LXM32M

Ethernet TCP/IP Module (Protocol EtherNet/IP)

User Guide

Original instructions

0198441113802.01
06/2021

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.

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 EtherNet/IP) 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 EtherNet/IP) - User Guide (this user guide)	0198441113802 (eng) 0198441113803 (fre) 0198441113801 (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.

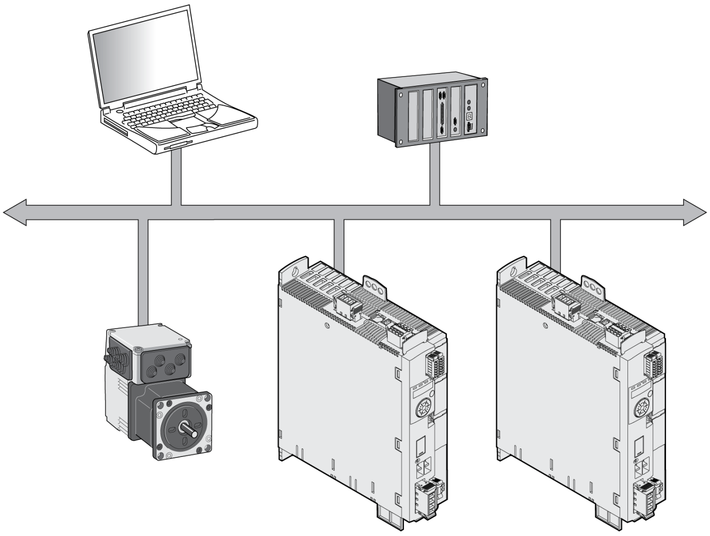
Introduction

Fieldbus Devices on the EtherNet/IP Network

Overview

EtherNet/IP is a fieldbus based on TCP and UDP. EtherNet/IP extends Ethernet by an advanced industrial protocol (CIP, Common Industrial Protocol) as an application layer for automation applications. Products from different manufacturers can be networked without the need for special interface adaptation. The majority of the required network components correspond to the Ethernet components used in the PC world.

Different products with an EtherNet/IP interface can be operated in the same fieldbus. EtherNet/IP provides a common basis for interchanging commands and data between the network devices.



Features

The product supports the following functions via EtherNet/IP:

- 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.

EtherNet/IP Fieldbus

General

ODVA

The ODVA is the proprietor of the specifications for the EtherNet/IP network and EtherNet/IP data terminal equipment. For more information on the ODVA see:

http://www.odva.org

Number of Nodes

The number of nodes in an EtherNet/IP network depends on the subnet size and on whether or not a CIP router is used. For example, 254 nodes are possible in a class C subnet.

Cable Length

The maximum cable length is 100 m (328 ft) between EtherNet/IP terminal points and 90 m (295 ft) between infrastructure components. However, interference in industrial environments may require you to use shorter cables.

Drive Profiles

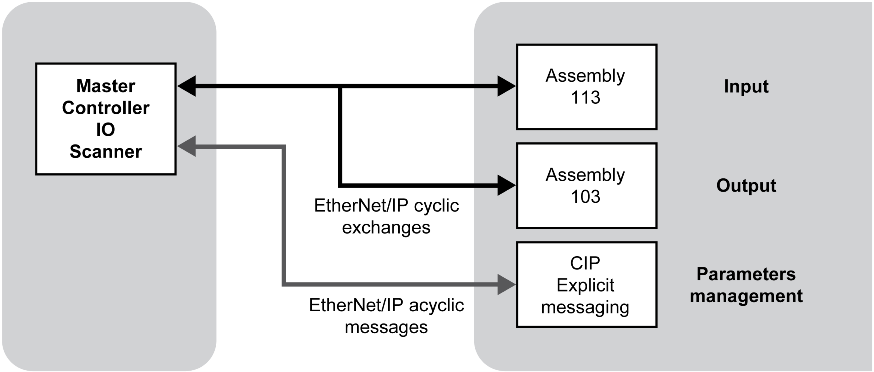
The product supports the following drive profiles:

- “Drive Profile Lexium” (manufacturer-specific)

Communication Means

The product supports the following communication means:

Overview of communication means


Master Controller IO Scanner
EtherNet/IP cyclic exchanges
EtherNet/IP acyclic messages
Assembly 113
Assembly 103
CIP Explicit messaging
Input
Output
Parameters management

- Class 1 connections via assemblies:
  - Maximum number of connections: 4
  - Supported actual packet intervals (API): 2 ... 3200 ms
  - Supported connection types product to PLC (Target to Originator): point-to-point, Multicast
  - Supported connection types PLC to product (Originator to Target): point-to-point
  - Supported triggers: cyclic, state transition
- Class 3 connections via Explicit Messages:
  - Maximum number of connections: 16
  - Supported actual packet intervals (API): 10 ... 10000 ms
  - Supported connection types product to PLC (Target to Originator): point-to-point
  - Supported connection types PLC to product (Originator to Target): point-to-point

The product identifies itself as CIP “Generic Device” (Device Type = 0 hex).

Data Link Layer

The EtherNet/IP data link layer uses the transmission mechanisms as per IEEE 802.3 Standard Ethernet specification (edition 2002).

Physical Layer

Industrial EtherNet/IP specifies minimum requirements in terms of ambient conditions, cabling, and connectors, based on IEC, ANSI, TIA, and EIA standards.

The connectors required for Industrial EtherNet/IP include M12-4 connectors, D-coded. Use CAT5e or CAT6 cables for Industrial EtherNet/IP.

Copper media may be used only for distances up to 100 m (328 ft).

Object Class, Instance, Attribute, Service

The EtherNet/IP approach is object-oriented. CIP defines object classes; one or more instances (objects) can be derived from such object classes. The attributes of an object class or the instance derived from it contain the various parameters. Services are actions that are possible with these attributes.

Example

Class	Instance	Attribute	Attribute value	Service
Motor data	Motor_1	MaxSpeed	4000 RPM	Get
Motor data	Motor_2	MaxSpeed	3000 RPM	Get

CIP Object Model

The following object classes from the CIP object model are available:

Object class	Class ID	Instance ID
Identity Object	1 (01 hex)	1
Message Router Object	2 (02 hex)	1
Assembly Object	4 (04 hex)	103 = Output Assembly, consuming 113 = Input Assembly, producing
Connection Manager Object	6 (06 hex)	1= Explicit Message
Vendor-specific objects	101 ... 163 (65 ... A3 hex)	1
Port Object	244 (F4 hex)	1
TCP/IP Object	245 (F5 hex)	1
Ethernet Link Object	246 (F6 hex)	2

The vendor-specific object class IDs 101 to 163 correspond to the object dictionary (class ID = object group + 100). The attributes of a class correspond to the subindex entry within the object group.

