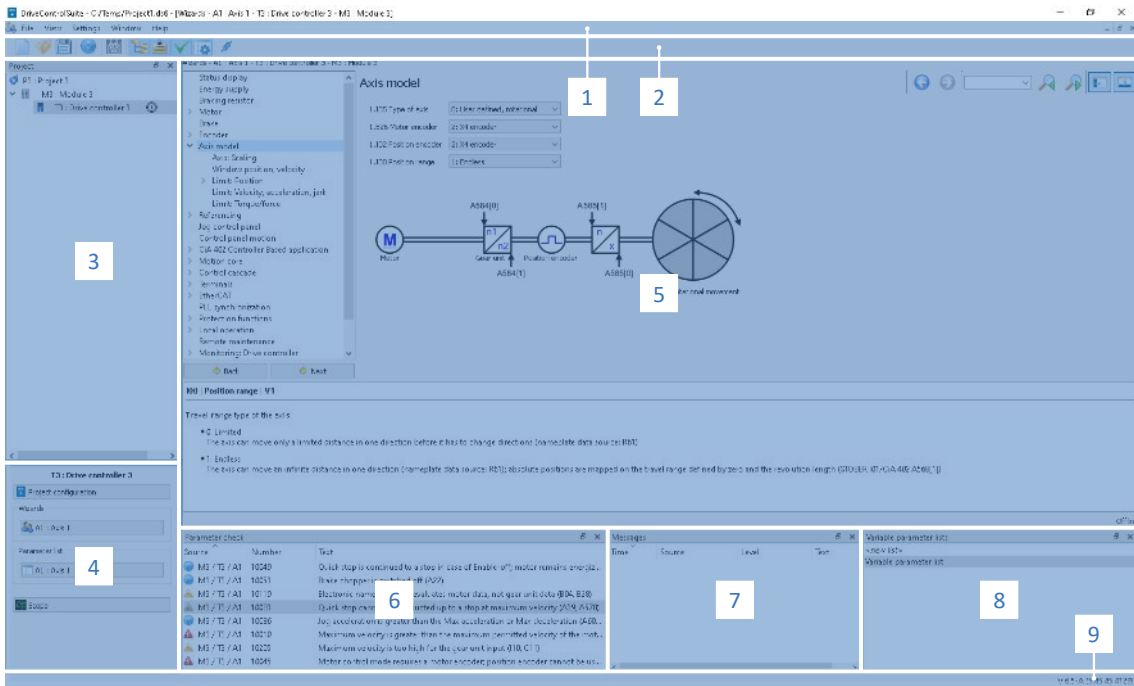


Fig. 1: DS6: Program interface





| No. | Area | Description |
|-----|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | Menu bar | Using the File, View, Settings and Window menus, you can open and save projects, display and hide program windows, select the interface language and access level and change between different windows in the workspace. |
| 2 | Toolbar | The toolbar enables quick access to frequently needed functions, like opening and saving projects and hiding and displaying windows in the program interface. |
| 3 | Project tree | The project tree forms the structure of your drive project in the form of modules and drive controllers. Select an element using the project tree first in order to edit it using the project menu. |
| 4 | Project menu | The project menu offers you various functions for editing the project, module and drive controller. The project menu adapts to the element that you selected in the project tree. |
| 5 | Workspace | The different windows which can be used to edit your drive project, such as the configuration dialog, wizards, the parameter list or the scope analysis tool, open in the workspace. |
| 6 | Parameter check | The parameter check points out irregularities and inconsistencies that were detected in the plausibility check of calculable parameters. |
| 7 | Messages | The entries in the messages log the connection and communication status of the drive controllers, incorrect inputs caught by the system, errors when opening a project or rule violations in the graphical programming. |
| 8 | Variable parameter lists | You can use variable parameter lists to compile any parameters in individual parameter lists for a quick overview. |
| 9 | Status bar | In the status bar, you can find the specifications of the software version and get additional information about the project file, the devices and the progress of the process during processes such as loading projects. |

4.1.1 Configuring the view

In DriveControlSuite, you can change the visibility and arrangement of areas and windows, such as to optimize the available space in the workspace when working with smaller screens.

Showing/hiding areas

Use the icons in the toolbar or the items in the View menu to show or hide specific areas in DriveControlSuite as needed.

| Icon | Item | Description |
|-----------------------------------------------------------------------------------|--------------------------|--------------------------------------------------------------|
| – | Reset | Resets the view to factory settings. |
|  | Project | Shows/hides the Project window (project tree, project menu). |
|  | Messages | Shows/hides the Messages window. |
|  | Parameter check | Shows/hides the Parameter check window. |
|  | Variable parameter lists | Shows/hides the Variable parameter lists window. |

Arrange and group areas

You can undock and rearrange the individual areas via drag and drop. If you drag an undocked window to the edge of DriveControlSuite, you can release it there in a color-highlighted area either next to or on top of another window to redock it.

When you release the window onto another window, the two areas are merged into one window where you can use tabs to switch between the areas.

4.1.2 Navigation using sensitive circuit diagrams

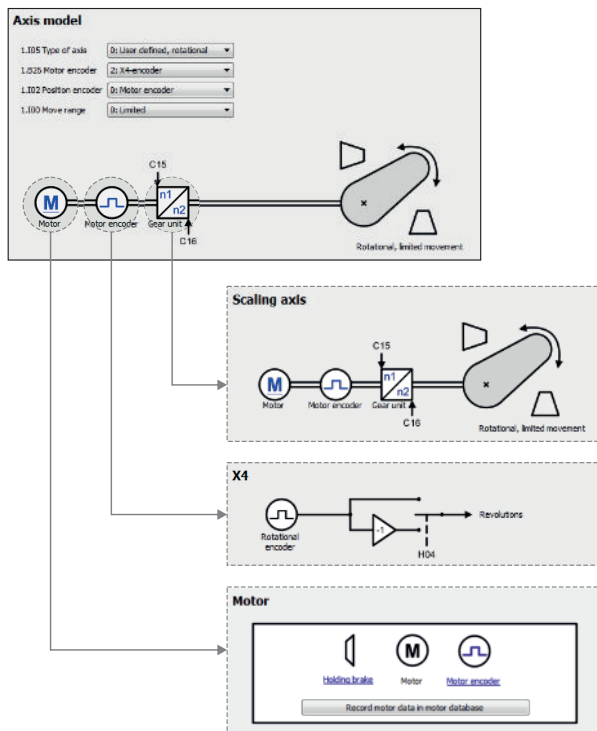


Fig. 2: DriveControlSuite: Navigation using text links and symbols

In order to illustrate graphically the processing sequence of actual and set values, the use of signals or certain drive component arrangements and to make configuring the accompanying parameters easier, they are displayed on the respective wizard pages of the workspace in the form of circuit diagrams.

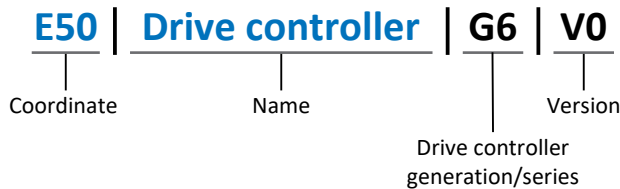
Blue text links or clickable icons indicate links within the program. These refer to the corresponding wizard pages and, as a result, allow you to reach additional helpful detail pages with just a click.

4.2 Meaning of parameters

You can use parameters to adapt the function of the drive controller to your individual application. In addition, parameters visualize the current actual values (actual velocity, actual torque, etc.) and trigger actions such as Save values, Test phase, etc.

Interpretation of parameter identification

Parameter identification consists of the following elements, where short forms are also possible, i.e. only specifying a coordinate or the combination of coordinate and name.



4.2.1 Parameter groups

Parameters are assigned to individual groups by topic. The 6th generation of drive controllers differentiates between the following parameter groups.

| Group | Topic |
|-------|----------------------------------------------------------------------------------------------------|
| A | Drive controllers, communication, cycle times |
| B | Motor |
| C | Machine, velocity, torque/force, comparators |
| D | Set value |
| E | Display |
| F | Terminals, analog and digital inputs and outputs, brake |
| G | Technology – Part 1 (application-dependent) |
| H | Encoder |
| I | Motion (all motion settings) |
| J | Motion blocks |
| K | Control panel |
| L | Technology – Part 2 (application-dependent) |
| M | Profiles (application-dependent) |
| N | Additional functions (application-dependent; e.g. extended cam control unit) |
| P | Customer-specific parameters (programming) |
| Q | Customer-specific parameters, instance-dependent (programming) |
| R | Production data for the drive controller, motor, brakes, motor adapter, gear unit and geared motor |
| S | Safety (safety technology) |
| T | Scope |
| U | Protection functions |
| Z | Fault counter |

Tab. 1: Parameter groups

4.2.2 Parameter types and data types

In addition to topic-based sorting in individual groups, all parameters belong to a certain data type and parameter type. The data type of a parameter is displayed in the parameter list, properties table. The connections between parameter types, data types and their value range can be found in the following table.

| Data type | Parameter type | Length | Value range (decimal) |
|---------------------------------------------|------------------------------------|-----------------------------------|---------------------------------------------------------|
| INT8 | Integer or selection | 1 byte (signed) | -128 – 127 |
| INT16 | Integer | 2 bytes (1 word, signed) | -32768 – 32767 |
| INT32 | Integer or position | 4 bytes (1 double word, signed) | -2 147 483 648 – 2 147 483 647 |
| BOOL | Binary number | 1 bit (internal: LSB in 1 byte) | 0, 1 |
| BYTE | Binary number | 1 byte (unsigned) | 0 – 255 |
| WORD | Binary number | 2 bytes (1 word, unsigned) | 0 – 65535 |
| DWORD | Binary number or parameter address | 4 bytes (1 double word, unsigned) | 0 – 4 294 967 295 |
| REAL32 (single type according to IEE754) | Floating-point number | 4 bytes (1 double word, signed) | -3.40282×10^{38} – 3.40282×10^{38} |
| STR8 | Text | 8 characters | — |
| STR16 | Text | 16 characters | — |
| STR80 | Text | 80 characters | — |

Tab. 2: Parameters: data types, parameter types, possible values

Parameter types: Use

- Integer, floating-point number
For general computing processes
Example: Set and actual values
- Selection
Numeric value to which a direct meaning is assigned
Example: Sources for signals or set values
- Binary number
Bit-oriented parameter information that is collected in binary
Example: Control and status words
- Position
Integer combined with associated units and decimal places
Example: Actual and set values of positions
- Velocity, acceleration, deceleration, jerk
Floating-point number combined with associated units
Example: Actual and set values for velocity, acceleration, deceleration, jerk
- Parameter address
Referencing of a parameter
Example: In F40 AO1 source, for example, E08 n-motor filtered can be parameterized
- Text
Outputs or messages

4.2.3 Parameter types

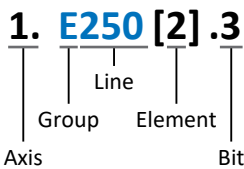
The following types of parameters are differentiated.

| Parameter type | Description | Example |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Simple parameters | Consist of one group and one line with a defined value. | A21 Brake resistor R: Value = 100 ohms |
| Array parameters | Consist of a group, a line and multiple sequential (listed) elements, which have the same properties but different values. | A10 Access level <ul style="list-style-type: none"> ▪ A10[0] access level: Value = Access level via operating unit ▪ A10[2] access level: Value = Access level via CANopen and EtherCAT ▪ A10[4] access level: Value = Access level via PROFINET |
| Record parameters | Consist of a group, a line and multiple sequential (listed) elements, which can have different properties and different values. | A00 Save values <ul style="list-style-type: none"> ▪ A00[0] Start: Value = Start action ▪ A00[1] Progress: Value = Display action progress ▪ A00[2] Result: Value = Display action result |

Tab. 3: Parameter types

4.2.4 Parameter structure

Every parameter has specific coordinates with the following structure.



- Axis (optional)
In case of multiple axes, the one to which a parameter is assigned; not applicable for global parameters (value range: 1 – 4).
- Group
The thematic group to which a parameter belongs (value range: A – Z).
- Line
Distinguishes the parameters within a parameter group (value range: 0 – 999).
- Element (optional)
Elements of an array or record parameter (value range: 0 – 16000).
- Bit (optional)
Selection of a single bit for complete data addressing; depends on the data type (value range: 0 – 31).

4.2.5 Parameter visibility

The visibility of a parameter is primarily controlled by the access level you set in DriveControlSuite and by the properties you project for the respective drive controller (e.g. hardware, firmware and application). A parameter can also be shown or hidden depending on other parameters or settings. For example, the parameters of an additional function are only shown as soon as you activate the relevant additional function.

Access level

The access options for the individual software parameters are ranked hierarchically and divided into individual levels. This means that parameters can be hidden for a specific purpose and, relatedly, their configuration options can be locked starting from a specific level.

Each parameter has one access level for read access (visibility) and one access level for write access (editability). The following levels are present:

- Level 0
Elementary parameters
- Level 1
Important parameters of an application
- Level 2
Important parameters for service with extensive diagnostic options
- Level 3
All parameters needed for commissioning and optimizing an application

The parameter A10 Access level controls general access to parameters:

- Over the drive controller display (A10[0])
- Over CANopen or EtherCAT (A10[2])
- Over PROFINET (A10[3])

Information

It is not possible to write to or read the parameter hidden in DriveControlSuite during communication via fieldbus.

Hardware

Which parameters are available to you in DriveControlSuite is determined by which series you select in the configuration dialog for the drive controller, for example, or whether you project an option module. Basically, the only parameters that are displayed are the ones you need to parameterize the configured hardware.

For example, a drive controller can evaluate an encoder using terminal X120, provided that a corresponding terminal module has been installed. The accompanying evaluation is activated using parameter H120. However, this parameter is visible only if the terminal module was initially selected during the drive project configuration.

Firmware

Due to the further development and updating of functions for the 6th generation of drive controllers, new parameters and also new versions of existing parameters are continuously being implemented in DriveControlSuite and in the firmware. The parameters are displayed in the software according to the DriveControlSuite version used and the configured firmware version of the respective drive controller.

Applications

Applications generally differ in terms of functions and their control. For this reason, different parameters are available with each application.

4.3 Signal sources and process data mapping

The transmission of control signals and set values in DriveControlSuite meets the following principles.

Signal sources

Drive controllers are controlled either over a fieldbus, using mixed operation consisting of a fieldbus system and terminals or exclusively using terminals.

You can use the corresponding selection parameters, referred to as signal sources, to configure whether the control signals and set values of the application are obtained over a fieldbus or using terminals.

In case of activation over a fieldbus, parameters that are selected as data sources for control signals or set values must be part of the subsequent process data mapping. In the case of activation using terminals, the respective analog or digital inputs are specified directly.

Process data mapping

If you are working with a fieldbus system and have selected the source parameters for control signals and set values, configure the fieldbus-specific settings, e.g. the assignment of the process data channels for transmitting receive and transmit process data, as the last step. The respective procedure can be found in the accompanying STOBER fieldbus manuals.

4.4 Non-volatile memory

All project configurations, parameterizations and related changes to parameter values are in effect after transmission to the drive controller, but are only stored in volatile memory.

Saving to a drive controller

To save the configuration in non-volatile memory on a drive controller, you have the following options:

- Saving the configuration using the *Save values wizard*:
Project menu > Wizards area > Projected axis > Save values wizard: Select the Save values action
- Saving the configuration using the parameter list:
Project menu > Parameter list area > Projected axis > Group A: Drive controller > A00 Save values: Set the parameter A00[0] to the value 1: Active
- Saving the configuration using the operating unit:
Drive controller with operating unit: Press the save button for 3 seconds

Saving to all drive controllers within a project

To save the configuration in non-volatile memory on several drive controllers, you have the following options:

- Saving the configuration using the toolbar:
Toolbar > Save values icon: Click the Save values icon
- Saving the configuration using the *Online functions window*:
Project menu > Online connection button > Online functions window: Click on Save values (A00)

Information

Do not shut off the drive controller while saving. If the supply voltage to the control unit is interrupted while saving, the drive controller will start without an executable configuration the next time it is switched on. In this case, the configuration must be transferred to the drive controller again and stored in non-volatile memory.

5 Commissioning

The following chapters describe how to commission your drive system using DriveControlSuite.

For the components of your axis model, **as an example**, we require a synchronous servo motor with EnDat 2.1/2.2 digital encoder and optional brake. These motors are saved along with all relevant data for the project configuration in the DriveControlSuite motor database and in the electronic nameplate.

Upon selecting the motor from the database, such as when reading out the nameplate, all data is transferred to the corresponding parameters. There is no need for complex parameterization of the motor, brake or encoder.

For asynchronous motors, the motor data relevant for project configuration is also taken from the motor database. All other motor types need to have their parameters configured manually.

Before you begin commissioning, make sure that the system nodes are hooked up and supplied with control voltage.

Information

Always perform the steps described below in the specified order!

Some parameters are interdependent and do not become accessible to you until you have first configured certain settings. Follow the steps in the specified sequence so that you can finish the parameterization completely.

5.1 Initiating the project

In order to be able to configure all drive controllers and axes of your drive system using DriveControlSuite, you must record them as part of a project.

5.1.1 Projecting the drive controller and axis

Create a new project and project the first drive controller along with the accompanying axis.

Information

Make sure that you project the correct series in the *Drive controller* tab. The projected series cannot be changed afterwards.

Creating a new project

1. Start DriveControlSuite.
2. On the start screen, click *Create new project*.
 - ⇒ The new project is created and the configuration dialog for the first drive controller opens.
 - ⇒ The *Drive controller* button is active.

Projecting the drive controller

1. **Properties tab:**

Establish the relationship between your circuit diagram and the drive controller to be projected in DriveControlSuite.
Reference: Specify the reference code (equipment code) of the drive controller.
Designation: Give the drive controller a unique name.
Version: Version your project configuration.
Description: If necessary, specify additional supporting information, such as the change history of the project configuration.
2. **Drive controller tab:**

Select the series, device type and firmware version of the drive controller.
3. **Option modules tab:**

Depending on the drive controller series, various option modules are available for you to select.
Communication module: If the drive controller communicates with a controller via a fieldbus, select the corresponding communication module.
Terminal module: If you are controlling the drive controller using analog and digital inputs, select the corresponding terminal module (in addition to the communication module in mixed operation).
Safety module: If the drive controller is part of a safety circuit, select the corresponding safety module.
4. **Device control tab:**

Device control: Select *Drive Based*.
Receive process data (Rx), transmit process data (Tx): If you control the drive controller using a fieldbus, select the fieldbus-specific receive and transmit process data.

Projecting the axis

1. Click on Axis 1.
 2. **Properties tab:**
Establish the connection between your circuit diagram and the axis to be projected in DriveControlSuite.
Reference: Specify the reference code (equipment code) of the axis.
Designation: Give the axis a unique name.
Version: Version your project configuration.
Description: If necessary, specify additional supporting information, such as the change history of the project configuration.
 3. **Application tab:**
Select Drive Based Synchronous.
 4. **Motor tab:**
Select the type of motor operated using this axis.
 5. Confirm with OK.
- ⇒ The configuration dialog closes and the drive controller is created in a new module in the project tree.

5.1.2 Configuring safety technology

If the drive controller is part of a safety circuit, you must configure the safety technology in accordance with the commissioning steps outlined in the corresponding manual in the next step (see Detailed information).

5.1.3 Creating other drive controllers and modules

In DriveControlSuite, all drive controllers within a project are grouped using modules. If you add a new drive controller to your project, be sure to always assign it to an existing module. Group drive controllers in a module if, for example, they are located in the same control cabinet or work together to operate the same machine part.

Creating a drive controller

1. In the project tree, select your project P1 > module M1 > context menu **Create new drive controller**.
⇒ The drive controller is created in the project tree and the configuration dialog opens.
2. Project the drive controller as described in [Projecting the drive controller and axis \[► 21\]](#).
3. Repeat the steps for all other drive controllers that you want to project.

Creating a module

1. In the project tree, select your project P1 > context menu **Create new module**.
⇒ The module is created in the project tree.
2. Project the module as described in [Projecting the module \[► 23\]](#).
3. Repeat the steps for all other modules that you want to project.

5.1.4 Projecting the module

Give your module a unique name, enter the reference code and, as an option, store additional information like the version and change history of the module.

1. Select the module in the project tree and click on **Project configuration** in the project menu.
⇒ The configuration dialog for the module opens.
2. Establish the relationship between your circuit diagram and the module in DriveControlSuite.
Reference: Specify the reference code (equipment code) of the module.
Designation: Give the module a unique name.
Version: Version the module.
Description: If necessary, specify additional supporting information, such as the change history of the module.
3. Confirm with OK.

5.1.5 Projecting the project

Give your project a unique name, enter the reference code and, as an option, store additional information like the version and change history of the project.

1. Mark the project in the project tree and click on **Project configuration** in the project menu.
⇒ The configuration dialog for the project opens.
2. Establish the relationship between your circuit diagram and the project in DriveControlSuite.
Reference: Specify the reference code (equipment code) of the project.
Designation: Give the project a unique name.
Version: Version the project.
Description: If necessary, specify additional supporting information, such as the change history of the project.
3. Confirm with OK.

5.2 Mapping the mechanical axis model

To be able to put your real drive train with one or more drive controllers into operation, you must map your complete mechanical environment in DriveControlSuite.

The 6th generation of STOBER drive controllers are specially developed for communication between the drive controller and controller on the basis of the actual variables at the output (° or mm of real axis movement). The scaling of the axis model is calculated by the drive controller's firmware independently of the encoder model, without any rounding errors or drift.

5.2.1 Parameterizing the motor

You have projected a synchronous servo motor with EnDat 2.1/2.2 digital encoder and optional brake.

By projecting the corresponding motor, limiting values for currents and torques as well as associated temperature data are automatically transferred to the respective parameters of the individual wizards. All additional data on the brake and encoder is transferred at the same time.

Motor protection

All models of the 6th drive controller generation feature an i^2t model – a computational model for thermal monitoring of the motor. To activate it and set up the protective function, configure the following settings (deviating from the presets): U10 = 2: Warning and U11 = 1.00 s. This model can be used instead of or in addition to temperature-monitored motor protection.

5.2.2 Parameterizing the axis model

Parameterize the setup of your drive in this order:

- Define the axis model
- Scale the axis
- Parameterize the position and velocity window
- Limit the axis (optional)
 - Limit the position
 - Limit the velocity, acceleration and jerk
 - Limit the torque and force

5.2.2.1 Define the axis model

1. Select the relevant drive controller in the project tree and click on the desired projected axis in the Project menu > Wizard area.
2. Select the Axis model wizard.
3. I05 Type of axis:
Define whether the axis type is rotational or translational.
 - 3.1. If you would like to configure the units of measure and the number of decimal places individually for specifying and displaying positions, velocities, accelerations and jerk, select 0: User defined, rotational or 1: User defined, translational.
 - 3.2. If the units of measure and the number of decimal places for specifying and displaying positions, velocities, accelerations and jerk are to be fixed, select 2: Rotational or 3: Translational.
4. B26 Motor encoder:
Define the interface to which the motor encoder is connected.
5. I02 Position encoder (optional):
Define the interface to which the position encoder is connected.
6. I00 Position range:
Define whether the travel range of the axis is limited or endless (modulo).
7. If you select I00 = 1: Endless, parameterize a revolution length when you scale the axis.

Information

When you parameterize I05 Type of axis, you can either use selection 0: User defined, rotational or 1: User defined, translational to configure units of measure and the number of decimal places for the axis model individually or use selections 2: Rotational and 3: Translational to revert to preset values.

Selection 0: User defined, rotational and selection 1: User defined, translational let you configure the unit of measure (I09) and the decimal places (I06) individually. Velocity, acceleration and jerk are represented as the derivative of the unit of measure with respect to time.

Selection 2: Rotational sets the following units of measure for the axis model: position in $^{\circ}$, velocity in rpm, acceleration in rad/s^2 , jerk in rad/s^3 .

Selection 3: Translational sets the following units of measure for the axis model: position in mm, velocity in m/s, acceleration in m/s^2 , jerk in m/s^3 .

Information

If you do not parameterize it differently for I02 Position encoder, B26 Motor encoder is used for position control as standard.

5.2.2.2 Scale the axis

1. Select the relevant drive controller in the project tree and click on the desired projected axis in the Project menu > Wizard area.
2. Select the Axis model wizard > Axis: Scaling.
3. Scale the axis by configuring the overall gear ratio between the motor and output.
To simplify this scaling for you, you are provided with the scaling calculator Conversion of positions, velocities, accelerations, torque/force, which calculates the effects of changed motion variables on the entire system.
4. I01 Circular length:
If you have selected 1: Endless for I00 Position range, enter the revolution length.
5. I06 Decimal places position (optional):
If you have selected 0: User defined, rotational or 1: User defined, translational for I05 Type of axis, define the desired number of decimal places.
6. I09 Measure unit (optional):
If you have selected 0: User defined, rotational or 1: User defined, translational for I05 Type of axis, define the desired unit of measure.
7. I03 Axis polarity:
Use the polarity to specify the direction of interpretation between the axis movement and motor movement.

| |
|--------------------|
| Information |
|--------------------|

A change to parameter I06 moves the decimal separator for all axis-specific position values! Ideally, define I06 before parameterizing other position values and then check them.

If the axis receives set value specifications from a controller or follows the master values of a master, the resolution of position values directly impacts the smooth operation of the axis. Therefore, you should define a sufficient number of decimal places appropriate for your application.

| |
|--------------------|
| Information |
|--------------------|

Parameter I297 Maximum speed position encoder must be parameterized according to your application case. If I297 is set too low, the permitted maximum speed is exceeded even at normal operating speeds. On the other hand, if I297 is set too high, measuring errors of the encoder can be overlooked.

I297 depends on the following parameters: I05 Type of axis, I06 Decimal places position, I09 Measure unit as well as I07 Distance factor numerator position and I08 Distance factor denominator position or A585 Feed constant for CiA 402. If you have made changes to one of the parameters listed, select I297 accordingly as well.

5.2.2.3 Parameterizing the position and velocity window

Enter position limits and velocity zones for set values. To do so, parameterize boundary values for reaching a position or velocity.

1. Select the **Axis model wizard** > **Window position, velocity**.
2. **C40 Velocity window:**
Parameterize a tolerance window for velocity tests.
3. **I22 Target window:**
Parameterize a tolerance window for position tests.
4. **I87 Actual position in window time:**
Parameterize how long a drive must stay in the specified position window before a corresponding status message is output.
5. **I21 Maximal following error:**
Parameterize a tolerance window for lag tests.

For more information on monitoring the lag, see [Lag monitoring \[► 154\]](#).

5.2.2.4 Limiting the axis

If necessary, limit the movement variables for position, velocity, acceleration, jerk as well as torque/force according to the applicable conditions for your axis model.

Limiting the position (optional)

1. Select the relevant drive controller in the project tree and click on the desired projected axis in the Project menu > Wizard area.
2. Select the Axis model wizard > Limit: Position.
3. If necessary, limit the position of your axis using a software or hardware limit switch to secure the travel range.

Limiting velocity, acceleration, jerk (optional)

The default values are designed for slow velocities without gear units. For this reason, adapt the saved values.

For example, verify the maximum velocity of the motor (B83) against the velocity of the output (I10).

1. Select the Motor wizard.
2. Determine the maximum possible motor velocity in parameter B83 v-max motor.
3. Select the Axis model wizard > Axis: Scaling > Conversion of positions, velocities, accelerations, torque/force area.
4. Velocity line:
Enter the maximum motor velocity from B83 in the Velocity line of the Motor column and confirm with ENTER.
⇒ The maximum velocity of the motor is converted to the output.
5. Repeat the procedure for other limits, such as for the gear unit input speed (C11).
6. Select the Axis model wizard > Limit: Velocity, acceleration, jerk.
7. I10 Maximal speed:
Limit the maximum velocity of the output, taking into account the determined system limits.
8. Determine the limiting values for acceleration and jerk if necessary and enter them into the associated parameters.

Limiting torque/force (optional)

The default values take into account the rated operation together with the overload reserves.

1. Select the Axis model wizard > Limit: Torque/force.
2. If the motor force must be limited, adapt the saved values as necessary.

5.3 Referencing absolute position

When commissioning a system with position measurement systems, the relation of a measured to an actual axis position must be determined to be able to work with absolute positions.

If you are working with absolute positions, reference the axis now. Otherwise, proceed by parameterizing application-specific conditions and responses of the Drive Based device control.

WARNING!

Injury to persons and material damage due to slave axis movement!

By activating this function, the master value changes immediately without a motion profile. As a result, coupled slave axes immediately follow the movement of the master.

- Only execute this function if there are no slave axes coupled with the master.

5.3.1 Defining the referencing method

1. Select the relevant drive controller in the project tree and click on the desired projected axis in the Project menu > Wizard area.
2. Select the Referencing wizard.
3. I30 Referencing type, I31 Referencing direction, I35 Referencing with zero pulse:
Define the characteristics for referencing.
For information on possible combinations of the named characteristics as well as detailed descriptions of the individual referencing methods, see [Referencing methods \[▶ 122\]](#).
4. I43 Move to reference position:
If the axis is to automatically move to the referenced position after referencing, set this parameter to 1: Active.

5.3.2 Parameterizing the reference switch

If you are working with absolute positions and would like to determine a reference position while referencing using the reference switch, proceed as follows.

1. Select the Drive Based Synchronous application wizard > Data sources > Application digital signals: Data source.
2. I103 Source reference switch:
 - 2.1. If a digital input (direct or inverted) acts as the source, select the corresponding input.
 - 2.2. If bit 3 of the control word I210 of the application is the source, select 2: Parameter.

5.3.3 Setting the reference

If you are working with absolute positions and would like to reference without using a referencing run, the value of the current actual position is applied directly as a parameter value from I34 using the Set reference signal.

1. Select the Drive Based Synchronous application wizard > Data sources > Application digital signals: Data source.
2. I111 Source set reference:
 - 2.1. If a digital input (direct or inverted) acts as the source, select the corresponding input.
 - 2.2. If bit 11 of the control word I210 of the Drive Based Synchronous application is the signal source, select 2: Parameter.

5.3.4 Parameterizing the limit switch

Using the limit switch, you can limit the travel range in the positive and negative direction using sensor signals, regardless of the operating mode. Both signals are protected against wire breaks (= low active, i.e. a low level leads to a limit switch fault).

Parameterizing a positive limit switch

1. Select the Drive Based Synchronous application wizard > Data sources > Application digital signals: Data source.
2. I101 Source positive /limit switch:

To limit the travel range in the positive direction of motion, select the source for the positive limit switch.

 - 2.1. If a digital input (direct or inverted) acts as the source, select the corresponding input.
 - 2.2. If bit 1 of the control word I210 of the application is the source, select 2: Parameter.

Parameterizing a negative limit switch

1. Select the Drive Based Synchronous application wizard > Data sources > Application digital signals: Data source.
2. I102 Source negative /limit switch:

To limit the travel range in the positive direction of motion, select the source for the negative limit switch.

 - 2.1. If a digital input (direct or inverted) acts as the source, select the corresponding input.
 - 2.2. If bit 2 of the control word I210 of the application is the source, select 2: Parameter.

5.4 Parameterizing Drive Based device control

The Drive Based device control describes the control sequence of a drive controller based on a state machine, i.e. a drive controller switches its state due to an event. Some of the conditions and responses coupled with the state transitions can be influenced depending on the application; for example, it is possible to define the end of a quick stop or enable signal delays tailored to the respective application case.

Detailed information on the device control and Drive Based device state machine can be found in the chapter [Drive Based device control](#) [▶ 145].

5.4.1 Parameterizing transition conditions

- ✓ You have projected a drive controller with the Drive Based device control.
- 1. Select the relevant drive controller in the project tree and click on the desired projected axis in the Project menu > Wizard area.
- 2. Select the Drive Based device control wizard.
- 3. A34 Auto start:
If the drive controller is to switch directly to the Operation enabled state after the Switch on disabled state, activate this parameter.
- 4. A43 Enable switch-on/off delay
If you need to hide possible fault or test pulses on the enable signal input, enter the maximum internal activation and deactivation delay.
- 5. A44 Quick stop at enable off:
If the drive is to be stopped with a quick stop in the event of a deactivated enable signal, activate this parameter.
- 6. A60 Source additional enable:
If you are working with multiple enable signals, define the source of additional signals in this parameter.
- 7. A29 Quick stop in case of fault:
If the drive is to be stopped with a quick stop—if possible—in case of a fault, activate this parameter. If the parameter is deactivated, motion of the axis is no longer controlled by the drive controller.
- 8. A39 Maximum quick stop duration for enable off:
If you have selected 1: Active for parameter A44, define the maximum time after which the power unit is switched off in A39.
- 9. A45 Quick stop end:
In this parameter, define whether a quick stop is considered ended when the drive is at a standstill or when the quick stop request is canceled.
- 10. A62 Source /quick stop:
Define how a quick stop is triggered in this parameter.
- 11. E48 Device control state:
Shows the current state of the drive controller.

5.5 Parameterizing synchronous operation

The following chapters describe the basic procedure for parameterizing the master and slave axes for synchronous operation. Different configuration options are available in DriveControlSuite to operate one or more axes in synchronous operation.

You can find more information on synchronous operation and the various configuration options under [Synchronous operation – Concept](#) [▶ 107].

WARNING!

Injury to persons and material damage due to axis movement!

When referencing the master (axis, master encoder or virtual master) or using the preset function for phasing, the master value changes immediately and without a motion profile. As a result, coupled slave axes immediately follow the motion of the master.

- Only execute the functions if there are no slave axes coupled with the master.

5.5.1 Projecting the IGB motion bus network

Project an IGB motion bus network as the basis for the data exchange of multiple SD6 drive controllers in synchronous operation.

