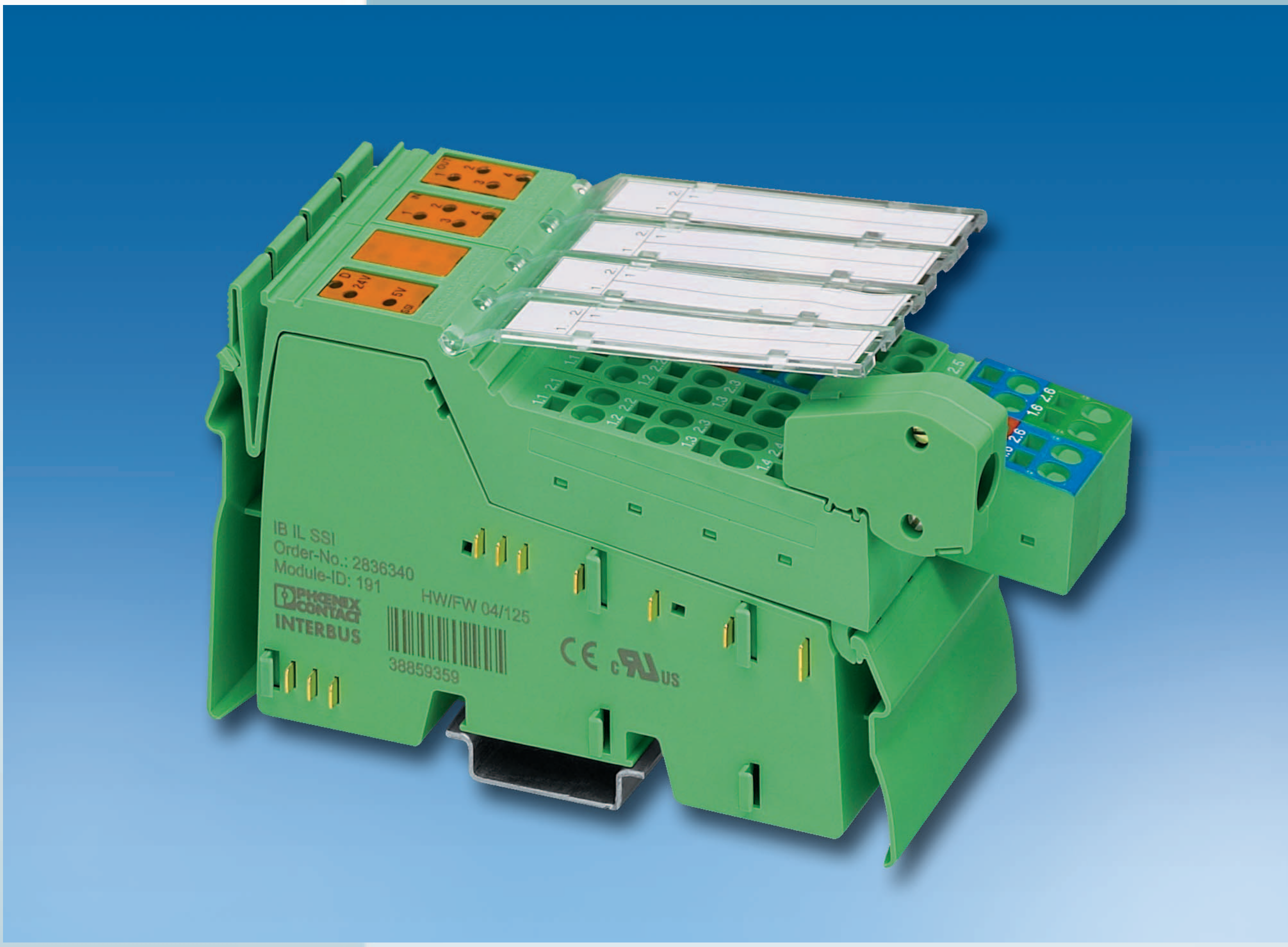
AUTOMATIONWORX



User Manual

IB IL SSI UM E

Order No.: 2698465

Positioning Terminal for Absolute Encoders
IB IL SSI, IB IL SSI-PAC and IB IL SSI-2MBD

PHOENIX CONTACT
INSPIRING INNOVATIONS

AUTOMATIONWORX

User Manual

Positioning Terminal for Absolute Encoders IB IL SSI, IB IL SSI-PAC and IB IL SSI-2MBD

03/2007

Designation: IB IL SSI UM E

Revision: 01

Order No.: 2698465

This user manual is valid for:

Designation	Order No.
IB IL SSI	2836340
IB IL SSI-PAC	2861865
IB IL SSI-2MBD	2855729

Please Observe the Following Notes

In order to ensure the safe use of the product described, we recommend that you read this manual carefully. The following notes provide information on how to use this manual.

User Group of This Manual

The use of products described in this manual is oriented exclusively to

– qualified electricians or persons instructed by them, who are familiar with applicable standards and other regulations regarding electrical engineering and, in particular, the relevant safety concepts.
– qualified application programmers and software engineers, who are familiar with the safety concepts of automation technology and applicable standards.

Phoenix Contact accepts no liability for erroneous handling or damage to products from Phoenix Contact or third-party products resulting from disregard of information contained in this manual.

Explanation of Symbols Used

The *attention* symbol refers to an operating procedure which, if not carefully followed, could result in damage to hardware and software or personal injury.

The *note* symbol informs you of conditions that must strictly be observed to achieve error-free operation. It also gives you tips and advice on the efficient use of hardware and on software optimization to save you extra work.

The *text* symbol refers to detailed sources of information (manuals, data sheets, literature, etc.) on the subject matter, product, etc. This text also provides helpful information for the orientation in the manual.

We Are Interested in Your Opinion

We are constantly striving to improve the quality of our manuals.

Should you have any suggestions or recommendations for improvement of the contents and layout of our manuals, please send us your comments.

PHOENIX CONTACT GmbH & Co. KG
Documentation Services
32823 Blomberg
Germany

Phone +49 - 52 35 - 30 0
Fax +49 - 52 35 - 34 20 21
E-mail tecdoc@phoenixcontact.com

General Terms and Conditions of Use for Technical Documentation

Phoenix Contact GmbH & Co. KG reserves the right to alter, correct, and/or improve the technical documentation and the products described in the technical documentation at its own discretion and without giving any notice.

The provision of technical documentation (in particular data sheets, installation instructions, manuals, etc.) does not constitute any further duty on the part of Phoenix Contact GmbH & Co. KG to furnish information on alterations to products and/or technical documentation. Any other agreement shall only apply if expressly confirmed in writing by Phoenix Contact GmbH & Co. KG. Please note that the supplied documentation is product-specific documentation only.

Although Phoenix Contact GmbH & Co. KG makes every effort to ensure that the information content is accurate, up-to-date, and state-of-the-art, technical inaccuracies and/or printing errors in the information cannot be ruled out.
Phoenix Contact GmbH & Co. KG does not offer any guarantees as to the reliability, accuracy or completeness of the information provided. Phoenix Contact GmbH & Co. KG accepts no liability or responsibility for errors or omissions in the content of the technical documentation (in particular data sheets, installation instructions, manuals, etc.).

As far as is permissible by applicable jurisdiction, no guarantee or claim for liability for defects whatsoever shall be granted in conjunction with the information available in the technical documentation, whether expressly mentioned or implied. This information does not include any guarantees on quality, does not describe any fair marketable quality and does not make any claims as to quality guarantees or guarantees on the suitability for a special purpose.
Phoenix Contact GmbH & Co. KG reserves the right to alter, correct, and/or improve the information and the products described in the information at its own discretion and without giving any notice.

Internet

Up-to-date information on Phoenix Contact products can be found on the Internet at:

www.phoenixcontact.com

Make sure you always use the latest documentation.
It can be downloaded at:

www.download.phoenixcontact.com

A conversion table is available on the Internet at:

www.download.phoenixcontact.com/general/7000_en_00.pdf

Table of Contents

1 Function and Structure of the Positioning Terminal

☞ The IB IL SSI and IB IL SSI-PAC terminals only differ in the scope of supply (see Section 7.3, "Ordering Data"). Their function and technical data are identical.
Other than that, the IB IL SSI-2MBD terminal is available with a transmission speed of 2 Mbps.
For greater clarity, the designation IB IL SSI ... is used throughout this document.

1.1 Function Description

The IB IL SSI ... terminal belongs to the Inline product range. It is a universal positioning module designed for use within an Inline station.

The IB IL SSI ... terminal can be used for point-to-point positioning on both rotary and linear axes. In this context, it has sole responsibility for drive control in respect of approaching the required position. The position values are either assigned directly by the higher-level open-loop control or up to two setpoint positions are stored in the terminal and these are approached automatically by the module following the call.

As the digital 24 V outputs supply the control signals for the drive (forward/reverse in addition to rapid motion/creeping motion), both electric (AC or DC) and pneumatic and hydraulic drives can be controlled.

The IB IL SSI ... is suitable for applications in which the drive is not to apply any force in the target position. This is the case, for example, on a conveyor container positioned for loading and on which, during loading, no forces are applied in either the direction of transport or the opposite direction. Nor do drives need to apply forces in the target position if gears or spindles are used in an application for what is known as self-locking.

A prerequisite for the use of the IB IL SSI ... is that absolute encoders (linear or rotary encoders) are used for position detection. Symmetrical encoders with a 5 V signal and 5 V or 24 V supply can be connected. The associated limit switches can be used as an option. Four digital 24 V outputs enable the drives to respond quickly to prevailing conditions.

Positioning terminals are often used in woodworking machinery and in paper processing and conversion, as well as in packaging machinery and for metalworking. As well as supporting the positioning of transport equipment such as conveyor belts, chain conveyors or lifting gear, the terminals can also be used for format adjustments affecting limit stops, guideways or pressure rollers.

The IB IL SSI ... is a compact positioning terminal for an axis. It can be used to create non-bus-specific control system solutions for point-to-point positioning according to the rapid motion/creeping motion principle (also known as switching axes).

In addition to the complete function logic for the control system for the positioning process, the IB IL SSI ... positioning terminal supports actual-position detection (input for absolute encoder) and drive activation (four digital inputs).

The higher-level control system assigns a target position to the IB IL SSI ... positioning terminal and sends the start signal for positioning. The positioning terminal then carries out positioning automatically and confirms the result to the higher-level control system. For confirmation to be positive, the axis must have been able to actually stop in the target window on stopping.

The point-to-point positioning terminal can be operated on any fieldbus for which there is an Inline bus coupler, e.g., INTERBUS, CANopen, PROFIBUS, Ethernet, and DeviceNet. It can be installed anywhere on the Inline station.

Its sister terminal, the IB IL INC, which has been designed for the connection of incremental encoders, is also available.

The positioning terminal can be used with any absolute encoder with an SSI interface. It has been designed for the connection of an encoder. The terminal supplies the encoder with 5 V DC and 24 V DC.

The encoder supply is taken from the Inline U_M main circuit. A short-circuit-proof power supply provides +5 V DC or +24 V DC to the connection terminals for the encoder supply. This enables both the initiator voltage and the output voltage to be monitored.

The isolation between the encoder input and bus interface is designed for a test voltage of 500 V AC.

The terminal has three digital inputs of 24 V DC, four digital outputs of 24 V DC, 500 mA and a connection for enabling the hand-held operator panel (HHOP). Limit switches can be connected to the digital inputs in positioning mode. The signals for activating the drive (e.g., for direction of travel and speed of travel) are output directly via the digital outputs.

If you want to use the positioning terminal to simply determine the position of a drive, subject to certain restrictions, you can use the three inputs and four outputs as digital inputs and outputs independent of all positioning functions (see "Using the Terminal for Position Detection" on page 6-6).

The terminal is configured (encoder data, resolution, etc.) and parameterized (start window, pre-stop window, etc.) via the bus system. Once these settings have been made, the required position values to be approached automatically by the terminal once it has been called can be assigned via the control program. If the application permits, up to two parameter records can be saved. The module monitors the positioning and sends a status message to the control system. In the event of an error, the drive is stopped immediately. Following configuration, the terminal operates independently of the bus and control system.

Application examples

- Format adjustments
- Compound table positioning
- Position monitoring at valves
- Position control in transport systems (goods, containers)
- Positioning of lifting systems

In the case of many machines, there is a cost problem: In order to increase flexibility, automated axes are being used more and more frequently. This increases the cost of equipping and commissioning positioning control systems so significantly that the machines are no longer able to compete on the global market. One of the reasons for this is that excessively powerful and, therefore, overly expensive control systems are required for simple motion control tasks, such as those involving the positioning of:

– Transport equipment (conveyor belts, cranes, chain and monorail conveyors, etc.)
– Format adjustments (e.g., settings for limit stops, guideways and rollers)
– Tools (spindles, saw blades, cutting and bending tools)

The IB IL SSI ... can provide a cost-effective solution for many of these positioning applications. This can prove particularly worthwhile if the axes have gears.

What the terminal looks like

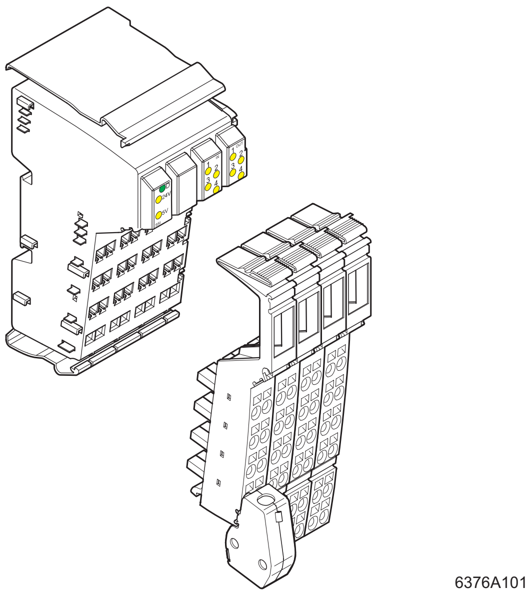


Figure 1-1 IB IL SSI ... terminal with associated connectors

1.2 Terminal Structure

1.2.1 Housing Dimensions

Small I/O stations are frequently installed in standard control boxes with a depth of 80 mm. The Inline terminal has been designed for use in this type of control box.

The housing dimensions of the positioning terminal are determined by the dimensions of the electronics base and those of the connector.

The electronics base for the terminal has an overall width of 48.8 mm.

It accepts four 12.2 mm wide connectors.

With connectors, the terminal is 71.5 mm deep and 132 mm high (height of the shield connector).

The IB IL SSI/INC-PLSET connector set is used on the positioning terminal. It contains a shield connector, a standard connector and two extended double signal connectors.

Electronics base

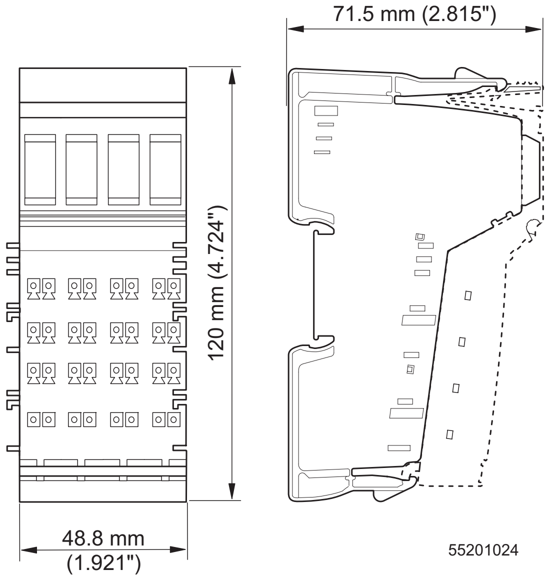


Figure 1-2 Electronics base dimensions

Connectors

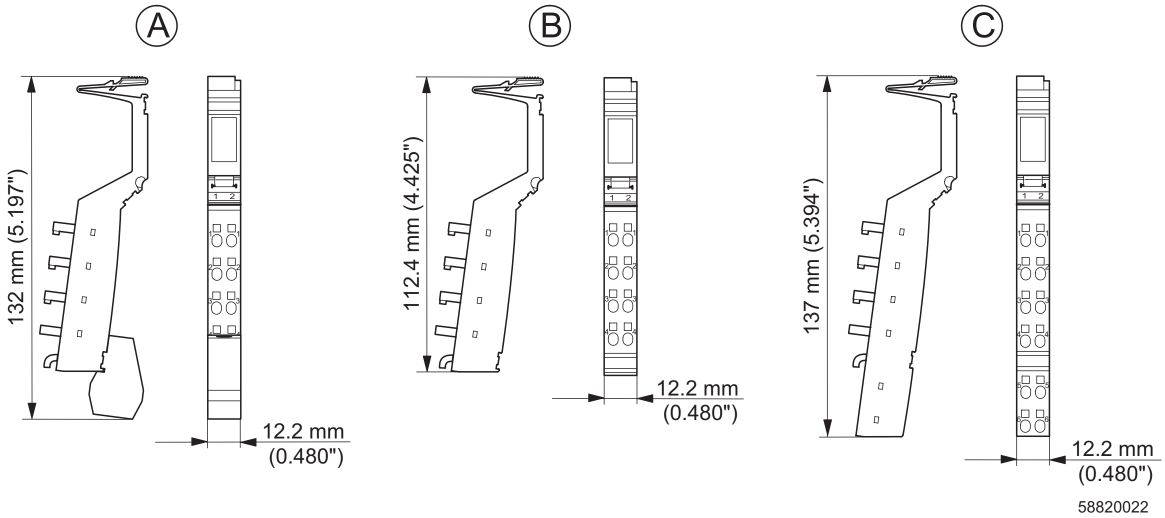


Figure 1-3 Connector dimensions

Key:

A	Shield connector
B	Standard connector
C	Extended double signal connector

The depth of the connector does not influence the overall depth of the terminal.

1.2.2 Terminal Point Assignment

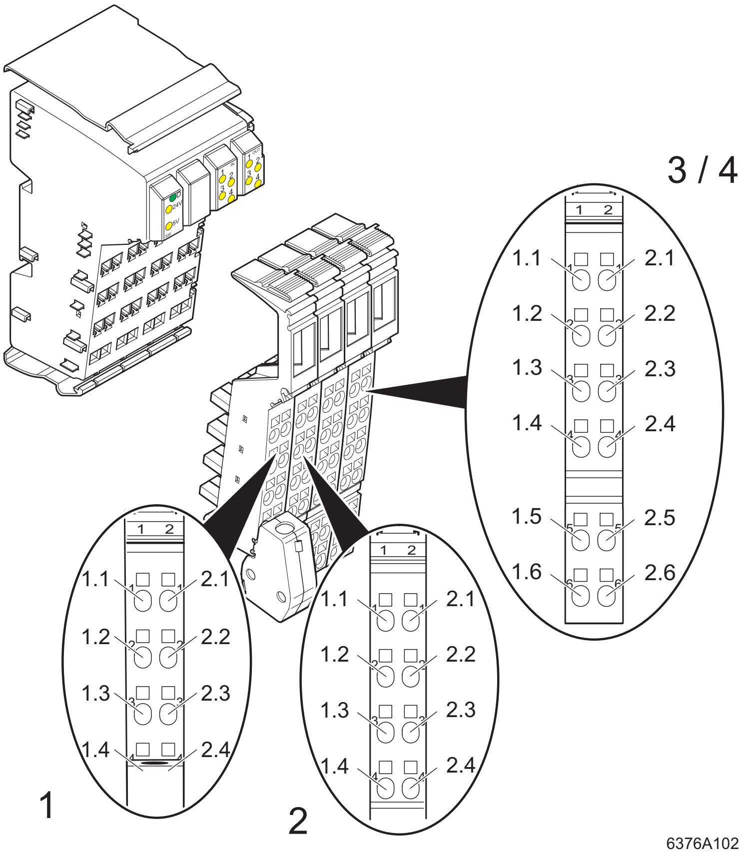


Figure 1-4 Terminal point assignment for the IB IL SSI ... terminal

Terminal point assignment

Connector	Terminal Point	Signal	Assignment
1 Shield connector	**1.1**	**24 V**	+24 V DC -encoder supply
	1.2	**GND**	Reference ground for the encoder supply
	1.3	**5 V**	+5 V DC -encoder supply
	1.4	**Shield**	Shield connection (high resistance and capacitance to FE)
	2.1, 2.2 2.3, 2.4	**–**	Not used
2 Standard connector	**1.1**	**T**	Clock
	2.1	$\overline{T}$	Clock inverted
	1.2	**D**	Data
	2.2	$\overline{D}$	Data inverted
	1.3, 2.3 1.4, 2.4	**–**	Not used
3 Extended double signal connector	**1.1**	**IN1**	Input 1
	2.1	**IN2**	Input 2
	1.2, 2.2	**24 V**	Supply voltage +24 V DC (U_M)
	1.3, 2.3	**GND**	GND of the supply voltage
	1.4	**IN3**	Input 3
	2.4	**HHOP**	Enable hand-held operator panel mode (HHOP enable)
	1.5, 2.5	**24 V**	Supply voltage +24 V DC (U_M)
	1.6, 2.6	**GND**	GND of the supply voltage
4 Extended double signal connector	**1.1**	**OUT1**	Output 1
	2.1	**OUT2**	Output 2
	1.2, 2.2	**GND**	GND of the supply voltage
	1.3, 2.3	**FE**	Functional earth ground
	1.4	**OUT3**	Output 3
	2.4	**OUT4**	Output 4
	1.5, 2.5	**GND**	GND of the supply voltage
	1.6, 2.6	**FE**	Functional earth ground

The encoder supply is generated from the main voltage U_M.

1.2.3 Diagnostic and Status Indicators

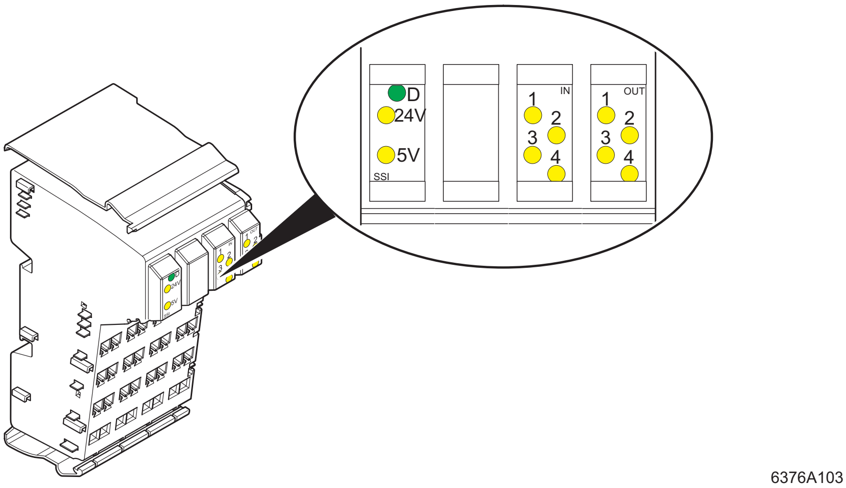


Figure 1-5 Diagnostic and status indicators

Diagnostics

The following states can be read from the positioning terminal:

LED	Color	Meaning	State	Description of the LED States
D	Green	**D**iagnostics	ON:	Bus active
			Flashing:	
			0.5 Hz: (slow)	Communications power present, bus not active
			2 Hz: (medium)	Communications power present, bus active, I/O error
			4 Hz: (fast)	Communications power present, bus connection for the flashing terminal has failed; terminals to the right of the flashing terminal are not part of the configuration frame
			OFF:	Communications power not present, bus not active
24V	Green	**24 V** encodersupply	ON:	24 V encoder supply present
			OFF:	24 V encoder supply not present
5V	Green	**5 V** encoder supply	ON:	5 V encoder supply present
			OFF:	5 V encoder supply not present

Status

The status of the inputs and outputs can be read from the relevant yellow LEDs:

LED	Color	Meaning	State	Description of the LED States
Slot 3 (IN)				
1, 2, 3	Yellow	Status of the corresponding input	ON:	Input set
			OFF:	Input not set
4	Yellow	HHOP status	ON:	Hand-held operator panel mode active
			OFF:	Hand-held operator panel mode not active
Slot 4 (OUT)				
1, 2, 3, 4	Yellow	Status of the corresponding output	ON:	Output set
			OFF:	Output not set

1.2.4 Circuit Diagram

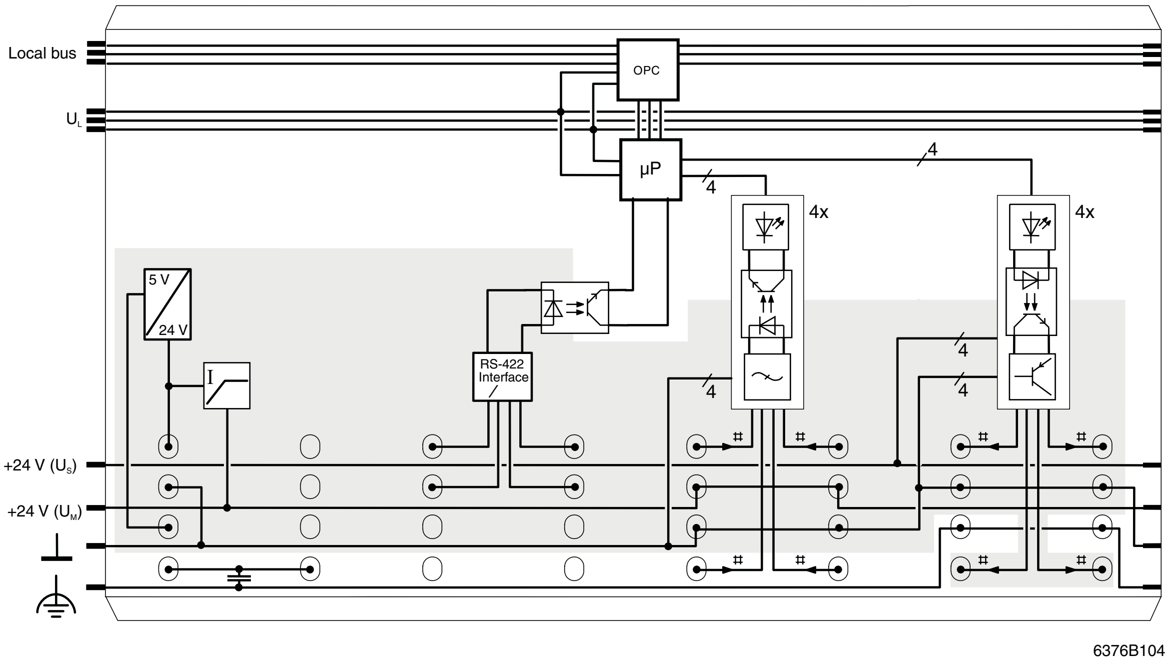
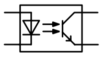


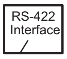
Figure 1-6 Circuit diagram of the IB IL SSI ...

Key:

Protocol chip (bus logic including voltage conditioning)

Microprocessor

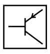

Optocoupler



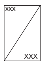
RS-422 interface


LED(s)


Filter


Transistor


Capacitor


Power supply unit without electrical isolation


Encoder supply U_G with short-circuit protection


Digital input


Digital output


Ground


Functional earth ground


Terminal point

Potential or data jumper with jumper contacts on the side

2 Mounting/Removing the Positioning Terminal and Connecting Cables

2.1 Installation Instructions

2.1.1 Unpacking a Terminal

ESD Regulations

> Persons who handle Inline terminals must protect them by observing ESD regulations before packing or unpacking the terminals, opening control boxes or control cabinets, and before touching the terminals.

