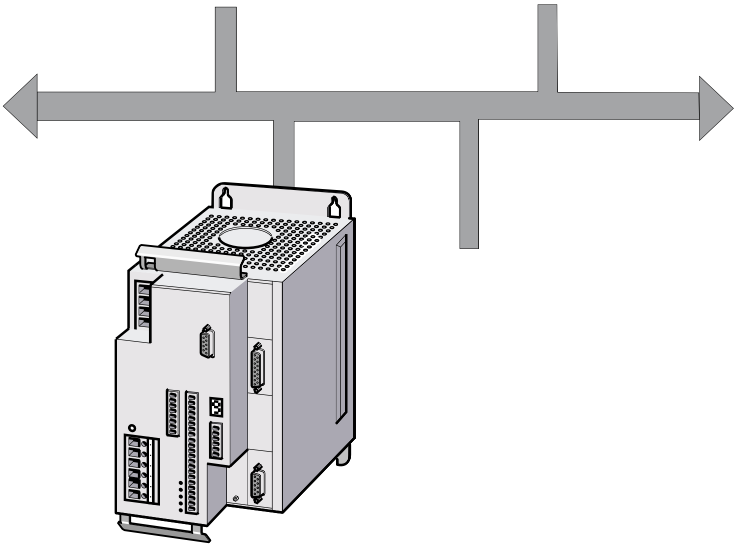
BERGER LAHR

Technical documentation



Online command processing for Twin Line units via field bus

Profibus-DP

Order no.: 9844 1113 125

Edition: a121, 07.02

Twin Line

Table of contents

5 Installation and setup

6 Examples for field bus operation

7 Error handling

8 **Service**

9 **Accessories**

Glossaries

Abbreviations

Abbreviation	Meaning
AC	Alternating current
ASCII	American Standard Code for Information Interchange
COS	Controller Operating System
DC	Direct current
DP	Decentralized Periphery
EC	European Community
EMC	Electromagnetic compatibility
GSD	File with device master data
HMI	Human-Machine Interface, plug-in hand-held operating unit
I	Incremental encoder
Inc	Increment
I/O	Inputs/Outputs
LED	Light Emitting Diode
LWL	Optic fiber
M	Motor
PC	Personal Computer
PLC	Programmable Logic Controller
ZK	DC link

Product name

Abbreviation	Product designation	Term used
TLC5xx	Twin Line Controller 5xx	Positioning controller
TL HMI	Twin Line HMI	HMI hand-held operating unit
TL CT	Twin Line Control Tool	Operating Software
TL HBC	Twin Line Holding Brake Controller	Holding Brake Controller
TL BRC	Twin Line Ballast Resistor Controller	Ballast Resistor Controller

Technical terms

Actual position of the drive system The actual position of the drive system gives the absolute or relative positions of moving components in the system.

Actual position of the motor See Angular position of the motor

Low/open Signal status of an input or output signal; when no signal is present, signal voltage is low (low level).

Asynchronous error Error detected and reported by the internal controller monitoring equipment.

Calibration factors Calibration factors are used to modify the precision of a user-defined unit referring to one motor revolution.

Control dynamics Speed at which a controller reacts to a disturbance or a change in the input signal.

DC link The DC link generates the necessary direct current for operating the motor and provides the amplifier with the necessary energy. The DC link acts as a buffer for energy fed back by the motor.

Default values Preset values for the parameters of the Twin Line unit before initial commissioning, factory settings.

Direction of rotation Rotation of the motor shaft in a clockwise or anticlockwise direction. A clockwise direction of rotation is defined as the motor shaft rotating clockwise as the observer faces the end of the protruding shaft.

Drive solution The drive solution comprises the drive system with the Twin Line Unit and motor with the system mechanics forming an integral part of the chain of motion.

Drive system The drive system consists of the Twin Line Unit and the motor.

Electronic gear An input speed is recalculated by the Twin Line Unit using the values of an adjustable gear ratio to derive a new output speed for the motor movement.

Encoder Sensor for recording the angular position of a rotating element. The encoder is mounted on the motor and signals the angular position of the rotor.

Fault class Response of the Twin Line Unit to an operational malfunction corresponding to one of five fault classes.

Forcing To change signal states regardless of the hardware switching status in the unit; with the control tool, for example. The hardware signals remain unchanged.

HIFA-C Module with Hiperface interface for connecting a Stegmann incremental encoder.

High/open Signal status of an input or output signal; when no signal is present, signal voltage is high (high level).

HMI Hand-held operating unit which can be plugged into the Twin Line Unit. HMI: Human Machine Interface. Human Machine Interface

Incremental encoder See encoder

Incremental signals Angular steps of an encoder in the form of square-wave pulse sequences. The pulses signal changes in position.

Index pulse Encoder signal for referencing the rotor position in the motor. The encoder sends one index pulse per revolution.

Input device Input device is the device which can be connected to the RS232 interface for the purpose of commissioning; it is either the HMI hand-held operating unit or a PC with the operating software.

Internal units Resolution of the power amplifier with which the motor is directed to the new setpoint. Internal units are given in increments.

I^2t monitoring Predictive temperature monitoring. The expected temperature rise of unit components is calculated in advance on the basis of the motor current Should a limit value be exceeded, the Twin Line Unit reduces the unit current.

Limit switch Switches that signal an overrun of the permissible travel range.

Node guarding Monitoring function at the RS232 interface or the field bus interface

Optically isolated Electrical transmission of signals with electrical isolation

Parameter Device data and values that can be set by the user.

PBDP-C Field bus module with which the positioning controller can be integrated into a Profibus-DP network.

Power amplifier This is the unit that controls the motor. The power amplifier generates currents for controlling the motor in accordance with the positioning signals from the control unit.

Pulse direction signals Digital signals with variable pulse frequencies which signal changes in position and rotation direction via separate signal wires.

Quick-Stop This function is used in the event of faults, the stop command or for fast braking of the motor in an emergency.

Resolver Analog encoder for determining the angular position of the rotor. It is used for returning the actual position of the motor for phase-accurate control of the motor.

RS232 interface Communications interface of the Twin Line Unit for the connection of a PC or the HMI hand-held operating unit.

RS422 level The signal status is calculated from the differential voltage of one positive and one inverted negative signal. Two signal wires must therefore be connected for one signal.

RS485 level The signal status is calculated from the differential voltage of one positive and one inverted negative signal. Two signal wires must therefore be connected for one signal.

Sincoder An encoder for registering the position of the servomotor rotor as an analog sine-cosine signal and as digital position data over the HIFA-C module. Motor data are held in the Sincoder and are read into the unit once the Twin Line Unit is switched on.

Synchronous error Error reported by the controller when it is unable to execute a command sent by the master.

User-defined unit A user-defined unit corresponds to the maximum precision at which a distance, speed or acceleration value can be input.

Watchdog A facility in the device for detecting internal system errors. When an error is detected, the power amplifier switches off immediately.

Written conventions and note symbols

Action symbols "▶" This action symbol is used for step-by-step instructions which can be carried out as they are described. If one of the instructions results in a noticeable response from the unit, this will be described after the description of the action to be carried out. This will give you direct confirmation that a particular step has been correctly carried out.

Enumeration symbol "•" The enumeration symbol is used for listing individual points in a given information group in summary form. If the result of steps or sequences is described, the step to be carried out is described first.

Menu paths "→" In the Twin Line Control Tool operating software, actions can be initiated via "Menu → menu item →...". For example, selecting "File → Save" in the "File" menu under menu item "Save" will save data from a PC'S memory to a data carrier.

Parameter names Parameters are written in this manual as follows:

Parameter group.parameter name (Index:Subindex)

Example: "CtrlBlock1.n_max (19:5)" for the maximum speed parameter in the group CtrlBlock1.

For additional information on parameters see the parameter overview in the controller manual. They are listed there sorted by groups.

ℹ *This symbol is used for general notes which give additional information about the unit.*

☎ *Passages preceded by this symbol may have to be discussed in more detail with the local customer service.*

1 Profibus technology

1.1 Profibus transmission technology

The Profibus is a serial field bus system which enables devices from different manufacturers to be networked without the need for special interface adaptation. Profibus is available in three versions, with which time-critical, complex communications tasks can be performed:

- Profibus FMS
- Profibus-PA
- Profibus-DP

Profibus-FMS (FMS: Fieldbus Message Specification) is a universal, flexible solution for communications tasks in general automation technology, and is used, for example, for communication between manufacturing units.

Profibus-PA (PA: Process Automation) is used mainly in process technology, e.g. in process automation. Profibus-PA networks are characterized by their ability to use sensors and actuators in explosion-endangered areas, and to provide data communication and power to devices over the bus.

Profibus-DP (DP: Distributed Periphery) is the fast Profibus version. It is specially tailored for communications in manufacturing areas. It is characterized by simple integration of new devices into the bus, low connection costs and high transmission speeds.

The controllers from your supplier are equipped for operation in Profibus-DP-networks.

1.2 Network topology

A Profibus-DP network consists of one or more master devices and slave devices. All devices are connected to one another via the Profibus-DP network cable.

Master Masters are active bus devices that control the data traffic in the network. The following are examples of master devices:

- automation devices, e. g. PLCs
- PCs
- programming devices

Slave Slaves are passive bus devices. They receive control commands and supply data to the master. Examples of slave devices are:

- input/output modules
- drive controllers, e.g. Twin Line Units
- sensors and actuators

1.3 Accessing procedures

There are two possible accessing procedures resulting from the arrangement of network devices on the bus:

- the Token-Passing method
- the Master-Slave method

The Token-Passing method

The Token-Passing method is used between several master devices in a Profibus-DP network. The master devices form a logical token ring in which each master in succession receives transmission rights for a certain period.

Drive control units such as Twin Line units are integrated into the network as slaves, and therefore do not operate by the token-passing method.

The Master-Slave method

Data are exchanged with drive controllers with the master-slave method. The slave device has a transmission and reception buffer which is used to prepare and receive data. The master device reserves an equal amount of storage space with a transmission and reception buffer for each slave device.

Data exchange between master and slave is cyclical. The master device sends commands to the slave device and receives data back from the slave in the next cycle.

The transmission and reception buffers for data exchange in the controller from your supplier is 8 bytes in size each.

1.4 Transmission technology in Profibus-DP network

Profibus-DP networks can be set up with fiber-optic cables or with RS485 technology.

Fiber-optic technology

Transmission via fiber-optic cable is used mainly in environments subject to strong electromagnetic interference and for heavy data transmission over wide areas.

RS485 technology

RS485 twisted-pair transmission is a simple method. Transmission rates between 9.6 kbits/s and 12 Mbits/s are possible with this method.

Twin Line units work with RS485 technology and are connected to a Profibus-DP via twisted-pair cables. Intermediate terminal boxes can be used as an interface with fiber-optic cables.

1.5 Device identification

Device master data file

The specific characteristics of a Profibus device type are described in the device master data file (GSD file). This file is supplied with the device by the manufacturer, and must be read by the network configuration program.

The GSD file contains all the information required for operating the device in the Profibus-DP network, such as details of the manufacturer, device name, baud rates supported, signal assignment on plugs, time intervals for monitoring cycles and device-specific values for other network devices such as input/output settings.

The GSD file for a Twin Line unit in a Profibus-DP network is supplied on a diskette.

Identity number

A master device uses the identity number to identify the device class of the connected slave device. The Profibus user organization allocates a unique identity number for specific device classes; for TLC devices it is 0482h.

2 The field bus device

2.1 System requirements

The Twin Line Unit must be fitted with the PBDP-C field bus interface for operation in the field bus.

Item	Qty.	Designation	Order no.
1	1	Twin Line Unit with module PBDP-C	Type code
2	1	Diskette with the master device data (GSD file)	
3	1	Manual for Profibus-DP	9844 1113 125

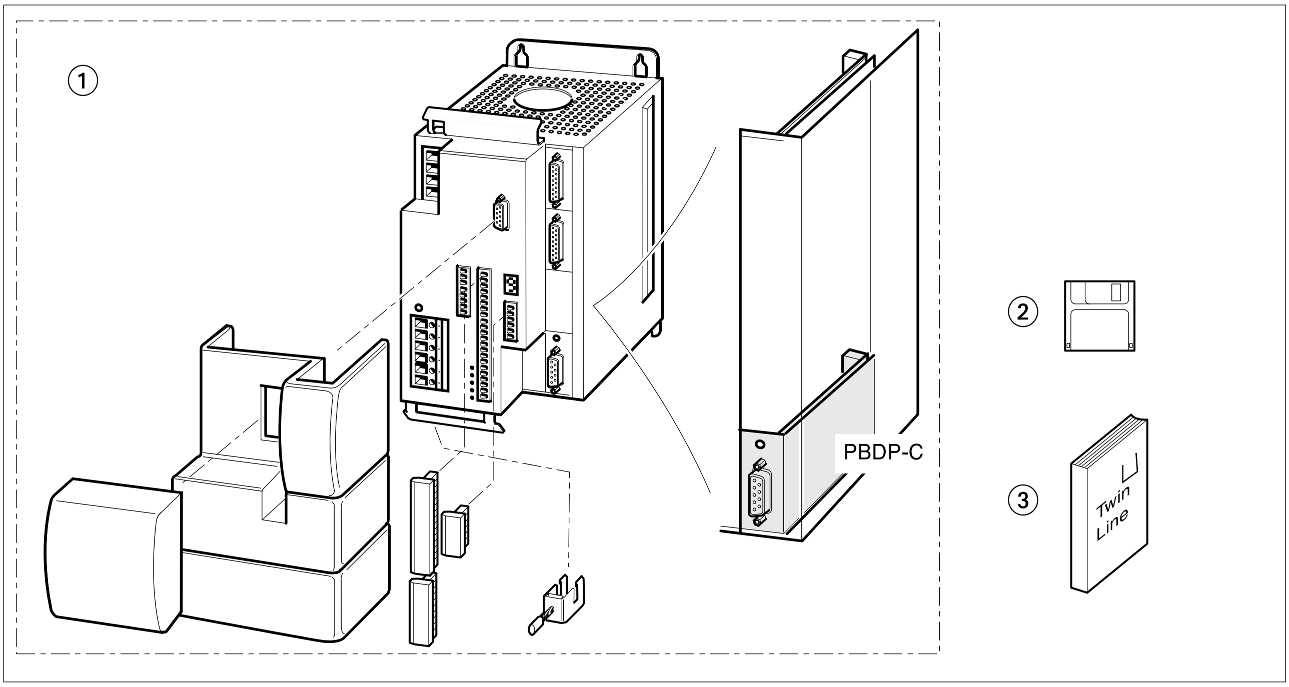


Fig. 2.1 Twin Line Unit with field bus fittings

2.2 Field bus devices in Profibus-DP network

The manufacturer's field bus devices can be operated in the same field bus. The commands for Twin Line Units differ however from those for other devices from the manufacturer in the field bus.