Communication Model

EtherNet/IP uses the producer-consumer communication model. The nodes monitor the bus as to whether a data packet with the Identifier they support is available. Data packets that are sent by producers can only be received by the consumers of these packets.

Groups of Connections

EtherNet/IP is a connection-oriented network. Connections must be established and managed between two nodes. There are 4 connection groups with different priorities:

Group 1	Top-priority process data (highest priority)
Group 2	For master-slave connections
Group 3	For Explicit Messages
Group 4	Reserved group (lowest priority)

EDS File (Electronic Data Sheet)

The EDS file contains device-specific and vendor-specific descriptions of the parameters for a drive. The EDS file also contains the fieldbus-specific communication parameters.

Messaging and Message Types

General

EtherNet/IP is based on TCP/IP and UDP/IP technologies that are used without modification. TCP/IP is used for the transmission of Explicit Messages while UDP/IP is used for I/O Messages.

Messaging and Message Types

EtherNet/IP defines several message types for communication. The drive uses the message types “Explicit Message” and “I/O Message”.

Explicit Messages

Explicit Messaging connections are point-to-point connections between two network nodes that are used for transactions of the type request - response. The data field of Explicit Messages contains both protocol data and application-specific commands.

An Explicit Message (EtherNet/IP-specific or vendor-specific) is used to read or write an individual parameter.

The parameter is accessed by means of Class.Instance.Attribute as per CIP.

I/O Messages

I/O Messages, also referred to as Implicit Messages, are transmitted via UDP/IP. I/O Message connections are often established as One-to-Many relationships in the producer-consumer multicast model of EtherNet/IP. The data fields of I/O Messages contain no protocol information, but only time-critical I/O data. I/O Messages are considerably smaller than Explicit Messages, thus allowing for faster processing. These messages are used to transport application-specific I/O data over the network at regular intervals. The meaning of the data is defined at the time the connection is established. I/O Messages can contain so-called Assemblies of several parameters that can be transmitted with a single message. The parameters for configuring EtherNet/IP communication are described in chapter Commissioning, page 24.

Command Processing: Transmit Data and Receive Data

The master sends a command to the drive system (slave) to execute a motion command, activate functions or request information from the slave. The slave executes the command and acknowledges it with a response message that may contain an error message if an error has been detected.

The master can send new commands as soon as it has received acknowledgment concerning the current command. Acknowledgment information and error messages are included in the transmitted data in bit-coded form. The master must then continuously monitor for completion of the command by evaluating the acknowledgment from the slave. I/O messages are a special case. I/O messages are not acknowledged by the slave.

EtherNet/IP Communication

Communication via I/O Messages

Overview

An I/O Message is used for realtime exchange of process data. Transmission is very fast because the data is sent without administration data and a transmission acknowledgment from the recipient is not required.

The master can control the operating states of the slave by means of I/O Message, for example, enable and disable the power stage, trigger a Quick Stop, reset detected errors and activate operating modes.

Changing operating states and activating operating modes must be done separately. An operating mode can only be activated in the operating state “Operation Enabled”.

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

Assembly

I/O Messages contain a collection (Assembly) of different parameters that are transmitted with a single message.

The following Assemblies are available:

- Output Assembly, instance 103
- Input Assembly, instance 113

Polled I/O Connection

The Assemblies are used in a Polled I/O Connection. A Polled I/O Connection is initiated by the master with a Poll Command. The Slave responds with a Poll Response.

Output Assembly, Instance 103

Overview

The table below shows the memory image for Output Assembly data. See the user guide of the drive for a description of the parameters.

Byte	Name	Parameter address CIP
0 ... 3	PCTRLms	-
4 ... 7	PVms	-
8 ... 9	dmControl	-
10 ... 13	RefA32	-
14 ... 17	RefB32	-
18 ... 21	Ramp_v_acc	CIP 106.1.10
22 ... 25	Ramp_v_dec	CIP 106.1.11
26 ... 29	EthOptMapOut1	CIP 168.1.46
30 ... 33	EthOptMapOut2	CIP 168.1.47
34 ... 37	EthOptMapOut3	CIP 168.1.48

Double Words “PCTRLms” and “PVms”

The two double words "PCTRLms" and "PVms" are used to read and write parameters, see 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 29 and Starting and Changing an Operating Mode, page 31 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 two 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. The user guide of the drive provides descriptions of the parameters EthOptMapOut1 ... EthOptMapOut3 which explain parameter mapping.

Input Assembly, Instance 113

Overview

The table below shows the memory image for Input Assembly data. See the user guide of the drive for a description of the parameters.

Byte	Name	Parameter address CIP
0 ... 3	PCTRLsm	-
4 ... 7	PVsm	-
8 ... 9	driveStat	-
10 ... 11	mfStat	-
12 ... 13	motionStat	-
14 ... 15	driveInput	-
16 ... 19	_p_act	CIP 130.1.13
20 ... 23	_v_act	CIP 130.1.32
24 ... 25	_I_act	CIP 130.1.3
26 ... 29	EthOptMapInp1	CIP 168.1.52
30 ... 33	EthOptMapInp2	CIP 168.1.53
34 ... 37	EthOptMapInp3	CIP 168.1.54

Double Words “PCTRLsm” and “PVsm”

The two double words "PCTRLsm" and "PVsm" are used to read and write parameters, see 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 29.

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 30.

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. The user guide of the drive provides descriptions of the parameters EthOptMapInp1 ... EthOptMapInp3 which explain parameter mapping.

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" must contain the address of the object class.

Example parameter *_prgNoDEV*: Parameter address **101**.1.1

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

Byte “Subindex”

The byte "Subindex" must contain the address of the attribute.

Example parameter *_prgNoDEV*: Parameter address 101.1.**1**

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
03 hex	No request
13 hex	Read request
23 hex	Write request (word)
33 hex	Write request (double word)

Receive data:

Ctrl	Function
03 hex	Request not yet completed
13 hex	Read request or write request successfully completed (word)
23 hex	Read request or write request successfully completed (double word)
73 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 03 hex to 13 hex. Write requests requests are only executed by the slave if the value changes from 03 hex to 23 hex or to 33 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 101.1.1.