Further information on the IGB motion bus network can be found in the chapter [IGB motion bus network](#) [▶ 110].

- ✓ You have configured all drive controllers for synchronous operation in the IGB motion bus network.
1. Select the relevant module in the project tree and click **IGB motion bus** in the project menu.
 - ⇒ The IGB assignment window opens.
 - ⇒ The IGB-capable drive controllers of the module are grouped under **Not assigned**.
 2. Click **Add IGB network**.
 - ⇒ A new IGB network is added.
 3. **Drive controller structure area, IGB network column:**
 Drag and drop the drive controllers from **Unassigned** into the IGB network.
 - ⇒ The IGB target number column shows the number of assigned drive controllers for the IGB network.
 4. Click **Assign IGB addresses automatically**.
 - ⇒ The drive controllers of the IGB network automatically receive a valid IGB address.
 5. **Drive controller structure area, Scaling column:**
 Make sure that the scaling of the parameter values is consistent within the IGB network.

Information

You can change IGB addresses manually by double-clicking on the relevant IGB address and assigning a new unused value. Numerical values from 0 up to and including 31 are accepted.

5.5.2 Parameterizing the master/slave assignment

In the IGB motion bus network, parameterize the assignment of the slave axes to the master and the scaling of the master axis models.

Information

If you use the *Master/Slave* window to perform the master/slave assignment, the mapping of the parameters for the position and time stamp happens automatically in the background and is displayed in the IGB motion bus mapping dialog box (parameters G120, G121, G127, G128).

The same applies to using the master to monitor the slave axes (parameter G209).

Selecting a master

- ✓ You are in the IGB assignment window.
- ✓ You have configured an IGB motion bus network with valid IGB addresses.
 1. IGB networks area:
Click the *Master/Slave* button next to the relevant IGB network.
 - ⇒ The *Master/slave management window* opens.
 2. Click *Select master*.
 - ⇒ The *Select Master dialog box* opens.
 3. Select the master drive controller and confirm with *OK*.
 - ⇒ *Master/slave management window, Master column*:
The selected drive controller is displayed as the master for the IGB network.

Selecting a master value source

- ✓ You are in the *Master/slave management window*.
- ✓ You have selected the master drive controller for the IGB network.
 1. Double-click the master.
 - ⇒ The *Master/Slave window* opens.
 2. Master axis model area, master axis:
Define the master value source for the master axis.
 - 2.1. If a master encoder is acting as the master value source, select 1: Master encoder.
 - 2.2. If a virtual master is acting as the master value source, select 2: Virtual master.
 - 2.3. If the actual position of the master axis is acting as the master value source, select 4: Actual position I80.
- ⇒ Depending on the master value source, the parameters of the master axis model or the axis model of the master axis are displayed.

Assigning slave axes

- ✓ You are in the Master/Slave window.
- ✓ You have selected the master value source for synchronous operation.
- 1. Master axis model area, Slave axes:
Click Add slave axes.
⇒ The Add slave axes window opens.
- 2. Select the desired slave axes and confirm with OK.
- ⇒ Master axis model area, Slave axes:
The assigned slave axes are displayed.

Parameterizing the master axis model

- ✓ You are in the Master/slave management window.
- ✓ You have assigned the master and slave axes.
- 1. Master axis model area:
Parameterize the settings for the travel range, decimal places and unit of measure of the master axis model.
- 2. Master axis model area, Slave axes:
To apply the scaling of the master axis model to all slave axes, click Apply master axis scaling for all slave axes.
- 3. To apply the settings, close the window.

Information

A change to parameter G46 moves the decimal separator for all master position values. Therefore, change G46 before parameterizing other master position values and then check them.

The resolution of master position values has a direct effect on the smooth operation of the coupled slave axes. Therefore, define a sufficient number of decimal places appropriate for your application.

Information

For the synchronous operation of multiple SD6 drive controllers over an IGB motion bus network, the master axis models of the slave axes must be scaled identically to the axis model of the master value source. Scale the axis model identically from the start so that you do not have to adapt position settings to changed scaling afterwards.

5.5.3 Monitoring synchronous operation

In synchronous operation, you have the option to monitor whether selected slave axes are still running synchronously with the master. As soon as a monitored slave axis is no longer synchronous, a quick stop can bring the master to a standstill and thus also stop all slave axes.

Information

When monitoring synchronous operation, an override function is available for commissioning, emergency operation or maintenance and repair work. You can use this to temporarily deactivate the quick-stop behavior of the master that is defined in parameter G108 (override: G116).

Information

An axis is considered synchronous if it has the status I89 = 5: Synchronous motion. This does not take into account whether the source for the master position is inactive, for example, or whether a factor of 0 or an override of 0% has been parameterized for the ratio of the electronic gear unit (master position source: G27; gear ratio: I417, I418; override: G116).

IGB motion bus network: Monitoring synchronous operation

- ✓ You have projected an IGB motion bus network.
 - ✓ You have carried out the mapping for synchronous operation.
 - ✓ You are in the *Master/Slave* window.
1. To activate synchronous operation monitoring, select the *Monitoring* option.
 - ⇒ The function is active and the associated parameters are displayed.
 2. *Slave axes area*:
Select which slave axes the master is to monitor for synchronism.
 3. *G108 Sync monitoring quick stop*:
Select how the master is to behave if one of the monitored axes is no longer synchronous.
 - 3.1. If a virtual master is acting as the master value source, select 1: Quick stop VM.
 - 3.2. If a master encoder or the axis model of the master axis is acting as the master value source, select 2: Quick stop axis.
 4. To accept the settings, close the window by clicking on X.

Information

If you subsequently remove a drive controller that is part of monitoring from the module or project, you must configure the settings for synchronous operation monitoring again.

5.5.4 Parameterizing the master axis

Parameterize the master axis according to your application as described below.

5.5.4.1 Parameterizing the virtual master

If a virtual master is used as the master value source, activate it in the first step and then limit the position, velocity, acceleration and jerk for the virtual master.

Activating the virtual master

1. Select the relevant drive controller in the project tree and click on the desired projected axis in the Project menu > Wizard area.
2. Select the Drive Based Synchronous application wizard > Additional functions.
3. Activate the option Virtual master.
⇒ In the list of wizards, the Virtual master wizard is displayed.

Information

To be able to control the additional virtual master function via fieldbus, expand the process data mapping using the respective wizard (EtherCAT: Received process data RxPDO; PROFINET: Received process data RxPZD).

An overview of the virtual master parameters that can be written by the controller via fieldbus is provided by the Command operating mode: Virtual master wizard. Add the required parameters to both the receive process data of the drive controller and the transmit process data of the controller (motion command: G483; set value specifications: G484 – G488; control words: G10, G12).

Parameterizing the virtual master

- ✓ You have activated the additional virtual master function.
1. Select the relevant drive controller in the project tree and click on the desired projected axis in the Project menu > Wizard area.
 2. Select the Drive Based Synchronous application wizard > Additional functions > Virtual master > Limit: virtual master.
 3. G53 Virtual master move direction:
If you selected 1: Endless for G30 when parameterizing the master axis model, you define the permitted direction of motion for the virtual master.
 - 3.1. If motions in both directions are permitted, select 0: Positive and negative.
 - 3.2. If motions are only permitted in one direction, select 1: Positive or 2: Negative accordingly.
 4. G50 Virtual master maximal speed:
Define the maximum velocity of the virtual master.
 5. G51 Virtual master maximal acceleration:
Define the maximum acceleration of the virtual master.
 6. G54 Virtual master quickstop deceleration:
Define the maximum quick-stop acceleration of the virtual master.
 7. G52 Virtual master maximal jerk:
Define the maximum jerk of the virtual master.

Information

The limiting values for the velocity, acceleration, quick stop and jerk of the virtual master (G50, G51, G52, G54) depend on the limiting values of the slave axes.

To determine the maximum velocity for the virtual master, for example, convert the maximum velocity of each individual slave axis together with the respective gear ratio I417 Gear ratio numerator and I418 Gear ratio denominator into an alternative maximum velocity. The slave axis with the lowest maximum velocity is simultaneously the velocity limit value of the virtual master.

Proceed in the same manner for the limit values for acceleration, quick stop and jerk.

Virtual master: Limiting the travel range

- ✓ You have activated the additional virtual master function.
 - ✓ The travel range of the master is limited (G30 = 0: Limited).
 - 1. Select the relevant drive controller in the project tree and click on the desired projected axis in the Project menu > Wizard area.
 - 2. Select the Drive Based Synchronous application wizard > Additional functions > Virtual master > Limit: virtual master.
 - 3. G146 Virtual master software limit switch +, G147 Virtual master software limit switch -:
In order to protect the travel range, limit the position of your axis if necessary using a positive or negative software limit switch.
 - 4. Select the Protection functions wizard.
 - 5. U24 Level virtual master limit switch:
Select 3: Fault.
 - 6. Select the Drive Based Synchronous application wizard > Additional functions > Virtual master.
 - 7. G57 Virtual master quickstop additional source:
Select 1: Error or 3: Enable or error.
- ⇒ If the software limit switches are overshot, a quick stop of the virtual master is triggered.

5.5.4.2 Parameterizing the master encoder

If a master encoder is used as the master value source, activate the interface you used to connect the master encoder to the drive controller in the first step and then parameterize the master encoder together with the encoder interface.

Parameterizing the master encoder

1. Select the relevant drive controller in the project tree and click on the desired projected axis in the Project menu > Wizard area.
2. Select the Encoder wizard > Master encoder: Scaling.
3. G104 Source master encoder:
Select the interface over which the master encoder is connected.
4. G47 Distance factor numerator master and G48 Distance factor denominator master:
Enter the conversion factor for the path of the master encoder in relation to the master position.
5. G297 maximum-speed master-encoder
Define the value for the permitted maximum velocity of the master encoder.
6. If you have connected the master encoder over the X4 interface, select the Encoder wizard > X4.
 - 6.1. H00 X4 function:
Select the encoder model that is connected to the interface.
⇒ The appropriate parameters are displayed based on the selected encoder.
 - 6.2. H03 Encoder version:
Select whether the encoder design is rotational or translational.
 - 6.3. Parameterize the interface according to the properties of the master encoder.
7. If you have connected the master encoder over the X120 interface, select the Encoder wizard > X120.
 - 7.1. H120 X120 function:
Select the encoder model that is connected to the interface.
⇒ The appropriate parameters are displayed based on the selected encoder.
 - 7.2. H123 Encoder version:
Select whether the encoder design is rotational or translational.
 - 7.3. Parameterize the interface according to the properties of the master encoder.
8. If you have connected the master encoder over the X140 interface, select the Encoder wizard > X140.
 - 8.1. H140 X140 function:
Select the encoder model that is connected to the interface.
⇒ The appropriate parameters are displayed based on the selected encoder.
 - 8.2. H143 Encoder version:
Select whether the encoder design is rotational or translational.
 - 8.3. Parameterize the interface according to the properties of the master encoder.

Information

Parameterize G297 maximum-speed master-encoder according to your application case: If G297 is set too low, the permitted maximum speed is exceeded even at normal operating speeds. If G297 is set too high, measuring errors of the encoder can be overlooked.

G297 depends on the following parameters: G46 Decimal places, G47 Distance factor numerator master, G48 Distance factor denominator master and G49 Master measure unit. If you have made changes to one of the parameters listed, adjust G297 accordingly as well.

5.5.5 Parameterizing slave axes

Parameterize the slave axes according to your application as described below.

5.5.5.1 Parameterizing an electronic gear unit

As an option, you can define a factor for the gear ratio of the electronic gear unit.

1. Select the relevant drive controller in the project tree and click on the desired slave axis in the project menu > Wizard area.
2. Select the Drive Based Synchronous application wizard > Synchronous operation > Electronic gear unit.
3. I418 Gear ratio denominator, I417 Gear ratio numerator:
Define a factor for the gear ratio of the master position and axis movement in the form of a quotient (numerator and denominator).

5.6 Transmitting and saving a configuration

In order to transmit and save the configuration to one or more drive controllers, you must connect your PC and the drive controllers over the network.

WARNING!

Injury to persons and material damage due to axis movement!

If there is an online connection between DriveControlSuite and the drive controller, changes to the configuration can lead to unexpected axis movements.

- Only change the configuration if you have visual contact with the axis.
- Make sure that no people or objects are within the travel range.
- For access via remote maintenance, there must be a communication link between you and a person on site with eye contact to the axis.

Information

During the search, all drive controllers within the broadcast domain are found via IPv4 limited broadcast.

Requirements for finding a drive controller in the network:

- Network supports IPv4 limited broadcast
- All drive controllers and the PC are in the same subnet (broadcast domain)

5.6.1 Transmitting the configuration

The steps for transmitting the configuration vary depending on the safety technology.

Drive controller without SE6 safety module

- ✓ You have verified the plausibility of the predefined test motion variables.
 - ✓ The drive controllers are switched on.
1. In the project tree, select the module under which you have recorded your drive controller and click **Online connection** in the project menu.
 - ⇒ The **Add connection** dialog box opens. All drive controllers found via IPv4 limited broadcast are displayed.
 2. **Direct connection** tab > **IP address** column:
 Activate the IP addresses in question and confirm your selection with **OK**.
 - ⇒ The **Online functions** window opens. All drive controllers connected through the previously selected IP addresses are displayed.
 3. Select the drive controller to which you want to transmit a configuration and change the selection of the transmission type from **Read** to **Send**.
 4. Change the selection **Create new drive controller**:
 Select the configuration that you would like to transfer to the drive controller.
 5. Repeat steps 3 and 4 for all other drive controllers to which you would like to transfer your configuration.
 6. **Online** tab:
 Click **Establish online connection**.
 - ⇒ The configurations are transferred to the drive controllers.

Drive controller with SE6 safety module

- ✓ You have verified the plausibility of the predefined test motion variables.
- ✓ The drive controllers are switched on.
- 1. In the project tree, select the module under which you have recorded your drive controller and click **Online connection** in the project menu.
 - ⇒ The **Add connection** dialog box opens. All drive controllers found via IPv4 limited broadcast are displayed.
- 2. **Direct connection tab > IP address column:**
Activate the IP addresses in question and confirm your selection with **OK**.
 - ⇒ The **Online functions** window opens. All drive controllers connected through the previously selected IP addresses are displayed.
- 3. Select the drive controller to which you want to transmit a configuration and change the selection of the transmission type from **Read** to **Send**.
- 4. Change the selection **Create new drive controller:**
Select the configuration that you would like to transfer to the drive controller.
- 5. Repeat steps 3 and 4 for all other drive controllers to which you would like to transfer your configuration.
- 6. **Online tab:**
Click **Establish online connection**.
 - ⇒ The configurations are transferred to the drive controllers.
 - ⇒ A dialog box prompts you to open the **PASmotion** configuration tool.
- 1. Confirm the dialog box with **Yes**.
 - ⇒ **PASmotion** opens.
- 2. In the **PASmotion** project administration, navigate to the safety module for the drive controller and double-click to open it.
 - ⇒ The dialog box for the password prompt opens.
- 3. Enter the password and confirm with **OK**.
 - ⇒ The wizard for device synchronization opens.
 - ⇒ Device configuration and configuration are checked against each other automatically.
- 4. If the configurations match, click on **Done** after device synchronization has finished.
- 5. Optional: If the configurations do not match, click on **Next** after device synchronization has finished.
 - 5.1. Confirm the production number of the safety module and click **Next**.
 - 5.2. Enter the password for the configuration on the safety module and click **Next**.
 - 5.3. Click **Upload** to transfer the device configuration to the project.
 - 5.4. After the successful transfer, click **Done**.
- 6. Exit **PASmotion**.
 - ⇒ The safety configuration is transferred to the selected drive controller.

5.6.2 Saving the configuration

- ✓ You have successfully transmitted the configuration.
- 1. Online functions window:
Click Save values (A00).
 - ⇒ The Save values (A00) window opens.
- 2. Click Start action.
 - ⇒ The configuration is stored on the drive controllers in non-volatile memory.
- 3. Close the Save values (A00) window.

Information

For the configuration to take effect on the drive controller, a restart is required when the configuration is saved on the drive controller for the first time or when changes are made to the firmware or process data mapping.

Restarting a drive controller

- ✓ You have stored the configuration on the drive controller in non-volatile memory.
- 1. Online functions window:
Click Restart (A09).
 - ⇒ The Restart (A09) window opens.
- 2. Select which of the connected drive controllers you want to restart.
- 3. Click Start action.
- 4. Confirm the safety note with OK.
 - ⇒ The Restart (A09) window closes.
- ⇒ The fieldbus communication and connection between DriveControlSuite and drive controllers are interrupted.
- ⇒ The selected drive controllers restart.

5.7 Testing the configuration

Check the configuration of your projected axis model before continuing with the parameterization of the application. Use **Jog control panel** to test the electronic and mechanical data of your project configuration and parameterization for plausibility. **Control panel motion** gives you the option to test the basic movements of your axis using motion commands.

Information

Check the suitability of the default values for your application before starting the test. If they appear too large or unsuitable compared with the results of the scaling calculator, replace them with values that are more suitable for test operation.

If you are working with a virtual master or master encoder as the master value source, first reference its absolute position and test its functionality. Then, check the functionality of the master/slave coupling (GearIn).

5.7.1 Testing jog mode

Jog control panel provides various commands for jog mode which you can use to test the configuration of your projected axis model for plausibility.

WARNING!

Injury to persons and material damage due to axis movement!

When you activate the control panel, DriveControlSuite gives you sole control of the motions of the axis. If you are using a controller, it no longer monitors the axis movements after the control panel is activated. The controller cannot intervene to prevent collisions. The controller takes over control again when the control panel is deactivated, which can cause unexpected axis movements.

- Do not switch to other windows when the control panel is active.
- Only use the control panel if you have visual contact with the axis.
- Make sure that no people or objects are within the travel range.
- For access via remote maintenance, there must be a communication link between you and a person on site with eye contact to the axis.

Information

If you are using an SD6 drive controller, you can also use jog mode via the operating unit of the SD6.

Information

Make sure that the values of the control panel are compatible with your projected axis model and adjust the default values if necessary. This is the only way to get suitable test results which you can use to optimize your configuration for the respective axis.

The scaling calculator is available under the **Axis model wizard > Axis: Scaling** to recalculate the values for the control panel according to your projected axis model.

Testing the configuration using the jog control panel

- ✓ The drive controller is switched on.
- ✓ You have established an online connection to the drive controller.
- 1. Select the relevant drive controller in the project tree and click on the desired projected axis in the Project menu > Wizard area.
- 2. Select the Jog control panel wizard.
- 3. Click Control panel on and then Enable.
 - ⇒ The axis is monitored via the active control panel.
- 4. Check the default values of the control panel and adjust them to your projected axis model if necessary.
- 5. To test the configuration of your projected axis for direction of motion, velocity, etc., move the axis gradually using the Jog+, Jog-, Jog step+ and Jog step- buttons.
- 6. Use your test results to optimize your configuration as necessary.
- 7. To deactivate the control panel, click on Control panel off.

Information

Jog+ and Jog- cause a continual manual movement in the positive or negative direction. Jog step+ and Jog step- move the axis relative to the current actual position by the increment specified in I14.

Jog+ and Jog- have a higher priority than Jog step+ and Jog step-.

5.7.2 Testing the motion commands

Control panel motion provides a standard set of motion commands which you can use to test the basic movements of your axis. The command set is based on the PLCopen standard and supplemented by manufacturer-specific motion commands.

If you are working with absolute position measuring systems and need the MC_MoveAbsolute motion command for testing, you must reference an absolute position in advance (see [Referencing absolute position](#) [▶ 29]).

WARNING!

Injury to persons and material damage due to axis movement!

When you activate the control panel, DriveControlSuite gives you sole control of the motions of the axis. If you are using a controller, it no longer monitors the axis movements after the control panel is activated. The controller cannot intervene to prevent collisions. The controller takes over control again when the control panel is deactivated, which can cause unexpected axis movements.

- Do not switch to other windows when the control panel is active.
- Only use the control panel if you have visual contact with the axis.
- Make sure that no people or objects are within the travel range.
- For access via remote maintenance, there must be a communication link between you and a person on site with eye contact to the axis.

Information

Make sure that the values of the control panel are compatible with your projected axis model and adjust the default values if necessary. This is the only way to get suitable test results which you can use to optimize your configuration for the respective axis.

The scaling calculator is available under the *Axis model wizard* > *Axis: Scaling* to recalculate the values for the control panel according to your projected axis model.

Testing the configuration using the motion control panel

- ✓ The drive controller is switched on.
 - ✓ You have established an online connection to the drive controller.
1. Select the relevant drive controller in the project tree and click on the desired projected axis in the Project menu > Wizard area.
 2. Select the Control panel motion wizard.
 3. Click Control panel on and then Enable.
 - ⇒ The axis is monitored via the active control panel.
 4. Check the default values of the control panel and adjust them to your projected axis model if necessary.
 5. K402 Command:
 - Select the desired command and define the position, velocity, acceleration, deceleration and jerk.
 6. Execute the command by clicking Start.
 7. To test the configuration of your projected axis for direction of motion, velocity, etc., move the axis precisely using the individual commands.
 8. Use your test results to optimize your configuration as necessary.
 9. To deactivate the control panel, click on Control panel off.

5.7.3 Testing synchronous operation

If you are working with a virtual master or master encoder as the master value source, first reference its absolute position and test its functionality. Then, check the functionality of the master/slave coupling (GearIn).

5.7.3.1 Referencing and testing the virtual master

If you are working with a virtual master as the master value source, reference it and test its functionality using the Control panel motion: Virtual master wizard.

WARNING!

Injury to persons and material damage due to slave axis movement!

By activating this function, the master value changes immediately without a motion profile. As a result, coupled slave axes immediately follow the movement of the master.

- Only execute this function if there are no slave axes coupled with the master.

Referencing the virtual master

- ✓ The drive controller is switched on.
 - ✓ You have established an online connection to the drive controller.
 - ✓ You have completed and read in the configuration for synchronous operation.
1. Select the master drive controller in the project tree and click on the desired projected axis in the project menu > Wizard area.
 2. Select the Drive Based Synchronous application wizard > Additional functions > Virtual master > Control panel motion: Virtual master.
 3. Activate the control panel by clicking Control panel on.
 - ⇒ The control panel is active: The parameters and buttons which you can use to control the virtual master are displayed.
 4. In order to reference the actual position of the virtual master manually, click Set pos.
 - ⇒ The value of parameter G58 Virtual master preset position is transmitted to parameter G190 Actual position virtual master.

Testing the virtual master

- ✓ The drive controller is switched on.
 - ✓ You have established an online connection to the drive controller.
 - ✓ You have completed and read in the configuration for synchronous operation.
1. Select the master drive controller in the project tree and click on the desired projected axis in the project menu > Wizard area.
 2. Select the Drive Based Synchronous application wizard > Additional functions > Virtual master > Control panel motion: Virtual master.
 3. Activate the control panel by clicking Control panel on.
 - ⇒ The control panel is active: The parameters and buttons which you can use to control the virtual master are displayed.
 4. To test the virtual master, assign values to the following parameters.
 5. G483 Virtual master command:
Select the motion command you would like to use to move the virtual master.
 6. G484 Virtual master target position:
Define the set position to which the virtual master is to travel.
 7. G485 Virtual master velocity:
Define the set velocity at which the virtual master is to travel.
 8. G486 Virtual master acceleration:
Define the acceleration at which the virtual master is to travel.
 9. G487 Virtual master deceleration:
Define the deceleration at which the virtual master is to travel.
 10. G488 Virtual master jerk:
Define the jerk at which the virtual master is to travel.
 11. To execute the motion command, click on Start.
 - ⇒ The values of parameters G190 Actual position virtual master, G191 Actual velocity virtual master and G159 Virtual master PLCopen-state change according to the parameterization.

Information

The values of Control panel motion: Virtual master for the set velocity, acceleration and jerk of the virtual master are limited using the Limit: virtual master wizard (parameters G485, G486 and G488 or parameters G50, G51 and G52).

5.7.3.2 Referencing and testing the master encoder

If you are working with an external master encoder as the master value source, reference it and test its function by moving it manually.

 **WARNING!**

Injury to persons and material damage due to slave axis movement!

By activating this function, the master value changes immediately without a motion profile. As a result, coupled slave axes immediately follow the movement of the master.

- Only execute this function if there are no slave axes coupled with the master.

- ✓ The drive controller is switched on.
 - ✓ You have established an online connection to the drive controller.
 - ✓ You have completed and read in the configuration for synchronous operation.
1. Select the master drive controller in the project tree and click on the desired projected axis in the project menu > Wizard area.
 2. Select the Drive Based Synchronous application wizard > Synchronous operation > Master digital signals: Data source.
 3. G453 Source master reference set:
Enter the signal source for triggering the referencing.
 4. To reference the master encoder, trigger the referencing signal.
 5. Select the Drive Based Synchronous application wizard > Synchronous operation > Master.
 6. Move the master encoder mechanically.
- ⇒ The value of parameter G122 Lead-position producer changes according to the movements of the master encoder.

5.7.3.3 Testing the master/slave coupling

Test the master/slave coupling using Control panel motion and the 12: MC_GearIn motion command.

 **WARNING!**

Injury to persons and material damage due to axis movement!

When you activate the control panel, DriveControlSuite gives you sole control of the motions of the axis. If you are using a controller, it no longer monitors the axis movements after the control panel is activated. The controller cannot intervene to prevent collisions. The controller takes over control again when the control panel is deactivated, which can cause unexpected axis movements.

- Do not switch to other windows when the control panel is active.
- Only use the control panel if you have visual contact with the axis.
- Make sure that no people or objects are within the travel range.
- For access via remote maintenance, there must be a communication link between you and a person on site with eye contact to the axis.

Information

To prevent an enabled axis from moving unnoticed in the background, active control panels are deactivated as soon as you open a new wizard.

To test the master/slave coupling (Gear-In), you need multiple control panels at the same time. Therefore, open all necessary wizards in the first step. In the second step, arrange them next to each other in window mode and only activate the required control panels in the third step.

Opening wizards

1. Select the desired drive controller in the project tree and click on the master axis in the project menu > Wizard area.
 2. Open the wizard according to the master value source.
 - 2.1. Virtual master:
Select the Drive Based Synchronous application wizard > Additional functions > Virtual master > Jog control panel: Virtual master.
 - 2.2. Master encoder:
Select the Drive Based Synchronous application wizard > Synchronous operation > Master.
 - 2.3. Motor axis model of the master axis:
Select the Drive Based Synchronous application wizard > Jog control panel.
 3. Select the desired drive controller in the project tree, hold down [CTRL] and click on the first slave axis in the project menu > Wizard area.
 4. Select the Control panel motion wizard.
 5. Select Windows in the menu bar > Show windows side by side.
 - ⇒ The wizards are arranged next to each other.
- ⇒ You can now move the master and check the behavior of the slaves at the same time without the control panels being automatically deactivated.

Coupling the slave axis

- ✓ The drive controller is switched on.
 - ✓ You have established an online connection to the drive controller.
 - ✓ You have completed and read in the configuration for synchronous operation.
 - ✓ You have opened all necessary wizards and arranged them next to each other in window mode.
1. Select the previously opened Control panel motion wizard of the slave axis.
 2. Activate the control panel by clicking Control panel on.
 - ⇒ The control panel is active: The parameters and buttons which you can use to move the axis are displayed.
 3. K402 Command:
Select 12: MC_GearIn.
 4. Click on Start.
 - ⇒ Parameter E80 Operating condition indicates the state 29: Synchronous motion.
 - ⇒ The slave axis is active and coupled to the master.

Moving the master

- ✓ The drive controller is switched on.
 - ✓ You have established an online connection to the drive controller.
 - ✓ You have completed and read in the configuration for synchronous operation.
 - ✓ You have opened all necessary wizards and arranged them next to each other in window mode.
1. Open the wizard according to the master value source.
 2. Virtual master:
 - 2.1. Select the Jog control panel: Virtual master wizard.
 - 2.2. Activate the control panel by clicking Control panel on.
 - 2.3. Move the virtual master using jog.
 - ⇒ The value in parameter G190 changes according to the position of the virtual master.
 3. Master encoder:
 - 3.1. Select the Master wizard.
 - 3.2. Move the master encoder mechanically.
 - ⇒ The value in parameter G122 changes according to the position of the master encoder.
 4. Motor axis model of the master:
 - 4.1. Select the previously opened Jog control panel wizard of the master axis.
 - 4.2. Activate the control panel by clicking Control panel on.
 - 4.3. Move the master axis using jog.
 - ⇒ The value in parameter I80 changes according to the position of the master axis.
 - ⇒ The actual position of the slave axis changes depending on the master position (Control panel motion wizard, parameter I80).

5.8 Parameterizing the Drive Based Synchronous application

The Drive Based Synchronous application provides two operating modes: Command operating mode and motion block operating mode

The operating modes provide you with a standard set of motion commands based on PLCopen that are supplemented with unique motion commands. The operating mode that is best suited to your application depends on your application case. If you are coordinating motion sequences using a controller, use the command operating mode. If you are not using a controller and are instead storing the motion profiles on the respective drive controller, use the motion block operating mode.

The general motion variables are grouped in the mode-independent *Data sources* wizard. Depending on the operating mode, unique wizards are also available which you can use to parameterize operating mode-specific motion variables. The general or operating mode-specific motion variables you need to parameterize depend on your drive project.

If you would like to use set values from external sources for your project, parameterize them in the first step. Then, continue with the operating mode-specific motion variables in the respective wizards.

5.8.1 General motion variables and signal sources

First, parameterize the general motion variables and signal sources before you begin parameterization of the operating mode-specific motion variables. The motion variables that you need to parameterize depend on your drive project.

5.8.1.1 Velocity – Parameterizing sources

If you are obtaining the set value for velocity from external sources for applications of the Drive Based Synchronous type, enter this as described below.

If you transmit data via fieldbus, the set value can either be obtained directly from an internal parameter or indirectly by reading out a parameter (such as with PID controllers).

Set velocity – Parameterizing the source

If you are obtaining the set value for velocity from an external source, enter the source for the main set value.

✓ Use the motion command MC_MoveSpeed or MC_MoveVelocity.

1. Select the Drive Based Synchronous application wizard > Data sources > External velocity: Data source.
2. G461 Source external velocity:
 - 2.1. If an analog input acts as the source, select the corresponding input.
 - 2.2. If you are transmitting data via fieldbus and would like to configure the value as the general data source, select 4: Parameter G460.
 - 2.3. If the set value is read out indirectly via a parameter, select 5: Indirect read parameter G811.
3. G460 External velocity:

If you have selected 4: Parameter G460 for G461, enter the set velocity here.
4. G811 Indirect read external velocity:

If you have selected 5: Indirect read parameter G811 for G461, enter the associated parameter coordinate here.

Additional velocity – Parameterizing the source

If you also want to regulate the set velocity, enter the source for the additional set value.

- ✓ Use the motion command MC_MoveSpeed or MC_MoveVelocity.
 - ✓ You obtain the set velocity from an external source.
1. Select the Drive Based Synchronous application wizard > Data sources > External additional velocity: Data source.
 2. G464 Source external additional velocity:
 - 2.1. If an analog input acts as the source, select the corresponding input.
 - 2.2. If you are transmitting data via fieldbus and would like to configure the value as the general data source, select 4: Parameter G463.
 - 2.3. If the set value is read out indirectly via a parameter, select 5: Indirect read parameter G812.
 3. G463 External additional velocity:

If you have selected 4: Parameter G463 for G464, enter the set velocity here.
 4. G812 Indirect read additional external velocity:

If you have selected 5: Indirect read parameter G812 for G463, enter the parameter coordinate here.

5.8.1.2 Velocity override – Parameterizing the source

If you would like to use a velocity override to scale the velocity profile of your motion commands, enter the source for the velocity override.

If you transmit data via fieldbus, the set value can either be obtained directly from an internal parameter or indirectly by reading out a parameter (such as with PID controllers).

1. Select the Drive Based Synchronous application wizard > Data sources > Velocity override: Data source.
2. G467 Source velocity override:
 - 2.1. If an analog input acts as the source, select the corresponding input.
 - 2.2. If you are transmitting data via fieldbus and would like to configure the value as the general data source, select 4: Parameter G466.
 - 2.3. If the set value is read out indirectly via a parameter, select 5: Indirect read parameter G813.
3. G466 Velocity override:

If you have selected 4: Parameter G466 for G467, enter the set velocity here.
4. G813 Indirect read velocity override:

If you have selected 5: Indirect read parameter G813 for G467, enter the parameter coordinate here.

5.8.1.3 Torque/Force – Parameterizing sources

If you have selected the motion command MC_TorqueControl for Drive Based Synchronous type applications and obtain set values from external sources, enter the sources for the set torque/force and velocity bracketing.

If you transmit data via fieldbus, the set value can either be obtained directly from an internal parameter or indirectly by reading out a parameter (such as with PID controllers).

Set torque/force – Parameterizing the source

If you obtain the set value for torque/force from an external source, enter the source for the set torque/force.

✓ You are using the 9: MC_TorqueControl motion command.

1. Select the Drive Based Synchronous application wizard > Data sources > Set torque/force, velocity bracketing: Data source.
2. G470 Source torque/force reference:
 - 2.1. If an analog input acts as the source, select the corresponding input.
 - 2.2. If you are transmitting data via fieldbus and would like to configure the value as the general data source, select 4: Parameter G469.
 - 2.3. If the set value is read out indirectly via a parameter, select 5: Indirect read parameter G814.
3. G469 Torque/Force reference:

If you have selected 4: Parameter G469 for G470, define the value for set torque/force here. The value relates to C03 Maximum positive torque/force and C05 Maximum negative torque/force.
4. G814 Indirect read torque/force reference:

If you have selected 5: Indirect read parameter G814 for G470, define the associated parameter coordinate here.
5. G500 Torque/force ramp removal, G501 Torque/force ramp removal:

Define the ramps for the build-up or reduction of torque/force.

