LXM32M

Ethernet TCP/IP Module (Protocol Modbus TCP)

User Guide

Original instructions

0198441113843.01
06/2021

Modbus

www.se.com

Schneider Electric

Legal Information

The Schneider Electric brand and any trademarks of Schneider Electric SE and its subsidiaries referred to in this guide are the property of Schneider Electric SE or its subsidiaries. All other brands may be trademarks of their respective owners.

This guide and its content are protected under applicable copyright laws and furnished for informational use only. No part of this guide may be reproduced or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), for any purpose, without the prior written permission of Schneider Electric.

Schneider Electric does not grant any right or license for commercial use of the guide or its content, except for a non-exclusive and personal license to consult it on an "as is" basis. Schneider Electric products and equipment should be installed, operated, serviced, and maintained only by qualified personnel.

As standards, specifications, and designs change from time to time, information contained in this guide may be subject to change without notice.

To the extent permitted by applicable law, no responsibility or liability is assumed by Schneider Electric and its subsidiaries for any errors or omissions in the informational content of this material or consequences arising out of or resulting from the use of the information contained herein.
© 2021 Schneider Electric. All Rights Reserved.

Table of Contents

Safety Information

Important Information

Read these instructions carefully, and look at the equipment to become familiar with the device before trying to install, operate, service, or maintain it. The following special messages may appear throughout this documentation or on the equipment to warn of potential hazards or to call attention to information that clarifies or simplifies a procedure.

The addition of this symbol to a “Danger” or “Warning” safety label indicates that an electrical hazard exists which will result in personal injury if the instructions are not followed.

This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

⚠ DANGER

DANGER indicates a hazardous situation which, if not avoided, **will result in** death or serious injury.

⚠ WARNING

WARNING indicates a hazardous situation which, if not avoided, **could result in** death or serious injury.

⚠ CAUTION

CAUTION indicates a hazardous situation which, if not avoided, **could result** in minor or moderate injury.

NOTICE

NOTICE is used to address practices not related to physical injury.

Please Note

Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material.

A qualified person is one who has skills and knowledge related to the construction and operation of electrical equipment and its installation, and has received safety training to recognize and avoid the hazards involved.

Qualification of Personnel

Only appropriately trained persons who are familiar with and understand the contents of this manual and all other pertinent product documentation are authorized to work on and with this product. These persons must have sufficient technical training, knowledge and experience and be able to foresee and detect potential hazards that may be caused by using the product, by modifying the settings and by the mechanical, electrical and electronic equipment of the entire system in which the product is used.

The qualified person must be able to detect possible hazards that may arise from parameterization, modifying parameter values and generally from mechanical, electrical, or electronic equipment.

The qualified person must be familiar with the standards, provisions, and regulations for the prevention of industrial accidents, which they must observe when designing and implementing the system.

Intended Use

The products described or affected by this document are, along with software, accessories and options, servo-drive systems for three-phase servo motors.

The products are intended for industrial use according to the instructions, directions, examples, and safety information contained in the present user guide and other supporting documentation.

The product may only be used in compliance with all applicable safety regulations and directives, the specified requirements and the technical data.

Prior to using the products, you must perform a risk assessment in view of the planned application. Based on the results, the appropriate safety-related measures must be implemented.

Since the products are used as components in an overall machine or process, you must ensure the safety of persons by means of the design of this overall machine or process.

Operate the products only with the specified cables and accessories. Use only genuine accessories and spare parts.

Any use other than the use explicitly permitted as described herein is prohibited and may result in unanticipated hazards.

About the Book

Document Scope

The information provided in this user guide supplements the user guide of the servo drive LXM32M.

The functions described in this user guide are only intended for use with the associated product. You must read and understand the appropriate user guide of the drive.

Validity Note

This user guide applies to the module Ethernet TCP/IP (protocol Modbus-TCP) for the servo drive LXM32M, module identification ETH (VW3A3616).

For product compliance and environmental information (RoHS, REACH, PEP, EOLI, etc.), go to www.se.com/ww/en/work/support/green-premium/.

The characteristics that are described in the present document, as well as those described in the documents included in the Related Documents section below, can be found online. To access the information online, go to the Schneider Electric home page www.se.com/ww/en/download/.

The characteristics that are described in the present document should be the same as those characteristics that appear online. In line with our policy of constant improvement, we may revise content over time to improve clarity and accuracy. If you see a difference between the document and online information, use the online information as your reference.

Related Documents

Title of documentation	Reference number
LXM32M - Ethernet TCP/IP Module (Protocol Modbus-TCP) - User Guide (this user guide)	0198441113843 (eng) 0198441113844 (fre) 0198441113842 (ger)
Lexium 32M - Servo Drive - User Guide	0198441113767 (eng) 0198441113768 (fre) 0198441113766 (ger) 0198441113770 (spa) 0198441113769 (ita) 0198441113771 (chi)

Product Related Information

⚠WARNING

LOSS OF CONTROL

- The designer of any control scheme must consider the potential failure modes of control paths and, for certain critical control functions, provide a means to achieve a safe state during and after a path failure. Examples of critical control functions are emergency stop and overtravel stop, power outage and restart.
- Separate or redundant control paths must be provided for critical control functions.
- System control paths may include communication links. Consideration must be given to the implications of unanticipated transmission delays or failures of the link.
- Observe all accident prevention regulations and local safety guidelines.[1]
- Each implementation of this equipment must be individually and thoroughly tested for proper operation before being placed into service.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

[1] For additional information, refer to NEMA ICS 1.1 (latest edition), “Safety Guidelines for the Application, Installation, and Maintenance of Solid State Control” and to NEMA ICS 7.1 (latest edition), “Safety Standards for Construction and Guide for Selection, Installation and Operation of Adjustable-Speed Drive Systems” or their equivalent governing your particular location.

For reasons of Internet security, for those devices that have a native Ethernet connection, TCP/IP forwarding is disabled by default. Therefore, you must manually enable TCP/IP forwarding. However, doing so may expose your network to possible cyberattacks if you do not take additional measures to protect your enterprise. In addition, you may be subject to laws and regulations concerning cybersecurity.

⚠WARNING

UNAUTHENTICATED ACCESS AND SUBSEQUENT NETWORK INTRUSION

- Observe and respect any and all pertinent national, regional and local cybersecurity and/or personal data laws and regulations when enabling TCP/IP forwarding on an industrial network.
- Isolate your industrial network from other networks inside your company.
- Protect any network against unintended access by using firewalls, VPN, or other, proven security measures.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Consult the Schneider Electric Cybersecurity Best Practices for additional information.

Use the latest firmware version. Visit https://www.se.com or contact your Schneider Electric representative for information on firmware updates that may involve Ethernet connections.

Terminology Derived from Standards

The technical terms, terminology, symbols and the corresponding descriptions in this manual, or that appear in or on the products themselves, are generally derived from the terms or definitions of international standards.

In the area of functional safety systems, drives and general automation, this may include, but is not limited to, terms such as *safety*, *safety function*, *safe state*, *fault*, *fault reset*, *malfunction*, *failure*, *error*, *error message*, *dangerous*, etc.

Among others, these standards include:

Standard	Description
IEC 61131-2:2007	Programmable controllers, part 2: Equipment requirements and tests.
ISO 13849-1:2015	Safety of machinery: Safety related parts of control systems. General principles for design.
EN 61496-1:2013	Safety of machinery: Electro-sensitive protective equipment. Part 1: General requirements and tests.
ISO 12100:2010	Safety of machinery - General principles for design - Risk assessment and risk reduction
EN 60204-1:2006	Safety of machinery - Electrical equipment of machines - Part 1: General requirements
ISO 14119:2013	Safety of machinery - Interlocking devices associated with guards - Principles for design and selection
ISO 13850:2015	Safety of machinery - Emergency stop - Principles for design
IEC 62061:2015	Safety of machinery - Functional safety of safety-related electrical, electronic, and electronic programmable control systems
IEC 61508-1:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems: General requirements.
IEC 61508-2:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems: Requirements for electrical/electronic/programmable electronic safety-related systems.
IEC 61508-3:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems: Software requirements.
IEC 61784-3:2016	Industrial communication networks - Profiles - Part 3: Functional safety fieldbuses - General rules and profile definitions.
2006/42/EC	Machinery Directive
2014/30/EU	Electromagnetic Compatibility Directive
2014/35/EU	Low Voltage Directive

In addition, terms used in the present document may tangentially be used as they are derived from other standards such as:

Standard	Description
IEC 60034 series	Rotating electrical machines
IEC 61800 series	Adjustable speed electrical power drive systems
IEC 61158 series	Digital data communications for measurement and control – Fieldbus for use in industrial control systems

Finally, the term *zone of operation* may be used in conjunction with the description of specific hazards, and is defined as it is for a *hazard zone* or *danger zone* in the *Machinery Directive* (*2006/42/EC*) and *ISO 12100:2010*.

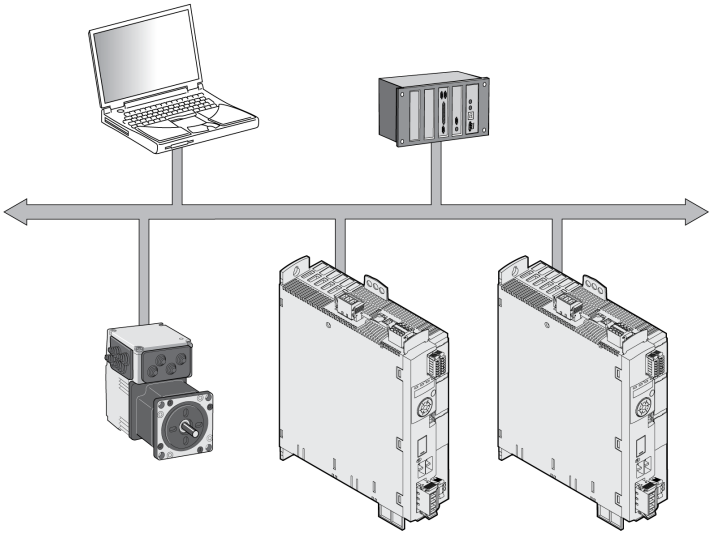
NOTE: The aforementioned standards may or may not apply to the specific products cited in the present documentation. For more information concerning the individual standards applicable to the products described herein, see the characteristics tables for those product references.

Introduction

Fieldbus Devices on the Modbus TCP Network

Overview

Different products with a Modbus TCP interface can be operated via a fieldbus. Modbus TCP is a fieldbus that allows multiple devices to be networked.



Features

The product supports the following functions via Modbus TCP:

- Automatic IP address assignment via BOOTP/DHCP or manual IP address
- Commissioning via commissioning software
- Reading and writing parameters
- Controlling the drive with or without motion libraries
- Monitoring inputs and outputs
- Diagnostics and monitoring functions

Basics

The information contained in this chapter provides a general overview of the various protocols of the fieldbus as it applies to the equipment in the present document. It is not intended as a thorough treatment of the subject, nor is it a sufficient basis to design and deploy a fieldbus network in any given application.