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
13 hex	01 hex	00 65 hex	00 00 00 00 hex

Receive data:

Byte Ctrl	Byte Subindex	Word Index	Double word ParameterValue
23 hex	01 hex	00 65 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 103.1.50. The value of the parameters is to be modified to 222 (DE hex).

Before the slave can accept a new request, the value 03 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 73 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
33 hex	32 hex	00 67 hex	00 00 00 DE hex

Receive data:

Byte Ctrl	Byte Subindex	Word Index	Double word ParameterValue
73 hex	32 hex	00 67 hex	00 00 11 01 hex

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

Handshake via the Bit "MT" (Mode Toggle)

Mode Toggle

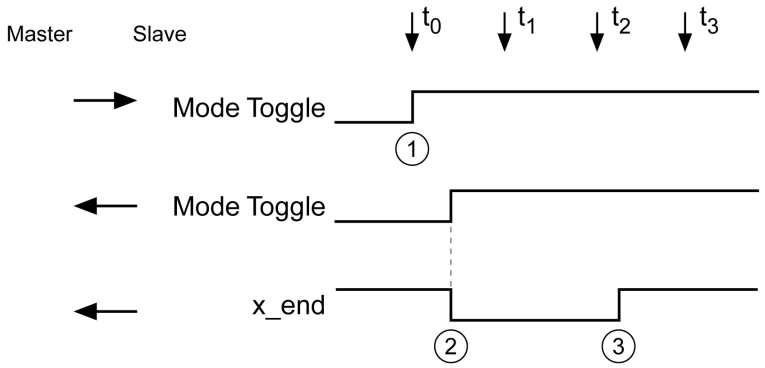
Synchronized processing can be carried out with the transmit data in the word "dmControl" bit "MT" (Mode Toggle) and the receive data in the word "mfStat", bit "ME" (Mode Error) and bit "MT" (Mode Toggle). Synchronized processing means that the master waits for feedback messages from the slave so it can respond appropriately.

The bit "MT" (Mode Toggle) is effective with a rising edge and a falling edge.

Example 1: Positioning

The master starts a movement. At points in time t_1, t_2 ..., the master verifies the responses from the slave. It waits for the end of the movement. The end is identified by bit "x_end" = 1.

Mode Toggle Handshake:



1 Master starts movement with “Mode Toggle” = 1

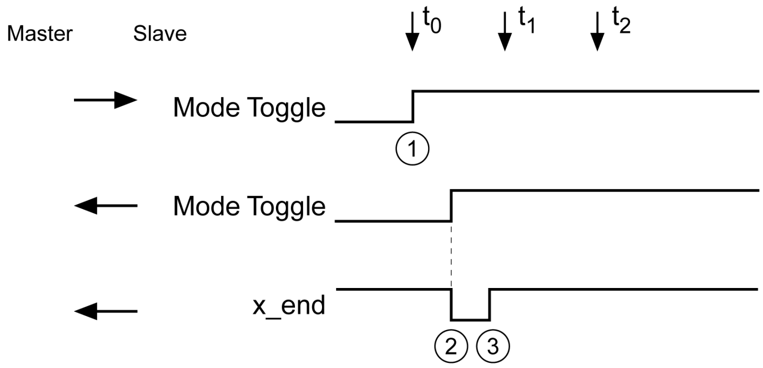
2 Slave signals that movement is running with “Mode Toggle” = 1; at the same time “x_end” = 0

3 Slave signals “movement terminated” with “x_end” = 1

Example 2: Short-Distance Movement

The master starts a movement whose duration is shorter than the request cycle of the master. At point in time t_1, the movement is already terminated. Bit “x_end” does not allow the master to detect whether the movement has already been terminated or has not yet started. However, it can identify this with the bit “MT“ (Mode Toggle).

Mode Toggle Handshake, short-distance movement:



1 Master starts movement with “Mode Toggle” = 1

2 Slave signals that movement is running with “Mode Toggle” = 1; at the same time “x_end” = 0

3 Slave signals “movement terminated” with “x_end” = 1

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.

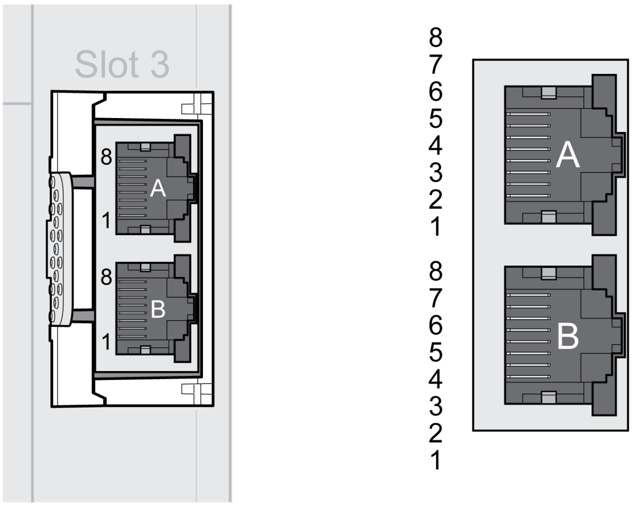
Install the module according to the instructions in the user guide of the drive.

Cable Specifications

Shield:	Required
Twisted Pair:	Required
PELV:	Required
Cable composition:	8 x 0.25 mm^2 (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
- EtherNet/IP master
- Lexium 32M Drive User Guide and this user guide, LXM32M Ethernet TCP/IP Module (Protocol EtherNet/IP) 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.