Velocity bracketing – Parameterizing the source

If you obtain values for limiting the velocity from an external source, enter the source for the velocity bracketing.

- ✓ You are using the 9: MC_TorqueControl motion command.
- 1. Select the Drive Based Synchronous application wizard > Data sources > Set torque/force, velocity bracketing: Data source.
- 2. G473 Source velocity bracketing positiv:
 - 2.1. If an analog input acts as the source, select the corresponding input.
 - 2.2. If you are transmitting data via fieldbus, select 4: Parameter G472.
- 3. G472 Velocity bracketing positiv:

If you have selected 4: Parameter G472 for G473, define the maximum limit that you want to permit for your mechanical system.
- 4. G476 Source velocity bracketing negative:
 - 4.1. If an analog input acts as the source, select the corresponding input.
 - 4.2. If you are transmitting data via fieldbus, select 4: Parameter G475.
- 5. G475 Velocity bracketing negative:

If you have selected 4: Parameter G475 for G476, define the maximum limit that you want to permit for your mechanical system here.

Information

If you define the velocity bracketing using G472 or G475, you should allow the control enough room. Therefore, select a value for the velocity bracketing that is greater than 1%. G472 and G475 refer to I10 Maximal speed.

5.8.1.4 Parameterizing analog inputs

If the analog inputs serve as a source for the application in your drive project, parameterize the respective analog input as described below.

For more information and continuous function charts for analog inputs AI1 – AI3, see [Analog inputs \[► 87\]](#).

5.8.1.4.1 Parameterizing analog input AI1

Parameterize the analog input as a voltage source by defining a time constant for the associated filter and calibrating using the maximum positive, maximum negative and 0 V voltage. After that, you can scale the analog input as needed.

If the analog input is the current source, calibrate using the maximum and minimum currents instead, and optionally enable wire break monitoring.

Parameterizing analog input AI1

1. Select the **Terminals wizard** > **Analog input 1**.
2. **F13 AI1 lowpass filter**:
If the quality of the signal requires it, adjust the time constant for the filter accordingly.
3. **F15 AI1 wire break monitoring**:
If you want to use analog input AI1 as a current source and enable wire break monitoring, select 1: Active.
4. **F116 AI1 operating mode**:
Select the operating mode of analog input AI1.
 - 4.1. If analog input AI1 is the voltage source, select 0: -10V to 10V.
 - 4.2. If analog input AI1 is the current source, select 1: 0 to 20mA or 2: 4 to 20mA.

Calibrating analog input AI1 (voltage source)

- ✓ Analog input AI1 is interpreted as a voltage source (F116 = 0: -10V to 10V).
1. Select the **Terminals wizard** > **Analog input 1** > **Analog input 1: Calibration**.
 2. Set the voltage source to the minimum voltage.
 - 2.1. **F112 AI1 maximum negative value**:
Apply the value from parameter E110[1] for parameter F112.
 - 2.2. **F114 AI1 maximum negative % value**:
Define the associated value for F112 in % in parameter F114.
 3. Set the voltage source to the maximum voltage.
 - 3.1. **F111 AI1 maximum positive value**:
Apply the value from parameter E110[1] for parameter F111.
 - 3.2. **F115 AI1 maximum positive % value**:
Define the associated value for F111 in % in parameter F115.
 4. Set the voltage source to 0 V voltage.
 - 4.1. **F110 AI1 zero point offset**:
Apply the value from parameter E110[1] for parameter F110.
 5. **F113 AI1 deadband**:
Optionally define a deadband to compensate for any noise of the 0 V voltage at the analog input.

Calibrating analog input AI1 (current source)

- ✓ Analog input AI1 is interpreted as a current source (F116 = 1: 0 to 20mA or 2: 4 to 20mA).
- 1. Select the **Terminals wizard** > **Analog input 1** > **Analog input 1: Calibration**.
- 2. Set the current source to the minimum current.
 - 2.1. F110 AI1 zero point offset:
Apply the value from parameter E110[1] for parameter F110.
 - 2.2. F114 AI1 maximum negative % value:
Define the associated value for F110 in % in parameter F114.
- 3. Set the current source to the maximum current.
 - 3.1. F111 AI1 maximum positive value:
Apply the value from parameter E110[1] for parameter F111.
 - 3.2. F115 AI1 maximum positive % value:
Define the associated value for F111 in % in parameter F115.

Scaling analog input AI1

- 1. Select the **Drive Based Center Winder application wizard** > **Analog inputs: Scaling** > **Analog input 1: Scaling**.
- 2. G260 AI1 scaling:
Select the scaling for the analog input.
 - 2.1. If you want to use the value unscaled, select 0: Without scaling.
 - 2.2. If you want to use the value unsigned, select 1: Absolute value.
 - 2.3. If you want to use the value inverted, select 2: Inverting.
 - 2.4. If you want to scale the value using a curve, select 3: X/Y characteristic.
⇒ Parameters G261 and G262 for scaling the analog input using a curve are displayed.
- 3. G261 AI1 X-value scaling, G262 AI1 Y-value scaling:
If you have selected 3: X/Y characteristic for G260, define the value pairs of the calibrated value and scaled value for the scaling of the analog input.
 - ⇒ The scaling of the analog input is completed.
 - ⇒ The scaled value of analog input AI1 is displayed in G270.

5.8.1.4.2 Parameterizing analog input AI2

Parameterize the analog input as a voltage source by defining a time constant for the associated filter and calibrating using the maximum positive, maximum negative and 0 V voltage. After that, you can scale the analog input as needed.

Parameterizing analog input AI2

1. Select the **Terminals wizard** > **Analog input 2**.
2. **F23 AI2 lowpass filter:**
If the quality of the signal requires it, adjust the time constant for the filter accordingly.

Calibrating analog input AI2

1. Select the **Terminals wizard** > **Analog input 2** > **Analog input 2: Calibration**.
2. Set the voltage source to the minimum voltage.
 - 2.1. **F122 AI2 maximum negative value:**
Apply the value from parameter E111[1] for parameter F122.
 - 2.2. **F124 AI2 maximum negative % value:**
Define the associated value for F122 in % in parameter F124.
3. Set the voltage source to the maximum voltage.
 - 3.1. **F121 AI2 maximum positive value:**
Apply the value from parameter E111[1] for parameter F121.
 - 3.2. **F125 AI2 maximum positive % value:**
Define the associated value for F121 in % in parameter F125.
4. Set the voltage source to 0 V voltage.
 - 4.1. **F120 AI2 zero point offset:**
Apply the value from parameter E111[1] for parameter F120.
5. **F123 AI2 deadband:**
Optionally define a deadband to compensate for any noise of the 0 V voltage at the analog input.

Scaling analog input AI2

1. Select the **Drive Based Center Winder application wizard** > **Analog inputs: Scaling** > **Analog input 2: Scaling**.
2. **G263 AI2 scaling:**
Select the scaling for the analog input.
 - 2.1. If you want to use the value unscaled, select 0: Without scaling.
 - 2.2. If you want to use the value unsigned, select 1: Absolute value.
 - 2.3. If you want to use the value inverted, select 2: Inverting.
 - 2.4. If you want to scale the value using a curve, select 3: X/Y characteristic.
⇒ Parameters G264 and G265 for scaling the analog input using a curve are displayed.
3. **G264 AI2 X scaling, G265 AI2 Y scaling:**
If you have selected 3: X/Y characteristic for G263, define the value pairs of the calibrated value and scaled value for the scaling of the analog input.
⇒ The scaling of the analog input is completed.
⇒ The scaled value of analog input AI2 is displayed in G271.

5.8.1.4.3 Parameterizing analog input AI3

Parameterize the analog input as a voltage source by defining a time constant for the associated filter and calibrating using the maximum positive, maximum negative and 0 V voltage. After that, you can scale the analog input as needed.

Parameterizing analog input AI3

1. Select the **Terminals wizard** > **Analog input 3**.
2. **F33 AI3 lowp. filter:**
If the quality of the signal requires it, adjust the time constant for the filter accordingly.

Calibrating analog input AI3

1. Select the **Terminals wizard** > **Analog input 3** > **Analog input 3: Calibration**.
2. Set the voltage source to the minimum voltage.
 - 2.1. **F132 AI3 maximum negative value:**
Apply the value from parameter E112[1] for parameter F132.
 - 2.2. **F134 AI3 maximum negative % value:**
Define the associated value for F132 in % in parameter F134.
3. Set the voltage source to the maximum voltage.
 - 3.1. **F131 AI3 maximum positive value:**
Apply the value from parameter E112[1] for parameter F131.
 - 3.2. **F135 AI3 maximum positive % value:**
Define the associated value for F131 in % in parameter F135.
4. Set the voltage source to 0 V voltage.
 - 4.1. **F130 AI3 zero point offset:**
Apply the value from parameter E112[1] for parameter F130.
5. **F133 AI3 deadband:**
Optionally define a deadband to compensate for any noise of the 0 V voltage at the analog input.

Scaling analog input AI3

1. Select the **Drive Based Center Winder application wizard** > **Analog inputs: Scaling** > **Analog input 3: Scaling**.
2. **G266 AI3 scaling:**
Select the scaling for the analog input.
 - 2.1. If you want to use the value unscaled, select 0: Without scaling.
 - 2.2. If you want to use the value unsigned, select 1: Absolute value.
 - 2.3. If you want to use the value inverted, select 2: Inverting.
 - 2.4. If you want to scale the value using a curve, select 3: X/Y characteristic.
⇒ Parameters G267 and G268 for scaling the analog input using a curve are displayed.
3. **G267 AI3 X-value scaling, G268 AI3 Y-value scaling:**
If you have selected 3: X/Y characteristic for G266, define the value pairs of the calibrated value and scaled value for the scaling of the analog input.
⇒ The scaling of the analog input is completed.
⇒ The scaled value of analog input AI3 is displayed in G272.

5.8.2 Parameterizing motion block operating mode

The following graphic shows the signal flows of motion block operating mode. The elements shown in a lighter shade are optional.

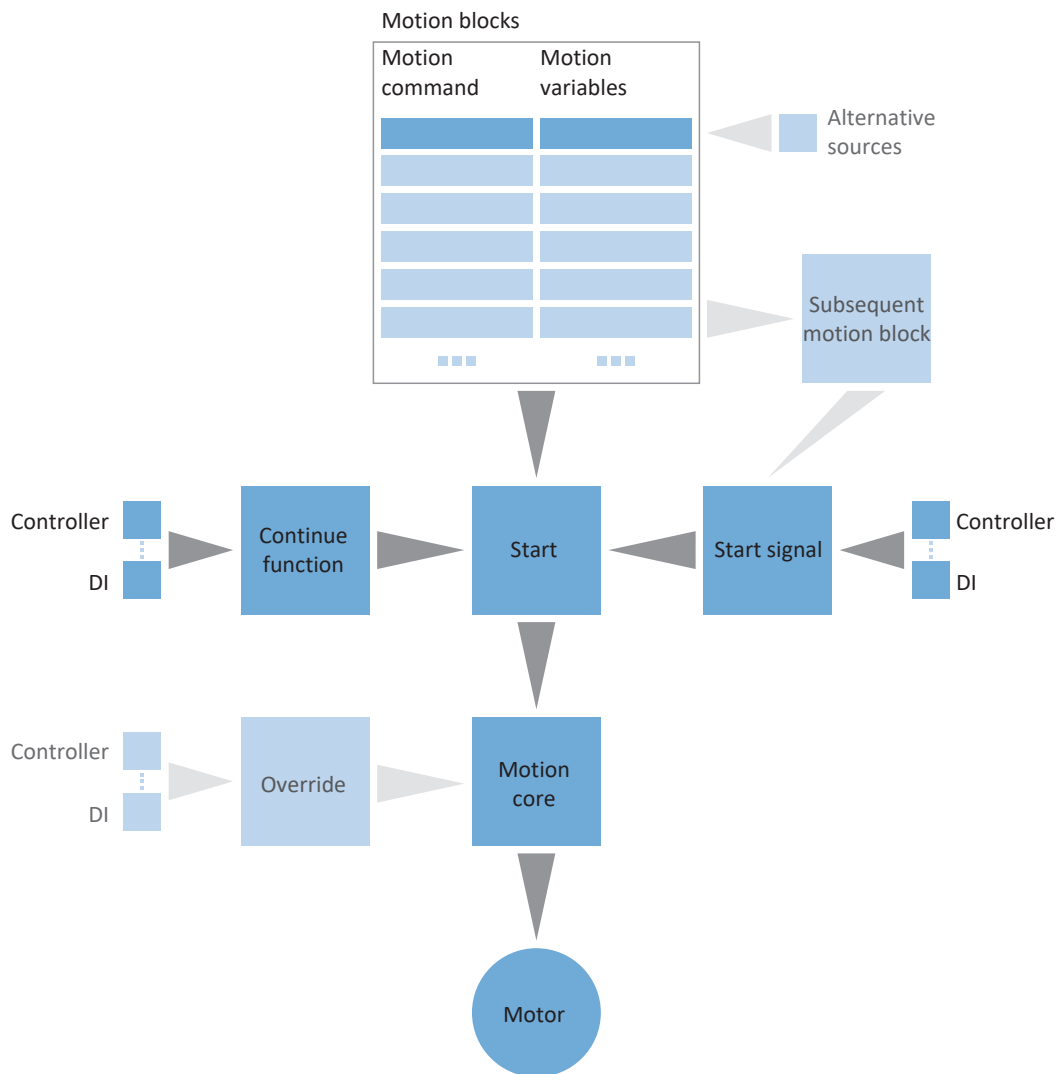


Fig. 3: Motion block operating mode: Signal flows

In the motion block operating mode, motion profiles are permanently stored on the drive controller in the form of motion blocks. A motion block includes a PLCopen-compliant motion command which defines the basic motion method of the axis. The associated motion variables supply specifications on velocity, acceleration, deceleration and jerk, which the motion core of the drive controller uses to calculate the motion profile.

Motion blocks can be selected individually using motion block numbers and they can be linked with each other, i.e. they are automatically processed in succession. In motion block linking, a linear, defined sequence as well as branches based on conditions are both possible for the further motion block action. You can parameterize specifications on velocity override and brake for each motion block individually.

In motion block operating mode, drive controllers are either activated using a fieldbus or by a mixed operation of a fieldbus and terminals. The number of axes and motion blocks used determines the memory you need for your project.

Proceed as follows ...

- Activate motion block operating mode.
- Parameterize one or more motion blocks.
- Test the individual motion blocks using the motion block control panel.
- If you want to link motion blocks with each other, define subsequent motion blocks and triggers for continuation.
- Test the motion block links using the motion block control panel.
- Define how the motion blocks are started.
- Parameterize the source for the continue, reset and, if necessary, stop signals.

Information

The motion variables available for this operating mode that you configure in the software depend on your application case as well as on additional project-specific factors such as the use of a controller or the type of data transmission (fieldbus, terminals).

5.8.2.1 Activating motion block operating mode

1. Select the relevant drive controller in the project tree and click on the desired projected axis in the Project menu > Wizard area.
 2. Select the Drive Based Synchronous application wizard.
 3. Operating modes area:
Activate the Motion block operating mode option.
- ⇒ The corresponding wizards are displayed.

5.8.2.2 Limiting the torque/force using the operating mode

Define whether you also want to limit torque/force using the operating mode. Limiting using the mechanical system takes place in the project configuration of the axis model. You have the option of specifying an additional limit for all motion blocks together or for each motion block individually.

1. Select the Drive Based synchronous application wizard > Motion block operating mode > Motion block > Torque/force limit: Data source.
2. J08 Source maximum T/F:
 - 2.1. If you would like to specify the maximum value from the axis model for all motion blocks centrally, select 3: 100%.
 - 2.2. If you would like to specify the maximum value using the operating mode for all motion blocks centrally, select 1: Parameter J09.
 - 2.3. If you would like to define an individual maximum value for each motion block, select 2: Parameter J25.
3. J09 Maximum torque/force:

If you have selected 1: Parameter J09 for J08, enter the maximum permitted value for torque/force.
4. If you have selected 2: Parameter J25 for J08, enter the maximum permitted value for torque/force during the parameterization of individual motion blocks in J25 Maximum torque/force.

To do so, proceed as described in the chapter [Parameterizing the motion block](#) [▶ 63].

Information

The mechanical limit for torque/force is executed by C03 Maximum positive torque/force and C05 Maximum negative torque/force using the axis model. The additional limit using the operating mode is optional and refers to the values that you defined in the axis model.

5.8.2.3 Parameterizing motion blocks

Before you start parameterizing the individual motion blocks, determine whether you also want to limit torque/force using the operating mode. You have already defined the maximum value for torque/force in the axis model, but you have the option of specifying an additional limit for all motion blocks together or for each motion block individually.

For each motion block, select an appropriate motion command and parameterize the associated motion variables. You can copy parameterized motion blocks completely to create other motion blocks.

Then, test the parameterization of the individual motion blocks using [Motion block control panel](#).

Information

Before you start parameterizing the operating mode-specific motion variables, parameterize the general motion variables and signal sources. If you obtain set values for velocity, velocity override or torque/force for your drive project from external sources, proceed as described in the chapter [General motion variables and signal sources](#) [[▶ 52](#)].

5.8.2.3.1 Parameterizing the motion block

To parameterize a motion block, first select the desired motion command, then parameterize the associated motion variables according to your drive project.

- ✓ You have activated motion block operating mode.
- 1. Select the [Drive Based Synchronous application wizard](#) > [Motion block operating mode](#).
 - ⇒ By default, 16 motion blocks are available to you which you can parameterize individually.
- 2. To parameterize the first motion block, click its motion block number in the [Number](#) column.
 - ⇒ The details view for the first motion block is displayed.
- 3. [J10 Name](#):
Define the name of the motion block.
- 4. [J11 Command](#):
Select the desired motion command.
 - ⇒ The associated parameters are displayed.
- 5. [J25 Maximum torque/force](#):
If you have also limited torque/force using the operating mode and have selected [2: Parameter J25 for J08](#), enter the maximum permitted value for torque/force for each motion block.
- 6. Parameterize the selected motion command based on its associated parameter.

Information

Using [J28 Velocity override enable](#), you activate the scaling of the velocity profile of a motion block. For motion blocks with final velocity, the velocity override influences [J14 Velocity](#) and [J18 End velocity](#) equally, i.e. the relationship between [J14](#) and [J18](#) remains the same.

Information

Use [J21 Command Mode](#) to define how the motion block, as the subsequent motion block, is executed for motion block linking. If you select [0: Abort current command](#), the ongoing motion block is interrupted when the trigger is activated and the subsequent motion block is executed immediately. If you select [1: Start when current command has finished](#), the ongoing motion block is completed first before the subsequent motion block is started.

Moving the axis with a set velocity

- ✓ You have selected the motion command MC_MoveSpeed or MC_MoveVelocity.
- 1. J30 Source velocity:
Select the source from which you will obtain the set value for the velocity.
 - 1.1. If you obtain the set value via fieldbus, select 0: Parameter J14.
 - 1.2. If you obtain the set value from an external source, select 1: Parameter G462.
- 2. J14 Velocity:
If you have selected 0: Parameter J14 for J30, enter the set velocity.
- 3. J15 Acceleration, J16 Deceleration, J17 Jerk:
Enter the set values for acceleration, deceleration and jerk.
- 4. J28 Velocity override enable:
If you would like to apply a velocity override to the motion block, select 1: Active.
- 5. J21 Command Mode:
Select the mode in which the selected motion block, as the subsequent motion block, is to be executed.

Moving the axis with a set torque/force

- ✓ You have selected the motion command MC_TorqueControl.
- 1. J32 Source reference torque/force:
Select the source from which you will obtain the set value for torque/force.
 - 1.1. If you obtain the set value via fieldbus, select 0: Parameter J26.
 - 1.2. If you obtain the value from external sources, select 1: Parameter G471.
- 2. J26 Torque/Force reference:
If you have selected 0: Parameter J26 for J32, enter the set torque or set force.
- 3. J16 Deceleration, J17 Jerk:
Enter the set values for deceleration and jerk.
- 4. J28 Velocity override enable:
If you would like to apply a velocity override to the motion block, select 1: Active.
- 5. J21 Command Mode:
Select the mode for the motion block in which it, as the subsequent motion block, is to be executed.

Moving the axis to the set position

- ✓ You have selected the motion command MC_MoveAbsolute, MC_MoveAdditive or MC_MoveRelative.
- 1. J13 Position:
Enter the absolute or relative position to which the axis is to be moved.
- 2. J14 Velocity:
Enter the set value for the velocity.
- 3. J15 Acceleration, J16 Deceleration, J17 Jerk:
Enter the set values for acceleration, deceleration and jerk.
- 4. J18 End velocity:
Enter the final velocity with which you want to reach the set position J13.
- 5. J24 Waiting time at the end:
If a subsequent motion block is not to be started immediately, enter the wait time at the end of the motion block.
- 6. J27 Engage brake at the end:
If you have parameterized a brake and it is to engage at the end of the motion block, select 1: Active.
- 7. J28 Velocity override enable:
If you would like to apply a velocity override to the motion block, select 1: Active.
- 8. J21 Command Mode:
Select the mode for the motion block in which it, as the subsequent motion block, is to be executed.

Information

For motion blocks with a final velocity that is not 0, a change of travel direction by the subsequent motion block is not permitted.

Referencing the axis

- ✓ You have selected the motion command MC_Home.
- 1. I30 Referencing type:
Indicates what referencing type you have parameterized for the referencing process.
- 2. J16 Deceleration, J17 Jerk:
Enter the set values for deceleration and jerk with which the axis is stopped if referencing is interrupted by a control panel.
- 3. J24 Waiting time at the end:
If a subsequent motion block is not to be started immediately, enter the wait time at the end of the motion block.
- 4. J27 Engage brake at the end:
If you have parameterized a brake and it is to engage at the end of the motion block, activate J27.
- 5. J28 Velocity override enable:
If you have parameterized a velocity override and would like to apply it to the motion block, select 1: Active.

Information

To parameterize the referencing process, click the [Referencing](#) link and proceed as described in the chapter [Referencing absolute position](#) [► 29].

Stopping the axis

- ✓ You have selected the motion command MC_Halt or MC_Stop.
- 1. J16 Deceleration, J17 Jerk:
Enter the set values for deceleration and jerk with which the axis is stopped.
- 2. J24 Waiting time at the end:
If a subsequent motion block is not to be started immediately, enter the wait time at the end of the motion block.
- 3. J27 Engage brake at the end:
If you have parameterized a brake and it is to engage at the end of the motion block, activate J27.

Move the axis synchronously

- ✓ You have selected the motion command MC_GearIn or MC_GearInPosition.
- 1. J15 Acceleration, J16 Deceleration, J17 Jerk:
Enter the set values for acceleration, deceleration and jerk.
- 2. J13 Position:
If you have selected MC_GearInPosition, enter the position of the axis from which the axis is to move synchronously with the master.
- 3. J20 Position D (Master Sync Position):
If you have selected MC_GearInPosition, enter the position of the master from which the axis is to move synchronously with the master.
- 4. J28 Velocity override enable:
If you would like to apply a velocity override to the motion block, select 1: Active.

| |
|--------------------|
| Information |
|--------------------|

To decouple a slave axis from synchronous operation, you can use the motion commands MC_GearOut or MC_Halt, for example. MC_GearOut decouples the axis and retains the last velocity before decoupling; MC_Halt brings the axis to a standstill after decoupling.

5.8.2.3.2 Parameterizing the motion block duration

If you are using an endless motion command to move the axis with a set velocity or set torque/force, for example, you can switch to a subsequent motion block after a defined time using the additional motion block duration function and an indirect trigger.

- ✓ You are using an endless motion command.
- 1. Select the Drive Based Synchronous application wizard > Motion block operating mode > Subsequent motion block > Indirect trigger: Data source.
- 2. Activate the Motion block duration option.
 - ⇒ The additional function is activated and the associated parameter J452 FSM block duration is shown in the Motion block wizard.
- 3. Click on the Motion block link.
 - ⇒ The Motion block wizard opens.
- 4. Number:
Use the arrow keys to select the number of the motion block for which you would like to define a motion block duration.
- 5. J452 FSM block duration:
Enter the duration for which the motion block is to be executed.

Information

As soon as the motion block duration is reached, the information is stored in bit 0 of parameter J453 FSM end of block duration. To continue on to a subsequent motion block once the motion block duration is reached, parameterize a subsequent motion block using the interrupt signal with an indirect trigger as described in [Parameterizing the subsequent motion block with a trigger](#) [▶ 69].

5.8.2.3.3 Copying a motion block

To make parameterization of similar motion blocks easier, you can apply the values from existing motion blocks.

- ✓ You are in the Motion block wizard.
- ✓ You have parameterized at least one motion block.
- 1. Number:
Select the number of the motion block to which you want to copy the values from an existing motion block.
- 2. Apply from:
Click on **Apply from** and select the existing motion block whose values you want to copy.
- 3. Confirm by clicking on **OK**.
 - ⇒ The values of the existing motion block are applied to the current motion block.
- 4. Adjust the copied values, such as J10 Name, if necessary.

5.8.2.3.4 Testing the motion block parameterization

Motion block control panel provides all functions you need to test the parameterization of individual motion blocks.

WARNING!

Injury to persons and material damage due to axis movement!

By activating the control panel, you have sole control over the movements of the axis using DriveControlSuite. If you are using a controller, it no longer monitors the axis movements after the control panel is activated. The controller cannot intervene to prevent collisions. The controller takes over control again when the control panel is deactivated, which can cause unexpected axis movements.

- Only use the control panel if you have visual contact with the axis.
- Make sure that no people or objects are within the travel range.

✓ The drive controller is switched on.

✓ You have established an online connection to the drive controller.

✓ You have parameterized at least one motion block.

1. Select the Drive Based synchronous application wizard > Motion block operating mode > Motion block control panel.
2. Click Control panel on and then Enable.
⇒ The drive is controlled using the active control panel.
3. J311 Motion block selection:
Select the motion block whose parameterization you would like to test.
4. Test the positions, velocities, accelerations and brakes and optimize your parameterization based on your test results if necessary.
5. If you want to link motion blocks with each other as subsequent motion blocks, first test the parameterization for each motion block individually to be able to rule out errors in motion block linking.
6. In order to deactivate the control panel, click Control panel off.

5.8.2.4 Parameterizing motion block linking

In order to configure rolling programs automatically, you can connect multiple motion blocks with each other in sequence. To do so, link the motion blocks in question with each other as subsequent motion blocks and enter the associated signal sources for continuation. The subsequent motion blocks that are assigned to a motion block are displayed in the motion block overview.

Then, test the linking of the individual motion blocks using [Motion block control panel](#).

5.8.2.4.1 Parameterizing subsequent motion blocks

You can define one or more possible subsequent motion blocks for each motion block. Continuation to the subsequent motion block either happens automatically at the end of the motion block or is caused by a trigger. The trigger can either be a direct trigger signal or a trigger condition that is configurable in the software (indirect trigger). Buffered subsequent motion blocks are parameterized without triggers and are intended for the case where the axis is moved to a set position in the previous motion block that is reached with the final velocity.

You can find more information on motion block linking and subsequent motion block under [Motion block](#) [▶ 85].

5.8.2.4.1.1 Parameterizing the subsequent motion block with a trigger

You can define up to five subsequent motion blocks that are each activated by a trigger signal or trigger condition (indirect trigger) for each motion block in the [Subsequent motion block wizard](#). Depending on the configuration, the subsequent motion block is only executed at the end of the previous motion block, or the ongoing motion block is interrupted when an interrupt signal is triggered or an interrupt condition occurs and the subsequent motion block is started.

For each subsequent motion block, define the signal sources for continuation by parameterizing either a direct trigger signal or an indirect trigger condition. You can configure multiple triggers for each motion block. Each trigger can either activate a subsequent motion block through interruption or a subsequent motion block at the end of the previous motion block. If multiple triggers are active simultaneously, the sequence continues with the subsequent motion block with the highest priority (A – E).

Information

Depending on the application case, you can either parameterize a regular subsequent motion block or a buffered subsequent motion block at the end of the previous motion block. The use of a regular subsequent motion block at the end rules out the simultaneous use of a buffered subsequent motion block for the same motion block, and vice versa. You can combine subsequent motion blocks through interruption with regular subsequent motion blocks at the end and with buffered subsequent motion blocks.

Information

J450 Source trigger and J439 Source indirect trigger are array parameters. The first element (J450[0], J439[0]) corresponds to trigger 1, the second element (J450[1], J439[1]) corresponds to trigger 2, etc.

Parameterizing the subsequent motion block at the end of motion block

Use subsequent motion blocks at the end of the motion block if you want to move the axis with a finite motion command and automatically switch to the subsequent motion block when the specified set value is reached.

The following section explains the process as an example, based on the parameterization of subsequent motion block A. For the parameterization of subsequent motion block B – E, proceed in the same manner.

- ✓ You have parameterized at least two motion blocks.
 - ✓ You are moving the axis with MC_MoveAbsolute, MC_MoveRelative or MC_MoveAdditive in the motion block to which you would like to add a subsequent motion block.
1. Select the Drive Based Synchronous application wizard > Motion block operating mode > Subsequent motion block.
 2. Number:
Use the arrow keys to select the number of the motion block to which you would like to add a subsequent motion block.
 3. J281 Next motion block A at end:
If the subsequent motion block is to be executed once the motion block selected in Number has been completed, select the subsequent motion block here.
 4. J251 Source trigger next motion block A:
Select the trigger that activates continuation to the subsequent motion block.
 5. Click on the Trigger: Source link.
⇒ The Trigger: Data source wizard opens.
 6. J450[0] Source trigger:
Select the source of the trigger signal used to start the subsequent motion block.
 - 6.1. If you would like the subsequent motion block to be activated directly at the end of the previous motion block, select 1: High.
 - 6.2. If a digital input (direct or inverted) acts as the source, select the corresponding input.
 - 6.3. If the subsequent motion block is to be executed when a configurable condition is fulfilled (indirect trigger), select 2: Parameter and proceed as described in the section **Parameterizing the indirect trigger**.

| |
|--------------------|
| Information |
|--------------------|

For subsequent motion blocks at the end of the motion block, the trigger condition is only evaluated at the end of the ongoing motion block by using J281 – J285. Since the evaluation only takes place at the motion block end, J21 Command Mode does not influence when the subsequent motion block is started. The subsequent motion block always starts at the motion block end if the trigger condition is active.

Parameterizing the subsequent motion block due to interruption

Use subsequent motion blocks due to interruption if you would like to interrupt the ongoing motion block when the trigger is activated and continue to the subsequent motion block.

The following section explains the process as an example, based on the parameterization of subsequent motion block A. For the parameterization of subsequent motion block B – E, proceed in the same manner.

- ✓ You have parameterized at least two motion blocks.
 - ✓ You have selected 0: Abort current command for the subsequent motion block in J21 Command Mode.
1. Select the Drive Based Synchronous application wizard > Motion block operating mode > Subsequent motion block.
 2. Number:
Use the arrow keys to select the number of the motion block to which you would like to add a subsequent motion block.
 3. J271 Next motion block A by interrupt:
If the subsequent motion block is to be executed as soon as the motion selected in Number is interrupted by an interrupt signal, select the subsequent motion block here.
 4. J251 Source trigger next motion block A:
Select the trigger that activates continuation to the subsequent motion block.
 5. Click on the Trigger: Source link.
⇒ The Trigger: Data source wizard opens.
 6. J450[0] Source trigger:
Select the source of the trigger signal that activates continuation to the subsequent motion block.
 - 6.1. If a digital input (direct or inverted) acts as the source, select the corresponding input.
 - 6.2. If the subsequent motion block is to be executed when a configurable condition is fulfilled (indirect trigger), select 2: Parameter and proceed as described in **Parameterizing the indirect trigger**.

Information

For subsequent motion blocks due to interruption, the trigger condition is evaluated while the previous motion is still running by using J271 – J275. J21 Command Mode controls when the triggered subsequent motion block is started. When 0: Abort current command is selected, the subsequent motion block is executed immediately after the trigger condition is active. When 1: Start when current command has finished is selected, the subsequent motion block is buffered during activation of the trigger condition, but not executed until the motion block end.

Information

For motion blocks with a final velocity, you can parameterize a special case: a buffered subsequent motion block with trigger condition. To do so, use a subsequent motion block due to interruption (J271 – J275) with J21 Command Mode = 1: Start when current command has finished.

When 1: Start when current command has finished is selected, the subsequent motion block is buffered during activation of the trigger condition, but not executed until the motion block end. This enables a fluid transition to the subsequent motion block, depending on a trigger condition.

Parameterizing the indirect trigger

Parameterize the trigger condition for motion block continuation. If you parameterize a subsequent motion block that is to be triggered at the end of the previous motion block, the condition for continuation must be fulfilled before the set position is reached.

- ✓ You have selected 2: Parameter for J450[0] Source trigger.
- 1. Select the Drive Based Synchronous application wizard > Motion block operating mode > Subsequent motion block > Indirect trigger: Data source.
- 2. J439[0] Source indirect trigger:
 - Activate the desired indirect trigger and enter the condition for continuation to subsequent motion block A.
 - 2.1. If a simple parameter is used as the condition for continuation, select the BOOL notation: `Coordinate`.
 - 2.2. If a certain bit of a simple parameter is used as the condition for continuation, select the bit notation: `Coordinate.Bit`.
 - 2.3. If an array or record parameter is used as the condition for continuation, select the array notation: `Coordinate[Element]`.
 - 2.4. If a certain bit of an array or record parameter is used as the condition for continuation, select the following notation: `Coordinate[Element].Bit`.