The following information is intended to be consulted in an as needed, as is basis. Only appropriately trained persons who are familiar with and have the education and training necessary to understand the contents of this information, as well as all other pertinent product documentation, are authorized to work on and with this equipment.

Modbus TCP Fieldbus

Modbus TCP Technology

Function Principle

Modbus TCP is an Ethernet fieldbus. Modbus TCP describes the transmission of the Modbus protocol via the Ethernet interface and the TCP/IP transport and network layers.

The Modbus TCP client (master) connects to the Modbus TCP server (slave). Once the connection is established, the client sends Modbus requests to the server. These requests are processed by the server. The result is returned to the client as a Modbus response.

The Modbus TCP services are identical to the Modbus RTU services.

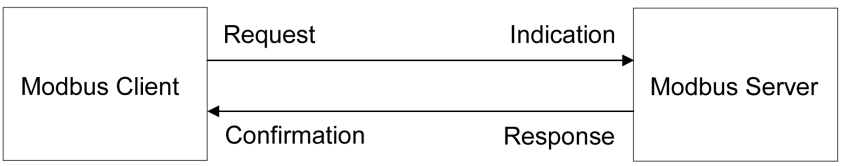
Bus Topology

Modbus TCP allows for the use of hubs or switches. Use switches in the case of high bus loads with many devices.

The maximum length of a segment is 100 m (328 ft). A segment consists of devices and hubs. A network can be subdivided into several segments by means of gateways or switches. For a fast bus cycle, keep the cables short and use a star topology.

The transmission rate is 10 or 100 MBit/s in half-duplex mode. If switches are used, transmission is also possible in full duplex mode.

Client-Server Model



The Modbus messaging service implements client-server communication between devices connected by means of a TCP/IP network. Modbus TCP does not use an object dictionary.

The client-server model is based on four types of messages:

- **Request**: Message sent by the client to initiate a transaction.
- **Indication**: Request as received by the server.
- **Response**: Response message to the request sent by the server.
- **Confirmation**: Response as received by the client.

A communication cycle consists of the request from the client (request from the fieldbus master) and a response from the server (response from the fieldbus slave). Modbus request and Modbus response have the same structure. If an error is detected on receipt of the Modbus request or if the slave cannot execute the action, the slave sends an error message in the Modbus response.

The drive analyzes the Modus requests received. Depending on the Modbus request, the drive triggers actions or provides requested data.

Network Service SNMP

The Network Management System can exchange data with SNMP devices. The tasks of the network management system comprise monitoring, control, and configuration of network components as well as error detection and error messaging.

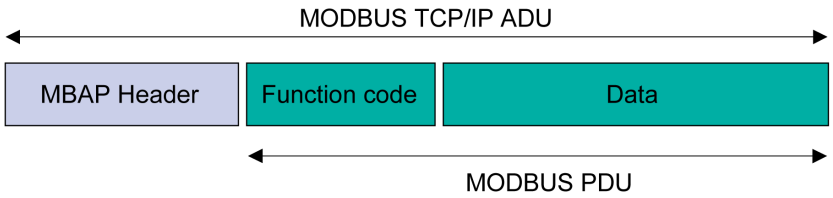
The product supports SNMP version 1.0. An SNMP agent must be used to monitor a network with SNMP.

Modbus TCP Protocol

Overview

The Modbus protocol defines a Modbus PDU (Protocol Data Unit) which is independent of the underlying communication layers. This Modbus PDU consists of the fields “Function Code” and “Data”. Depending on the mapping to the different network protocols, the Modbus PDU is extended by additional fields in the Modbus ADU (Application Data Unit). The Modbus PDU and the Modbus ADU constitute the Modbus message, also referred to as “Frame”.

Structure of a Modbus message



The “Function Code” of a message specifies the Modbus service to be triggered. The “Data” field can contain additional information, depending on the “Function Code”.

Due to the encapsulation of “Function Code” and “Data” in the Modbus PDU, different Modbus versions can use same Modbus services and object model.

The maximum size of a Modbus ADU is 260 bytes. The size of an embedded Modbus PDU is 253 bytes.

Modbus Application Protocol (MBAP) Header

The MBAP header contains the information allowing the recipient to uniquely identify a message.

The MBAP header has a length of seven bytes and contains the following fields:

Field	Length	Description
Transaction Identifier	2 bytes	Identification of a Modbus request or Modbus response.
Protocol Identifier	2 bytes	Value 0 means Modbus protocol.
Length	2 bytes	Byte counter for the subsequent fields (“Unit Identifier”, “Function Code” and "Data").
Unit Identifier	1 byte	Identification of a slave connected to another bus via a serial line.

Modbus TCP Communication

Connection management

Establishing of a Connection

The Modbus TCP server allows for TCP connections via the default port 502. A client can establish a new connection via this port. If the client is to exchange data with a remote server, a new client connection via remote port 502 must be established.

Closing a Connection

After the Modbus communication between the client and a server is finished, the client causes the connection used to be closed.

The server does not close the connection under normal circumstances.

However, when errors are detected, the server closes the connection, for example:

- Communication error detected
- Communication inactivity
- Maximum number of connections reached

The product can manage up to 8 TCP connections. If an attempt is made to establish a further connection beyond this maximum, the oldest unused connection is closed. If it is impossible to close the oldest unused connection, the new connection cannot be established.

Modbus Response to a Modbus Request

Overview

The Modus server generates a Modbus response after having processed a Modbus request.

Two types of Modbus responses are possible:

- Positive Modbus response
  - The "Function Code" in the Modbus response corresponds to the "Function Code" in the Modbus request.
- Negative Modbus response
  - The client receives pertinent information on error detection during processing;
  - The "Function Code" in the Modbus response corresponds to the "Function Code" in the Modbus request + 80_h.
  - The "Exception Code" indicates the cause of the error.

If a syntactically incorrect Modbus PDU (Protocol Data Unit) is transmitted, the connection is terminated. In the case of other errors, a negative Modbus response is sent.

Exception Code	Modbus name (as per Modbus specifications)	Description
01	Illegal Function	The "Function Code" cannot be processed by the server.
02	Illegal Data Address	Depends on the Modbus request
03	Illegal Data Value	Depends on the Modbus request
04	Server Failure	The server was unable to properly process the request. You can use Function Code 8 to read the vendor-specific error code.

Exception Code	Modbus name (as per Modbus specifications)	Description
05	Acknowledge	The server has accepted the Modbus request. However, the execution takes a relatively long time. The server therefore only returns an acknowledgement confirming receipt of the service request.
06	Server Busy	The server was unable to accept the Modbus request. It is the responsibility of the application on the client to determine whether and when to re-send the request.
0A	Gateway Problem	The gateway path is unavailable.
0B	Gateway Problem	The targeted device does not respond. This condition is detected by the gateway.

Reading and Writing Parameters

Overview

Parameters are processed as 32 bit values. 16 bit values must also be processed as 32 bit values. Two consecutive 16 bit parameters must be read or written to process a 16 bit parameter. The Modbus address of the first 16-bit parameter must be used.

If several consecutive parameters are to be processed, a single Modbus command with the corresponding Modbus address and the length indication is sufficient.

NOTE: This does not apply to reading and writing parameters with addresses in the range from 17408 (4400 hex) to 17663 (44FF hex).

In this range, only a single parameter can be addressed with one Modbus command.

Example

Reading the parameter *CTRL1_KPp* "Position controller P gain".

Modbus address: 4614

When the parameter *CTRL1_KPp* with the Modbus parameter address 4614 and length 2 is read, the two parameter addresses 4614 and 4615 are read. Result:

Address	Value
4614	0000 hex
4615	00C8 hex

I/O Scanning to "Drive Profile Lexium"

Overview

I/O scanning is used for cyclic interchange of data between master and slave.

I/O scanning must be configured on the master. The master can use 2 different approaches for I/O scanning:

- "Function Code" 23 (17_h), Read-Write Multiple Registers
- "Function Code" 3 (03_h), Read Multiple Registers and "Function Code" 16 (10_h), Write Multiple Registers

  The read value is 0 until the first write command is executed.

Settings

The following setting must be made on the master before you can use I/O scanning:

- The "Unit Identifier" is 255.
- The Modbus parameter address is 0.
- The data length is 13.

  In addition, you can use up to 3 mappable parameters. If these parameters are used, the data length changes to 15, 17 or 19.

The Modbus addresses for I/O scanning do not differ from the addresses for normal Modbus access.

Output - Input

Output and input refer to the direction of data transmission from the perspective of the master.

- Output: Commands from the master to the slave
- Input: Status messages from the slave to the master

I/O Scanning - Output

Overview

The table below shows the structure of the cyclic data for the commands from the master to the product. See the user guide of the drive for a description of the parameters.

Byte	Name	Parameter address
0 ... 3	PCTRLms	-
4 ... 7	PVms	-
8 ... 9	dmControl	-
10 ... 13	RefA32	-
14 ... 17	RefB32	-
18 ... 21	Ramp_v_acc	Parameter *Ramp_v_acc*, Modbus 1556
22 ... 25	Ramp_v_dec	Parameter *Ramp_v_dec*, Modbus 1558
26 ... 29	EthOptMapOut1	Parameter *EthOptMapOut1*, Modbus 17500
30 ... 33	EthOptMapOut2	Parameter *EthOptMapOut2*, Modbus 17502
34 ... 37	EthOptMapOut3	Parameter *EthOptMapOut3*, Modbus 17504

Double Words “PCTRLms” and “PVms”

The two double words "PCTRLms" and "PVms" are used to read and write parameters, see I/O Scanning - Parameter Channel, page 18.

Word “dmControl”

The word "dmControl" is used to set the operating state and the operating mode.

See Changing the Operating State via Fieldbus, page 41 and Starting and Changing an Operating Mode, page 43 for a detailed description of the bits.

Double Words “RefA32” and “RefB32”

The two double words "RefA32" and "RefB32" are used to set two operating mode-specific values. The meaning is described in the sections on the individual operating modes.

Double Words “Ramp_v_acc” and “Ramp_v_dec”

The double words "Ramp_v_acc" and "Ramp_v_dec" are used to set the acceleration and the deceleration. They correspond to the parameters of the same name.

Double Words “EthOptMapOut1 ... EthOptMapOut3”

The double words EthOptMapOut1 ... EthOptMapOut3 contain selectable parameters, see Setting the Mapping for I/O Scanning, page 37.

I/O Scanning - Input

Overview

The table below shows the structure of the cyclic data for the status messages from the product to the master. See the user guide for a description of the parameters.