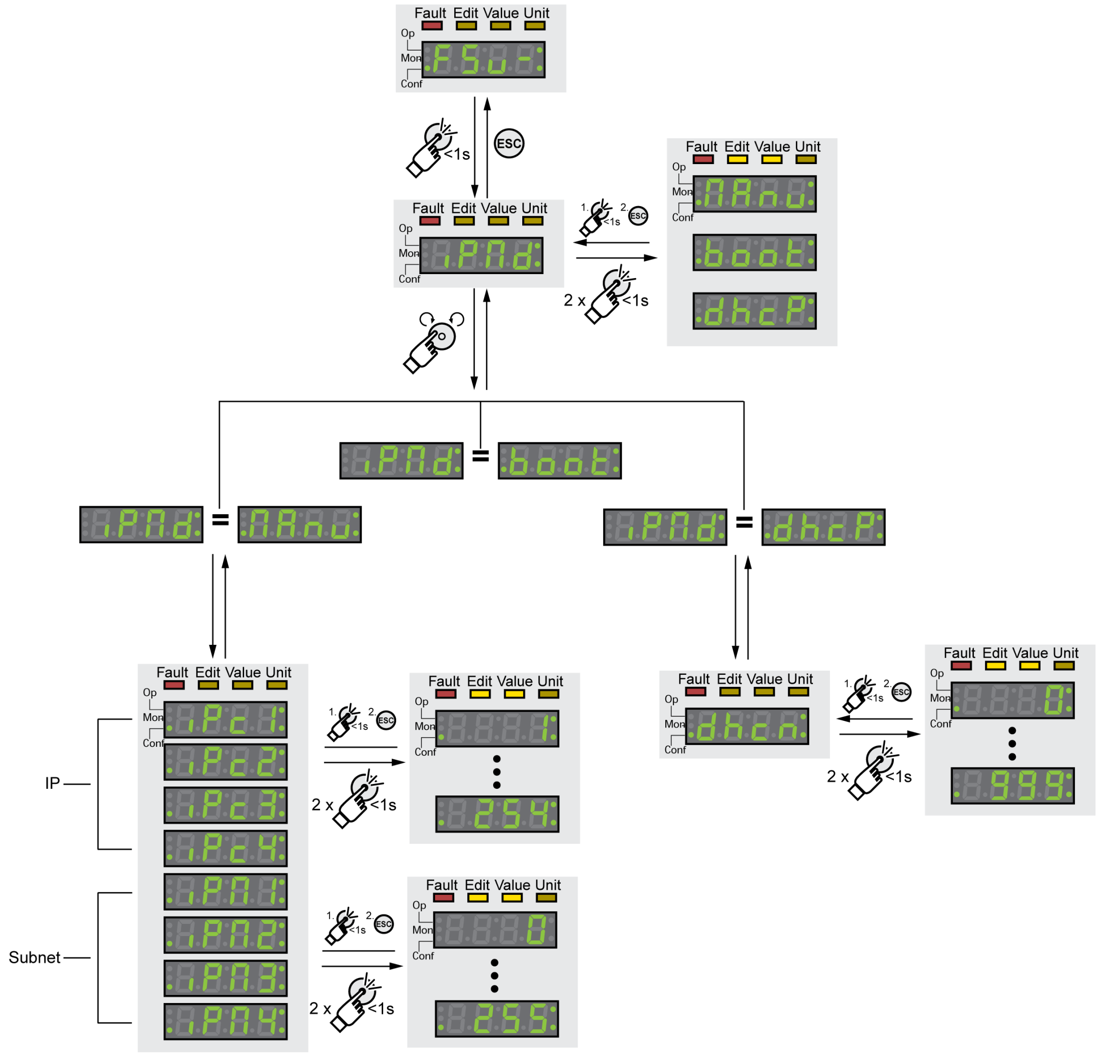
- Disconnect the drive from the fieldbus during commissioning in order to avoid conflicts by simultaneous access.
- Power on the controller supply.

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* (iPNd).

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 3044:5$_h$ Modbus 17418 Profibus 17418 CIP 168.1.5 ModbusTCP 17418 EtherCAT 3044:5$_h$ 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- iPc1	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 3044:7$_h$ Modbus 17422 Profibus 17422 CIP 168.1.7 ModbusTCP 17422 EtherCAT 3044:7$_h$ PROFINET 17422
EthIPmodule2 ConF → CoM- iPc2	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 3044:8$_h$ Modbus 17424 Profibus 17424 CIP 168.1.8 ModbusTCP 17424 EtherCAT 3044:8$_h$ PROFINET 17424
EthIPmodule3 ConF → CoM- iPc3	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 3044:9$_h$ Modbus 17426 Profibus 17426 CIP 168.1.9 ModbusTCP 17426 EtherCAT 3044:9$_h$ 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 ConF → CoM- iPc4	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 ConF → CoM- iPM1	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 ConF → CoM- iPM2	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 ConF → CoM- iPM3	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 ConF → CoM- iPM4	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* = boot)

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

Assignment of the Network Address via DHCP (*EthIpMode* = d h c P)

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 d h c n.

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

  Example: LEXIUM_SERVO**001**
- Set the new device name of the drive in the configuration of the DHCP server and verify correctness.

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

  - If the device name is modified after the "First Setup", the unique number set via d h c n is only taken into account if the device name (without counting the unique number) consists of 12 characters.
  - If the device name is modified after the "First Setup", set the modified device name of the drive in the configuration of the DHCP server and verify correctness.

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.

Operating States and Operating Modes

Operating States

Indication of the Operating State via Fieldbus

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

driveStat

15	14	13	12	11	10	9	8
X_ERR	X_END	X_ADD1	-	-	QS	RF	HALT

7	6	5	4	3	2	1	0
WARN	ERROR	-	-	STATE			

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.

dmControl

15	14	13	12	11	10	9	8
CU	CH	SH	-	FR	QS	EN	DS

7	6	5	4	3	2	1	0
MT	ACTION		MODE				

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.

mfStat

15	14	13	12	11	10	9	8
-	-	-	-	CAP2 1	CAP2 0	CAP1 1	CAP1 0

7	6	5	4	3	2	1	0
MT	ME	DE	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.

dmControl

15	14	13	12	11	10	9	8
CU	CH	SH	-	FR	QS	EN	DS

7	6	5	4	3	2	1	0
MT	ACTION		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	ACTION	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 should 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 and Parameterization

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

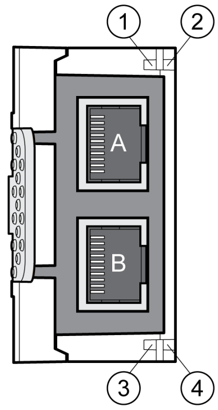
- Addressing: Each network device must have a unique IP address.
- Parameterization: "Vendor ID" and "Product Code" must match the values stored in the EDS file.

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 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 for the module status.

Color	LED	Meaning for EtherNet/IP
-	Off	No power supply
Green/red	Flashing	Start-up
Green	Flashing	Module is not configured or scanner is Idle
Green	On	Ready for operation
Red	Flashing	Recoverable error
Red	On	Irrecoverable error

Network Status LED 4

The table below shows the meaning for the network status.