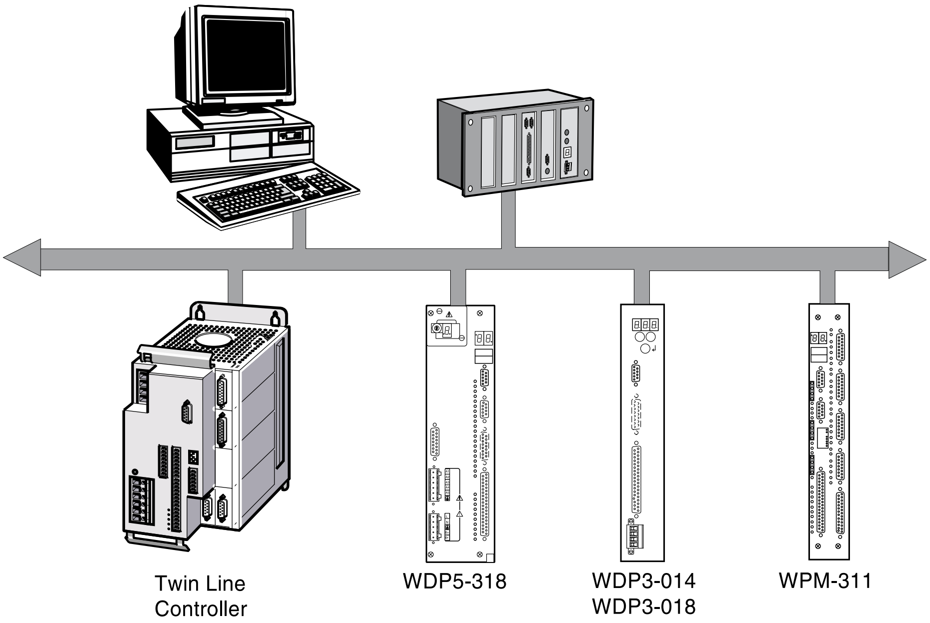


Fig. 2.2 Field bus devices in the network

2.3 Operating modes and functions in field bus operation

Depending on their equipment and model, Twin Line Units operate with the following operating modes and functions in a field bus:

- Relative and absolute positioning,
- Speed mode
- Electronic gear
- Reference movement and dimension setting
- Manual movement
- Data set mode
- Oscillator mode
- Current regulation

Operating functions include

- List control
- Teach-in
- Calibration
- Ramp functions
- Braking function
- Quick-Stop function
- Reversal
- Standstill window
- Fast position capture
- Monitoring functions.

The parameter settings for Twin Line units can be opened and modified, inputs can be monitored, outputs controlled and diagnostic and troubleshooting monitoring functions can be activated over the field bus.

2.4 Twin Line manuals and literature references

Twin Line manuals Twin Line Controller 53x, TLC53x controller manual for positioning controllers

Twin Line HMI manual for HMI handheld unit

TL CT operating software manual for Twin Line Control Tool software

Literature references Field bus manual, Berger Lahr controllers (German):
Online-Befehlsbearbeitung über Profibus-DP für WDP3-014 und WDP3-018 und Geräte Serie 300
(Online command processing via Profibus-DP for WDP3-014 and WDP3-018 and Series 300 devices),
DS no. 222.267/DD 04.97

Addresses Profibus international in the Internet: http://www.profibus.com

2.5 Regulations, standards

DIN 19245, Parts 1 to 3: PROFIBUS-FMS

EN 50170, field bus standard

3 Safety

3.1 Danger categories

Safety notes and general information are indicated by special symbols in the manual. In addition you will find symbols and instructions affixed to your Twin Line Unit which warn you of possible dangers and help you to operate the unit safely.

Danger symbols are divided into three danger categories classified by how serious a danger is.. The symbols shown emphasize the danger situation to which the warning applies.

DANGER!
This indicates direct personal danger.
Can lead to serious injuries with fatal consequences if not observed.

WARNING!
Indication of a recognizable danger.
Can result in serious injuries with fatal consequences and destruction of the unit or system component if not observed.

CAUTION!
Indication of a danger.
If this is ignored, minor personal injury and light damage to the unit or system may be the result.

3.2 Safety notes

DANGER!
Electric shock from high voltage!
Follow safety rules when working on electrical systems:

- Switch off the power to the unit.
- Make sure the unit cannot be switched on again inadvertently.
- Confirm that no voltage is present.
- Cover or shield any neighboring system parts which are live.

◉ > 6 min

DANGER!
Electric shock from high voltage!
Before starting work on the connections of the power unit or on the motor terminals, wait for the 4 minutes discharge time and then measure the residual voltage on the DC link terminals DC+ and DC-. The residual voltage must not exceed 48 V_{DC} before you start work on the connections. If additional DC-link capacitors are connected, the discharge period is increased to 10 minutes. Wait for this period and then measure the residual voltage.

DANGER!
Danger of personal injury and damage to system parts by uncontrolled movement of the controller!
When starting field bus operation the connected controllers are generally out of the operator's sight and cannot be directly observed. Only start field bus operation when no persons are in the action radius of moving system components and the system can be operated safely.

DANGER!
Danger of personal injury and damage to system parts resulting from processing of faulty control commands!
If a PLC is used as the master device, the exchange of data can lead to inconsistent transmission data as a result of field bus and PLC cycles not operating synchronously.
You will find details on the use of PLCs in a field bus in the chapter on "PLC as a field bus master device" on page 4-15.

3.3 Intended use

Twin Line units with the PBDP-C module may be used in Profibus-DP networks with wiring conforming to the RS485 standard.

The Twin Line units must be correctly installed before network operation and their functions must be tested in an initial commissioning test.

The network wiring must be installed to EMC specifications before a device may be started in the network.

3.4 Qualification of the personnel

Only personnel qualified as electrical technicians and controller technicians in accordance with IEV 826-09-01 (modified) and who are familiar with the contents of this manual are permitted to set parameters, commission and operate the TL unit. The technicians must be able to detect potential dangers that may be caused by setting parameters, changing parameter values and generally by the mechanical, electrical and electronic equipment.

Personnel can have the technical training, knowledge and experience to assess the work to be done and to recognize and avoid possible dangers or non-qualified personnel can have the same knowledge as qualified personnel after several years of work in the same area.

Qualified personnel must be familiar with the current standards, regulations and accident prevention regulations which have to be observed when working on the unit.

4 Communication in the field bus

4.1 Control access to Twin Line units

Local and remote access

Data exchange and control of Twin Line units is possible over various access channels:

- locally via the RS232 interface with the HMI hand-held control unit or the TL CT operating software or the signal interface
- remote control over the field bus.

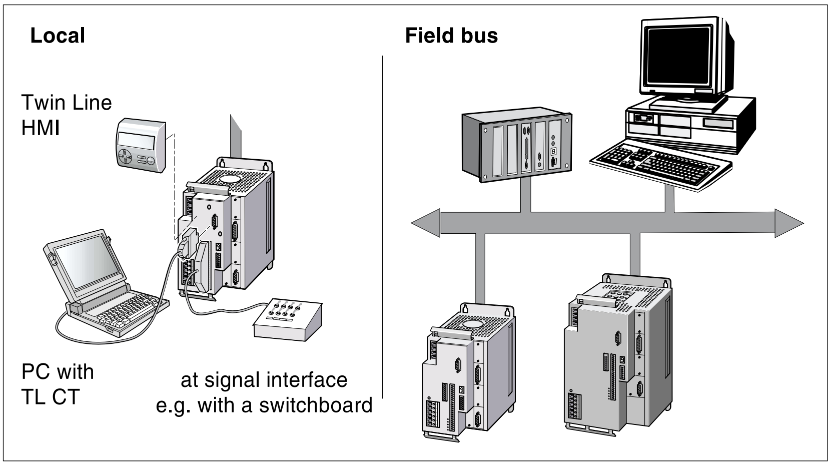


Fig. 4.1 Local and remote access to Twin Line units

Locally the Twin Line unit controls access rights and channel clearance automatically. A running movement command via local devices cannot be interrupted by field bus commands.

If a movement command is triggered over the field bus, a movement cannot be triggered simultaneously via a local device. This ensures that a running movement command can be completed in a controlled manner through the field bus.

Access by local control devices to the Twin Line unit can also be blocked and released by field bus commands. The local channels become available when the master device releases them or if field bus mode is interrupted.

Communication via parameters

The parameters of the Twin Line units are the basis for communication between Twin Line units and the field bus master device. A field bus command addresses every parameter via an index and subindex.

Initiating absolute point-to-point positioning

parameter name:	PTP.p_absPTP
index: sub-index:	35:01
target position:	4650 inkrcements

field bus command

...	01	35	4650

Fig. 4.2 Example of a parameter in a field bus command

The number of usable parameters depends on the Twin Line unit. A list of all parameters is included in the Twin Line unit manual. The parameters are listed first in their functional context to every operating mode and are listed again in a group overview at the end of the manual.

The number format of the parameter values in a field bus command can be found in the group overview in the "Parameters" chapter of the manual.

Example: the conversion factor from current strength to values to be input for the parameter I_max (18:2) is: **100=1Apk**. If you input 100, you are setting a current of 1Apk.

4.1.1 Data exchange

Online command processing

The master device sends a command to the Twin Line unit to have a movement assignment carried out, activate operating functions or request information from the controller. The controller carries out the command and acknowledges its successful execution.

The exchange of data follows a fixed routine. The process is always viewed from the point of view of the master device:

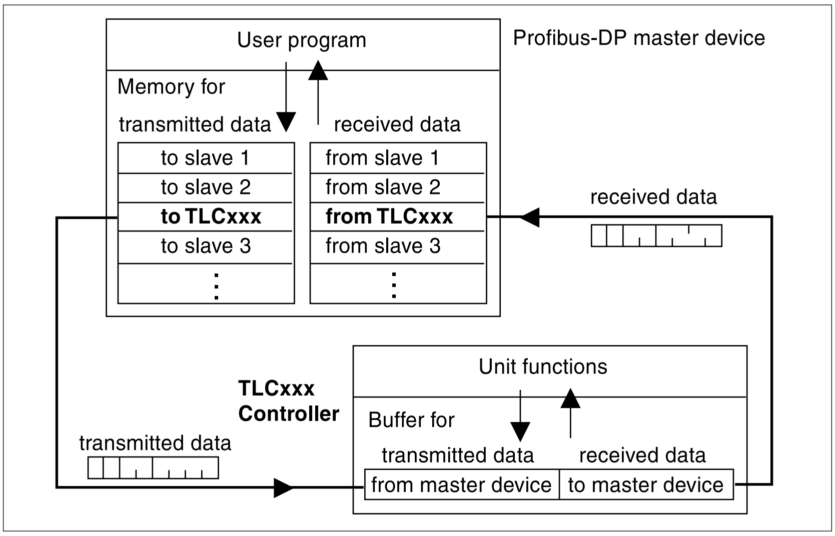


Fig. 4.3 Communications between master device and controller

- "Transmitted data" to the controller: The master device places a command in the data transmission memory. From there it is transmitted to the controller and carried out.
- "Received data" from the controller: The controller acknowledges the execution status of the command in the received data. If the master device receives an acknowledgment with no error message, the command has been correctly executed.

The master device can send new commands as soon as it has received acknowledgment of the current command.

Acknowledgment information and error information are included in the transmitted data in bit-coded form.

In the case of cyclical field bus transmission, the master device automatically receives the latest received data from the slave device at every cycle. It can recognize via the acknowledgment mechanism whether the received data were status information from the slave device or an answer to a command sent previously. The slave device also uses the acknowledgment mechanism to recognize a new command.

Commands The master device transmits control commands and action commands with the transmitted data. After sending a control command, it receives an acknowledgment from the controller confirming whether the processing operation has been successfully carried out and completed.

In the case of an action command the controller merely reports whether an action or movement command has been successfully initiated. The master device must then continuously monitor for the completion of the process by evaluating the data received from the controller.

Details of both commands are described in the "Action and control commands" chapter from page 4-12.

4.1.2 Data structure

In addition to command and control information, transmitted and received data also contain administration data for monitoring network operations. Administration data are provided by the user program in the master device.

To enable communication with the controller on the network, transmitted and received data must be exchanged, programmed and evaluated in an eight-byte data frame.

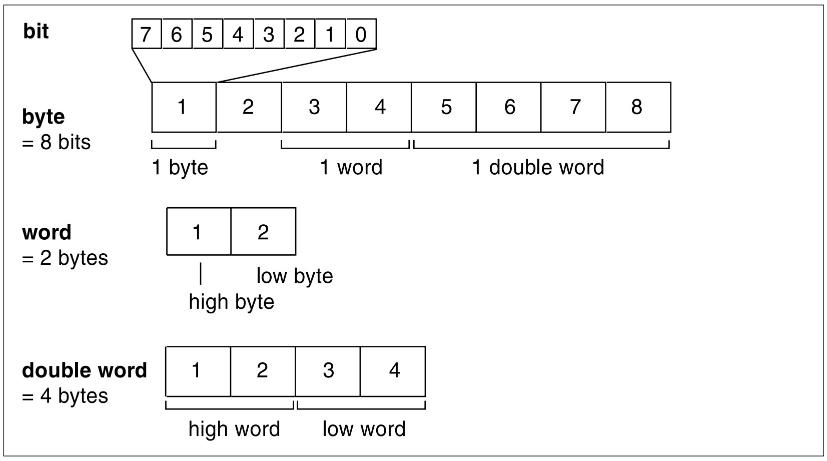


Fig. 4.4 Data structure

The first byte transmitted (byte 1) in the eight-byte block contains acknowledgment information for coordinating data exchange.

The data frame with transmitted and received data and all byte, word and double word values is given in hexadecimal notation in the manual. Hexadecimal values are indicated with an "h" after the value, e.g. "31h".

4.1.3 Data frame for transmitted data

The master device uses transmitted data to send a control or action command to the controller.

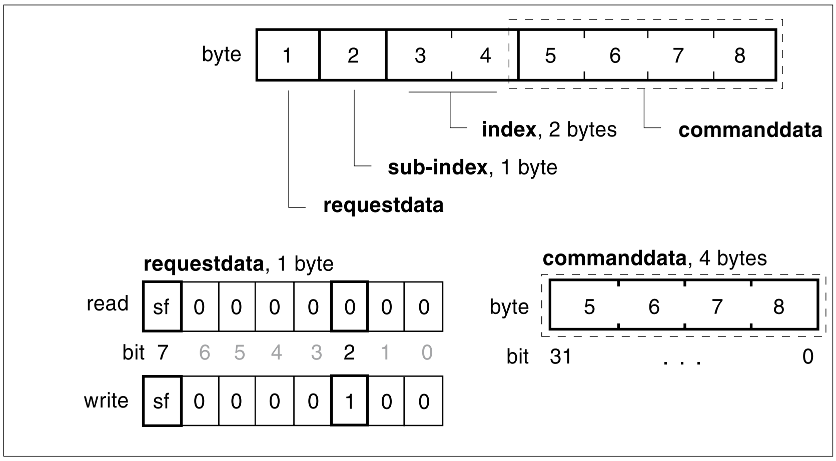


Fig. 4.5 Transmitted data frame

Byte 1: requestdata

This byte contains control information for acknowledging and synchronizing as well as for differentiating whether the command is a write command or a read command.

Bit	Name	Meaning
2	–	0: Read value: The controller reads a value defined by index and subindex and places it on the bus. 1: Write value: A parameter is written to the controller.
7	sf	(sf: sendflag) The master device flags a new command for the controller by changing the signal to the "sf" bit. Used together with "rf" in received data.

The acknowledgment mechanism via "sf" and "rf" is described on page 4-8.

Byte 2..0.4: subindex, index

Index and subindex are used to address the parameter which is evaluated as a command in the controller. The allocation of parameters for index and subindex and the parameter settings are described in the controller manual.

Byte 5...8: commanddata

The four bytes contain settings for the parameter transmitted to the controller as a command, e.g. the set speed for a PTP positioning maneuver.

When a parameter of type INT16 or UINT16 is transmitted, the value is stored in commanddata in bytes 7 and 8. The entries in bytes 5 and 6 have the value 0.

The data types used can be found in the positioning controller manual.

4.1.4 Data frame for received data

Data frame for received data

Received data supply the controller's answer to a command. At the same time they include information on the controller's operating status.