Byte	Name	Parameter address
0 ... 3	PCTRLsm	-
4 ... 7	PVsm	-
8 ... 9	driveStat	-
10 ... 11	mfStat	-
12 ... 13	motionStat	-
14 ... 15	driveInput	-
16 ... 19	_p_act	Parameter *_p_act*, Modbus 7706
20 ... 23	_v_act	Parameter *_v_act*, Modbus 7744
24 ... 25	_I_act	Parameter *_I_act*, Modbus 7686
26 ... 29	EthOptMapInp1	Parameter *EthOptMapInp1*, Modbus 17512
30 ... 33	EthOptMapInp2	Parameter *EthOptMapInp2*, Modbus 17514
34 ... 37	EthOptMapInp3	Parameter *EthOptMapInp3*, Modbus 17516

Double Words “PCTRLsm” and “PVsm”

The two double words "PCTRLsm" and "PVsm" are used to read and write parameters, see I/O Scanning - Parameter Channel, page 18.

Word “driveStat”

The current operating state is indicated with the "driveStat" word.

For a detailed description of the bits, see Indication of the Operating State via Fieldbus, page 41.

Word “mfStat”

The word "mfStat" is used to indicate the current operating mode.

For a detailed description of the bits, see Indicating an Operating Mode, page 42.

Word “motionStat”

The word "motionStat" is used to provide information on the motor and profile generator.

Bit	Meaning
0	Positive limit switch triggered(1)
1	Negative limit switch triggered(1)
2 ... 5	Reserved
6	MOTZ: Motor at a standstill
7	MOTP: Motor movement in positive direction
8	MOTN: Motor movement in negative direction
9	Setting via parameter *DS402intLim*
10	Setting via parameter *DPL_intLim*
11	TAR0: Profile generator at standstill
12	DEC: Profile generator decelerates
13	ACC: Profile generator accelerates
14	CNST: Profile generator moves at constant velocity
15	Reserved

(1)	With firmware version ≥V01.14

Word “driveInput”

The word "driveInput" is used to indicate the state of the digital signal inputs.

Bit	Signal	Factory setting
0	*DI0*	Signal input function Freely Available
1	*DI1*	Signal input function Reference Switch (REF)
2	*DI2*	Signal input function Positive Limit Switch (LIMP)
3	*DI3*	Signal input function Negative Limit Switch (LIMN)
4	*DI4*	Signal input function Freely Available
5	*DI5*	Signal input function Freely Available
6 ... 7	-	Reserved
8	*DI11* (module IOM1)	Signal input function Freely Available
9	*DI12* (module IOM1)	Signal input function Freely Available
10	*DI13* (module IOM1)	Signal input function Freely Available
11	*DI14* (module IOM1)	Signal input function Freely Available
12 ... 15	-	Reserved

Double Word “_p_act”

The double word "_p_act" indicates the actual position. The value corresponds to the parameter *_p_act*.

Double Word “_v_act”

The double word "_v_act" indicates the actual velocity. The value corresponds to the parameter *_v_act*.

Word “_I_act”

The word "_I_act" indicates the actual current. The value corresponds to the parameter *_I_act*.

Double Words “EthOptMapInp1 ... EthOptMapInp3”

The double words EthOptMapInp1 ... EthOptMapInp3 contain selectable parameters, see Setting the Mapping for I/O Scanning, page 37.

I/O Scanning - Parameter Channel

Overview

The master can request a parameter value from the slave or modify a parameter value via the parameter channel. Each parameter can be addressed via the index and subindex.

Byte	Name	Description
0 ... 3	PCTRLms and PCTRLsm	Bits 0 ... 15: Word “Index” Bits 16 ... 23: Byte “Subindex” Bits 24 ... 31: Byte “Ctrl”
4 ... 7	PVms and PVsm	Double Word “ParameterValue”

Word “Index”

The word "Index" contains the parameter address.

See the user guide of the drive for a list of the parameters.

Byte “Subindex”

The byte "Subindex" must be set to the value 00 hex.

Byte “Ctrl”

Byte "Ctrl" contains the request to read or write a parameter.

The transmit data contains the information whether a parameter is to be read or written. The receive data contains the information whether the read request or the write request were successful.

Transmit data:

Ctrl	Function
02 hex	No request
12 hex	Read request
22 hex	Write request (word)
32 hex	Write request (double word)

Receive data:

Ctrl	Function
02 hex	Request not yet completed
12 hex	Read request or write request successfully completed (word)
22 hex	Read request or write request successfully completed (double word)
72 hex	Error message

Only one request can be processed at a time. The slave provides the response until the master sends a new request. If a response includes parameter values, the slave responds with the current value in the case of a repetition.

Read requests are only executed by the slave if the value changes from 02 hex to 12 hex. Write requests requests are only executed by the slave if the value changes from 02 hex to 22 hex or to 32 hex.

Double Word “ParameterValue”

The double word "ParameterValue" contains the parameter value.

In the case of a read request, the value in the transmit data has no significance. The receive data contains the parameter value.

In the case of a write request, the transmit data contains the value to be written to the parameter. The receive data contains the parameter value.

If a read request or a write request were not successful, the double word "ParameterValue" contains the error number of the error.

Example: Reading a Parameter

In the example, the program number of the product is read from the parameter *_prgNoDEV*. The parameter *_prgNoDEV* has the parameter address 258 (01 02 hex).

The parameter value read has the decimal value 91200 which corresponds to 01 64 40 hex.

Transmit data:

Byte Ctrl	Byte Subindex	Word Index	Double word ParameterValue
12 hex	00 hex	01 02 hex	00 00 00 00 hex

Receive data:

Byte Ctrl	Byte Subindex	Word Index	Double word ParameterValue
22 hex	00 hex	01 02 hex	00 01 64 40 hex

Example: Writing of an Invalid Parameter

In this example, the value of a non-existent parameter is to be modified. The parameter has the parameter address 101 (00 65 hex). The value of the parameters is to be modified to 222 (DE hex).

Before the slave can accept a new request, the value 02 hex must be transmitted in byte "Ctrl".

Since the slave cannot address the parameter, a synchronous error message is transmitted with the receive data. Byte "Ctrl" is set to 72 hex. Double word "PV" is set to the error number (error number 11 01 hex: Parameter does not exist).

Transmit data:

Byte Ctrl	Byte Subindex	Word Index	Double word ParameterValue
32 hex	00 hex	00 65 hex	00 00 00 DE hex

Receive data:

Byte Ctrl	Byte Subindex	Word Index	Double word ParameterValue
72 hex	00 hex	00 65 hex	00 00 11 01 hex

See the user guide of the drive for information on the error numbers.

Modbus Services - "Function Code"

"Function Code" 3 (Read Multiple Registers)

Description

The "Function Code" 3 (Read Multiple Registers) allows you to read several consecutive parameters, starting at any address.

Modbus Request

Structure of the Modbus request:

Field	Bytes	Value	Meaning
Function Code	1	3 = 03 hex	Read Multiple Registers
Starting Address	2	(various)	Address of the first parameter to be read
Quantity Of Registers	2	2 * n	Number of 16 bit values to be read (1 parameter has the value 2 since a parameter consists of a 32 bit value)

Modbus Response

Structure of the positive Modbus response:

Field	Bytes	Value	Meaning
Function Code	1	3 = 03 hex	Read Multiple Registers
Byte Count	1	4 * n	Number of data bytes
Registers Value	4 * n	(various)	Parameter values

Structure of the negative Modbus response

Field	Bytes	Value	Meaning
Function Code	1	03 hex + 80 hex = 83 hex	Read Multiple Registers
Exception Code	1	01 hex ... 04 hex	See Modbus Response to a Modbus Request, page 13

"Function Code" 8 (Diagnostics)

Description

The "Function Code" 8 (Diagnostics) allows you to read diagnostics data of the slave.

Modbus Request

Structure of the Modbus request:

Field	Bytes	Value	Meaning
Function Code	1	8 = 08 hex	Diagnostics
Sub-function Code	2	(various)	Diagnostics function
Data	2	(various)	Data (depending on diagnostics function)

Modbus Response

Structure of the positive Modbus response:

Field	Bytes	Value	Meaning
Function Code	1	8 = 08 hex	Diagnostics
Sub-function Code	2	(various)	Diagnostics function
Data	2	(various)	Diagnostics data

Structure of the negative Modbus response

Field	Bytes	Value	Meaning
Function Code	1	08 hex + 80 hex = 88 hex	Diagnostics
Exception Code	1	01 hex ... 04 hex	See Modbus Response to a Modbus Request, page 13

Sub-Function Code

The following diagnostics functions are available:

Sub-function Code		Diagnostics function
00	Return Query Data	Return request as a response
01	Restart Communication Option	Re-initialize the communication port
02	Return Diagnostic Register	Return the error code in the case of synchronous errors
03	(reserved)	–
04	Force Listen Only Mode	Force Listen Only mode of slave
05 ... 09	(reserved)	–
10	Clear Counters and Diagnostic Register	Clear the statistical counters
11	Return Bus Message Count	Return number of detected Bus Message
12	Return Bus Communication Error Count	Return number of detected Bus Communication Error
13	Return Bus Exception Error Count	Return number of detected Bus Exception Error
14 ... 15	(reserved)	–
16	Return Slave NAK Count	Return number of detected Slave Not-Acknowledged
17	Return Slave Busy Count	Return number of detected Slave Busy
18	Return Bus Char Overrun Count	Return number of detected Bus Char Overrun
>18	(reserved)	–

"Function Code" 16 (Write Multiple Registers)

Description

The "Function Code" 16 (Write Multiple Registers) allows you to write several consecutive parameters, starting at any address.

Modbus Request

Structure of the Modbus request:

Field	Bytes	Value	Meaning
Function Code	1	16 = 10 hex	Write Multiple Registers
Starting Address	2	(various)	Address of the first parameter to be written
Quantity Of Registers	2	2 * m	Number of 16 bit values to be written (1 parameter has the value 2 since a parameter consists of a 32 bit value)
Byte Count	1	4 * m	Number of data bytes
Registers Value	2 * m	(various)	Parameter values

Modbus Response

Structure of the positive Modbus response:

Field	Bytes	Value	Meaning
Function Code	1	16 = 10 hex	Write Multiple Registers
Starting Address	2	(various)	Corresponds to the Modbus request
Quantity Of Registers	2	2 * m	Corresponds to the Modbus request

Structure of the negative Modbus response

Field	Bytes	Value	Meaning
Function Code	1	10 hex + 80 hex = 90 hex	Write Multiple Registers
Exception Code	1	01 hex ... 04 hex	See Modbus Response to a Modbus Request, page 13

"Function Code" 23 (ReadWrite Multiple Registers)

Description

The "Function Code" 23 (ReadWrite Multiple Registers) allows you to read and write several consecutive parameters, starting at any address.