Information

Using J439, you can specify a parameter, a single bit or an element of an array parameter as the source for the indirect trigger. Possible sources for indirect triggers include individual bits of the status or control word for control via fieldbus or signals from additional functions such as motion block duration, comparator or cams.

| Notation | Input format | Example |
|-------------------|--------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| BOOL notation | <code>Coordinate</code> | J439[0] = I86 The subsequent motion block is only triggered if I86 In reference is fulfilled, i.e. if the axis is referenced. |
| Bit notation | <code>Coordinate.Bit</code> | J439[0] = J453.0 You use J452 FSM block duration and the triggering of the subsequent motion block is to be time-controlled. The information "Motion block duration is reached" is stored in bit 0 of J453 FSM end of block duration. |
| Array notation | <code>Coordinate[Element]</code> | J439[0] = I64[0] The subsequent motion block is to be triggered once the first cam position is reached. The information for the first cam is stored in array element 0 of I64 Cam result. |
| Combined notation | <code>Coordinate[Element].Bit</code> | – |

Tab. 4: Indirect trigger: Notations

5.8.2.4.1.2 Parameterizing the buffered subsequent motion block

In the *Subsequent motion block wizard*, you can define a buffered subsequent motion block for each motion block. This means that the subsequent motion block is already loaded while the previous motion block is still being executed. Continuation therefore takes place automatically at the end of the previous motion block, without a delay and without a trigger signal.

Information

Depending on the application case, you can either parameterize a regular subsequent motion block or a buffered subsequent motion block at the end of the previous motion block. The use of a regular subsequent motion block at the end rules out the simultaneous use of a buffered subsequent motion block for the same motion block, and vice versa. You can combine subsequent motion blocks through interruption with regular subsequent motion blocks at the end and with buffered subsequent motion blocks.

Parameterizing the buffered subsequent motion block

Use a buffered subsequent motion block if you are moving the axis in the previous motion block with a set position (MC_MoveAbsolute, MC_MoveRelative, MC_MoveAdditive) and would like to transition fluidly into the next motion upon reaching the set position with the final velocity.

- ✓ You have parameterized at least two motion blocks.
 - ✓ You are moving the axis in the previous motion block with MC_MoveAbsolute, MC_MoveRelative or MC_MoveAdditive.
 - ✓ You have selected 1: Start when current command has finished for the motion block intended as the subsequent motion block in J21 Command Mode.
1. Select the Drive Based Synchronous application wizard > Motion block operating mode > Subsequent motion block.
 2. Number:
Use the arrow keys to select the number of the motion block to which you would like to add a buffered subsequent motion block.
 3. J286 Buffered motion block:
Select the motion block that is to follow the motion block selected in Number without a delay or trigger signal.

Information

If you add a buffered subsequent motion block using J286 Buffered motion block, make sure that J281 Next motion block A at end up to J285 Next motion block E at end is set to -1: Inactive.

Information

For motion blocks with a final velocity, you can parameterize a special case: a buffered subsequent motion block with trigger condition. To do so, use a subsequent motion block due to interruption (J271 – J275) with J21 Command Mode = 1: Start when current command has finished.

When 1: Start when current command has finished is selected, the subsequent motion block is buffered during activation of the trigger condition, but not executed until the motion block end. This enables a fluid transition to the subsequent motion block, depending on a trigger condition.

5.8.2.4.2 Testing the motion block linking

Motion block control panel provides all functions that you need to test the motion block linking.

WARNING!

Injury to persons and material damage due to axis movement!

By activating the control panel, you have sole control over the movements of the axis using DriveControlSuite. If you are using a controller, it no longer monitors the axis movements after the control panel is activated. The controller cannot intervene to prevent collisions. The controller takes over control again when the control panel is deactivated, which can cause unexpected axis movements.

- Only use the control panel if you have visual contact with the axis.
- Make sure that no people or objects are within the travel range.

-
- ✓ The drive controller is switched on.
 - ✓ You have established an online connection to the drive controller.
 - ✓ You have parameterized at least one motion block.
1. Select the Drive Based synchronous application wizard > Motion block operating mode > Motion block control panel.
 2. Click Control panel on and then Enable.
 - ⇒ The drive is controlled using the active control panel.
 3. J311 Motion block selection:
Select the motion block whose linking with the subsequent motion block you would like to check.
 4. Test the positions, velocities, accelerations and brakes and optimize your parameterization based on your test results if necessary.
 5. Repeat the tests for each additional motion block for which you have defined subsequent motion blocks.
 6. In order to deactivate the control panel, click Control panel off.

5.8.2.5 Parameterizing the type of motion block

Define how the motion blocks are to be started. Motion block operating mode offers the option to parameterize the direct start of up to three motion blocks without a controller. If you would like to use a controller, parameterize the start signal by defining the source for the execute signal and the source for the selection of the motion block to be started.

5.8.2.5.1 Parameterizing the direct motion block type

Motion block operating mode offers the option to parameterize the direct start of up to three motion blocks of your choice without a controller.

1. Select the Drive Based Synchronous application wizard > Motion block operating mode > Motion block > Direct motion block start: Data source.
2. J80 Activation direct motion block start:
In order to activate the direct motion block start, select 1: Active.
3. J87 Selected direct motion block start 1:
Enter the number of the motion block that you would like to start directly.
4. J81 Source direct motion block start 1:
 - 4.1. If a digital input (direct or inverted) acts as the source, select the corresponding input.
 - 4.2. If you would like to activate the motion block directly, select 2: Parameter J84.
5. J84 Direct motion block start 1:
If you have selected 2: Parameter J84 for J81 and want to start the motion block directly from DriveControlSuite, select 1: High.
6. Configure the direct start of a second and third motion block in the same way as the first motion block, if necessary.
 - 6.1. If you want to parameterize the direct start for a second motion block, repeat the steps in the same way for J88, J82 and J85.
 - 6.2. If you want to parameterize the direct start for a third motion block, repeat the steps in the same way for J89, J83 and J86.

Information

If you have configured the direct start of multiple motion blocks, they are activated with a simultaneous start signal according to your priority: J83 Source direct motion block start 3 > J82 Source direct motion block start 2 > J81 Source direct motion block start 1.

5.8.2.5.2 Parameterizing the execute signal

If you want to use a controller and start the motion block using an execute signal, define the source for the execute signal and the source for the selection of the motion block that is to be started by the execute signal.

Defining a source for the execute signal

1. Select the Drive Based Synchronous application wizard > Data sources > Application digital signals: Data source.
2. I100 Source execute:

To start a motion block via a controller, select the source for the execute signal.

 - 2.1. If a digital input (direct or inverted) acts as the source, select the corresponding input.
 - 2.2. If bit 0 of the control word I210 of the application is the source, select 2: Parameter.

Source for motion block selection

1. Select the Drive Based Synchronous application wizard > Motion block operating mode > Motion block > Motion block selection: Data source.
2. J03 Source motion block selection:

To be able to select a motion block via a controller, select the source for the motion block selection signal.

 - 2.1. If the control word J02 of motion block operating mode acts as the data source, select 1: Parameter J02 and enter the number of the motion block to be started in the event of an execute signal in J02Reference motion block.
 - 2.2. If digital inputs act as the source, select 2: Digital inputs and enter the digital inputs (direct or inverted) that activate a controller or binary switch.

5.8.2.6 Parameterizing the motion block end control unit

The motion block end control unit allows you to monitor the status of up to 5 freely selectable motion blocks, making it easier for you to keep track of the motion block linking.

1. Select the Drive Based Synchronous application wizard > Motion block operating mode > Motion block end control unit.
2. Activate the desired number of motion blocks to be monitored using the respective option.

⇒ Parameter J455[0] – [4] can be parameterized according to your selection.
3. J455 Move reference block:

Select the number of the motion block whose status you wish to monitor.

⇒ The state of the selected reference motion blocks can be observed in J457[0] – [4].

Information

The motion block end control unit indicates that the end of motion block has been reached (J457 = 1: High) as soon as the motion core has completed the motion command to be executed (I92 = 1: Active) and the actual motion block matches the reference motion block (J300 = J455).

If J455 is subsequently changed, the motion block chain must first be run through completely before J457 displays the correct state. If a motion block is executed several times, J457 is reset to 0: Low until the motion command is ended again.

5.8.2.7 Parameterizing the continue, reset and halt signal

To use the continue function to resume an interrupted motion block, configure the sources of the continue and reset signals. If you would like to use the motion command MC_Halt to stop a motion block during its run, also add the associated signal source here.

You can find more information on the continue function, continuable motion commands and reasons for canceling under [Continue function \[► 152\]](#).

1. **Select the Drive Based Synchronous application wizard > Motion block operating mode > Motion block > Operating mode digital signals: Data source.**
2. **J917 Source continue:**

To use the continue function to resume interrupted motion commands, select the source for the continue signal.

 - 2.1. If a digital input (direct or inverted) acts as the source, select the corresponding input.
 - 2.2. If the control byte J01 of the application acts as the source, select 2: Parameter J01.
⇒ Bit 2 of the control byte is set as the source.
3. **J07 Source reset motion block:**

To delete a canceled motion block from the continue memory and enable a different motion block start, enter the source for the reset signal.

 - 3.1. If a digital input (direct or inverted) acts as the source, select the corresponding input.
 - 3.2. If the control word J01 is used as the source for the motion block operating mode, select 2: Parameter J01.
⇒ Bit 1 of the control word is set as the data source.
4. **J06 Source MC_Halt motion block:**

To be able to stop a motion block in general, select the source for the stop signal.

 - 4.1. If a digital input (direct or inverted) acts as the source, select the corresponding input.
 - 4.2. If the control word J01 is used as the source for the motion block operating mode, select 2: Parameter J01.
⇒ Bit 0 of the control word is set as the data source.

5.8.3 Parameterizing command operating mode

The following graphic shows the signal flows of the command operating mode. The elements shown in a lighter shade are optional.

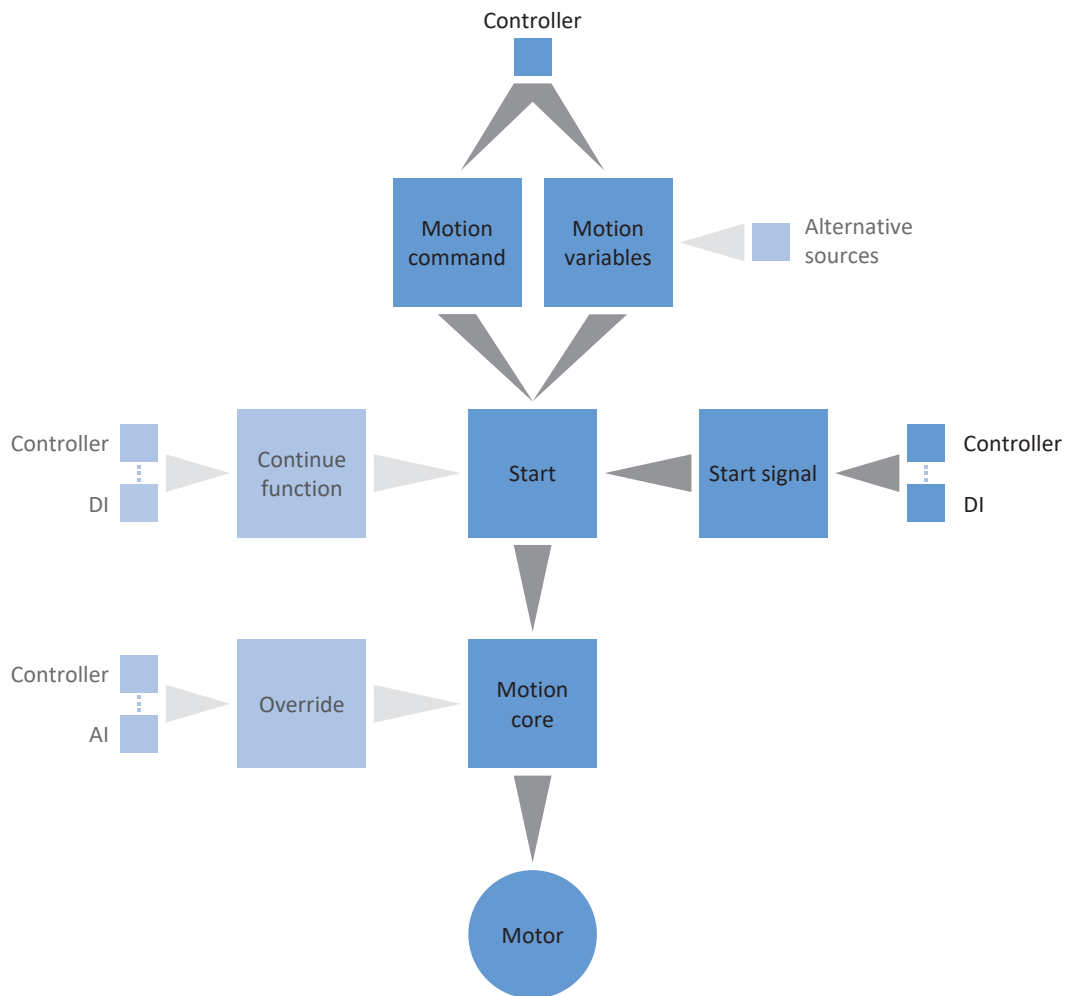


Fig. 4: Command operating mode: signal flows

In command operating mode, motion profiles are sent from a controller to the drive controller in the form of motion tasks. A motion task includes a PLCopen-compliant motion command which defines the basic motion method of the axis. The associated motion variables supply specifications on velocity, acceleration, deceleration and jerk, from which the motion core of the drive controller calculates the motion profile.

The controller coordinates the time progression and selects the motion tasks for the drive controller that the positioning, velocity and torque/force commands process and run. In the controller, you define the motion tasks. In the DS6, you parameterize the signal sources from which the drive controller receives the motion tasks from the controller.

In command operating mode, drive controllers are either activated using a fieldbus or by a mixed operation of a fieldbus and terminals.

Proceed as follows ...

- Activate command operating mode.
- Optional: Limit torque/force using the operating mode.
- Parameterize the command-specific motion variables.
- Define the source for the start signal.
- Optional: Define the source for the continue signal.

| |
|--------------------|
| Information |
|--------------------|

The motion variables available for this operating mode that you configure in the software depend on your application case as well as on additional project-specific factors such as the use of a controller or the type of data transmission (fieldbus, terminals).

5.8.3.1 Activating command operating mode

1. Select the relevant drive controller in the project tree and click on the desired projected axis in the Project menu > Wizard area.
 2. Select the Drive Based Synchronous application wizard.
 3. Operating modes area:
Activate the Command operating mode option.
- ⇒ The corresponding wizards are displayed.

5.8.3.2 Limiting the torque/force using the operating mode

Define whether you also want to limit torque/force using the operating mode. Limiting using the mechanical system takes place in the project configuration of the axis model. An additional limit using the operating mode is optional.

1. Select the Drive Based Synchronous application wizard > Data sources > Torque/force limit: Data source.
2. C132 Source maximum positiv T/F, C133 Source maximum negative T/F:
Select the source for the torque/force limit.
 - 2.1. If an analog input acts as the source, select the corresponding input.
 - 2.2. If the limiting values are transmitted over a fieldbus, select 4: Parameter C232 or 4: Parameter C233.
3. If the limiting values are transmitted over a fieldbus, select Drive Based Synchronous application > Command operating mode wizard.
4. J47 Maximum positive torque/force, J48 Maximum negative torque/force:
Enter the maximum permitted positive and negative values for torque/force.

| |
|--------------------|
| Information |
|--------------------|

The mechanical limit for torque/force is executed by C03 Maximum positive torque/force and C05 Maximum negative torque/force using the axis model. The additional limit using the operating mode is optional and refers to the values that you defined in the axis model.

| |
|--------------------|
| Information |
|--------------------|

If an analog input serves as the source for the application, parameterize, calibrate and scale the respective analog input as described in [Parameterizing analog inputs](#) [▶ 56].

5.8.3.3 Parameterizing command-specific motion variables

The parameters for the acceleration, deceleration and jerk motion variables are not included as standard in the process data mapping. You can either store the values for the motion variables permanently on the drive controller or add the corresponding parameters to the process data mapping in order to receive the values from the controller.

Information

Before you start parameterizing the operating mode-specific motion variables, parameterize the general motion variables and signal sources. If you obtain set values for velocity, velocity override or torque/force for your drive project from external sources, proceed as described in the chapter [General motion variables and signal sources](#) [► 52].

5.8.3.3.1 Storing motion variables: Drive controller

If you want to store the specifications for acceleration, deceleration and jerk permanently on the drive controller, check the default values in the *Command operating mode wizard* and, if necessary, adjust them to your drive project.

- ✓ You have activated command operating mode.
 - ✓ Parameters J44, J45 and J46 are not part of the process data mapping.
 - 1. Select the *Drive Based Synchronous application wizard* > *Command operating mode*.
 - 2. J44 Acceleration, J45 Deceleration, J46 Jerk:
Define the set values for acceleration, deceleration and jerk.
- ⇒ The specifications for acceleration, deceleration and jerk are permanently stored on the drive controller.

Information

The parameters in the *Command operating mode wizard* are written by the controller once there is an online connection between the drive controller and controller. In this state, you can read off the values in the parameters that the drive controller receives from the controller. Using parameter J41 Motion-ID, you can identify the ongoing motion task in order to assign the status bit correctly.

5.8.3.3.2 Storing motion variables: Process data mapping

If you want to specify the settings for acceleration, deceleration and jerk via a controller, add the required parameters to both the receive process data of the drive controller and the transmit process data of the controller.

EtherCAT: Supplementing motion variables in PDO mapping

If you want to specify values for motion variables using a controller, supplement the desired parameters in the receive process data of the drive controller.

✓ You have activated command operating mode.

1. Select the EtherCAT wizard > Received process data RxPDO.

2. Coordinate column:

Enter the coordinate of the parameter that you want to apply to the process data mapping in the Coordinate column.

2.1. If you want to obtain the values for acceleration from the controller, enter J44.

2.2. If you want to obtain the values for deceleration from the controller, enter J45.

2.3. If you want to obtain the values for jerk from the controller, enter J46.

⇒ The parameter is applied to the receive process data of the drive controller.

In the Name, Data type and Length columns, your information about the parameters is displayed.

3. Add your changes to the process data mapping also to the transmit process data of the controller.

PROFINET: Supplementing motion variables in PZD mapping

If you want to specify values for motion variables using a controller, supplement the desired parameters in the receive process data of the drive controller.

✓ You have activated command operating mode.

1. Select the PROFINET wizard > Received process data RxPZD.

2. Coordinate column:

Enter the coordinate of the parameter that you want to apply to the process data mapping in the Coordinate column.

2.1. If you want to obtain the values for acceleration from the controller, enter J44.

2.2. If you want to obtain the values for deceleration from the controller, enter J45.

2.3. If you want to obtain the values for jerk from the controller, enter J46.

⇒ The parameter is applied to the receive process data of the drive controller.

In the Name, Data type and Length columns, your information about the parameters is displayed.

3. Add your changes to the process data mapping also to the transmit process data of the controller.

5.8.3.4 Parameterizing the start signal

Define the source for the execute signal in order to start a motion command selected in J40 in command operating mode.

1. Select the Drive Based Synchronous application > Data sources > Application digital signals: Data source.
2. I100 Source execute:
 - 2.1. If a digital input (direct or inverted) acts as the source, select the corresponding input.
 - 2.2. If the control word I210 of the application acts as the source, select 2: Parameter.
⇒ Bit 0 of the control word is set as the data source.

5.8.3.5 Parameterizing the continue signal

To use the continue function to resume an interrupted movement command, configure the source of the continue signal. If you do not use the continue function, select a new command using J40 and start it using the start signal (execute).

You can find more information on the continue function, continuable motion commands and reasons for canceling under [Continue function \[► 152\]](#).

1. Select the Drive Based Synchronous application > Command operating mode > Operating mode digital signals: Data source.
2. J38 Source continue:

To use the continue function to resume interrupted motion commands, select the source for the continue signal.

 - 2.1. If a digital input (direct or inverted) acts as the source, select the corresponding input.
 - 2.2. If the control byte J37 for the command operating mode acts as the source, select 2: Parameter.
⇒ Bit 0 of the control byte is set as the source.

6 More on Drive Based Synchronous?

The following chapters summarize the important terms, modules and relationships concerning Drive Based Synchronous.

6.1 Drive Based Synchronous – Concept

Applications such as Drive Based Synchronous, which calculate and adjust the motions in the drive itself, are referred to as drive-based systems. They are either networked via a fieldbus or obtain signals and set values using analog and digital hardware inputs.

The Drive Based Synchronous application provides you with a standard set of motion commands based on PLCopen, which is supplemented by your own motion commands and thus offers flexible drive-based motion control for the position, velocity and torque/force control types.

The motion commands are combined into corresponding operating modes for different applications. Depending on the selected operating mode, the motion variables required for parameterizing individual motion profiles, such as set or limit values, are recorded.

In addition, the Drive Based Synchronous application provides you with various additional functions for different applications, such as jerk limitation, motion block linking, cam control unit or PID controller.

Various technological functions are available for the synchronous operation of multiple axes (multi-axis movements). As a rule, a master specifies set values which the coupled slave axes follow according to the specified proportional ratio. For example, a real axis, master encoder or virtual master can serve as the master.

Based on all parameterized data, the motion core finally calculates an individual motion profile. Specified set values are transmitted to the control cascade, which in turn activates the motor.

Additional wizards such as control panels or jog mode help with commissioning and the configuration test or are designed for emergency operation.

The following graphic shows the components and configuration steps for the Drive Based Synchronous application. The elements shown in a lighter shade are optional.

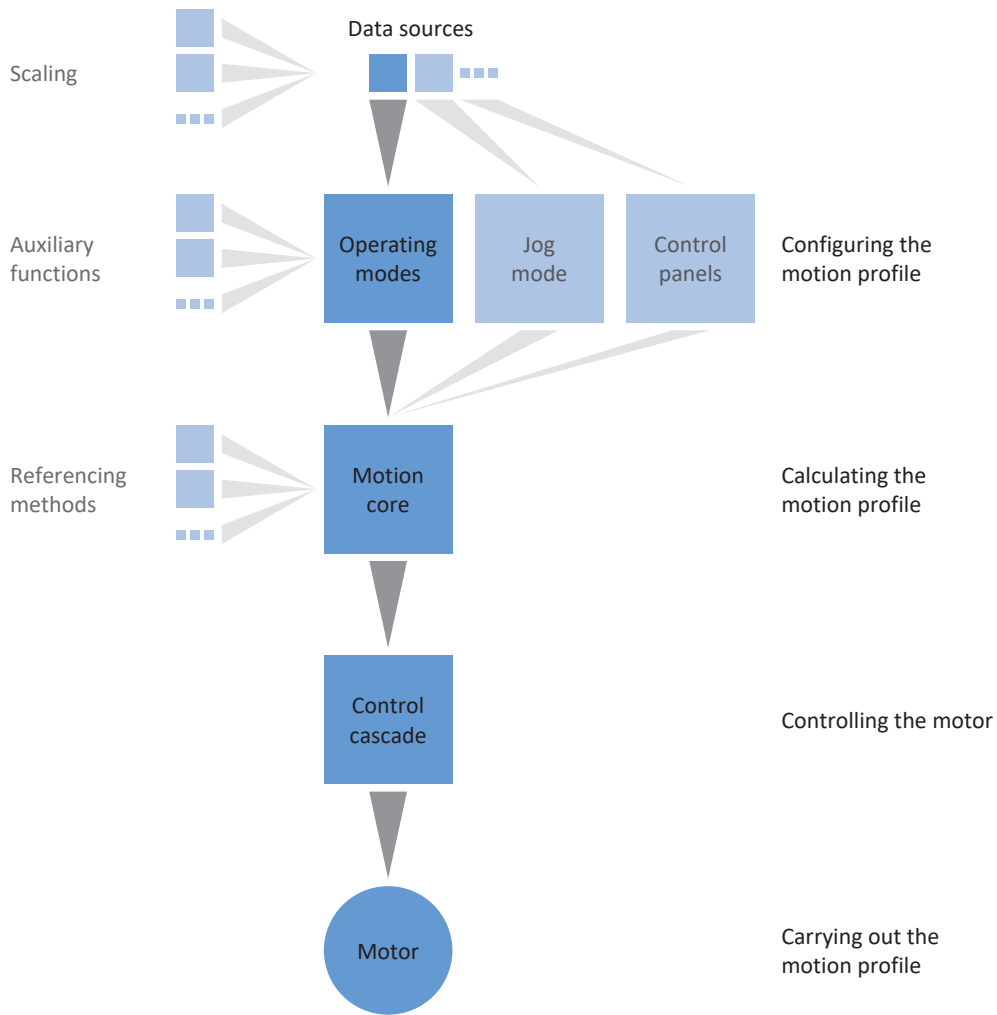


Fig. 5: Components and configuration steps

6.1.1 Operating modes

The operating modes of the Drive Based Synchronous application involve command sets bundled by application for configuring individual motion profiles for production operation.

The drive functions of these operating modes span from simple functions like position, velocity and torque/force control types to technology functions for individual axis applications to a synchronous operation of multi-axis systems.

Thanks to flexible, user-defined adjustment options for the operating modes, the Drive Based Synchronous application offers a compact – yet extremely variable – scope of functions that is suitable for a wide range of application cases in drive technology. After selecting the suitable operating mode for the respective application case, users concentrate just on parameterizing the most important sequences for their application.

Note that a drive controller can only process one operating mode per axis.

6.1.1.1 Motion block

In motion block operating mode, motion properties are predefined in the form of motion blocks. Special linking options allow for complete motion sequences to be set up that make it possible to execute sequences of motions quickly – regardless of whether an external controller is connected or the motion sequences are executed through digital signals.

Typical application examples are small machine automations or machine modules with predefined motion sequences.

Motion block linking

You can define one or more possible subsequent motion blocks for each motion block. Continuation to the subsequent motion block either happens automatically at the end of the motion block or is caused by a trigger. The trigger can either be a direct trigger signal or a trigger condition that is configurable in the software (indirect trigger).

For subsequent motion blocks with triggers, a distinction is made between subsequent motion blocks that are triggered at the end of the previous motion block and subsequent motion blocks that interrupt the previous motion block when an interrupt signal is triggered. The continuation can be executed by a direct or indirect trigger in either case.

Subsequent motion blocks at the end of a motion block are triggered as soon as the set value of a finite motion command is reached and the trigger is active. For subsequent motion blocks due to interruption, a finite or endless motion is interrupted and continued in the subsequent motion block as soon as the trigger is active. For example, the ongoing motion can be interrupted when a sensor or light barrier is triggered and the axis stopped.

Buffered subsequent motion blocks are parameterized without triggers and are intended for the case where the axis is moved to a set position in the previous motion block that is reached with the final velocity. In order to continue the motion fluidly and start the next motion command without a delay or standstill, the subsequent motion block is loaded in advance (buffered) while the previous motion block is still being executed. The continuation happens automatically at the end of the previous motions block.

6.1.1.2 Command

Command operating mode enables a drive controller to run parameterizable motions. In this process, a range of motion commands are processed that correspond to the behavior of the motion control blocks of the PLCopen standard.

A controller coordinates the time sequences by selecting motion commands based on PLCopen, such as MC_MoveAbsolute (move to absolute set position) or MC_MoveRelative (move relative to actual position).

Parameters such as set position, velocity or torque limit can be defined individually.

Typical application examples are individual axis movements that are communicated by a controller (PLC) to the drive controller.

6.1.2 Jog mode

For commissioning, emergency operation and maintenance or repair work, the manufacturer-specific jog mode (manual movement) is available. Using jog mode, you can move the drive independently of the controller, for example.

You can use jog mode either through the operating unit of the SD6 drive controller, the jog control panel or a controller that applies the manual movement.

6.1.3 Control panels

The control panels are special wizards in DriveControlSuite, which you can use to take control of the axis. With the help of the control panels, you can manually release and move an axis even if the drive controller has no operating unit or is difficult to access.

Using control panels, you can check the connection wiring, the project configuration of your physical axis model or the parameterization of your application in the respective operating mode before you switch to normal operation.

The following control panels are available:

- Jog control panel is used to check the projected axis model in jog mode.
- Control panel motion provides you with a standard set of motion commands based on PLCopen. Via the control panel, independent of the application and fieldbus interface, you can parameterize a motion profile directly for the motion core of the axis in order to check the basic functions of the drive controller.
- Motion block control panel is used to check the parameterized motion block operating mode and provides you with all the functions you need to test the parameterization of individual motion blocks or motion block linking.
- Jog control panel: Virtual master enables you to move the virtual master in jog mode to test synchronous operation.
- Control panel motion: Virtual master enables you to move the virtual master via PLCopen compliant motion commands to test synchronous operation.

Because control panels override normal operation, they can be activated only with a deactivated enable signal and should be operated by experienced users only.

6.1.4 Motion core

Based on the projected and parameterized data, the motion core calculates a motion profile and associated detail movements as a basis for the drive controller as well as mandatory set values for the control cascade.

6.1.5 Data sources

The signals that activate a drive controller can be obtained from different data sources.

Each of the possible operating modes has a set of set values that is permanently stored in the software. Furthermore, signals exist for the motion start or motion limits or velocity specifications that generally come from external data sources.

A fieldbus commonly acts as the signal source, though analog or digital hardware inputs or even a mixed operation made up of the above-listed data sources are also possible.

Values that are obtained from external data sources are usually adapted to the stored reference values automatically, i.e. calibrated and calculated with scaling.

6.1.5.1 Analog inputs

The analog inputs of the drive controller can serve as sources for set and actual values of the application in your drive project if, for example, you use sensors to measure motion variables such as velocities or torques/forces.

The available connections vary depending on the series and, if applicable, the terminal module of the drive controller.

For more information on the available connections, refer to the manual of the respective drive controller (see [Detailed information \[▶ 157\]](#)).

Information

If an analog input serves as the source for the application, parameterize, calibrate and scale the respective analog input as described in [Parameterizing analog inputs \[▶ 56\]](#).

Analog input AI1

Depending on whether analog input AI1 is used as a voltage source or current source, the following continuous function charts serve as examples (operating mode: F116).

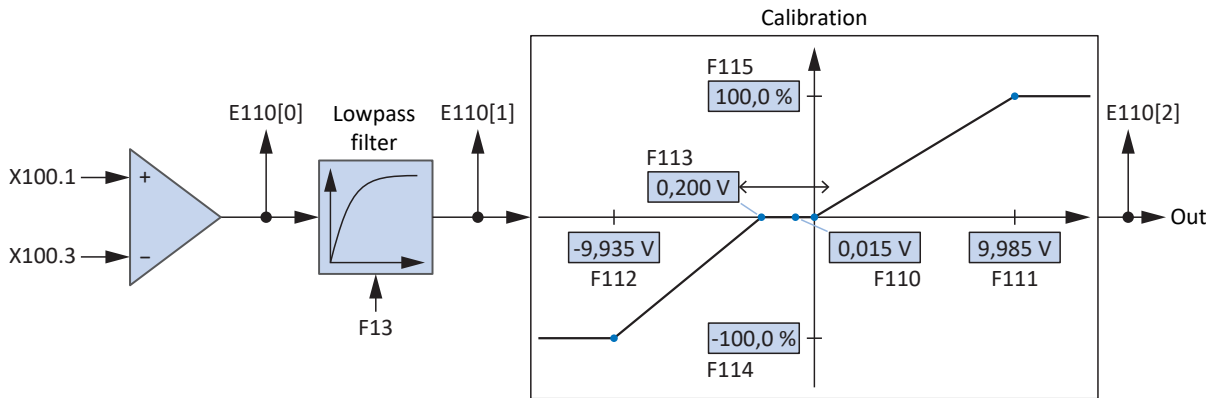


Fig. 6: Continuous Function Chart: analog input AI1 (F116 = 0: -10V to 10V)

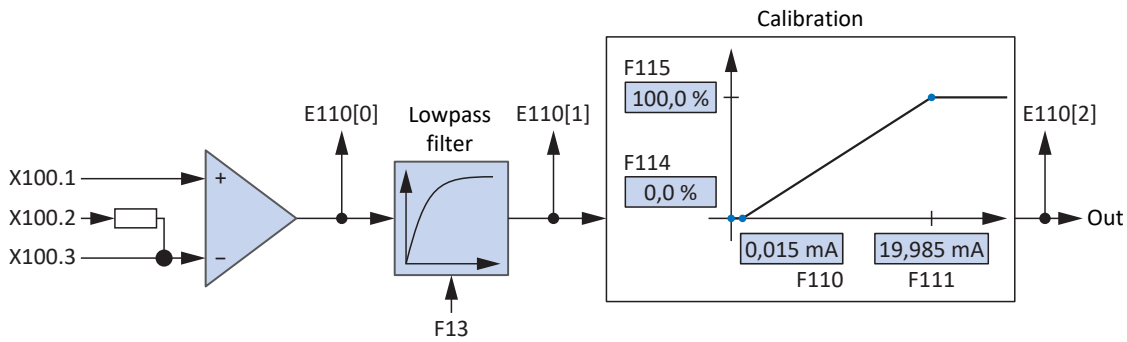


Fig. 7: Continuous Function Chart: analog input AI1 (F116 = 1: 0 to 20mA)

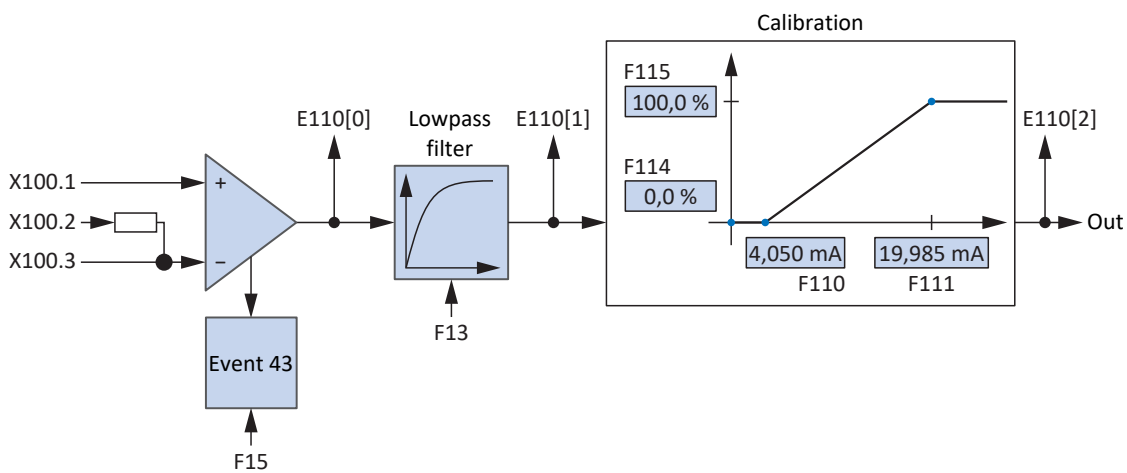


Fig. 8: Continuous Function Chart: analog input AI1 (F116 = 2: 4 to 20mA)

Analog input AI2

For analog input AI2 as a current source, the following continuous function chart serves as an example.