Unpacking the Terminal

The terminal is supplied in an ESD box together with a package slip with installation instructions. Please read the complete package slip carefully before unpacking the terminal.

> Only qualified persons should pack, unpack, mount, and remove a terminal. ESD regulations must be observed at all times.

2.1.2 Replacing Terminals

> **Do not replace terminals while the power is connected.**
>
> Before working on a terminal, removing a terminal from the station or inserting a terminal in the station, disconnect the power to the entire station. Make sure the entire station is completely reassembled before switching the power back on.

2.2 Mounting and Removing the Terminal

The positioning terminal is designed for use within an Inline station.

An Inline station is set up by mounting the individual components side by side. No tools are required. Mounting the components side by side automatically creates potential and bus signal connections between the individual station components.

All Inline terminals are mounted on 35 mm standard DIN rails. The terminals are mounted perpendicular to the DIN rails. This ensures that they can be easily mounted and removed even when space is at a premium.

After a station has been set up, individual terminals can be exchanged by pulling them out or plugging them in. Tools are not required.

Setting up an Inline station and the terminal mounting and removal procedure are described in the "Configuring and Installing the INTERBUS Inline Product Range" user manual IB IL SYS PRO UM E and in the system manual for the bus system you are using.

In addition, the sequence of the terminals is specified in "Sequence of the Inline terminals" on page 6-1.

2.3 Power Supply

The terminal is supplied with power via the potential jumpers. No additional power connections are needed.

2.4 Connecting Encoders, Sensors and Actuators

2.4.1 Connection Notes and Shielding

Encoders should always be connected using **shielded** cables. Unshielded cables may lead to erroneous results in environments subject to heavy noise. On the terminal side, there is a capacitive connection between the shield and the functional earth ground (FE) via the shield connector. On the encoder side, the shield must be connected with the grounded encoder housing.

The encoders, sensors and actuators are connected to the terminals using connectors. The IB IL SSI/INC-PLSET connector set is designed for the IB IL SSI ... terminal. It contains a shield connector, a standard connector and two extended double signal connectors.

Connect the encoder using the shield connector, and all other cables using connectors without shield connection.

Connect unshielded cables as described in "Connecting Cables to the Connector Without Shield Connection" on page 2-6.

Connect shielded cables as described in "Connecting Shielded Cables Using the Shield Connector" on page 2-8.

2.4.2 Connection Methods for Sensors and Actuators

Sensors

The sensors can be connected using the following methods:

– 2-wire (signal and 24 V)
– 3-wire (signal, 24 V and GND)

Actuators

The actuators can be connected using the following methods:

– 2-wire (signal and GND)
– 3-wire (signal, GND and FE)

Sensor Connection

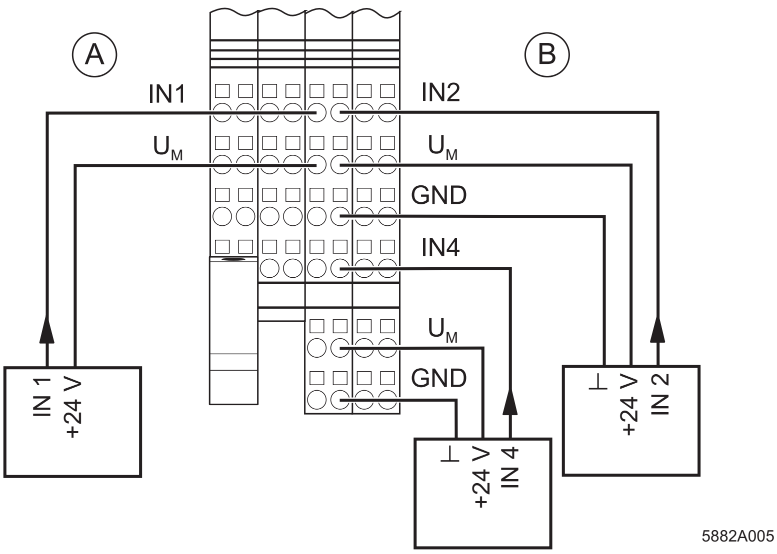


Figure 2-1 Sensor connection

2-wire technology

The left side (A) shows the connection of a 2-wire sensor. The sensor signal is routed to terminal point IN1. The sensor power supply comes from voltage U_M.

3-wire technology

The right side (B) shows the connection of two 3-wire sensors. The sensor signals are routed to terminal points IN2 and IN4. The sensor power supply comes via terminal points U_M and GND.

Actuator Connection

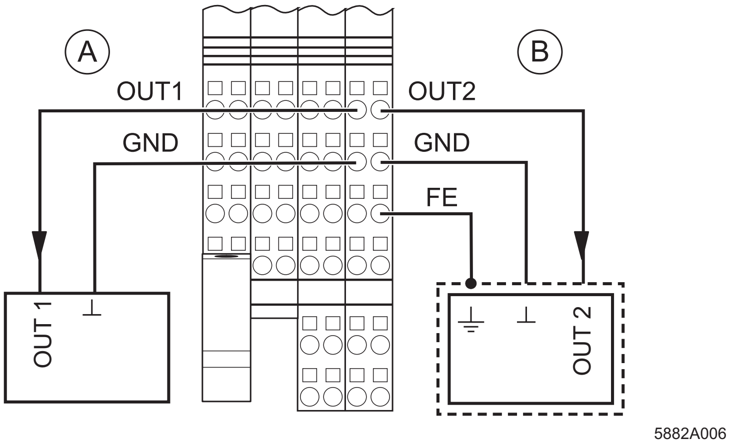


Figure 2-2 Actuator connection

2-wire technology

The left side (A) shows the connection of a 2-wire actuator. The actuator power is supplied via output OUT1. The load is switched directly via the output.

3-wire technology

The right side (B) shows the connection of a shielded actuator. The actuator power is supplied via output OUT2. The load is switched directly via the output.

The 500 mA maximum current carrying capacity for each output must not be exceeded.

2.5 Connecting Cables

Both shielded and unshielded cables are used with the IB IL SSI ... terminal.

A shielded cable is used to connect the encoder. The shield is connected via the shield connector, and the encoder via connectors 1 and 2 according to "Terminal Point Assignment" on page 1-6.

Sensors and actuators are connected via extended double signal connectors. If shielded actuators are used, the shield is connected via the FE connection.

The I/O cables are connected using the spring-cage connection method. This method supports the connection of cables with a conductor cross section of 0.2 mm^2 to 1.5 mm^2 (AWG 24 - 16).

2.5.1 Connecting Cables to the Connector Without Shield Connection

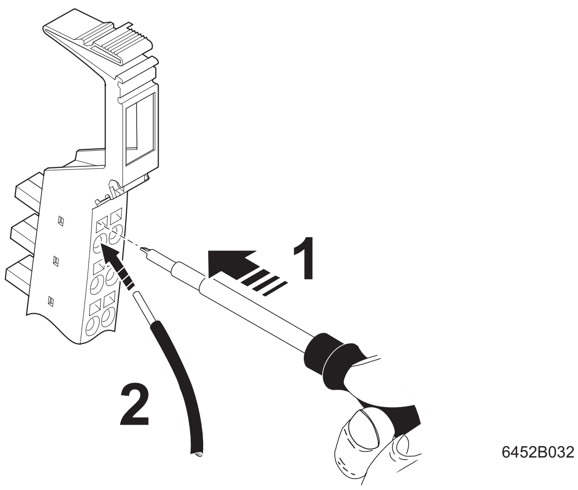


Figure 2-3 Connecting unshielded cables

Wire the connectors as required for your application.

> The connector pin assignment is specified in "Terminal Point Assignment" on page 1-6.

When wiring, proceed as follows:

- Strip 8 mm off the cable.

> Inline wiring is normally done without ferrules. However, it is possible to use ferrules. If using ferrules, make sure they are properly crimped.

- Push a screwdriver into the slot above the appropriate terminal point (1 in Figure 2-3), so that you can plug the wire into the spring opening.
  Phoenix Contact recommends using a SZF 1 - 0,6X3,5 screwdriver
  (Order No. 1204517; see Phoenix Contact "CLIPLINE" Catalog).
- Insert the wire (2 in Figure 2-3). Remove the screwdriver from the opening. This clamps the wire.

After installation, the wires and the terminal points should be labeled (see the "Configuring and Installing the INTERBUS-Inline Product Family" IB IL SYS PRO UM E user manual or the system manual for the bus system you are using.

2.5.2 Connecting Shielded Cables Using the Shield Connector

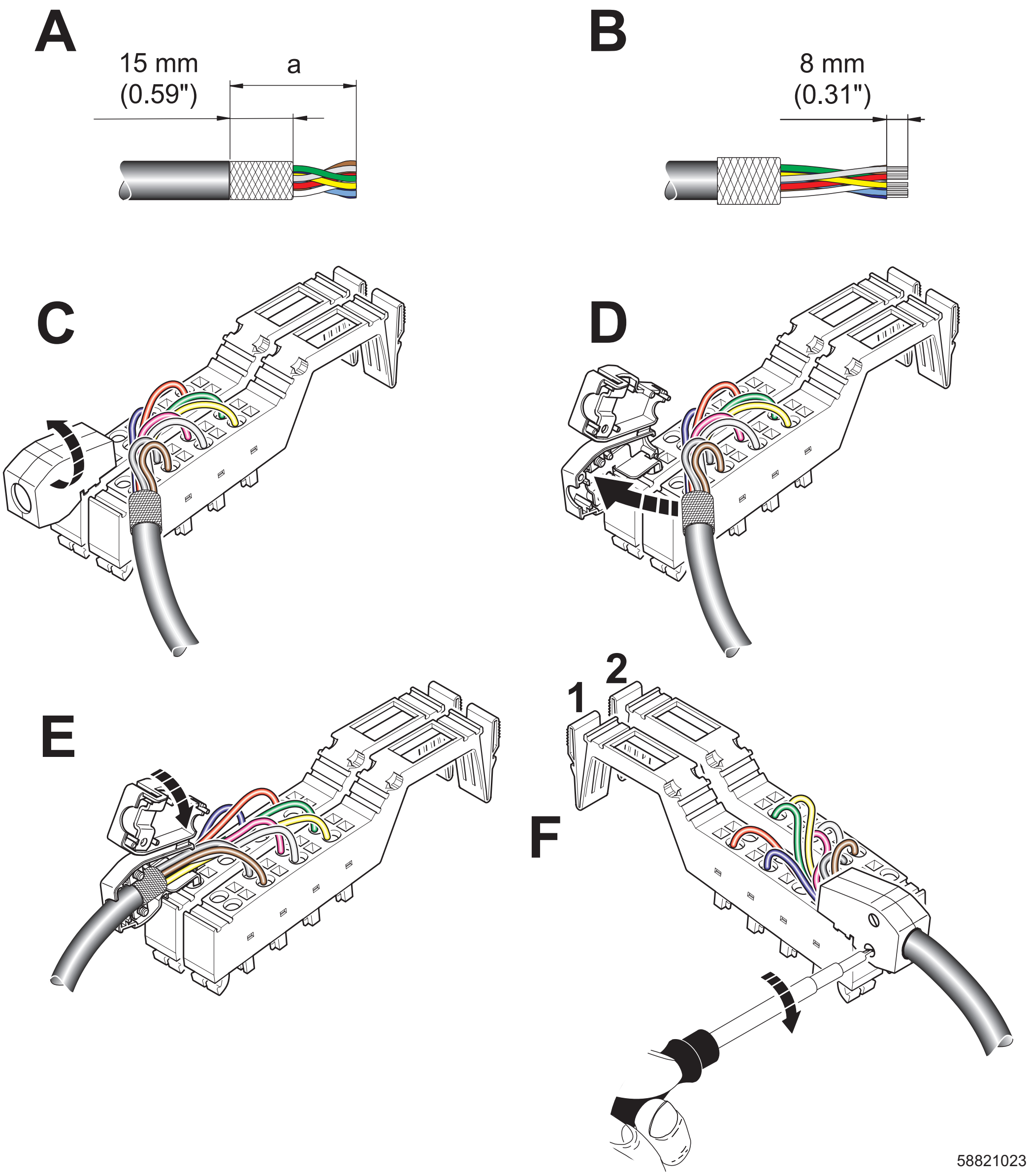


Figure 2-4 Connecting the shield to the shield connector

The encoder is connected using shielded cables. The basic procedure for connecting shielded cables is described here. The connection method differs for incremental encoders. The specific connector pin assignment is given in "Terminal Point Assignment" on page 1-6.

When connecting the shielded cable, proceed as follows:

Stripping cables

- Strip the outer cable sheath to the desired length (a) (A).

  Choose a length (a) that also enables proper connection of the cable to connector 2. The required length (a) also depends on whether the wires are to be generous or tight between the connection points and the shield connection.
- Shorten the braided shield to 15 mm (A).
- Fold the braided shield back over the outer sheath (B).
- Remove the protective foil.
- Strip 8 mm off the wires (B).

Inline wiring is normally done without ferrules. However, it is possible to use ferrules. If using ferrules, make sure they are properly crimped.

Wiring the connectors

Wire the connectors according to "Connecting Cables to the Connector Without Shield Connection" on page 2-6.

For the connector assignments, please refer to "Terminal Point Assignment" on page 1-6.

Connecting the shielding

- Open the shield connector (C).
- Place the shield clamp in the shield connector corresponding to the cable cross section (see Figure 2-5 or Figure 2-6).
- Place the cable with the folded braided shield in the shield connector (D).
- Close the shield connector (E).
- Use a screwdriver to tighten the screws for the shield connector (F).

Shield connection clamp

The shield connection clamp (a in Figure 2-5, B and Figure 2-6, B) in the shield connector can be used in various ways depending on the cross section of the cable. For thinner cables, the dip in the clamp must be turned towards the cable (Figure 2-5, C). For thicker cables, the dip in the clamp must be turned away from the cable (Figure 2-6, C).

Connecting a shield connection clamp for thin cables

To position the shield connection clamp for thin cables, proceed as shown in **Figure 2-5**:

- Open the shield connector housing (A).
- Connect the shield connection clamp as shown in Figure 2-5, B.
- Figure C shows the position of the shield connection clamp for thin cables.

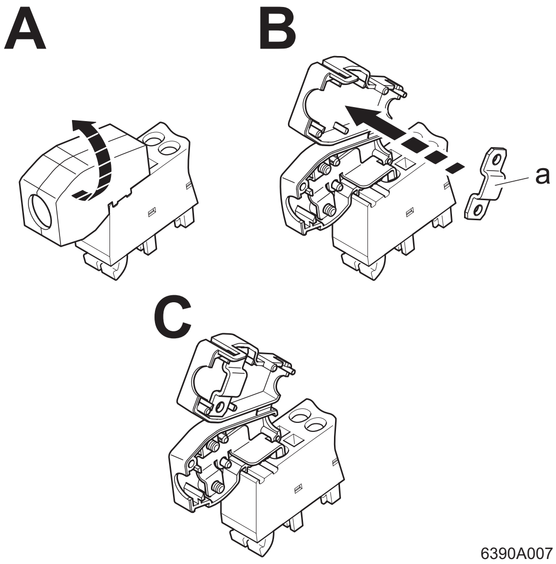


Figure 2-5 Position of the shield connection clamp for thin cables

Connecting a shield connection clamp for thick cables

To position the shield connection clamp for thick cables, proceed as shown in **Figure 2-6**:

- Open the shield connector housing (A).
- Connect the shield connection clamp as shown in Figure 2-6, B.
- Figure C shows the position of the shield connection clamp for thick cables.

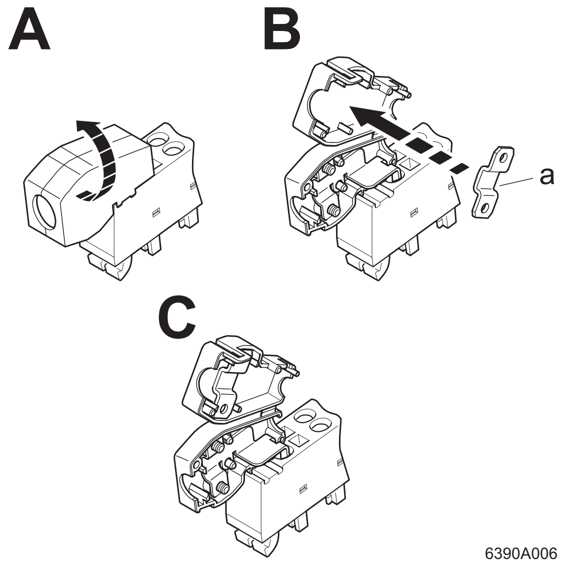


Figure 2-6   Position of the shield connection clamp for thick cables

2.6 Connection Examples

2.6.1 Example for Wiring Inputs and Outputs

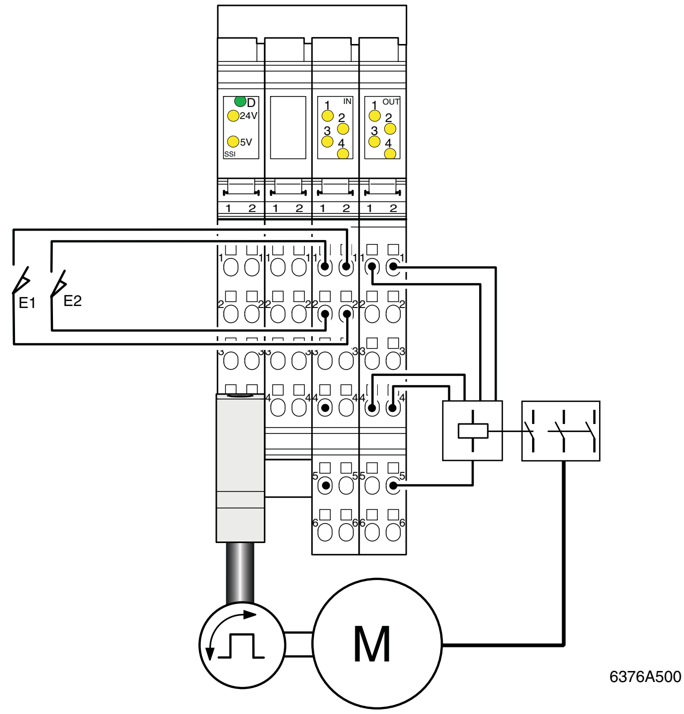


Figure 2-7 Example for wiring inputs and outputs

I1	Limit switch 1 (minimum limit switch)
I2	Limit switch 2 (maximum limit switch)

Connect the four digital outputs to the power contactors according to their output version (see "Function of the Switching Outputs (OutputFunction)" on page 5-14).

Configure the inputs and outputs according to ""Configuration of Axis Types, Behavior in the Event of a Bus Failure, Initiators and Switching Outputs" (DefineInOut) Command" on page 5-11.

2.6.2 Wiring the Positioning Terminal When Using Hand-Held Operator Panel Mode

Detailed information about hand-held operator panel mode can be found in "Hand-Held Operator Panel Mode" on page 3-27.

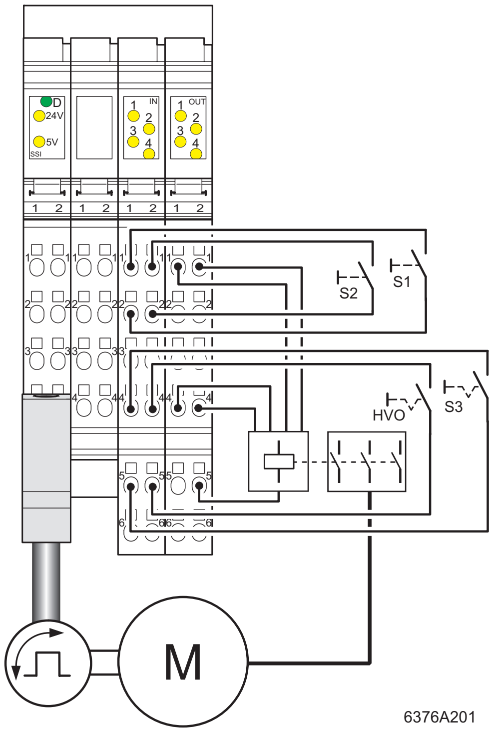


Figure 2-8 Wiring an IB IL SSI ... when using hand-held operator panel mode

> Connect the four digital outputs to the power contactors according to their output version (see "Function of the Switching Outputs (OutputFunction)" on page 5-14).

The switches and buttons have the following meaning:

HHOP	Hand-held operator panel
S1	Slow negative run
S2	Slow positive run
S3	0 - Output version 1 or 2 1 - Output version 3

> **Check that your configuration is correct!**
>
> Avoid an incorrect configuration of the output versions, as this may lead to errors or damage the connected components.

Connector for HHOP mode

In hand-held operator panel mode, the extended double signal connector, which is connected with the limit switches, is removed and replaced by a double signal connector, which is wired for hand-held operator panel mode. The connector for hand-held operator panel mode is connected with two manual switches and two manual buttons. Activating the HHOP switch enables hand-held operator panel mode. Switch S3 is used to select the desired output version. Buttons S1 and S2 can be used to drive the motor slowly in a positive or negative direction.

When terminating hand-held operator panel mode, deactivate hand-held operator panel mode and re-insert the connector that was originally connected to the limit switches.

> **Avoid malfunctions!**
>
> Terminal point 2.4 of connector 3 (HHOP enable) must only be used for hand-held operator panel mode. If this terminal point is connected inadvertently, the positioning terminal interprets this as hand-held operator panel mode, leading to errors.

2.7 Startup

After installation the terminal is in the initial state. In order to work with the terminal, it must be parameterized.

The commands to parameterize the terminal can be found in Section 5, "Commands for Working With the Positioning Terminal".

A process plan for parameterizing a position can be found on Figure 5-9 on page 5-51. To parameterize your terminal and to read or approach a position, carry out the steps in the specified order.

Please note that in the event of a communications power failure on the terminal, the parameterization is **not** saved. In this case, the terminal must be parameterized again once the communications power has been restored.

Some parameterization examples can be found in Section 6.2, "Examples".

3 Positioning

The higher-level control system assigns a target position to the IB IL SSI ... positioning control system. Positioning can be started by the higher-level control system or via an input. The positioning control system then carries out positioning automatically and confirms the result to the higher-level control system.

The positioning control system uses the rapid motion/creeping motion principle for positioning. The drives travel at two fixed speeds (first rapid motion, then creeping motion) until they reach the stop point. Then the drive is switched off. This means that positioning is achieved by means of rapid motion/creeping motion, whereby the positioning control system activates and switches over the output signals for drive activation following the start of positioning should the axis overshoot the predefined position thresholds. The signals are first switched from rapid motion to creeping motion and then from creeping motion to stop. Once the target position has been reached, the drive is stopped (i.e., it does not supply torque in the target position).

Due to its fast-response logic, the positioning terminal can achieve positioning accuracy rates of micrometers using this simple method.

The terminal controls a positioning process by comparing the current position value with the specified target position and the parameterized window limits. The direction of travel is calculated on the basis of the type of axis and the result of the comparison and the corresponding output bits are activated to control the drive. This means that four switching outputs control the speed and direction of travel of the drive.

The positioning terminal differentiates between linear axes and rotary axes in order to ascertain the path to be taken to the target. For linear axes, the target is approached in a positive or negative direction, depending on whether the difference between the target position and actual position is positive or negative. In the case of rotary axes, either the shorter path to the target is selected or the target continues to be approached in one direction (if the other direction of travel has been disabled).

The current position value can be read in every bus cycle.

In addition to the input for the encoder signals, the terminal also has three digital 24 V inputs for limit switches.

2- or 3-wire initiators can be connected to these inputs.

The drive stops when the limit switch is activated.

To ensure the correct operation of the terminal, start by defining the encoder, the initiators and the switching outputs. Then parameterize all windows, which belong to a position, and the target position. By default, all windows are set to 0 and must be adapted if necessary.

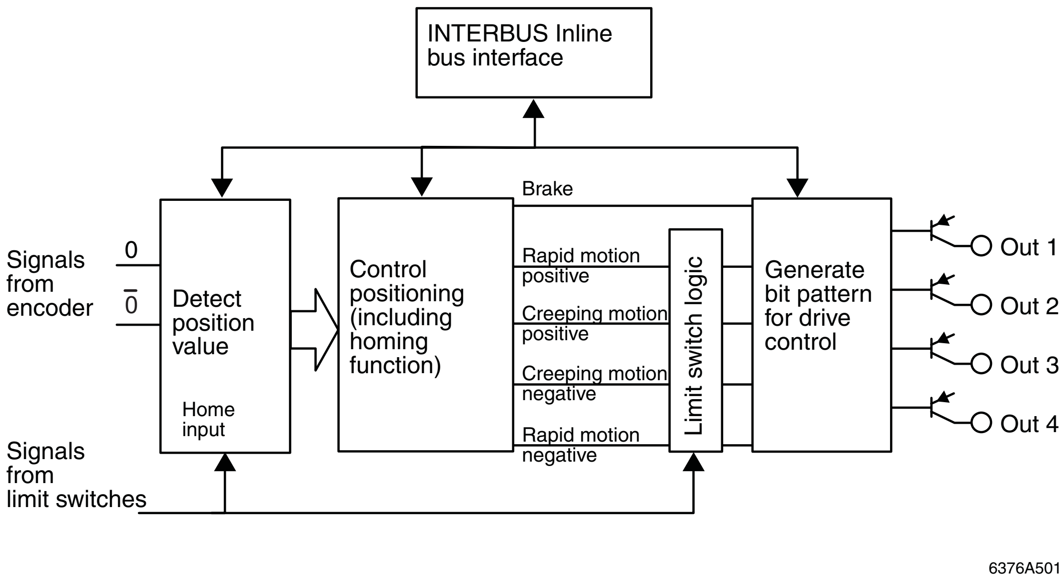


Figure 3-1 Function block of the IB IL SSI ... terminal

3.1 Positioning and Structure of a Position

When approaching the specified target position, the drive is controlled independently by the terminal according to the terminal parameterization.

The value for the target position and the parameters defining the switching positions are written to the terminal. During positioning, the terminal recognizes the individual states (e.g., negative creeping motion) and controls the drive accordingly. The output bits are set accordingly for the various drives (pole-changing motors, Dahlander circuit, etc.) (the settings are not the same for each drive).

The bit combination the terminal works with is defined via the "Function of the switching outputs" parameter. There are five versions, which are listed in "Function of the Switching Outputs (OutputFunction)" on page 5-14. Depending on the parameterization of the output behavior, the position will be approached at two or three speeds (for the version for controlling variable speed drives, e.g., frequency inverters).

For positioning with two speeds, creeping motion and rapid motion are used. For positioning with three speeds, fast motion is available in addition to creeping motion and rapid motion.

Once the positioning process is complete, i.e., once the drive has stopped, the terminal checks whether the required position has been reached. To do this, it checks whether the drive is in stop by waiting until the latter is moving more slowing than defined in the time interval and path parameters under "Drive stop".

It then ascertains whether the axis position is inside the target window and sends positive confirmation of the positioning process. If the axis is not inside the target window, either a negative confirmation will be sent or positioning is repeated until such confirmation can be sent (target repeating counter REPEATING COUNT 1 or REPEATING COUNT 2).