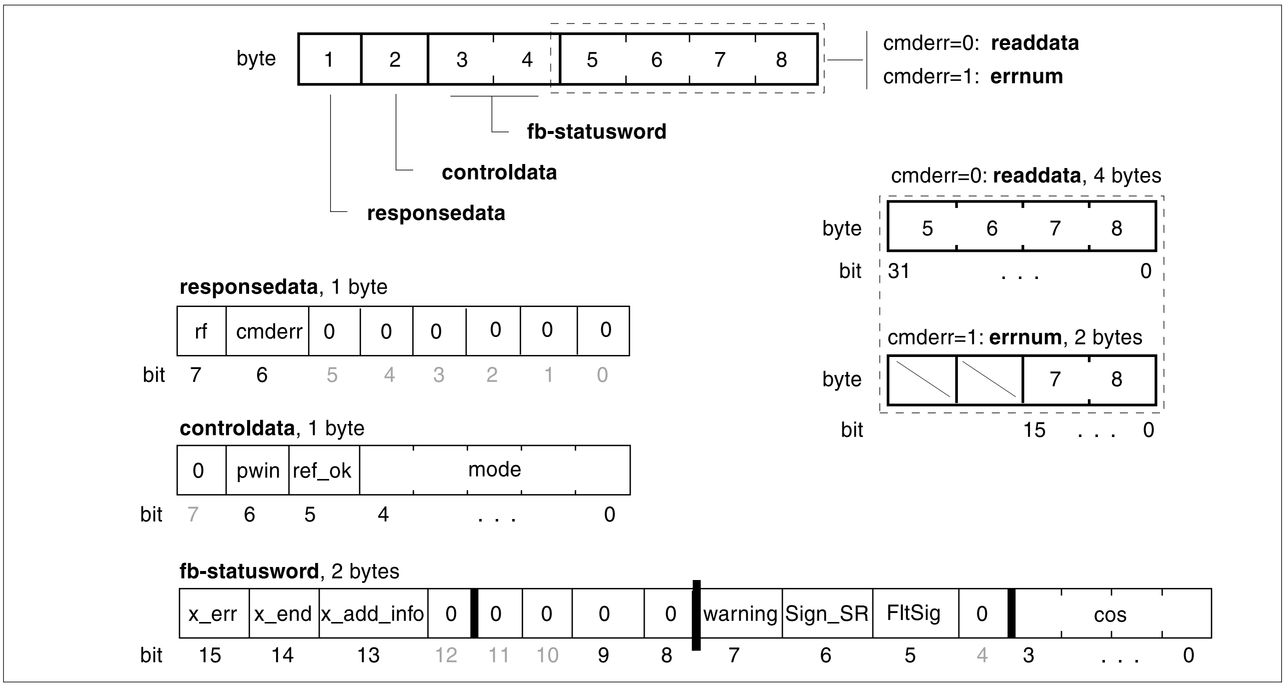


Fig. 4.6 Received data frame

Byte 1: responsedata

The first byte contains the response data from the controller on acknowledgment, synchronization, error detection and identification of the field bus service.

Bit	Name	Meaning
6	cmderr	Command error (cmderr: command error), signal is only valid after correct acknowledgment of a data package. 0: Command was executed without error. 1: command error, bytes 7 and 8 contain the error number "errnum".
7	rf	Controller's acknowledgment of receipt by changing "rf" bit (rf: receiveflag). rf = sf: Command received and initiated. rf ≠ sf: New command not yet processed.

The acknowledgment mechanism via "sf" and "rf" is described on page 4-8.

Byte 2: controldata

The controller uses the control data to provide information on the set operating mode and also supplies axis data. The control data can also be determined via the low byte of the "Status.xMode_act" parameter (28:3).

Bit	Name	Meaning
0..4	mode	Current axis mode, bit-coded Example: 00011 – PTP positioning 1: manual movement 2: referencing 3: PTP positioning 4: speed mode 5: electronic gear
5	ref_ok	0: No reference point specified. 1: axis has been referenced.
6	pwin	Standstill window, permissible control deviation 0: no standstill 1: motor stopped in standstill window.

Byte 3, 4: fb-statusword

The operating status of the controller is monitored with the status signals. This information can also be queried via the low word of the "driveStat" parameter (28:2).

Bit	Name	Meaning
0..3	cos	Operating status of the controller, bit-coded Details on the display and detection of operating states are described in the chapter on diagnostics and troubleshooting in the controller manual.
5	FltSig	Internal monitoring signals 0: no error detected 1: error detected, cause via parameter "Status.FltSig_SR" (28:18) and parameter "Status.IntSigSr" (29:34)
6	Sign_SR	External monitoring signals 0: no error detected 1: error detected, cause via parameter "Status.Sign_SR" (28:15)
7	warning	Warning message 0: no warning message 1: warning message Cause via parameter "Status.FltSig_SR" (28:18) and parameter "Status.IntSigSr" (29:34)
13	x_add_info	Status bit for monitoring the process status, see page 4-12
14	x_end	Status bit for monitoring the process status, see page 4-12
15	x_err	Status bit for monitoring the process status, see page 4-12

Byte 5..8: readdata The "readdata" read data contain controller information, such as the current motor position. The controller sends this updated information to the master with every received data set. The master device requests new read data with a command with the status "read value", see "Data frame for transmitted data", page 4-4, byte 1.

During transmission of a parameter of data type INT16 or UINT16 the value is saved in read data in byte 7 and 8; the entries in byte 5 and 6 have no significance.

The data types used can be found in the positioning controller manual.

If no read value has been requested, the controller sends the current axis position.

Read data are transmitted when the command could be executed without errors.

Byte 7, 8: errnum If a command is not executed correctly, the command error bit "cmderr" reports an error in byte 1. The cause of the error can then be found via the error number "errnum". There is a list with the error numbers in the chapter on diagnosis and troubleshooting in the controller manual.

4.2 Acknowledgment mechanism

A special acknowledgment procedure is required for a slave device to identify and confirm a new command. The acknowledgment mechanism supervises the exchange of data between master device and control unit, and synchronizes both devices when data transmission is cyclical on the field bus.

Only when a command has been acknowledged

- the other received data are valid as a response to the command
- a new command can be processed by the controller.

Three bits are significant for the acknowledgment mechanism:

- "sf" bit in the transmitted data
- "rf" bit in the received data
- "cmderr" bit in the received data.

4.2.1 Function of "sf" and "rf" bit

When field bus operation is started, the "sf" and "rf" bits are defined at "0" level.

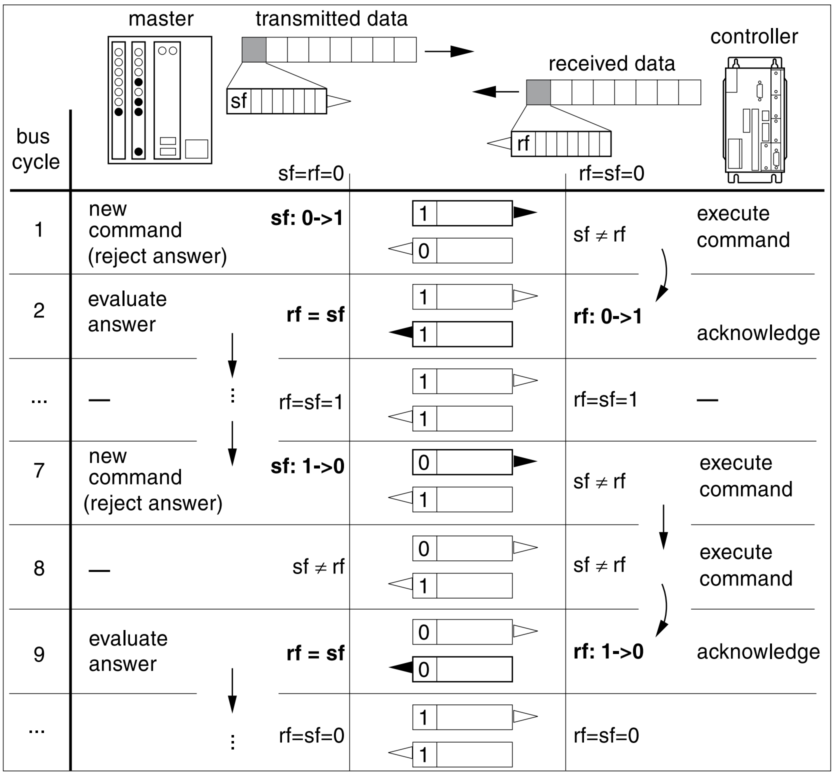
New command from master device

The master device user software marks a new command by switching the "sf" bit.

Evaluation by slave device

The controller compares the "sf" and "rf" bits

- "sf" ≠ "rf": command is new - execute
- "sf" = "rf": command has already been processed - reject



bus cycle		sf=rf=0		rf=sf=0	
1	new command (reject answer)	**sf: 0->1**	1 0	sf ≠ rf	execute command
2	evaluate answer	**rf = sf**	1 1	**rf: 0->1**	acknowledge
...	—	rf=sf=1	1 1	rf=sf=1	—
7	new command (reject answer)	**sf: 1->0**	0 1	sf ≠ rf	execute command
8	—	sf ≠ rf	0 1	sf ≠ rf	execute command
9	evaluate answer	**rf = sf**	0 0	**rf: 1->0**	acknowledge
...		rf=sf=0	1 1	rf=sf=0	

Fig. 4.7 Data exchange and synchronization with "rf" and "sf" bits

Response from the slave

If the command has been carried out, the controller switches over the "rf" bit and sends the signal together with the response data back to the master device.

Evaluation by the master device

The master device checks the "rf" bit:

- "sf" = "rf": command executed – evaluate received data.
- "sf" ≠ "rf": command not yet acknowledged – wait.

An unacknowledged command has either not yet been carried out, or it has not been received and must be sent again with the same "sf" bit. This happens automatically with continuous data transmission.

4.2.2 Command error bit "cmderr"

The command error bit is valid when the command has been acknowledged.

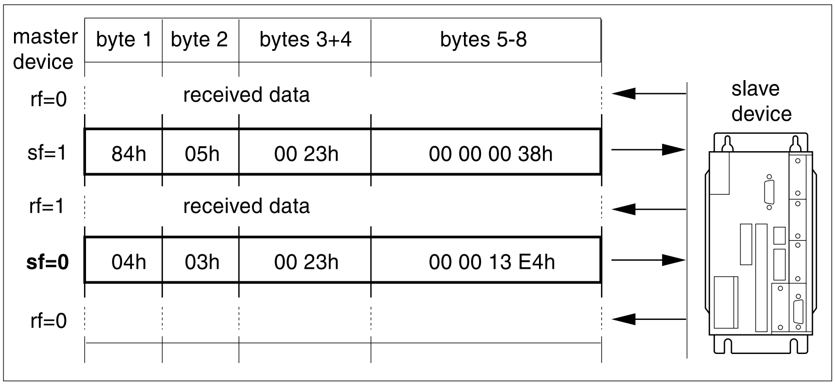
- "cmderr"=0: command has been successfully carried out
- "cmderr"=1: a synchronous error has occurred.

In the event of a synchronous error, the controller returns an error number "errnum" in byte 7 and 8 of the received data from which the cause of the error can be determined. The error numbers are listed in the chapter on diagnosis and troubleshooting.

You will find information on synchronous errors in the chapter on "Error handling" on page 7-1.

4.2.3 Example of a positioning command

A relative positioning command is transmitted to the controller.



master device	byte 1	byte 2	bytes 3+4	bytes 5-8
rf=0	received data			
sf=1	84h	05h	00 23h	00 00 00 38h
rf=1	received data			
sf=0	04h	03h	00 23h	00 00 13 E4h
rf=0				

Fig. 4.8 Positioning command

The first command sets the speed, and the second initiates a positioning process. The bytes mean:

- Byte 1 (requestdata): Send new command: switch over "sf" bit; write access with bit2 = 1: 4h.
- Byte 2.-4: subindex and index for the command 05h:00 23h for PTP set speed "PTP.v_target" (35:5) 03h:00 23h for relative positioning "PTP.p_relPTP" (35:3).
- Byte 5-8 (commanddata): setting value for the command
  00 00 00 38h: PTP speed, here 38h = 56 units
  00 00 13 E4h: positioning path, here 13E4h = 5.092 units.

You will find detailed examples on all operating modes of Twin Line units in the chapter on "Examples for field bus operation" from page 6-1.

4.2.4 Monitoring communications with field bus

Diagnostic values are available for monitoring activities on the field bus. They can be used to read the transmitted and received data and display statistical information.

The following diagnostic values are available to monitor field bus communications.

- Content of **controller** transmitted data
- Content of **controller** received data
- Bus statistics for determining the frequency of communications errors

Note that the diagnostic values show the content of the transmitted and received data from the point of view of the slave device and not from that of the master device.

Transmitted and received data

The transmitted and received data of the controller can be shown with the TL CT commissioning software or the HMI hand-held unit. Diagnostics over the field bus is not feasible, because the data are changed by the access.

The current content of the transmitted and received data can be determined with the following diagnostic values.

Parameter			**Explanation and unit [ ]**	**Range of values**	**Default**	**R/W**
Group.Name	**Idx:Sidx**	**TL-HMI**			**Value**	**rem.**
M4.busTxD	24:33	2.6.2	Transmitted data online command processing (Byte 1... 4)	UINT32 0...4294967295	0	R/– –
M4.busTxD5_8	24:34	2.6.2	Transmitted data online command processing (Byte 5... 8)	UINT32 0...4294967295	0	R/– –
M4.busRxD	24:28	2.6.1	Received data online command processing (Byte 1... 4)	UINT32 0...4294967295	0	R/– –
M4.busRxD5_8	24:29	2.6.1	Received data online command processing (Byte 5... 8)	UINT32 0...4294967295	0	R/– –

Bus statistics The statistical data can be queried with the TL CT commissioning software, the HMI hand-held unit or over the field bus.

The bus statistics can be used to find information on the number of timeout errors and bus cycles. The total of all errors that resulted in breaking a connection can be determined. The following diagnostic values in the M4 parameter group are available:

The various statistical values can be reset to zero by a write access with the value 0.

Parameter			Explanation and unit [ ]	Range of values	Default	R/W
Group.Name	Idx:Sidx	TL-HMI			Value	rem.
M4.busTout	24:31	2.6.6	Bus statistics timeout: Total number of broken connections caused by timeout (node guarding)	UINT16 0...65535	0	R/W rem.
M4.busError	24:32	2.6.7	Bus statistics transmission error: Total of all errors that resulted in a connection break	UINT16 0...65535	0	R/W rem.
M4.busCycle	24:35	2.6.8	Bus statistics bus cycles Total number of all processed bus cycles	UINT32 0...4294967295	0	R/W rem.

4.3 Action and control commands

The master can send two types of commands: Action commands or control commands. The controller reacts differently depending on the type of commands.

Control commands Control commands are executed immediately and are complete when the received data are returned. Control commands can be used for items such as changing parameters or switching outputs.

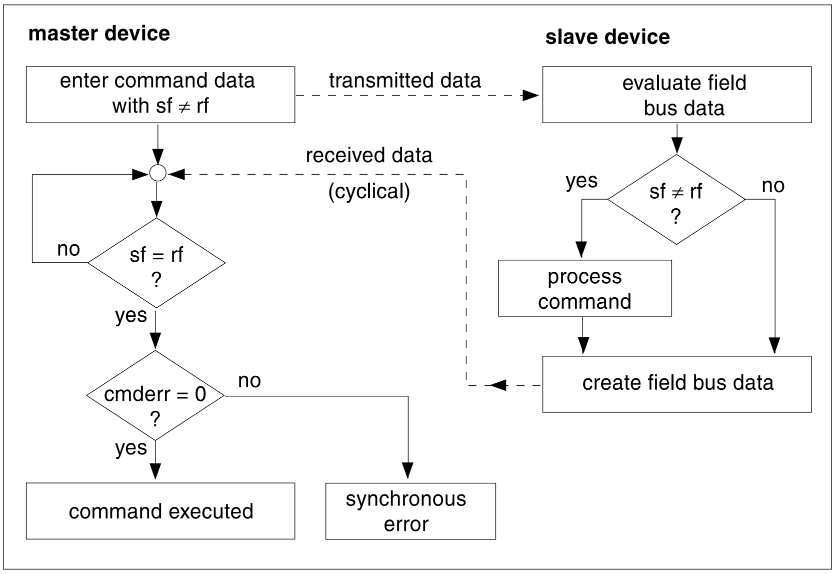


Fig. 4.9 Carrying out a control command

If a command could not be correctly executed, the controller sets the command error bit "cmderr" to "1" and reports a synchronous error.