Color	LED	Meaning for EtherNet/IP
-	Off	No IP address or no power supply
Green/red	Flashing	Start-up
Green	On	Connected
Green	Flashing	No connection

Color	LED	Meaning for EtherNet/IP
Red	Flashing	Timeout
Red	On	IP address conflict

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 word "driveStat" (bit 6 "ERROR")
- Error number is written to parameter *_LastError*

Synchronous Errors

Synchronous errors are errors that are detected immediately in response to a fieldbus command. They comprise, for instance:

- Error during execution of an action command or control command
- Parameter value outside the permissible value range
- Invalid action command or control command during processing
- Access to indeterminable parameter

Synchronous Errors - Explicit Error Response

If an Explicit Request message cannot be processed by the slave, the master receives an error message in the associated Explicit Response. This response message contains 2 bytes:

- General Error Code
- Additional Error Code

Error codes can be read with object 100.1.1. If the general error code has the value =1F hex, the field "additional error code" contains vendor-specific error numbers in coded form.

Synchronous Errors - Response During I/O Connection

The slave responds to an incorrect I/O command in the next I/O response by setting bit 6 (ME) in the word "mfStat". This does not interrupt the current process. To determine the cause of the error, the master can read the error number with the object 100.1.1.

The error indication is reset when the next valid data assembly is transmitted.

Synchronous Errors - Table of General Error Codes

The error codes that can be contained in the "General Error Code" field are listed in the following table:

Error code	Name of general error	Meaning
00 hex	Success	The service was successfully executed by the specified object.
05 hex	Path destination unknown	The path refers to an object class, an instance or a structure element that is indeterminable or not contained in the processing node. Path processing is terminated if an error is detected that is due to an indeterminable path destination.
09 hex	Invalid attribute value	Invalid attribute data was detected.
0C hex	Object state conflict	The object cannot execute the requested service in its current mode/state.
0E hex	Attribute not settable	A request to modify an attribute that cannot be set was received.
0F hex	Privilege denied	Verification of an authorization/privilege was unsuccessful.
10 hex	Device state conflict	The requested service cannot be executed in the current mode/state of the device.
13 hex	Not enough data	The service did not deliver enough data to execute the specified operation.
14 hex	Attribute not supported	The attribute specified in the request is not supported.
15 hex	Too much data	The service delivered more data than expected.
1F hex	Vendor specific error	A vendor-specific error was detected. The vendor-specific error code can be read with object 100.1.1. The vendor-specific error codes can be found in the user guide of the drive.

Object Dictionary

Classes

Overview

This chapter describes the communication parameters supported by the product.

The following classes are supported:

- Identity Object (class 1)
- Message Router Object (class 2)
- Assembly Object (class 4)
- Connection Manager Object (class 6)
- Port Object (class 244)
- TCP/IP Interface Object (class 245)
- Ethernet Link Object (class 246)

Acronyms

NV: Persistent (Non-Volatile)

V: Not persistent (Volatile)

RO: Read Only

RW: Read Write

Vendor-Specific Objects

The vendor-specific objects (parameters) are described in the user guide of the drive.

Structure of the address of an object:

Class.Instance.Attribute

Identity Object (class 1)

Description

The object contains the identification data of the product.

For more details on the object see also ODVA standard "The CIP Networks Library Volume 1: Common Industrial Protocol", chapter "Identity Object".

Services

Class services:

ID	Name	Description
01 hex	Get_Attribute_All	Returns a predefined listing of this object’s attributes
0E hex	Get_Attribute_Single	Returns the contents of the specified attribute

Instance services:

ID	Name	Description
01 hex	Get_Attribute_All	Returns a predefined listing of this object's attributes
0E hex	Get_Attribute_Single	Returns the contents of the specified attribute
05 hex	Reset	Restart of the drive

Class Attributes

ID	Access	Name	Data type	Values	Description
1	Get	Revision	UINT	-	Revision
2	Get	Max instance	UINT	00 01 hex	Greatest currently existing instance number of an object derived from this class
3	Get	Number of instances	UINT	00 01 hex	Number of instances
4	Get	Optional attribute list	-	-	-
6	Get	Max ID of class attributes	UINT	00 07 hex	Greatest currently existing attribute number of a class
7	Get	Max ID of instance attributes	UINT	00 07 hex	Greatest currently existing attribute number of an instance derived from this class

Instance Attributes

ID	Access	Name	Data type	Values	Description
1	Get	Vendor ID	UINT	243 (F3 hex)	Unique vendor number
2	Get	Device type	UINT	0 (Generic Device)	Device family
3	Get	Product code	UINT	0A 04 hex	Unique device type
4	Get	Revision	STRUCT: USINT USINT	-	Revision of module
5	Get	Status	WORD	-	Summarized device status
6	Get	Serial number	UDINT	-	CIP serial number
7	Get	Product name	SHORT_ STRING	Lexium 32	Device name in text form

Message Router Object (class 2)

Description

The object Message Router provides a connection to address a service for an object class or instance.

For more details on the object see also ODVA standard "The CIP Networks Library Volume 1: Common Industrial Protocol", chapter "Message Router Object".

Services

Class services:

ID	Name	Description
0E hex	Get_Attribute_Single	Returns the contents of the specified attribute

Instance services:

ID	Name	Description
0E hex	Get_Attribute_Single	Returns the contents of the specified attribute

Class Attributes

ID	Access	Name	Data type	Values	Description
1	Get	Revision	UINT	-	Revision
2	Get	Max instance	UINT	00 01 hex	Greatest currently existing instance number of an object derived from this class
3	Get	Number of instances	UINT	00 01 hex	Number of instances
4	Get	Optional attribute list	-	2, 3	List of optional instance attributes used in an object class implementation
6	Get	Max ID of class attributes	UINT	00 07 hex	Greatest currently existing attribute number of a class
7	Get	Max ID of instance attributes	UINT	00 03 hex	Greatest currently existing attribute number of an instance derived from this class

Instance Attributes

ID	Access	Name	Data type	Values	Description
2	Get	Number available	UINT	00 10 hex	Maximum connections
3	Get	Number active	UINT	-	Number of the currently active system connection to this object.

Assembly Object (Class 4)

Description

An Assembly object is a container that contains one or more attributes of other objects. This way, multiple attributes can be transmitted from a slave or to a slave simultaneously with a single connection.