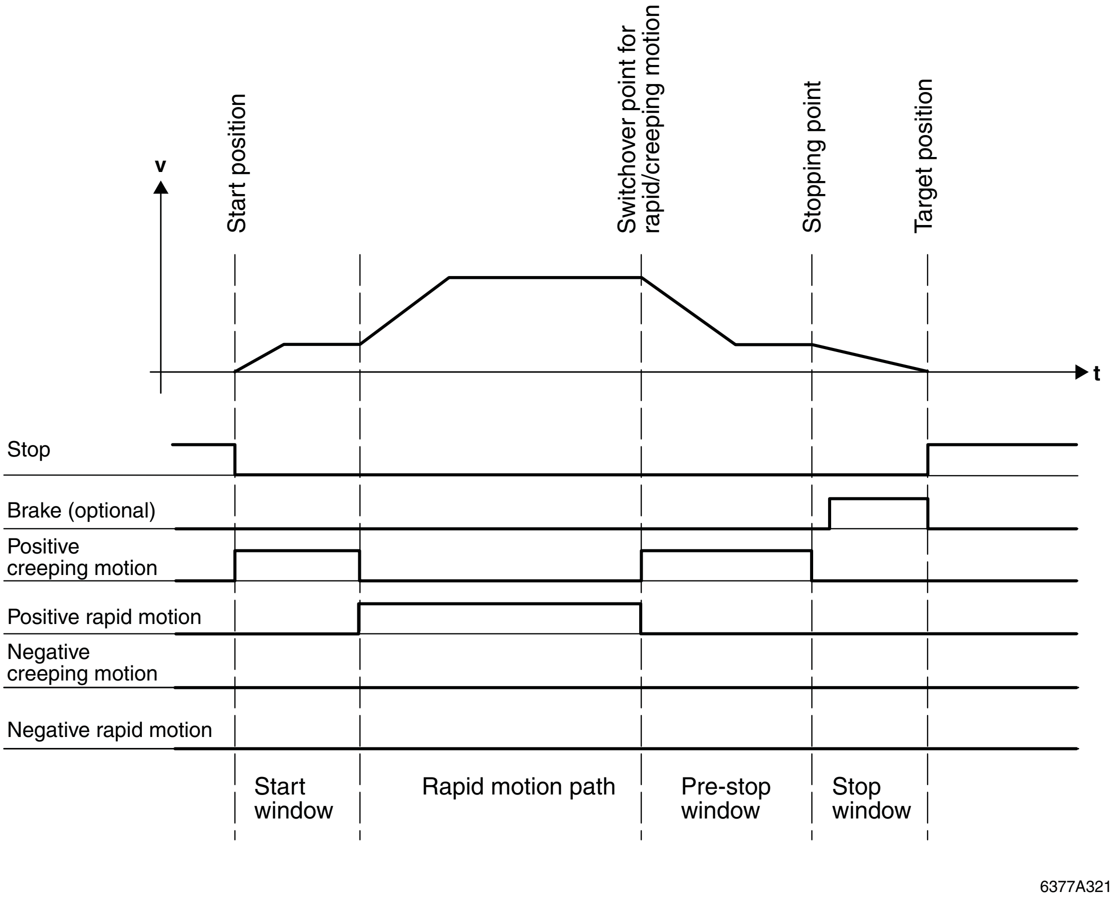


Figure 3-2 States for positioning in a positive direction

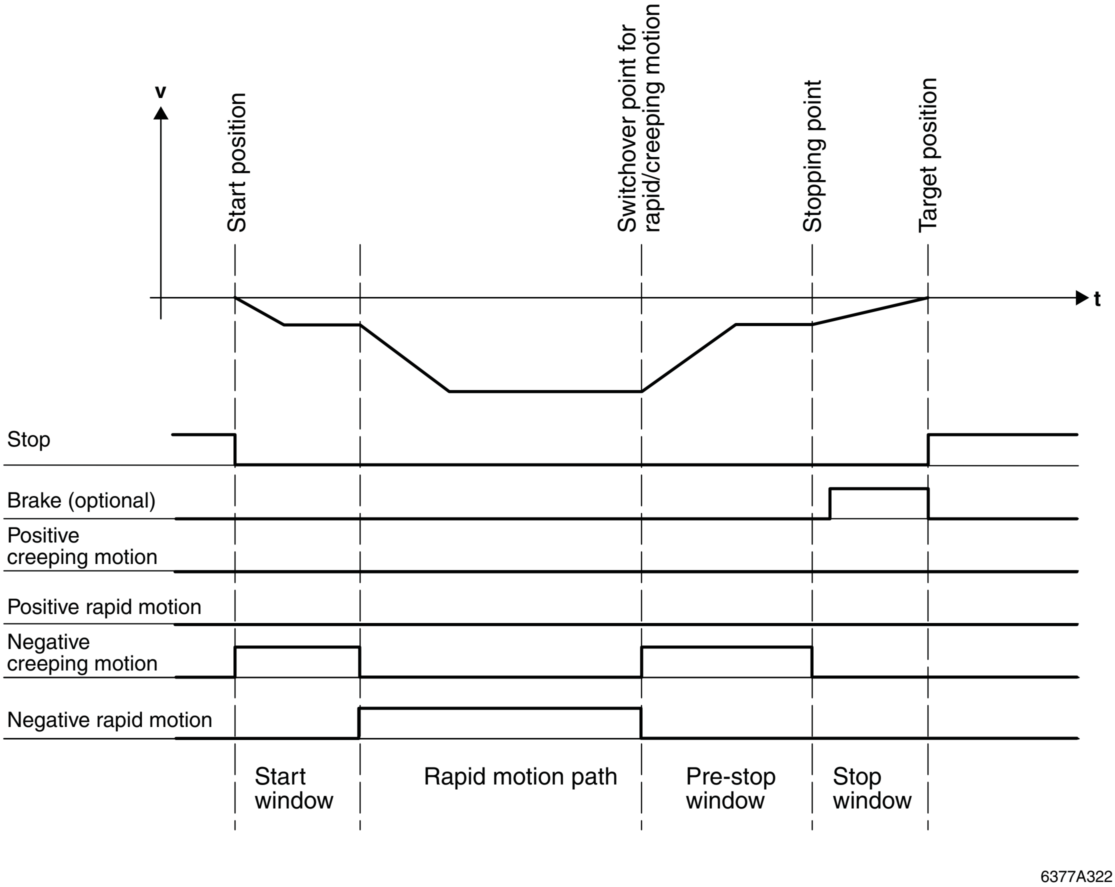


Figure 3-3 States for positioning in a negative direction

Structure of a positioning data record

The parameter record for a position comprises the following values:

- TargetPosition
- TargetWindow
- StopWindow
- Pre-StopWindow
- StartWindow
- LubFricCompValue

Plus, if version 4 has been selected under "Function of switching outputs":

- RapidStopWindow
- RapidStartWindow

Each of these parameters is defined independently of the others using the corresponding commands (see "Commands for Parameterizing the Path" on page 5-24). Examples of the arrangement of the individual switching points and the stop point during positioning are provided in Figure 3-5 and Figure 3-4.

On the positioning terminal, two parameter records can be defined for target positions.

Positioning phases

The phases (states) of a positioning process with **two** speeds (function of switching outputs, versions 1, 2 and 3) are illustrated in Figure 3-4:

- Start the drive and run at creeping speed.
- Switch the drive to rapid motion (on leaving the start window).
- Switch the drive to creeping speed (pre-stop window reached).
- Stop the drive: Roll to a stop in the target window (stop window reached).

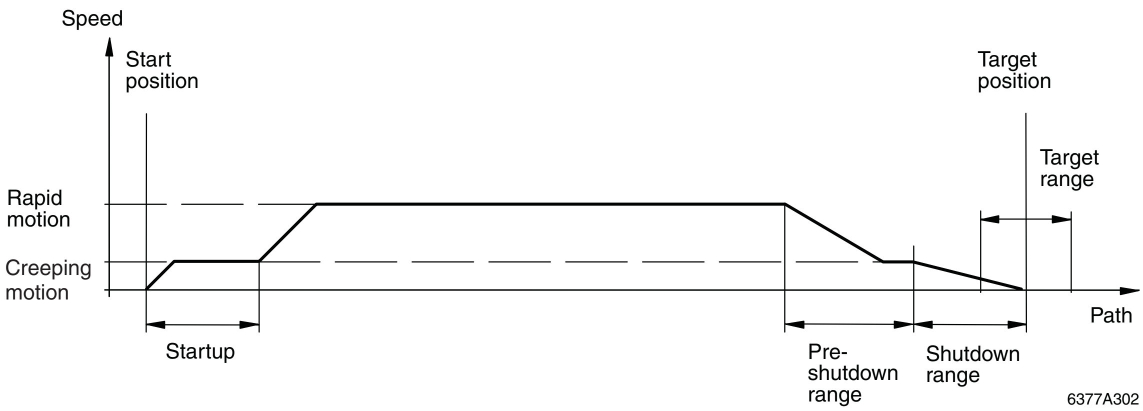


Figure 3-4 Curve for positioning with two speeds

The phases (states) of a positioning process with **three** speeds (switching output behavior, version 4, see page 5-15) are illustrated in Figure 3-5:

- Start the drive and run in rapid motion.
- Switch the drive to fast motion (on leaving the rapid start window).
- Switch the drive to rapid motion (rapid stop window reached).
- Switch the drive to creeping speed (pre-stop window reached).
- Stop the drive: Roll to a stop in the target window (stop window reached).

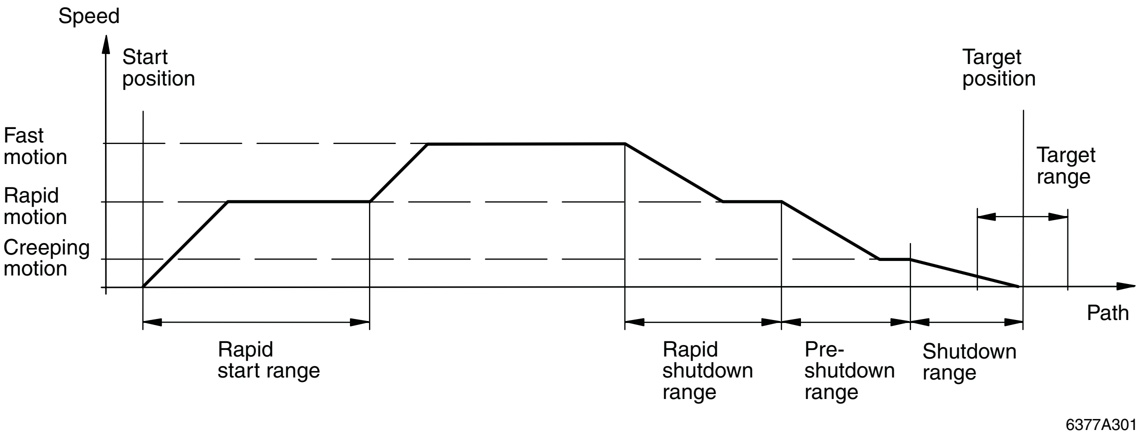


Figure 3-5 Curve for positioning with three speeds

In order to be support the maximum possible number of applications, the terminal is very flexible in terms of adapting position detection to the requirements of the mechanics (gear ratio).

In addition to its positioning function, the terminal also features a number of monitoring functions, which promote error-free operation of the automated axes. First, the function of the position encoder is monitored continuously. Second, the error-free operation of the drive is checked (direction of rotation and motion).

> Changing a saved position once a positioning process is underway has no effect on that process. The new values only take effect when the next positioning process starts.

The target position and the remaining parameters can be specified using control commands in the output words (see "Control Commands" on page 5-42).

Configuration

To ensure the correct operation of the terminal, first define the encoder, initiators and switching outputs.

Finally, you need to parameterize the positioning data record.

By default, all areas are set to 0. You only need to parameterize the values you need for your application.

Please also note that the firmware cycle time is 500 µs. This means that the positioning error caused by the positioning terminal corresponds to the distance traveled at creeping speed in 500 µs. The actual position is read in with the same time offset.

Example

At a creeping speed of 1 cm/s, the distance will be:

$$v = \frac{s}{t}$$

$$s = v \times t$$

$$s = 1\ cm/s \times 500\ \mu s = 5\ \mu m$$

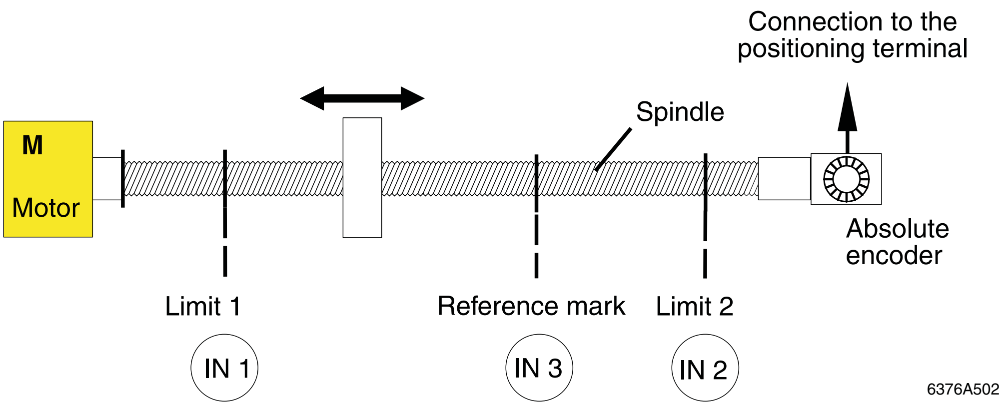


Figure 3-6   Application example for the inputs for linear axis positioning

3.2 Drive Stop

The module uses the speed threshold set by the drive stop for two functions:

1 Monitoring the drive during positioning
During positioning, the module checks whether the drive is **running correctly**, i.e., whether it is running in the right direction faster than when the drive is in stop.

2 Detection that the drive has actually come to a stop after stopping (if the target window has been reached)
During positioning, once the drive has stopped, the system waits until the terminal has detected the stop.

To define the drive stop, a path and time interval must be specified. If a path less than that specified is covered in the defined time interval, the drive detects this as a drive stop.

The parameters for detecting a drive stop can be specified using the *Define drive stop* command (see ""Drive Stop" (DefineDrvStop) Command" on page 5-16).

This path/time combination is the maximum speed, which the terminal still interprets as "drive stopped". If, when the time has expired, the terminal detects that the count value of the increment has not increased (forward running) or reduced (backward running) by at least the predefined value, it is presumed that the drive has stopped.

The module uses the speed threshold set by the drive stop for two functions:

Monitoring the drive during positioning

During positioning, the module checks whether the drive is **running correctly**, i.e., whether it is running in the right direction faster than when the drive is in stop. The module checks whether, within the set time interval, the positioning counter has increased (forward running) or reduced (backward running) by the value that was specified for the increment, since it was last checked. If the desired path was not covered within the time, the module switches to the error status "Drive stop detected" (error message 14_{dec}). The drive is switched off in this error status.

> Set the parameters according to your application. Please observe the following points to prevent the error message being triggered unnecessarily:

- If the motor is to start up against a large load or against a brake, the "Delay time for detection of direction and stop" parameter must be defined in addition to the "Drive stop" parameter. This means that the drive can start moving within this time without the error message "Drive stop detected" being generated.
  Drive stop monitoring only starts once the delay time for detection of direction and stop has elapsed (see ""Delay Time for Detection of Direction and Stop and Output Delay Time" (DefineDsdOd) Command" on page 5-22).
- If vibrations occur in the drive train, the time for defining the drive stop must be increased so that it is greater than the vibrations.

Once the drive has stopped, the module only detects this stop once the set time has expired (the worst-case scenario is the expiry of a period of time that is twice as long as the set time). Therefore, do not select a time any longer than necessary.

Detection that the drive has actually come to a stop after stopping (if the target window has been reached)

During positioning, after drive shutdown the system waits until the terminal has detected the stop. The terminal checks whether the drive is moving slower than defined in "drive stop", i.e., has traveled less than the set increments in the predefined time. Once the stop has been detected, the stop bit is set to "1" (see ""Control Position and Read Status" (ControlPosition, ReadStatus) Command" on page 5-47).

Only after this stop has been detected is a check made to determine whether the drive is in the target window. Next, the positioning process is completed by entering the result in the status word.

If you read the status after positioning, the status words indicate the result of the positioning process (see ""Control Position and Read Status" (ControlPosition, ReadStatus) Command" on page 5-47).

If the target window has been reached, IN[1] bit 2 (positioning process with parameter record 1 completed successfully) or IN[1] bit 4 (positioning process with parameter record 2 completed successfully) is set in the status word.

If the target window has not been reached, IN[0] bit 15 (error) is set in the status word, and an error code (17_{dec}: "Target window could not be reached") is generated.

If the parameters for monitoring the drive stop are set too low, effects such as vibrations on the axis may mean that no stop is detected. In this case, the positioning process is not completed, and can only be aborted by the user sending the stop command.

Please note that the selected time interval is added to the duration of the positioning cycle. This means that once the drive stops, the time interval selected will continue to run until the terminal detects the drive stop and interprets the positioning process as being at an end.

If drive stop monitoring is not active, the stop bit is constantly set. As positioning without a drive stop makes no sense, it is not possible to enter a value of zero for the drive stop.

3.3 Looping

If the difference between the start and target positions is less than the sum of the start window and stop window, it will not be possible to approach this target position directly. Looping can be used to exit the range automatically and approach the position again.

Looping must be enabled for this. It is enabled using output word OUT[1] bit 12 for the *Control positioning* command (see "Control Commands" on page 5-42).

The direction of travel on exiting the target window is the opposite of the approach direction specified in OUT[1] bit 11 (Approach direction of the target position for active backlash compensation or looping).

The direction of the loop is determined by OUT[1] bit 11 (Approach direction of the target position for active backlash compensation or looping).

A new positioning process is carried out according to the parameterization.

If looping is required to reach the target position, the terminal executes the process automatically, provided that looping is enabled. The current target position is only approached again once the drive has stopped.

Example

In Figure 3-7, A, the end position of the last positioning process is the start position of the next positioning process. This start position is within the sum of the start and stop windows and, therefore, cannot be approached directly. The drive must be moved out of the start/stop windows using looping. The end point of this looping process is the start position for approaching the target position (Figure 3-7, B).

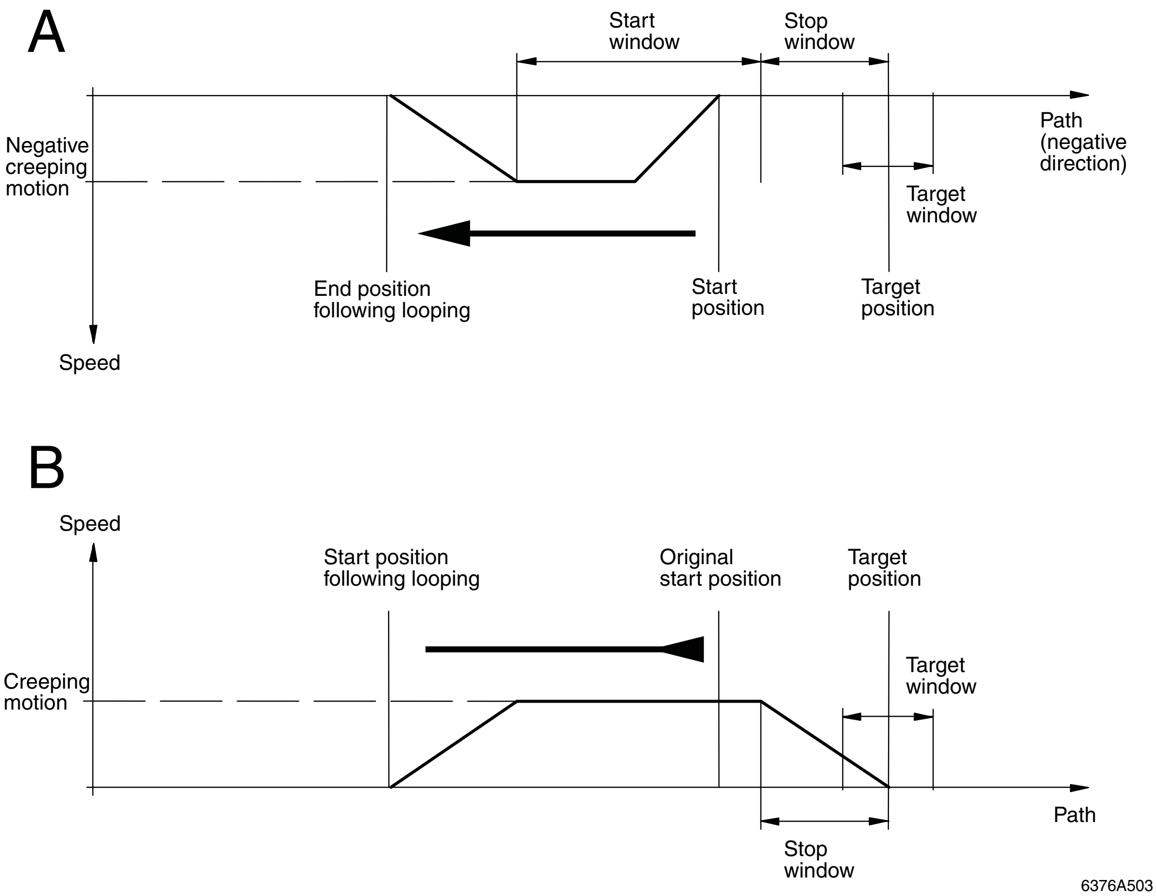


Figure 3-7 Example of a looping process

3.4 Backlash Compensation

Usually, drive systems have clearance, referred to as "backlash" in practice. Every time the direction changes, the backlash causes a motor rotation without changing the drive position. If the position encoder is linked to the motor axis, this leads to a reduction in positioning accuracy. The positioning terminal enables you to compensate the backlash by approaching all positions from one direction.

Backlash compensation can be activated and deactivated via OUT[1] bit 10 (Activate backlash compensation, ActBacklashComp) of the control word. The approach direction of the position can be specified using OUT[1] bit 11 of the control word (Approach direction of the target position for active backlash compensation, DefDrvDir).

Active backlash compensation monitors whether the software limit switches are overrun during the positioning process. If this happens, the position will not be approached. The terminal generates an error message (error code 10_{dec}: "Function cannot be executed, as software limit switches would be overrun", see: Table 5-10 on page 5-49).

Example

Figure 3-8, A:
If the specified approach direction is positive (OUT[1] bit 11 = 0), and the position is approached in a negative direction, then with backlash compensation activated, the target position will first be overrun. Once the rapid stop window is reached, the drive is stopped and comes to a standstill outside the range calculated by adding together the stop window and the start window.

Figure 3-8, B:
Since the target position was overrun, the drive now changes direction, and approaches the target position again in a positive direction.

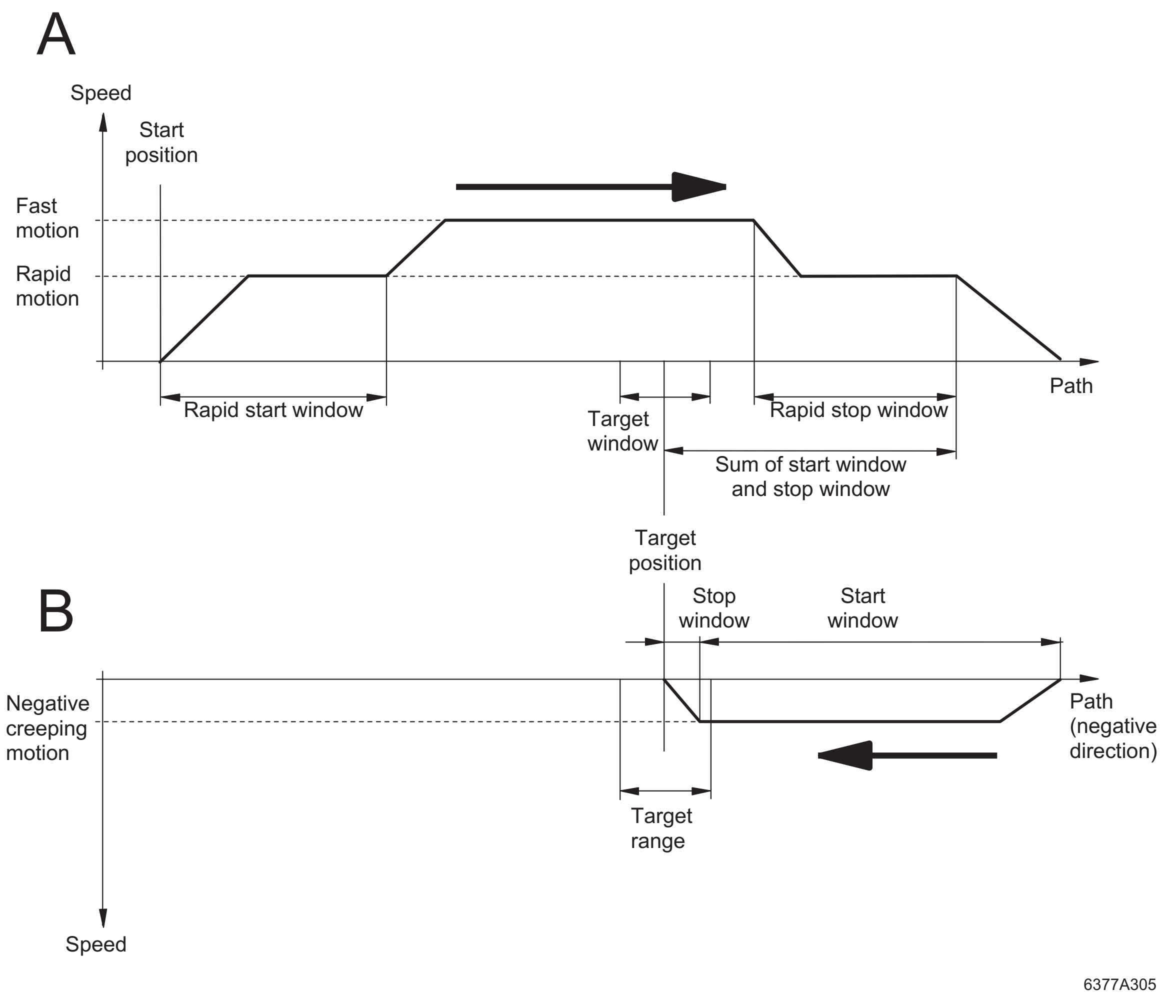


Figure 3-8 Example of approaching a position with backlash compensation

3.5 Lubrication and Friction Compensation

The disadvantage of positioning processes with switched drives (rapid motion/creeping motion principle) is that positioning accuracy is very much dependent upon the stability and reproducibility of the shutdown times of the power switch and the braking response of the mechanics.

If, over time, these parameters change continuously and slowly, the resulting positioning error can be corrected by moving the stop point (stop window) back. To do this, after every positioning process, the positioning terminal ascertains the positioning error and adds this to the friction compensation value. When the next positioning process gets underway, the drive is stopped at the position calculated from the difference between the stop point and the friction compensation value. It is for this reason that the positioning terminal supports lubrication and friction compensation.

Lubrication and friction compensation will **only** work if the shutdown time of the power switch or the braking response of the mechanics change **continuously**.

If lubrication and friction compensation has been activated by setting the bit of the same name in the control word (OUT[1] bit 14 = 1), the target position is corrected by this value.

Once the positioning process has been completed, the friction compensation value is recalculated. The compensation values from previous positioning processes are not taken into account.

The friction compensation value can be set and requested at any time by a higher-level control system.

3.6 Software Limit Switches

Software limit switches are used to protect your system. These software limit switches prevent the drive traveling to a position beyond the software limit switches in normal mode, and thus prevent it driving right up to the limits (hardware limit switches). Hardware limit switches can also be used (these will continue to function even if the position encoder fails).

Software limit switches can be defined using the *Define minimum software limit switch* (see page 5-20) and *Define maximum software limit switch* (see page 5-21) commands.