Action commands

Action commands start a movement. The controller activates the relevant operating mode and loads the required parameters. It reports the start of the movement assignment to the master as a command confirmation. For example, an action command can start a positioning process.

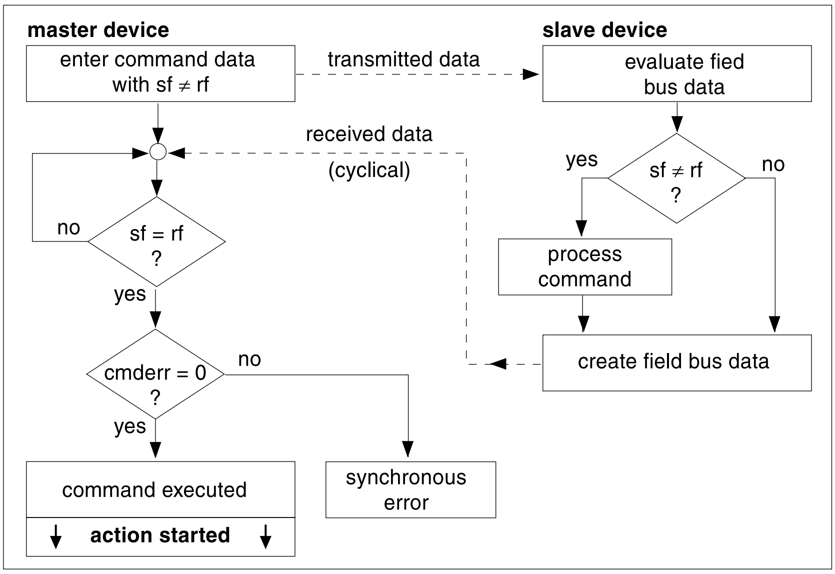


Fig. 4.10 Carrying out an action command

If an action could not be correctly initiated, the controller sets the command error bit "cmderr" to "1" and reports a synchronous error.

Monitoring the operating status

The operating status and the completion of the movement command must be continuously monitored by the master with the statusword "fb-statusword" in the received data.

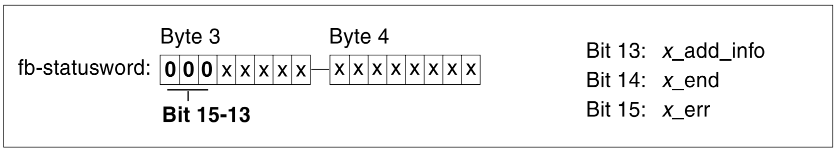


Fig. 4.11 Monitoring the controller operating status

Status bits

The controller manages operating states internally for every mode with separate status bits. It reports only the status of the current mode over the field bus. The status bits have the following meaning:

- bit13, "x_add_info": mode-specific message.
- bit14, "x_end": Processing status of operating mode
  0: process running
  1: process finished, motor stopped.
- bit15, "x_err": Error status during processing:
  0: error-free operation 1: error has occurred

The information on the current mode allows the status message to be evaluated specific to the mode. The current operating mode is in bit0 to 4 d("mode") in "controldata".

The following table shows the allocation of operating mode and status bits:

Operating mode	mode	x_add_info	x_end	x_err
Manual mode	1	0	manu_end	manu_err
Referencing	2	0	ref_end	ref_err
PTP positioning	3	Set position reached	motion_end	motion_err
Speed mode	4	Set speed reached	vel_end	vel_err
Electronic gear	5	gear_sync_ window	gear_end	gear_err
Data set mode	7	VEL data set: Set speed reaches PTP data set: Motor standstill in target position, drive in standstill window	record_end	record_err
Current control	17	curr_ctrl_nact_ zero	curr_ctrl_end	curr_ctrl_err
Oscillator mode	18	Set speed reached	oscillator_end	oscillator_err

As soon as processing is initiated by an action command, bit14 "x_end" switches to "0". When the process is finished, bit14 changes back to '1' thereby indicating that further process steps can now be executed. The signal change is suppressed if a process is followed immediately by a new process in a different mode.

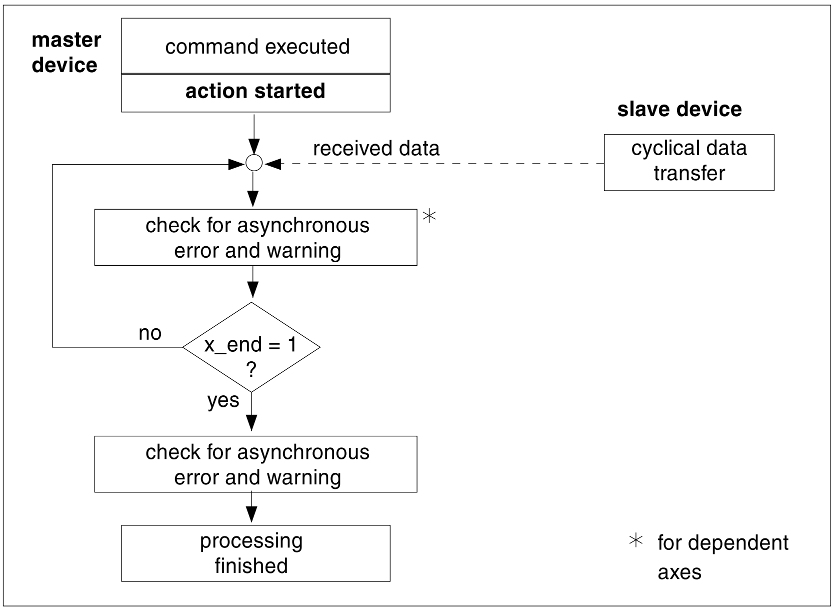


Fig. 4.12 Monitoring the execution of an action command

If bit15 "x_err" switches to 1, there is an error that must be corrected before any further processing. The path for checking an asynchronous error or a warning is described in chapter "Error handling" from page 7-1. A check during processing is only required if, for example, additional dependent drives must be stopped immediately.

Changing operating mode

The controller can execute additional commands during the execution of an action command, for example to change the setpoint speed of a positioning in progress or to change the operating mode. The command error bit "cmderr" shows information on the successful execution of the command.

4.4 PLC as a field bus master device

The field bus master device allocates each connected device its own memory space for transmitted and received data. Data exchange between PLC memory and field bus master module can be carried out via the peripheral or process image area.

The field bus transmission runs asynchronously to the application program's read/write accesses of transmitted and received data. As a result, data from the field bus master module may be read from the PLC memory before the PLC has had time to update the data completely.

DANGER!
Danger of personal injury and damage to system parts resulting from processing of faulty control commands!
If a PLC is used as the master device, the exchange of data can lead to inconsistent transmission data as a result of field bus and PLC cycles not operating synchronously.

Observe the following to ensure safe operation of a master PLC:

- Copy data from high to low addresses.
- Transmit "sf" control bit last.
- When exchanging data via the process image, the transmitted data must be copied from the process image memory to the field bus master module memory. This copy process should ensure that no inconsistent data are created on the field bus.

The "sf" bit denotes a new command. After entering all data in the peripheral memory, the "sf" transmission sequence bit is inverted and thus denotes a new command from the master device to the slave device.

Data exchange via peripheral memory

Data integrity is ensured in exchanging data via the peripheral memory if the "sf" bit is entered last. The controller ignores the transmitted data as long as "sf" and "rf" bits have the same value.

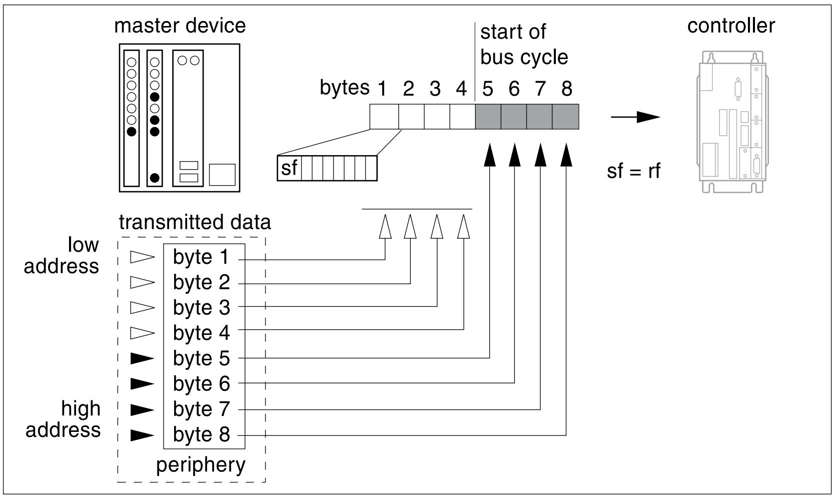


Fig. 4.13 Guaranteed data integrity, "sf" bit is copied last.

Data exchange via process image memory

Data integrity when exchanging data over the process image memory is only guaranteed when the data in the peripheral memory are not accessed during the copy process between process image and peripheral memories from the lower to the higher address.

The following example shows the problem of unsafe data integrity.

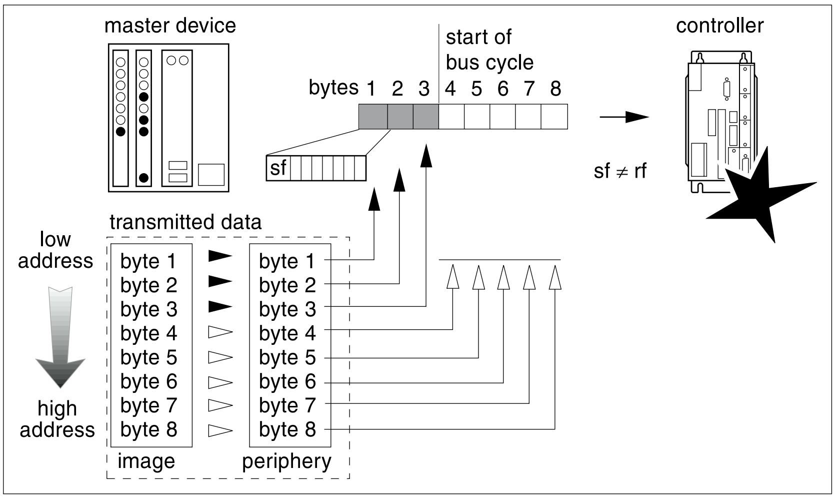


Fig. 4.14 Insecure data integrity

The data have been copied correctly to the process image area. When copying from the process image memory to the peripheral memory from a low to a high address, bytes 1 to 3 were updated.

As a result of the copy process being interrupted by the field bus cycle, the updated values are transmitted to the controller for bytes 1 to 3 but the old values for bytes 4 to 8.

Data integrity is not ensured in this case.

You must refer to the manufacturer's specifications for the field bus unit to see whether data integrity can be guaranteed by the field bus master device in use.

4.5 Replacing devices

After a slave unit has been replaced, the new unit should operate in exactly the same way as the old one. The new unit must have the same parameter settings to ensure this.

In operating mode case IO_mode = 0 the unit detects the values of the field bus parameters via the signal interface inputs when starting up. In the case of IO_mode ≠ 0, the field bus parameters must be set previously via the HMI or the TL CT software, otherwise no communication will be possible with the device in the field bus network.

If the default values of other parameters need to be changed, these values can be stored on the master controller. They must be transmitted every time the Twin Line units starts up, e.g. in "ReadyToSwitchOn" status.

Your local representative can supply drivers for controlling Twin Line units with PLC controllers made by Siemens. If you require these drivers, please contact your local representative.

5 Installation and setup

5.1 EMC

EMC requirements must be taken into account when laying and connecting cables in an electromagnetically charged environment.

EMC measures

EMC requirements must be taken into account when laying and connecting cables in an electromagnetically charged environment.

EMC measures	Effect
Use cables with braided and foil shields	Deflect interference voltages
Field bus cables can be laid in one conduit with signal wires and analog wires. Do not lay them with DC and AC cables carrying over 60 V.	Avoid mutual interference coupling
Use bonding lines in wide-area systems with - wide-area installations – different voltage feeds – networking between buildings	Deflect interference currents
Use fine-core bonding conductors Deflect even high-frequency interference currents	Deflect even high-frequency interference currents

In digital cables the shields are connected at both ends to protect against interference. Potential differences can result in excessive currents on the shield and must be prevented by using bonding lines. For cables up to 200 m in length an cross section of 16 mm^2 is sufficient, but for longer cables a cross section of 20 mm^2 must be used.

5.2 Installation

5.2.1 Installation and setup of unit

The Twin Line unit requires correct mechanical and electrical installation and successful setup are required before installation in the network.

Set up the unit following the manual. This procedure prepares the Twin Line unit for operation in the network.

5.2.2 Setting the address

Every device in the field bus is identified by a unique address. The network address for a Twin Line unit is preset to 126.

Setting the address There are two options for inputting the address. The "Settings.IO_mode" parameter (29:31) sets up the unit for one of the two options:

- "IO_mode" = "0": Setting via the signal interface
- "IO_mode" = "1" or "2": Setting via parameters.

When making a setting via the signal interface, the controller interprets the switching status of the ADR_1 to ADR64 inputs as the network address. Assignment and settings options for the interface are described in the controller manual in the chapter on connecting the signal interface.

When setting the address via parameters, the address is entered locally with an operating unit in the "M4.adrPdb" parameter (24:21).

Baud rate The baud rate of the controller automatically adapts to the rate set by the master device.

5.2.3 Connecting the Twin Line unit

Twin Line units are fitted with the module PBDP-C in slot M4 for connection to a Profibus-DP network.

Module interface

The PBDP-C module is fitted with a 9-pole, UNC-threaded, sub-D connector.

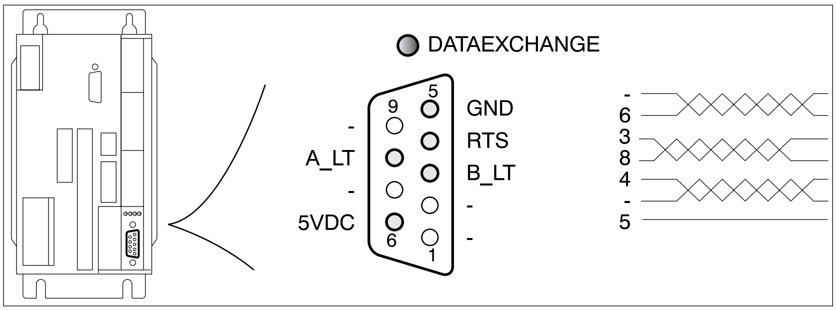


Fig. 5.1 Field bus module interface connection

Pin	Signal	Color [1]	Pairs	Meaning	I/O
1	-	-	1	not assigned	-
6	5VDC	white	1	power supply, max. 10 mA for terminator	O
2	-	-	-	not assigned	
7	-	-	-	not assigned	
3	B_LT	green	2	data wire, inverted	I, O
8	A_LT	yellow	2	data wire	I, O
4	RTS	gray	3	transmission request	O
9	-	-	3	not assigned	-
5	GND	black	-	ground	-

[1] Color details refer to the cable available as an accessory.

A bus terminal forms the docking station to the Profibus. Data lines between module and bus terminal are wired up 1:1.

Terminals A_LT and B_LT must be connected to wires A and B in the network in accordance with the minimum wiring requirement.

Cable specification for connection to a bus terminal

- shielded cable
- minimum cross-section of signal wires 0.14 mm^2
- twisted-pair cables
- shield grounded at both ends
- maximum length 100 m

For units with a hood, the cable must be led downwards from the point of connection.

Field bus cable

The permissible cable length depends on the transmission speed and the type of cable. The Profibus standard specifies cable types A and B. Cable type A is suitable for fast transmission and long distances, and cable type B for lower thresholds.