- Output Assemblies are commands from the network to the device.
- Input Assemblies are status messages from the device to the network.

The following instances of the object Assembly are implemented in the device:

ID	Type	Name	Number of bytes
103	EtherNet/IP Output Assemby, consuming	Vendor-dependent extended profile	38
113	EtherNet/IP Input Assemby, producing	Vendor-dependent extended profile	38

For more details on the object see also ODVA standard "The CIP Networks Library Volume 1: Common Industrial Protocol", chapter "Assembly Object".

Services

Class services:

ID	Name	Description
0E hex	Get_Attribute_Single	Returns the contents of the specified attribute

Instance services:

ID	Name	Description
0E hex	Get_Attribute_Single	Returns the contents of the specified attribute
10 hex	Set_Attribute_Single	Sets the contents of the specified attribute

Class Attributes

ID	Access	Name	Data type	Values	Description
1	Get	Revision	UINT	-	Revision
2	Get	Max instance	UINT	00 C7 hex	Greatest currently existing instance number of an object derived from this class
3	Get	Number of instances	UINT	00 04 hex	Number of instances
6	Get	Max ID of class attributes	UINT	00 07 hex	Greatest currently existing attribute number of a class
7	Get	Max ID of instance attributes	UINT	00 04 hex	Greatest currently existing attribute number of an instance derived from this class

Instance Attributes

ID	Access	Name	Data type	Values	Description
3	Get	Data	Array USINT	See Communication via I/O Messages, page 14	Data from module
4	Get	Size	UINT	-38	Data size of instance

Connection Manager Object (class 6)

Description

The Connection Manager object allocates and manages the internal resources associated with both I/O-Messaging and Explicit Messaging connections.

For more details on the object see also ODVA standard "The CIP Networks Library Volume 1: Common Industrial Protocol", chapter "Connection Manager Object".

Services

Class services:

ID	Name	Description
0E hex	Get_Attribute_Single	Returns the contents of the specified attribute

Instance services:

ID	Name	Description
0E hex	Get_Attribute_Single	Returns the contents of the specified attribute
54 hex	Forward_Open	Opens a connection, maximum data size 511 bytes.
4E hex	Forward_Close	Closes a connection

Class Attributes

ID	Access	Name	Data type	Values	Description
1	Get	Revision	UINT	-	Revision
2	Get	Max instance	UINT	00 01 hex	Greatest currently existing instance number of an object derived from this class
3	Get	Number of instances	UINT	00 01 hex	Number of instances
4	Get	Optional attribute list	-	1,2,3,4,5,6,7,8	List of optional instance attributes used in an object class implementation
6	Get	Max ID of class attributes	UINT	00 07 hex	Greatest currently existing attribute number of a class
7	Get	Max ID of instance attributes	UINT	00 08 hex	Greatest currently existing attribute number of an instance derived from this class

Instance Attributes

ID	Access	Name	Data type	Values	Description
1	Get	Open Requests	UINT	-	Number of the received Forward Open Service Request
2	Get	Open Format Rejects	UINT	-	Number of the rejected Forward Open Service Request (incorrect format)
3	Get	Open Resource Rejects	UINT	-	Number of the rejected Forward Open Service Request (busy)
4	Get	Open Other Rejects	UINT	-	Number of the rejected Forward Open Service Request (other reason)
5	Get	Close Requests	UINT	-	Number of the received Forward Open Service Request
6	Get	Close Format Rejects	UINT	-	Number of the rejected Forward Close Service Request (incorrect format)
7	Get	Close Other Rejects	UINT	-	Number of the rejected Forward Close Service Request (busy)
8	Get	Connection Timeouts	UINT	-	Number of the rejected Forward Close Service Request (other reason)

Port Object (class 244)

Description

The object Port describes the available CIP ports.

For more details on the object see also ODVA standard "The CIP Networks Library Volume 1: Common Industrial Protocol", chapter "Port Object Class Definition".

Services

Class services:

ID	Access	Description
01 hex	Get_Attribute_All	Returns a predefined listing of this object's attributes.
0E hex	Get_Attribute_Single	Returns the contents of the specified attribute.

Instance services:

ID	Access	Description
01 hex	Get_Attribute_All	Returns a predefined listing of this object's attributes.
0E hex	Get_Attribute_Single	Returns the contents of the specified attribute.

Class Attributes

ID	Access	Name	Data type	Values	Description
1	Get	Revision	UINT	-	Revision
2	Get	Max Instance	UINT	00 01 hex	Greatest currently existing instance number of an object derived from this class
3	Get	Number of Instance	UINT	00 01 hex	Number of instances
6	Get	Maximum ID Number Class Attributes	UINT	00 09 hex	The attribute ID number of the last class attribute of the class definition implemented in the device
7	Get	Maximum ID Number Instance Attributes	UINT	00 0A hex	The attribute ID number of the last instance attribute of the class definition implemented in the device
8	Get	Entry Port	UINT	00 01 hex	Instance in Port object where the request came from.
9	Get	All Port	STRUCT: PortType PortNumber	-	-

Instance Attributes

ID	Access	Name	Data type	Values	Description
1	Get	Port type	DWORD	00 hex	Type of CIP port.
2	Get	Port number	DWORD	02 hex	Vendor specific numbering of port.
3	Get	Link object	STRUCT: Path Length (UINT) Padded EPATH	02 00 20 F5 24 01 hex	EPATH to Ethernet Link object.
4	Get	Port name	SHORT_ STRING	0B 45 74 68 65 72 4E 65 74 2F 49 50 hex	String that names the physical network port. Name = “EtherNet/IP”

ID	Access	Name	Data type	Values	Description
7	Get	Node Address	Padded EPATH	-	Node number of the port.
10[1]	Get	Port routing capabilities	UINT32	00 hex	Bit 0 = 1: Routing of incoming Unconnected Messaging supported. Bit 1 = 1: Routing of outgoing Unconnected Messaging supported.

(1)	With firmware version ≥V01.16 of the module.

TCP/IP Interface Object (class 245)

Description

The object TCP/IP Interface provides a mechanism to configure a device's TCP/IP network interface. Configurable items include the device's IP address, network mask and gateway address.

The object maintains link-specific counters and status information for a Ethernet 802.3 communication interface.

For more details on the object see also ODVA standard "The CIP Networks Library Volume 2: EtherNet/IP Adaptation of CIP", chapter "TCP/IP Interface Object".