Modbus Request

Structure of the Modbus request:

Field	Bytes	Value	Meaning
Function Code	1	23 = 17 hex	Read/Write Multiple Registers
Read Starting Address	2	(various)	Address of the first parameter to be read
Quantity To Read	2	2 * n	Number of 16 bit values to be read (1 parameter has the value 2 since a parameter consists of a 32 bit value)
Write Starting Address	2	(various)	Address of the first parameter to be written
Quantity To Write	2	2 * m	Number of 16 bit values to be written (1 parameter has the value 2 since a parameter consists of a 32 bit value)
Write Byte Count	1	4 * m	Number of data bytes
Write Registers Value	4 * m	(various)	Parameter values

Modbus Response

Structure of the positive Modbus response:

Field	Bytes	Value	Meaning
Function Code	1	23 = 17 hex	Read/Write Multiple Registers
Byte Count	1	2 * n	Number of data bytes
Read Registers Value	2 * n	(various)	Parameter values

Structure of the negative Modbus response

Field	Bytes	Value	Meaning
Function Code	1	17 hex + 80 hex = 97 hex	Read/Write Multiple Registers
Exception Code	1	01 hex ... 04 hex	See Modbus Response to a Modbus Request, page 13

"Function Code" 43 (Encapsulated Interface Transport)

Description

The "Function Code" 43 / 14 (Read Device Identification) allows you to read device-specific data.

Modbus Request

Structure of the Modbus request:

Field	Bytes	Value	Meaning
Function Code	1	43 = 2B hex	Encapsulated Interface Transport
Modbus Encapsulated Interface Type	1	14 = 0E hex	Fixed value 14 (Read Device Identification)
Read Device ID Code	1	01	Read the objects
Object ID	1	0 x 00	Object ID

Modbus Response

Structure of the positive Modbus response:

Field	Bytes	Value	Meaning
Function Code	1	43 = 2B hex	Encapsulated Interface Transport
Modbus Encapsulated Interface Type	1	14 = 0E hex	Fixed value 14 (Read Device Identification)
Read Device ID Code	1	01	Corresponds to the Modbus request
Conformity Level	1	02	Fixed value
More Follows	1	00	Fixed value
Next Object ID	1	00	Fixed value
Number Of Objects	1	03	Number of objects
Object ID	1		Object ID, page 24
Object Length	1		Object length
Object Value		(various)	Object data (various)

Structure of the negative Modbus response

Field	Bytes	Value	Meaning
Function Code	1	2B hex + 80 hex = AB hex	Encapsulated Interface Transport
Exception Code	1	01 hex ... 04 hex	See Modbus Response to a Modbus Request, page 13

Object ID

The following object IDs are available:

Object ID	Object name	Value
00 hex	vendor name	Manufacturer name
01 hex	product code	„xxxxxxxxxxx“ (see type code of the drive)
03 hex	revision	"Vxx.yyy" (for example "V02.001")

Example of "Function Code" 3

Description

Reading an error memory entry. Since the Modbus addresses of the parameters of an error memory entry are contiguous (ascending order), a single Modbus request is sufficient.

Parameters *_ERR_number (15362)*, *_ERR_class (15364)*, *_ERR_time (15366)* and *_ERR_qual (15368)*.

Modbus Request

Structure of the Modbus request:

Field	Bytes	Value	Meaning
Function Code	1	3	Read Multiple Registers
Starting Address	2	15362 (3C02 hex)	Address of the first parameter to be read
Quantity Of Registers	2	8	Number of the 16 bit values to be read = 8

Modbus Response

Structure of the positive Modbus response:

Field	Bytes	Value	Meaning
Function Code	1	3	Read Multiple Registers
Byte Count	1	16	Number of bytes: 8 bytes of data
Registers Value	16	32 bit value	*_ERR_number*, 15362 (error number)
		32 bit value	*_ERR_class*, 15364 (error class)
		32 bit value	*_ERR_time*, 15366 (error time)
		32 bit value	*_ERR_qual*, 15368 (error qualifier)

Example of "Function Code" 16

Description

Writing of the software limit switches. Since these parameters have consecutive addresses, a single Modbus request is sufficient:

Parameters *MON_swLimP (1544)* and *MON_swLimN (1546)*.

Modbus Request

Structure of the Modbus request:

Field	Bytes	Value	Meaning
Function Code	1	16	Write Multiple Registers
Starting Address	2	1544 (608 hex)	Address of the first parameter to be written
Quantity Of Registers	2	4	Number of parameters = 4 (8 bytes of data)
Byte Count	1	8	Number of bytes: 8 bytes of data
Registers Value	8	32 bit value 32 bit value	*MON_swLimP*, 1544 *MON_swLimN*, 1546

Modbus Response

Structure of the positive Modbus response:

Field	Bytes	Value	Meaning
Function Code	1	16	Write Multiple Registers
Starting Address	2	1544 (608 hex)	Address of the parameter
Quantity Of Registers	2	4	Number of parameters = 4 (8 bytes of data)

Installation

Installation of the Module

Mechanical Installation

Electrostatic discharge (ESD) may permanently damage the module either immediately or over time.

NOTICE
EQUIPMENT DAMAGE DUE TO ESD • Use suitable ESD measures (for example, ESD gloves) when handling the module. • Do not touch internal components. **Failure to follow these instructions can result in equipment damage.**

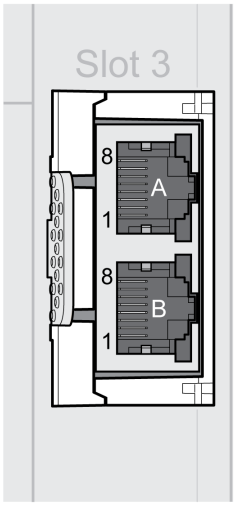
Install the module according to the instructions in the user guide of the drive.

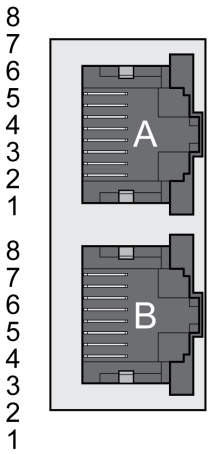
Cable Specifications

Shield:	Required
Twisted Pair:	Required
PELV:	Required
Cable composition:	8 x 0.25 mm² (8 x AWG 22)
Maximum cable length:	100 m (328 ft)

- Note the pertinent information on equipotential bonding conductors in the user guide of the drive.
- Use pre-assembled cables to reduce the risk of wiring errors.

Pin Assignment





A Port A

B Port B

Pin	Signal	Meaning
1	*Tx+*	Ethernet transmit signal +
2	*Tx-*	Ethernet transmit signal -
3	*Rx+*	Ethernet receive signal +
4	-	-
5	-	-
6	*Rx-*	Ethernet receive signal -
7	-	-
8	-	-

Commissioning

Preparation

This chapter describes how to commission the product.

The product is unable to detect an interruption of the network link if connection monitoring is not active.

⚠WARNING

LOSS OF CONTROL

- Ensure that connection monitoring is enabled.
- Set the shortest, practical monitoring time cycles to detect communication interruptions as quickly as possible.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

⚠WARNING

UNINTENDED EQUIPMENT OPERATION

- Only start the system if there are no persons or obstructions in the zone of operation.
- Do not write values to reserved parameters.
- Do not write values to parameters unless you fully understand the function.
- Run initial tests without coupled loads.
- Verify correct word order for fieldbus communication.
- Do not establish a fieldbus connection unless you have fully understood the communication principles.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Required Components

The following is required for commissioning:

- Commissioning software “Lexium32 DTM Library”

  https://www.se.com/ww/en/download/document/Lexium_DTM_Library/
- Fieldbus converter for the commissioning software for connection via the commissioning interface
- Modbus TCP master
- Lexium 32M Drive User Guide and this user guide, LXM32M Ethernet TCP/IP Module (Protocol Modbus TCP) User Guide

Performing a “First Setup”

Powering on the Drive

A "First Setup" is required when the controller supply is powered on for the first time or after the factory settings have been restored.

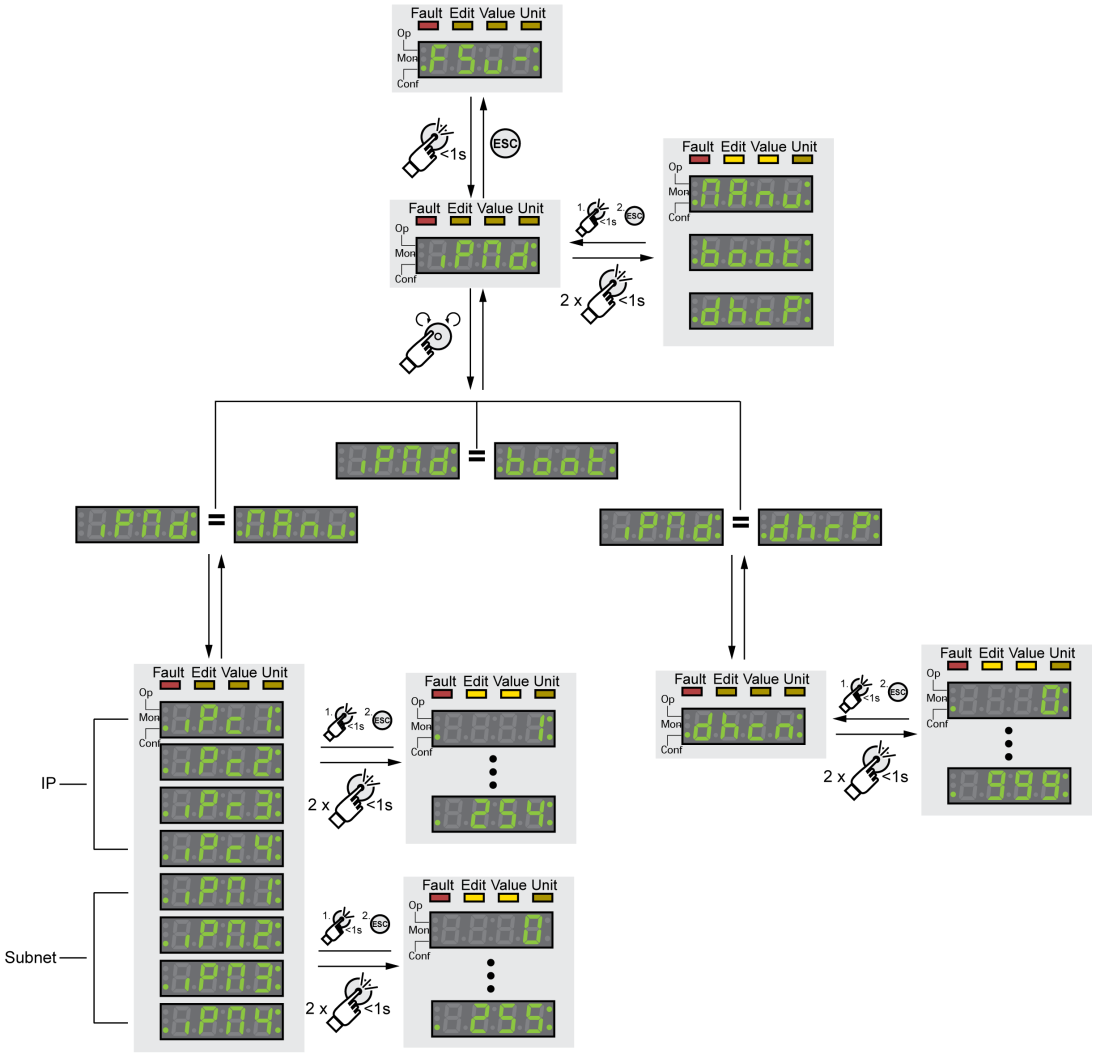
- Disconnect the drive from the fieldbus during commissioning in order to avoid conflicts by simultaneous access.
- Power on the controller supply.