Baud rate (kbit/s)	**9,6 - 33,7**	**187,5**	**500**	**1500**	**12000**
distance/segment for cable type A	1,200 m	1,000 m	400 m	200 m	100 m

5.2.4 Termination

Field bus devices are connected in segments with a linear structure, consisting of 32 master and slave devices. Each segment must be terminated at both ends with an active bus terminator. When using the Twin Line unit's network accessory cable, it can be terminated with a switch in the connecting plug to external bus devices.

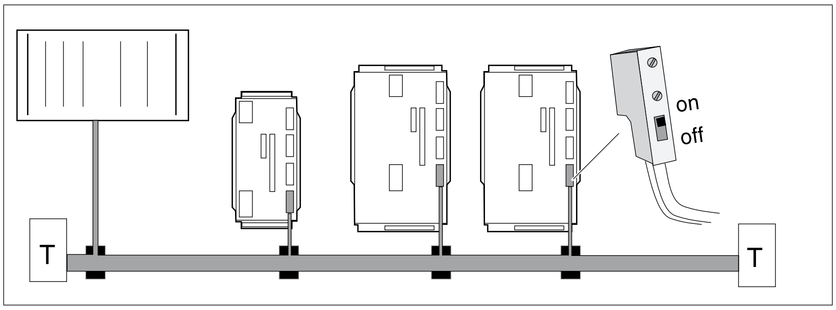


Fig. 5.2 Termination of a bus segment via plug or terminator (T)

Where there are more than 32 devices, individual segments are connected by repeaters. Every segment must then be terminated separately.

5.2.5 Configuring network master devices

The Twin Line unit comes with a device master data file (GSD file), which must be read in via the master device's network configuration program to integrate the controller into the network as a slave device.

The master device thereby registers the controller in the network and records its transmission properties.

5.3 Commissioning

5.3.1 Initiating network operation

Network operation with the Twin Line unit is started via a master device. This can be a PLC or a PC with the appropriate application software with which commands can be entered and received data read.

When you plug the HMI hand-held operating unit into the Twin Line unit, you can watch transmitted and received data on the display in the "2.6 field bus diagnostics" menu.

- Switch on the Twin Line unit.
- Start network operation. Received data from the controller are transmitted to the master device in the bus cycle.

If the received data contain a value not equal to zero, logical data connection has been established between master device and controller.

For example, you can now read off the current position of the motor with the "Status.p_act" parameter (31:6). Index is 1Fh, subindex 6h.

- Enter the following command:
  80h 06h 001Fh 0000.0000h

  Response from the controller: 80h xxh 000xh 000.0138h.

This is the current position 138h=312 increments. The values of "x" depend on the current operating status of the controller.

5.3.2 Troubleshooting

If you receive the value zero in the received data, check the following settings:

- Units switched on and master device started for network operation?
- Cables mechanically sound?
- Is the LED on the field bus input on the controller on? If not, either network operation or addressing is not working. The LED shows data traffic over the network interface.
- Address correctly set in controller?
- Has the correct device master data file been installed on the master device?
- Does the status display on the Twin Line unit show "3", "4" or "6" without flashing? If not, there is an operating error in the unit. See the controller manual for information on the cause of errors and troubleshooting.

6 Examples for field bus operation

6.1 Structure of program examples

Overview

The program examples show practical applications for the use of Twin Line units in networks. The following is shown:

- description of task
- initial conditions
- commands required in transmitted data frame
- response of the unit in receiving
- possible limitations on command execution.

You should be aware of the following to be able to reproduce the examples:

- operating concept and functional scope of your Twin Line unit. You will find information on this in the manual.
- field bus protocol and connection to the master controller
- functional scope of the field bus profile

Because the field bus manuals are used for different device types and some commands are not available for all device types, various examples indicate for which device types the functions described are available under "Applicable Device Types". If no information is given, the examples can be run with all device types.

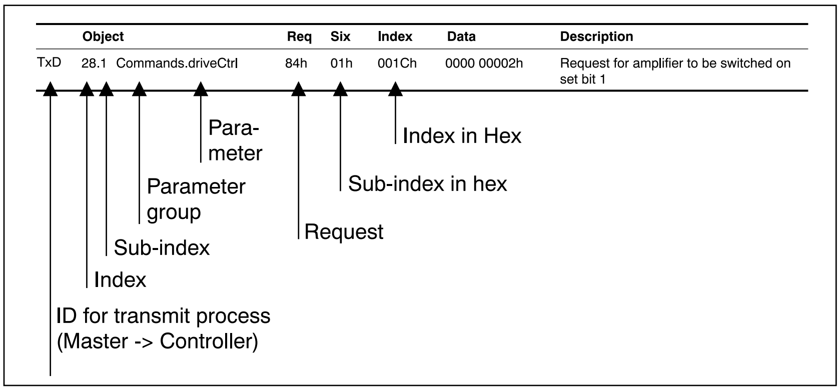
Transmitted and received data

Transmitted and received data are shown in hexadecimal form.

The master command is given. The response of the device after correct execution is only shown if this is necessary for the continuing description. Otherwise a positive acknowledgement of the command is assumed:

Transmitted data

	Object	Req	Six	Index	Data	Description
TxD	28.1 Commands.driveCtrl	84h	01h	001Ch	0000 0002h	Request to switch on power amplifier: set bit1



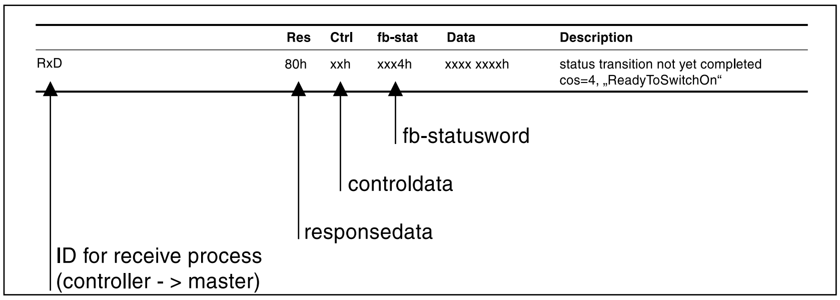
	Object	Req	Six	Index	Data	Description
TxD	28.1 Commands.driveCtrl	84h	01h	001Ch	0000 00002h	Request for amplifier to be switched on set bit 1

Fig. 6.1 Transmitted data coding

The data type of the value written can be taken from the "value range" column in the parameter description of the manual. On transfer of a INT16 or UINT16 value, the value is saved to bytes 7 and 8 – 0 must be entered in bytes 5 and 6.

Received data

	Res	Ctrl	fb-stat	Data	Description
RxD	80h	xxh	xxx4h	xxxx xxxxh	Status transition still unsuccessful: cos=4, "ReadyToSwitchOn"



	Res	Ctrl	fb-stat	Data	Description
RxD	80h	xxh	xxx4h	xxxx xxxxh	status transition not yet completed cos=4, „ReadyToSwitchOn“

Fig. 6.2 Received data coding

The data type of the value written can be taken from the "value range" column in the parameter description of the manual. When an INT16 or UINT16 value is read, the value is saved to bytes 7 and 8 – bytes 5 and 6 show 0. The error number of a synchronous error message is saved as a UINT16 value.

Irrelevant values Values which are not relevant for the example are shown with an x.

Value units

If read or write values are shown in user-defined units [usr], they must still be set off with calibration factors. You will find information on this in your manual in the "Functions" chapter under "Calibration" and on page 6-28.

Acknowledgment bits

In all examples the value "0" is assumed for the acknowledgment bits "sf" and "rf" before the first transmit command. The first command must therefore transmit with "sf" = 1. This also applies when the description of a subject is split across several examples. A new command can be recognized by the change in level.

Data frame

In the examples only the 8 byte frame for Twin Line units is shown. The control byte specific to the field bus must be added by the master device user program in accordance with the field bus protocol.

Manual

The examples are designed to complement the function descriptions in the manuals. The manual describes the basic functions of the operating modes and functions. You will also find all parameters regarding the operating modes and functions listed there.

The number format of the parameter values in a field bus command can be found in the group overview in the "Parameters" chapter of the manual.

Example: the conversion factor from current strength to values to be input for the parameter I_max (18:2) is: **100=1Apk**. If you input 100, you are setting a current of 1Apk.

6.2 Operating status

The unit must be ready to start and correctly initialized before an operating mode can be started.

Detailed information on the unit operating states and a status diagram with the various states numbered can be found in the manual for your Twin Line unit in "Operating displays and transitions".

6.2.1 Checking the operating status

The 7-segment display on the Twin Line unit shows the operating status. The unit is ready for operation when the display shows "6".

The operating status is evaluated over the field bus by the first four bits in the statusword "fb-statusword". The status word is transmitted with every received data set.

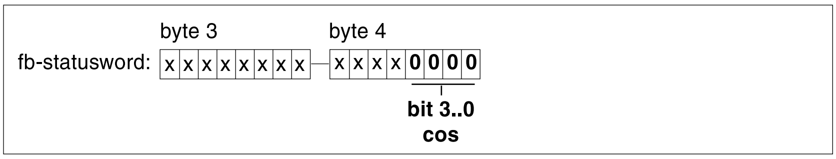


Fig. 6.3 Displaying the operating status of the unit

Bit3..0	Changing the operating	Meaning
-	-	24 V switched on
...0001	1 - Start	Initialization of the unit electronics
...0010	2 - Not ready to switch on	Power amplifier not ready to switch on
...0011	3 - Switch on disabled	Power amplifier cannot be switched on.
...0100	4 - Ready to switch on	Power amplifier is ready to switch on.
...0101	5 - Switched on	Power amplifier is switched on.
...0110	6 - Operation enable	Unit is working in the set operating mode.
...0111	7 – Quick Stop active	Quick Stop is executed
...1000	8 - Fault reaction active	Fault reaction activated
...1001	9 - Fault	Fault display

Operating states 0..3, 5, 8 and 9 are transition states in which the unit does not remain when it is operating correctly.

If the unit remains in operating status 1, 2 or 3 after switching on the 24 V power supply, a fault has occurred during initialization of the unit. Information on troubleshooting can be found in the manual at the end of the chapter on installation.

6.2.2 Changing the operating status

The operating status of the Twin Line unit is sent with bits "cos" in "fb-statusword". The coding corresponds to the status display on the unit.

Depending on the status of the unit, the operating status can be changed with field bus commands. For example, the status "ReadyToSwitchOn" can only be activated when the following conditions are met:

- unit started up after the 24 V power supply has been activated
- DC link power activated
- no fault present.

Changes of status in the unit are made with the "Commands.driveCtrl" parameter (28:1). The value is always with the result that write access to a bit automatically triggers an edge change 0 → 1.

Bit3..0	Control word	Meaning
0 0 0 1	Disable	Switching off the power amplifier
0 0 1 0	Enable	Switching on the power amplifier
0 1 0 0	Quick-Stop	Trigger stop with Quick-Stop
1 0 0 0	Fault Reset	Acknowledging fault message

Switching on the power amplifier

▶ Switch on the power amplifier.

This requires that the number "4" is visible on the 7-segment display of the unit, that the unit is in the "ReadyToSwitchOn" status and that the parameter setting in the unit is correct.

	Object	Req	Six	Index	Data	Description
TxD	28.1 Commands.driveCtrl	84h	01h	001Ch	0000 0002h	Request to switch on power amplifier: set bit1

	Res	Ctrl	fb-stat	Data	Description
RxD	80h	xxh	xxx4h	xxxx xxxxh	Status transition still unsuccessful: cos=4, "ReadyToSwitchOn"

If the status transition to the "OperationEnable" status was successful, the unit reports the following:

	Res	Ctrl	fb-stat	Data	Description
RxD	80h	xxh	xxx6h	xxxx xxxxh	Status transition complete: cos=6, "OperationEnable"

Internal monitoring signals are activated in bit5 (FltSig) and bit15 (x_err) in "fb-statusword" until the "OperationEnable" status has been reached

Status	fb-statusword x_err (Bit15)	fb-statusword FltSig (Bit5)
≠ OperationEnable	1	1
= OperationEnable	0	0

Switching off the power amplifier

▶ Switch off the power amplifier.

The unit must be in the "OperationEnable" status, i.e. the number "6" must be displayed on the 7-segment display.

	Object	Req	Six	Index	Data	Description
TxD	28.1 Commands.driveCtrl	84h	01h	001Ch	0000 0001h	Request to switch off power amplifier: set bit0

		Res	Ctrl	fb-stat	Data	Description
RxD		80h	xxh	xxx6h	xxxx xxxxh	Status transition still unsuccessful: cos=6, "OperationEnable"

If the status transfer to the "ReadyToSwitchOn" status was successful, the unit reports the following:

		Res	Ctrl	fb-stat	Data	Description
RxD		80h	xxh	xxx4h	xxxx xxxxh	Status transition complete: cos=4, "ReadyToSwitchOn"

Internal monitoring signals are activated in bit5 (FltSig) and bit15 (x_err) in the fb-statusword as soon as the unit exits the "OperationEnable" status.

Status	fb-statusword x_err (Bit15)	fb-statusword FltSig (Bit5)
≠ OperationEnable	1	1
= OperationEnable	0	0

Movement interruption by software stop

▶ Interrupt a current positioning process by a software stop.

The following conditions must be met:

- The unit must be in the OperationEnable status, see page 6-5.
- All required function settings must have been made, see page 6-24.
- All required settings must have been made via the operating modes, see page 6-13.

	Object	Req	Six	Index	Data	Description
TxD	28.1 Commands.driveCtrl	84h	01h	001Ch	0000 0004h	Request software stop: set bit2

If the "QuickStopActive" status was successfully activated, the following message is sent:

	Res	Ctrl	fb-stat	Data	Description
RxD	80h	xxh	8027h	xxxx xxxxh	In fb-statusword: x_err=1,FltSig=1,cos=7: "QuickStopActive"

As soon as the drive is stopped, the following message is sent:

	Res	Ctrl	fb-stat	Data	Description
RxD	80h	xxh	C027h	xxxx xxxxh	In fb-statusword: x_err=1,x_end=1,FltSig=1,cos=7: "QuickStopActive"

If the drive is at a standstill, the interruption status can be canceled with "Fault reset" – see page 6-35.

Acknowledging fault message

To acknowledge a fault message, see "Resetting faults (FaultReset)" on page 6-35.

6.3 Setting processing parameters

6.3.1 Non-operating-mode-specific processing parameters

The various settings for the non-mode-specific processing parameters are independent of the mode set by the operator. The non-mode-specific parameters are assigned to one of the following parameter groups:

- Settings
- Commands
- Servomotors
- CtrlBlock1, CtrlBlock2
- Motion
- Teach
- List
- List1Data0..List1Data63, List2Data0..List2Data63
- Capture
- I/O
- M1..M4
- ErrMem0..ErrMem19

Signal enable for monitoring parameters

▶ Deactivate limit switch monitoring, because the system only has reference switches and stop switches.

The unit must be set to the "ReadyToSwitchOn" status to be able to run the task, see 6-6.

	Object	Req	Six	Index	Data	Description
TxD	28.13 Settings.SignEnabl	84h	0Dh	001Ch	0000 000Ch	Activate monitoring inputs REF and STOP

6.3.2 Operating-mode-specific processing parameters

The various settings for the mode-specific processing parameters depend on the mode set by the operator. The non-mode-specific parameters are assigned to one of the following parameter groups:

- Manual
- VEL
- PTP
- Gear
- Home
- Record
- RecoData0..RecoData49
- Oscillator
- CurrentControl

Setting the speed for fast manual movement

▶ Set the speed for fast manual movement to 500 usr = 01 F4h usr.

This requires that the set speed is no greater than the entry in "Motion.v_target0", that the unit is in "Operation Enable" status (see "Changing the operating status" on page 6-5) and that the parameter "Manual.n_fastMan" is available on your Twin Line unit.