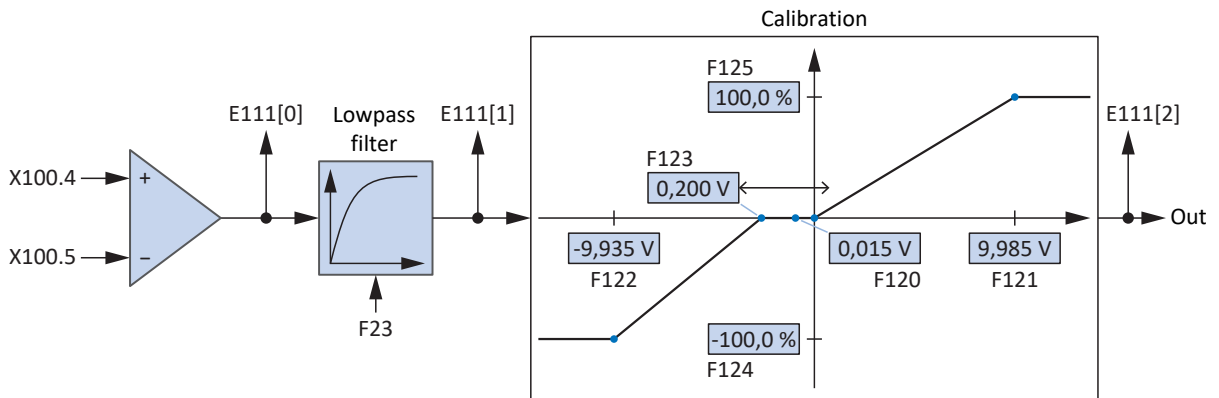


Fig. 9: Continuous Function Chart: analog input AI2

Analog input AI3

For analog input AI3 as a current source, the following continuous function chart serves as an example.

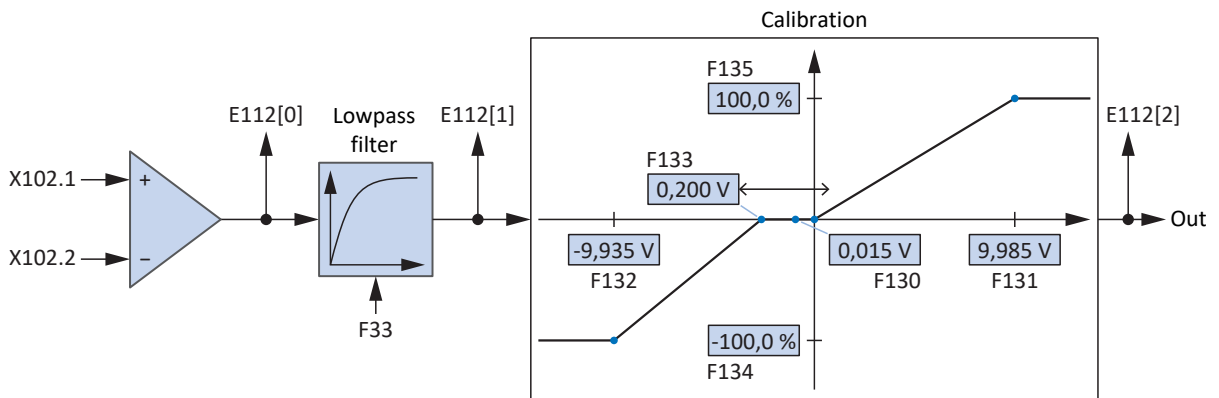


Fig. 10: Continuous Function Chart: analog input AI3

6.1.6 Additional functions

Each of the available operating modes can be expanded by drive-based additional functions. For example, these offer convenient monitoring of process variables such as position, velocity or torque/force (cams, comparators) or control of external process variables (PID controller).

6.1.6.1 Additional counter function

The additional counter function in Drive Based type applications provides you with up to 4 independent counters with which you can implement smaller automation tasks directly in the drive controller, such as direct or indirect control of the digital outputs.

Function

For each counter, you define a comparison value, a digital signal for increasing the counter reading and a digital signal for resetting the counter reading (comparison value: N41; increase source: N43; reset source: N46). With each rising edge of the counter signal, the counter reading is increased by 1 until the comparison value is reached (counter reading: N44). When the comparison value is reached, a status signal is issued and further counter signals are ignored until the counter reading is reset or the comparison value is increased (status: N42).

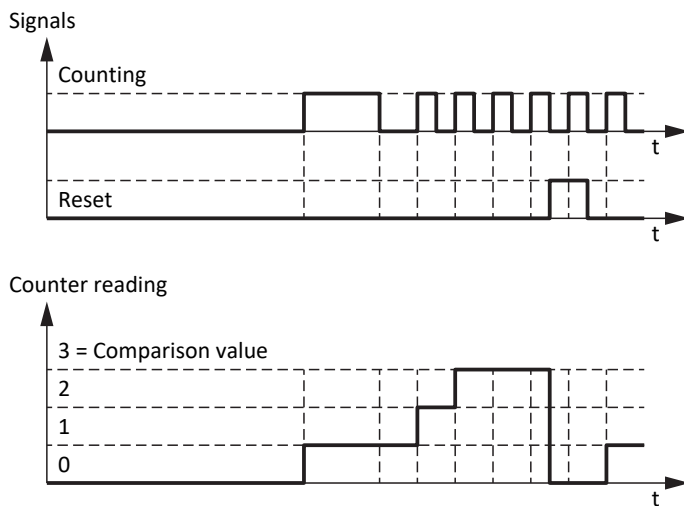


Fig. 11: Additional counter function: Example

The digital inputs of the drive controller (direct or inverted), parameters with the BOOL data type or individual bits of parameters with the BYTE, WORD or DWORD data type (example bit addressing: E49.4 for Switch on inhibit reason = STO) can serve as the source for the signals for increasing and resetting the counter status. The reset signal has priority and is executed immediately; as long as it is active, the counter reading remains at 0 and rising edges to increase the counter reading are ignored.

6.1.6.1.1 Parameterizing counters

To parameterize the additional counter function, proceed as described below.

1. Select the relevant drive controller in the project tree and click on the desired projected axis in the Project menu > Wizard area.
2. Select the Drive Based Synchronous application wizard > Additional functions.
3. Activate the Counter option.
 - ⇒ The additional function is activated and the associated wizards and parameters are displayed.
4. Select the Drive Based Synchronous application wizard > Additional functions > Counters.
5. Number:
 - Select the desired counter and activate it using the corresponding option.
 - ⇒ The associated parameters are displayed.
6. N41 Counter comparison value:
 - Define the comparison value.
7. N43 Source counter:
 - Select the source for the signal that increases the counter reading by 1 until the comparison value is reached.
 - 7.1. If a parameter is the source, select 2: Parameter.
 - 7.2. If a digital input (direct or inverted) acts as the source, select the corresponding input.
8. N45 Counter indirect reading:
 - If a parameter is used as the source for increasing the counter, define the desired coordinate, if necessary with bit addressing.
9. N46 Source reset counter reading:
 - Select the source for the signal that resets the counter reading to 0.
 - 9.1. If a parameter is the source, select 2: Parameter.
 - 9.2. If a digital input (direct or inverted) acts as the source, select the corresponding input.
10. N47 Reset counter reading via parameter:
 - If a parameter is used as the source for resetting the counter, define the desired coordinate, if necessary with bit addressing.

6.1.6.1.2 Parameterizing counters: Examples

For example, you can use the additional counter function to control the digital outputs directly or indirectly (using the additional INT32 comparator function) by reaching the comparison value.

In motion block operating mode, the additional counter function offers you options for automating motion block continuation, e.g. for repeating a motion block as often as required or interrupting a motion block after a certain number of events.

You can find more information on parameterizing motion blocks, subsequent motion blocks and triggers under [Parameterizing motion block operating mode \[▶ 60\]](#).

Controlling a digital output directly

To control a digital output directly via the counter, you can use the status of the counter as a source.

1. Activate the additional counter function.
 2. Select the **Counters wizard**.
 3. Parameterize the counter according to your application case.
 4. Select the status of the counter as the source for the desired digital output (e.g. DO1: F61 = N42).
- ⇒ The digital output is controlled directly via the status of the counter.

Controlling a digital output indirectly (INT32 comparator)

To indirectly control a digital output, you can use the additional counter and INT32 comparator functions in combination with each other. If you use the INT32 comparator to check whether the current counter reading fulfills a certain condition, you can control a digital output depending on the result of the comparator.

1. Activate the additional counter and comparator functions.
 2. Select the **Counters wizard**.
 3. Parameterize the counter according to your application case.
 4. Select the **INT32 comparators wizard**.
 5. Parameterize the INT32 comparator according to your application case.
 - 5.1. **C78 Source INT32 comparator:**
Select the counter reading N44 as the source for the INT32 comparator.
 6. Select the result of the INT32 comparator as the source for the desired digital output (e.g. DO1: F61 = C83).
- ⇒ The digital output is controlled indirectly via the counter reading.

Interrupting a motion block

To interrupt a motion block after a certain number of events, parameterize a motion block with an endless motion command and a subsequent motion block using interrupt with a motion command to stop an ongoing movement. The subsequent motion block is triggered by an indirect trigger, using the status of the counter as the source.

1. Activate the additional counter function.
 2. Select the **Counters wizard**.
 3. Parameterize the counter according to your application case.
 4. Select the **Motion block wizard**.
 5. Parameterize at least motion block 0 and motion block 1 according to your application.
 - 5.1. **J11[0] Command:**
Select for motion block 0 an endless motion command (e.g. 8: MC_MoveSpeed).
 - 5.2. **J11[1] Command:**
Select a motion command to stop for motion block 1 (e.g. 5: MC_Stop).
 6. For motion block 0, parameterize the motion block start (e.g. J80 = 1: Active; J81 = 3: DI1; J87 = 0).
 7. Select the **Subsequent motion block wizard**.
 8. **J271[0] Next motion block A by interrupt:**
Select motion block 1 as the subsequent motion block using interrupt for motion block 0.
 9. Select the **Trigger: Data source wizard**.
 10. **J450[0] Source trigger:**
To parameterize the status of the counter as an indirect trigger for the subsequent motion block, select 2: Parameter.
 11. Select the **Indirect trigger: Data source wizard**.
 12. **J439[0] Source indirect trigger:**
Activate the indirect trigger and define the status of the counter N42 as the source.
- ⇒ As soon as the counter has reached the comparison value, motion block 0 (8: MC_MoveSpeed) is interrupted by motion block 1 (5: MC_Stop) as a subsequent motion block by interrupt.

Repeating a motion block

To execute a motion block for a certain number of repetitions, parameterize a motion block with a finite motion command and a subsequent motion block at the end with the 0: MC_DoNothing motion command. The subsequent motion block is triggered by an indirect trigger using the status of the counter as a source as soon as the desired number of repetitions has been reached.

1. Activate the additional counter function.
2. Select the **Counters wizard**.
3. Parameterize the counter according to your application case.
 - 3.1. N41 Counter comparison value:
Define the desired number of motion block repetitions as the comparison value.
 - 3.2. N43 Source counter, N45 Counter indirect reading:
To count the number of times a motion block has been completed using the status byte of motion block operating mode, select N43 = 2: Parameter and N45 = J302.3.
4. Select the **Motion block wizard**.
5. Parameterize at least motion block 0 and motion block 1 according to your application.
 - 5.1. J11[0] Command:
Select a finite motion command for motion block 0 (e.g. 2: MC_MoveRelative).
 - 5.2. J11[1] Command:
Select motion command 0: MC_DoNothing for motion block 1.
6. For motion block 0, parameterize the motion block start (e.g. J80 = 1: Active; J81 = 3: DI1; J87 = 0).
7. Select the **Subsequent motion block wizard**.
8. To be able to both repeat and interrupt motion block 0, parameterize two subsequent motion blocks for motion block 0.
 - 8.1. J281[0] Next motion block A at end:
Select motion block 1 as the subsequent motion block at the end of motion block 0 (0: MC_DoNothing).
 - 8.2. J282[0] Next motion block B at end:
Select motion block 0 as the subsequent motion block at the end of motion block 0 (2: MC_MoveRelative).

⇒ Subsequent motion block A is executed with higher priority than subsequent motion block B if both subsequent motion blocks are triggered simultaneously.
9. Select the **Trigger: Data source wizard**.
10. J450[0] Source trigger:
To parameterize the status of the counter as an indirect trigger for subsequent motion block A, select 2: Parameter.
11. J450[1] Source trigger:
To automatically repeat motion block 0, select 1: High as the trigger for subsequent motion block B.
⇒ Subsequent motion block B is executed until subsequent motion block A is triggered (priority: J281 > J282).
12. Select the **Indirect trigger: Data source wizard**.
13. J439[0] Source indirect trigger:
Activate the indirect trigger and parameterize the status of the counter N42 as the source.
⇒ As soon as the counter has reached the desired number of repetitions, motion block 1 (0: MC_DoNothing) is executed at the end of motion block 0 (2: MC_MoveRelative) due to its higher priority.

6.1.6.2 Additional motor potentiometer function

In Drive Based Synchronous type applications, the additional motor potentiometer (MOP) function provides you with the option of adjusting an electromechanical potentiometer, for example for set value specification, for direct control of the analog outputs or for indirect control of the digital outputs.

Function

The motor potentiometer can be adjusted continuously via digital up and down signals, for example for specifying set values for axis movements via the output value (output value: G373). The output value can be limited by a maximum positive and a maximum negative value (limit: G362, G363; limit value reached: G374). Either the digital inputs of the drive controller or the graphical programming can serve as the source for the up and down signals (source: G364, G365). The initialization signal has priority over the up and down signals and is executed immediately; the initialization value is freely definable (source: G369; value: G366).

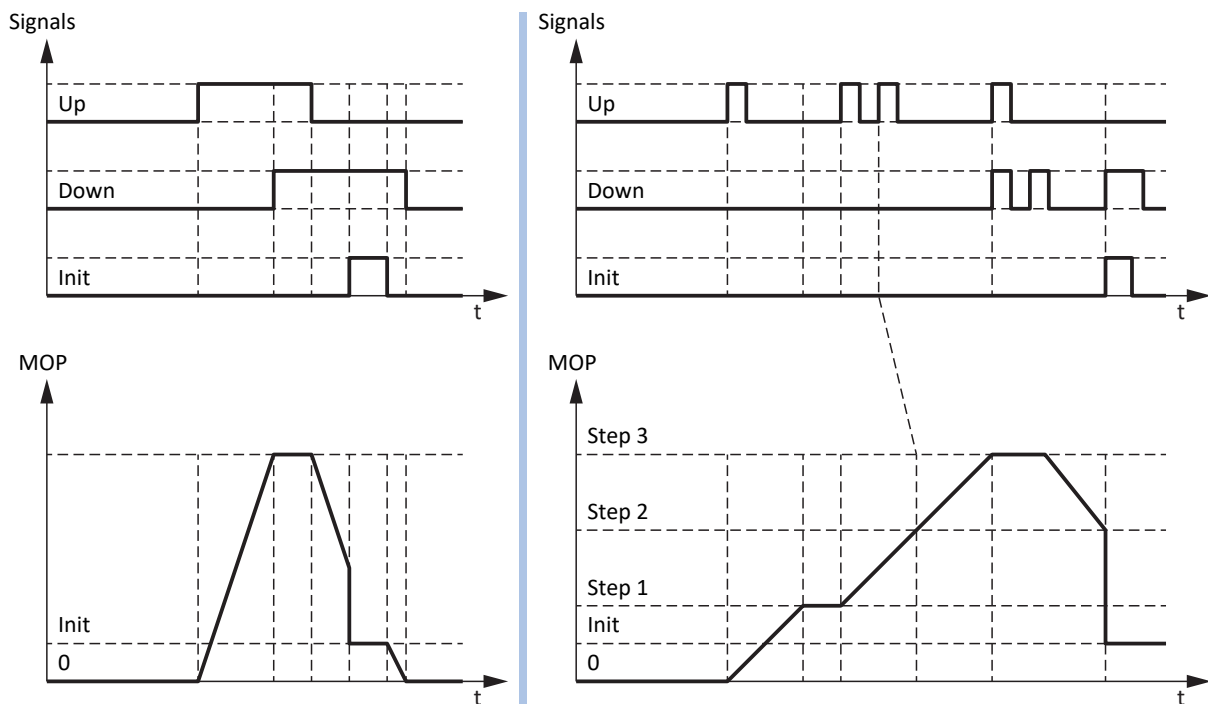


Fig. 12: Motor potentiometer: Linear and stepwise calculation

You can use the operating mode of the motor potentiometer to influence both the calculation of the output value and the storage behavior (operating mode: G368). The output value is calculated either linearly or in steps and can be saved either with an active enable signal, prior to the next restart of the drive controller or in non-volatile memory. The parameterized ramp (ramp: G361) applies to both the linear and stepwise calculations. With a linear calculation, the output value is changed for the duration of the incoming up or down signal. With a stepwise calculation, the output value is changed by the parameterized increment at signal input (increment: G367). If another up or down signal is received while the set value has not yet been reached, it is adjusted by the corresponding increment. If the up and down signals are active at the same time, the output value remains unchanged in both operating modes.

6.1.6.2.1 Parameterizing motor potentiometers

To parameterize the additional motor potentiometer function, proceed as described below.

1. Select the relevant drive controller in the project tree and click on the desired projected axis in the Project menu > Wizard area.
2. Select the Drive Based Synchronous application wizard > Additional functions.
3. Activate the Motor potentiometer option.
 - ⇒ The additional function is activated and the associated wizards and parameters are displayed.
4. Select the Drive Based Synchronous application wizard > Additional functions > Motor potentiometer.
5. G361 Motor potentiometer ramp:
Define the ramp for calculating the output value.
6. G362 Motor potentiometer positive maximum value, G363 Motor potentiometer negative maximum value:
Define the maximum permissible positive and negative output values of the motor potentiometer.
7. G364 Source motor potentiometer up, G365 Source motor potentiometer down:
Select the sources for the up and down signals of the motor potentiometer.
 - 7.1. If the graphical programming is used as the source, select 2: Parameter.
 - 7.2. If a digital input (direct or inverted) acts as the source, select the corresponding input.
8. G368 Motor potentiometer operating mode:
Select the desired operating mode for the motor potentiometer.
 - 8.1. To calculate the output value of the motor potentiometer linearly, select 0: Linear, 2: Linear (remanent) or 4: Linear (enable-dependent), depending on the desired storage behavior.
 - 8.2. To calculate the output value of the motor potentiometer in steps, select 1: Step-by-step, 3: Step-by-step (remanent) or 5: Step-by-step (enable-dependent), depending on the desired storage behavior.
9. G367 Motor potentiometer increment:
If you calculate the output value of the motor potentiometer in steps, define the desired increment.
10. G366 Motor potentiometer initialization value:
Optionally define the desired initialization value.
11. G369 Source motor potentiometer initialization:
Select the source for the initialization signal of the motor potentiometer.

6.1.6.2.2 Parameterizing motor potentiometers: Examples

For example, you can use the additional motor potentiometer function to specify set values for the velocity, velocity override or torque/force and thus control the axis directly in local operation, e.g. during commissioning, if the controller fails or if the machine has no controller at all. Using the motor potentiometer, you can also control the analog outputs directly or the digital outputs indirectly (using the additional REAL32 comparator function).

Controlling an analog output directly

To control an analog output directly, you can use the output value of the motor potentiometer as the source.

1. Activate the additional motor potentiometer function.
 2. Select the **Motor potentiometer wizard**.
 3. Parameterize the motor potentiometer according to your application case.
 4. Select the **Analog output 1** or **Analog output 2 wizard**.
 5. F40 AO1 source, F50 AO2 source:
Select the output value of the motor potentiometer G373 as the source for the desired analog output AO1 or AO2.
 6. F41 AO1 reference value source, F51 AO2 reference value source:
If necessary, delete the reference value of the desired analog output.
- ⇒ The analog output is controlled directly via the motor potentiometer.

Controlling a digital output indirectly (REAL32 comparator)

To indirectly control a digital output, you can use the additional motor potentiometer and INT32 comparator functions in combination with each other. If you use the REAL32 comparator to check whether the output value of the motor potentiometer fulfills a certain condition, you can control a digital output depending on the result of the comparator.

1. Activate the additional motor potentiometer and REAL32 comparator functions.
 2. Select the **Motor potentiometer wizard**.
 3. Parameterize the motor potentiometer according to your application case.
 4. Select the **REAL32 comparators wizard**.
 5. Parameterize the REAL32 comparator according to your application case.
 - 5.1. C65 Source REAL32 comparator:
Select the output value of the motor potentiometer G373 as the source for the REAL32 comparator.
 6. Select the result of the REAL32 comparator as the source for the desired digital output (e.g. DO1: F61 = C75).
- ⇒ The digital output is controlled indirectly via the motor potentiometer.

Specifying a set velocity (external velocity)

To specify a set velocity via the motor potentiometer, select its output value as the source for the external velocity and use this as the set value source for the motion command.

1. Activate the additional motor potentiometer function.
 2. Select the Motor potentiometer wizard.
 3. Parameterize the motor potentiometer according to your application case.
 4. Select the External velocity: Data source wizard.
 5. G461 Source external velocity:
Select 5: Indirect read parameter G811.
 6. G811 Indirect read external velocity:
Select the output value of the motor potentiometer G373 as the source for the external velocity.
 - ⇒ The output value of the motor potentiometer serves as the source for the external velocity.
 - ⇒ The external velocity can be provided as a set value specification by accessing parameter G462.
 7. For the motion block operating mode, parameterize the motion command and set value source via parameters J11 and J30.
 - 7.1. J11 Command:
Select the motion command 4: MC_MoveVelocity or 8: MC_MoveSpeed.
 - 7.2. J30 Source velocity:
Select 1: Parameter G462 as the source for the external velocity.
 8. For the command operating mode, parameterize the motion command and set value source via parameters J40 and J52.
 - 8.1. J40 Command:
Select the motion command 4: MC_MoveVelocity or 8: MC_MoveSpeed.
 - 8.2. J52 Source velocity 1:
Select 1: Parameter G462 as the source for the external velocity.
- ⇒ The output value of the motor potentiometer serves as the set value specification; the conversion to a set velocity is carried out using maximum permissible speed I10.

Specifying a set velocity (external additional velocity)

To additionally regulate the set velocity via the motor potentiometer, select its output value as the source for the external additional velocity and use this as the set value source for the motion command.

- ✓ You are using command operating mode or velocity, torque/force operating mode.
 - 1. Activate the additional motor potentiometer function.
 - 2. Select the **Motor potentiometer wizard**.
 - 3. Parameterize the motor potentiometer according to your application case.
 - 4. Select the **External additional velocity: Data source wizard**.
 - 5. **G464 Source external additional velocity:**
Select 5: Indirect read parameter G811.
 - 6. **G812 Indirect read additional external velocity:**
Select the output value of the motor potentiometer G373 as the source for the external additional velocity.
 - ⇒ The output value of the motor potentiometer serves as the source for the external additional velocity.
 - ⇒ The external velocity can be provided as a set value specification by accessing parameter G465.
 - 7. For the command operating mode, parameterize the motion command and set value source via parameters J40 and J54.
 - 7.1. **J40 Command:**
Select the motion command 4: MC_MoveVelocity or 8: MC_MoveSpeed.
 - 7.2. **J54 Source velocity 2:**
Select 1: Parameter G465 as the source for the external additional velocity.
- ⇒ The output value of the motor potentiometer serves as the set value specification; the conversion to a set velocity is carried out using maximum permissible speed I10.

Specifying the velocity override

To specify a velocity override via the motor potentiometer, select its output value as the set value source and apply the velocity override to the motion command.

1. Activate the additional motor potentiometer function.
 2. Select the **Motor potentiometer wizard**.
 3. Parameterize the motor potentiometer according to your application case.
 4. Select the **Velocity override: Data source wizard**.
 5. **G467 Source velocity override:**
Select 5: Indirect read parameter G813.
 6. **G813 Indirect read velocity override:**
Select the output value of the motor potentiometer G373 as the source for the velocity override.
 - ⇒ The output value of the motor potentiometer serves as the source for the velocity override.
 - ⇒ The velocity override can be provided as a set value specification by accessing parameter G468.
 7. For the motion block operating mode, parameterize the motion command and set value source via parameters J11 and J28.
 - 7.1. **J11 Command:**
Select the motion command 4: MC_MoveVelocity or 8: MC_MoveSpeed.
 - 7.2. **J28 Velocity override enable:**
To apply the velocity override to the desired motion block, select 1: Active.
 8. For the command operating mode, parameterize the motion command and set value source via parameters J40 and J51.
 - 8.1. **J40 Command:**
Select the motion command 4: MC_MoveVelocity or 8: MC_MoveSpeed.
 - 8.2. **J51 Source velocity override:**
Select 1: Parameter G468 as the source for the velocity override.
- ⇒ The output value of the motor potentiometer is used as the set value specification for the velocity override.

Specifying the set torque/set force

To specify a set torque/force via the motor potentiometer, select its output value as the set value source for the motion command.

1. Activate the additional motor potentiometer function.
 2. Select the Motor potentiometer wizard.
 3. Parameterize the motor potentiometer according to your application case.
 4. Select the Set torque/force, velocity bracketing: Data source wizard.
 5. G470 Source torque/force reference:
Select 5: Indirect read parameter G814.
 6. G814 Indirect read torque/force reference:
Select the output value of the motor potentiometer G373 as the source for the set torque/force.
 - ⇒ The output value of the motor potentiometer serves as the source for the set torque/force.
 - ⇒ The set torque/force can be provided as a set value specification by accessing parameter G471.
 7. For the motion block operating mode, parameterize the motion command and set value source via parameters J11 and J32.
 - 7.1. J11 Command:
Select the 9: MC_TorqueControl motion command.
 - 7.2. J32 Source reference torque/force:
Select 1: Parameter G471 as the source for the set torque/force.
 8. For the command operating mode, parameterize the motion command and set value source via parameter J40.
 - 8.1. J40 Command:
Select the 9: MC_TorqueControl motion command.
 - ⇒ The parameterized set value source is applied automatically; no further settings are necessary.
- ⇒ The output value of the motor potentiometer serves as the set value specification for the set torque/force.

6.1.6.3 Additional fieldbus-controlled analog output function

In Drive Based Synchronous type applications, the additional fieldbus-controlled analog output function allows you to control the respective analog output directly via a controller, e.g. for operating simple actuators such as pumps, fans, or valves. The additional function can be used to specify an individual set value for each analog output (enable: G303; set value: G306, G309). When fieldbus communication is active, the set value point is written by the controller via the process data mapping (G304, G307). In addition, a fallback value can be defined which supplies the set value at the respective analog output if fieldbus communication fails (G305, G308).

6.1.6.3.1 Parameterizing a fieldbus-controlled analogue output

To parameterize the additional function for controlling analog outputs via fieldbus, proceed as described below.

Information

To use the additional function for controlling the analog outputs via fieldbus, expand the process data mapping of the application, depending on the fieldbus used. Further information on process data mapping can be found in the relevant fieldbus manual.

Parameterizing fieldbus-controlled analogue output AO1

1. Select the relevant drive controller in the project tree and click on the desired projected axis in the Project menu > Wizard area.
2. Select the Drive Based Synchronous application wizard > Additional functions.
3. Activate the Fieldbus-controlled analog output 1 option.
⇒ The additional function is activated and the associated wizard is displayed.
4. Select the Drive Based Synchronous application wizard > Additional functions > Fieldbus-controlled analog output 1.
5. G305 Set value AO1 fallback:
Define the fallback value used to control the analog output if fieldbus communication fails.
6. Select the Terminals wizard > Analog output 1.
 - 6.1. F40 AO1 source:
For set value specification via fieldbus, select parameter G306 as the source for analog output AO1.
 - 6.2. F41 AO1 reference value source:
If necessary, delete the reference value of analog output AO1.
7. For fieldbus communication via CANopen, add parameter G304 to the receive process data RxPDO A225 – A228.
8. For fieldbus communication via EtherCAT, add parameter G304 to the receive process data RxPDO A225 – A228 and create a new ESI file for the controller if necessary.
9. For fieldbus communication via PROFINET, add parameter G304 to the receive process data RxPZD A90 – A91.

Parameterizing fieldbus-controlled analogue output AO2

1. Select the relevant drive controller in the project tree and click on the desired projected axis in the Project menu > Wizard area.
2. Select the Drive Based Synchronous application wizard > Additional functions.
3. Activate the Fieldbus-controlled analog output 2 option.
⇒ The additional function is activated and the associated wizard is displayed.
4. Select the Drive Based Synchronous application wizard > Additional functions > Fieldbus-controlled analog output 2.
5. G308 Set value AO2 fallback:
Define the fallback value used to control the analog output if fieldbus communication fails.
6. Select the Terminals wizard > Analog output 2.
 - 6.1. F50 AO2 source:
For set value specification via fieldbus, select parameter G309 as the source for analog output AO2.
 - 6.2. F51 AO2 reference value source:
If necessary, delete the reference value of analog output AO2.
7. For fieldbus communication via CANopen, add parameter G307 to the receive process data RXPDO A225 – A228.
8. For fieldbus communication via EtherCAT, add parameter G307 to the receive process data RXPDO A225 – A228 and create a new ESI file for the controller if necessary.
9. For fieldbus communication via PROFINET, add parameter G304 to the receive process data RxPZD A90 – A91.

6.1.6.4 Additional phasing function

In Drive Based Synchronous applications, the additional phasing function allows you to make corrections to the master position while the slave axis is coupled to the master. For example, the additional phasing function can be used if position-based coupling of the slave axis via 14: MC_GearInPosition is not possible or to correct the master value of slave axes if slipping occurs in the master.

Function

Phasing takes place within the slave axis. The slave axis receives the master position according to the selected master value source (source: G27). In order to influence the position of the slave axis, a phase shift to the received master position is specified by the set position of the phasing (set position: G223). The phase shift is added to the received master position (total master position: G80).

The slave axis attempts to compensate for the phase shift by accelerating or decelerating. The compensation movement has its own motion profile, which can be limited in terms of velocity, acceleration and jerk as well as permitted direction of motion (motion profile: G224 – G227; limitation: G70 – G73). The compensation movement can therefore be executed while the slave axis is coupled to the master.

Phasing can be started either via the digital inputs of the drive controller or via fieldbus (source: G230). The phase shift is maintained until a new set position is specified by another phasing command. The phase shift can be executed with an absolute set position or a set position relative to the existing phase shift, i.e. to the actual position of the phasing at the start of the command (motion command: G234; actual position: G214).

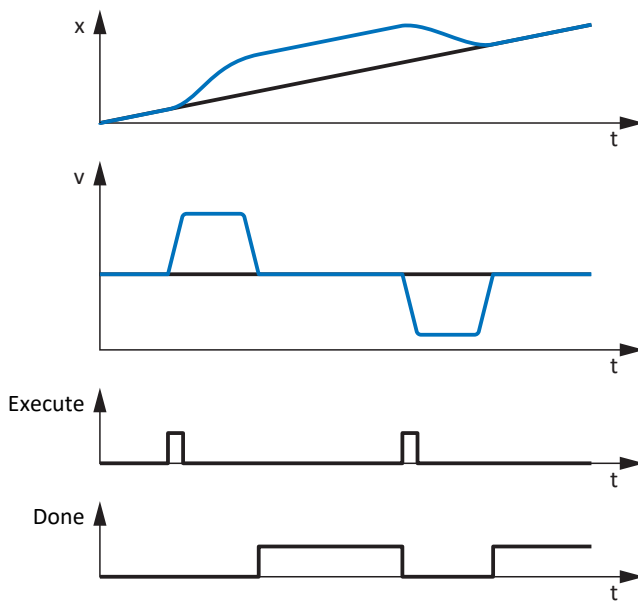


Fig. 13: Additional phasing function: Example

The example shows a phasing compensation movement with motion command G234 = 2: MC_MoveRelative. When the Execute signal is triggered, the slave axis accelerates or decelerates according to the parameterized motion profile in order to compensate for the phase shift specified via G223. The difference between the master position received (source: G27) and the master position G80 of the slave axis used for motion commands corresponds to G214.

To ensure that the axis interrupts the active phasing motion command during the compensation movement, a separate quick stop can be parameterized for the phasing (quick stop: G74, G216).

A preset function is available for phasing (preset: G78, G79, G240) in order to set the slave axis to a specific position immediately and without a motion profile when decoupled.