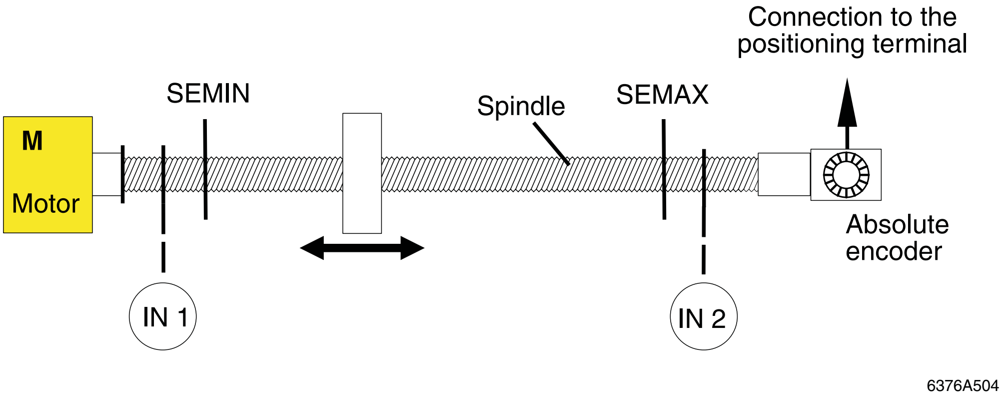


Figure 3-9 Example software limit switch

IN 1	Limit position 1
IN 2	Limit position 2
SEMIN	Minimum software limit switch
SEMAX	Maximum software limit switch

In rotary axis mode and for homing, the software limit switches are **deactivated**.

In jog mode, both the software and hardware limit switches are **deactivated** for terminals with **firmware version 1.1** or earlier.

For terminals with **firmware version 1.1 or later** both the software and hardware limit switches are **activated** in jog mode, although they can be deactivated. The hardware limit switches are deactivated using the *Define initiators and switching outputs* command (see page 5-11). The software limit switches are deactivated using the *Define software limit switches* command (see page 5-20).

The software limit switches are effective immediately.

3.7 Axis Types

The positioning terminal can be used to execute positioning with both linear and rotary axes. The desired axis type must be defined during parameterization. Depending on the axis type, there are differences during parameterization and operation.

Linear axis

A linear axis is an axis with a limited path. The path is limited by parameterized software limit switches (SEMIN, SEMAX) and/or hardware limit switches (IN1, IN2).

The maximum path is the permitted position range of -2^{25} to 2^{25}-1 increments.

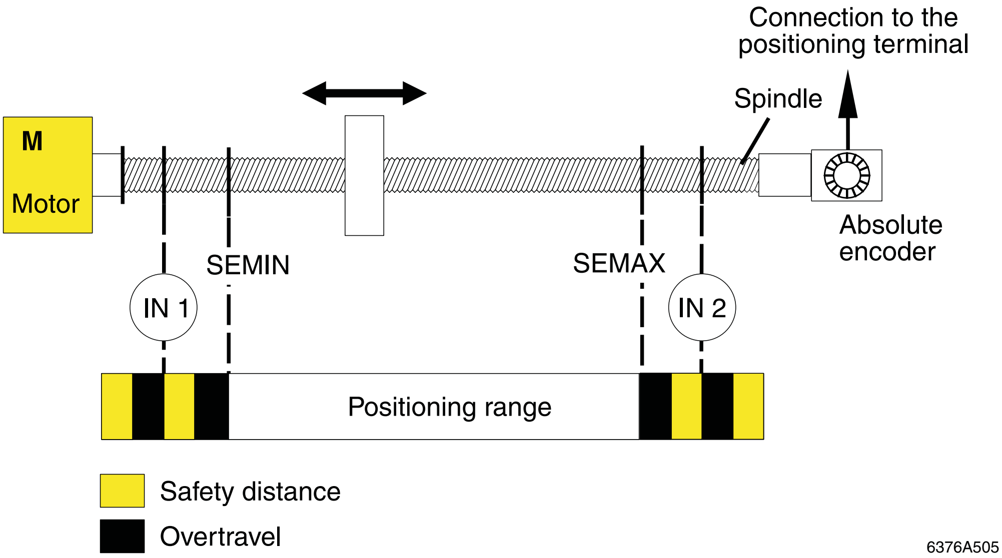


Figure 3-10 Example of a linear axis

IN 1	Limit position 1
IN 2	Limit position 2
SEMIN	Minimum software limit switch
SEMAX	Maximum software limit switch

Rotary axis

A rotary axis is an axis with an unlimited path (rotary table, continuous conveyor belt). For a rotary axis, the path starts and ends at the same physical point on the axis. Hardware and software limit switches are not considered.

For rotary axes, the path (increments) for a revolution is referred to as a modulo value. The modulo value corresponds to the resolution of the absolute encoder.

For rotary axes, limitations can be specified with reference to the direction of travel. Some rotary axes can only travel in a negative or in a positive direction, others can travel in both directions.

The permissible directions of travel are determined by bits 4 and 5 of the output word OUT[0] of the *Configure initiators and switching outputs* command (see page 5-11).

If rotary axis mode is active, the drive must be freely rotatable. If this is not possible, the axis will not meet the requirement for operation as a rotary axis and must, therefore, be parameterized as a linear axis.

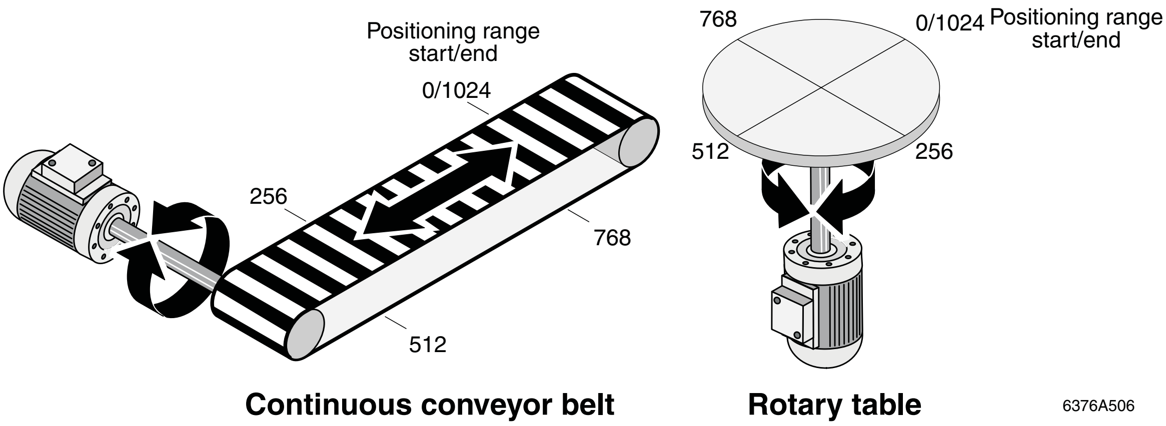


Figure 3-11 Example rotary axes

In Figure 3-11 the modulo value is set to 1024. This value has the same position on the axis as the value 0. The actual value range is indicated by the terminal as 0 to 1023. The value 1024 is not indicated.

The module value is not converted via the gear ratio, i.e., it has to be specified in increments.

Path optimization

For rotary axes, which can travel in both a positive and a negative direction, the direction of travel is selected so that the target position is reached via the shortest path.

Example

On a rotary table, the encoder resolution (modulo value) is set to 10 bits. The current position is 896 and the specified target position is 128. If it is possible to travel in a positive and a negative direction, the table takes the shortest path to position 128, namely via the start/end of the path (0/1024).

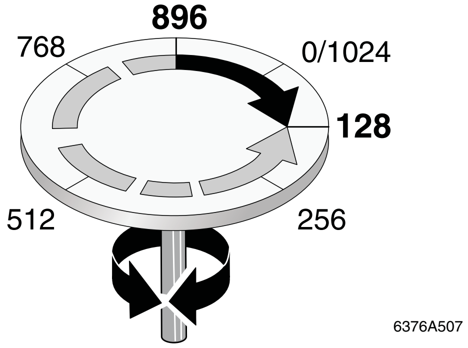


Figure 3-12 Example of path optimization

If travel were only permitted in the negative direction, the table would travel from position 896 through positions 768, 512, and 256 to position 128.

3.8 Jog Mode

In jog mode, the two external bits routed via the bus system ensure that the positioning terminal is able to control the movement of the drive in a positive or negative direction. In jog mode, the drive can only travel at creeping speed.

There are two variants of the jog mode function:

1 Drive travel as long as the control bit is activated

For this purpose, the "Delay time for continuous signal in jog mode" must be shorter than the "Monoflop time for jog mode".

2 Drive travel for a predefined period of time

Here, a low-high edge on the "Trigger jog mode" bit will initiate the output of a drive control pulse of a predefined length. This makes it possible for the drive to approach a setpoint position manually in very small increments, step by step. As the pulse time used to control the drive can be shorter than the response time of the operator (or even shorter than, e.g., the INTERBUS cycle time), this function supports more precise increments than are possible with variant 1.

The functions of the two variants can also be combined. In this case, a low-high edge on the control bit will initiate a drive control pulse on which the drive will travel a short distance. Following this, if the control bit remains pending, the drive will be controlled again once the "Delay time for continuous signal in jog mode" has elapsed until the control pulse is withdrawn.

In jog mode, you can set bit 2 (jog mode negative direction) or bit 1 (jog mode positive direction) of the control word (OUT[0]) to activate travel in the corresponding direction (see Section 5.13, "Control Commands").

Jog mode has the highest priority. This means that while jog mode is active, no other processes (positioning, looping) can be started. However, activating jog mode does not abort other processes.

For terminals with **firmware version 1.1 or later** both the software and hardware limit switches are **activated** for jog mode, although they can be deactivated.

3.9 Encoder Offset

When the encoder is installed, it might not be set to zero. An encoder offset can be set to avoid manually turning the encoder to the zero position.

Example

An encoder with a resolution of 12 bits is used. Therefore, a range from 0 to 4096 increments can be used.

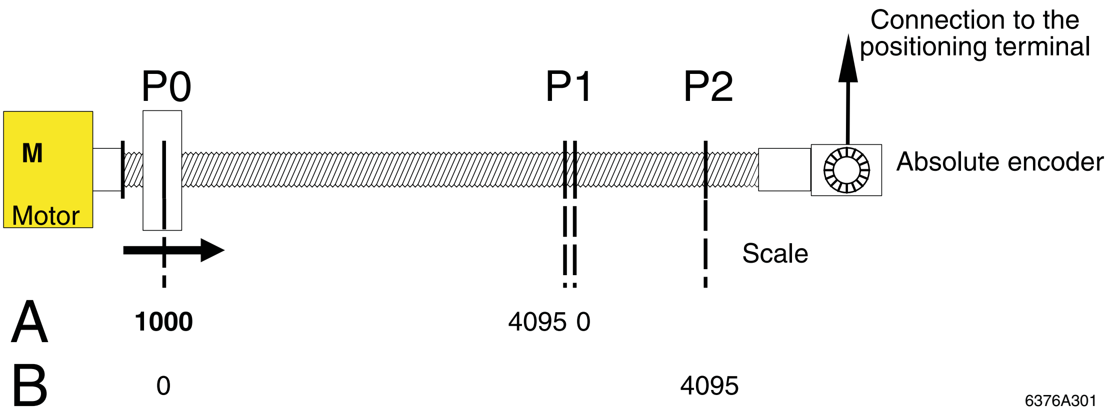


Figure 3-13 Example for encoder offset

The encoder has been installed and the drive is in the presumed zero position (P0).

Read the current position (see ""Control Position and Read Position" (ControlPosition, ReadPosition) Command" on page 5-45).

In the example, this position is equal to 1000 (A in Figure 3-13). If the drive now travels through position 4095 (P1), error message 18 "Counter overflow" is generated (see Table 5-10 "Error codes" on page 5-49). It is not possible to approach position P2.

To use the entire range of 4096 increments, position P0 must have the value 0. The encoder can also be removed and turned backwards. However, the terminal offers the option of assigning a certain position to this value using the *Define encoder offset* command. In the example, position P0 is to have the value "0 increments".

Define the encoder offset as 1000 (see ""Encoder Offset" (DefineEncoderOffset) Command" on page 5-23).

> A gear ratio should not yet have been set when defining the encoder offset.

> If you define the encoder offset with a value, which is not the same as the position read, you will still be operating with a limited range.

3.10 Actual Position Detection

The IB IL SSI ... terminal generates the actual position, taking into consideration the counting direction from the data flow, the logic offset and the gear ratio.

Reading the absolute encoder

To transmit the data from the absolute encoder correctly, the pulse train shown in Figure 3-14 is generated by the IB IL SSI ... terminal.

Data transmission is implemented by generating a clock pulse train every 33 pulses. The first pulse latches the position of the absolute encoder. The next 32 pulses are used to transmit the position of the encoder to the IB IL SSI .... There is a pause t_P of at least 100 µs between clock pulse trains.

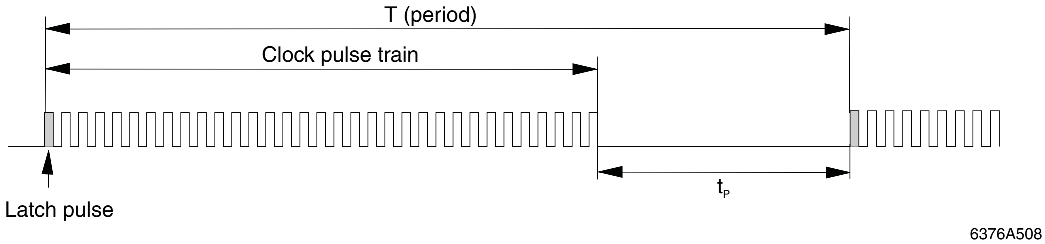


Figure 3-14 Clock sequence during data transmission

Latch pulse	Start signal for data transmission	
t_P	Pause between two clock pulse trains	t_P > 100 µs

The data transmission sequence is illustrated in Figure 3-15.

On the first pulse of a clock pulse train, the encoder writes the current position to its register. On the next falling edge of the clock signal, the most significant bit (D_n, MSB) is read, and on each subsequent falling edge, the next bit is read. If the least significant bit (D_0, LSB) is read, up to two other bits (special bits e.g., parity) are read for encoders with a corresponding function.

If, following transmission of bit D_0 or the special bits, signals are still pending at the clock input, the transmission sequence will be repeated starting with the MSB (D_n). D_X represents the last bit transmitted.

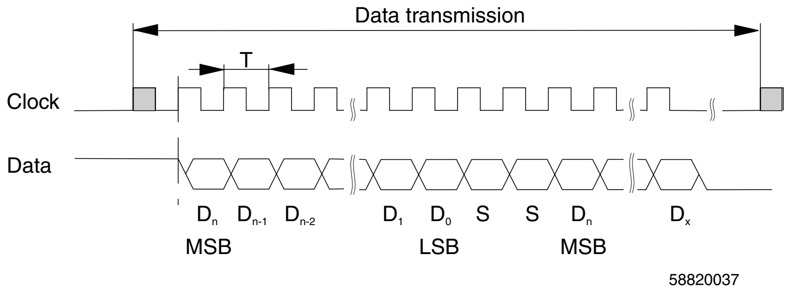


Figure 3-15 Data transmission

S	Special bit	
T	Duration of a clock signal	T = 2.5 µs

The encoder must be defined appropriately to enable the received data to be evaluated correctly (see ""Encoder Configuration" (DefineEncoder) Command" on page 5-8).

When data transmission is complete, the corresponding number of data bits is evaluated (masked out) by the module firmware, according to the configuration of the "Resolution" parameter.

The data flow determined is converted into an absolute position value according to the code (Gray code or binary code). The actual position is determined from this position value, taking into consideration the gear ratio and the logic offset.

The current actual position can be requested using the *Read position* command (see ""Control Position and Read Position" (ControlPosition, ReadPosition) Command" on page 5-45).

The resolution can be determined using the *Define encoder* command (see also ""Encoder Configuration" (DefineEncoder) Command" on page 5-8).

Count range for actual position

The defined range of values for the actual position is between -2^{25} and $2^{25} - 1$ increments.

On exiting the defined actual value range, the counter enters the overflow range. Pulse detection continues, but error code 18_{dec} is set in the status word ("Counter overflow"; see Table 5-10 "Error codes" on page 5-49).

If a rotary axis is being parameterized, the actual position value is always mapped in the specified path, which is determined for the IB IL SSI ... terminal by the resolution of the encoder.

Effect of the logic offset

The positioning terminal calculates the value displayed for the actual position on the basis of the gear ratio and the logic offset in accordance with the following formula:

$$P = G' \times \frac{A}{B} + O$$

Where:

P	Current position
G´	Value of the encoder minus encoder offset (if applicable, see page 3-20)
A	Gear ratio numerator
B	Gear ratio denominator
O	Logic offset

The offset is calculated using the formula:

$$O = S - G' \times \frac{A}{B}$$

Where:

O	Logic offset
G´	Value of the encoder minus encoder offset (if applicable, see page 3-20).
S	Setpoint position

G´ is calculated using the formula:

$G' = G - G0$

Where:

G	Encoder value
G0	Encoder offset (see page 3-20)

> Please note that when determining the current position, the logic offset must equal 0.

The current position can be determined using the *Read position* command.
(see ""Control Position and Read Position" (ControlPosition, ReadPosition) Command" on page 5-45).

Example

An encoder with a resolution of 12 bits is used. Therefore, a range from 0 to 4095 increments can be used.

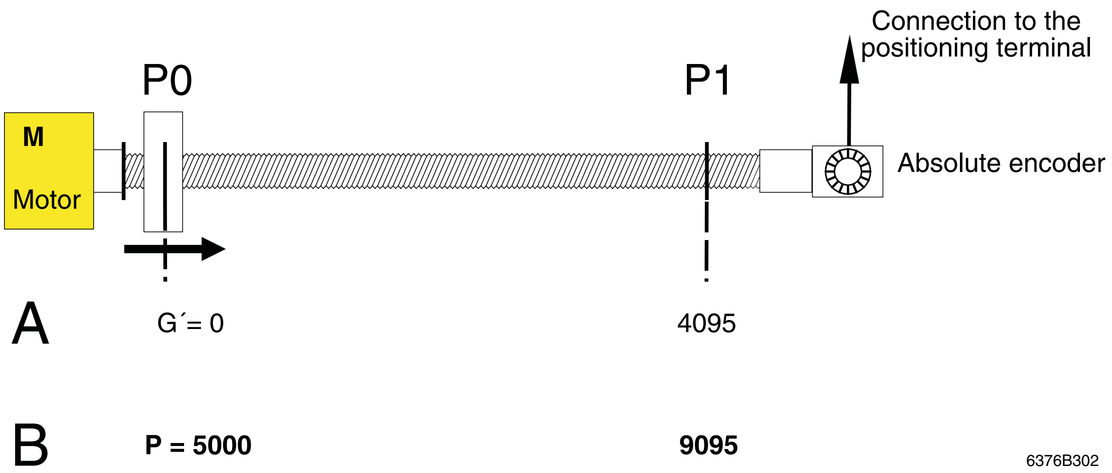


Figure 3-16 Logic offset example

The encoder has been installed and the drive is in the zero position (P0). This state is either reached immediately of after defining the encoder offset (see "Encoder Offset" on page 3-20).

It is now possible to position within the range 0 to 4095 increments (A). However, if you want to shift your positioning range to another system of coordinates, you can do this by defining the logic offset.

Set the logic offset, e.g., to 5000 (see ""Logic Offset" (DefineLogicOffset) Command" on page 5-19).

The range of values is now between 5000 and 9095 increments, as shown in B.

3.11 Gear Ratio

The *Define gear ratio* command can be used to specify the gear ratio, i.e., how many units of a measurement unit correspond to a certain number of increments (see also ""Gear Ratio" (DefineGearRatio) Command" on page 5-18).

A gear ratio can be used for linear and rotary axes.

Example

You want to specify the position in centimeters rather than increments. You know that the entire path is 200 centimeters long and that this represents 4000 increments.

Therefore

– Gear ratio numerator	GRN = 200 (cm)	(1 to 1023)
– Gear ratio denominator	GRD = 4000	(1 to $2^{16} - 1$)

$$\frac{GRN}{GRD} = \frac{200}{4000} = \frac{1}{20} \text{ (cm)}$$

All other values must now be specified in centimeters:

- Software limit switches
- Parameter records (start window, rapid start window, pre-stop window, stop window, target window, target position)
- Drive stop
- Logic offset

The terminal will also indicate all values (e.g., actual position) in centimeters.

> Do **not** change the gear ratio during operation.
> If the parameters specified above have not been defined for the corresponding gear ratio, entering or changing the gear ratio during operation will corrupt the positioning process.

> The user must set a numerator/denominator ratio for the gear ratio as appropriate for the mechanics. The gear ratio must be ≤ 1!

If you wish to have the resolution in micrometers, you must use an encoder on the axis, which supplies at least one increment per micrometer. Otherwise, you will not be able to set the required resolution via the gear ratio.

> Stay within the ranges of values for the numerator and denominator!

Example of a correct range of values

$$\frac{GRN}{GRD} = \frac{4000}{16000}$$

A GRN > value of 1023 will corrupt the positioning process.

The following value is, therefore, preferable:

$$\frac{GRN}{GRD} = \frac{400}{1600} = \frac{1}{4}$$

3.12 Output Assignment

The terminal uses digital outputs to control the speed and direction of travel of the drive. The output bits must be set accordingly for the various drives (pole-changing motors, Dahlander circuit, etc.) (the settings are not the same for each drive).

The bit combination the terminal works with is defined via the "Function of the switching outputs" parameter (see ""Configuration of Axis Types, Behavior in the Event of a Bus Failure, Initiators and Switching Outputs" (DefineInOut) Command" on page 5-11).

The available output versions are described in ""Configuration of Axis Types, Behavior in the Event of a Bus Failure, Initiators and Switching Outputs" (DefineInOut) Command" on page 5-11.

Do not damage the system!

Avoid damage by selecting a bit combination that is appropriate for your drive.

3.13 Hand-Held Operator Panel Mode

Hand-held operator panel mode is only designed for startup. It is activated by enabling (with a logic 1) hand-held operator panel mode at connector 3 (see "Terminal Point Assignment" on page 1-6).

An example of how to wire terminals can be found in "Wiring the Positioning Terminal When Using Hand-Held Operator Panel Mode" on page 2-13.

This section also describes the cable assembly of a connector specifically for hand-held operator panel mode. When hand-held operator panel mode is activated, the commands are generated via this connector and forwarded directly to the drive. Although the limit switches are monitored and this is indicated in the process data, they have no effect on drive control.

4 Process Data Operation

Process data is used to configure, control and read the positioning terminal.

4.1 Process Data Channel Assignment

Each positioning terminal's process image features two data words.

Ensure data consistency

Ensure data consistency of two words to prevent the possibility of the values being misinterpreted.

See also "Tips for Working With the Positioning Terminal" on page 6-1.

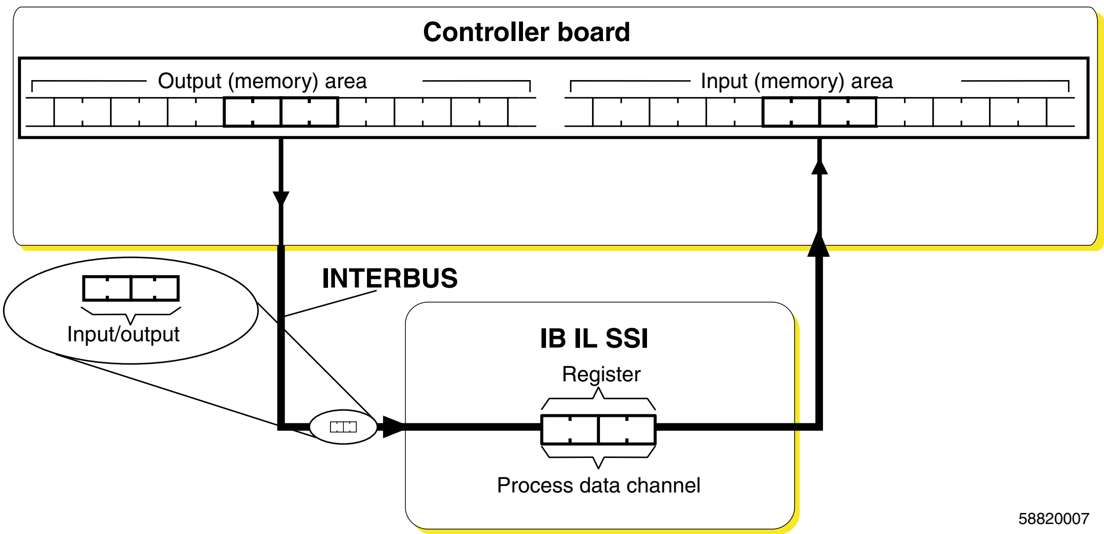


Figure 4-1 Process image in the I/O (memory) area of the controller board

The data words are located in the process data (memory) area on the controller board. This memory area is a process image of the entire application, i.e., the bus configuration. The addresses are assigned by means of the automatic or logical addressing of the controller board.

The process data (memory) area comprises an output (memory) area and an input (memory) area. The two memory areas do not necessarily have to be different.

Definition

Direction of output data flow:	From the controller board to the terminal
Direction of input data flow:	From the terminal to the controller board

4.2 Output Words

The IB IL SSI ... terminal is configured and controlled by means of different commands transmitted via the two output words.

The command code and, if necessary, the associated parameters, are transmitted from the controller board to the terminal via the output words. If no parameters are required, the assignment of the parameter bits is irrelevant.

Specify whether the system should write to a register of the terminal or read from this register using bit 15 of the first output data word (Read/Write; R/W). If R/W = 1, data is written to the register; if R/W = 0, data is read from the register.

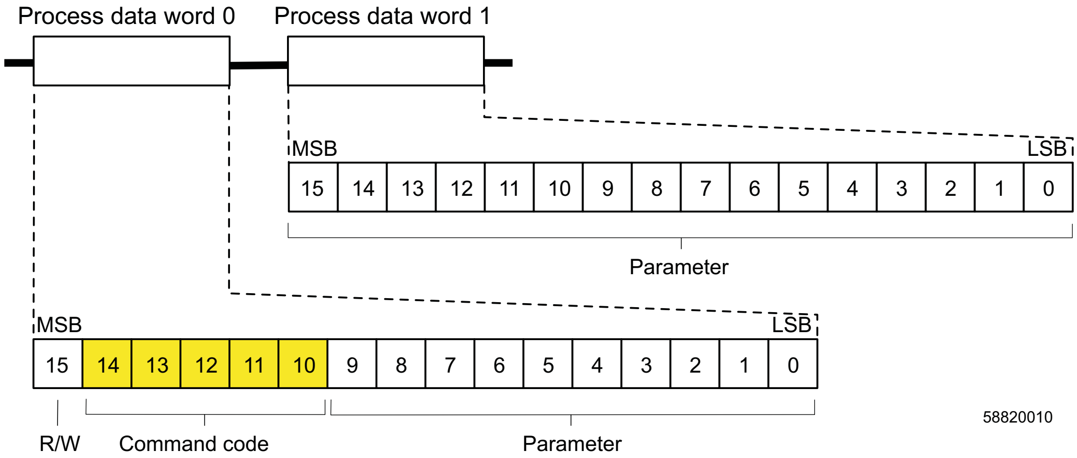


Figure 4-2 Assignment of the output words

Valid command codes are listed in "Commands for Working With the Positioning Terminal" on page 5-1.

4.3 Input Words

Each IB IL SSI ... terminal uses two input words.