	Object	Req	Six	Index	Data	Description
TxD	41.5 Manual.n_fastMan	84h	05h	0029h	0000 01F4h	Setting fast manual movement speed 500 usr = 1F4h usr

You can set the speed for fast manual movement before initiating the manual movement or during a manual movement.

6.4 Reading device information

After a read access the following actions are executed:

- The current processing value is output.
- The data that should be output in the event of another access to the read data are defined. The data are output until new values are specified by a new read access.

See the parameter description in chapter 12 of the manual for your Twin Line unit for whether a parameter value can be read and the data type to which it corresponds. All values marked with "R" in the "R/W" column of the parameter list can be read; the data type is shown in the "Value Range" column.

6.4.1 Reading parameter settings

The current setting of a processing value can be read from the unit with a read value.

Reading the speed for fast manual movement

▶ Read the current setting for the speed of fast manual movement from the unit.

This requires that the read value is available on your unit and in the current unit status.

	Object	Req	Six	Index	Data	Description
TxD	41.5 Manual.n_fastMan	80h	05h	0029h	0000 0000h	Requesting speed [usr] for fast manual movement

		Res	Ctrl	fb-stat	Data	Description
RxD		80h	xxh	xxx6h	0000 00B4h	Speed in bytes 5 to 8: B4h = 180 usr

In the event of a read access to a 16-bit value, the value read is in bytes 7 and 8; the value of bytes 5 and 6 is 0.

6.4.2 Reading status information

Various read values are available in the status parameter group. They can be used to read the processing status from the unit.

Status information can be dependent on or independent of the operating mode. Mode-independent status information is communicated in controldata and fb-statusword; the coding corresponds to the assignment of the "Status.driveStat" parameter.

Operating-mode-independent status information

For example, determine the motor speed Status.n_act [rpm], and set the motor speed as a cyclic read value.

This requires that the read value is available on your unit and in the current unit status.

	Object	Req	Six	Index	Data	Description
TxD	31.9 Status.n_act	80h	09h	001Fh	0000 0000h	Requesting the actual speed in [rpm]

		Res	Ctrl	fb-stat	Data	Description
RxD		80h	xxh	xxx6h	0000 03E8h	Actual speed in byte 7 and 8: 3E8h=1000 rpm

In the event of a read access to a 16-bit value, the value read is in bytes 7 and 8; the value of bytes 5 and 6 is 0.

Operating-mode-dependent status information

Every mode has its own acknowledgement object. As an example, the "PTP.StatePTP" parameter shows the status information of the PTP positioning.

▶ Request detailed status information on the PTP positioning mode.

This requires that the read value is available on your unit and in the current unit status.

	Object	Req	Six	Index	Data	Description
TxD	35.2 PTP.StatePTP	80h	02h	0023h	0000 0000h	Requesting the status information on PTP positioning

		Res	Ctrl	fb-stat	Data	Description
RxD		80h	xxh	6xx6h	0000 6000h	Acknowledgement in byte 7 and 8: "motion_end" and "set position reached" active.

In the event of a read access to a 16-bit value, the value read is in bytes 7 and 8; the value of bytes 5 and 6 is 0.

6.5 Processing inputs/outputs

Depending on the setting in "Settings.IO_mode", specific inputs and outputs are available to the operator. For further information see the chapter on "Connection to the signal interface" in your manual.

Reading inputs

▶ Read the signal level at input word 0.

	Object	Req	Six	Index	Data	Description
TxD	33.1 I/O.IW0_act	80h	01h	0021h	0000 0000h	Requesting the signal level at input word 0

		Res	Ctrl	fb-stat	Data	Description
RxD		80h	xxh	xxxxh	0000 000Fh	High level at inputs I0..I3

Writing to outputs

▶ Set output Q0 to "high" and outputs Q1..Q4 to "low".

The condition is that "Settings.IO_mode" does not equal "2: inputs/ outputs are assigned with function".

	Object	Req	Six	Index	Data	Description
TxD	34.1 I/O.QW0_act	84h	01h	0022h	0000 0001h	Output Q0=high, outputs Q1..Q4=low

Write accesses are only possible on the free outputs Q0..Q4.

Reading the status of outputs

▶ Determine the instantaneous signal level at the outputs.

	Object	Req	Six	Index	Data	Description
TxD	34.1 I/O.QW0_act	80h	01h	0022h	0000 0000h	Requesting the signal level at output word 0

		Res	Ctrl	fb-stat	Data	Description
RxD		80h	xxh	xxx6h	0000 0036h	Outputs Q1, Q2, Q4 and ACTIVE_CON are set.

In the received data bit0 to bit4 correspond to outputs Q0 to Q4, bit5 to output ACTIVE_CON and bit6 to output TRIGGER/ALARM.

6.6 Use of operating modes

The Twin Line unit can operate in various operating modes. Mode-specific and non-mode-specific processing values are available for configuring the processing sequence.

The operating modes can be determined via specific parameters, the processing status non-mode-specific or mode-specific.

The status of the process or movement can be read from "x_err" and "x_end" in "fb-statusword":

x_err	x_end	Process status
0	0	Process/movement active and no fault
0	1	Process/movement terminated and no fault
1	0	Process/movement active and fault detected
1	1	Process/movement terminated and fault detected

If an asynchronous error occurs during processing, "x_err" is set to 1 (bit15) immediately and the drive is stopped. Because the drive stops x_end is set to bit1 (bit14).

If "2:I/O assigned with function" has been set via "Settings.IO_Mode", input "AUTOM" must equal 1 to enable access to the operating modes.

6.6.1 Point-to-point positioning

Device types that can be used TLC5xx, TLC6xx

A pallet must be moved two stations on an endless conveyor belt at a speed of 200 usr (e. g. 200 rpm). All values are given in user-defined units [usr], because the real values depend on the calibration factor used.

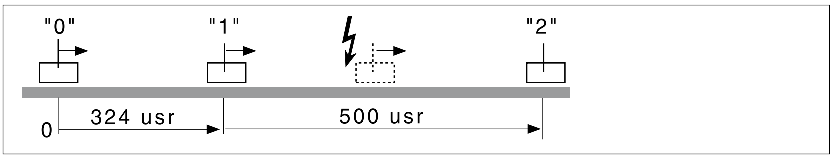


Fig. 6.4 Moving the pallet

The first station is 324 usr (e. g. 324 mm) away from the reference position "0". The station is approached with absolute positioning. The second station is another 500 usr away and is reached by relative positioning.

The movement to position "2" is interrupted by the STOP signal. After correction of the fault the interrupted movement must be resumed and completed.

The following conditions must be met to be able to carry out the task:

- All required function settings must have been made, see page 6-24.
- The 0 position must be defined, see page 6-19.
- No referencing movement is active.

Carrying out absolute positioning

▶ Set the absolute motor movement to position +324 usr with a set speed of 200 usr.

This requires the unit to be in "OperationEnable" status, see page 6-5.

	Object	Req	Six	Index	Data	Description
TxD	35.5 PTP.v_target	84h	05h	0023h	0000 00C8h	Setting the set speed to 200 usr = 00C8h usr

	Object	Req	Six	Index	Data	Description
TxD	35.1 PTP.p_absPTP	04h	01h	0023h	0000 0144h	Start of absolute positioning at 324 usr = 0144h usr

	Res	Ctrl	fb-stat	Data	Description
RxD	00h	x3h	0006h	xxxx xxxxh	Motor movement active

The process status can be taken from "x_end" in fb-statusword. If the process or movement is complete, x_end switches from 0 to 1.

	Res	Ctrl	fb-stat	Data	Description
RxD	00h	2xh	6xx6h	xxxx xxxxh	Motor movement complete

The set speed can be changed during the movement with a write access to "PTP.v_target".

Carrying out relative positioning

▶ Initiate a relative motor movement of +500 usr at the set speed.

This requires the unit to be in "OperationEnable" status, see page 6-5.

	Object	Req	Six	Index	Data	Description
TxD	35.3 PTP.p_relPTP	84h	03h	0023h	0000 01F4h	Start of relative positioning at +500 usr = 01F4h usr

	Res	Ctrl	fb-stat	Data	Description
RxD	80h	x3h	0006h	xxxx xxxxh	Motor movement active

The process status can be taken from "x_end" in fb-statusword. If the process or movement is complete, x_end switches from 0 to 1.

	Res	Ctrl	fb-stat	Data	Description
RxD	80h	23h	6006h	xxxx xxxxh	Motor movement complete

The set speed can be changed during the movement with a write access to "PTP.v_target".

Completing interrupted movement

▶ Complete the movement interrupted by the STOP signal.

The following conditions must be met to be able to resume and complete the interrupted movement:

- PTP positioning was interrupted by STOP.
- The unit is in the "QuickStopActive" status.
- The cause of the fault has been corrected, i.e. the STOP signal is no longer active.

The drive must first be changed to the "OperationEnable" status by "FaultReset", see page 6-35.

	Object	Req	Six	Index	Data	Description
TxD	35.4 PTP.continue	84h	04h	0023h	0000 0000h	Initiating resumption of interrupted positioning operation

	Res	Ctrl	fb-stat	Data	Description
RxD	80h	23h	0006h	xxxx xxxxh	Motor movement active

The process status can be taken from "x_end" in fb-statusword. If the process or movement is complete, x_end switches from 0 to 1.

	Res	Ctrl	fb-stat	Data	Description
RxD	80h	23h	6006h	xxxx xxxxh	Motor movement complete

The set speed can be changed during the movement with a write access to "PTP.v_target".

6.6.2 Speed mode

Device types that can be used TLC5xx, TLC6xx

▶ Set the speed to 2000 usr – we want to monitor that the speed has been reached.

▶ Bring the drive to a standstill by setting speed = 0, and check that the movement has been completed.

- All required function settings must have been made, see page 6-24.
- The unit is in the "OperationEnable" status, see page 6-5
- No referencing movement is active.

Setting the speed:

	Object	Req	Six	Index	Data	Description
TxD	36.1 VEL.velocity	84h	01h	0024h	0000 07D0h	Setting the speed: +2000 usr = 07D0h usr

"x_add_info" in fb-statusword can be used to check whether the set speed has been reached. If the set speed has been reached, x_add_info switches from 0 to 1.

		Res	Ctrl	fb-stat	Data	Description
RxD		80h	x4h	2006h	xxxx xxxxh	Motor movement active, set speed reached

Bringing drive to standstill:

	Object	Req	Six	Index	Data	Description
TxD	36.1 VEL.velocity	04h	01h	0024h	0000 0000h	Setting the speed: 0 usr = 0h usr

The process status can be taken from "x_end" in fb-statusword. If the process or movement is complete, x_end switches from 0 to 1.

		Res	Ctrl	fb-stat	Data	Description
RxD		00h	x4h	6006h	xxxx xxxxh	Motor movement complete

6.6.3 Electronic gear

Device types that can be used TLC5xx, TLC6xx

Processing reference pulses

▶ Apply the gear ratio of 7/5 to pulses arriving at the guidance sensor. Only consider the pulses that are detected after activation of the gear (immediate synchronization).

▶ Correct the gear ratio to 8/5 during operation.

▶ Set the gear to the "Disable" status, and wait for the drive to stop.

- All required function settings must have been made, see page 6-24.
- The unit is in the "OperationEnable" status, see page 6-5
- No referencing movement is active.

Apply gear factor 7/5 to the pulses:

	Object	Req	Six	Index	Data	Description
TxD	38.8 Gear.denGear	84h	08h	0026h	0000 0005h	Gear factor denominator = 5

	Object	Req	Six	Index	Data	Description
TxD	38.7 Gear.numGear	04h	07h	0026h	0000 0007h	Gear factor numerator = 7 Imports the denominator.

	Object	Req	Six	Index	Data	Description
TxD	38.1 Gear.startGear	84h	01h	0026h	0000 0001h	Start gear process with immediate synchronization (commanddata=1)

The gear process is now active.

Correcting the gear factor to 8/5:

	Object	Req	Six	Index	Data	Description
TxD	38.7 Gear.numGear	04h	07h	0026h	0000 0008h	Gear factor numerator = 8 Denominator is retained.

Set gear to "Disable" status and wait until the drive is at a standstill:

	Object	Req	Six	Index	Data	Description
TxD	38.1 Gear.startGear	84h	01h	0026h	0000 0000h	Deactivating gear process (commanddata=0)

		Res	Ctrl	fb-stat	Data	Description
RxD		80h	x5h	0006h	xxxx xxxxh	Motor movement active

The process status can be taken from "x_end" in fb-statusword. If the process or movement is complete, x_end switches from 0 to 1.

	Res	Ctrl	fb-stat	Data	Description
RxD	80h	x5h	4006h	xxxx xxxxh	Motor movement complete

The new gear factor is activated on transfer of the numerator.

The "Gear" parameter group includes various setting options for processing the operating mode.

Overlay of a PTP-offset positioning process

▶ Correct the initial gear position by the relative offset of -100 Inc with the set speed of 200 rpm of the offset positioning process.

- All required function settings must have been made, see page 6-24.
- The unit is in the "OperationEnable" status, see page 6-5
- No referencing movement is active.
- The gear process is active.

	Object	Req	Six	Index	Data	Description
TxD	39.5 Gear.n_tarOffs	84h	03h	0027h	0000 00C8h	Setting the set speed for the offset positioning process to 200 rpm = 00C8h rpm

	Object	Req	Six	Index	Data	Description
TxD	39.3 Gear.p_relOffs	04h	03h	0027h	0000 0064h	Starting the relative offset positioning process at 100 Inc = 64h Inc

	Object	Req	Six	Index	Data	Description
TxD	39.2 Gear.StateOffs	80h	02h	0027h	0000 0000h	Request: acknowledgement of offset positioning

	Res	Ctrl	fb-stat	Data	Description
RxD	80h	x6h	xxx6h	0000 0000h	Acknowledgement in bytes 7 and 8: offset_motion_end = 0, offset set position reached = 0. offset process running.

The process status of the offset positioning process is in the read data in "offset_motion_end". If the offset positioning is complete, "offset_motion_end" switches from 0 to 1.

	Res	Ctrl	fb-stat	Data	Description
RxD	80h	x6h	xxx6h	0000 6000h	Acknowledgement in bytes 7 and 8: offset_motion_end = 1, offset set position reached = 1. offset process complete.

The setpoint speed of the offset movement can be changed during movement with a write access to "Gear.n_tarOffs".

The "Gear" parameter group includes various setting options for processing the PTP offset positioning.

6.6.4 Referencing

The following conditions must be met to be able to run the referencing examples:

- All required function settings must have been made, see page 6-24.
- The unit is in the "OperationEnable" status, see page 6-5
- No referencing movement is active.

Dimension setting

▶ Set the current drive position to the value of 1000 usr. The value acts as a reference point for further movements.

This task requires the drive to be at a standstill.

	Object	Req	Six	Index	Data	Description
TxD	40.3 Home.startSetp	84h	03h	0028h	0000 03E8h	Dimension setting position +1000 usr = 03E8h usr

Dimension setting is run immediately when the unit is called. The success of the operation can be checked with "x_end" and "x_err" in fb-statusword.

	Res	Ctrl	fb-stat	Data	Description
RxD	80h	x2h	4006h	xxxx xxxxh	Dimension setting successfully completed

After successful referencing bit "ref_ok" in controldata is set to 1.

Carrying out a referencing movement

▶ Carry out a referencing movement in a negative direction to the additional reference switch. The search movement to the switch should be run at a speed of 500 usr.

	Object	Req	Six	Index	Data	Description
TxD	40.4 Home.v_Home	84h	04h	0028h	0000 01F4h	Speed for search of reference switch: 500 usr = 01F4h usr

	Object	Req	Six	Index	Data	Description
TxD	40.1 Home.startHome	04h	01h	0028h	0000 0003h	Start referencing movement to the additional reference switch in a negative direction

	Res	Ctrl	fb-stat	Data	Description
RxD	00h	x2h	0006h	xxxx xxxxh	Referencing movement active

The process status can be taken from "x_end" in fb-statusword. If the process or movement is complete, x_end switches from 0 to 1.