Application example

You are using a controller and operating the Drive Based Synchronous application in command operating mode. The master position is measured using a measuring wheel that is in frictional contact with a material web, i.e. a master encoder is acting as the master value source for the slave axes. There are print marks at regular intervals on the material web, e.g. plastic film, which can be measured using a sensor. If slipping occurs and the measuring wheel briefly loses frictional contact with the material web, the controller can detect this using the print marks and compensate for it with the help of phasing.

6.1.6.4.1 Parameterizing phasing

For phasing, parameterize the limiting values and the source for the Execute signal as described below. The parameterization of the quick stop and preset function is optional.

Information

To be able to control the additional phasing function via fieldbus, expand the process data mapping using the respective wizard (EtherCAT: Received process data RxPDO; PROFINET: Received process data RxPZD).

An overview of the phasing parameters that can be written by the controller via fieldbus is provided by the **Command operating mode: Phasing wizard**. Add the required parameters to both the receive process data of the drive controller and the transmit process data of the controller (motion command: G234; set value specifications: G223 – G227; control word: G10).

Parameterizing phasing

Parameterize the additional phasing function as described below.

- ✓ You have completed the parameterization of synchronous operation.
- 1. Select the relevant drive controller in the project tree and click on the desired projected axis in the Project menu > Wizard area.
- 2. Select the Drive Based Synchronous application wizard > Additional functions > Phasing > Limit: Phasing.
- 3. G73 Phasing permitted direction:
If you selected 1: Endless for G30 when configuring synchronous operation, define the permitted direction of motion for phasing.
 - 3.1. If motions in both directions are permitted, select 0: Positive and negative.
 - 3.2. If motions are only permitted in one direction, select 1: Positive or 2: Negative accordingly.
- 4. G70 Phasing maximum velocity:
Define the maximum permitted velocity for phasing.
- 5. G71 Phasing maximum acceleration:
Define the maximum permitted acceleration for phasing.
- 6. G72 Phasing maximum jerk:
Define the maximum permitted jerk for phasing.
- 7. Select the Drive Based Synchronous application wizard > Synchronous operation > Master digital signals: Data source.
- 8. G230 Source phasing execute:
Select the source for the digital signal that is used to start a motion command for phasing.
 - 8.1. If a control unit is used as the source, select 2: Parameter.
 - 8.2. If a digital input (direct or inverted) acts as the source, select the corresponding input.

Phasing: Parameterizing a quick stop

To interrupt an active phasing motion command with a quick stop, proceed as described below.

1. Select the relevant drive controller in the project tree and click on the desired projected axis in the Project menu > Wizard area.
2. Select the Drive Based Synchronous application wizard > Additional functions > Phasing.
3. G216 Phasing quick stop additional cause:
Select the device state of the drive controller that is to be an additional cause for triggering a quick stop during phasing.
 - 3.1. To trigger a quick stop when the axis has a fault, select 1: Error.
 - 3.2. To trigger a quick stop when the axis is not enabled, select 2: Enable.
 - 3.3. To trigger a quick stop when the axis is not enabled or has a fault, select 3: Enable or error.
4. Select the Drive Based Synchronous application wizard > Additional functions > Phasing > Limit: Phasing.
5. G74 Phasing quick stop deceleration:
Define the delay for the quick stop of the phasing.

Phasing: Parameterizing a preset

To use the phasing preset function, proceed as described below.



Injury to persons and material damage due to slave axis movement!

By activating this function, the master value changes immediately without a motion profile. As a result, coupled slave axes immediately follow the movement of the master.

- Only execute this function if there are no slave axes coupled with the master.

1. Select the relevant drive controller in the project tree and click on the desired projected axis in the Project menu > Wizard area.
2. Select the Drive Based Synchronous application wizard > Additional functions > Phasing.
3. G78 Phasing preset position:
Define the set position for the phasing preset.
4. Select the Drive Based Synchronous application wizard > Synchronous operation > Master digital signals: Data source.
5. G240 Source phasing preset:
Select the source for the digital signal that is used to set the axis to the preset phasing position immediately and without a motion profile.

6.2 Synchronous operation – Concept

In DriveControlSuite, you have various options for parameterizing synchronous operation, which differ in terms of the master value source and number of axes involved.

For example, you can use a virtual master as a master value source and provide the master value via the master axis model to optimize the synchronism of multiple axes (example: smooth motion of a platform). Using a master encoder as a master value source enables you to synchronize the motion of one or more axes with an upstream machine (example: synchronizing conveyor belts). Using the actual position of the master axis as a master value allows you to synchronize slave axes to a main axis.

6.2.1 Model 1: Virtual master or master encoder, multiple slave axes

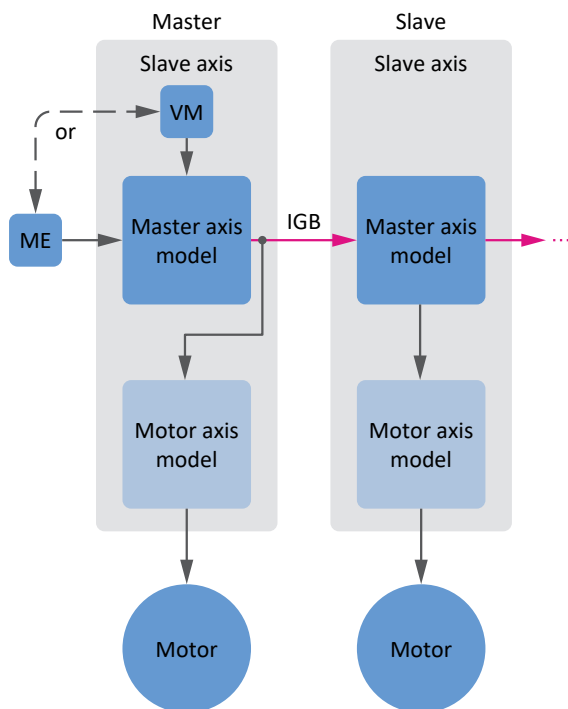


Fig. 14: Synchronous operation: Model 1

In this model, a virtual master or master encoder acts as the master value source. The master axis itself follows the master value as a slave via the IGB motion bus and makes it available to the master axis models of other slave axes across all devices.

This model for the configuration of synchronous operation allows you to synchronize the motion of multiple axes to a common master value of a virtual master or to the master encoder of an upstream machine, for example.

6.2.2 Model 2: Master encoder, individual slave axis

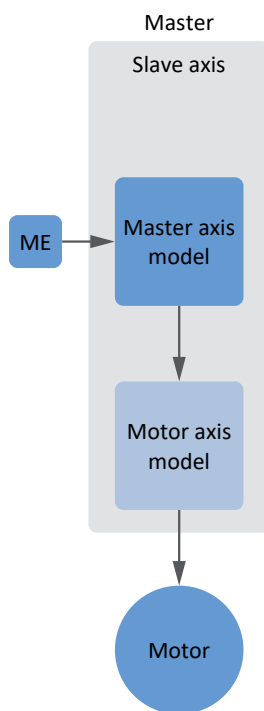


Fig. 15: Synchronous operation: Model 2

In this model, a master encoder acts as the master value source. An individual axis follows the master value that is provided by an internal device master axis model. There is no need to parameterize additional slave axes or an IGB motion bus network.

This model for the configuration of synchronous operation allows you to synchronize the motion of a single axis to the master encoder of an upstream machine, for example.

6.2.3 Model 3: Master axis, one or more slave axes

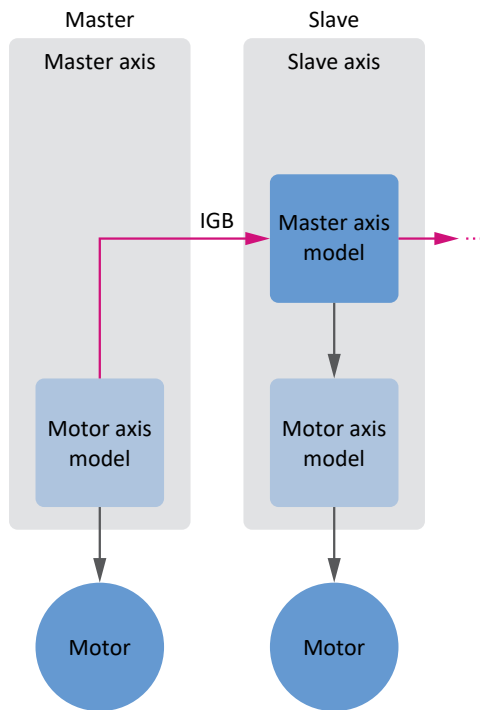


Fig. 16: Synchronous operation: Model 3

In this model, the actual position of the master axis acts as the master value. The master axis makes the master value available to the master axis models of other slave axes across all devices via the IGB motion bus.

This model for the configuration of synchronous operation allows you to synchronize slave axes to the actual position of the master axis, for example.

6.3 IGB motion bus network

The IGB motion bus transmits signals between SD6 series drive controllers. In an IGB motion bus network, data is synchronized in millisecond cycles and transferred cyclically, in real time, according to the producer/consumer network model.

Application cases

For synchronous operation, an IGB motion bus network transmits actual and set values between network nodes, such as the master value for the position or information about the velocities or torque/force of the individual axes.

System requirements

- At least 2 and up to 32 drive controllers can be networked with each other
- All network nodes must be connected with each other directly, without any intermediary hubs or switches
- The IGB motion bus network must follow a line topology
- The X3A interfaces may be connected only to the X3B interfaces of other drive controllers and vice versa
- The use of a suitable Ethernet cable is required for a functioning network
- The total length of the network must not exceed 100 m

Features

In the IGB motion bus network, you can network up to 32 drive controllers with each other, eliminating complicated fieldbus configuration.

In the role of a producer, any drive controller in an IGB motion bus network can place 32 bytes on the bus. Of those 32 bytes, 6 bytes are preassigned with system data that can be used at any time. The remaining 26 bytes can be assigned freely with PDO-capable parameters.

In the role of a consumer, any drive controller can receive up to 64 parameters from the IGB motion bus network from any other drive controller.

The IGB motion bus function of an IGB network is automatically active as soon as a drive controller is dragged and dropped into the newly created network when you are projecting an IGB motion bus network. Correct operation of an IGB motion bus is required for the device state machine to switch from the Switch on disabled state to the Ready to switch on state.

Each node of an IGB motion bus network requires a unique IGB motion bus address, which can be assigned automatically during network project configuration. The address can be edited manually. In addition, the number of drive controllers in the entire IGB motion bus group must be known to each drive controller of an IGB motion bus network. The associated IGB target number is automatically parameterized during IGB motion bus mapping and transferred to the individual drive controllers along with the overall configuration.

6.4 Axis model

With the help of the axis model, you map the real mechanical environment of your drive project in DriveControlSuite by parameterizing the axis type and the arrangement of existing encoders. The parameterization of the axis model is required for the smooth operation and easy diagnosis of your drive train.

With I05 Type of axis, you select a rotational or a translational axis model and define whether the scaling of the axis is done using either predefined or individually configured units of measure. With I00 Position range, you parameterize either an endless or a limited travel range. You parameterize the encoder arrangement by using B26 Motor encoder and I02 Position encoder.

The 6th generation of STOBER drive controllers are specially developed for communication between the drive controller and controller on the basis of the actual variables at the output (° or mm of real axis movement). The scaling of the axis model is calculated by the drive controller's firmware independently of the encoder model, without any rounding errors or drift.

If your axis model is not followed by any other gear ratios, you can operate the axis with output-side motion variables where all nominal and actual values correspond to the real axis movement.

Information

The firmware processes values for the motion variables velocity, acceleration and jerk in the data type REAL32 (floating-point number, 32 bits). Position values are processed in the INT32 data type (integer, 32 bits) to eliminate rounding errors and enable precise movements.

| Abbreviation | Meaning |
|--------------|------------------|
| LinM | Linear Motor |
| M | Motor |
| MEnc | Motor Encoder |
| PEnc | Position Encoder |

Rotational axis models

The following figures each show a rotational axis model consisting of a motor, a gear unit and a rotary table (endless rotational) or a pointer (limited rotational). Rotational axis models support rotational motor encoders as well as rotational position encoders.

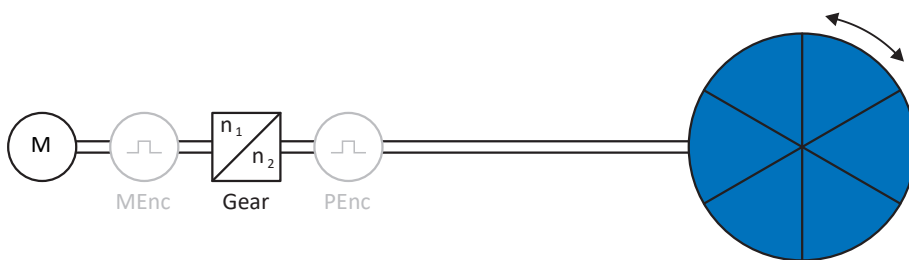


Fig. 17: Endless-rotational motion: Rotary table

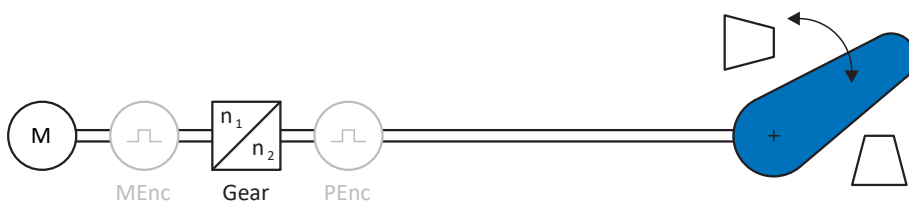


Fig. 18: Limited-rotational motion: Pointer

Translational axis models

The following figures each show a translational axis model consisting of a motor, gear unit, feed and conveyor belt (endless translational) or tool slide (limited translational). Translational axis models support rotational motor encoders as well as rotational or translational position encoders.

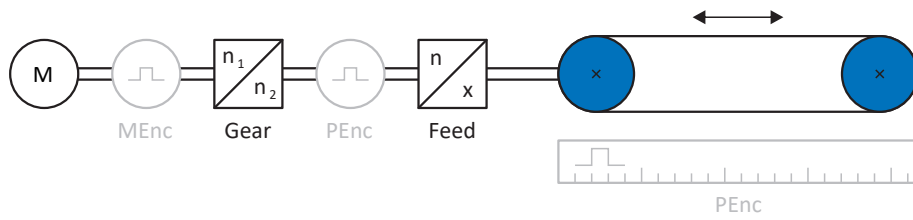


Fig. 19: Endless translational motion: Conveyor belt

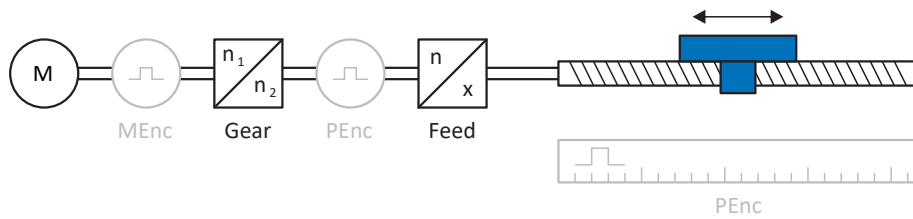


Fig. 20: Limited translational motion: Tool slide

Translational axis model: Linear motor

The following figure shows a limited translation axis model using the example of a linear motor. Linear motors only support translational motor encoders and translational position encoders.

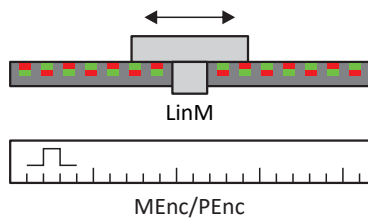


Fig. 21: Limited translational motion: Linear motor

Encoder arrangement

You parameterize the encoder arrangement by using B26 Motor encoder and I02 Position encoder. The motor encoder for velocity control is located on the motor shaft. The position encoder for position control is located on the gear unit output. If you use only one of the two encoders, it will be used for both velocity and position control.

| Encoder | Parameterization | Encoder arrangement |
|----------------------------------|---------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Motor encoder | B26 ≠ 0: Inactive I02 = 0: Motor encoder | <p>The diagram shows a motor (M) connected to a motor encoder (MEnc). The motor encoder is connected to a gear unit with input speed n_1 and output speed n_2. The gear unit is connected to a position encoder (PEnc). The position encoder is connected to two output lines.</p> |
| Position encoder | B26 ≠ 0: Inactive I02 = B26 | <p>The diagram shows a motor (M) connected to a motor encoder (MEnc). The motor encoder is connected to a gear unit with input speed n_1 and output speed n_2. The gear unit is connected to a position encoder (PEnc). The position encoder is connected to two output lines.</p> |
| Motor encoder & position encoder | B26 ≠ 0: Inactive I02 ≠ B26 | <p>The diagram shows a motor (M) connected to a motor encoder (MEnc). The motor encoder is connected to a gear unit with input speed n_1 and output speed n_2. The gear unit is connected to a position encoder (PEnc). The position encoder is connected to two output lines.</p> |

6.5 Limit switches

Limit switches are sensors that detect when a certain position has been reached.

A distinction is made here between hardware limit switches and software limit switches. While a hardware limit switch is a real switch (hardware), a software limit switch refers to the position limiting or position monitoring implemented in the software.

There are hardware and software limit switches on real axes, while virtual masters only have software limit switches.

Behavior in jog mode, reaching the +/- 31 bit computing limit and simultaneous triggering of positive and negative limit switches are special cases.

6.5.1 Real axes

There are hardware and software limit switches on real axes.

6.5.1.1 Faults

If a hardware limit switch is overshot, a fault is triggered immediately.

If a software limit switch is overshot, a fault is triggered if the following applies:

- For a positive software limit switch: actual position > limit switch position + I22
- For a negative software limit switch: actual position < limit switch position - I22

Faults

53: Limit switch

Causes

- 1: Hardware-Limit-Switch positive
- 2: Hardware-Limit-Switch negative
- 3: SW-limit switch positive
- 4: SW-limit switch negative
- 5: +/- 31 bit computing limit reached
- 6: Linear motor moving range
- 7: Both limit switches not connected

The fault can be acknowledged. Note that movement is only possible in the opposite direction of the limit switch, since the direction of the triggered hardware or software limit switch is locked. A direction lock is displayed in parameter I196.

A fault can be triggered again when the direction lock is no longer active.

6.5.1.2 Denial

If the axis is at a hardware limit switch (I441 or I442 = High) or if it is beyond of the software limit switch, movement is denied in the locked direction.

A direction lock is displayed in parameter I196. The direction lock is deactivated if the following applies:

- For a positive software limit switch: actual position < limit switch position – I22
- For a negative software limit switch: actual position > limit switch position + I22

When the direction lock is active, parameter I91 Error = 1: Active.

The cause is output in parameter I90.

Possible causes are:

- 1: Illegal direction
- 2: SW-limit switch positive
- 3: SW-limit switch negative
- 10: Denied because position is not in circular length
- 11: HW-limit switch positive
- 12: HW-limit switch negative

| |
|--------------------|
| Information |
|--------------------|

There are no software limit switches for an infinite axis, since this would contradict the nature of an infinite axis.

A movement limitation is also displayed in parameter E80:

- If E80 = 20: Limit switch, check the parameterization and the connection of the limit switches.
- If E80 = 15: Illegal direction, check the set values and a possible direction lock in parameter I196.

Also check I196 if the permitted direction was limited with I04.

| |
|--------------------|
| Information |
|--------------------|

Parameter I04 is only available for infinite axes.

6.5.1.3 Hardware limit switches

Hardware limit switches apply if your sources are defined in parameters I101 and I102.

Relevant parameters

- I101 Source positive /limit switch
- I102 Source negative /limit switch
- I441 Signal /HW limit switch positive
- I442 Signal /HW limit switch negative
- I805 Actual value HW-Limit switch positive
- I806 Actual value HW-Limit switch negative
- I52 Delete limit switch memory
- I196 Direction blocking

If I441 and I442 = 0: Inactive, fault 53 is triggered with cause 7: Both limit switches not connected. Therefore, after parameterizing I101 and I102, check whether the hardware limit switches are also connected physically.

If I101 and I102 = 2: Parameter, the fault is triggered by the control word of the application (I210). In this case, check the controller connection.

Passable hardware limit switches

The hardware limit switches can be passed over. It is therefore possible to use an infinite cam as a hardware limit switch. The limit switch is detected if the signal goes to 0: Inactive (parameters I441 and I442).

If the hardware limit switch is overshoot, the position at which the limit switch was detected is saved. When resetting limit switches, the axis must have reached or passed back over this saved position before the signal is valid again.

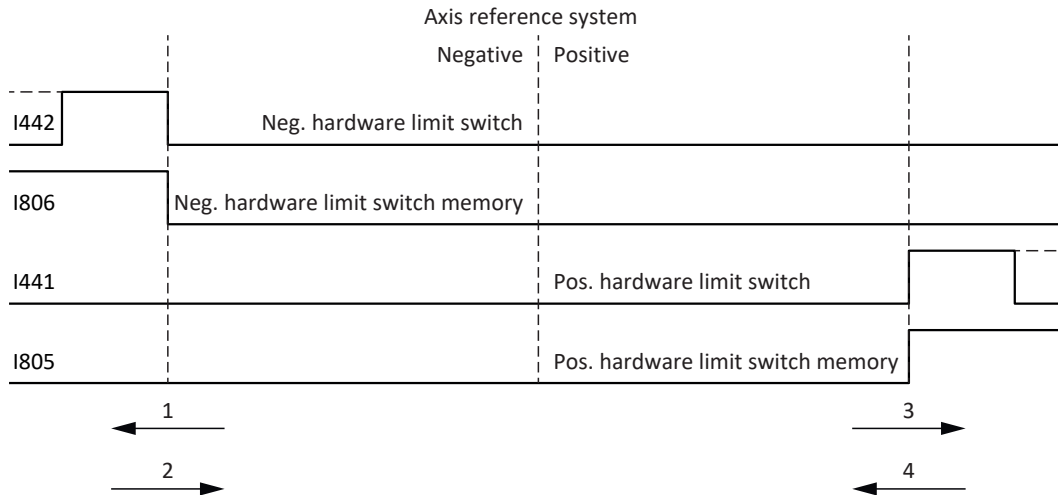


Fig. 22: Hardware limit switch memory

- 1 Setting the memory (negative hardware limit switch) with rising edge
- 2 Resetting the memory (negative hardware limit switch) with falling edge
- 3 Setting the memory (positive hardware limit switch) with rising edge
- 4 Resetting the memory (positive hardware limit switch) with falling edge

Information

The passable hardware limit switch is no longer effective once the position at which the edge of the limit switch was detected has been passed over again.

Note that the positions are not saved in non-volatile memory. This means that if the axis is beyond a passable limit switch after activation, it must first be reset to the regular travel range.

The entire system is simpler if you do not use any passable limit switches.

During commissioning or in case of defective hardware limit switch connections, problems can arise with the saved positions. These can be deleted with parameter I52. However, deletion only takes place if the corresponding limit switch signal is inactive. Stored positions can also be deleted by referencing, setting a reference can be deleted using I452 or by restarting the drive controller.

During referencing, the hardware limit switches are not evaluated as limit switches. There are referencing methods in which the hardware limit switches are used for referencing.

Example

A positive limit switch ranges from position 100 to 120.

When moving in the positive direction, it is detected at 100.

On the return journey, due to tolerances, it can be the case that the limit switch has already been left at 101.

Despite this, position 100 must be undershot in order to leave the effective limit switch.

6.5.1.4 Hardware limit switches when referencing

Hardware limit switches assume a special position when referencing.

Limit switch as reference switch

The hardware limit switches can be used in place of a reference switch (I30 = 2: Limit switch).

Change of direction

In other referencing types, a hardware limit switch signal leads to a change of direction. However, the signal only causes a change if it matches the direction of travel.

If the detected hardware limit switch does not match the direction of travel (e.g. negative limit switch with positive direction of travel), a fault is then triggered.

Example

At the start of referencing, the axis is between the reference switch and positive limit switch. The referencing direction is positive. The axis moves in the positive direction and first encounters the positive limit switch, not the reference switch. The axis reverses and searches for the reference switch in the other direction.

6.5.1.5 Software limit switches

Software limit switches only apply if the axis has been referenced.

Only limited axes have software limit switches (I00 = 0: Limited).

Relevant parameters

- I50 Software stop positive
- I51 Software stop negative
- I196 Direction blocking

If the values defined in parameter I50 and parameter I51 are the same, the software limit switches are deactivated.

6.5.1.6 Software limit switches when referencing

Software limit switches are not evaluated when referencing.

6.5.1.7 Shutting down by the limit switch

If a hardware limit switch is approached from the permitted travel range, a fault is immediately triggered after detecting the limit switch edge.

If a software limit switch is approached from the permissible travel range, a fault is triggered if the following applies:

- For a positive software limit switch: actual position > limit switch position + I22
- For a negative software limit switch: actual position < limit switch position + I22

If the current position is on or beyond the limit switch, a fault can be triggered again when the direction lock is no longer active (I196).

6.5.2 Virtual master

Virtual masters only have software limit switches.

Relevant parameters

- G146 Virtual master software limit switch +
- G147 Virtual master software limit switch -

Faults

If a software limit switch is overshot, fault 51: Virtual master software limit switch is triggered.

Causes

- 1: SW-limit switch positive
- 2: SW-limit switch negative
- 3: +/- 31 bit computing limit reached

The fault can be acknowledged. It can only be shut down in the opposite direction of the software limit switch. The direction of the triggered software limit switch is locked.

Denial

If the axis is beyond the software limit switch, movement in the wrong direction is denied.

In this case, parameter G161 = 1: Active.

The cause is output in parameter G163:

- 2: SW-limit switch positive
- 3: SW-limit switch negative

Information

There are no software limit switches for an infinite axis, since this would contradict the nature of an infinite axis.

Information

In order to trigger the quick stop of the virtual master when event 51: Virtual master software limit switch occurs, the level of the associated protective function must be selected accordingly (U24 = 3: Fault) and the axis fault must be defined as an additional quick-stop cause for the virtual master (G57 = 1: Error/3: Enable or error).

To trigger the quick stop of the virtual master when event 51: Virtual master software limit switch occurs, proceed as described in [Parameterizing the virtual master](#) [▶ 36].

6.5.3 Special cases

A few special cases are described below.

6.5.3.1 Behavior in jog mode



Risk of fatal injury due to gravity-loaded vertical axis!

If you select 0: Velocity control as the control type for jog mode, a gravity-loaded vertical axis descends as soon as jog mode is activated via the control panel, the application or the operating unit (SD6) (parameter I26).

- Only use the 0: Velocity control control type with axes that do not have a gravity load.

The jog behavior does not depend on the control type selected for jog mode (I26).

As soon as the axis moves to a software limit switch, a quick stop is triggered instead of a fault. As a result, the axis does not stop precisely on the software limit switch, but shortly thereafter depending on the quick stop ramp. It can immediately be shut down again by the software limit switch. The "incorrect" direction is locked.

6.5.3.2 Reaching the +/- 31 bit computing limit

Fault:

53: Limit switch

Cause:

5: +/- 31 bit computing limit reached

The computing limit can be reached if, for example, so many commands follow after each other for the moving axis that the total distance to be traveled (including decimal places) adds up to over 2^{31} .

6.5.3.3 Linear motor travel range

Fault:

53: Limit switch

Cause:

6: Linear motor moving range

This fault is triggered if the axis is more than 200 m away from the commutation reference point, which means that the actual position of the motor has left the permissible range (parameter: E09). Check the axis model and correct the parameterization if necessary.

6.5.3.4 Simultaneous detection of positive and negative hardware limit switches

Fault:

53: Limit switch

Cause:

7: Both limit switches not connected

This fault is triggered if I441 and I442 are both 0: Inactive. Check whether the hardware limit switches are also connected physically.

If I101 and I102 = 2: Parameter, the fault is triggered by the control word of the application (I210). In this case, check the controller connection.

The fault can be acknowledged with I52.

6.5.3.5 Behavior upon controller start-up

Fault:

53: Limit switch

Cause:

7: Both limit switches not connected

If a controller is used as the source for the digital signals for evaluating the hardware limit switches and this is still in device start-up while the drive controller and fieldbus communication are already active, fault 53: Limit switch is triggered and the limit switch memory is set (I805, I806 = 1: Active).

If the power unit has not yet been enabled since the supply voltage was switched on, the limit switch memory is automatically reset as soon as the controller correctly transmits the signals for evaluating the hardware limit switches.

6.6 Referencing

To be able to work with a system with absolute positions, the relation of a measured to an actual axis position must be determined.

During initial commissioning or after changes to the axis model, the actual position of the axis is unknown. A defined initial position is necessary.

As a rule, this is either identified through a reference search or referencing setting. The associated process is referred to as referencing.

Absolute movements can be performed only in a referenced state.

6.6.1 Referencing methods

The following table shows an overview of the possible referencing methods.

| Abbreviation | Meaning |
|--------------|-----------------|
| S | Switch |
| M/F | Torque or force |

| | Method | Initial movement | Zero pulse | Feature |
|--|--------|------------------|------------|----------------------------------------|
| | A | Negative | Yes | Negative limit switch |
| | B | Negative | Yes | Negatively positioned reference switch |
| | C | Negative | – | Negative limit switch |
| | D | Negative | – | Negatively positioned reference switch |
| | E | Positive | Yes | Positive limit switch |
| | F | Positive | Yes | Positively positioned reference switch |
| | G | Positive | – | Positive limit switch |
| | H | Positive | – | Positively positioned reference switch |
| | I | Positive | Yes | Centrally positioned reference switch |
| | J | Positive | – | Centrally positioned reference switch |
| | K | Negative | Yes | Centrally positioned reference switch |
| | L | Negative | – | Centrally positioned reference switch |
| | M | Negative | Yes | Zero pulse |
| | N | Positive | Yes | Zero pulse |
| | O | – | – | Setting the reference |
| | P | Positive | – | Torque/force stop |
| | Q | Positive | Yes | Torque/force stop |
| | R | Negative | – | Torque/force stop |
| | S | Negative | Yes | Torque/force stop |

Tab. 5: Referencing methods

6.6.1.1 Referencing methods in detail

The following chapters show the individual referencing methods in detail.

The following abbreviations are used in the graphs on the referencing methods:

| Abbreviation | Meaning |
|--------------|------------------|
| ALT | Alternative |
| LS | Limit Switch |
| RS | Reference Switch |
| ZP | Zero Pulse |

Information

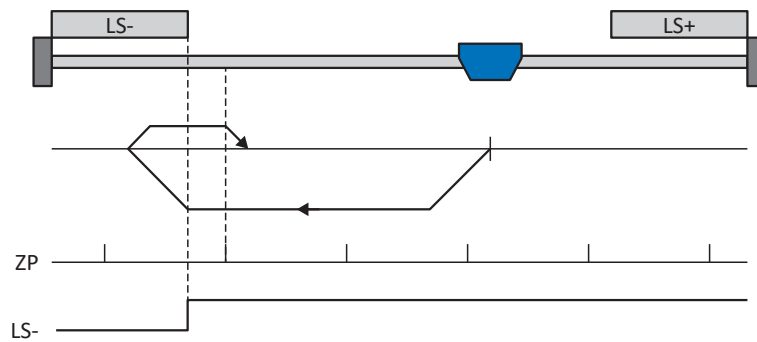
For graphically representing axes, the left is the smallest and the right is the largest position value. This means a positive movement goes to the right and a negative to the left.

The following parameters are used in the descriptions of the referencing methods:

| Coordinate | Name |
|------------|--------------------------------|
| I28 | Homing torque/force limit |
| I29 | Time homing torque/force limit |
| I30 | Referencing type |
| I31 | Referencing direction |
| I32 | Referencing speed fast |
| I33 | Referencing speed slow |
| I34 | Reference position |
| I35 | Referencing with zero pulse |
| I39 | Referencing acceleration |
| I43 | Move to reference position |
| I44 | Referencing jerk |
| I53 | Index search offset |
| I101 | Source positive /limit switch |
| I102 | Source negative /limit switch |
| I103 | Source reference switch |

6.6.1.1.1 Referencing method A

Referencing method A determines a reference using a run to the negative limit switch and zero pulse.



Preparation

1. Activate referencing method A by setting the following parameters to the specified values:
 I30 to 2: Limit switch,
 I31 to 1: Negative,
 I35 to 1: Active.
2. I102:
 Enter the data source for the negative limit switch.
3. I32, I33, I39, I44, I34:
 Define the set values necessary for referencing.
4. I53:
 Define the start of the search for the zero pulse search

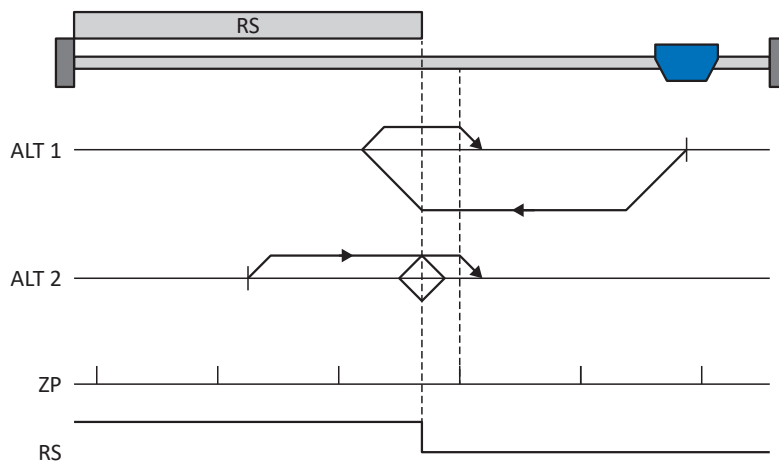
Referencing

If the PLCopen command MC_Home is active, referencing takes places as follows:

1. The drive starts in the negative direction with acceleration I39 and velocity I32.
2. It changes its direction upon reaching the negative limit switch and continues its run with velocity I33 until it reaches the next zero pulse after leaving the limit switch.
3. The current actual position is set to the value of reference position I34 upon reaching the zero pulse.
4. The drive comes to a standstill with deceleration I39.
5. If I43 is set to 1: Active, the drive is positioned at reference position I34.