During parameterization, the output words are mirrored in the input words (command code and if applicable, appropriate parameters).

When reading parameters and the actual position, the required value will be written to the input word.

In bit 15 of the first input data word, the R/W bit of the output data word is not mirrored. An error bit is set here, if one of the following is true:

- The terminal has not yet been completely configured.
- There is an invalid parameter in the default operating mode.
- A reserved bit has been set.

Read the status for more information about the error.

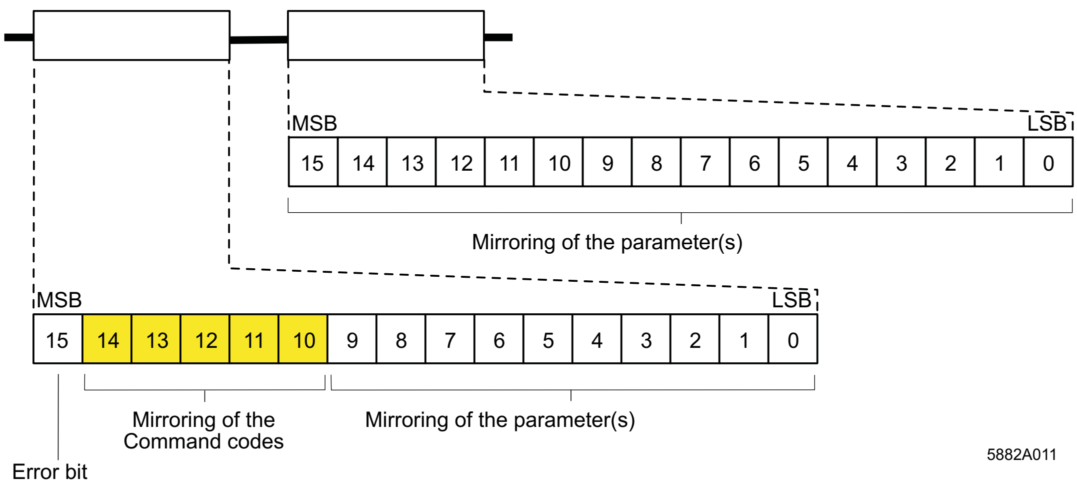


Figure 4-3 Assignment of the input words

5 Commands for Working With the Positioning Terminal

Various types of commands are available for working with the positioning terminal:
– Commands for configuring position value detection
– Commands for configuring the positioning system
– Commands for parameterizing the path
– Commands for controlling functions

In order to run a positioning process, the commands highlighted in bold in Table 5-2 on page 5-4 need to be executed as a minimum.

5.1 Overview of Commands

Communication with the positioning terminal can only take place via the process data channel. It works like this:
– A command is sent to the module.
– The module detects the command and acknowledges this by mirroring the command.
– In the case of commands used to write parameters, the module also mirrors the parameters.
– Commands used to read a value are mirrored and the value read is sent in the remaining bits.
– Commands used to send a parameter comprise both the command code and the parameter.

Generating Codes for Parameterization

The basic code for the relevant command is entered as the command code.

You will need to add the code associated with the required parameters to this basic code or link the two codes (basic code and parameter code) using the OR function.

Example

Define encoder command code (see page 5-4)	**90**00 0000
Parameters (see page 5-8)	PAR = 01_{bin}; REV = 1_{bin}; RESOLUTION = 00101_{bin}; CODE = 1_{bin}
Parameter code (leading zeros added)	00**11 0501**
Code for parameterization	**9011 0501**

Command Descriptions (Default Values)

The **default values** preset on the positioning terminal appear in **bold** in the data relating to valid ranges of values for parameters.

For some parameters, the default value is not a valid value for parameterization. In this case, a value **appropriate for your system configuration must** be set (e.g., switching outputs, see page 5-12). If you do not make this setting, the terminal will generate an error message.

The command and bit names appear in *italics*. This mode of representation is the same as that used in Phoenix Contact programs (as, for example, in function blocks).

Representation of Output Words and Codes

For the assignment of output words, the bit assignment is displayed in binary and hexadecimal representation.

A	X (4)

In order to more effectively distinguish between adjacent nibbles, these are displayed against a black/white background.

Here, the code is indicated in hexadecimal representation.

A The code is unique.

X The code is determined by the assignment of the individual bits.

4 or X
- X The code is determined by the assignment of the individual bits.
- (4) Code for basic code; bits, which do not belong to the basic code, are still at 0.

C/E Depending on a bit position, the code will either be C or E.

In a code, nibble, byte or word limits are indicated by a blank space.

Data Types Associated With Parameters:

The data type of a parameter is indicated for each individual parameter. The following data types are used:

Table 5-1 Data types associated with parameters

Designation	Example	Meaning
INTx	INT26	Signed value, displayed in x bits.
USIGNx	USIGN5 USIGN8 USIGN10 USIGN12 USIGN16 USIGN26	Unsigned value, displayed in x bits.
–		Parameters for which a data type has not been indicated are displayed as bit strings.

In respect of the representation of values in your control or computer system, please note:

– In the case of an INT26 value displayed in 32 bits, the sign bit (bit 25; word[1], bit 9) is copied to the free most significant bits (bits 26 to 31).
  If bit 25 = 0, bits 26 to 31 must be set to 0.
  If bit 25 = 1, bits 26 to 31 must be set to 1.
– In the case of a USIGN value, the free most significant bits must be set to 0.
  Example: A USIGN10 value is saved in a word (16 bits). This means that bits 10 to 15 are set to 0.

Table 5-2 Overview of commands for working with the positioning terminal

Bits 15 Through 10 (bin) (OUT[0])	Command Code (Hex) (Basic Code) Write/Read	Remark	*Command*	Default Value	Page
Configuration of position detection					
1001 00 0001 00	**9000 0000/ 1000 0000**	**Define encoder/Read encoder configuration**	***DefineEncoder/ ReadEncoderConfiguration***	See page 5-8	5-8
1001 01 0001 00	**9400 0000/ 1400 0000**	**Define configuration of axis types, behavior in the event of a bus failure, initiators and switching outputs/Read configuration of axis types, behavior in the event of a bus failure, initiators and switching outputs**	***DefineInOut/ ReadInOut***	See page 5-11	5-11
1010 00 0010 00	A000 0000/ 2000 0000	Define gear ratio/ Read gear ratio	*DefineGearRatio/ ReadGearRatio*	01 0001_{hex} Factor 1	5-18
1010 01 0010 01	A400 0000/ 2400 0000	Define logic offset/ Read logic offset	*DefineLogicOffset/ Read logic offset*	0	5-19
1011 00 0011 00	B000 0000/ 3000 0000	Define encoder offset/ Read encoder offset	*Define encoderOffset/ Read encoder offset*	0	5-23

Table 5-2 Overview of commands for working with the positioning terminal (Continued)

Bits 15 Through 10 (bin) (OUT[0])	Command Code (Hex) (Basic Code) Write/Read	Remark	*Command*	Default Value	Page
Configuration of the positioning system					
1001 10	9800 0000/ 1800 0000	Define drive stop/Read drive stop	*DefineDrvStop/* *ReadDrvStop*	0A 0001_{hex} 1 increment/s	5-16
1001 11 0001 11	9C00 0000/ 1C00 0000	Define delay time for detection of direction and stop and output delay time/ Read delay time for detection of direction and stop and output delay time	*DefineDsdOd* *ReadDsdOd*	00 $000A_{hex}$ 0 ms/10 ms	5-22
1010 10 0010 10	A800 0000/ 2800 0000	Define minimum software limit switch/ Read minimum software limit switch	*DefineMinSWLimSwitch/* *ReadMinSWLimSwitch*	-2^{25}	5-20
1010 11 0010 11	AC00 0000/ 2C00 0000	Define maximum software limit switch/ Read maximum software limit switch	*DefineMaxSWLimSwitch/* *ReadMaxSWLimSwitch*	$2^{25} - 1$	5-21
1011 01	B400 0000/ 3400 0000	Reserved	Reserved		
1011 10	B800 0000/ 3800 0000	Reserved	Reserved		
0011 11	3C00 0000	Read firmware version	*ReadFirmwareVersion*	–	5-17
1001 01 0001 01	9500 0000/ 1500 0000	Define jog mode/ Read jog mode	*DefineJogMode/* *ReadJogMode*	–	5-40

Table 5-2 Overview of commands for working with the positioning terminal (Continued)

Bits 15 Through 10 (bin) (OUT[0])	**Command Code (Hex) (Basic Code) Write/Read**	**Remark**	***Command***	**Default Value**	**Page**
Parameter for path (record 1) (*Parameter record 1*)					
1100 00 0100 00	C000 0000/ 4000 0000	Define start window/ Read start window	*DefineStartWindow/ ReadStartWindow*	0	5-25
1100 01 0100 01	C400 0000/ 4400 0000	Define rapid start window/ Read rapid start window	*Define RapidStartWindow/ ReadRapidStartWindow*	0	5-27
1100 10 0100 10	C800 0000/ 4800 0000	Define rapid stop window/ Read rapid stop window	*DefineRapidStopWindow/ ReadRapidStopWindow*	0	5-29
1100 11 0100 11	CC00 0000/ 4C00 0000	Define pre-stop window/ Read pre-stop window	*DefinePre-StopWindow/ ReadPre-StopWindow*	0	5-31
1101 00 0101 00	D000 0000/ 5000 0000	Define stop window/ Read stop window	*DefineStopWindow/ ReadStopWindow*	0	5-33
1101 01 0101 01	D400 0000/ 5400 0000	Define target window/ Read target window	*DefineTargetWindow/ ReadTargetWindow*	0	5-35
1101 10 0101 10	**D800 0000/ 5800 0000**	**Define target position**/ **Read target position**	***DefineTargetPosition/ ReadTargetPosition***	0	5-37
1101 11 0101 11	DC00 0000/ 5C00 0000	Define lubrication and friction compensation value/ Read lubrication and friction compensation value	*DefineLubricationAndFriction CompensationValue/ ReadLubricationAndFriction CompensationValue*	0	5-39
Parameter for path (record 2) (*Parameter record 2*)					
1110 00 0110 00	E000 0000/ 6000 0000	Define start window/ Read start window	*DefineStartWindow/ ReadStartWindow*	0	5-25
1110 01 0110 01	E400 0000/ 6400 0000	Define rapid start window/Read rapid start window	*Define RapidStartWindow/ ReadRapidStartWindow*	0	5-27
1110 10 0110 10	E800 0000/ 6800 0000	Define rapid stop window/ Read rapid stop window	*DefineRapidStopWindow/ ReadRapidStopWindow*	0	5-29
1110 11 0110 11	EC00 0000/ 6C00 0000	Define pre-stop window/ Read pre-stop window	*DefinePre-StopWindow/ ReadPre-StopWindow*	0	5-31
1111 00 0111 00	F000 0000/ 7000 0000	Define stop window/ Read stop window	*DefineStopWindow/ ReadStopWindow*	0	5-33
1111 01 0111 01	F400 0000/ 7400 0000	Define target window/ Read target window	*DefineTargetWindow/ ReadTargetWindow*	0	5-35
1111 10 0111 10	**F800 0000**/ 7800 0000	**Define target position/ Read target position**	***DefineTargetPosition/ ReadTargetPosition***	0	5-37
1111 11 0111 11	FC00 0000/ 7C00 0000	Define lubrication and friction compensation value/ Read lubrication and friction compensation value	*DefineLubricationAndFriction CompensationValue/ ReadLubricationAndFriction CompensationValue*	0	5-39

Table 5-2 Overview of commands for working with the positioning terminal (Continued)

Bits 15 Through 10 (bin) (OUT[0])	**Command Code (Hex) (Basic Code) Write/Read**	**Remark**	***Command***	**Default Value**	**Page**
Control functions					
1000 00 0000 00	**8000 0000/ 0000 0000**	**Control position, read position/ Read position**	***ControlPosition, ReadPosition/ ReadPosition***	0	5-45
1000 01 0000 01	8400 0000/ 0400 0000	Control position, read velocity/ Read velocity	*ControlPosition, ReadVelocity/ ReadVelocity*	–	5-46
1000 10 0000 10	8800 0000/ 0800 0000	Control position, read status/ Read status	*ControlPosition, ReadStatus/ ReadStatus*	–	5-47

5.2 "Encoder Configuration" *(DefineEncoder)* Command

You can use command 9000 0000_{hex} to define the type of encoder connected to the terminal and how that encoder is evaluated. Define the encoder by using the OR function to link command 9000 0000_{hex} to the bits listed below. x indicates dependence on the value of the parameter.

To read the encoder definition from the corresponding register on the terminal, you need to send command 1000 0000_{hex} (bit 15 = 0).

Output word 0 (OUT[0])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1/0	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**PAR**		**0**	**0**	**0**	**ID**
9 (Write)/1 (Read)				**0**				**X**				**X**			

Output word 1 (OUT[1])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	**0**	**0**	**RESO**					**0**	**0**	**0**	**0**	**0**	**0**	**0**	**CODE**
X				**X**				**0**				**X**			

Permissible Values for *DefineEncoder*:

Code (bin)	PAR: Parity *(Parity)*
00	**None**
01	Even
10	Odd
11	Reserved

Code (bin)	ID: Encoder installation direction *(Enc Rotation)*
0	**Normal**
1	Inverted

Code (bin)	RESO: Resolution *(Resolution)*	Code (bin)	RESO: Resolution *(Resolution)*
00000	**Invalid**	10000	23
00001	8	10001	24
00010	9	10010	25
00011	10	10011	26*
00100	11	10100	Reserved
00101	12	10101	
00110	13	10110	
00111	14	10111	
01000	15	11000	
01001	16	11001	
01010	17	11010	
01011	18	11011	
01100	19	11100	
01101	20	11101	
01110	21	11110	
01111	22	11111	

Code (bin)	CODE: Code *(Code)*
0	**Binary code**
1	Gray code

PAR: Parity

If you have configured a parity check for your encoder and the encoder supports this function, the parity check will be carried out when the current actual position is detected. In the event of this parity check returning an error twice in succession, the error message will be generated with error code 5_{dec} ("A parity error has occurred") (see Table 5-10 on page 5-49).

ID: Encoder installation direction

The installation direction allows you to invert the interpretation of the encoder's code string (count direction of the encoder and coordinate system).

If, for example, you have installed an encoder so that the position counter counts up when the axis is rotating in a positive direction, you can set the ID bit to parameterize the positioning terminal to count down when the axis is rotating in a positive direction.

This provides you, for example, with the advantage that one and the same encoder can be used on both sides of the axis without range limits having to be changed.

RESO: Resolution

The resolution for the IB IL SSI ... terminal must be the same as the resolution supported by the absolute encoder.

Example:

A multi-turn encoder is used with the 12 bits/12 bits parameter. This encoder indicates a position in the following terms:

– Number of revolutions (displayed in 12-bit format)
– Resolution per revolution (displayed in 12-bit format)

The resolution is, therefore, 24 bits (12 bits + 12 bits).

CODE: Code

Depending on type, an encoder will work in Gray code or binary code. The Code parameter must be set accordingly.

5.3 "Configuration of Axis Types, Behavior in the Event of a Bus Failure, Initiators and Switching Outputs" *(DefineInOut)* Command

You can use command 9400 0000hex to write the parameters for:

– Axis type
– Behavior in the event of a bus system failure
– Switching outputs
– Initiator function

To read the parameters from the corresponding register on the terminal, you need to send command 1400 0000_{hex} (bit 15 = 0).

Output word 0 (OUT[0])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1/0	**0**	**0**	**1**	**0**	**1**	**0**	**0**	**0**	**0**	**AXIS**		**0**	**0**	**0**	**AABR**
9 (Write)/1 (Read)				**4**				**X**				**X**			

Output word 1 (OUT[1])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	**OUTF**			**0**	**IN3F**			**0**	**IN2F**			**0**	**IN1F**		
X				**X**				**X**				**X**			

Permissible Values for *DefineInOut*:

Code (bin)	AXIS: Axis Type (Rotary Axis/Linear Axis) *(AxisType)* (See Also "Axis Types" on page 3-16)
00	**Linear axis**
01	Rotary axis: Movement only possible in negative direction of rotation
10	Rotary axis: Movement only possible in positive direction of rotation
11	Rotary axis: Negative and positive direction of rotation

Code (bin)	AABR: Action of Outputs at Bus Reset *(Action at Bus Reset)*
0	**In the event of a bus reset, the switching outputs are reset and, as a consequence, any positioning processes in progress stopped.**
1	Resetting the bus has no effect on positioning processes. This means that the inputs can continue to start positioning processes.

Code (bin)	OUTF: Switching Outputs (Output Version) *(Output Functions)*
000	**Invalid**
001	Version 1
010	Version 2
011	Version 3
100	Version 4
101	Version 5
110	Reserved
111	Reserved

The possible versions are listed in "Function of the Switching Outputs (OutputFunction)" on page 5-14.

If, in the case of the following parameters, the bit marked "x" = 1, the inverse input function will be executed. This means that the switches will be active at low or the function will be triggered on an edge which is the opposite of that indicated.

Code (bin)	Active	IN3F: Function initiator 3 *(Initiator3Function)*
x00	–	**Reserved**
x01	↑	Start positioning with parameter record 2
x10	↑	Start positioning with parameter record 1
x11	•	None (digital input only)

Code (bin)	Active	IN2F: Function initiator 2 *(Initiator2Function)*
x00	•	**Maximum limit switch**
x01	↓	Maximum limit switch
x10	↑	Start positioning with parameter record 2
x11	•	None (digital input only)

Code (bin)	Active	IN1F: Function initiator 1 *(Initiator1Function)*
x00	•	**Minimum limit switch**
x01	↓	Minimum limit switch
x10	↑	Start positioning with parameter record 1
x11	•	None (digital input only)

• Signal is level-driven (signal is evaluated when set to 1)

↑ Signal is edge-driven (signal is evaluated when the bit changes from 0 to 1)

↓ Signal is edge-driven (signal is evaluated when the bit changes from 1 to 0)

AXIS: Axis type

The positioning terminal can be used to automate both linear and rotary axes. The "Axis Type" variable is used to select the operating mode of the axis. The positioning function will differ depending on the type of axis:

– If a rotary axis is selected, the hardware limit switches and the software limit switches will have no function (see "Axis Types" on page 3-16).

AABR: Action at bus reset

Bit 0 in the output word OUT[0] (AABR) can be used to parameterize the behavior of the switching outputs in the event of a bus reset (bus failure). If the bit is reset (AABR = 0), the switching outputs will also be reset in the event of a bus reset and, as a consequence, any positioning processes in progress will be stopped. If the bit is set (AABR = 1) resetting the bus will have no effect on positioning. This means that the inputs can continue to start new positioning processes.

OUTF: Switching outputs

The terminal uses digital outputs to control the speed at and direction in which the drive is traveling. The output bits must be set accordingly for the various drives (pole-changing motors, Dahlander circuit, etc.) (the settings are not the same for each drive).

The bit combination the terminal works with is defined via the "Function of the switching outputs" parameter (OUTF).

INxF: Initiators

The initiators can be used as limit switches or as control inputs. If the initiators are used as control inputs, positioning processes can be started by means of external events. In order that the control function can be executed, as well as the initiator concerned being parameterized accordingly, the parameter record must also be enabled by means of the control command for external positioning (ELC2 or ELC1 in the control word; see "Control Commands" on page 5-42). An external control command will then only be accepted if no other positioning processes are in progress at the time.

5.3.1 Function of the Switching Outputs *(OutputFunction)*

Table 5-3 Version 1

State	Output 1	Output 2	Output 3	Output 4
Stop	0	0	0	0
Positive creeping motion	0	1	1	0
Positive rapid motion	1	1	1	0
Negative creeping motion	0	1	0	1
Negative rapid motion	1	1	0	1

Table 5-4 Version 2

State	Output 1	Output 2	Output 3	Output 4
Stop	0	0	0	0
Positive creeping motion	0	1	1	0
Positive rapid motion	1	0	1	0
Negative creeping motion	0	1	0	1
Negative rapid motion	1	0	0	1

Table 5-5 Version 3

State	Output 1	Output 2	Output 3	Output 4
Stop	0	0	0	0
Positive creeping motion	0	1	0	0
Positive rapid motion	1	0	0	0
Negative creeping motion	0	0	1	0
Negative rapid motion	0	0	0	1

Table 5-6 Version 4

State	Output 1	Output 2	Output 3	Output 4
Stop	0	0	0	0
Brake	0	0	1	1
Positive creeping motion	1	0	0	0
Positive rapid motion	1	1	0	0
Positive fast motion	1	1	1	0
Negative creeping motion	1	0	0	1
Negative rapid motion	1	1	0	1
Negative fast motion	1	1	1	1

Table 5-7 Version 5

State	Output 1	Output 2	Output 3	Output 4
Outputs, which can be freely controlled via a control command (see Table 5-8 on page 5-42)	OUT 1	OUT 2	OUT 3	OUT 4

5.4 "Drive Stop" *(DefineDrvStop)* Command

Define the drive stop by assigning a maximum path to be traveled within a specific period of time. If this maximum path is traveled within the time, the terminal will detect a drive stop.

See also Section 3.2, "Drive Stop".

Define the drive stop by using the OR function to link command 9800 0000_{hex} to the variables for Time and Distance.

To read the parameters, you need to send command 1800 0000_{hex} (bit 15 = 0).

Output word 0 (OUT[0])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1/0	**0**	**0**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**TIME (USIGN5)**				
9 (Write)/1 (Read)				**8**				**X**				**X**			

Output word 1 (OUT[1])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSTD (USIGN16)															
X				**X**				**X**				**X**			

Permissible ranges of values for *DefineDrvStop*:

Code (dec)	TIME: Time for Drive Stop *(DefineStopTime)*
0	No monitoring of drive stop
1 to 31	0.1 s to 3.1 s
10	**1 s**

Code (dec)	DSTD: Drive Stop Distance *(DefineStopDistance)*
0 to $2^{16} - 1$	0 increments to $2^{16} - 1$ increments
1	**1 increment**

5.5 "Read Firmware Version" Command *(ReadFirmwareVersion)*

You can use command 3C00 0000_{hex} to read the firmware version of your positioning terminal. This command can be used at any time. The result is shown immediately in input word IN[1].

Output word 0 (OUT[0])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	**0**	**1**	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
3 (Read)				**C**				**0**				**0**			

The output word 1 is 0000_{hex}.

Input word 0 (IN[0])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST	**0**	**1**	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
X (3)				**C**				**0**				**0**			

Example

Input word 1 (IN[1])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Firmware version															
0	**0**	**0**	**1**	**0**	**0**	**1**	**0**	**0**	**0**	**1**	**1**	**X**	**X**	**X**	**X**
1				**2**				**3**				**A**			

In this example input word IN[1] has the value $123A_{hex}$. The firmware version is 1.23. The value of bits 3 to 0 of input word 1 is not relevant.

5.6 "Gear Ratio" *(DefineGearRatio)* Command

Define the gear ratio by using the OR function to link command A000 0000_{hex} with the numerator and denominator of the gear ratio.

To read the parameters, you need to send command 2000 0000_{hex} (bit 15 = 0).

The gear ratio can be used to scale the position value. The gear ratio can, therefore, be used to assign all paths in a unit of your choice (e.g., mm or cm, see also "Gear Ratio" on page 3-25).

Output word 0 (OUT[0])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1/0	**0**	**1**	**0**	**0**	**0**	**NGR (USIGN10)**									
A (Write)/2 (Read)				**0 or X**				**X**				**X**			

Output word 1 (OUT[1])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DGR (USIGN16)															
X				**X**				**X**				**X**			

Permissible ranges of values for *DefineGearRatio*:

Code (dec)	NGR: Gear Ratio Numerator *(NumeratorGearRatio)*
1 to 1023	**1 increment** to 1023 increments

Code (dec)	DGR: Gear Ratio Denominator *(DenominatorGearRatio)*
1 to $2^{16} - 1$	**1 increment** to $2^{16} - 1$ increments

5.7 "Logic Offset" *(DefineLogicOffset)* Command

The position values can be shifted by defining the logic offset.

> You should not define the logic offset until **after** you have defined the gear ratio, as the logic offset has to be indicated in the corresponding unit (see also "Effect of the logic offset" on page 3-23).

> Remember the difference between the logic offset and the encoder offset! See ""Encoder Offset" (DefineEncoderOffset) Command" on page 5-23 and "Encoder Offset" on page 3-20.

Define the logic offset by using the OR function to link command A400 0000_{hex} to the 26-bit variables for the logic offset.

To read the logic offset, you need to send command 2400 0000_{hex} (bit 15 = 0).

The logic offset is a 26-bit value.

Output word 0 (OUT[0])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1/0	**0**	**1**	**0**	**0**	**1**	**LO (26-bit value; INT26)**									
A (Write)/2 (Read)				**4 or X**				**X**				**X**			

Output word 1 (OUT[1])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LO (26-bit value; INT26)															
X				**X**				**X**				**X**			

Permissible range of values for *DefineLogicOffset*:

Code (dec)	LO: Logic offset *(LogicOffset)*
-2^{25} to $+(2^{25} - 1)$	-2^{25} increments to $+(2^{25} - 1)$ increments
0	**0 increments**

5.8 "Software Limit Switch" *(MaximumSoftwareLimitSwitch, MinimumSoftwareLimitSwitch)* Command

Taking into account the gear ratio, the software limit switches define the minimum and maximum position of the path. The software limit switches are specified as 26-bit values.

Before positioning starts, the system checks whether the software limit switches have been overrun. If the target position is outside the software limit switches, positioning will not start and an error message with error code 10_{dec}will be generated (see "Error codes" on page 5-49).

Please read the notes about software limit switches in rotary axis mode and jog mode in "Software Limit Switches" on page 3-15.

5.8.1 "Minimum Software Limit Switch" *(DefineMinSWLimSwitch)* Command

Define the minimum software limit switch by using the OR function to link command A800 0000_{hex} to the variables for the minimum software limit switch.

To read the minimum software limit switch, you need to send command 2800 0000_{hex} (bit 15 = 0).

Output word 0 (OUT[0])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1/0	**0**	**1**	**0**	**1**	**0**	**MINL (26-bit value; INT26)**									
A (Write)/2 (Read)				**8 or X**				**X**				**X**			

Output word 1 (OUT[1])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MINL (26-bit value; INT26)															
X				**X**				**X**				**X**			

Permissible range of values for *DefineMinSWLimSwitch*:

Code (dec)	MINL: Minimum software limit switch *(MinSWLimSwitch)*
-2^{25} to $+(2^{25} - 1)$	-2^{25} **increments** to $+(2^{25} - 1)$ increments

5.8.2 "Maximum Software Limit Switch" *(DefineMaxSWLimSwitch)* Command

Define the maximum software limit switch by using the OR function to link command AC00 0000_{hex} to the variables for the maximum software limit switch.

To read the maximum software limit switch, you need to send command 2C00 0000_{hex} (bit 15 = 0).

Output word 0 (OUT[0])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1/0	**0**	**1**	**0**	**1**	**1**	**MAXL (26-bit value; INT26)**									
A (Write)/2 (Read)				**C or X**				**X**				**X**			

Output word 1 (OUT[1])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAXL (26-bit value; INT26)															
X				**X**				**X**				**X**			

Permissible range of values for *DefineMaxSWLimSwitch*:

Code (dec)	MAXL: Maximum software limit switch *(MaxSWLimSwitch)*
-2^{25} to **$+(2^{25} - 1)$**	-2^{25} increments to **$+(2^{25} - 1)$ increments**

5.9 "Delay Time for Detection of Direction and Stop and Output Delay Time" *(DefineDsdOd)* Command

Define the Delay time for detection of direction and stop and Output delay time parameters by using the OR function to link command 9C00 0000_{hex} with the variables.