	Res	Ctrl	fb-stat	Data	Description
RxD	00h	x2h	4006h	xxxx xxxxh	Referencing movement complete

After successful referencing bit "ref_ok" in controldata is set to 1.

The "Home" parameter group includes various setting options for processing the operating mode.

6.6.5 Manual movement

Two different movement profiles for manual movement are available. You can select the type with "Manual.typeMan".

▶ Initiate a classical manual movement in a positive direction with a **slow** direction of rotation set. Then change the values to run a classical manual movement in a positive direction at a **fast** speed.

▶ Terminate the manual movement and check whether the movement has been ended.

The following conditions must be met to be able to carry out the example:

- All required function settings must have been made, see page 6-24.
- The unit is in the "OperationEnable" status, see page 6-5
- No referencing movement is active.

Carrying out a manual movement and changing the speed:

	Object	Req	Six	Index	Data	Description
TxD	41.1 Manual.startMan	84h	01h	0029h	0000 0001h	Selection: positive direction of rotation, slow speed (Manual.n_slowMan)

	Object	Req	Six	Index	Data	Description
TxD	41.1 Manual.startMan	04h	01h	0029h	0000 0005h	Selection: positive direction of rotation, fast speed (Manual.n_fastMan)

End manual movement and check for standstill:

	Object	Req	Six	Index	Data	Description
TxD	41.1 Manual.startMan	84h	01h	0029h	0000 0000h	Selection: no direction of rotation, i.e. complete manual movement

The process status can be taken from "x_end" in fb-statusword. If the process or movement is complete, x_end switches from 0 to 1.

	Res	Ctrl	fb-stat	Data	Description
RxD	80h	x4h	6006h	xxxx xxxxh	Manual movement complete

The "Manual" parameter group includes various setting options for processing the operating mode.

6.6.6 Data set mode

The movement data for "data set mode" operating mode are stored in the data set memory slots RecoData0..RecoData49. "PTP set" or "VEL set" can be selected with "RecoData0.TypeReco"; the data set mode with the specified type can be initiated with "Reco.startReco".

The following examples describe the sequence for setting up the PTP data set, initiating data set mode and resuming an interrupted data set mode with "Record.continue".

All required settings must have been made via the functions, see page 6-24.

Device types that can be used TLC4xx, TLC6xx

Setting PTP data sets

▶ Set up the data sets on the PTP data set.

▶ Set up data set 15 with the following processing units:

- Dimension system: relative
- set position: 1000 usr
- set speed: 300 usr
- ramp selection: 2
  (settings in "Record.UpRamp2" and "Record.DownRamp2").

This requires all ramp parameters to be set in "Record.UpRamp2" and "Record.DownRamp2".

Setting the data set data to the PTP data set:

	Object	Req	Six	Index	Data	Description
TxD	1000.1 RecoData0.TypeReco	84h	01h	03E8h	0000 0001h	Selection: PTP data set

Processing values in data set 15:

	Object	Req	Six	Index	Data	Description
TxD	1015.2 RecoData15.PosSystem	04h	02h	03F7h	0000 0002h	Data set 15, selection of dimension system: relative

	Object	Req	Six	Index	Data	Description
TxD	1015.3 RecoData15.PosReco	84h	03h	03F7h	0000 03E8h	Data set 15, selection of set position: 1000 usr = 03E8h usr

	Object	Req	Six	Index	Data	Description
TxD	1015.4 RecoData15.VelReco	04h	04h	03F7h	0000 012Ch	Data set 15, selection of set speed: 300 usr = 012Ch usr

	Object	Req	Six	Index	Data	Description
TxD	1015.5 RecoData15.RmpChoice	84h	05h	03F7h	0000 0002h	Data set 15, selection of ramp selection: 2

Initiating PTP data set mode

▶ Activate PTP data set 15 and check the movement sequence.

The following conditions must be met to be able to carry out the example:

- The unit is in the "OperationEnable" status, see page 6-5
- The data set data are configured.

	Object	Req	Six	Index	Data	Description
TxD	45.1 Record.startReco	84h	01h	002Dh	0000 000Fh	Start data set 15

		Res	Ctrl	fb-stat	Data	Description
RxD		80h	x7h	0006h	xxxx xxxxh	Motor movement active

The process status can be taken from "x_end" in fb-statusword. If the process or movement is complete, x_end switches from 0 to 1.

		Res	Ctrl	fb-stat	Data	Description
RxD		80h	27h	6006h	xxxx xxxxh	Motor movement complete

Completing interrupted data set mode

▶ Complete data set mode interrupted by the STOP signal.

The following conditions must be met to be able to resume and complete the interrupted data set mode:

- Data set mode was interrupted by STOP.
- The unit is in the "QuickStopActive" status.
- The cause of the fault has been corrected, i.e. the STOP signal is no longer active.

The drive must first be changed to the "OperationEnable" status by "FaultReset", see page 6-35.

	Object	Req	Six	Index	Data	Description
TxD	45.17 Record.continue	84h	11h	002Dh	0000 0000h	Initiating resumption of the interrupted data set mode

		Res	Ctrl	fb-stat	Data	Description
RxD		80h	x7h	0006h	xxxx xxxxh	Motor movement active

The process status can be taken from "x_end" in fb-statusword. If the process or movement is complete, x_end switches from 0 to 1.

		Res	Ctrl	fb-stat	Data	Description
RxD		80h	x7h	6006h	xxxx xxxxh	Motor movement complete

6.7 Using operating functions

6.7.1 List control

The processing data for the list control function are stored in the list data memory slots L1Data0..L1Data63 (list 1) or L2Data0..L2Data63 (list 2). "Pos-SignalList" or "SpeedList" can be selected with "L1Data0.typeList" or "L2Data0.typeList". A list process is initiated by List.startList.

The following example describes the sequence for setting up a position/ signal list on list 1 and initiation of list processing with monitoring of the processing status.

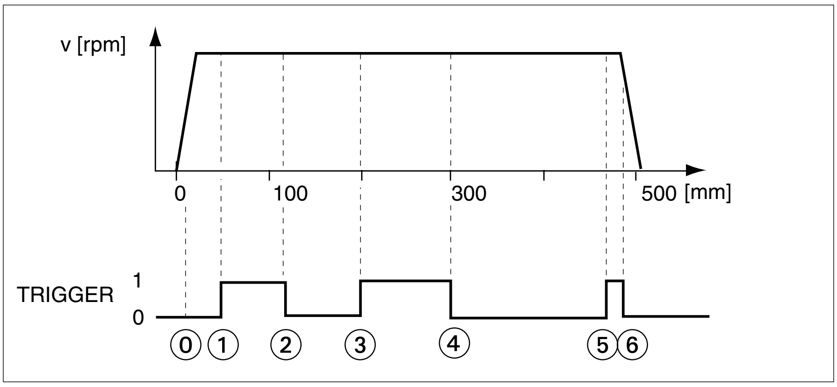


Fig. 6.5 Positioning with position/signal list

Graphics point	List number 1100:x...1163:x	List type 1xxx:1	Position 1xxx:2	Trigger signal 1xxx:3	Speed 1xxx:4
0	1100	1	10	0	0
1	1101	1	50	1	0
2	1102	1	120	0	0
3	1103	1	200	1	0
4	1104	1	300	0	0
5	1105	1	470	1	0
6	1106	1	490	0	0
-	...	...	...	0	0

All the required settings must have been made using the functions described in "Using operating functions".

Setting list data

▶ Create List1 as position/signal list

▶ Set the values in the list. As an example the solution is shown for list entry 0 with the following values:

- comparative position: 10 usr
- trigger signal level: 0

▶ Set the starting and finishing number of the list range:

- starting number = 0
- finishing number = 6

Setting up List1 as a position/signal list:

	Object	Req	Six	Index	Data	Description
TxD	1100.1 L1Data0.typeList	84h	01h	044Ch	0000 0001h	Selection: 1 = position/signal list

Example of list entry 0 with processing values:

	Object	Req	Six	Index	Data	Description
TxD	1100.2 L1Data0.posList1	04h	02h	044Ch	0000 000Ah	List1.entry0: comparative position = 10 usr = 000Ah usr

	Object	Req	Six	Index	Data	Description
TxD	1100.3 L1Data0.sign.List1	84h	03h	044Ch	0000 0000h	List1.entry0: Signal status = 0

The other list items 1..6 are set similarly via index L1Data1..L1Data6.

Setting the starting and finishing number of the list range:

	Object	Req	Six	Index	Data	Description
TxD	44.6 List.bgnList1	04h	06h	002Ch	0000 0000h	Setting the starting number of list 1 to 0

	Object	Req	Six	Index	Data	Description
TxD	44.7 List.endList1	84h	07h	002Ch	0000 0006h	Setting finishing number of list 1 to 6

Activating list control

▶ Activate List1 from the above example and monitor the process sequence.

The following conditions must be met for the example:

- The unit must be in the OperationEnable status, see page 6-5.
- List type, list data, starting and finishing number of the range that is to be processed are set in list 1.

	Object	Req	Six	Index	Data	Description
TxD	44.1 List.startList	84h	01h	002Ch	0000 0001h	Activating list processing 1 = list1

	Object	Req	Six	Index	Data	Description
TxD	44.2 List.stateList	00h	02h	002Ch	0000 0000h	Reading the status of list processing

		Res	Ctrl	fb-stat	Data	Description
RxD		00h	xxh	xxx6h	xxxx 0001h	Acknowledgement in byte 7 and 8: list_err = list_quit = 0 bits 0 and 1: 1 = list1 active list control running

If the process is complete, "list_end" switches from 0 to 1.

		Res	Ctrl	fb-stat	Data	Description
RxD		00h	xxh	xxx6h	0000 4000h	Acknowledgement in byte 7 and 8: list_quit = 1list control completed

The number of the last activated list entry can be queried with "List.actList".

The level of the "TRIGGER" output can be set via "I/O.OutTrig" with list control inactive.

6.7.2 Teach-in

Device types that can be used TLC4xx, TLC5xx

The current absolute position in user-defined units [usr] can be stored in a selected list or data set memory as a position value with the teach-in function.

The following example describes the process for setting up list 1 for teach-in as a position or signal list and how the actual teach-in procedure is run. Execution is monitored.

All the required settings must have been made using the functions described in "Using operating functions".

Preparing teach-in

▶ Set list 1 as the memory for teach-in processes.

▶ Set list 1 as a position/signal list.

	Object	Req	Six	Index	Data	Description
TxD	43.3 Teach.memNrTeac	84h	03h	002Bh	0000 0001h	Selection: 1 = list1

	Object	Req	Six	Index	Data	Description
TxD	1100.1L1Data0.typeList	04h	01h	044Ch	0000 0001h	Selection: 1 = position/signal list

Carrying out teach-in

▶ Carry out teach-in to save the current motor position in list 1 under entry 5. Monitor the processing status.

	Object	Req	Six	Index	Data	Description
TxD	43.1 Teach.storeTeac	84h	01h	002Bh	0000 0005h	Activation of the teach process: Current position value in [usr] is stored in L1Data5.posList1

The teach-in process is run as soon as it is called. If a fault is detected, it is reported as a synchronous fault message.

The processing status of the teach process can be read with "Teach.stateTeac". If processing has been successfully carried out, teach_err=0 and teach_end=1 are returned.

6.7.3 Calibration

Position, speed and acceleration values are saved in the unit in user-defined units [usr]. Calibration converts user-defined units to positioning controller units and vice versa. The calibration factor is set by specifying the numerator and denominator; the value is imported on transfer of the numerator. For more information see "Calibration" in the controller manual.

Values can only be changed when the power amplifier is switched off, e.g. in "ReadyToSwitchOn" status. The values in [usr] are converted to controller values when the power amplifier is activated and the limit values are checked.

Values stored in the unit in [usr] initiate changes in processing values when the associated calibration factor is changed. For example, the safety distance of the referencing movement "Home.p_outHome" must be adjusted if the positioning calibration is changed.

The calibration factor describes the connection between the value in user-defined units [usr] and the value in controller units:

$$\boxed{\text{normalisation factor=}\ \frac{\text{controller value}}{\text{user value}}}$$

Fig. 6.6 Calculation of the calibration factor

value user	calibration factor	value controller
position [usr]	position calibration	motor revolution [rev]
speed [usr]	speed calibration	motor speed [1 rpm]
acceleration/ deceleration [usr]	acceleration calibration	Motor acceleration [1 rev/(min*s)]

Setting positioning calibration

▶ Set positioning calibration so the change of the user position by 1000 usr initiates one motor revolution.

value user = 1000 usr

value control = 1 rev

$$\text{position normalisation} = \frac{\text{position value controller}}{\text{position value user}} = \frac{1\ \text{rev}}{1000\ \text{usr}}$$

Fig. 6.7 Calculation of positioning calibration factor

The power amplifier must be switched off for this task, i.e. the controller is in "ReadyToSwitchOn" status, see page 6-5.

	Object	Req	Six	Index	Data	Description
TxD	29.8 Motion.pNormDen	84h	08h	001Dh	0000 03E8h	Position calibration factor: denominator = 1000 = 03E8h

	Object	Req	Six	Index	Data	Description
TxD	29.7 Motion.pNormNum	04h	07h	001Dh	0000 0001h	Position calibration factor: numerator = 1; results in adoption of numerator and denominator

6.7.4 Fast position value capture

Device types that can be used TLC5xx, TLC6xx

Preparing position capture The current position of the motor in [Inc] must be detected as soon as the signal level at the "CAPTURE1" input changes from 0 to 1.

▶ Connect position capture on channel 1 to the "CAPTURE1" input.

▶ Set the signal level for capture to a 0→1 edge.

All the required settings must have been made using the functions described in "Using operating functions".

	Object	Req	Six	Index	Data	Description
TxD	20.13 Capture.TrigSign	84h	0Dh	0014h	0000 0000h	Selection: bit0..1 = 0, i.e. setting up recording via channel 1, triggered by the "CAPTURE1" input

	Object	Req	Six	Index	Data	Description
TxD	20.15 Capture.TrigLevl	04h	0Fh	0014h	0000 0001h	Selection: bit0 = 1, i.e. setting trigger level to 0→1-edge on channel 1

Starting and monitoring position capture

- ▶ Start capture recording.
- ▶ Monitor the processing status.
- ▶ Determine the value recorded.

The following conditions must be met:

- All required settings must have been made via the functions that are described in the chapter "Using operating functions".
- Position capture must be set up, see previous example.
- The module for the position sensor must be present and connected to the sensor.

	Object	Req	Six	Index	Data	Description
TxD	20.16 Capture.TrigStart	84h	10h	0014h	0000 0001h	Selection: bit0 = 1, i.e. position capture on channel 1

	Object	Req	Six	Index	Data	Description
TxD	20.17 Capture.TrigStat	00h	11h	0014h	0000 0000h	Reading the status of position capture

The processing status must be monitored cyclically; the status of the process is shown in the returned data.

		Res	Ctrl	fb-stat	Data	Description
RxD		00h	xxh	xxxxh	xxxx 0001h	Acknowledgement in byte 7 and 8: bit0 = 1, i.e. position detection carried out on channel 1

The position value saved in [Inc] can now be read:

	Object	Req	Six	Index	Data	Description
TxD	20.18 Capture.TrigPact1	80h	12h	0014h	0000 0000h	Reading the saved position value

		Res	Ctrl	fb-stat	Data	Description
RxD		80h	xxh	xxxxh	xxxx 1234h	Position value in [Inc] is given in the returned data: 1234h = 4660 Inc.

6.8 Examples of fault processing

You will find in-depth information on fault handling infield bus mode in the chapter on "Error handling" on page 7-1.