6.6.1.1.2 Referencing method B

Referencing method B determines the reference using a run to the negatively positioned reference switch and zero pulse.



Preparation

1. Activate referencing method B by setting the following parameters to the specified values:
I30 to 1: Reference input,
I31 to 1: Negative,
I35 to 1: Active.
2. I103:
Enter the data source for the reference switch.
3. I32, I33, I39, I44, I34:
Define the set values necessary for referencing.
4. I53:
Define the start of the search for the zero pulse search

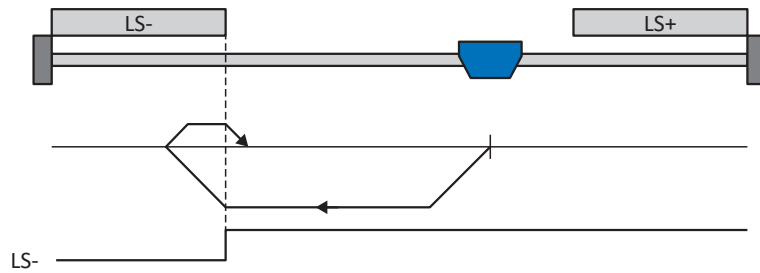
Referencing

If the PLCopen command MC_Home is active, a distinction is made between two referencing variants.

- ✓ Alternative 1: Drive is positioned in front of the reference switch
 1. The drive starts in the negative direction with acceleration I39 and velocity I32.
 2. It changes its direction upon reaching the reference switch and continues its run with velocity I33 until it reaches the next zero pulse after leaving the reference switch.
 3. The current actual position is set to the value of reference position I34 upon reaching the zero pulse.
 4. The drive comes to a standstill with deceleration I39.
 5. If I43 is set to 1: Active, the drive is positioned at reference position I34.
- ✓ Alternative 2: Drive stopped at reference switch
 1. The drive starts in the positive direction with acceleration I39 and velocity I33.
 2. It changes its direction upon reaching the reference switch and continues its run with velocity I32.
 3. When the drive comes to a stop after the reference switch, it changes direction again and continues at velocity I33 until it reaches the zero pulse.
 4. The current actual position is set to the value of reference position I34 upon reaching the zero pulse.
 5. The drive comes to a standstill with deceleration I39.
 6. If I43 is set to 1: Active, the drive is positioned at reference position I34.

6.6.1.1.3 Referencing method C

Referencing method C determines the reference using a run to the negative limit switch.



Preparation

1. Activate referencing method C by setting the following parameters to the specified values:
I30 to 2: Limit switch,
I31 to 1: Negative,
I35 to 0: Inactive.
2. I102:
Enter the data source for the negative limit switch.
3. I32, I33, I39, I44, I34:
Define the set values necessary for referencing.

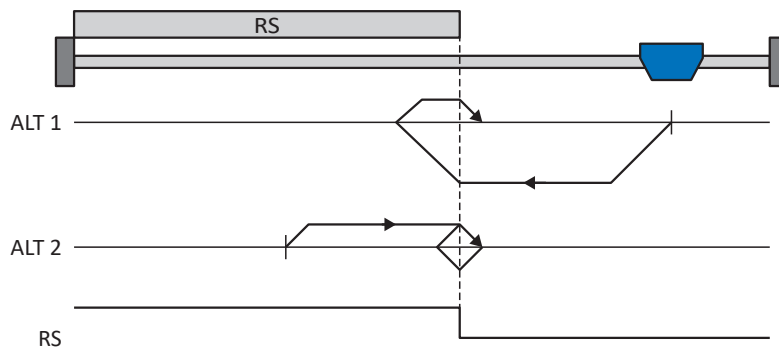
Referencing

If the PLCopen command MC_Home is active, referencing takes places as follows:

1. The drive starts in the negative direction with acceleration I39 and velocity I32.
2. It changes its direction upon reaching the negative limit switch and continues its run with velocity I33 until it leaves the limit switch again.
3. The current actual position is set to the value of the reference position I34 upon leaving the limit switch.
4. The drive comes to a standstill with deceleration I39.
5. If I43 is set to 1: Active, the drive is positioned at reference position I34.

6.6.1.1.4 Referencing method D

Referencing method D determines the reference using a run to the negatively positioned reference switch.



Preparation

1. Activate referencing method D by setting the following parameters to the specified values:
I30 to 1: Reference input,
I31 to 1: Negative,
I35 to 0: Inactive.
2. I103:
Enter the data source for the reference switch.
3. I32, I33, I39, I44, I34:
Define the set values necessary for referencing.

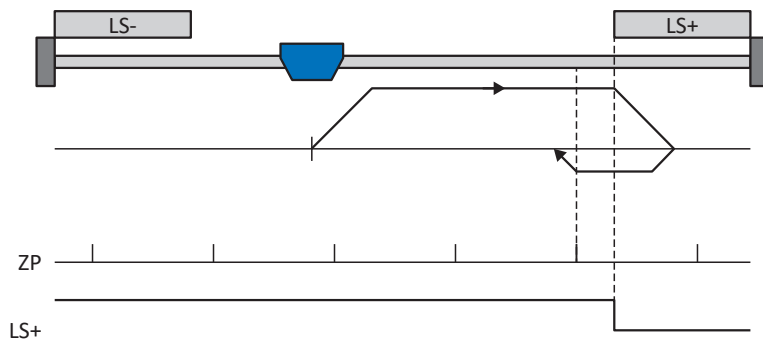
Referencing

If the PLCopen command MC_Home is active, a distinction is made between two referencing variants.

- ✓ Alternative 1: Drive is positioned in front of the reference switch
 1. The drive starts in the negative direction with acceleration I39 and velocity I32.
 2. It changes its direction upon reaching the reference switch and continues its run with velocity I33 until it leaves the reference switch again.
 3. The current actual position is set to the value of the reference position I34 upon leaving the reference switch.
 4. The drive comes to a standstill with deceleration I39.
 5. If I43 is set to 1: Active, the drive is positioned at reference position I34.
- ✓ Alternative 2: Drive stopped at reference switch
 1. The drive starts with acceleration I39 and velocity I33 in the positive direction until it leaves the reference switch.
 2. It changes its direction upon leaving the reference switch and continues its run with velocity I32.
 3. When the drive comes to a stop after the reference switch, it changes direction and continues its run at velocity I33 until it reaches the reference switch again.
 4. The current actual position is set to the value of the reference position I34 upon reaching the reference switch.
 5. The drive comes to a standstill with deceleration I39.
 6. If I43 is set to 1: Active, the drive is positioned at reference position I34.

6.6.1.1.5 Referencing method E

Referencing method E determines the reference using a run to the positive limit switch and zero pulse.



Preparation

1. Activate referencing method E by setting the following parameters to the specified values:
 I30 to 2: Limit switch,
 I31 to 0: Positive,
 I35 to 1: Active.
2. I101:
 Enter the data source for the positive limit switch.
3. I32, I33, I39, I44, I34:
 Define the set values necessary for referencing.
4. I53:
 Define the start of the search for the zero pulse search

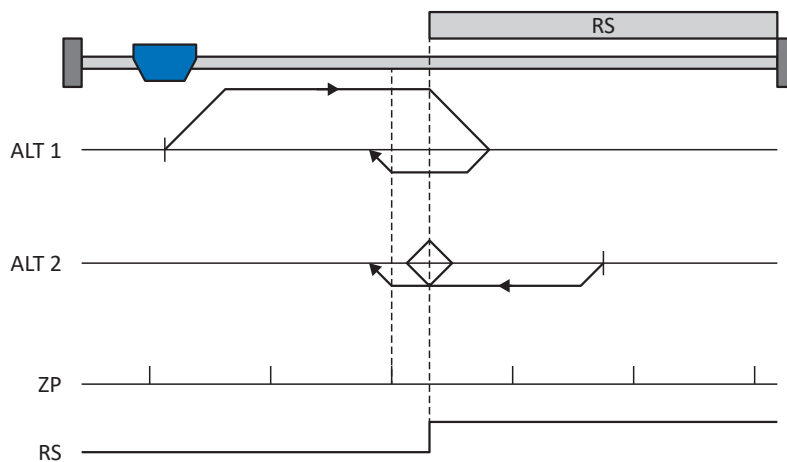
Referencing

If the PLCopen command MC_Home is active, referencing takes places as follows:

1. The drive starts in the positive direction with acceleration I39 and velocity I32.
2. It changes its direction upon reaching the positive limit switch and continues its run with velocity I33 until it reaches the next zero pulse after leaving the limit switch.
3. The current actual position is set to the value of reference position I34 upon reaching the zero pulse.
4. The drive comes to a standstill with deceleration I39.
5. If I43 is set to 1: Active, the drive is positioned at reference position I34.

6.6.1.1.6 Referencing method F

Referencing method F determines the reference using a run to the positively positioned reference switch and zero pulse.



Preparation

1. Activate referencing method F by setting the following parameters to the specified values:
I30 to 1: Reference input,
I31 to 0: Positive,
I35 to 1: Active.
2. I103:
Enter the data source for the reference switch.
3. I32, I33, I39, I44, I34:
Define the set values necessary for referencing.
4. I53:
Define the start of the search for the zero pulse search

Referencing

If the PLCopen command MC_Home is active, a distinction is made between two referencing variants.

✓ Alternative 1: Drive is positioned in front of the reference switch

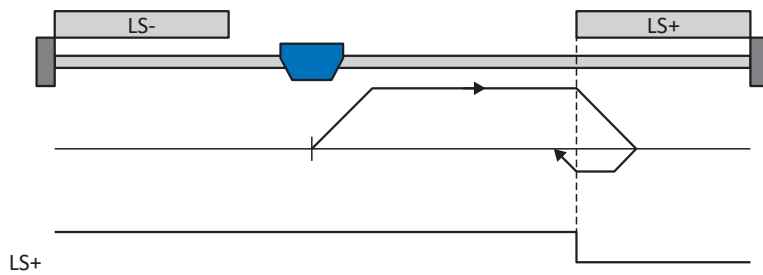
1. The drive starts in the positive direction with acceleration I39 and velocity I32.
2. It changes its direction upon reaching the reference switch and continues its run with velocity I33 until it reaches the next zero pulse after leaving the reference switch.
3. The current actual position is set to the value of reference position I34 upon reaching the zero pulse.
4. The drive comes to a standstill with deceleration I39.
5. If I43 is set to 1: Active, the drive is positioned at reference position I34.

✓ Alternative 2: Drive stopped at reference switch

1. The drive starts in the negative direction with acceleration I39 and velocity I33.
2. It changes its direction upon reaching the reference switch and continues its run with velocity I32.
3. When the drive comes to a stop after the reference switch, it changes direction again and continues at velocity I33 until it reaches the zero pulse.
4. The current actual position is set to the value of reference position I34 upon reaching the zero pulse.
5. The drive comes to a standstill with deceleration I39.
6. If I43 is set to 1: Active, the drive is positioned at reference position I34.

6.6.1.1.7 Referencing method G

Referencing method G determines the reference using a run to the positive limit switch.



Preparation

1. Activate referencing method G by setting the following parameters to the specified values:
 - I30 to 2: Limit switch,
 - I31 to 0: Positive,
 - I35 to 0: Inactive.
2. I101:
 - Enter the data source for the positive limit switch.
3. I32, I33, I39, I44, I34:
 - Define the set values necessary for referencing.

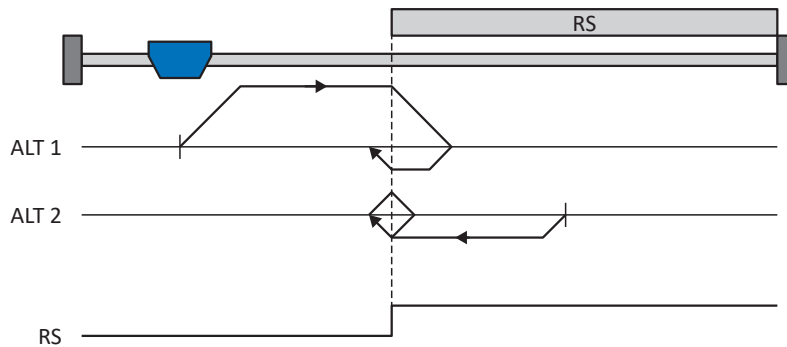
Referencing

If the PLCopen command MC_Home is active, referencing takes place as follows:

1. The drive starts in the positive direction with acceleration I39 and velocity I32.
2. It changes its direction upon reaching the positive limit switch and continues its run with velocity I33 until it leaves the limit switch again.
3. The current actual position is set to the value of the reference position I34 upon leaving the limit switch.
4. The drive comes to a standstill with deceleration I39.
5. If I43 is set to 1: Active, the drive is positioned at reference position I34.

6.6.1.1.8 Referencing method H

Referencing method H determines the reference using a run to the positively positioned reference switch.



Preparation

1. Activate referencing method H by setting the following parameters to the specified values:
I30 to 1: Reference input,
I31 to 0: Positive,
I35 to 0: Inactive.
2. I103:
Enter the data source for the reference switch.
3. I32, I33, I39, I44, I34:
Define the set values necessary for referencing.

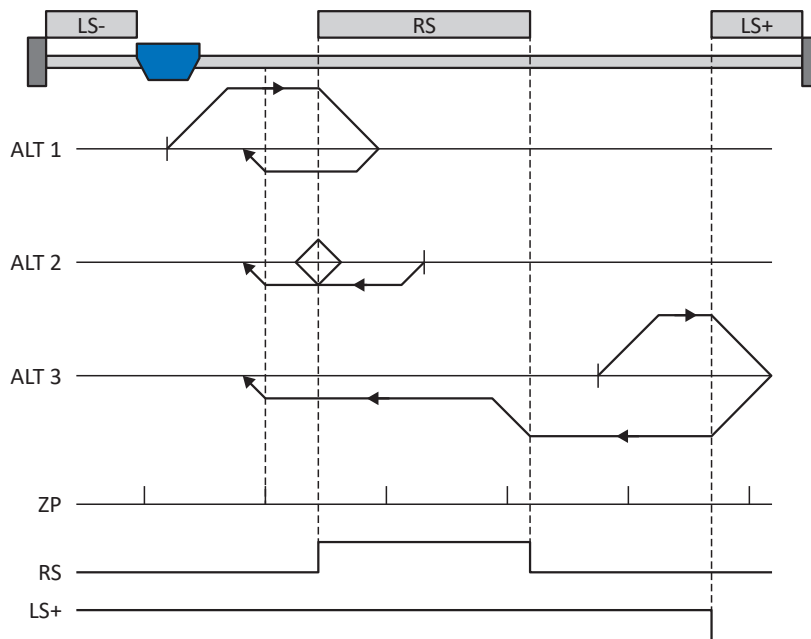
Referencing

If the PLCopen command MC_Home is active, a distinction is made between two referencing variants.

- ✓ Alternative 1: Drive is positioned in front of the reference switch
 1. The drive starts in the positive direction with acceleration I39 and velocity I32.
 2. It changes its direction upon reaching the reference switch and continues its run with velocity I33 until it leaves the reference switch again.
 3. The current actual position is set to the value of the reference position I34 upon leaving the reference switch.
 4. The drive comes to a standstill with deceleration I39.
 5. If I43 is set to 1: Active, the drive is positioned at reference position I34.
- ✓ Alternative 2: Drive stopped at reference switch
 1. The drive starts with acceleration I39 and velocity I33 in the negative direction until it leaves the reference switch.
 2. It changes its direction upon leaving the reference switch and continues its run with velocity I32.
 3. When the drive comes to a stop after the reference switch, it changes direction and continues its run at velocity I33 until it reaches the reference switch again.
 4. The current actual position is set to the value of the reference position I34 upon reaching the reference switch.
 5. The drive comes to a standstill with deceleration I39.
 6. If I43 is set to 1: Active, the drive is positioned at reference position I34.

6.6.1.1.9 Referencing method I

Referencing method I determines the reference using a run to the centered reference switch and zero pulse.



Preparation

1. Activate referencing method I by setting the following parameters to the specified values:
 - I30 to 1: Reference input,
 - I31 to 0: Positive,
 - I35 to 1: Active.
2. I103:
 - Enter the data source for the reference switch.
3. I32, I33, I39, I44, I34:
 - Define the set values necessary for referencing.
4. I53:
 - Define the start of the search for the zero pulse search

Referencing

If the PLCopen command MC_Home is active, a distinction is made between three referencing sequences.

- ✓ Alternative 1: Drive is positioned between the negative limit and reference switches
 1. The drive starts in the positive direction with acceleration I39 and velocity I32.
 2. It changes its direction upon reaching the reference switch and continues its run with velocity I33 until it reaches the next zero pulse after leaving the reference switch.
 3. The current actual position is set to the value of reference position I34 upon reaching the zero pulse.
 4. The drive comes to a standstill with deceleration I39.
 5. If I43 is set to 1: Active, the drive is positioned at reference position I34.

✓ Alternative 2: Drive stopped at reference switch

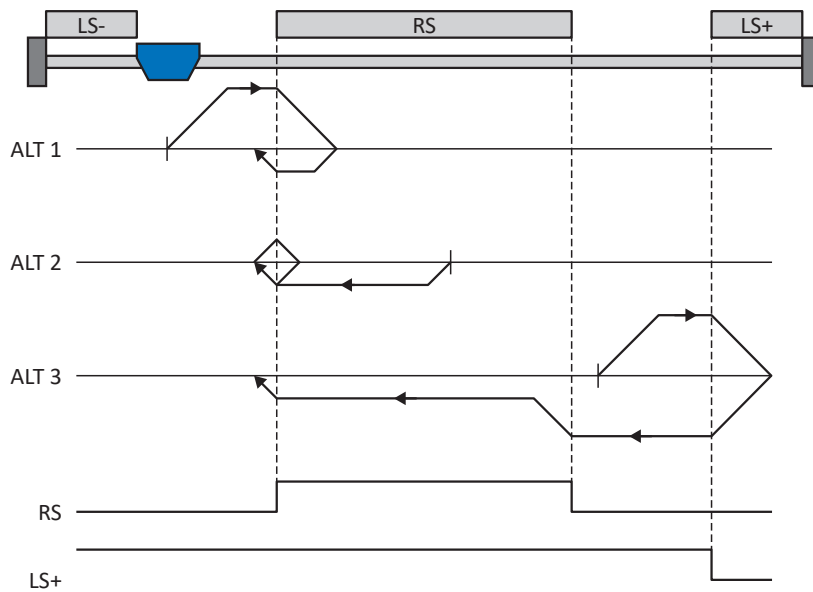
1. The drive starts in the negative direction with acceleration I39 and velocity I33.
2. It changes its direction upon reaching the reference switch and continues its run with velocity I32.
3. When the drive comes to a stop after the reference switch, it changes direction again and continues at velocity I33 until it reaches the zero pulse.
4. The current actual position is set to the value of reference position I34 upon reaching the zero pulse.
5. The drive comes to a standstill with deceleration I39.
6. If I43 is set to 1: Active, the drive is positioned at reference position I34.

✓ Alternative 3: Drive is positioned between the reference and positive limit switches

1. The drive starts in the positive direction with acceleration I39 and velocity I32.
2. It changes its direction upon reaching the positive limit switch until the reference switch is reached.
3. Upon reaching the reference switch, the drive changes its velocity to I33 until it leaves the reference switch again.
4. The current actual position is set to the value of the reference position I34 after leaving the reference switch and upon reaching the next zero pulse.
5. The drive comes to a standstill with deceleration I39.
6. If I43 is set to 1: Active, the drive is positioned at reference position I34.

6.6.1.1.10 Referencing method J

Referencing method J determines the reference using a run to the centered reference switch.



Preparation

1. Activate referencing method J by setting the following parameters to the specified values:
 I30 to 1: Reference input,
 I31 to 0: Positive,
 I35 to 0: Inactive.
2. I103:
 Enter the data source for the reference switch.
3. I32, I33, I39, I44, I34:
 Define the set values necessary for referencing.

Referencing

If the PLCopen command MC_Home is active, a distinction is made between three referencing sequences.

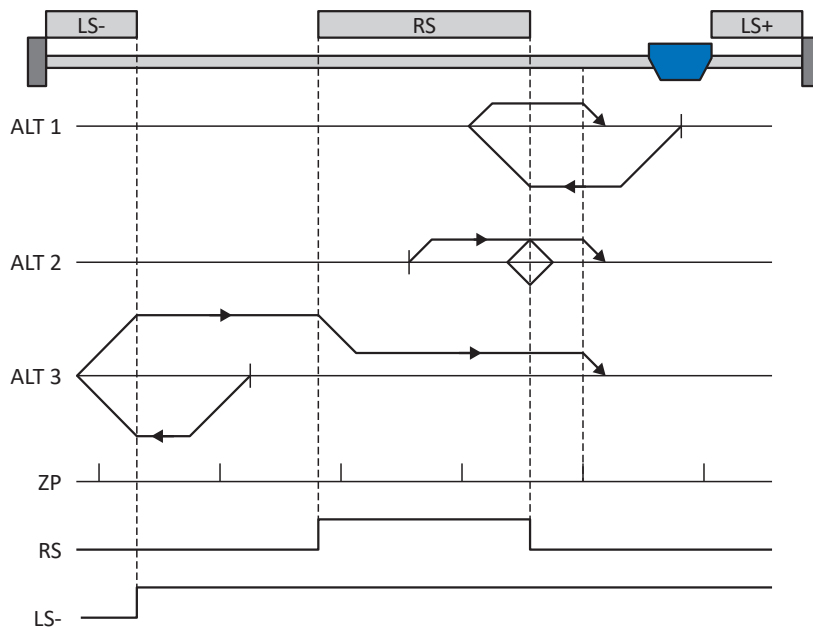
- ✓ Alternative 1: Drive is positioned between the negative limit and reference switches
1. The drive starts in the positive direction with acceleration I39 and velocity I32.
 2. It changes its direction upon reaching the reference switch and continues its run with velocity I33 until it leaves the reference switch again.
 3. The current actual position is set to the value of the reference position I34 upon leaving the reference switch.
 4. The drive comes to a standstill with deceleration I39.
 5. If I43 is set to 1: Active, the drive is positioned at reference position I34.

- ✓ Alternative 2: Drive stopped at reference switch
 1. The drive starts in the negative direction with acceleration I39 and velocity I33.
 2. It changes its direction upon reaching the reference switch and continues its run with velocity I32 until it leaves the reference switch.
 3. When the drive comes to a stop after the reference switch, it changes direction and continues its run at velocity I33 until it reaches the reference switch again.
 4. The current actual position is set to the value of the reference position I34 upon reaching the reference switch.
 5. The drive comes to a standstill with deceleration I39.
 6. If I43 is set to 1: Active, the drive is positioned at reference position I34.

- ✓ Alternative 3: Drive is positioned between the reference and positive limit switches
 1. The drive starts in the positive direction with acceleration I39 and velocity I32.
 2. Upon reaching the positive limit switch, the drive changes its direction and continues its run until it reaches the reference switch.
 3. Upon reaching the reference switch, the drive changes its velocity to I33 until it leaves the reference switch again.
 4. The current actual position is set to the value of the reference position I34 upon leaving the reference switch.
 5. The drive comes to a standstill with deceleration I39.
 6. If I43 is set to 1: Active, the drive is positioned at reference position I34.

6.6.1.1.11 Referencing method K

Referencing method K determines the reference using a run to the centered reference switch and zero pulse.



Preparation

1. Activate referencing method K by setting the following parameters to the specified values:
 I30 to 1: Reference input,
 I31 to 1: Negative,
 I35 to 1: Active.
2. I103:
 Enter the data source for the reference switch.
3. I32, I33, I39, I44, I34:
 Define the set values necessary for referencing.
4. I53:
 Define the start of the search for the zero pulse search

Referencing

If the PLCopen command MC_Home is active, a distinction is made between three referencing variants.

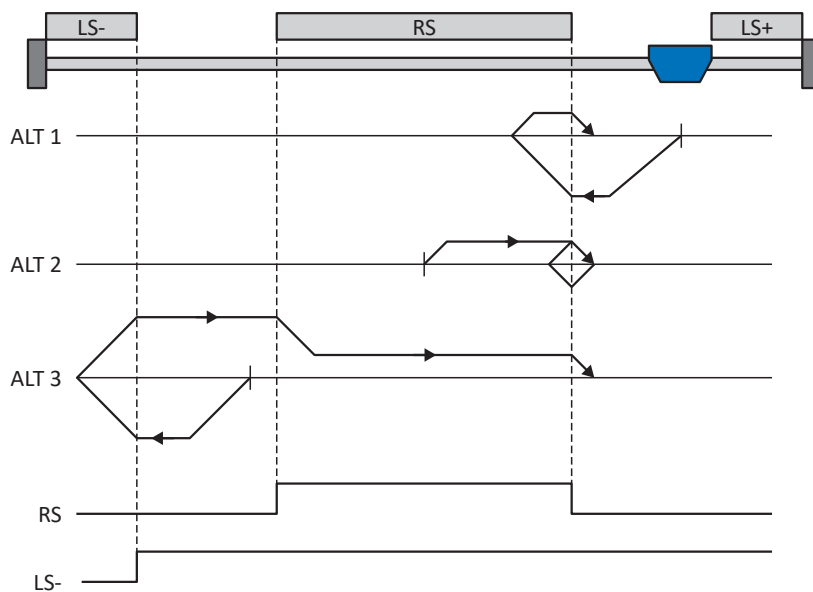
✓ Alternative 1: Drive is positioned between the reference and positive limit switches

1. The drive starts in the negative direction with acceleration I39 and velocity I32.
2. It changes its direction upon reaching the reference switch and continues its run with velocity I33 until it reaches the next zero pulse after leaving the reference switch.
3. The current actual position is set to the value of reference position I34 upon reaching the zero pulse.
4. The drive comes to a standstill with deceleration I39.
5. If I43 is set to 1: Active, the drive is positioned at reference position I34.

- ✓ Alternative 2: Drive stopped at reference switch
 1. The drive starts in the positive direction with acceleration I39 and velocity I33.
 2. It changes its direction upon reaching the reference switch and continues its run with velocity I32.
 3. When the drive comes to a stop after the reference switch, it changes direction again and continues at velocity I33 until it reaches the zero pulse.
 4. The current actual position is set to the value of reference position I34 upon reaching the zero pulse.
 5. The drive comes to a standstill with deceleration I39.
 6. If I43 is set to 1: Active, the drive is positioned at reference position I34.
- ✓ Alternative 3: Drive is positioned between the negative limit and reference switches
 1. The drive starts in the negative direction with acceleration I39 and velocity I32.
 2. It changes its direction upon reaching the negative limit switch and continues its run until the reference switch is reached.
 3. Upon reaching the reference switch, the drive changes its velocity to I33 and continues its run until it reaches the next zero pulse after leaving the reference switch.
 4. The current actual position is set to the value of reference position I34 upon reaching the zero pulse.
 5. The drive comes to a standstill with deceleration I39.
 6. If I43 is set to 1: Active, the drive is positioned at reference position I34.

6.6.1.1.12 Referencing method L

Referencing method L determines the reference using a run to the centered reference switch.



Preparation

1. Activate referencing method L by setting the following parameters to the specified values:
 I30 to 1: Reference input,
 I31 to 1: Negative,
 I35 to 0: Inactive.
2. I103:
 Enter the data source for the reference switch.
3. I32, I33, I39, I44, I34:
 Define the set values necessary for referencing.

Referencing

If the PLCopen command MC_Home is active, a distinction is made between three referencing variants.

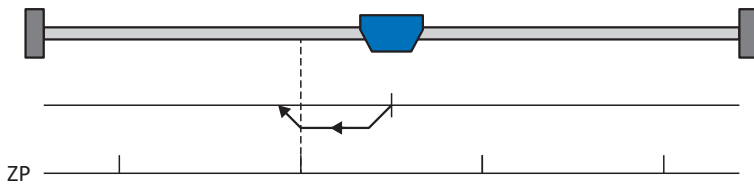
✓ Alternative 1: Drive is positioned between the reference and positive limit switches

1. The drive starts in the negative direction with acceleration I39 and velocity I32.
2. It changes its direction upon reaching the reference switch and continues its run with velocity I33 until it leaves the reference switch again.
3. The current actual position is set to the value of the reference position I34 upon leaving the reference switch.
4. The drive comes to a standstill with deceleration I39.
5. If I43 is set to 1: Active, the drive is positioned at reference position I34.

- ✓ Alternative 2: Drive stopped at reference switch
 1. The drive starts with acceleration I39 and velocity I33 in the positive direction until it leaves the reference switch.
 2. It changes its direction upon leaving the reference switch and continues its run with velocity I32.
 3. When the drive comes to a stop after the reference switch, it changes direction and continues its run at velocity I33 until it reaches the reference switch again.
 4. The current actual position is set to the value of the reference position I34 upon reaching the reference switch.
 5. The drive comes to a standstill with deceleration I39.
 6. If I43 is set to 1: Active, the drive is positioned at reference position I34.
- ✓ Alternative 3: Drive is positioned between the negative limit and reference switches
 1. The drive starts in the negative direction with acceleration I39 and velocity I32.
 2. It changes its direction upon reaching the negative limit switch and continues its run until the reference switch is reached.
 3. When the reference switch is reached, the drive changes its velocity to I33 and continues its run until it leaves the reference switch again.
 4. The current actual position is set to the value of the reference position I34 upon leaving the reference switch.
 5. The drive comes to a standstill with deceleration I39.
 6. If I43 is set to 1: Active, the drive is positioned at reference position I34.

6.6.1.1.13 Referencing method M

This method determines the reference using a run to the zero pulse.



Preparation

1. Activate referencing method M by setting the following parameters to the specified values:
 I30 to 3: Zero pulse,
 I31 to 1: Negative,
2. I32, I39, I44, I34:
 Define the set values necessary for the referencing.

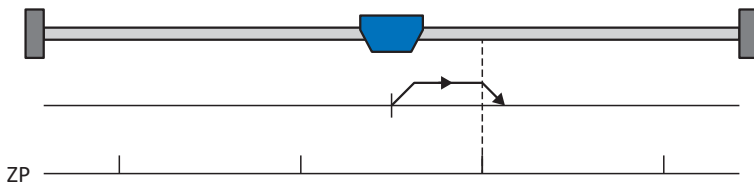
Referencing

If the PLCopen command MC_Home is active, referencing takes places as follows:

1. The drive starts in the negative direction with acceleration I39 and velocity I32.
2. The current actual position is set to the value of reference position I34 upon reaching the zero pulse.
3. The drive comes to a standstill with deceleration I39.
4. If I43 is set to 1: Active, the drive is positioned at reference position I34.

6.6.1.1.14 Referencing method N

Referencing method N determines the reference using a run to the zero pulse.



Preparation

1. Activate referencing method N by setting the following parameters to the specified values:
 I30 to 3: Zero pulse,
 I31 to 0: Positive,
2. I32, I39, I44, I34:
 Define the set values necessary for the referencing.

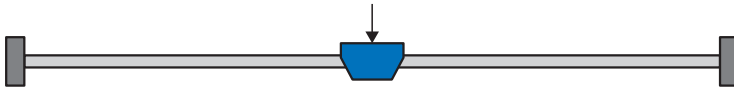
Referencing

If the PLCopen command MC_Home is active, referencing takes places as follows:

1. The drive starts in the positive direction with acceleration I39 and velocity I32.
2. The current actual position is set to the value of reference position I34 upon reaching the zero pulse.
3. The drive comes to a standstill with deceleration I39.
4. If I43 is set to 1: Active, the drive is positioned at reference position I34.

6.6.1.15 Referencing method O

Referencing method O determines the reference by setting the reference to any position.



Preparation

1. I30:
Activate referencing method O by setting this parameter to 5: Define home.
2. I34:
Define the reference position.

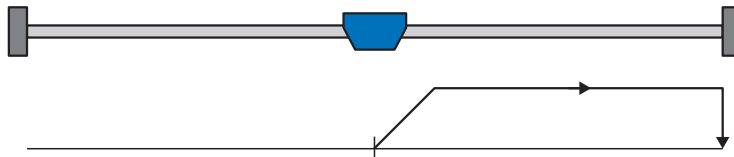
Referencing

If the PLCopen command MC_Home is active, referencing takes places as follows:

The current actual position is set to the value of the reference position I34.

6.6.1.16 Referencing method P

Referencing method P determines the reference using a run with a torque/force stop.



Preparation

1. Activate referencing method P by setting the following parameters to the specified values:
I30 to 4: Torque/force stopper,
I31 to 0: Positive,
I35 to 0: Inactive.
2. I32, I39, I44, I34:
Define the set values necessary for the referencing.
3. I28, I29:
Define the torque/force limit. If the actual torque for the time stored in I29 is above the limit defined in I28 for an ongoing period, the torque/force limit is reached.

Information

In case of a selected value for the torque/force variable that is too big, the machine can be damaged. In case of a selected value that is too small, an incorrect reference position may be applied.

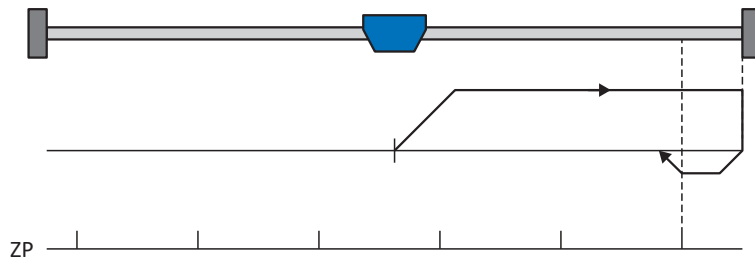
Referencing

If the PLCopen command MC_Home is active, referencing takes places as follows:

1. The drive starts in the positive direction with acceleration I39 and velocity I32.
2. The current actual position is set to the value of the reference position I34 when the torque/force limit is reached and the time stored in I29 has elapsed.
3. The set values are set to the value 0 with deceleration I39.
4. If I43 is set to 1: Active, the drive is positioned at reference position I34.