To read the parameters, you need to send command 1C00 0000_{hex} (bit 15 = 0).

The delay time for detection of direction and stop defines the time during which monitoring of the drive stop is deactivated once the drive has started. This enables the mechanics to start up during a positioning process without a stop being detected if drive stop monitoring is active.

If contactors are connected to the outputs of the IB IL SSI ... positioning terminal to control the drive, prior to activation, delay times must be observed prior to the individual contactors being triggered. Otherwise, there is a risk of short circuits if a contactor picks up more quickly than the previous one drops out. These delay times are defined using the "Output delay time" variable.

Output word 0 (OUT[0])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1/0	**0**	**0**	**1**	**1**	**1**	**0**	**0**	**DSD (USIGN8)**							
9 (Write)/1 (Read)				**C**				**X**				**X**			

Output word 1 (OUT[1])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	**0**	**0**	**0**	**ODT (USIGN12)**											
0				**X**				**X**				**X**			

Permissible ranges of values for *DefineDsdOd*:

Code (dec)	DSD: Delay Time for Detection of Direction and Stop *(DriveStartingDelay)*
0 to 255	**0 s** to 25.5 s

Code (dec)	ODT: Output Delay Time *(OutDelayTime)*
0 to 4095	0 ms to 4095 ms
10	**10 ms**

5.10 "Encoder Offset" *(DefineEncoderOffset)* Command

Define the encoder offset by using the OR function to link command B000 0000_{hex} to the encoder offset and sending this command to the terminal.

To read the encoder offset, you need to send command 3000 0000_{hex} (bit 15 = 0).

The encoder offset physically shifts the representation range on the basis of increments.

> Define the encoder offset **before** you define the gear ratio (see also "Encoder Offset" on page 3-20).

> Remember the difference between the encoder offset and the logic offset! See ""Logic Offset" (DefineLogicOffset) Command" on page 5-19 and "Effect of the logic offset" on page 3-23.

The encoder offset is specified as a 26-bit value.

Output word 0 (OUT[0])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1/0	**0**	**1**	**1**	**0**	**0**	**ENCO (26-bit value; USIGN26)**									
B (Write)/3 (Read)				**X (0)**				**X**				**X**			

Output word 1 (OUT[1])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENCO (26-bit value; USIGN26)															
X				**X**				**X**				**X**			

Permissible range of values for *DefineEncoderOffset*:

Code (dec)	ENCO: Encoder Offset *(EncoderOffset)*
0	**No** encoder offset
1 to $2^{26} - 1$	1 increment to $2^{26} -$ 1 increments

> The value for the encoder offset must not exceed the resolution of the encoder.

5.11 Commands for Parameterizing the Path

See also "Positioning and Structure of a Position" on page 3-3.

The positioning terminal supports an option to use two positioning data records to parameterize two independent paths.

A path is defined by the following parameters:

– Start window
– Rapid start window (version 5 of "Function of switching outputs" only)
– Rapid stop window (version 5 of "Function of switching outputs" only)
– Pre-stop window
– Stop window
– Target window
– Target position
– Friction compensation value (only if bit 14 (ALFC) in control word 1 is set)

Bit 13 of output word 0 is used to select the parameter record for each individual parameter (parameter record x; **P***arameter* **r***ecord* **x**; PRx).

– PRx = 0: Parameterization for parameter record 1
– PRx = 1: Parameterization for parameter record 2

5.11.1 "Start Window" *(StartWindow)* Command

Define the start window by using the OR function to link command C000 0000_{hex} for parameter record 1 or E000 0000_{hex} for parameter record 2 to the value for the start window and sending this command to the terminal.

To read the value, you need to send command 4000 0000_{hex} for parameter record 1 (bit 15 = 0) or 6000 0000_{hex} (bit 15 = 0) for parameter record 2.

Output word 0 (OUT[0])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1/0	**1**	**PRx**	**0**	**0**	**0**	**SW1/SW2 (26-bit value; USIGN26)**									
C/E (Write)/4/6 (Read)				**0 or X**				**X**				**X**			

Output word 1 (OUT[1])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SW1/SW2 (26-bit value; USIGN26)															
X				**X**				**X**				**X**			

PRx = 0: Parameterization for parameter record 1 (***P**arameter **r**ecord 1)*
PRx = 1: Parameterization for parameter record 2 (***P**arameter **r**ecord 2*)

If output version 4 is parameterized, the start window will only be taken into account during looping (looping, backlash compensation).

Permissible range of values for *StartWindow*:

Code (dec)	SW1/SW2: Start Window (*StartWindow*)
0 to $2^{26} – 1$	**0 increments** to 2^{26} – 1 increments

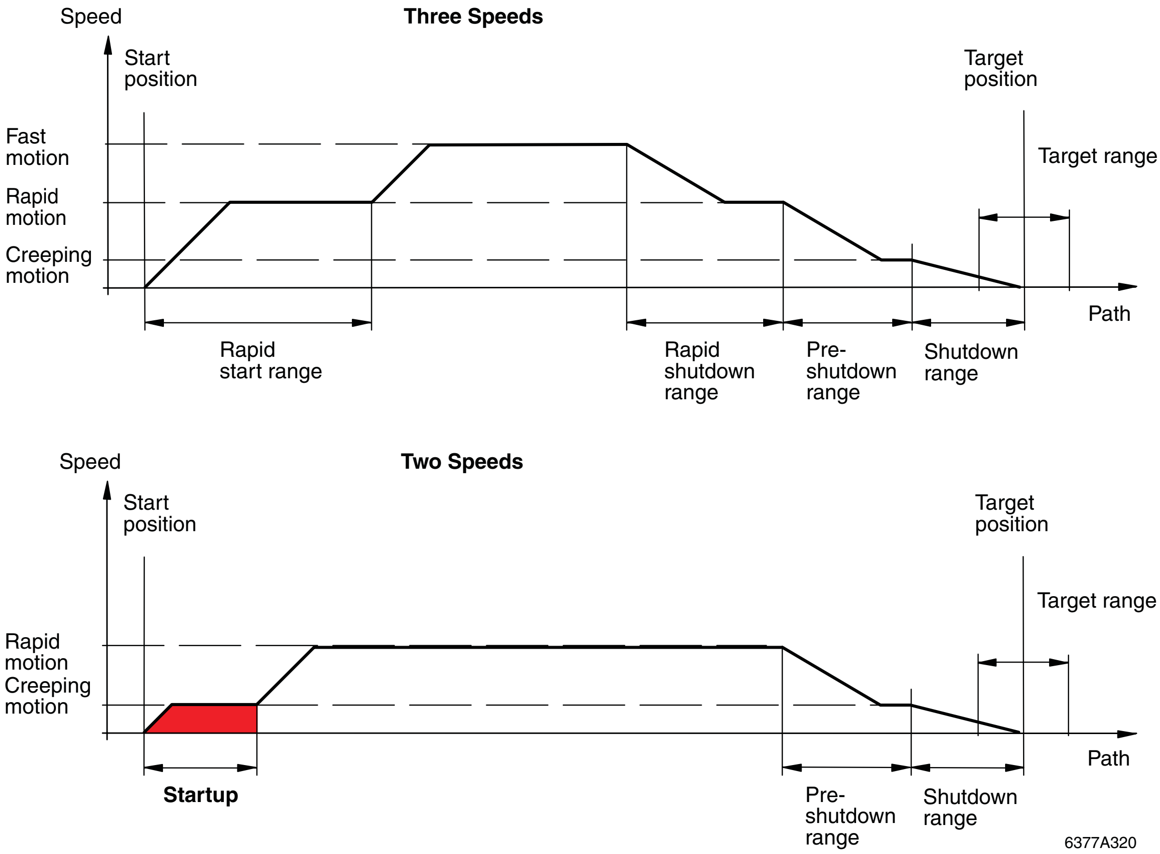


Figure 5-1 Start window

5.11.2 "Rapid Start Window" *(RapidStartWindow)* Command

This command is only needed if you have selected version 4 for "Function of switching outputs".

Define the rapid start window by using the OR function to link command C400 0000_{hex} for parameter record 1 or E400 0000_{hex} for parameter record 2 to the value for the rapid start window and sending this command to the terminal.

To read the value, you need to send command 4400 0000_{hex} for parameter record 1 (bit 15 = 0) or 6400 0000_{hex} (bit 15 = 0) for parameter record 2.

Output word 0 (OUT[0])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1/0	**1**	**PRx**	**0**	**0**	**1**	**RSW1/RSW2 (26-bit value; USIGN26)**									
C/E (Write)/4/6 (Read)				**4 or X**				**X**				**X**			

Output word 1 (OUT[1])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSW1/RSW2 (26-bit value; USIGN26)															
X				**X**				**X**				**X**			

PRx = 0: Parameterization for parameter record 1 (***P**arameter **r**ecord 1*)
PRx = 1: Parameterization for parameter record 2 (***P**arameter **r**ecord 2*)

Permissible range of values for *RapidStartWindow*:

Code (dec)	RSW1/RSW2: Rapid Start Window *(RapidStartWindow)*
0 to $2^{26} - 1$	**0 increments** to $2^{26} - 1$ increments

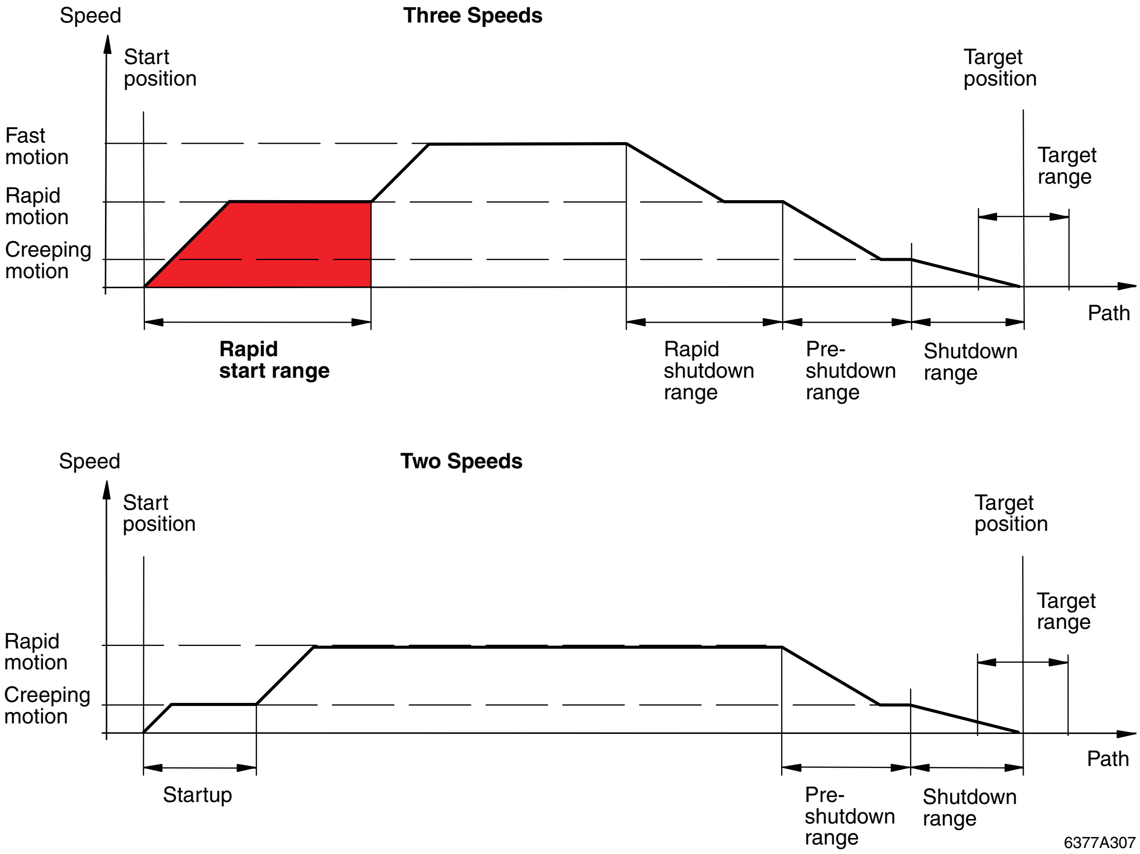


Figure 5-2 Rapid start window

5.11.3 "Rapid Stop Window" *(RapidStopWindow)* Command

This command is only needed if you have selected version 4 for "Function of switching outputs".

Define the rapid stop window by using the OR function to link command C800 0000_{hex} for parameter record 1 or E800 0000_{hex} for parameter record 2 to the value for the rapid stop window and sending this command to the terminal.

To read the value, you need to send command 4800 0000_{hex} for parameter record 1 (bit 15 = 0) or 6800 0000_{hex} (bit 15 = 0) for parameter record 2.

Output word 1 (OUT[0])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1/0	**1**	**PRx**	**0**	**1**	**0**	**RTW1/RTW2 (26-bit value; USIGN26)**									
C/E (Write)/4/6 (Read)				**8 or X**				**X**				**X**			

Output word 1 (OUT[1])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTW1/RTW2 (26-bit value; USIGN26)															
X				**X**				**X**				**X**			

PRx = 0: Parameterization for parameter record 1 (***P**arameter **r**ecord 1*)
PRx = 1: Parameterization for parameter record 2 (***P**arameter **r**ecord 2*)

Permissible range of values for *RapidStopWindow*:

Code (dec)	RTW1/RTW2: Rapid Stop Window *(RapidStopWindow)*
0 to $2^{26} - 1$	**0 increments** to $2^{26} - 1$ increments

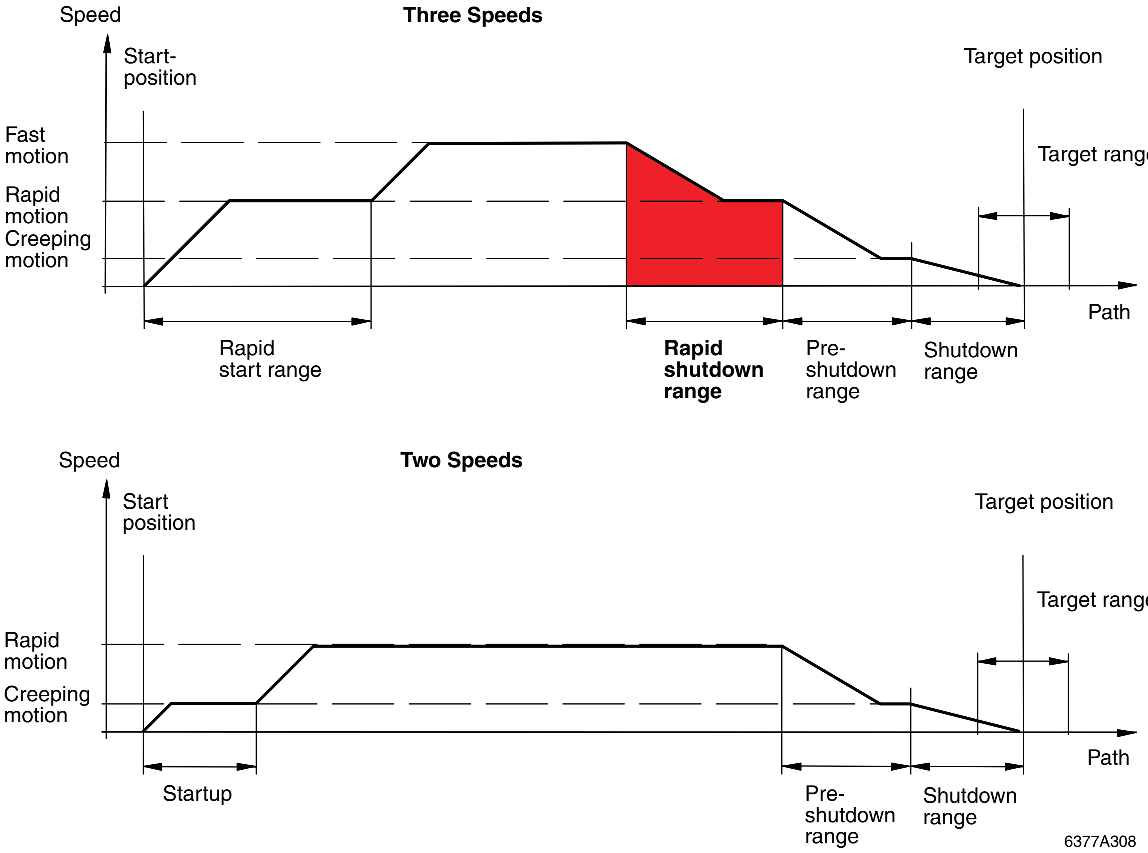


Figure 5-3 Rapid stop window

5.11.4 "Pre-Stop Window" *(Pre-StopWindow)* Command

Define the pre-stop window by using the OR function to link command CC00 0000_{hex} for parameter record 1 or EC00 0000_{hex} for parameter record 2 to the value for the pre-stop window and sending this command to the terminal.

To read the value, you need to send command 4C00 0000_{hex} for parameter record 1 (bit 15 = 0) or 6C00 0000_{hex} (bit 15 = 0) for parameter record 2.

Output word 0 (OUT[0])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1/0	**1**	**PRx**	**0**	**1**	**1**	**PSW1/PSW2 (26-bit value; USIGN26)**									
C/E (Write)/4/6 (Read)				**C or X**				**X**				**X**			

Output word 1 (OUT[1])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSW1/PSW2 (26-bit value; USIGN26)															
X				**X**				**X**				**X**			

PRx = 0: Parameterization for parameter record 1 (***P**arameter **r**ecord 1*)
PRx = 1: Parameterization for parameter record 2 (***P**arameter **r**ecord 2*)

Permissible range of values for *Pre-StopWindow*:

Code (dec)	PSW1/PSW2: Pre-Stop Window *(Pre-StopWindow)*
0 to $2^{26} - 1$	**0 increments** to $2^{26} - 1$ increments

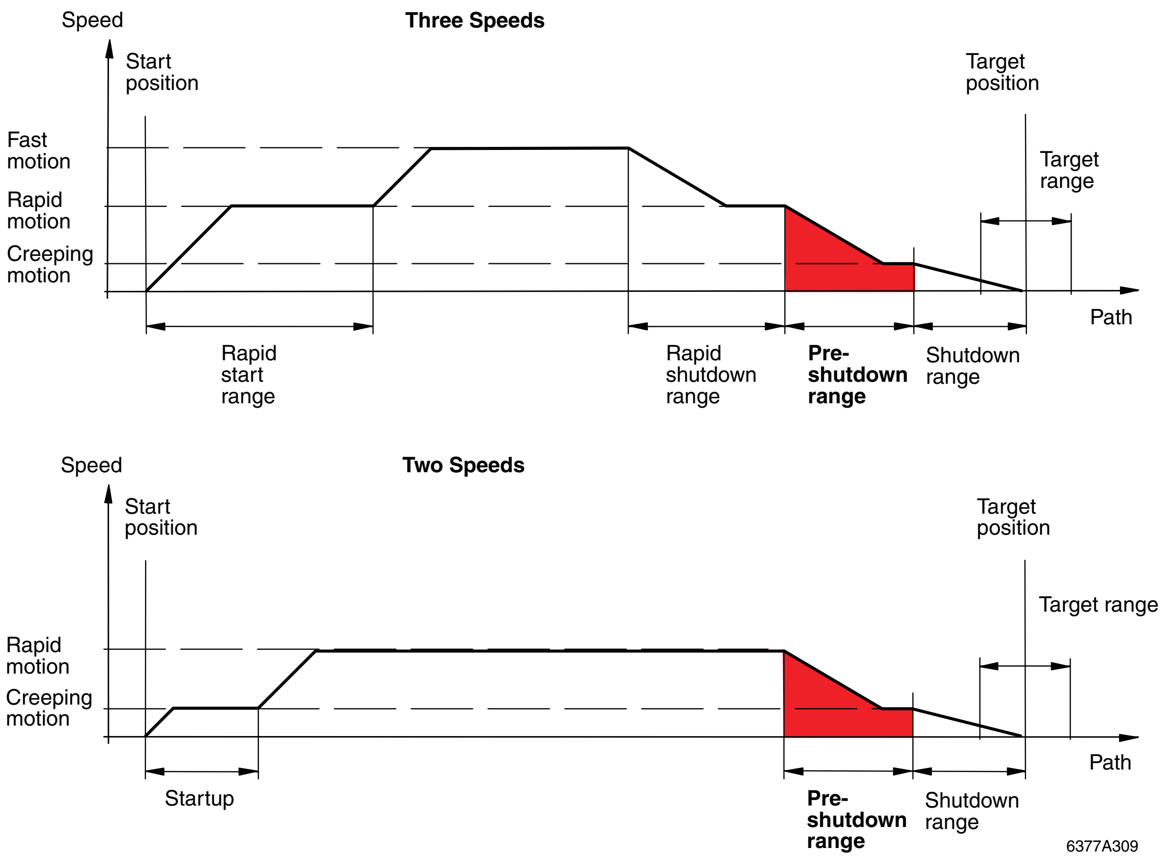


Figure 5-4 Pre-stop window

5.11.5 "Stop Window" *(StopWindow)* Command

Define the stop window by using the OR function to link command D000 0000_{hex} for parameter record 1 or F000 0000_{hex} for parameter record 2 to the value for the stop window and sending this command to the terminal.

To read the value, you need to send command 5000 0000_{hex} for parameter record 1 (bit 15 = 0) or 7000 0000_{hex} (bit 15 = 0) for parameter record 2.

Output word 1 (OUT[0])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1/0	**1**	**PRx**	**1**	**0**	**0**	**STW1/STW2 (26-bit value; USIGN26)**									
D/F (Write)/5/7 (Read)				**0 or X**				**X**				**X**			

Output word 1 (OUT[1])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STW1/STW2 (26-bit value; USIGN26)															
X				**X**				**X**				**X**			

PRx = 0: Parameterization for parameter record 1 (***P**arameter **r**ecord 1*)
PRx = 1: Parameterization for parameter record 2 (***P**arameter **r**ecord 2*)

To read the value, you need to send the corresponding command with bit 15 = 0.

Permissible range of values for *StopWindow*:

Code (dec)	STW1/STW2: Stop Window *(StopWindow)*
0 to $2^{26}-1$	**0 increments** to $2^{26}-1$ increments

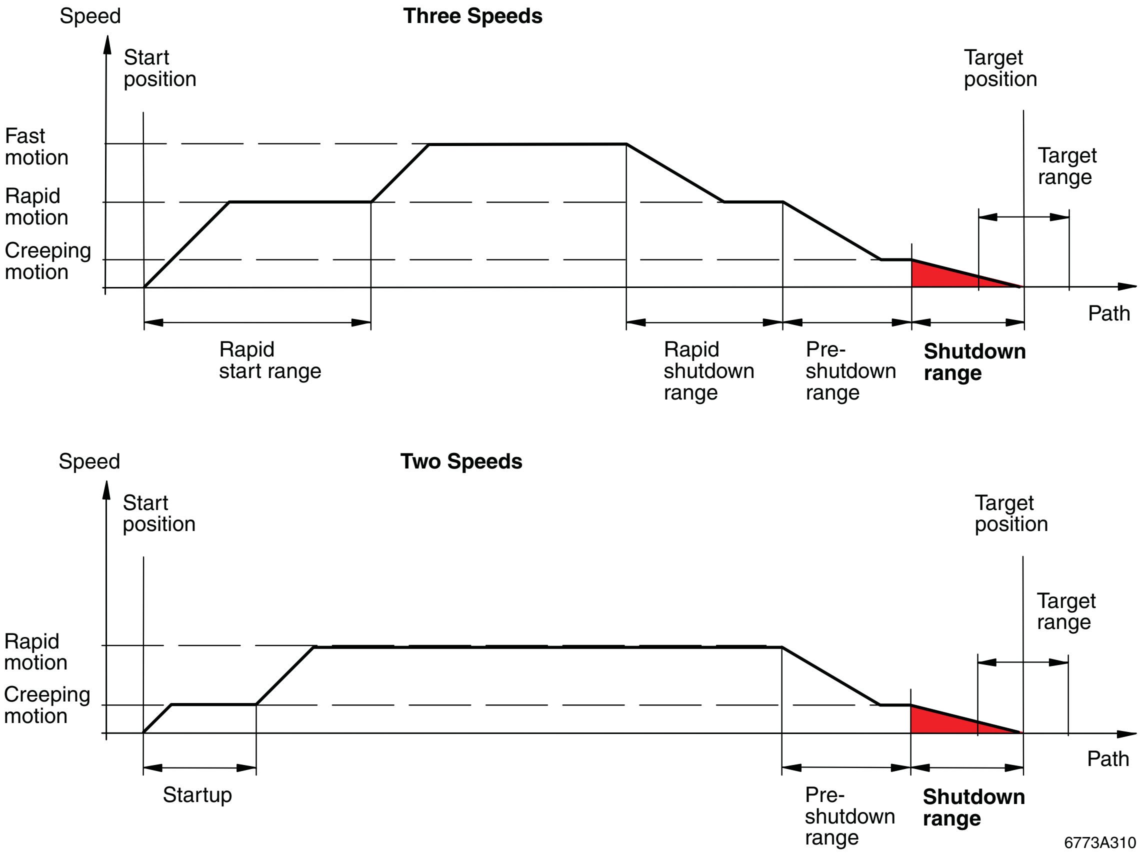


Figure 5-5 Stop window

5.11.6 "Target Window" *(TargetWindow)* Command

Define the target window by using the OR function to link command D400 0000_{hex} for parameter record 1 or F400 0000_{hex} for parameter record 2 to the value for the repeating counter and the value for the target window, and sending this command to the terminal.

To read the value, you need to send command 5400 0000_{hex} for parameter record 1 (bit 15 = 0) or 7400 0000_{hex} (bit 15 = 0) for parameter record 2.

If the target window is not reached, the terminal can automatically start a new process to approach the target position. In order that a new attempt can be made to approach the target position, looping must first be activated. The maximum number of repeat attempts is set by the repeating counter. Between 0 and 15 repeat attempts can be made.