6.8.1 Synchronous errors

Synchronous faults only occur in response to a command. When a command is sent, it is immediately checked for whether it can be correctly executed. If this is not the case, the unit returns an error number in response to the command and cmderr=1 is set in the "responsedata" byte. This action does not change the unit status.

Creating a synchronous fault

▶ Execute a write access to a non-existent parameter (index: 0, subindex: FFh).

This requires the unit to be in "OperationEnable" status, see page 6-5.

	Object	Req	Six	Index	Data	Description
TxD	0.255 to non-existent parameter	84h	FFh	0000h	xxxx xxxxh	Write access to a non-existent parameter

	Res	Ctrl	fb-stat	Data	Description
RxD	C0h	xxh	6xx6h	0000 1003h	cmderr=1 The error number (errnum) is contained in bytes 7 and 8.

6.8.2 Asynchronous errors

Asynchronous faults occur independently of transmitted commands. If the external and internal monitoring signals detect a fault, the unit switches to a fault status. The unit status changes depending on the fault class. The fault status can be queried or, in the case of cyclic transmission, is in fb-statusword.

Internal monitoring signals

▶ The 7-segment display on your unit shows a flashing 1, i. e. there is a fault. Evaluate the cause of the fault.

The following conditions must be met to generate the fault:

- The unit is in the "OperationEnable" status, see page 6-5
- The 220 V power supply is switched off. If the capacitors are fully discharged, the 7-segment display shows a flashing 1, which indicates undervoltage.

	Res	Ctrl	fb-stat	Data	Description
RxD	00h	xxh	E029h	xxxx xxxxh	In fb-statusword: x_err=1, x_end=1, FltSig=1, cos=9: "Fault"

The cause of the fault can be evaluated in detail with the internal monitoring signals.

	Object	Req	Six	Index	Data	Description
TxD	28.18 Status.FltSig_SR	80h	12h	001Ch	xxxx xxxxh	

	Res	Ctrl	fb-stat	Data	Description
RxD	80h	xxh	E029h	0000 0002h	Internal monitoring signal: bit1 active, i. e. cause of fault "DC link undervoltage Lim1"

	Object	Req	Six	Index	Data	Description
TxD	29.34 Status.IntSigSR	00h	22h	001Dh	xxxx xxxxh	

	Res	Ctrl	fb-stat	Data	Description
RxD	00h	xxh	E029h	0000 8000h	Internal monitoring signal: bit15 active, i.e. "power amplifier not active"

If bit7 in fb-statusword is active, the warning messages can be read with "Status.FltSig_SR" and "Status.IntSigSR".

If the drive is at a standstill, the interruption status can be canceled with "Fault reset" – see page 6-35.

External monitoring signals

▶ A positioning operation was interrupted by a light barrier at the "STOP" input. Evaluate the cause of the fault.

The following conditions must be met to generate the fault:

- The "STOP" monitoring signal is enabled with "Settings.SignEnabl"; STOP is activated.
- The unit is in the "QuickStopActive" status.

	Res	Ctrl	fb-stat	Data	Description
RxD	00h	xxh	E047h	xxxx xxxxh	In fb-statusword: x_err=1, x_end=1, SignSr=1, cos=7: "QuickStopActive"

The cause of the fault can be evaluated in detail with the internal monitoring signals.

	Object	Req	Six	Index	Data	Description
TxD	28.15 Status.Sign_SR	80h	0Fh	001Ch	xxxx xxxxh	

	Res	Ctrl	fb-stat	Data	Description
RxD	80h	xxh	E047h	0000 0004h	Internal monitoring signal: bit2 active, i.e. "STOP"

If the drive is at a standstill, the interruption status can be canceled with "Fault reset" – see page 6-35.

6.8.3 Other faults

If "x_err" (bit15) in fb-statusword is activated but neither "SignSr" (bit6) nor "FltSig" (bit5) is set, an internal controller fault has been detected, which can only be read via "Status.StopFault" (32:7) as an error number.

Determining the error number

Faults that force the unit out of the OperationEnable status are not only entered into the bit line for external and/or internal monitoring signals but also into the error memory. You can access the cause of the last status change directly.

▶ Read the cause of the last interruption from the unit error memory.

This requires that the unit was interrupted via the "STOP" input.

	Object	Req	Six	Index	Data	Description
TxD	28.15 Status.StopFault	80h	07h	0020h	0000 0000h	Request: cause of last interruption

		Res	Ctrl	fb-stat	Data	Description
RxD		80h	xxh	Cx47h	xxxx 1846h	In fb-statusword: x_err=1, x_end=1, SignSr=1. error number 1846h in bytes 7 and 8

If the faults have been reset or the 24 V power supply switched off and on again, the cause of the last interruption is deleted.

If more than one fault was detected, only the fault that resulted in the drive being interrupted, i.e. leaving the "OperationEnable" status, is saved as the cause of the interruption. Any subsequently occurring faults are saved in the standard error memory in order of occurrence.

6.8.4 Resetting faults (FaultReset)

The "Quick-Stop" or "Fault" fault states can be exited using "FaultReset" if the cause of the fault is no longer active, otherwise the fault status is retained.

Once "FaultReset" has been successfully performed, the error number of the last interruption is deleted. A "FaultReset" is only possible if the drive is at a standstill (x_end = 1).

FaultReset ▶ Reset the interruption to the movement caused by the "STOP" input.

The following conditions must be met to be able to carry out the task:

- As a result of activating the STOP input the unit is in the "QuickStopActive" status.
- The cause of the fault is no longer active, i.e. the STOP input is deactivated.
- The drive is stopped (x_end = 1).

	Res	Ctrl	fb-stat	Data	Description
RxD	00h	xxh	E047h	xxxx xxxxh	In fb-statusword: x_err=1, x_end=1, SignSr=1, cos=7: "QuickStopActive"

	Object	Req	Six	Index	Data	Description
TxD	28.1 Commands.driveCtrl	84h	01h	001Ch	0000 0008h	Request: set bit 3 "FaultReset"

If the fault has been successfully reset, the fault status is terminated. The unit enters "OperationEnable" status.

	Res	Ctrl	fb-stat	Data	Description
RxD	80h	xxh	6xx6h	xxxx xxxxh	In fb-statusword: x_err=0, cos=6: "OperationEnable"

This action will not delete the fault entry in the error memory.

6.8.5 Reading and deleting the error memory

All error messages are entered into the unit error memory in order of occurrence. The error memory can contain a maximum of 20 entries.

Apart from the error number, the following information can be read from the error memory:

- Fault class
- time of fault since power amplifier was switched on
- number of AMPON cycles (activation of power amplifier)
- additional information on fault

ErrMem0 contains the information on the oldest fault entry, ErrMem1 the information on the second oldest etc. If an entry in the memory is empty, the error number = 0.

Faults of fault class 2 and higher are stored in non-volatile memory, i.e. they are not deleted when the unit is switched off.

A special command is available to delete the error memory.

Reading the error memory

The oldest fault entry is a movement interruption caused by the "STOP" input. The unit is in the "QuickStopActive" status.

▶ Read all information on the oldest fault entry from the error memory.

	Object	Req	Six	Index	Data	Description
TxD	900.1 ErrMem0.ErrNum	80h	01h	0384h	0000 0000h	Request: error number in ErrMem0

		Res	Ctrl	fb-stat	Data	Description
RxD		80h	xxh	Cx47h	0000 1846h	In fb-statusword: x_err=1, x_end=1, SignSr=1. error number 1846h in bytes 7 and 8

	Object	Req	Six	Index	Data	Description
TxD	900.2 ErrMem0.Class	00h	02h	0384h	0000 0000h	Request: Fault class

		Res	Ctrl	fb-stat	Data	Description
RxD		00h	xxh	Cx47h	0000 0001h	Fault class = 1 in byte 7 and 8

	Object	Req	Six	Index	Data	Description
TxD	900.3 ErrMem0.Time	80h	03h	0384h	0000 0000h	Request: time of fault since power amplifier was switched on

		Res	Ctrl	fb-stat	Data	Description
RxD		80h	xxh	Cx47h	0000 035Dh	time of fault 035Dh = 861 sec

	Object	Req	Six	Index	Data	Description
TxD	900.4 ErrMem0.AmpOnCnt	00h	04h	0384h	0000 0000h	Request: number of power amplifier switch-on cycles

		Res	Ctrl	fb-stat	Data	Description
RxD		00h	xxh	Cx47h	0000 006Eh	switch-on cycles 6Eh = 110 cycles

	Object	Req	Six	Index	Data	Description
TxD	900.5 ErrMem0.ErrQual	80h	05h	0384h	0000 0000h	Request: additional information for assessing fault

		Res	Ctrl	fb-stat	Data	Description
RxD		80h	xxh	Cx47h	0000 0000h	value = 0; no additional information available

Access to the other fault entries ErrMem1 to ErrMem19 is similar, and the index for the parameters must be changed accordingly; e.g. ErrMem5, index = 905 = 389h

Deleting the error memory

▶ Delete all entries in the error memory

	Object	Req	Six	Index	Data	Description
TxD	32.2 Commands.del_err	80h	02h	0020h	0000 0000h	Request: delete contents of error memory

The entries in the error memory are deleted regardless of whether the cause of the fault is still active.

7 Error handling

7.1 Error messages

The master device receives error messages with the received data during network operation. A distinction is made between reports of

- synchronous errors
- asynchronous errors

The master receives the message regarding a synchronous error directly from the controller as a response if the transmitted command could not be executed. A synchronous error is detected by the "cmderr" bit.

Asynchronous errors are reported by the monitoring devices in the controller as soon as a fault in the device occurs. The master continuously monitors the status word "fb_statusword" to detect asynchronous errors.

The controller supplies the status information cyclically in bus frequency.

7.2 Synchronous error

A synchronous error is evaluated with the "cmderr" command error bit in the first byte of the received data:

- "cmderr"=0: command has been successfully carried out
- "cmderr"=1: error has occurred

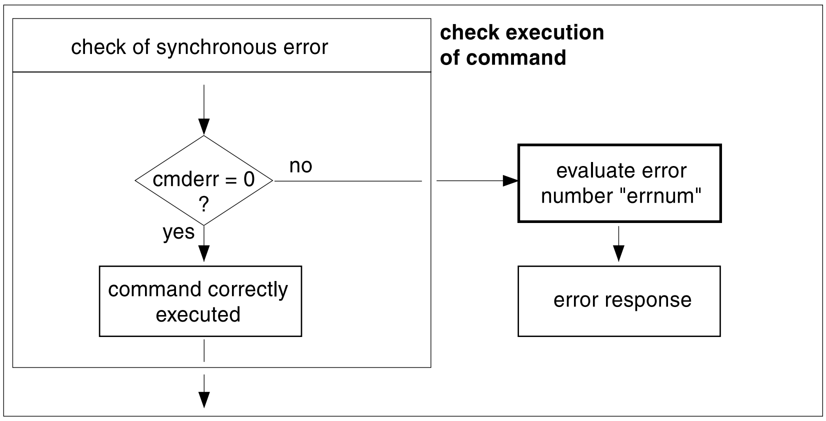


Fig. 7.1 Evaluation of synchronous errors

Causes of errors

Possible causes of a synchronous error are:

- unknown command, syntax error or incorrect transmitted data frame
- parameter value outside the permissible value range
- illegal action or control command during a running process
- error while executing an action or control command.

"cmderr" is only valid when the command has been acknowledged The controller returns an error number "errnum" in bytes 7 and 8 of the received data. The cause of the error can be determined from this number.

The table with the error numbers is in the manual in the chapter on diagnostics and troubleshooting.

7.3 Asynchronous errors

Error bits must be monitored in the "fb_statusword" status word to detect an asynchronous error.

- Bit15, "x_err": Error status during process, evaluate cause via bit5 and bit6
- Bit7, 'warning': controller warning message
  e. g. I^2T power amplifier fault.
- Bit6, 'Sign_SR': message from external monitoring signal,
  e. g. movement interrupted by STOP input
- Bit5, 'FltSig': message from internal monitoring signal
  e. g. overtemperature power amplifier.

Signal status '1' indicates an error or warning message.

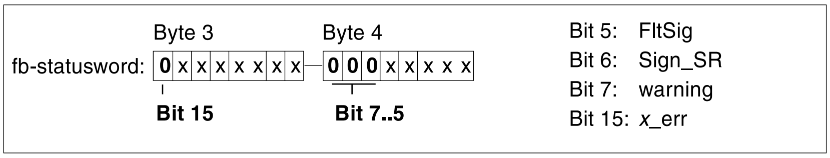


Fig. 7.2 Status word for evaluating asynchronous errors

Warning message

In a warning message the movement command processing continues and the error information is input to the "Status.FltSig_SR" (28:18) or "Status.IntSigSr" parameter (29:34).

Error message In a warning message the movement command processing continues and the error information is input to the "Status.FltSig_SR" (28:18) or "Status.IntSigSr" parameter (29:34).

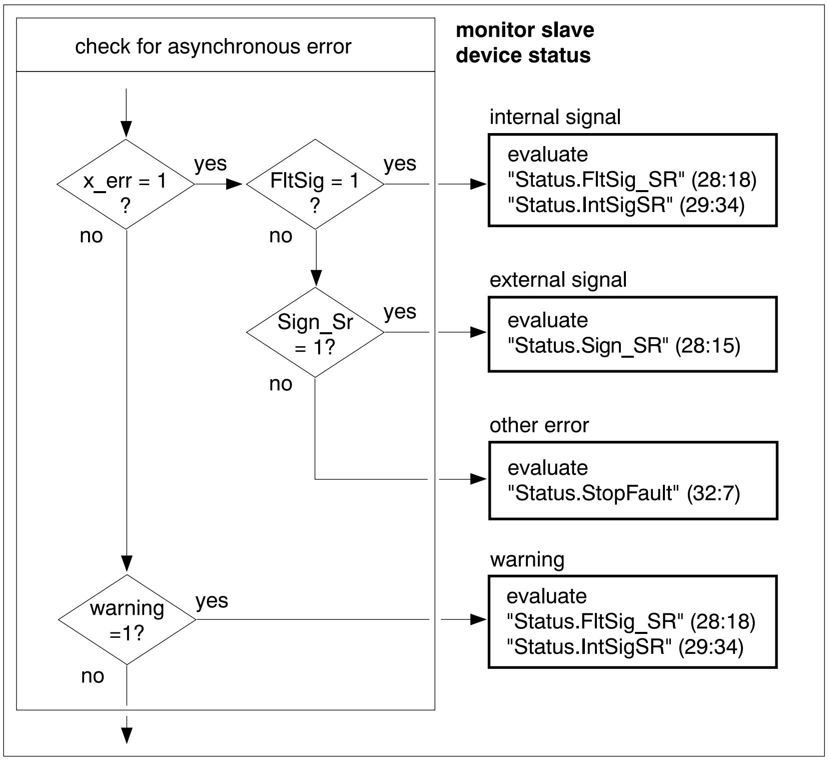


Fig. 7.3 Evaluation of asynchronous errors

Parameters, fault classes and troubleshooting actions are described in the controller manual in the chapter on diagnostics and troubleshooting.

8 Service

8.1 Service address

Contact your local dealer with any questions or problems. Your dealer will be happy to give you the name of a customer service outlet in your area.

9 Accessories

9.1 List of accessories

Accessories for the positioning controller are:

Qty.	Designation	Order no.
1	TL CT operating software with online documentation on data carrier, German	6250 1101 803
1	RS232 programming cable, 5 m; RS232 programming cable, 10 m	6250 1441 050 6250 1441 100
1	HMI hand-held operating unit with manual	9844 1113 091
1	matching TL HMI cable	62501442 yyy [1)]
1	Network cable for Profibus-DP	Siemens

1) Cable length yyy: 005, 015, 030, 050: 0.5 m, 1.5 m, 3 m, 5 m.

Index