6.6.1.17 Referencing method Q

Referencing method Q determines the reference using a run with a torque/force stop and zero pulse.



Preparation

1. Activate referencing method Q by setting the following parameters to the specified values:
 I30 to 4: Torque/force stopper,
 I31 to 0: Positive,
 I35 to 1: Active.
2. I32, I33, I39, I44, I34:
 Define the set values necessary for referencing.
3. I28, I29:
 Define the torque/force limit. If the actual torque for the time stored in I29 is above the limit defined in I28 for an ongoing period, the torque/force limit is reached.
4. I53:
 Define the start of the search for the zero pulse search.

Information

In case of a selected value for the torque/force variable that is too big, the machine can be damaged. In case of a selected value that is too small, an incorrect reference position may be applied.

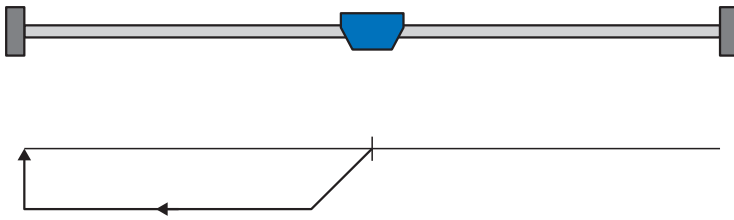
Referencing

If the PLCopen command MC_Home is active, referencing takes places as follows:

1. The drive starts in the positive direction with acceleration I39 and velocity I32.
2. It changes its direction upon reaching the torque/force stop and when the time stored in I29 has passed, and continues its run with velocity I33 until it reaches the next zero pulse.
3. The current actual position is set to the value of reference position I34 upon reaching the zero pulse.
4. The set values are set to the value 0 with deceleration I39.
5. If I43 is set to 1: Active, the drive is positioned at reference position I34.

6.6.1.1.18 Referencing method R

Referencing method R determines the reference using a run with a torque/force stop.



Preparation

1. Activate referencing method R by setting the following parameters to the specified values:
 I30 to 4: Torque/force stopper,
 I31 to 1: Negative,
 I35 to 0: Inactive.
2. I32, I33, I39, I44, I34:
 Define the set values necessary for referencing.
3. I28, I29:
 Define the torque/force limit. If the actual torque for the time stored in I29 is above the limit defined in I28 for an ongoing period, the torque/force limit is reached.

Information

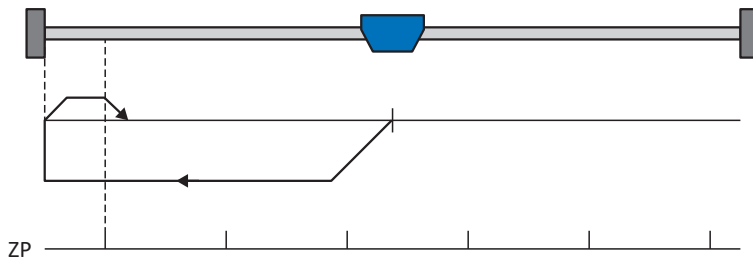
In case of a selected value for the torque/force variable that is too big, the machine can be damaged. In case of a selected value that is too small, an incorrect reference position may be applied.

Referencing

1. The drive starts in the negative direction with acceleration I39 and velocity I32.
2. The current actual position is set to the value of the reference position I34 when the torque/force stop is reached and the time stored in I29 has elapsed.
3. The set values are set to the value 0 with deceleration I39.
4. If I43 is set to 1: Active, the drive is positioned at reference position I34.

6.6.1.1.19 Referencing method S

Referencing method S determines the reference using a run with torque/force stop and zero pulse.



Preparation

1. Activate referencing method S by setting the following parameters to the specified values:
I30 to 4: Torque/force stopper,
I31 to 1: Negative,
I35 to 1: Active.
2. I32, I33, I39, I44, I34:
Define the set values necessary for referencing.
3. I28, I29:
Define the torque/force limit. If the actual torque for the time stored in I29 is above the limit defined in I28 for an ongoing period, the torque/force limit is reached.
4. I53:
Define the start of the search for the zero pulse search

Information

In case of a selected value for the torque/force variable that is too big, the machine can be damaged. In case of a selected value that is too small, an incorrect reference position may be applied.

Referencing

If the PLCopen command MC_Home is active, referencing takes places as follows:

1. The drive starts in the negative direction with acceleration I39 and velocity I32.
2. It changes its direction upon reaching the torque/force stop and after the time stored in I29 has passed, and continues its run with velocity I33 until it reaches the next zero pulse.
3. The current actual position is set to the value of reference position I34 upon reaching the zero pulse.
4. The set values are set to the value 0 with deceleration I39.
5. If I43 is set to 1: Active, the drive is positioned at reference position I34.

6.6.2 Reference position

Depending on the referencing type I30, the actual position I80 is replaced by the reference position I34 during the referencing event.

6.6.3 Reference retaining

STOBER offers a convenient drive-based referencing system. Depending on the encoder model used and the reference management, different types of reference retaining (I46) are offered.

6.7 Drive Based device control

Drive Based device control is based on DRIVECOM profile drive technology 21; this profile describes the control sequence of a drive controller based on a device state machine. Each device state here represents a certain behavior that can be changed only through defined events. These events are assigned to individual state transitions.

For example, device control regulates the device start-up when turning on the supply voltage, the enable state of the drive controller, the process of a quick stop, the behavior in case of fault and the restart after a fault.

Some of the conditions and responses coupled with the state transitions can be influenced specific to the user. For example, it is possible to define the end of a quick stop or enable delays in a way that is tailored to the respective application case (see wizard Drive Based device control).

The following chapters describe the device states and the possible state transitions associated with them. You will also learn whether certain actions are necessary on your part to reach the individual device states, what those actions are and which user-specific factors you can parameterize yourself.

6.7.1 Drive Based device state machine

The device state machine describes the different device states of the drive controller along with the possible state changes.

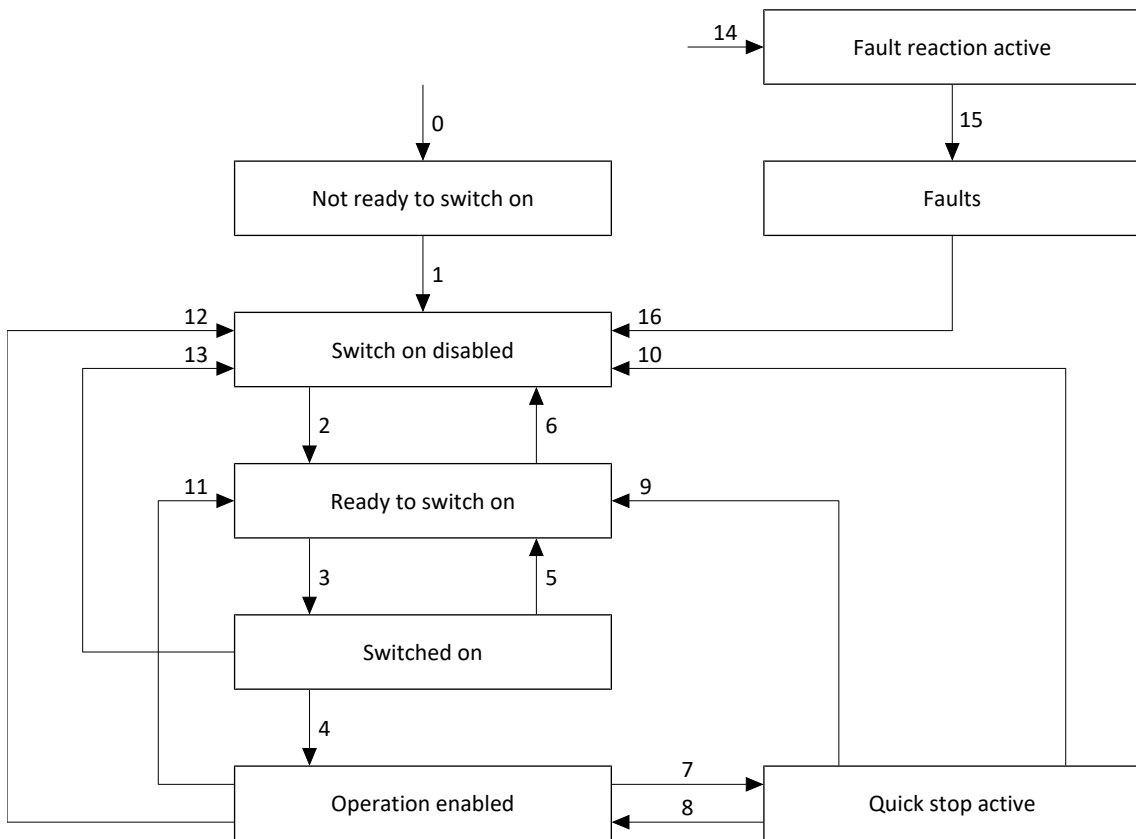


Fig. 23: Drive Based device state machine: device states and state changes

6.7.2 States, transitions and conditions

The states provided by Drive Based device control all have specific features. A state either transitions into another state automatically or requires certain actions from the user. There are also transition conditions that you can parameterize to be tailored to your application case using the Drive Based device control wizard.

Parameter E48 shows the current device state of a drive controller.

6.7.2.1 Not ready to switch on

Features

- The drive is being initialized, tested and associated parameters are being loaded
- The power unit as well as drive and activation functions are locked
- Brakes remain engaged

Transition to Switch on disabled (1)

The drive controller switches to Switch on disabled (1) automatically after successfully completing the self-test.

6.7.2.2 Switch on disabled

Features

- Initialization has been completed
- Self-test was successfully completed
- The power unit as well as drive and activation functions are locked
- Brakes are or remain engaged in the event of an inactive release override

Possible causes of a switch-on lockout include:

1. Insufficient or no supply voltage or deactivated DC link supply
2. Axis switching is active
3. Projected IGB motion bus is not active
4. STO safety function is active

Transition to Ready to switch on (2)

The exact cause of a switch-on lockout can be found in parameters E47 and E49.

The transition to Ready to switch on happens automatically upon remedying the relevant cause of the switch-on lockout.

Application-specific parameterization

A34 Auto start:

If this parameter is activated, the drive controller switches directly to the state `Operation enabled` if an enable signal is present.

WARNING!

Injury to persons and material damage due to unexpected motor startup!

Only activate autostart if the standards and regulations applicable to the system or machine in question permit a direct switch to the `Operation enabled` device state.

- In accordance with EN 61800-5-1, clearly mark an activated autostart on the system and in the associated system documentation.

6.7.2.3 Ready to switch on

Features

- Power unit and drive function are disabled
- Drive controller is ready to switch on
- Brakes are or remain engaged in the event of an inactive release override

Transitions to Switched on (3) or Switch on disabled (6)

Activate the enable signal (3) or one of the possible causes of a switch-on lockout is present (6).

Application-specific parameterization

In addition to the enable signal at terminal X1, the Drive Based device state machine also supports the use of an additional enable signal, for example over a fieldbus or using digital inputs. The enable signal and additional enable signals are coupled by an AND link. In this case, the drive controller is enabled only if the value of both signals is `HIGH`.

A60 Source additional enable:

If you are working with an additional enable signal, define the data source of the additional signal in this parameter. In the case of a fieldbus connection, A180 bit 0 supplies the additional enable signal.

6.7.2.4 Switched on

Features

- Power unit is ready for operation
- Drive function is locked; set values are not processed
- Brakes are or remain engaged in the event of an inactive release override

Transition to Operation enabled (4), Ready to switch on (5) or Switch on disabled (13)

Leave the enable signal active (4) or deactivate it (5). If one of the possible causes of a switch-on lockout is present, the drive controller automatically switches to Switch on disabled (13).

6.7.2.5 Operation enabled

Features

- Power unit is switched on
- Drive function is enabled, set values are processed

Transitions to Quick stop active (7), Ready to switch on (11) or Switch on disabled (12)

Request a quick stop (7) or deactivate the enable signal (11). If one of the possible causes of switch-on lockout is present, the drive controller automatically switches to Switch on disabled (12).

Application-specific parameterization

- A43 Switch-off delay X1 enable:
If the enable signal at terminal X1 of a safety relay is affected by OSSD test pulses, the drive controller cannot differentiate these signals from a functional enable-off. An incorrect interpretation of the test pulses can be avoided by defining a delay time in this parameter that is greater than the duration of the test pulses. This then eliminates unexpected switching of the drive controller to the states Ready to switch on or Quick stop active (see parameter A44).
- A44 Quick stop at enable off:
This parameter results in a drive being stopped with a quick stop in case of an inactive enable. In this case, the drive controller switches to the state quick stop active. If parameter A44 is not activated, the drive switches to the Ready to switch on state in case of Enable-off.

6.7.2.6 Quick stop active

Features

- Power unit is switched on; drive function is enabled
- A quick stop is executed
- Brakes remain released, brake engages at the end of the quick stop or at standstill
- Brake engaging ends the quick stop

Transition to Operation enabled (8), Ready to switch on (9) or Switch on disabled (10)

Cancel the quick stop request (8). If the quick stop was carried out fully while the enable signal is inactive, the drive controller automatically switches to Ready to switch on (9); if one of the possible causes of a switch-on lockout is present, the drive controller automatically switches to Switch on disabled (10).

Application-specific parameterization

- A39 Maximum quick stop duration for enable off:
If the drive is stopped with a quick stop in the case of an inactive enable signal (see A44), define the maximum time after which the power unit is switched off in this parameter.
- A45 Quick stop end:
Define whether a quick stop is ended with a standstill of the drive or by canceling the quick stop request in this parameter.
- A62 Source /quick stop:
Define how a requested quick stop is triggered in this parameter.

6.7.2.7 Fault reaction active

Features

- A drive error has occurred
- Fault response is executed depending on the respective fault event
- Brakes are activated depending on the respective fault response

Transition to fault (15)

After completed fault response, the drive controller switches to Fault (15) automatically.

Application-specific parameterization

A29 Quick stop in case of fault:

If this parameter is activated, the drive is stopped with a quick stop, if possible, in case of fault; if the parameter is inactive, the movement of the axis is no longer controlled by the drive controller.

6.7.2.8 Fault

Features

- A drive error has occurred
- The fault response is finished
- The power unit as well as drive and activation functions are locked
- Brakes are engaged in the event of an inactive release override

Transition to Switch on disabled (16)

After remedying the fault cause and then acknowledging the fault message, the drive controller automatically switches to Switch on disabled (16).

6.8 Motion commands

The operating modes of the application feature special motion commands that are based on the PLCopen standard and are supplemented by two manufacturer-specific motion commands (MC_DoNothing and MC_MoveSpeed). Any motion command – except MC_Stop – can be interrupted during execution. To be able to execute a motion command, the following requirements must be met:

- Local and jog mode must not be activated
- A drive controller must not be in the Switch on disabled or Fault device state

| Command | Description | Control type | Necessary motion variables |
|---------------------|-------------------------------------------------------------------------------------------------------------|--------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0: MC_DoNothing | — | — | — |
| 8: MC_MoveSpeed | Axis moves without end with set velocity (without position control) | Velocity | <ul style="list-style-type: none"> ▪ Velocity, override ▪ Acceleration ▪ Deceleration ▪ Jerk |
| 4: MC_MoveVelocity | Axis moves without end with set velocity (with position control) | Position | <ul style="list-style-type: none"> ▪ Velocity, override ▪ Acceleration ▪ Deceleration ▪ Jerk |
| 9: MC_TorqueControl | Axis moves without end with set torque/force | Torque/force | <ul style="list-style-type: none"> ▪ Torque/force |
| 2: MC_MoveRelative | Axis moves a relative distance; set position is relative to the actual position at the start of the command | Position | <ul style="list-style-type: none"> ▪ Position ▪ Velocity, override ▪ Acceleration ▪ Deceleration ▪ Jerk |
| 3: MC_MoveAdditive | Axis moves a relative distance; set position is relative to the set position of the previous motion command | Position | <ul style="list-style-type: none"> ▪ Position ▪ Velocity, override ▪ Acceleration ▪ Deceleration ▪ Jerk |
| 1: MC_MoveAbsolute | Axis moves to an absolute set position (referencing necessary) | Position | <ul style="list-style-type: none"> ▪ Position ▪ Velocity, override ▪ Acceleration ▪ Deceleration ▪ Jerk |
| 6: MC_Home | Axis is referenced | Depending on the selected referencing type | <ul style="list-style-type: none"> ▪ Velocity, override ▪ Acceleration ▪ Deceleration ▪ Jerk ▪ Torque/force |

| Command | Description | Control type | Necessary motion variables |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------|--------------|---------------------------------------------------------------------------------------------------------------------------------------|
| 11: MC_Halt | Axis is stopped; next command can be run before standstill | Velocity | <ul style="list-style-type: none"> ▪ Deceleration ▪ Jerk |
| 5: MC_Stop | Axis is brought to a standstill; next command can be executed after standstill | Velocity | <ul style="list-style-type: none"> ▪ Deceleration ▪ Jerk |
| 12: MC_GearIn | Axis moves synchronously with a master; velocity- based coupling (electronic gear unit) | Position | <ul style="list-style-type: none"> ▪ Acceleration ▪ Deceleration ▪ Jerk |
| 14: MC_GearInPosition | Axis moves synchronously with a master; position- based coupling (electronic gear unit) | Position | <ul style="list-style-type: none"> ▪ Position, master position |
| 13: MC_GearOut | Axis ends the ongoing synchronous motion and continues with the last synchronous velocity (electronic gear unit) | Position | <ul style="list-style-type: none"> ▪ Velocity override ▪ Acceleration ▪ Deceleration ▪ Jerk |

Tab. 6: Drive Based Synchronous: Available motion commands

For detailed information on the motion commands, refer to the associated manual (see [Detailed information \[▶ 157\]](#)).

6.9 Continue function

In Drive Based Synchronous type applications, the continue function allows you to resume interrupted motion commands, for example after a quick stop, the triggering of safety technology or after an event with a fault reaction. Using the continue function, you can also continue motion commands 2: MC_MoveRelative and 3: MC_MoveAdditive and end with the desired set position, which would not be possible when restarting using the execute signal.

Cancellation causes that can be continued

- Event with fault response
- Enable-off
- Motion command 5: MC_Stop, 11: MC_Halt
- Stop signal
- Quick stop signal
- SS1, SS2
- Jog

Motion commands

| Motion command (I401) | Continue | Note |
|-----------------------|----------|---------------------------------------------------------------------------------------------------------------------------------|
| 0: MC_DoNothing | – | – |
| 1: MC_MoveAbsolute | ✓ | Motion continues until the set position is reached; set position is absolute |
| 2: MC_MoveRelative | ✓ | Motion continues until the set position is reached; set position is relative to actual position at the start of the command |
| 3: MC_MoveAdditive | ✓ | Motion continues until the set position is reached; set position is relative to the set position of the previous motion command |
| 4: MC_MoveVelocity | ✓ | Motion continues without end at set velocity |
| 5: MC_Stop | – | – |
| 6: MC_Home | – | – |
| 8: MC_MoveSpeed | ✓ | Motion continues without end at set velocity |
| 9: MC_TorqueControl | ✓ | Motion continues without end with set torque/force |
| 11: MC_Halt | – | – |
| 12: MC_GearIn | – | – |
| 13: MC_GearOut | – | – |
| 14: MC_GearInPosition | – | – |

Motion block operating mode

In motion block operating mode, the digital inputs, control byte of motion block operating mode or Motion block control panel can serve as the source for the continue signal (control byte: J01; continue source: J917). The continue signal not only continues the interrupted motion command, but also the entire motion block chain.

| Cause of cancellation | Continue | Note |
|----------------------------------------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Event with fault response | ✓ | – |
| Enable-off | ✓ | – |
| Motion command 5: MC_Stop | – | Motion command 5: MC_Stop can be part of a motion block chain |
| Motion command 11: MC_Halt | ✓ | Motion command 11: MC_Halt can be part of a motion block chain; continue is only possible with stop signal via control byte (prerequisite: J06 = 2: Parameter J01) |
| Stop signal (I440 = 0: Inactive) | – | Stop signal is operated from the control panel in motion block operating mode and is used to stop and end the motion block chain |
| Quick stop signal (A302 = 0: Inactive) | ✓ | – |
| SS1 | ✓ | – |
| SS2 | ✓ | – |
| Jog (I443 = 1: Active) | – | – |

Motion block reset

You can use the continue function to continue an interrupted motion block if both the motion command and the cause of cancellation can be continued. If it is not possible to resume the motion block or you do not want to continue the interrupted motion block, it always has to be deleted from the continue memory by means of a reset before a new motion block can be started using the execute signal.

6.10 Electronic nameplate

STOBER synchronous servo motors are generally equipped with EnDat encoders that provide a special memory. This memory includes the electronic nameplate, i.e. all type-relevant master data as well as special mechanical and electronic values of a motor.

When you operate a drive controller with a STOBER synchronous servo motor and an EnDat encoder, the electronic nameplate is read and all data transferred if the drive controller is connected online. This specifically includes job data, current controller values, motor parameters, data on motor temperature sensors, brakes, the commutation offset and the measured electromotive force. These can be manually optimized later and resaved in the drive controller.

Every time a drive controller is restarted, it checks whether the projected motor, the brake (if applicable), the motor temperature sensor or commutation have been changed. If so, the changed data is read out. The optimizations saved in the drive controller are maintained whenever possible.

The automatic read-out of the electronic nameplate is activated at the factory (B04 Electronic nameplate = 64: Active).

6.11 Lag monitoring

In Drive Based Synchronous type applications, you can monitor the lag of the axis with active position control in order to detect increasing position deviations at an early stage. Monitoring the lag allows you to react quickly before material damage can occur, e.g. in the event of stiffness or a mechanical blockage of the output.

To monitor the lag, the difference between the actual position of the axis and the set position x_{2_set} of the control is determined and compared with the maximum permissible lag (actual position: I80; set position: I96; permissible lag: I21; result: I187). If the permitted lag is exceeded, event 54: Following error is triggered with the corresponding protection level (protection level: U22).

7 Appendix

7.1 Standard mapping of Drive Based Synchronous

Information

For fieldbus communication via PROFINET, the processing of process data in some controllers is WORD-oriented (16 bits). In Drive Based applications, the standard mapping is pre-assigned appropriately. If changes are made to the standard mapping, take the parameter data type that you are adding to or removing from the mapping into account.

If you add or remove parameters of the BYTE or INT8 data type (8 bit), this can cause problems in the data structures of the controller. If necessary, use parameter A101 Dummy byte to fill the 8-bit gaps that result in the process data and implement the necessary data structure for the controller.

7.1.1 SD6: Drive Based Synchronous receive process data

The following table shows the standard mapping for the receive process data in the Drive Based and Drive Based Synchronous applications. For fieldbus communication via PROFINET, the process data mapping is done via parameter A90 – A92. For fieldbus communication via EtherCAT, the process data mapping is done via parameter A225 – A228. If necessary, the standard mapping can be customized.

| Byte | Data type | Name | Parameters |
|---------|-----------|---------------------------|------------|
| 0 | BYTE | Device control byte | A180 |
| 1 | BYTE | Control byte command | 1.J37 |
| 2 – 3 | WORD | Control word application | I210 |
| 4 | INT8 | Command | 1.J40 |
| 5 | INT8 | Motion-ID | 1.J41 |
| 6 – 9 | INT32 | Position | 1.J42 |
| 10 – 13 | REAL32 | Velocity 1 | 1.J43 |
| 14 – 17 | REAL32 | Velocity Override | 1.J56 |
| 18 – 21 | REAL32 | Torque/Force reference | 1.G469 |
| 22 | BYTE | Control byte motion block | J01 |
| 23 – 24 | INT16 | Reference motion block | J02 |

Tab. 7: SD6: receive process data (standard mapping)

7.1.2 SD6: Drive Based Synchronous transmit process data

The following table shows the standard mapping for the transmit process data in the Drive Based and Drive Based Synchronous applications. For fieldbus communication via PROFINET, the process data mapping is done via parameter A94 – A96. For fieldbus communication via EtherCAT, the process data mapping is done via parameter A233 – A236. If necessary, the standard mapping can be customized.

| Byte | Data type | Name | Parameters |
|---------|-----------|--------------------------|------------|
| 0 | BYTE | Status byte device | E200[0] |
| 1 | BYTE | Status byte device | E200[1] |
| 2 – 3 | WORD | Status word 2 | E201 |
| 4 | BYTE | Status byte application | 1.I212 |
| 5 | BYTE | Status byte command | 1.J39 |
| 6 – 7 | WORD | Status word application | I200 |
| 8 – 11 | INT32 | Current position | 1.I80 |
| 12 – 15 | REAL32 | Actual speed | 1.I88 |
| 16 – 19 | REAL32 | Actual torque/force | E90 |
| 20 – 21 | WORD | Status word user-defined | A67 |
| 22 | INT8 | Operating condition | E80 |
| 23 | BYTE | Status byte motion block | J302 |
| 24 – 25 | INT16 | Actual motion block | J300 |
| 26 | INT8 | Device control state | E48 |

Tab. 8: SD6: transmit process data (standard mapping)

7.2 Detailed information

The documentation listed below provides you with further relevant information on the 6th STOBER drive controller generation. The current status of the documentation can be found in our download center at:

<http://www.stoeber.de/en/downloads/>.

Enter the ID of the documentation in the search field.

| Title | Documentation | Contents | ID |
|-------------------------------------------------------------|---------------|-----------------------------------------------------------------------------------------------------------------------------------------|--------|
| SD6 drive controller | Manual | System design, technical data, project configuration, storage, installation, connection, commissioning, operation, service, diagnostics | 442426 |
| PROFINET communication – SD6 | Manual | Installation, electrical installation, data transfer, commissioning, diagnostics, detailed information | 442710 |
| EtherCAT communication – SD6 | Manual | Installation, electrical installation, data transfer, commissioning, diagnostics, detailed information | 442516 |
| SE6 safety technology – Safe drive monitoring via terminals | Manual | Technical data, installation, commissioning, diagnostics | 442796 |
| ST6 safety technology – STO via terminals | Manual | Technical data, installation, commissioning, diagnostics, detailed information | 442478 |
| Motion commands | Manual | Control and status information, denials and limitations, motion | 443349 |

7.3 Abbreviations

| Abbreviation | Meaning |
|--------------|--------------------------------------------|
| AI | Analog Input |
| ALT | Alternative |
| CAN | Controller Area Network |
| CB | Controller Based |
| CiA | CAN in Automation |
| CNC | Computerized Numerical Control |
| DI | Digital Input |
| EtherCAT | Ethernet for Control Automation Technology |
| IGB | Integrated Bus |
| LinM | Linear Motor |
| LS | Limit Switch |
| LSB | Least Significant Bit |
| M | Motor |
| M/F | Torque or force |
| MEnc | Motor Encoder |
| PEnc | Position Encoder |
| PROFINET | Process Field Network |
| RS | Reference Switch |
| S | Switch |
| PLC | Programmable Logic Controller |
| VM | Virtual Master |
| ZP | Zero Pulse |

8 Contact

8.1 Consultation, service and address

We would be happy to help you!

We offer a wealth of information and services to go with our products on our website:

<http://www.stoeber.de/en/service>

For additional or personalized information, contact our consultation and support service:

<http://www.stoeber.de/en/support>

If you need our system support:

Phone +49 7231 582-3060

systemsupport@stoeber.de

If you need a replacement device:

Phone +49 7231 582-1128

replace@stoeber.de

Call our 24-hour service hotline:

Tel. +49 7231 582-3000

Our address:

STÖBER Antriebstechnik GmbH + Co. KG

Kieselbronner Strasse 12

75177 Pforzheim, Germany

8.2 Your opinion is important to us

We created this documentation to the best of our knowledge with the goal of helping you build and expand your expertise productively and efficiently with our products.

Your suggestions, opinions, wishes and constructive criticism help us to ensure and further develop the quality of our documentation.

If you want to contact us for a specific reason, we would be happy to receive an e-mail from you at:

documentation@stoeber.de

Thank you for your interest.

Your STOBER editorial team

8.3 Close to customers around the world

We offer you committed, expert advice and support in over 40 countries worldwide:

STOBER AUSTRIA

www.stoerber.at
+43 7613 7600-0
sales@stoerber.at

STOBER FRANCE

www.stoerber.fr
+33 478 98 91 80
sales@stoerber.fr

STOBER HUNGARY

www.stoerber.de
+36 53 5011140
info@emtc.hu

STOBER JAPAN

www.stoerber.co.jp
+81-3-5875-7583
sales@stoerber.co.jp

STOBER TAIWAN

www.stoerber.tw
+886 4 2358 6089
sales@stoerber.tw

STOBER UK

www.stoerber.co.uk
+44 1543 458 858
sales@stoerber.co.uk

STOBER CHINA

www.stoerber.cn
+86 512 5320 8850
sales@stoerber.cn

STOBER Germany

www.stoerber.de
+49 4 7231 582-0
sales@stoerber.de

STOBER ITALY

www.stoerber.it
+39 02 93909570
sales@stoerber.it

STOBER SWITZERLAND

www.stoerber.ch
+41 56 496 96 50
sales@stoerber.ch

STOBER TURKEY

www.stoerber.com
+90 216 510 2290
sales-turkey@stoerber.com

STOBER USA

www.stoerber.com
+1 606 759 5090
sales@stoerber.com

List of figures

| | | |
|---------|---------------------------------------------------------------------------|-----|
| Fig. 1 | DS6: Program interface | 12 |
| Fig. 2 | DriveControlSuite: Navigation using text links and symbols | 14 |
| Fig. 3 | Motion block operating mode: Signal flows..... | 60 |
| Fig. 4 | Command operating mode: signal flows..... | 78 |
| Fig. 5 | Components and configuration steps | 84 |
| Fig. 6 | Continuous Function Chart: analog input AI1 (F116 = 0: -10V to 10V) | 88 |
| Fig. 7 | Continuous Function Chart: analog input AI1 (F116 = 1: 0 to 20mA)..... | 88 |
| Fig. 8 | Continuous Function Chart: analog input AI1 (F116 = 2: 4 to 20mA)..... | 88 |
| Fig. 9 | Continuous Function Chart: analog input AI2 | 89 |
| Fig. 10 | Continuous Function Chart: analog input AI3 | 89 |
| Fig. 11 | Additional counter function: Example | 90 |
| Fig. 12 | Motor potentiometer: Linear and stepwise calculation | 95 |
| Fig. 13 | Additional phasing function: Example..... | 104 |
| Fig. 14 | Synchronous operation: Model 1..... | 107 |
| Fig. 15 | Synchronous operation: Model 2..... | 108 |
| Fig. 16 | Synchronous operation: Model 3..... | 109 |
| Fig. 17 | Endless-rotational motion: Rotary table | 111 |
| Fig. 18 | Limited-rotational motion: Pointer | 111 |
| Fig. 19 | Endless translational motion: Conveyor belt | 112 |
| Fig. 20 | Limited translational motion: Tool slide..... | 112 |
| Fig. 21 | Limited translational motion: Linear motor | 112 |
| Fig. 22 | Hardware limit switch memory..... | 117 |
| Fig. 23 | Drive Based device state machine: device states and state changes..... | 145 |

List of tables

| | | |
|--------|----------------------------------------------------------------|-----|
| Tab. 1 | Parameter groups..... | 15 |
| Tab. 2 | Parameters: data types, parameter types, possible values | 16 |
| Tab. 3 | Parameter types..... | 17 |
| Tab. 4 | Indirect trigger: Notations..... | 72 |
| Tab. 5 | Referencing methods | 122 |
| Tab. 6 | Drive Based Synchronous: Available motion commands | 150 |
| Tab. 7 | SD6: receive process data (standard mapping)..... | 156 |
| Tab. 8 | SD6: transmit process data (standard mapping)..... | 156 |

Glossary

Broadcast domain

Logical grouping of network devices within a local network that reaches all nodes via broadcast.

Control cascade

Complete model of the control structure with the position controller, velocity controller and current controller components.

Electronic nameplate

The synchronous servo motors are generally equipped with absolute encoders that provide special memory. This memory includes the electronic nameplate, i.e. all type-relevant master data as well as special mechanical and electronic values of a motor. When you operate a drive controller with a synchronous servo motor and an absolute encoder, the electronic nameplate is read and all motor data transferred if the drive controller is connected online. The drive controller automatically determines the associated limit values and control parameters from this data.

IPv4 limited broadcast

Type of broadcast in a network with IPv4 (Internet Protocol version 4). The IP address 255.255.255.255 is entered as the destination. The content of the broadcast is not forwarded by a router, which limits it to the local network.

PID controller

Universal controller type with a P-, I- and D-share. These three adjustment parameters make the controller flexible and ensure exact and highly dynamic control. However, by implication, it also necessitates a wide variety of variants. It is that much more important to ensure careful construction that is well-coordinated to the system. The application areas for this controller type are control circuits with systems of the second order and higher, which must be stabilized quickly and do not allow for any permanent control deviation.

PROFINET

Open Ethernet standard of PROFIBUS Nutzerorganisation e. V. (PNO) for automation.



4 4 3 0 4 6 . 0 3

05/2024

STÖBER Antriebstechnik GmbH + Co. KG
Kieselbronner Str. 12
75177 Pforzheim
Germany
Tel. +49 7231 582-0
mail@stoeber.de
www.stober.com

24 h Service Hotline
+49 7231 582-3000

www.stober.com