The drive goes through an initialization routine, the LEDs are tested, the segments of the 7-segment display and the status LEDs light up.

After the initialization, the fieldbus interface must be configured. The drive can be configured via the integrated HMI or the commissioning software.

First Setup via HMI

First Setup via the integrated HMI



Type of Network Address Assignment

Select the type of network address assignment.

The type of network address assignment is set via the parameter *EthIpMode* (iPMd).

Parameter name HMI menu HMI name	Description	Unit Minimum value Factory setting Maximum value	Data type R/W Persistent Expert	Parameter address via fieldbus
EthIpMode CONF → COM - IPMd	Method of obtaining IP address. **0 / Manual /** MAnu: Manual **1 / BOOTP /** boot: BOOTP **2 / DHCP /** dhcP: DHCP Modified settings become active immediately.	- 0 2 2	UINT16 R/W per. -	CANopen 3044:5h Modbus 17418 Profibus 17418 CIP 168.1.5 ModbusTCP 17418 EtherCAT 3044:5h PROFINET 17418

Manual Assignment of the Network Address (*EthIpMode* = MAnu)

Set the network addresses consisting of the IP address and the subnet mask.

The IP address is set via the parameters *EthIPmodule1* ... *EthIPmodule4*. The subnet mask is set via the parameters *EthIPmask1* ... *EthIPmask4*.

Parameter name HMI menu HMI name	Description	Unit Minimum value Factory setting Maximum value	Data type R/W Persistent Expert	Parameter address via fieldbus
EthIPmodule1 CONF → COM - IPc 1	IP address Ethernet module, byte 1. Byte 1 (x.0.0.0) of the IP address of the Ethernet module. Modified settings become active the next time the product is powered on.	- 0 0 255	UINT16 R/W per. -	CANopen 3044:7h Modbus 17422 Profibus 17422 CIP 168.1.7 ModbusTCP 17422 EtherCAT 3044:7h PROFINET 17422
EthIPmodule2 CONF → COM - IPc 2	IP address Ethernet module, byte 2. Modified settings become active the next time the product is powered on.	- 0 0 255	UINT16 R/W per. -	CANopen 3044:8h Modbus 17424 Profibus 17424 CIP 168.1.8 ModbusTCP 17424 EtherCAT 3044:8h PROFINET 17424
EthIPmodule3 CONF → COM - IPc 3	IP address Ethernet module, byte 3. Modified settings become active the next time the product is powered on.	- 0 0 255	UINT16 R/W per. -	CANopen 3044:9h Modbus 17426 Profibus 17426 CIP 168.1.9 ModbusTCP 17426 EtherCAT 3044:9h PROFINET 17426

Parameter name HMI menu HMI name	Description	Unit Minimum value Factory setting Maximum value	Data type R/W Persistent Expert	Parameter address via fieldbus
EthIPmodule4 C o n F → C o M - , P c 4	IP address Ethernet module, byte 4. Modified settings become active the next time the product is powered on.	- 0 0 255	UINT16 R/W per. -	CANopen 3044:A_h Modbus 17428 Profibus 17428 CIP 168.1.10 ModbusTCP 17428 EtherCAT 3044:A_h PROFINET 17428
EthIPmask1 C o n F → C o M - , P M 1	IP address subnet mask, byte 1. Modified settings become active the next time the product is powered on.	- 0 255 255	UINT16 R/W per. -	CANopen 3044:B_h Modbus 17430 Profibus 17430 CIP 168.1.11 ModbusTCP 17430 EtherCAT 3044:B_h PROFINET 17430
EthIPmask2 C o n F → C o M - , P M 2	IP address subnet mask, byte 2. Modified settings become active the next time the product is powered on.	- 0 255 255	UINT16 R/W per. -	CANopen 3044:C_h Modbus 17432 Profibus 17432 CIP 168.1.12 ModbusTCP 17432 EtherCAT 3044:C_h PROFINET 17432
EthIPmask3 C o n F → C o M - , P M 3	IP address subnet mask, byte 3. Modified settings become active the next time the product is powered on.	- 0 255 255	UINT16 R/W per. -	CANopen 3044:D_h Modbus 17434 Profibus 17434 CIP 168.1.13 ModbusTCP 17434 EtherCAT 3044:D_h PROFINET 17434
EthIPmask4 C o n F → C o M - , P M 4	IP address subnet mask, byte 4. Modified settings become active the next time the product is powered on.	- 0 0 255	UINT16 R/W per. -	CANopen 3044:E_h Modbus 17436 Profibus 17436 CIP 168.1.14 ModbusTCP 17436 EtherCAT 3044:E_h PROFINET 17436

Assignment of the Network Address via BOOTP (*EthIpMode* = b o o t)

Verify that an accessible BOOTP server is available on the network.

Assignment of the Network Address via DHCP (*EthIpMode* = dhcP)

Verify that an accessible DHCP server is available on the network.

The DHCP server must support the "DeviceName" configuration.

Procedure:

Set a number that is unique in the network via dhcn.

The number is entered at the 13th, 14th and 15th digit of the device name.

Example: LEXIUM_SERVO**001**

In the commissioning software, the full device name can be displayed and modified.

Restarting the Drive

A restart of the drive is required for the modifications to become effective. After the restart, the drive is ready for operation. The drive is in the operating mode Jog.

Network Settings

Setting the Transmission Rate

Set the transmission rate with the parameter *EthRateSet*.

Parameter name **HMI menu** **HMI name**	**Description**	**Unit** **Minimum value** **Factory setting** **Maximum value**	**Data type** **R/W** **Persistent** **Expert**	**Parameter address via fieldbus**
EthRateSet	Transmission rate setting. **0 / Autodetect**: Autodetect **1 / 10 Mbps Full**: 10 Mbps full duplex **2 / 10 Mbps Half**: 10 Mbps half duplex **3 / 100 Mbps Full**: 100 Mbps full duplex **4 / 100 Mbps Half**: 100 Mbps half duplex Modified settings become active immediately.	- 0 0 4	UINT16 R/W per. -	CANopen 3044:2$_h$ Modbus 17412 Profibus 17412 CIP 168.1.2 ModbusTCP 17412 EtherCAT 3044:2$_h$ PROFINET 17412

Setting the Protocol

The protocol is set by means of the parameter *EthMode*.

Set the parameter *EthMode* to "Modbus TCP".

Parameter name HMI menu HMI name	Description	Unit Minimum value Factory setting Maximum value	Data type R/W Persistent Expert	Parameter address via fieldbus
EthMode CONF → COM - ETNd	Protocol. **0 / Modbus TCP** / nTCP: Modbus TCP I/O scanning is enabled **1 / EtherNet/IP** / EtIP: EtherNet/IP communication is enabled Modbus TCP parameter access is possible irrespective of the selected setting. Modified settings become active the next time the product is powered on.	- 0 1 1	UINT16 R/W per. -	CANopen 3044:1$_h$ Modbus 17410 Profibus 17410 CIP 168.1.1 ModbusTCP 17410 EtherCAT 3044:1$_h$ PROFINET 17410

Setting the Gateway

Set the IP address of the gateway with the parameters *EthIPgate1* ... *EthIPgate4*.

Parameter name HMI menu HMI name	Description	Unit Minimum value Factory setting Maximum value	Data type R/W Persistent Expert	Parameter address via fieldbus
EthIPgate1 CONF → COM - IPG1	IP address gateway, byte 1. Modified settings become active the next time the product is powered on.	- 0 0 255	UINT16 R/W per. -	CANopen 3044:F$_h$ Modbus 17438 Profibus 17438 CIP 168.1.15 ModbusTCP 17438 EtherCAT 3044:F$_h$ PROFINET 17438
EthIPgate2 CONF → COM - IPG2	IP address gateway, byte 2. Modified settings become active the next time the product is powered on.	- 0 0 255	UINT16 R/W per. -	CANopen 3044:10$_h$ Modbus 17440 Profibus 17440 CIP 168.1.16 ModbusTCP 17440 EtherCAT 3044:10$_h$ PROFINET 17440

Parameter name HMI menu HMI name	Description	Unit Minimum value Factory setting Maximum value	Data type R/W Persistent Expert	Parameter address via fieldbus
EthIPgate3 CONF → CON - IPG3	IP address gateway, byte 3. Modified settings become active the next time the product is powered on.	- 0 0 255	UINT16 R/W per. -	CANopen 3044:11$_h$ Modbus 17442 Profibus 17442 CIP 168.1.17 ModbusTCP 17442 EtherCAT 3044:11$_h$ PROFINET 17442
EthIPgate4 CONF → CON - IPG4	IP address gateway, byte 4. Modified settings become active the next time the product is powered on.	- 0 0 255	UINT16 R/W per. -	CANopen 3044:12$_h$ Modbus 17444 Profibus 17444 CIP 168.1.18 ModbusTCP 17444 EtherCAT 3044:12$_h$ PROFINET 17444

Master with Word Swap

The IP address of a master with Word Swap is set by means of the parameters *EthMbIPswap1* ... *EthMbIPswap4*.

You may not set an IP address for a master without Word Swap.

- Check whether or not the master uses Word Swap.
- If the master uses Word Swap, set the IP address of the master with the parameters *EthMbIPswap1* ... *EthMbIPswap4*.