Services

Class services:

ID	Access	Description
01 hex	Get_Attribute_All	Returns a predefined listing of this object's attributes.
0E hex	Get_Attribute_Single	Returns the contents of the specified attribute.

Instance services:

ID	Access	Description
01 hex	Get_Attribute_All	Returns a predefined listing of this object's attributes.
0E hex	Get_Attribute_Single	Returns the contents of the specified attribute.
10 hex	Set_Attribute_Single	Modifies a single attribute.

Class Attributes

ID	Access	Name	Data type	Values	Description
1	Get	Revision	UINT	-	Revision
2	Get	Max Instance	UINT	00 01 hex	Greatest currently existing instance number of an object derived from this class
3	Get	Number of Instance	UINT	00 01 hex	Number of instances
4	Get	Optional attribute list	-	8, 9	List of optional instance attributes used in an object class implementation
5	Get	Optional service list	-	-	-

ID	Access	Name	Data type	Values	Description
6	Get	Maximum ID Number Class Attributes	UINT	00 07 hex	The attribute ID number of the last class attribute of the class definition implemented in the device
7	Get	Maximum ID Number Instance Attributes	UINT	00 0D hex	The attribute ID number of the last instance attribute of the class definition implemented in the device

Instance Attributes

ID	Access	Name	Data type	Values	Description
1	Get	Interface status	DWORD	See table Details for Instance Attribute ID 1, page 50.	Status of the interface
2	Get	Configuration capability	DWORD	See table Details for Instance Attribute ID 2, page 50.	Obtaining configuration via BOOTP or DHCP
3	Get/Set	Configuration control	DWORD	-	0 - Configuration via non-volatile memory 1 - Configuration via BOOTP 2 - Configuration via DHCP
4	Get	Physical Link Object	DWORD	-	Path to physical link object
		Path size	UINT	00 02 hex	2 words
		Path	Padded EPATH	20 F6 24 01 hex	Ethernet Link object, instance 1 (logical segments identifying the physical link object)
5	Get/Set	Interface Configuration	-	-	TCP/IP network interface configuration
		IP Address	UDINT	-	IP address of the device
		Network Mask	UDINT	-	Network mask of the device
		Gateway Address	UDINT	-	Default gateway address
		Name Server	UDINT	-	Primary name server
		Name Server 2	UDINT	-	Secondary name server
		Domain Name	STRING	-	Default domain name
6	Get/Set	Host name	STRING	-	Host name
8	Get/Set	TTL Value	USINT	1	TTL values for EtherNet/IP Multicast packets
9	Get/Set	Mcast Config	-	-	Address configuration IP Multicast
		Alloc Control	USINT	0	0 - Standard algorithm for calculation the Multicast address 1 - Multicast address is assigned on the basis of the values of Mcast and Mcast Start Addr
		Reserved	USINT	0	-
		Num Mcast	UINT	4	Number of Multicast addresses assigned to EtherNet/IP
		Mcast Start	USINT	-	Multicast address at which the assignment starts
10[(1)]	Get/Set	SelectAcd	BOOL	1	Activates the use of ACD
11[(1)]	Get/Set	LastConflictDetected	Struct of:	-	Structure containing information about last ACD conflict

ID	Access	Name	Data type	Values	Description
		AcdActivity	USINT	-	ACD state when conflict was detected
		RemoteMAC	Array of 6 USINT	-	MAC address of remote node from ARP PDU
		ArpPdu	Array of 28 USINT	-	Copy of raw ARP PDU when conflict was detected
13(1)	Get/Set	Encapsulation Inactivity Timeout	UINT	120 (default value)	Number of seconds of inactivity before TCP connection is closed. Value 0: Disable Values 1 ... 3600: Timeout in seconds

(1) With firmware version ≥V01.16 of the module.

Details for Instance Attribute ID 1

Bit	Name	Description
0 ... 3	Interface Configuration Status	Returns the interface status Value 1: The interface attribute contains a valid configuration.
4	Mcast Pending	Indicates a pending modification of the configuration of the TTL value or the Mcast Config attribute. This bit is set as soon as the TTL value or the Mcast Config attribute are modified; it is deleted during a re-start of the device.
5	Reserved	Reserved
6	AcdStatus	Value 1: Indicates when an IP address conflict has been detected by ACD.
7	AcdFault	Value 1: Indicates when an IP address conflict has been detected by ACD and when the port can no longer be used due to this conflict.
8 ... 31	Reserved	Reserved

Details for Instance Attribute ID 2

Bit	Name	Description
0	BOOTP client	Value 1: Network configuration could be obtained via BOOTP.
1	DNS Client	Value 1: Host names could be resolved via a DNS server.
2	DHCP client	Value 1: Network configuration could be obtained via DHCP.
3	DHCP-DNS update	Value 1: The product could send its host name in a DHCP request.
4	Configuration adjustable	Value 1: The interface configuration attribute is adjustable. Some devices such as a PC workstation do not allow such a configuration via the TCP/IP interface object.
5 ... 6	Reserved	Reserved
7	AcdCapable	Value 1: The device is ACD capable.
8 ... 31	Reserved	Reserved

Ethernet Link Object (class 246)

Description

The Ethernet Link object maintains link-specific counters and status information for a Ethernet 802.3 communication interface.

Each device supports exactly one instance of the object for each Ethernet IEEE 802.3 communication interface on the module.

A request to access instance 1 of the object refers to the instance associated with the communication interface via which the request was received.

For more details on the object see also ODVA standard "The CIP Networks Library Volume 2: EtherNet/IP Adaptation of CIP", chapter "Ethernet Link Object".