Output word 0 (OUT[0])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1/0	**1**	**PRx**	**1**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**RCN1/RCN2**			
D/F (Write)/5/7 (Read)				**4 or X**				**0**				**X**			

Output word 1 (OUT[1])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TW1/TW2 (16-bit value; USIGN16)															
X				**X**				**X**				**X**			

PRx = 0: Parameterization for parameter record 1 (***P**arameter **r**ecord 1*)
PRx = 1: Parameterization for parameter record 2 (***P**arameter **r**ecord 2*)

Permissible ranges of values for *TargetWindow*:

Code (dec)	RCN1/RCN2: Target Repeating Counter *(RepeatingCounter)*
0 to 15	**0 repeats** to 15 repeats

Code (dec)	TW1/TW2: Target window *(TargetWindow)*
0 to $2^{16} - 1$	**0 increments** to $2^{16} - 1$ increments

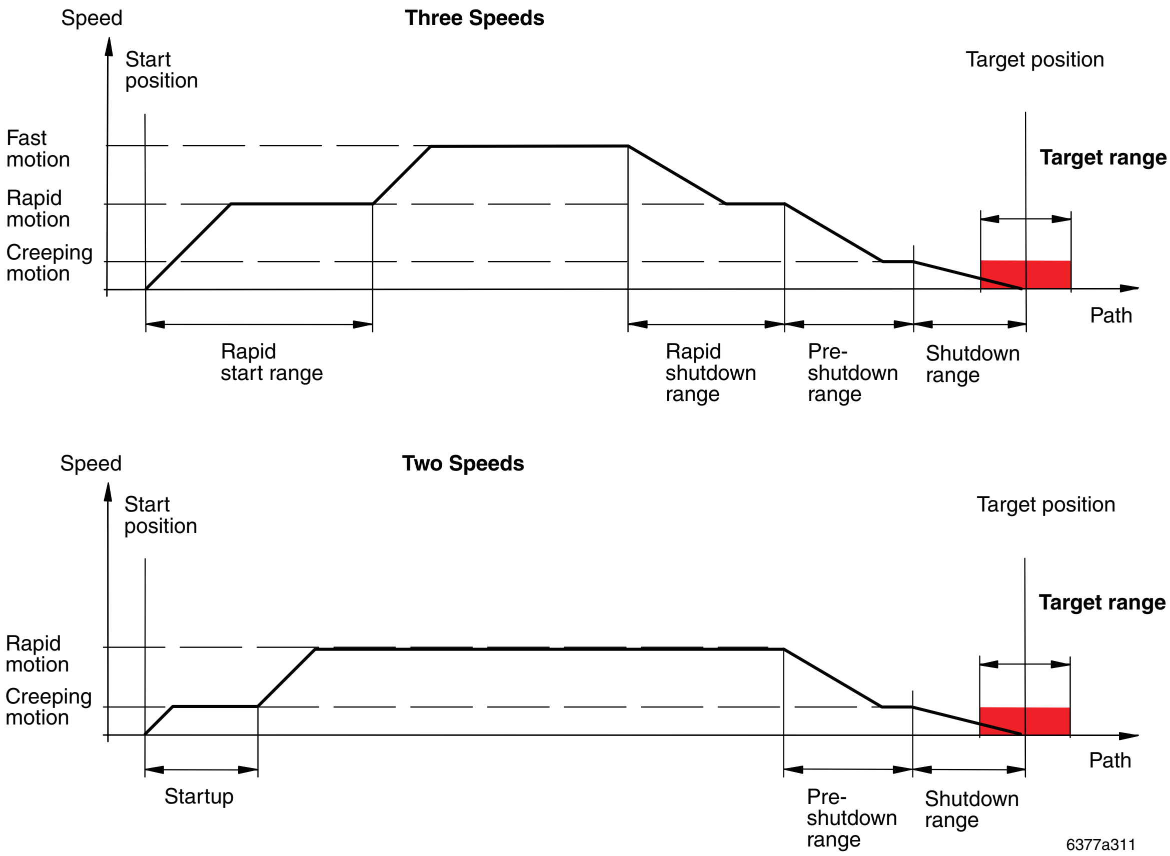


Figure 5-6 Target window

5.11.7 "Target position" *(TargetPosition)* Command

Define the target position by using the OR function to link command D800 0000_{hex} for parameter record 1 or F800 0000_{hex} for parameter record 2 to the value for the target position and sending this command to the terminal.

To read the value, you need to send command 5800 0000_{hex} for parameter record 1 (bit 15 = 0) or 7800 0000_{hex} (bit 15 = 0) for parameter record 2.

Output word 0 (OUT[0])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1/0	**1**	**PRx**	**1**	**1**	**0**	**TP1/TP2 (26-bit value; INT26)**									
D/F (Write)/5/7 (Read)				**8 or X**				**X**				**X**			

Output word 1 (OUT[1])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TP1/TP2 (26-bit value; INT26)															
X				**X**				**X**				**X**			

PRx = 0: Parameterization for parameter record 1 (***P**arameter **r**ecord 1*)
PRx = 1: Parameterization for parameter record 2 (***P**arameter **r**ecord 2*)

Permissible range of values for *TargetPosition*:

Code (dec)	TP1/TP2: Target position *(TargetPosition)*
-2^{25} to $+(2^{25}-1)$	-2^{25} increments to $+(2^{25}-1)$ increments
0	**0 increments**

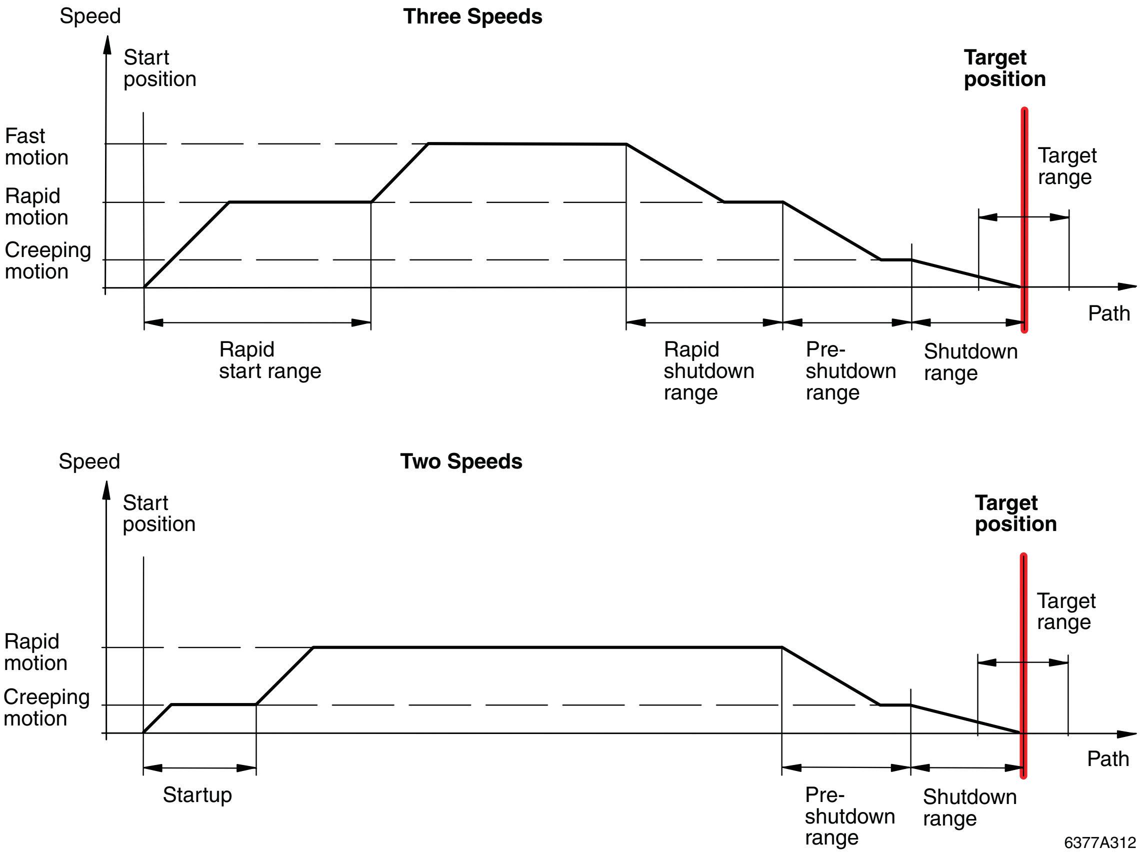


Figure 5-7 Target position

5.11.8 "Lubrication and Friction Compensation Value" *(DefineLubFricCompValue)* Command

Define the lubrication and friction compensation value by using the OR function to link command DC00 0000_{hex} for parameter record 1 or FC00 0000_{hex} for parameter record 2 to the compensation value and sending this command to the terminal.

To read the value, you need to send command 5C00 0000_{hex} for parameter record 1 (bit 15 = 0) or 7C00 0000_{hex} (bit 15 = 0) for parameter record 2.

Output word 0 (OUT[0])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1/0	**1**	**PRx**	**1**	**1**	**1**	**LFC1/LFC2 (26-bit-value)**									
D/F (Write)/5/7 (Read)				**C or X**				**X**				**X**			

Output word 1 (OUT[1])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LFC1/LFC2 (26-bit value)															
X				**X**				**X**				**X**			

PRx = 0: Parameterization for parameter record 1 (***P**arameter **r**ecord 1*)
PRx = 1: Parameterization for parameter record 2 (***P**arameter **r**ecord 2*)

Permissible range of values for *DefineLubFricCompValue*:

Code (dec)	LFC1/LFC2: Lubrication and Friction Compensation Value *(LubFricCompValue)*
-2^{25} to $+(2^{25}-1)$	-2^{25} increments to $+(2^{25}-1)$ increments
0	**0 increments**

5.12 "Jog Mode" *(DefineJogMode)* Command

Define jog mode by using the OR function to link command 9500 0000_{hex} to the monoflop time values and delay time and sending this command to the terminal.

To read the parameters, you need to send command 1500 0000_{hex} (bit 15 = 0).

Monoflop Time for Jog Mode (JogMonoflopTime)

You can use the monoflop time to set the time during which the drive should be set in motion for a jog pulse. The monoflop time can be set in millisecond increments; the setting range is between 0 and 65535 ms.

Delay Time for Continuous Signal in Jog Mode (JogDelayTimeCount)

If a control bit is "activated" for a prolonged period for jog mode via the "Delay time for continuous signal", once this time has elapsed, the position will be traversed in creeping motion (continuous signal) until the control bit is deactivated again.

The delay time for the continuous signal in jog mode is set in increments of 100 ms; the setting range is between 0 and 25.5 ms.

Output word 0 (OUT[0])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1/0	**0**	**0**	**1**	**0**	**1**	**0**	**1**	**JDTC (USIGN8)**							
9 (Write)/1 (Read)				**5**				**X**				**X**			

Output word 1 (OUT[1])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JMT (USIGN16)															
X				**X**				**X**				**X**			

Permissible ranges of values for *DefineJogMode*:

Code (dec)	JDTC: Delay Time for Continuous Signal in Jog Mode *(JogDelayTimeCount)*
0 to 255	**0 s** to 25.5 s

Code (dec)	JMT: Monoflop Time for Jog Mode *(JogMonoflopTime)*
0 to 65535	**0 ms** to 65535 ms

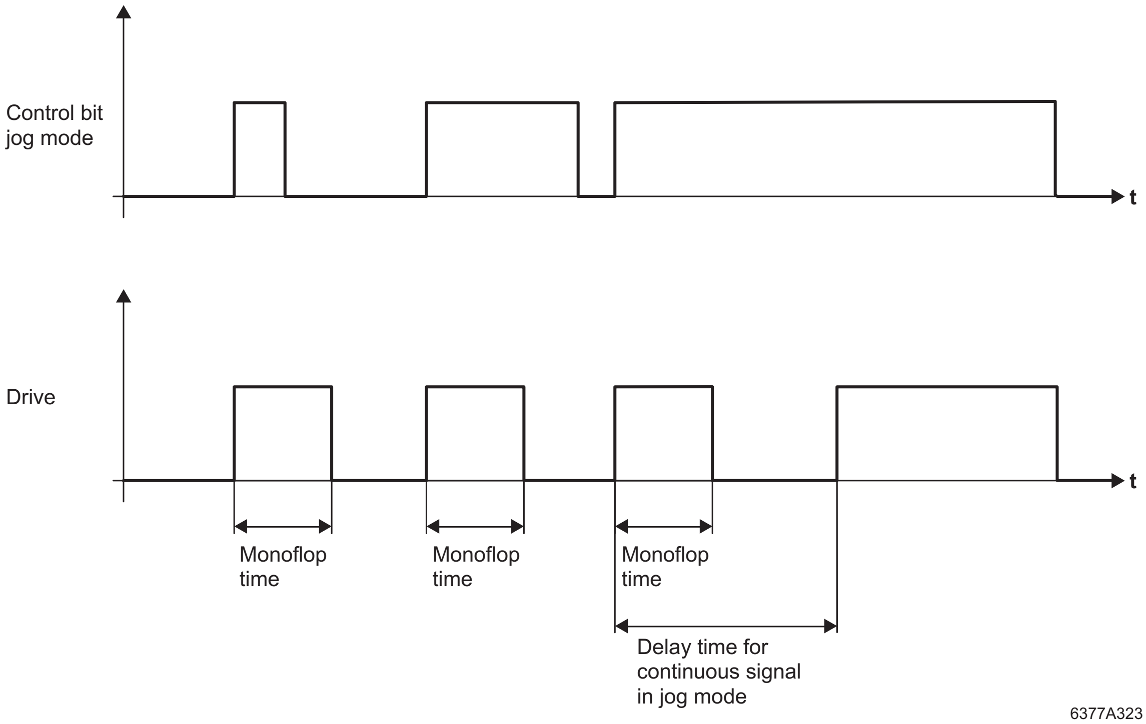


Figure 5-8 Times in jog mode

> If the default values (0) for the "Delay time for continuous signal in jog mode" and the "Monoflop time for jog mode" are set, the drive will follow the control bit directly.

5.13 Control Commands

These commands are used to control the positioning terminal and, simultaneously, to read specific information. Control is activated by setting bit 15 of the output word OUT[0] to 1. Set the bit to 0 to execute the read function only.

Control is via bits 7 to 0 of the output word OUT[0] and bits 15 to 0 of the output word OUT[1]. Depending on the command code used to transfer the control parameters, the following information can be queried simultaneously.

– Position of drive (page 5-45)
– Status of positioning terminal (seepage 5-47)

Please note that some control bits are level-driven and some are edge-driven.

Key for the following table:

•	Signal is level-driven (signal is evaluated when set to 1)
↑	Signal is edge-driven (signal is evaluated when bit changes from 0 to 1)

Table 5-8 Control bits for controlling the positioning terminal

Word	Bit	Active	Des.	Description	Description
OUT[0]	7	•	OUT4	Output 4 (***Out**put* **4**)	If output version 5 has been activated, otherwise not relevant
	6	•	OUT3	Output 3 (***Out**put* **3**)	
	5	•	OUT2	Output 2 (***Out**put* **2**)	
	4	•	OUT1	Output 1 (***Out**put* **1**)	
	3	0	0	Reserved	
	2	•	JOGN	Jog mode in negative direction (**JogN**egative: *Jogging in negative direction*)	The drive travels at creeping speed. The behavior in respect of the software limit switches is determined by the firmware version or the configuration (see "Software Limit Switches" on page 3-15).
	1	•	JOGP	Jog mode in positive direction (**JogP**ositive: *Jogging in positive direction*)	
	0	0	0	Reserved	

Table 5-8 Control bits for controlling the positioning terminal (Continued)

Word	Bit	Active	Des.	Description	Description
OUT[1]	15	–	0	Reserved	
	14	•	ALFC	Activate lubrication and friction compensation (**A**ct**L**ub**F**ric**C**omp: *Activate lubrication and friction compensation*)	
	13	–	0	Reserved	
	12	•	ELP	Enable looping (**E**nable**L**oo**p**ing: *Enable looping*)	
	11	•	DDD	Define drive direction of target position during active backlash compensation or looping (**D**ef**D**rv**D**ir: *Define drive direction of target position during active backlash compensation or looping*)	0: Positive direction; 1: Negative direction during active backlash compensation or looping
	10	•	ABC	Activate backlash compensation (**A**ct**B**acklash**C**omp: *Activate backlash compensation*)	The direction is set in bit 11
	9	–	0	Reserved	
	8	–	0	Reserved	

Table 5-8 Control bits for controlling the positioning terminal (Continued)

Word	Bit	Active	Des.	Description	Description
OUT[1]	7	–	0	Reserved	
	6	–	0	Reserved	
	5	•	ELC2	Enable local positioning control using digital inputs (initiators) with parameter record 2 (**E**nable**L**ocal**C**trl**2**: *Enable local positioning control using digital inputs (initiators) with parameter record 2*)	
	4	↑	SPD2	Start positioning with parameter record 2 (**S**tart**P**os**D**rv**2**: *Start positioning with parameter record 2*)	
	3	•	ELC1	Enable local positioning control using digital inputs (initiators) with parameter record 1 (**E**nable**L**ocal**C**trl**1**: *Enable local positioning control using digital inputs (initiators) with parameter record 1*)	
	2	↑	SPD1	Start positioning with parameter record 1 (**S**tart**P**os**D**rv**1**: *Start positioning with parameter record 1*)	
	1	↑	RES	Reset error (**Res**et: *Reset error*)	
	0	•	STOP	Stop positioning immediately. (**Stop**: *Stop positioning immediately*)	The command currently being executed is aborted. A new positioning process can only be started once the stop bit has been reset.

5.14 "Control Position and Read Position" *(ControlPosition, ReadPosition)* Command

You can control positioning and read the current position simultaneously by using the OR function to link command 8000 0000_{hex} to the control bit (Table 5-8 on page 5-42) and sending this command to the terminal.

To read the current position, you need to send command 0000 0000_{hex} (bit 15 = 0) without setting any parameters.

Output word 0 (OUT[0])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1/0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**OUT4**	**OUT3**	**OUT2**	**OUT1**	**0**	**JOGN**	**JOGP**	**0**
8 (Write)/0 (Read)				**0**				**0**				**0**			

Output word 1 (OUT[1])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	**ALFC**	**0**	**ELP**	**DDD**	**ABC**	**0**	**0**	**0**	**0**	**ELC2**	**SPD2**	**ELC1**	**SPD1**	**RES**	**STOP**
X				**X**				**X**				**X**			

The current position is transferred to the process input words. Representation is in two's complement format (as INT26).

If an error has occurred, this will be indicated in the status bit (***St**atus bit*; ST). The type of error that has occurred can be determined using the *Read status* command (see page 5-47).

Input word 0 (IN[0])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST	**0**	**0**	**0**	**0**	**0**	**POSI (26-bit value; INT26)**									

Input word 1 (IN[1])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POSI (26-bit value; INT26)															

Range of values for *ControlPosition, ReadPosition*:

Code (dec)	POSI: Value of current position *(CurrentPosition)*
-2^{25} to $+(2^{25} - 1)$	-2^{25} increments to $+(2^{25} - 1)$ increments

5.15 "Control Position and Read Velocity" *(ControlPosition, ReadVelocity)* Command

You can control the position and read the current velocity simultaneously by using the OR function to link command 8400 0000_{hex} to the control bit (Table 5-8 on page 5-42) and sending this command to the terminal.

To read the current position, you need to send command 0400 0000_{hex} (OUT[0] bit 15 = 0) without setting any parameters.

Output word 0 (OUT[0])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1/0	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**OUT4**	**OUT3**	**OUT2**	**OUT1**	**0**	**JOGN**	**JOGP**	**0**
8 (Write)/0 (Read)				**4**				**0**				**0**			

Output word 1 (OUT[1])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	**ALFC**	**0**	**ELP**	**DDD**	**ABC**	**0**	**0**	**0**	**0**	**ELC2**	**SPD2**	**ELC1**	**SPD1**	**RES**	**STOP**
X				**X**				**X**				**X**			

The current velocity (increments per time interval) is transferred to the process input words. Representation is in two's complement format (as INT26). The time interval is set using command *Define stop* (see page 5-16).

If an error has occurred, this will be indicated in the status bit (***St**atus bit*; ST). The type of error that has occurred can be determined using the *Read status* command (see page 5-47).

Input word 0 (IN[0])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST	**0**	**0**	**0**	**0**	**1**	**VELO (26-bit value; INT26)**									

Input word 1 (IN[1])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VELO (26-bit value; INT26)															

Range of values for *ControlPosition, ReadVelocity*:

Code (dec)	VELO: Value of current velocity *(CurrentVelocity)*
-2^{25} to $+(2^{25} - 1)$	-2^{25} increments/time interval to $+(2^{25} - 1)$ increments/ time interval

5.16 "Control Position and Read Status" *(ControlPosition, ReadStatus)* Command

You can control positioning and read the current status simultaneously by using the OR function to link command 8800 0000_{hex} to the control bit (Table 5-10 on page 5-49) and sending this command to the terminal.

To read the current status, you need to send command 0800 0000_{hex} (bit 15 = 0) without setting any parameters.

Output word 0 (OUT[0])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1/0	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**OUT4**	**OUT3**	**OUT2**	**OUT1**	**0**	**JOGN**	**JOGP**	**0**
8 (Write)/0 (Read)				**8**				**0**				**0**			

Output word 1 (OUT[1])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	**ALFC**	**0**	**ELP**	**DDD**	**ABC**	**0**	**0**	**0**	**0**	**ELC2**	**SPD2**	**ELC1**	**SPD1**	**RES**	**STOP**
X				**X**				**X**				**X**			

The current status is transferred to the process input words.

If an error has occurred, this will be indicated in the status bit (***St****atus bit*; ST). The type of error that has occurred can be determined using the *Read status* command (see page 5-47).

Input word 0 (IN[0])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**OUT4**	**OUT3**	**OUT2**	**OUT1**	**IN4**	**IN3**	**IN2**	**IN1**

Input word 1 (IN[1])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RNF	**0**	**0**	**ERRC**					**0**	**0**	**PPA2**	**PPE2**	**PPA1**	**PPE1**	**INCN**	**DS**

Table 5-9 Status bits

Word	Bit	Des.	Description
IN[0]	15	TS	*Status bit*
	...		
	7	OUT4	Status of output 4 *(Status of **out**put **4**)*
	6	OUT3	Status of output 3 *(Status of **out**put **3**)*
	5	OUT2	Status of output 2 *(Status of **out**put **2**)*
	4	OUT1	Status of output 1 *(Status of **out**put **1**)*
	3	IN4	Status of input 4 *(Status of **in**put **4**)*
	2	IN3	Status of input 3 *(Status of **in**put **3**)*
	1	IN2	Status of input 2 *(Status of **in**put **2**)*
	0	IN1	Status of input 1 *(Status of **in**put **1**)*
IN[1]	15	0	Reserved
	14	0	Reserved
	13	0	Reserved
	12	ER12	Error code n *(**Er**ror code **n**)* (see Table 5-10 on page 5-49)
	11	ER11	
	10	ER10	
	9	ER09	
	8	ER08	
	7	0	Reserved
	6	0	Reserved
	5	PPA2	Positioning with parameter record 2 is in progress *(**P**ositioning with **p**arameter record **2** **a**ctive)*
	4	PPE2	Positioning with parameter record 2 finished successfully *(**P**ositioning with **p**arameter record **2** finished successfully)*
	3	PPA1	Positioning with parameter record 1 is in progress *(**P**ositioning with **p**arameter record **1** **a**ctive)*
	2	PPE1	Positioning with parameter record 1 finished successfully *(**P**ositioning with **p**arameter record **1** finished successfully)*
	1	INCN	Terminal not yet completely initialized (encoder configuration, I/O configuration) *(Terminal **n**ot **c**ompletely **i**nitialized)*
	0	DS	Drive stop detected or drive stop monitoring switched off *(**D**rive **s**top detected or drive stop monitoring switched off)*

If a function is aborted with an error message, a new action can only be started once the error has been reset. The error must be reset with OUT[1] bit 1 (reset error, AERR) of the control command for positioning (see "Control Commands" on page 5-42).

Table 5-10 lists the meanings of the error codes and possible causes and provides tips for troubleshooting.

Table 5-10 Error codes

Error Code Bits 12 to 8		Meaning	Note/Cause and Tips for Troubleshooting
bin	dec		
0 0000	0	No errors have occurred	
0 0001	1	Terminal is in HHOP mode	Control via the bus is not possible in this mode.
0 0010	2	Output driver overload or short circuit	This error also triggers a module error message. • Remove the short circuit or overload.
0 0011	3	Encoder supply fault	This error also triggers a module error message. Cause: No encoder supply or encoder supply short circuit. • Connect the encoder supply or remove the short circuit.
0 0100	4	The encoder configuration is invalid.	Check the encoder configuration.
0 0101	5	A parity error has occurred.	Connection to the sensor is defective or sensor has been configured incorrectly. • Check connection and configuration.
0 0110	6	The initiator or switching output configuration is invalid.	• Check the configuration.
0 0111	7	Reserved	
0 1000	8	Invalid control command	• Check the control command.
0 1001	9	Position to be approached is out of the permissible range or is not defined.	• Check settings.
0 1010	10	Function cannot be executed, as software limit switches would be overrun.	• You may need to modify the software limit switches.
0 1011	11	Motor is not moving or is traveling in the wrong direction.	
0 1100	12	Reserved	
0 1101	13	The distance to the target position is shorter than the sum of the stop and start windows.	• Enable looping.
0 1110	14	Drive stop detected.	
0 1111	15	Software limit switches reached.	
1 0000	16	Hardware limit switches reached.	
1 0001	17	Target window could not be reached.	• Check the parameters of the stop and target windows.

Table 5-10 Error codes (Continued)

Error Code Bits 12 to 8		Meaning	Note/Cause and Tips for Troubleshooting
bin	dec		
1 0010	18	Counter overrun	The current position value is greater than the representation range. • Define a different gear ratio.
1 0011	19	Invalid action	An attempt has been made to start an invalid action. Example: A control command was still in progress. • Check action.
1 0100	20	Bus reset	
1 0101	21	Invalid system configuration.	• Check system configuration.