Parameter name HMI menu HMI name	Description	Unit Minimum value Factory setting Maximum value	Data type R/W Persistent Expert	Parameter address via fieldbus
EthMbIPswap1	IP address of master for Modbus word swap, byte 1. IP address of a Modbus master device. For this master, the word order is swapped to "Low word first", instead of the default "High word first". High word first: Modicon Quantum Low word first: Premium, HMI (Schneider Electric) Modified settings become active immediately.	- 0 0 255	UINT16 R/W per. -	CANopen 3044:50$_h$ Modbus 17568 Profibus 17568 CIP 168.1.80 ModbusTCP 17568 EtherCAT 3044:50$_h$ PROFINET 17568
EthMbIPswap2	IP address of master for Modbus word swap, byte 2. Modified settings become active immediately.	- 0 0 255	UINT16 R/W per. -	CANopen 3044:51$_h$ Modbus 17570 Profibus 17570 CIP 168.1.81 ModbusTCP 17570 EtherCAT 3044:51$_h$ PROFINET 17570

Parameter name HMI menu HMI name	Description	Unit Minimum value Factory setting Maximum value	Data type R/W Persistent Expert	Parameter address via fieldbus
EthMbIPswap3	IP address of master for Modbus word swap, byte 3. Modified settings become active immediately.	- 0 0 255	UINT16 R/W per. -	CANopen 3044:52$_h$ Modbus 17572 Profibus 17572 CIP 168.1.82 ModbusTCP 17572 EtherCAT 3044:52$_h$ PROFINET 17572
EthMbIPswap4	IP address of master for Modbus word swap, byte 4. Modified settings become active immediately.	- 0 0 255	UINT16 R/W per. -	CANopen 3044:53$_h$ Modbus 17574 Profibus 17574 CIP 168.1.83 ModbusTCP 17574 EtherCAT 3044:53$_h$ PROFINET 17574

Settings for Communication with I/O Scanning

Activating I/O Scanning

I/O scanning is activated/deactivated by means of the parameter *EthMbScanner*.

If you do not want to use I/O scanning, set the parameter *EthMbScanner* to "Off".

Parameter name HMI menu HMI name	Description	Unit Minimum value Factory setting Maximum value	Data type R/W Persistent Expert	Parameter address via fieldbus
EthMbScanner	Modbus TCP I/O scanning. **0 / Off**: Modbus TCP I/O scanning off **1 / On**: Modbus TCP I/O scanning on I/O scanning only works if the parameter EthMode is set to Modbus TCP. Modified settings become active immediately.	- 0 1 1	UINT16 R/W per. -	CANopen 3044:28$_h$ Modbus 17488 Profibus 17488 CIP 168.1.40 ModbusTCP 17488 EtherCAT 3044:28$_h$ PROFINET 17488

Setting the Master for I/O Scanning

Entering the IP address of a master reserves I/O scanning for this master. This means that no other master on the network can perform I/O scanning.

If the IP addresses are not set correctly, any network device may control the system or access by the master may be blocked.

⚠WARNING

UNINTENDED EQUIPMENT OPERATION DUE TO UNLIMITED ACCESS

Verify that you have set the correct master IP address.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Set the IP address of the master for I/O scanning with the parameters *EthIPmaster1* ... *EthIPmaster4*.

Parameter name **HMI menu** **HMI name**	**Description**	**Unit** **Minimum value** **Factory setting** **Maximum value**	**Data type** **R/W** **Persistent** **Expert**	**Parameter address via fieldbus**
EthIPmaster1	IP address master, byte 1. IP address of the master that is permitted to perform Modbus TCP I/O scanning. If set to 0.0.0.0 (default), any master can perform I/O scanning. Setting can only be modified if power stage is disabled. Modified settings become active immediately.	- 0 0 255	UINT16 R/W per. -	CANopen 3044:29$_h$ Modbus 17490 Profibus 17490 CIP 168.1.41 ModbusTCP 17490 EtherCAT 3044:29$_h$ PROFINET 17490
EthIPmaster2	IP address master, byte 2. Setting can only be modified if power stage is disabled. Modified settings become active immediately.	- 0 0 255	UINT16 R/W per. -	CANopen 3044:2A$_h$ Modbus 17492 Profibus 17492 CIP 168.1.42 ModbusTCP 17492 EtherCAT 3044:2A$_h$ PROFINET 17492
EthIPmaster3	IP address master, byte 3. Setting can only be modified if power stage is disabled. Modified settings become active immediately.	- 0 0 255	UINT16 R/W per. -	CANopen 3044:2B$_h$ Modbus 17494 Profibus 17494 CIP 168.1.43 ModbusTCP 17494 EtherCAT 3044:2B$_h$ PROFINET 17494
EthIPmaster4	IP address master, byte 4. Setting can only be modified if power stage is disabled. Modified settings become active immediately.	- 0 0 255	UINT16 R/W per. -	CANopen 3044:2C$_h$ Modbus 17496 Profibus 17496 CIP 168.1.44 ModbusTCP 17496 EtherCAT 3044:2C$_h$ PROFINET 17496

Setting the Mapping for I/O Scanning

The input mapping is set by means of the parameters *EthOptMapInp1* ... *EthOptMapInp3*.

The output mapping is set by means of the parameters *EthOptMapOut1* ... *EthOptMapOut3*.

Set the desired mapping values with the parameters *EthOptMapInp1* ... *EthOptMapInp3* and *EthOptMapOut1* ... *EthOptMapOut3*.

Parameter name **HMI menu** **HMI name**	**Description**	**Unit** **Minimum value** **Factory setting** **Maximum value**	**Data type** **R/W** **Persistent** **Expert**	**Parameter address via fieldbus**
EthOptMapInp1	Optionally mapped input parameter 1 (drive to controller). Modbus address of parameter which is optionally mapped to Ethernet/IP assembly or Modbus TCP I/O scanner data (drive to controller). Modified settings become active immediately.	- - 0 -	UINT16 R/W per. -	CANopen 3044:34$_h$ Modbus 17512 Profibus 17512 CIP 168.1.52 ModbusTCP 17512 EtherCAT 3044:34$_h$ PROFINET 17512
EthOptMapInp2	Optionally mapped input parameter 2 (drive to controller). Modbus address of parameter which is optionally mapped to Ethernet/IP assembly or Modbus TCP I/O scanner data (drive to controller). Modified settings become active immediately.	- - 0 -	UINT16 R/W per. -	CANopen 3044:35$_h$ Modbus 17514 Profibus 17514 CIP 168.1.53 ModbusTCP 17514 EtherCAT 3044:35$_h$ PROFINET 17514
EthOptMapInp3	Optionally mapped input parameter 3 (drive to controller). Modbus address of parameter which is optionally mapped to Ethernet/IP assembly or Modbus TCP I/O scanner data (drive to controller). Modified settings become active immediately.	- - 0 -	UINT16 R/W per. -	CANopen 3044:36$_h$ Modbus 17516 Profibus 17516 CIP 168.1.54 ModbusTCP 17516 EtherCAT 3044:36$_h$ PROFINET 17516
EthOptMapOut1	Optionally mapped output parameter 1 (controller to drive). Modbus address of parameter which is optionally mapped to Ethernet/IP assembly or Modbus TCP I/O scanner data (controller to drive). Modified settings become active immediately.	- - 0 -	UINT16 R/W per. -	CANopen 3044:2E$_h$ Modbus 17500 Profibus 17500 CIP 168.1.46 ModbusTCP 17500 EtherCAT 3044:2E$_h$ PROFINET 17500

Parameter name HMI menu HMI name	Description	Unit Minimum value Factory setting Maximum value	Data type R/W Persistent Expert	Parameter address via fieldbus
EthOptMapOut2	Optionally mapped output parameter 2 (controller to drive). Modbus address of parameter which is optionally mapped to Ethernet/IP assembly or Modbus TCP I/O scanner data (controller to drive). Modified settings become active immediately.	- - 0 -	UINT16 R/W per. -	CANopen 3044:2F$_h$ Modbus 17502 Profibus 17502 CIP 168.1.47 ModbusTCP 17502 EtherCAT 3044:2F$_h$ PROFINET 17502
EthOptMapOut3	Optionally mapped output parameter 3 (controller to drive). Modbus address of parameter which is optionally mapped to Ethernet/IP assembly or Modbus TCP I/O scanner data (controller to drive). Modified settings become active immediately.	- - 0 -	UINT16 R/W per. -	CANopen 3044:30$_h$ Modbus 17504 Profibus 17504 CIP 168.1.48 ModbusTCP 17504 EtherCAT 3044:30$_h$ PROFINET 17504

Setting Communication Monitoring for I/O Scanning

The product is unable to detect an interruption of the network link if connection monitoring is not active.

⚠WARNING

LOSS OF CONTROL

- Ensure that connection monitoring is enabled.
- Set the shortest, practical monitoring time cycles to detect communication interruptions as quickly as possible.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Set communication monitoring for I/O scanning with the parameter *EthMbScanTimeout*.

Parameter name HMI menu HMI name	Description	Unit Minimum value Factory setting Maximum value	Data type R/W Persistent Expert	Parameter address via fieldbus
EthMbScanTimeout	Modbus TCP I/O scanning timeout. Communication monitoring timeout for Modbus TCP. Value 0: Timeout monitoring disabled In increments of 0.1 s. Modified settings become active immediately.	s 0.0 2.0 60.0	UINT16 R/W per. -	CANopen 3044:2D$_h$ Modbus 17498 Profibus 17498 CIP 168.1.45 ModbusTCP 17498 EtherCAT 3044:2D$_h$ PROFINET 17498

Settings for Communication without I/O Scanning

Overview

It is also possible to establish communication without I/O scanning.

The following settings must be made to establish communication without I/O scanning:

- Activation of communication monitoring
- Exclusive use of access channel

Setting Communication Monitoring

The product is unable to detect an interruption of the network link if connection monitoring is not active.

⚠WARNING

LOSS OF CONTROL

- Ensure that connection monitoring is enabled.
- Set the shortest, practical monitoring time cycles to detect communication interruptions as quickly as possible.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Communication monitoring must be activated via the parameter *MBnode_guard*:

Parameter name **HMI menu** **HMI name**	**Description**	**Unit** **Minimum value** **Factory setting** **Maximum value**	**Data type** **R/W** **Persistent** **Expert**	**Parameter address via fieldbus**
MBnode_guard	Modbus Node Guarding. Value 0: Node Guarding inactive Value >0: Monitoring time A read request or a write request must be performed during the monitoring time. Modified settings become active immediately.	ms 0 0 10000	UINT16 R/W - -	CANopen 3016:6$_h$ Modbus 5644 Profibus 5644 CIP 122.1.6 ModbusTCP 5644 EtherCAT 3016:6$_h$ PROFINET 5644

Communication monitoring triggers an error of error class 2 if communication is interrupted. After the error message is reset, communication monitoring is active again.

Exclusive Use of Access Channel

In addition, the access channel must be used exclusively. Only after this it is possible to change operating states and start operating modes.

Once the access channel is used exclusively, it is no longer possible to change operating states and start operating modes via another access channel.

Writing the parameter *AccessExcl* sets the access channel to exclusive access.