Services

Class services:

ID	Name	Description
01 hex	Get_Attribute_All	Returns a predefined listing of this objects attributes
0E hex	Get_Attribute_Single	Returns the contents of the specified attribute

Instance services:

ID	Access	Description
01 hex	Get_Attribute_All	Returns a predefined listing of this objects attributes
0E hex	Get_Attribute_Single	Returns the contents of the specified attribute
10 hex	Set_Attribute_Single	Sets the contents of the specified attribute
4C hex	Get_and_Clear	Get and then clear the specified attribute (interface counters or media)

Class Attributes

ID	Access	Name	Data type	Values	Description
1	Get	Revision	UINT	-	Revision
2	Get	Max Instance	UINT	00 02 hex	Greatest currently existing instance number of an object derived from this class
3	Get	Num Instance	UINT	00 02 hex	Number of instances
6	Get	Max Class Attributes	UINT	00 07 hex	Greatest currently existing attribute number of a class
7	Get	Max Instance Attributes	UINT	00 0B hex	Greatest currently existing attribute number of an instance derived from this class

Instance Attributes

ID	Access	Name	Data type	Values	Description
1	Get	Interface Speed	UDINT	10 Mbit/sec 100 Mbit/sec	Currently used interface speed
2	Get	Interface status	DWORD	See table Details for Instance Attribute ID 2, page 53.	Status of the interface
3	Get	Physical address	ARRAY OF 6 USINTs	MAC address	MAC layer address
4	Get	Interface counters	Struct of:		
		Bytes (In)	UDINT	-	Bytes received
		Ucast Packets (In)	UDINT	-	Unicast packets received
		NUcast Packets (In)	UDINT	-	Non-unicast packets received

ID	Access	Name	Data type	Values	Description
		Discards (In)	UDINT	-	Inbound packets with indeterminable protocol
		Errors (In)	UDINT	-	Inbound packets with detected errors (without packets with indeterminable protocol)
		Unknown Protos (In)	UDINT	-	Inbound packets with indeterminable protocol
		Bytes (Out)	UDINT	-	Bytes sent
		Ucast Packets (Out)	UDINT	-	Unicast packets sent
		NUcast Packets (Out)	UDINT	-	Non-unicast packets sent
		Discards (Out)	UDINT	-	Outbound packets with indeterminable protocol
		Errors (Out)	UDINT	-	Outbound packets with detected errors (without packets with indeterminable protocol)
5	Get	Media counters	Struct of:		
		Alignment errors	UDINT	-	Received frames with a non-integral number of bytes in length.
		FCS errors	UDINT	-	Received frames with errors detected by FCS.
		Single collisions	UDINT	-	Successful received frames, but with one collision detected.
		Multiple collisions	UDINT	-	Successful received frames, but with more than one collision detected.
		SQE test errors	UDINT	0	Reserved
		Deferred transmissions	UDINT	-	Received frames with a delay.
		Late collisions	UDINT	-	Transmissions not successful due to late collisions.
		Excessive collisions	UDINT	-	Transmissions not successful due to excessive collisions.
		MAC transmit errors	UDINT	-	Sent frames with errors detected by the MAC sublayer.
		Carrier Sense errors	UDINT	-	Received frames with errors detected by Carrier Sense.
		Frame too long	UDINT	-	Received frames with exceeded frame size.
		MAC receive errors	UDINT	-	Received frames with errors detected by the MAC sublayer.
6	Get/Set	Interface control	Struct of:		
		Control bits	WORD	See table Details for Instance Attribute ID 6, page 53.	Interface control bits.
		Forced interface speed	UINT	-	Forced speed of the interface.
7	Get	Interface type	USINT	00 02 hex	value 2: Twisted pair
10	Get	Interface label	SHORT_ STRING	-	Interface 1: “Top” Interface 2: “Bottom”
11	Get	Interface capability	Struct of:		
		Capability bits	DWORD	-	Interface capabilities (except interface speed and duplex mode)
		Speed/duplex options	Struct of:		
		Speed/duplex array count	USINT	-	Number of elements
		Speed/duplex array	Array of Struct of:		

ID	Access	Name	Data type	Values	Description
		Interface speed	UINT	-	Interface speed in Mbits/sec
		Interface duplex mode	USINT	-	Value 0: half duplex Value 1: full duplex

Details for Instance Attribute ID 2

Bit	Name	Description
0	Link status	Value 0: Link inactive. Value 1: Link active.
1	Duplex mode	Value 0: Half duplex. Value 1: Full duplex.
2 ... 4	Auto-negotiation status	Value 0: Auto-negotiation in progress. Value 1: Auto-negotiation unsuccessful (interface speed and duplex mode is set to default). Value 2: Auto-negotiation unsuccessful (duplex mode is set to default). Value 3: Auto-negotiation successful. Value 4: Auto-negotiation not attempted. Forced duplex mode and interface speed is used.
5	Manuel setting require Reset	Value 0: Manuel settings are effective without a Reset of the object Identity Object (class 1). Value 1: Manuel settings need a Reset of the object Identity Object (class 1) to become effective.
6	Local hardware error detection	Value 0: No local hardware error detected. Value 1: Local hardware error detected.
7 ... 31	Reserved	Reserved

Details for Instance Attribute ID 6

Bit	Name	Description
1	Auto-negotiation setting	Value 0: Disable Auto-negotiation. Value 1: Enable Auto-negotiation.
2	Forced duplex mode	If Auto-negotiation is set to disabled: Value 0: Force half duplex. Value 1: Force full duplex.

Glossary

A

Assembly:

Various attributes are combined in one single data packet. Client and server know the structure of the packets. See also Explicit Message.

Attribute:

A single value of an object (in a network device) that can be read or written over the network. (see Class - Instance - Object - Attribute)

C

CIP:

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

Class:

DeviceNet and EtherNet/IP describes the behavior of a network node in so-called object classes. A class defines the behavior of (related) objects and consists of attributes and so-called services to work with these attributes (read/write)

for example: class vehicles, object car, attribute fuel level, service fill

(see Class - Instance - Object - Attribute)

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.

COS:

Change **Of S**tate: special I/O connection in which data is only transmitted when changes occur.

D

DOM:

Date **of 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

EDS:

(**E**lectronic **D**ata **S**heet); contains the specific properties of a product.

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.

Instance:

An actual object that is derived from a specific class. (see Class - Instance - Object - Attribute)

M

MAC ID:

Node address (MAC=Media Access Control); a unique address in the entire network.

Master:

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

O

Object dictionary:

List of the parameters, values and functions available in the device. Each entry is uniquely referenced via index (16 bit) and subindex (8 bit).

Object:

An object is a member of a specific class.

The object 'bicycle' is a member of the class 'vehicles'.

The object 'car' is a member of the class 'vehicles'.

(see Class - Instance - Object - Attribute)

ODVA:

Open **D**eviceNet **V**endor **A**ssociation.

User organization for DeviceNet and EtherNet/IP standards.

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.

S

Scanner:

Bus device that, as a master unit, controls all data transmission via the bus. Corresponds to the master.

U

User-defined unit:

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

Index

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.