5.17 Overview of a Command Sequence

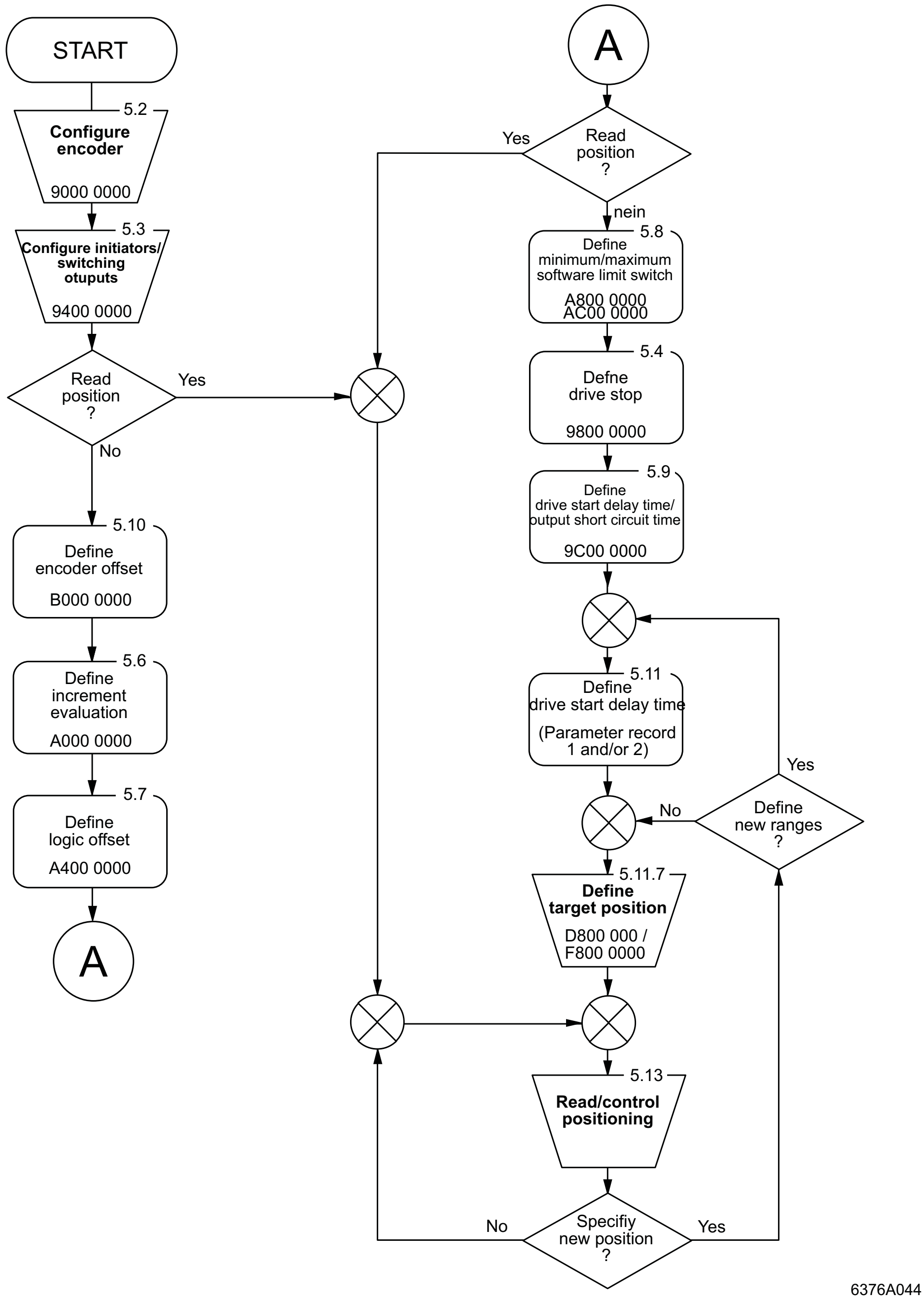
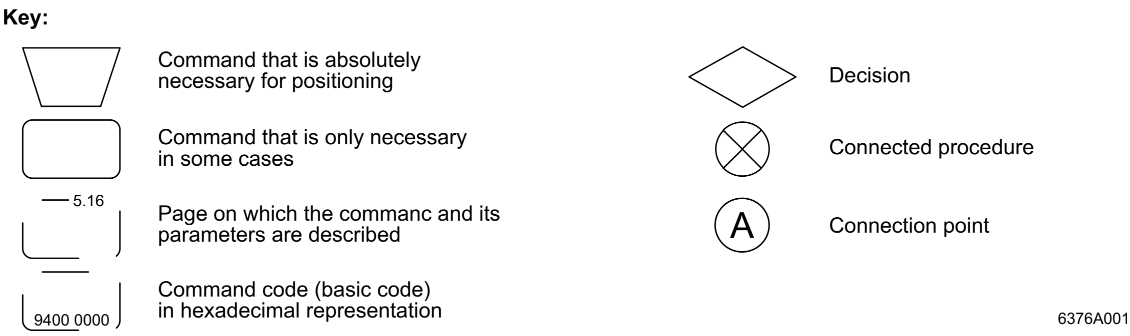


Figure 5-9 Process plan for parameterizing a positioning terminal



Define Parameter Record 1 and/or 2

	Parameter record 1/2	on page
Start window	C000 0000/E000 0000	5-25
Rapid start window	C400 0000/E400 0000	5-27
Rapid stop window	C800 0000/E800 0000	5-29
Pre-stop window	CC00 0000/EC00 0000	5-31
Stop window	D000 0000/F000 0000	5-33
Target window	D400 0000/F400 0000	5-35
Friction compensation value	DC00 0000/FC00 0000	5-39

Read/Control positioning

		on page
Control positioning/ Read position	8000 0000/ 0000 0000	5-45
Control positioning and read status/ Read status	8800 0000/ 0800 0000	5-47
Control positioning and read/ Read velocity	8400 0000/ 0400 0000	5-46

Control Bits for Control Commands (See Also "Control Commands" on page 5-42)

Output word 0 (OUT[0])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	0	0	0	X	X	0	0	OUT4	OUT3	OUT2	OUT1	0	JOGN	JOGP	0

Output word 1 (OUT[1])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	ALFC	0	ELP	DDD	ABC	0	0	0	0	ELC2	SPD2	ELC1	SPD1	RES	STOP

XXXXX One of the command codes for transmitting the control bits

Status Bits (See Also ""Control Position and Read Status" (ControlPosition, ReadStatus) Command" on page 5-47)

Input word 0 (IN[0])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST	0	0	0	1	0	0	0	OUT4	OUT3	OUT2	OUT1	IN4	IN3	IN2	IN1

Input word 1 (IN[1])

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	ERRC					0	0	PPA2	PPE2	PPA1	PPE1	INCN	DS

Some parameterization examples can be found in Section 6.2, "Examples".

6 Examples and Tips

Ensure data consistency

Always follow the notes on data consistency when programming.

6.1 Tips for Working With the Positioning Terminal

User manual

When configuring your system observe the information in the "Configuring and Installing the INTERBUS Inline Product Range" user manual or the system manual for the bus system you are using.

Ensure data consistency

Ensure a data consistency of two words (32 bits) to prevent the possibility of the values being misinterpreted.

Output word OUT[1] must be written first, followed by output word OUT[0], so that the terminal can ensure the required data consistency.

Sequence of the Inline terminals

The sequence of the terminals within an Inline station should depend on the current consumption of the I/O devices from the potential jumpers U_M and U_S.

As the voltage at every power terminal is fed back into the potential jumpers U_M and U_S, the section (main circuit) between bus coupler and power terminal or between power terminal and power terminal must always be considered when calculating the current. If power terminals are not used, the entire station is a main circuit.

Within a main circuit, place the I/O terminals with the highest current consumption first. This has the advantage that the high supply current does not flow through the entire main circuit.

This results in the following sequence:

1. Power-level terminals
2. Digital output terminals with 8-slot housing
3. Digital output terminals with 2-slot housing
4. Digital input terminals with 8-slot housing
5. Digital input terminals with 2-slot housing
6. Function modules in any order,
   including the **IB IL SSI ... positioning terminal**
7. Analog terminals in any order

The current consumption of the terminals is specified in the "Configuring and Installing the INTERBUS Inline Product Range" user manual IB IL SYS PRO UM E, in the system manual for the bus system you are using and in each terminal-specific data sheet.

6.2 Examples

6.2.1 Minimum Configuration for Reading a Position

The example shows the minimum commands that are required to read a position.

If you compare these steps with the process plan for parameterizing a positioning process in the foldout section of the front cover, these are the steps up to the first decision "Read position?" through to "Read/control positioning".

Table 6-1 Commands for reading a position

Action	Remark
OUT[0] = 0000_{hex}	Delete command code
OUT[0] = 9000_{hex} OUT[1] = 0100_{hex}	Command: *Define encoder* (page 5-8) – Parity: None – Mounting direction: Normal – Resolution: 8 bits – Code: Binary code
Wait until IN[0] = 1000_{hex} & IN[1] = 0100_{hex}	Wait for confirmation
OUT[0] = 9400hex OUT[1] = 2000hex	Command: *Define configuration of axis types, behavior in the event of a bus failure, initiators and switching outputs* (page 5-11) – Initiator 1: Limit switch – Initiator 2: Limit switch – Switching outputs: Version 2 Adjust the switching outputs to your application. – Bus reset: Stops a positioning process in progress – Axis type: Linear axis
Wait until IN[0] = 1400_{hex} & IN[1] = 2000_{hex}	Wait for confirmation
OUT[0] = 0000_{hex} OUT[1] = 0000_{hex}	Command: *Read position* (page 5-45)
IN[0] = $xxxx_{hex}$ IN[1] = $0xxx_{hex}$	The current position is specified in the input words.

6.2.2 Minimum Configuration for Reading a Position Using Gear Ratio

The example shows the minimum commands that are required to read a position, taking gear ratio into consideration.

If you compare these steps with the process plan for parameterizing a positioning process in the foldout section of the front cover, these are the steps up to the second decision "Read position?" through to "Read/control positioning".

Table 6-2 Commands for reading a position using gear ratio

Action	Remark
OUT[0] = 0000_{hex}	Delete command code
OUT[0] = 9000_{hex} OUT[1] = 0100_{hex}	Command: *Define encoder* (page 5-8) – Parity: None – Mounting direction: Normal – Resolution: 8 bits – Code: Binary code
Wait until IN[0] = 1000_{hex} & IN[1] = 0100_{hex}	Wait for confirmation
OUT[0] = 9400_{hex} OUT[1] = 2000_{hex}	Command: *Define configuration of axis types, behavior in the event of a bus failure, initiators and switching outputs* (page 5-11) – Initiator 1: Limit switch – Initiator 2: Limit switch – Switching outputs: Version 2 Adjust the switching outputs to your application. – Bus reset: Stops a positioning process in progress – Axis type: Linear axis
Wait until IN[0] = 1400_{hex} & IN[1] = 2000_{hex}	Wait for confirmation
	No rotary axis
	No encoder offset
OUT[0] = $A001_{hex}$ OUT[1] = $000A_{hex}$	Command: *Define gear ratio* (page 5-18) – Gear ratio 1/10
Wait until IN[0] = 2001_{hex} & IN[1] = $000A_{hex}$	Wait for confirmation
OUT[0] = $A400_{hex}$ OUT[1] = 0000_{hex}	Command: *Define logic offset* (page 5-19) – Set logic offset to 0

Table 6-2 Commands for reading a position using gear ratio (Continued)

Action	Remark
Wait until IN[0] = 2400_{hex} & IN[1] = 0000_{hex}	Wait for confirmation
OUT[0] = 0000_{hex} OUT[1] = 0000_{hex}	Command: *Read position* (page 5-45)
IN[0] = $xxxx_{hex}$ IN[1] = $0xxx_{hex}$	The current position is specified in the input words.

6.2.3 Minimum Configuration for Approaching a Position

The example shows the minimum commands that are required to approach a position.

The sequence of these commands is given in the process plan for parameterizing a positioning process in the foldout section of the front cover. Some of the commands, which are only necessary in some cases, are not listed in the example, since they are not required for this application. You must decide which parameterization is required for your application.

Table 6-3 Commands for approaching a position

Action	Remark
OUT[0] = 0000_{hex}	Delete command code
OUT[0] = 9000_{hex} OUT[1] = 0100_{hex}	Command: *Define encoder* (page 5-8) – Parity: None – Mounting direction: Normal – Resolution: 8 bits – Code: Binary code
Wait until IN[0] = 1000_{hex} & IN[1] = 0100_{hex}	Wait for confirmation
OUT[0] = 9400_{hex} OUT[1] = 2000_{hex}	Command: *Configure initiators and switching outputs* (page 5-11) – Initiator 1: Minimum limit switch – Initiator 2: Maximum limit switch – Switching outputs: Version 2 Adjust the switching outputs to your application. – Bus reset: Stops a positioning process in progress – Axis type: Linear axis
Wait until IN[0] = 1400_{hex} & IN[1] = 2000_{hex}	Wait for confirmation
OUT[0] = $D800_{hex}$ OUT[1] = $03E8_{hex}$	Command: *Define target position 1* (page 5-37) – Target position: 1000 ($3E8_{hex}$)
Wait until IN[0] = 5800_{hex} & IN[1] = $03E8_{hex}$	Wait for confirmation
OUT[0] = 8800_{hex} OUT[1] = 0004_{hex}	Command: *Read status and control positioning* (page 5-47) – Start positioning with parameter record 1. (approach position 1000) Please note that optimum positioning precision cannot yet be achieved because not all of the different ranges have been parameterized.
Wait until IN[0] = $08xx_{hex}$ & IN[1] = $xxx8_{hex}$ or $xxx4_{hex}$	Wait for confirmation

6.3 Using the Terminal for Position Detection

The positioning terminals can also be used to simply determine the position of a drive.

In this case, connect the encoder to the encoder interface. The position of the drive is determined using the input data words.

In this case, inputs 1 to 3 are available as digital inputs independent of the positioning terminal. The status of the inputs is determined using the input data words (IN[0] bits 2 to 0; see ""Control Position and Read Status" (ControlPosition, ReadStatus) Command" on page 5-47).

In addition, the outputs are freely available because they are not being used for positioning. In this case, select output version 5 (see "Function of the Switching Outputs (OutputFunction)" on page 5-14), and control the outputs independent of the positioning terminal using the output words (OUT[0] bits 7 to 4; see "Control Commands" on page 5-42).

6.4 Function Blocks and Documentation on the Internet

Function blocks for working with the positioning terminal are available on the Internet at www.download.phoenixcontact.com. Function blocks are available for various control systems.

The function blocks can be adapted to individual applications for parameterizing the positioning terminal.

Documentation for working with the function blocks is also available on the Internet.

7 Programming Data and Technical Data

This data is valid for the preferred mounting position (vertical).

The technical data does not claim to be complete. Technical modifications reserved.

The "Configuring and Installing the INTERBUS Inline Product Range" user manual IB IL SYS PRO UM E or the system manual for the bus system you are using contains additional technical data for the Inline product range.

7.1 Programming Data/Configuration Data

INTERBUS

ID code	BF_{hex} (191_{dec})
Length code	02_{hex}
Process data channel	32 bits
Input address area	4 bytes
Output address area	4 bytes
Parameter channel (PCP)	0 bytes
Register length (bus)	4 bytes

Other Bus Systems

Please refer to the relevant electronic data sheet (EDS), which is available at www.download.phoenixcontact.com, for configuration data for other bus systems.

Ensure data consistency

Ensure data consistency of two words to prevent the possibility of the values being misinterpreted.

7.2 Process Data Words

The "Configuring and Installing the INTERBUS Inline Product Range" user manual contains a description of INTERBUS software configuration.

The following tables can be used in association with the tables in the "INTERBUS Addressing" data sheet (DB D IBS SYS ADDRESS) to assign the positioning terminal's input and output words to the data words of your control or computer system.

Output Data Words for Configuring the Terminal
(See "Output Words" on page 4-2)

(Word.bit) view	Word	Word 0															
	Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
(Byte.bit) view	Byte	Byte 0								Byte 1							
	Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Word 0	Assignment	R/W	Command code					Parameter									

(Word.bit) view	Word	Word 1															
	Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
(Byte.bit) view	Byte	Byte 2								Byte 3							
	Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Word 1	Assignment	Parameter															

R/W: This bit indicates whether the specified parameter is to be read from the positioning terminal (R/W = 0) or written to the corresponding register on the terminal (R/W = 1).

Command code: The settings for these bits depend on the command to be transmitted.
Set the bits according to the application and the explanations in Section 5, "Commands for Working With the Positioning Terminal".

Parameter: The settings for these bits depend on the command to be transmitted.
Set the bits according to the application and the explanations in Section 5, "Commands for Working With the Positioning Terminal".

Input Data Words
(See "Input Words" on page 4-3)

Input Words During Parameterization

During parameterization the output words are mirrored in the input words.

(Word.bit) view	Word	Word 0															
	Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
(Byte.bit) view	Byte	Byte 0								Byte 1							
	Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Word 0	Assignment	ST	Mirroring of the command code					Result (parameter, position, status)									

(Word.bit) view	Word	Word 1															
	Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
(Byte.bit) view	Byte	Byte 2								Byte 3							
	Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Word 1	Assignment	Result (parameter, position, status)															

ST: Status bit

If bit 15 = 0, the command has been processed successfully.
If bit 15 = 1, an error occurred when the command was processed. The type of error that has occurred can be determined using the *Read status* command
(see ""Control Position and Read Status" (ControlPosition, ReadStatus) Command" on page 5-47).

7.3 Ordering Data

Products

Description	Type	Order No.	Pcs./Pck.
Positioning terminal for absolute encoder, with connector set and labeling fields; transmission speed 500 kbps	IB IL SSI-PAC	2861865	1
Positioning terminal for absolute encoder, without connector set and labeling fields; transmission speed 500 kbps	IB IL SSI	2836340	1
Positioning terminal for absolute encoder, without connector set and absolute encoder; transmission speed 2 Mbps	IB IL SSI-2MBD	2855729	1
Connector set for IB IL SSI;	IB IL SSI/INC-PLSET	2732473	1
Labeling field, can be snapped in, 8-slot terminal	IB IL FIELD 8	2727515	10

Accessories

Description	Type	Order No.	Pcs./Pck.
Keying profile	CP-MSTB See COMBICON catalog	1734634	100
Zack marker strip to label the terminal	ZBFM 6 ... see Phoenix Contact CLIPLINE catalog		
DIN EN 50022 DIN rail, 2 meters	NS 35/ 7.5 PERFORATED METERS NS 35/ 7.5 UNPERFORATED METERS	0801733 0801681	
Screwdriver according to DIN 5264, blade width 3.5 mm	SZF 1 - 0,6X3,5	1204517	1

Documentation

Description	Type	Order No.	Pcs./Pck.
User manual "Configuring and Installing the INTERBUS Inline Product Range"	IB IL SYS PRO UM E	2743048	1
User manual "Automation Terminals in the Inline Product Range"	IL SYS INST UM E	2698737	1
"General Introduction to the INTERBUS System" user manual	IBS SYS INTRO G4 UM E	2745211	1
"Configuring and Installing INTERBUS" user manual	IBS SYS PRO INST UM E	2743802	1
"INTERBUS Addressing" data sheet	DB GB IBS SYS ADDRESS	9000990	1

Make sure you always use the latest documentation. This can be downloaded from www.download.phoenixcontact.com.

7.4 Technical Data

General Data	
Housing dimensions (width x height x depth)	48.8 mm x 120 mm x 71.5 mm
Weight (without connectors)	130 g
Ambient temperature	
Ambient temperature (operation)	-25°C to +55°C
Ambient temperature (storage/transport)	-25°C to +85°C
Operating mode	Process data mode with 2 words
Connection method for sensors	2- and 3-wire technology
Connection method for actuators	2- and 3-wire technology
Connection method for all cables	Spring-cage terminals
Conductor cross section (typical)	0.2 mm² to 1.5 mm²

Ambient Conditions	
Regulations	Developed according to VDE 0160, UL 508
Humidity	
Humidity (operation/storage/transport)	10% to 95%, according to EN 61131-2
Air pressure	
Air pressure (operation/storage/transport)	70 kPa to 106 kPa (up to 3000 m above sea level)
Degree of protection according to DIN 40050, IEC 60529	IP20
Protection class according to DIN 57106-1	Class 3
Air and creepage distances	According to IEC 60644/ IEC 60664A/ DIN VDE 0110: 1989-01 and DIN VDE 0160: 1988-05
Housing material	Basic material: Arnite plastic PA6.6, self-extinguishing (V0)
Pollution degree according to EN 50178	2; condensation not permitted during operation
Surge voltage class	II (low-level signal) III (power level)
Gases that may endanger functions according to DIN 40046-36, DIN 40046-37	
Sulfur dioxide (SO_2)	Concentration 10 ± 0.3 ppm Ambient conditions - Temperature: 25°C (± 2°C) - Humidity: 75% (±5%) - Test duration: 10 days
Hydrogen sulfide (H_2S)	Concentration 1 ± 0.3 ppm Ambient conditions - Temperature: 25°C (±2°C) - Humidity: 75% (±5%) - Test duration: 4 days
Resistance of housing material to termites	Resistant
Resistance of housing material to fungal decay	Resistant

Power Consumption	500 kbps	2 Mbps
Communications power U_L	7.5 V	7.5 V
Current consumption at U_L	60 mA, maximum	85 mA, maximum
Power consumption at U_L	0.45 W, maximum	0.64 W, maximum
Segment supply voltage U_S	24 V DC (nominal value)	24 V DC (nominal value)
Nominal current consumption at U_S	2 A, maximum	2 A, maximum
Main supply voltage U_M	24 V DC (nominal value)	24 V DC (nominal value)
Nominal current consumption at U_M	1 A, maximum	1 A, maximum

Data Transfer	
Protocol	EN 50254; INTERBUS 2-wire protocol 500 kbps or 2 Mbps
Chip	Optical Protocol Chip
Transmission	Data jumper
Level	Logic level

Mechanical Requirements	
Vibration test Sinusoidal vibrations according to IEC 60068-2-6; EN 60068-2-6	2g load (low-level signal)
Shock test according to IEC 60068-2-27; EN 60068-2-27	25g load for 11 ms, half sinusoidal wave, three shocks in each space direction and orientation

Conformance With EMC Directive 89/336/EEC		
Noise Immunity Test According to EN 50082-2		
Electrostatic discharge (ESD)	EN 61000-4-2/ IEC 61000-4-2	Criterion B 6 kV contact discharge 6 kV air discharge (without labeling field) 8 kV air discharge (with labeling field in place)
Electromagnetic fields	EN 61000-4-3 IEC 61000-4-3	Criterion A Field strength: 10 V/m
Fast transients (burst)	EN 61000-4-4/ IEC 61000-4-4	Criterion B Supply lines: 2 kV I/O cables: 2 kV Criterion A All interfaces: 1 kV
Conducted interference	EN 61000-4-6 IEC 61000-4-6	Criterion A Test voltage 10 V
Noise Emission Test According to EN 50081-2		
Noise emission of housing	EN 55011	Class A

Interface	
Local bus	Through data routing

Supply of the Module Electronics and the I/O Through the Bus Coupler/Power Terminal (U_M, U_S, U_L)	
Connection method	Through potential routing

Absolute Encoder Inputs	
Number	1
Encoder signals	Clock, clock inverted, data, data inverted
Signal connection method	Shielded cables; unshielded cables may lead to erroneous results in environments prone to interference.

Encoder	
Type	Single-turn or multi-turn
Resolution	8 bits to 26 bits (can be parameterized)
Code type	Gray code, binary code
Parity monitoring	None, even, odd
Reversal of direction of rotation	Yes
Encoder supply	5 V (500 mA) or 24 V (500 mA)
Transmission frequency	400 kHz
Cable length	Less than 30 m for shielded cable

Encoder Supplies	
5 V Encoder Supply	
Voltage range	4.75 V to 5.25 V
Short-circuit protection	Electronic and thermal
Current carrying capacity	500 mA
24 V Encoder Supply	
Voltage range	19.2 V to 30.0 V
Short-circuit protection	Electronic and thermal
Current carrying capacity	500 mA

The state of the encoder supplies (5 V/24 V) is indicated via two LEDs. If the internal voltage for the encoder electronics fails, an I/O error is generated. This error is indicated by the diagnostic LED "D" flashing at 2 Hz and transmitted to the controller board.

Power Dissipation	
Power dissipation of the housing (PHOU)	2.7 W (within the permissible operating temperature)

Digital Inputs	
Number	4
Input design	According to EN 61131-2 Type 1
Signal range low	-30 V DC to +5 V DC (according to DIN 19240)
Signal range high	+13 V DC to +30 V DC (according to DIN 19240)
Common potentials	Main supply, ground
Nominal input voltage U_{IN}	24 V DC
Permissible range	-30 V DC < U_{IN} < +30 V DC
Nominal input current for U_{IN}	5 mA, typical
Connection method	2- and 3-wire technology

Digital Inputs (Continued)	
Current flow	Linear in the range 1 V < U_{IN} < 30 V
Delay time	< 1 ms
Permissible cable length to the sensor	30 m
Use of AC sensors	AC sensors in the voltage range < U_{IN} are limited in application

Input Characteristic Curve	
Input Voltage (V)	**Typical Input Current (mA)**
-30 < U_{IN} < 0.7	0
3	0.4
6	1.0
9	1.7
12	2.3
15	3.0
18	3.7
21	4.4
24	5.0
27	5.7
30	6.4

Digital Outputs	
Number	4
Connection method	2- and 3-wire technology
Nominal output voltage U_{OUT}	24 V DC
Differential voltage for I_{nom}	≤ 1 V
Nominal current per output I_{nom}	0,5 A
Tolerance of the nominal current	+10%
Total current of the outputs	2 A
Protection	Short-circuit; overload (thermal)
Nominal load	
Ohmic	48 Ω/12 W
Lamp	12 W
Inductive	12 VA (1.2 H, 50 Ω)
Signal delay upon power up of:	
- Ohmic nominal load	100 µs, typical
- Lamp nominal load	100 ms, typical (with switching frequencies up to 8 Hz; above this frequency the lamp load responds like an ohmic load)
- Inductive nominal load	100 ms (1.2 H, 50 Ω), typical
Signal delay upon power down of:	
- Ohmic nominal load	1 ms, typical
- Lamp nominal load	1 ms, typical
- Inductive nominal load	50 ms (1.2 H, 50 Ω), typical

Digital Outputs (Continued)

Switching frequency with:

- Ohmic nominal load	300 Hz, maximum

> This switching frequency is limited by the selected data rate, the number of bus devices, the bus structure, the software and the control or computer system used.

- Lamp nominal load	300 Hz, maximum

> This switching frequency is limited by the selected data rate, the number of bus devices, the bus structure, the software and the control or computer system used.

- Inductive nominal load	0.5 Hz, maximum at 500 mA (0.5 H, 48 Ω)
Overload response	Auto restart
Response with inductive overload	Output may be damaged
Response time in the event of a short circuit	400 ms, approximately
Reverse voltage protection against short pulses	Protected against reverse voltages
Resistance to polarity reversal of the supply voltage	Protective elements in the bus coupler or power terminal
Resistance to permanently applied surge voltage	No
Validity of output data after connecting the 24 V voltage supply (power up)	5 ms, typical
Response on power down	The output follows the supply voltage without delay.
One-time unsolicited energy	400 mJ, maximum
Protective circuit type	Integrated 38.6 V Zener diode in output
Overcurrent shutdown	0.7 A, minimum
Output current when switched off	100 μA, maximum
Output voltage when switched off	1 V, maximum

Output Characteristic Curve When Switched On (Typical)

Output Current (A)	Differential Output Voltage (V)
0	0
0.1	0.04
0.2	0.08
0.3	0.12
0.4	0.16
0.5	0.20

Limitation of Simultaneity, Derating

None

Safety Equipment

Surge voltage	Protective elements in the bus coupler or power terminal
Polarity reversal of the supply voltage	Protective elements in the bus coupler or power terminal the supply voltage must be protected. The power supply unit should be able to supply 4 times (400%) the nominal current of the fuse.
Short circuit protection for the outputs (segment circuit)	Short-circuit-proof (automatic restart)

Error Messages to the Higher-Level Control or Computer System

Short circuit/overload of an output	Yes

An error message is generated when an output is short-circuited and switched on. In addition, the diagnostic LED "D" flashes on the terminal at 2 Hz (medium).

Short circuit/Overload of the encoder supply	Yes

An error message is generated when an output is short-circuited and overloaded. In addition, the diagnostic LED "D" flashes on the terminal at 2 Hz (medium) under these conditions.

Failure of the main or segment voltage (U_M/U_S)	Yes
Failure of the internal voltage for the encoder electronics	Yes, I/O error to controller board. In addition, the diagnostic LED (D) flashes on the terminal at 2 Hz (medium) under these conditions.

Electrical Isolation/Isolation of the Voltage Areas

To provide electrical isolation between the logic level and the I/O area, power must be supplied to the station's bus coupler and the function terminal described in this document via the bus coupler or a power terminal from separate power supply units. Interconnection of the power supply units in the 24 V area is not permitted.

Common Potentials

The 24 V main voltage, 24 V segment voltage and GND have the same potential.
FE is a separate potential area.

Separate Potentials in the System Comprising Bus Coupler/Power Terminal and Function Terminal

- Test Distance	- Test Voltage
5 V supply incoming remote bus/7.5 V supply (bus logic)	500 V AC, 50 Hz, 1 min.
5 V supply outgoing remote bus/7.5 V supply (bus logic)	500 V AC, 50 Hz, 1 min.
7.5 V supply (bus logic)/24 V supply (I/O)	500 V AC, 50 Hz, 1 min.
7.5 V supply (bus logic)/functional earth ground of the encoder supply	500 V AC, 50 Hz, 1 min.
24 V supply (I/O)/functional earth ground	500 V AC, 50 Hz, 1 min.
24 V supply (I/O)/functional earth ground of the encoder supply	500 V AC, 50 Hz, 1 min.
Functional earth ground of the encoder supply/functional earth ground	500 V AC, 50 Hz, 1 min.

Approvals

You can download the latest approvals from www.download.phoenixcontact.com.

A Appendix

A 1 List of Figures

Section 1

Section 2

Section 3

Section 4

Section 5

A 2 List of Tables

Section 5

Section 6

A 3 Index

Numerics

R

S

T

U

V

W