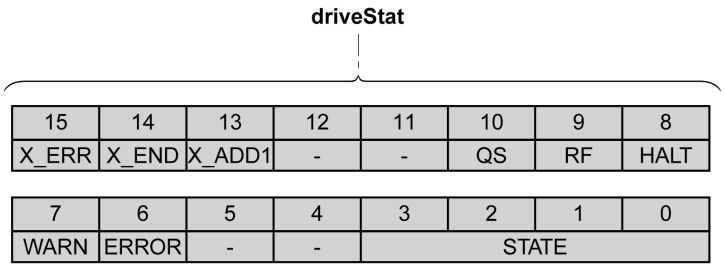
Parameter name **HMI menu** **HMI name**	**Description**	**Unit** **Minimum value** **Factory setting** **Maximum value**	**Data type** **R/W** **Persistent** **Expert**	**Parameter address via fieldbus**
AccessExcl	Get exclusive access to access channel. Write parameter: Value 0: Release access channel Value 1: Use exclusive access for access channel Read parameter: Value 0: Access channel is not used exclusively Value 1: Access channel is used exclusively (access channel used for reading) Modified settings become active immediately.	- - - - 	UINT16 R/W - - 	CANopen 3001:D_h Modbus 282 Profibus 282 CIP 101.1.13 ModbusTCP 282 EtherCAT 3001:D_h PROFINET 282

Operating States and Operating Modes

Operating States

Indication of the Operating State via Fieldbus

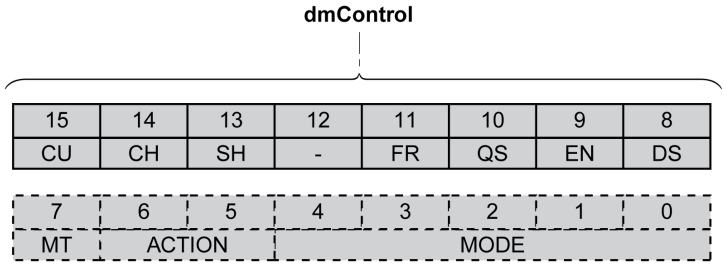
The operating state is indicated with the word "driveStat".



bit	Name	Meaning
0 ... 3	STATE	Operating state (binary coded) **1** Start **2** Not Ready To Switch On **3** Switch On Disabled **4** Ready To Switch On **5** Switched On **6** Operation Enabled **7** Quick Stop Active **8** Fault Reaction Active **9** Fault
4 ... 5	-	Reserved
6	ERROR	Error detected (error classes 1 ... 3)
7	WARN	Error detected (error class 0)
8	HALT	"Halt" is active
9	RF	Homing valid
10	QS	"Quick Stop" is active
11 ... 12	-	Reserved
13	X_ADD1	Operating mode-specific information
14	X_END	Operating mode terminated
15	X_ERR	Operating mode terminated with error

Changing the Operating State via Fieldbus

Bits 8 ... 15 of the word "dmControl" are used to set the operating state.



Bit	Name	Meaning	Operating state
8	DS	Disabling the power stage	**6** Operation Enabled -> **4** Ready To Switch On
9	EN	Enabling the power stage	**4** Ready To Switch On -> **6** Operation Enabled
10	QS	Perform "Quick Stop"	**6** Operation Enabled -> **7** Quick Stop Active
11	FR	Perform "Fault Reset"	**7** Quick Stop Active -> **6** Operation Enabled **9** Fault -> **4** Ready To Switch On
12	-	Reserved	Reserved
13	SH	Execute "Halt"	**6** Operation Enabled
14	CH	Clear "Halt"	**6** Operation Enabled
15	CU	Resume operating mode interrupted by "Halt"	**6** Operation Enabled

In the case of an access, the bits respond to a 0->1 change to trigger the corresponding function.

If a request for changing the operating state is not successful, this request is ignored. There is no error response.

If the bits 8 ... 15 are set to 0, the power stage will be disabled.

Ambivalent bit combinations are treated in accordance with the following priority list (highest priority bit 8, lowest priority bit 14 and bit 15):

- Bit 8 (disable power stage) prior to bit 9 (enable power stage)
- Bit 10 ("Quick Stop") prior to bit 11 ("Fault Reset")
- Bit 13 (execute "Halt") prior to bit 14 (clear "Halt") and bit 15 (resume operating mode interrupted by "Halt")

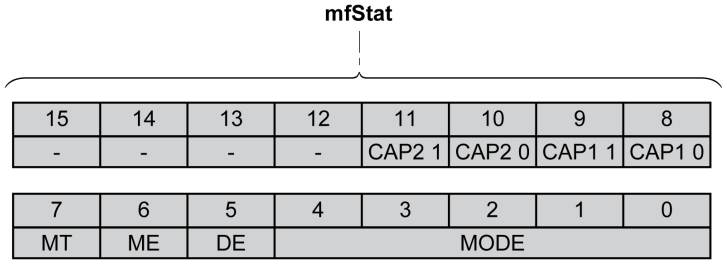
In the case of an error of error class 2 or error class 3, a "Fault Reset" can only be performed when bit 9 (enable power stage) is no longer set.

Operating Modes

Indicating an Operating Mode

Indicating an Operating Mode

The word “mfStat” is used to indicate the set operating mode.

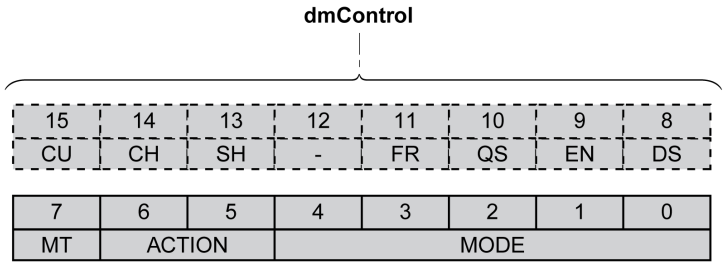


bit	Name	Description
0 ... 4	MODE	Indicates the set operating mode Value 01_h: Profile Position Value 03_h: Profile Velocity Value 04_h: Profile Torque Value 06_h: Homing Value $1D_h$: Motion Sequence Value $1E_h$: Electronic Gear Value $1F_h$: Jog
5	DE	The bit “DE” (Data Error) relates to parameters that are independent of the bit “MT” (Mode Toggle). The bit “DE” (Data Error) is set if a data value in the process data channel is invalid.
6	ME	The bit “ME” (Mode Error) relates to parameters that are dependent on the bit “MT” (Mode Toggle). The bit “ME” (Mode Error) is set if a request (for example, starting an operating mode) was rejected.
7	MT	Bit “MT” (Mode Toggle)
8 ... 9	CAP1	Bit 0 and bit 1 of parameter *_Cap1Count*
10 ... 11	CAP2	Bit 0 and bit 1 of parameter *_Cap2Count*
12 ... 15	-	Reserved

Starting and Changing an Operating Mode

Starting and Changing an Operating Mode

Bits 0 ... 7 in the word “dmControl” are used to set the operating mode.



bit	Name	Description
0 ... 4	MODE	Operating Mode Value 01_h: Profile Position Value 03_h: Profile Velocity Value 04_h: Profile Torque Value 06_h: Homing Value $1D_h$: Motion Sequence Value $1E_h$: Electronic Gear Value $1F_h$: Jog
5 ... 6	AC-TION	Operating mode-dependent
7	MT	Bit “MT” (Mode Toggle)

Via the following values the operating mode can be activated or target values can be changed:

- Target values, depending on required operating mode
- Operating mode in “dmControl”, bits 0 ... 4 (MODE).
- Action for this operating mode in bit 5 and bit 6 (ACTION)
- Toggle bit 7 (MT)

The following sections describe the possible operating modes, functions and the corresponding target values.

Overview of Operating Modes

Operating Mode	**dmControl** **Bits 0 ... 6** **MODE+ACTION**	**RefA32**	**RefB32**
JOG	$1F_h$	Value 0: No movement Value 1: Slow movement in positive direction Value 2: Slow movement in negative direction Value 5: Fast movement in positive direction Value 6: Fast movement in negative direction	-
Electronic Gear: Position synchronization without compensation movement	$1E_h$	As *GEARdenom*	As *GEARnum*
Electronic Gear: Position synchronization with compensation movement	$3E_h$	As *GEARdenom*	As *GEARnum*
Electronic Gear: Velocity synchronization	$5E_h$	As *GEARdenom*	As *GEARnum*
Profile Torque: Via analog input	04_h	-	-
Profile Torque: Via parameter	24_h	As *PTtq_target*	As *RAMP_tq_slope*
Profile Torque: Via PTI interface	44_h	-	-
Profile Velocity: Via analog input	03_h	-	-
Profile Velocity: Via parameter	23_h	As *PVv_target*	-
Profile Position: Absolute	01_h	As *PPv_target*	As *PPp_target*
Profile Position: Relative with reference to the currently set target position	21_h	As *PPv_target*	As *PPp_target*
Profile Position: Relative with reference to the motor position	41_h	As *PPv_target*	As *PPp_target*
Homing: Position setting	06_h	-	As *HMp_setP*
Homing: Reference Movement	26_h	As *HMmethod*	-
Motion Sequence: Start sequence	$1D_h$	Data set number	Value 1: Use data set number
Motion Sequence: Start individual data set	$3D_h$	Data set number	-

Operating Mode Jog

Starting the Operating Mode

The operating mode is set and started in the process data channel with the output data.

dmControl Bits 0 ... 6 MODE+ACTION	RefA32	RefB32
$1F_h$	Value 0: No movement Value 1: Slow movement in positive direction Value 2: Slow movement in negative direction Value 5: Fast movement in positive direction Value 6: Fast movement in negative direction	-

Status Information

The word "driveStat" provides information on the operating mode.

bit	Name	Meaning
13	X_ADD1	Reserved
14	X_END	0: Operating mode started 1: Operating mode terminated
15	X_ERR	0: No error detected 1: Error detected

Terminating the Operating Mode

The operating mode is terminated when the motor is at a standstill and one of the following conditions is met:

- Value 0 RefA
- Stop caused by "Halt" or "Quick Stop"
- Stop caused by a detected error

Operating Mode Electronic Gear

Starting the Operating Mode

The operating mode is set and started in the process data channel with the output data.

Method	dmControl Bits 0 ... 6 MODE+ACTION	RefA32	RefB32
Position synchronization without compensation movement	$1E_h$	As *GEARdenom*	As *GEARnum*
Position synchronization with compensation movement	$3E_h$	As *GEARdenom*	As *GEARnum*
Velocity synchronization	$5E_h$	As *GEARdenom*	As *GEARnum*

Status Information

The word "driveStat" provides information on the operating mode.

bit	Name	Meaning
13	X_ADD1	1: Reference velocity reached(1)
14	X_END	0: Operating mode started 1: Operating mode terminated
15	X_ERR	0: No error detected 1: Error detected

(1)	Only with method Velocity synchronization and with active velocity window.

Terminating the Operating Mode

The operating mode is terminated when the motor is at a standstill and one of the following conditions is met:

- Stop caused by "Halt" or "Quick Stop"
- Stop caused by a detected error

Operating Mode Profile Torque

Starting the Operating Mode

The operating mode is set and started in the process data channel with the output data.

Method	dmControl Bits 0 ... 6 MODE+ACTION	RefA32	RefB32
Via analog input	04_h	-	-
Via parameter	24_h	As *PTtq_target*	As *RAMP_tq_slope*
Via PTI interface	44_h	-	-

Status Information

The word "driveStat" provides information on the operating mode.

bit	Name	Meaning
13	X_ADD1	0: Target torque not reached 1: Target torque reached
14	X_END	0: Operating mode started 1: Operating mode terminated
15	X_ERR	0: No error detected 1: Error detected

Terminating the Operating Mode

The operating mode is terminated when the motor is at a standstill and one of the following conditions is met:

- Stop caused by "Halt" or "Quick Stop"
- Stop caused by a detected error

Operating Mode Profile Velocity

Starting the Operating Mode

The operating mode is set and started in the process data channel with the output data.

Method	dmControl Bits 0 ... 6 MODE+ACTION	RefA32	RefB32
Via analog input	03_h	-	-
Via parameter	23_h	As *PVv_target*	-

Status Information

The word "driveStat" provides information on the operating mode.

bit	Name	Meaning
13	X_ADD1	0: Target velocity not reached 1: Target velocity reached
14	X_END	0: Operating mode started 1: Operating mode terminated
15	X_ERR	0: No error detected 1: Error detected

Terminating the Operating Mode

The operating mode is terminated when the motor is at a standstill and one of the following conditions is met:

- Stop caused by "Halt" or "Quick Stop"
- Stop caused by a detected error

Operating Mode Profile Position

Starting the operating mode

The operating mode is set and started in the process data channel with the output data.

Method	dmControl Bits 0 ... 6 MODE+ACTION	RefA32	RefB32
Absolute	01_h	As *PPv_target*	As *PPp_target*
Relative with reference to the currently set target position	21_h	As *PPv_target*	As *PPp_target*
Relative with reference to the current motor position	41_h	As *PPv_target*	As *PPp_target*

Status Information

The word "driveStat" provides information on the operating mode.

bit	Name	Meaning
13	X_ADD1	0: Target position not reached 1: Target position reached
14	X_END	0: Operating mode started 1: Operating mode terminated
15	X_ERR	0: No error detected 1: Error detected

Terminating the Operating Mode

The operating mode is terminated when the motor is at a standstill and one of the following conditions is met:

- Target position reached
- Stop caused by "Halt" or "Quick Stop"
- Stop caused by a detected error

Operating Mode Homing

Starting the Operating Mode

The operating mode is set and started in the process data channel with the output data.

Method	dmControl Bits 0 ... 6 MODE+ACTION	RefA32	RefB32
Position setting	06_h	-	As *HMp_setP*
Reference movement	26_h	As *HMmethod*	-

Status Information

The word "driveStat" provides information on the operating mode.

bit	Name	Meaning
13	X_ADD1	Reserved
14	X_END	0: Operating mode started 1: Operating mode terminated
15	X_ERR	0: No error detected 1: Error detected

Terminating the Operating Mode

The operating mode is terminated when the motor is at a standstill and one of the following conditions is met:

- Homing successful
- Stop caused by "Halt" or "Quick Stop"
- Stop caused by a detected error

Operating Mode Motion Sequence

Starting the Operating Mode

The operating mode is set and started in the process data channel with the output data.

Method	dmControl Bits 0 ... 6 MODE+ACTION	RefA32	RefB32
Start sequence	$1D_h$	Data set number	Value 1: Use data set number
Start individual data set	$3D_h$	Data set number	-

Status Information

The word "driveStat" provides information on the operating mode.

bit	Name	Meaning
13	X_ADD1	1: End of a sequence
14	X_END	0: Operating mode started 1: Operating mode terminated
15	X_ERR	0: No error detected 1: Error detected

Terminating the Operating Mode

The operating mode is terminated when the motor is at a standstill and one of the following conditions is met:

- Individual data set terminated
- Individual data set of a sequence terminated (waiting for transition condition to be fulfilled)
- Sequence terminated
- Stop caused by "Halt" or "Quick Stop"
- Stop caused by a detected error

Diagnostics and Troubleshooting

Fieldbus Communication Error Diagnostics

Verifying Connections

A properly operating fieldbus is essential for evaluating status and error messages.

If the product cannot be addressed via the fieldbus, first verify the connections.

Verify the following connections:

- System power supply
- Supply connections
- Fieldbus cables and wiring
- Fieldbus connection

Fieldbus Function Test

If the connections are correct, verify that you can address the product on the fieldbus.

Fieldbus Test

Fieldbus Function Test

If the connections are correct, check the settings for the fieldbus addresses. After correct configuration of the transmission data, test fieldbus mode.

In addition to the master, a bus monitor can be installed that, as a passive device, displays messages.

- Switch the supply voltage of the drive system off and on.
- Observe the network messages that are generated briefly after the supply voltage is switched on. A bus monitor can be used to record the elapsed time between messages and the relevant information in the messages.

Possible Errors: Addressing

If it is impossible to connect to a device, verify the following:

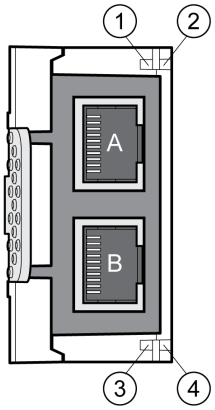
- Addressing: Each network device must have a unique IP address.

Fieldbus Status LEDs

Overview

The status of the module is indicated by four LEDs.

Overview of the LEDs at the module



1 Network activity interface A

2 Module status

3 Network activity interface B

4 Network status

Network Activity LED 1 and LED 3

The table below shows the meaning of the flashing signals for network activity.

Color	Status	Meaning
-	Off	No connection
Green	On	Connection with 100 MB/s
Yellow	On	Connection with 10 MB/s
Green	Flashing	Activity with 100 MB/s
Yellow	Flashing	Activity with 10 MB/s

Module Status LED 2

The table below shows the meaning of the flashing signals for the module status.

Color	LED	Meaning for Modbus TCP
-	Off	No IP address or no power supply
Green/red	Flashing	Start-up
Green	On	Ready for operation
Green	Flashing	Not ready (no connection, ...)
Red	Flashing	Recoverable error
Red	On	Irrecoverable error

Network Status LED 4

The table below shows the meaning of the flashing signals for the network status.

Color	LED	Meaning for Modbus TCP
-	Off	No IP address or no power supply
Green/red	Flashing	Start-up
Green	On	At least 1 port is connected and the IP address has been set

Color	LED	Meaning for Modbus TCP
Green	Flashing 3 times	No connection, IP address has been set
Green	Flashing 4 times	IP address conflict
Green	Flashing 5 times	BOOTP or DHCP active

Error Indication

Asynchronous Errors

Asynchronous errors are triggered by internal monitoring (for example, temperature) or by external monitoring (for example, limit switch). An error response is initiated if an asynchronous error is detected.

Asynchronous errors are indicated in the following way:

- Transition to operating state **7** Quick Stop Active or to operating state **9** Fault.
- Information in the words "driveStat", "mfStat", "motionStat" and "driveInput" during I/O scanning, see I/O Scanning - Input, page 16
- Error number is written to parameter *_LastError*

The parameters *_LastError* or *_LastWarning* can be used in the input mapping for I/O scanning. This way, error numbers are easy to read out.

Modbus Response

Depending on the type of processing, two types of Modbus responses are possible:

- Positive Modbus response
  - The "Function Code" in the Modbus response corresponds to the "Function Code" in the Modbus request.
- Negative Modbus response
  - The client receives pertinent information on error detection during processing;
  - The "Function Code" in Modbus response corresponds to the "Function Code" in the Modbus request + 80_h.
  - The "Exception Code" indicates the cause of the error.

If a syntactically incorrect Modbus PDU (Protocol Data Unit) is transmitted, the connection is terminated. In the case of other other error, a negative Modbus response is sent.

Exception Code	Modbus Name (as per Modbus specifications)	Description
01	Illegal Function	The "Function Code" cannot be processed by the server.
02	Illegal Data Address	Depends on the Modbus request
03	Illegal Data Value	Depends on the Modbus request
04	Server Failure	The server was unable to properly terminate processing.
05	Acknowledge	The server has accepted the Modbus request. However, the execution takes a relatively long time. The server therefore only returns an acknowledgement confirming receipt of the service request.
06	Server Busy	The server was unable to accept the Modbus request. It is the responsibility of the application on the client to determine whether and when to re-send the request.

Exception Code	Modbus Name (as per Modbus specifications)	Description
0A	Gateway Problem	The gateway path is unavailable.
0B	Gateway Problem	The targeted device does not respond. The gateway generates this error.

Glossary

C

CIP:

Common **I**ndustrial **P**rotocol, general specification for communication between fieldbus devices.

Client:

First transmitter, then recipient of fieldbus messages in the client-server relationship. Starts transmission with a transmission to the server; the reference point is the server object dictionary.

D

DOM:

Date **o**f **m**anufacturing: The nameplate of the product shows the date of manufacture in the format DD.MM.YY or in the format DD.MM.YYYY. For example:

31.12.19 corresponds to December 31, 2019

31.12.2019 corresponds to December 31, 2019

E

Error class :

Classification of errors into groups. The different error classes allow for specific responses to errors, for example by severity.

Error:

Discrepancy between a detected (computed, measured or signaled) value or condition and the specified or theoretically correct value or condition.

F

Factory setting:

Factory settings when the product is shipped

Fault reset:

A function used to restore the drive to an operational state after a detected error is cleared by removing the cause of the error so that the error is no longer active.

Fault:

Fault is a state that can be caused by an error. Further information can be found in the pertinent standards such as IEC 61800-7, ODVA Common Industrial Protocol (CIP).

I

Input:

Output and input refer to the direction of data transmission from the perspective of the master. Input: Status messages from the slave to the master, see also Output.

M

Master:

Active bus device that controls the data traffic on the network.

O

Output:

Output and input refer to the direction of data transmission from the perspective of the master. Output: Commands from the master to the slave, see also Input.

P

Parameter :

Device data and values that can be read and set (to a certain extent) by the user.

Persistent:

Indicates whether the value of the parameter remains in the memory after the device is switched off.

Q

Quick Stop:

The Quick Stop function can be used for fast deceleration of a movement in the case of an error or via a command.

U

User-defined unit:

Unit whose reference to motor movement can be determined by the user via parameters.

Index

I

M

O

P

Q

Schneider Electric
35 rue Joseph Monier
92500 Rueil Malmaison
France

+ 33 (0) 1 41 29 70 00

www.se.com

As standards, specifications, and design change from time to time, please ask for confirmation of the information given in this publication.

© 2021 – Schneider Electric. All rights reserved.

0198441113843.01