Operating instructions

IO-Link master with EtherNet/IP interface
PerformanceLine
4 ports
IP 65 / IP 66 / IP 67 / IP 69K

AL1421

Firmware: 3.1.x

English

80295493 / 02 10 / 2021

Contents

1 Preliminary note 5

1.1 Legal and copyright information ... 5
1.2 Purpose of the document ... 5
1.3 Explanation of Symbols ... 6
1.4 Modification history ... 6

2 Safety instructions 7

2.1 General ... 7
2.2 Required background knowledge ... 7
2.3 Safety symbols on the device ... 7
2.4 IT security ... 8

3 Intended use 9

3.1 Permitted use ... 9
3.2 Prohibited use ... 9

4 Function 10

4.1 Parameter setting ... 10
4.2 Visual indication ... 10
4.3 EtherNet/IP ... 10
4.4 Internet of Things (IoT) ... 11
4.5 IO-Link ... 11
4.5.1 IO-Link supply ... 11
4.5.2 Digital outputs ... 11
4.6 Voltage output ... 11

5 Mounting 12

5.1 Installing the device ... 12

6 Electrical connection 13

6.1 Remarks ... 13
6.2 Connect EtherNet/IP ports ... 14
6.3 Connect IO-Link ports ... 15
6.4 Ground the device ... 16
6.5 Connect the device ... 17

7 Operating and display elements 18

7.1 Overview ... 18
7.2 LED indicators ... 19
7.2.1 Status LEDs ... 19
7.2.2 Ethernet ports ... 19
7.2.3 Voltage supply ... 20
7.2.4 IO-Link Ports (Class B) ... 20

8 Setup 21

8.1 Parameter setting options 21

9 Configuration 22

9.1 LR DEVICE 23
9.1.1 Remarks 24
9.1.2 IoT: Configure IP settings 25
9.1.3 IoT: Configuring access rights 26
9.1.4 IoT: Configure the interface to LR AGENT or LR SMARTOBSERVER 27
9.1.5 Fieldbus: Configure IP settings 28
9.1.6 Fieldbus: set the configuration mode 29
9.1.7 IO-Link ports: Activate data transfer to LR AGENT or LR SMARTOBSERVER 30
9.1.8 IO-Link ports: Set the operating mode Pin 2 (UA) 30
9.1.9 IO-Link ports: setting the operating mode Pin 4 (US) 31
9.1.10 IO-Link ports: Limit the current intensity 32
9.1.11 IO-Link ports: Set the device validation and data storage 33
9.1.12 IO-Link ports: Setting fail-safe values 34
9.1.13 Info: Show device information 34
9.1.14 Firmware: Reset device to factory settings 35
9.1.15 Firmware: Reboot the device 35
9.1.16 Configure IO-Link devices 36
9.2 ifm IoT Core 37
9.2.1 Programmers' notes 38
9.2.2 First steps 43
9.2.3 General functions 43
9.2.4 Fieldbus: Configuring IP settings 47
9.2.5 Fieldbus: Selecting the configuration mode 47
9.2.6 Fieldbus: Setting failsafe values 48
9.2.7 IoT: Configuring access rights 48
9.2.8 IoT: Configuring the LR AGENT or LR SMARTOBSERVER interface 49
9.2.9 IO-Link ports: Limiting current values 49
9.2.10 IO-Link ports: Setting the operating mode of pin 4 (US) 49
9.2.11 IO-Link ports: Configuring device validation and data storage 50
9.2.12 IO-Link ports: Configuring data transfer to LR AGENT or LR SMARTOBSERVER 52
9.2.13 IO-Link ports: Reading and writing process data 52
9.2.14 IO-Link ports: Indicating port events 55
9.2.15 IO-Link devices: Accessing parameters 55
9.2.16 IO-Link devices: Reading an writing device information 57
9.2.17 IO-Link devices: Indicating IO-Link events 57
9.2.18 Gateway: Resetting, rebooting and localising the device 57
9.2.19 Gateway: Reading device information 58
9.2.20 Gateway: Reading status and diagnostic information 58
9.2.21 Gateway: Updating the firmware 59
9.2.22 Gateway: Setting the application tag 60
9.2.23 Subscribing to notifications 61
9.2.24 Using Web Socket 64
9.2.25 Using the IoT-Core Visualizer 66
9.3 EtherNet/IP 73
9.3.1 EtherNet/IP: Programmers' notes 73
9.3.2 Registration of the EDS file 74
9.3.3 Integrate the IO-Link Master into the EtherNet/IP project 74
9.3.4 Configure connection types 75
9.3.5 Configure the IO-Link master 76
9.3.6 Configure IO-Link ports 76
9.3.7 Configure IO-Link devices 77
9.3.8 Read process data 77
9.3.9 Write process data 78
9.3.10 Read diagnostic information and events 78
9.3.11 Use acyclic services 79

10 Operation 81
10.1 Use web-based management 81

11 Maintenance, repair and disposal 82
11.1 Cleaning process 82
11.2 Updating the firmware 82
11.3 Exchanging the IO-Link device 82

12 Factory Settings 83

13 Accessories 84

14 Appendix 85
14.1 Technical data 86
14.1.1 Application 86
14.1.2 Electrical data 86
14.1.3 Inputs/outputs 91
14.1.4 Inputs 91
14.1.5 Outputs 91
14.1.6 Interfaces 92
14.1.7 Environmental conditions 92
14.1.8 Approvals / tests 92
14.1.9 Mechanical data 93
14.1.10 Electrical connection 93
14.2 EtherNet/IP 94
14.2.1 Supported connection types 94
14.2.2 Parameter data 95
14.2.3 Cyclic data 100
14.2.4 Acyclic data 112
14.2.5 Field bus objects 124
14.3 ifm IoT Core 147
14.3.1 Overview: IoT profile 148
14.3.2 Overview: IoT types 154
14.3.3 Overview: IoT services 155

15 Index 168

1 Preliminary note

Content

Legal and copyright information 5
Purpose of the document 5
Explanation of Symbols 6
Modification history 6

33203

1.1 Legal and copyright information

33117

All product names, pictures, companies or other brands used on our pages are the property of the respective rights owners:

- AS-i is the property of the AS-International Association, (→ www.as-interface.net)
- CAN is the property of the CiA (CAN in Automation e.V.), Germany (→ www.can-cia.org)
- CODESYS™ is the property of the CODESYS GmbH, Germany (→ www.codesys.com)
- DeviceNet™ is the property of the ODVA™ (Open DeviceNet Vendor Association), USA (→ www.odva.org)
- EtherNet/IP® is the property of the → ODVA™
- EtherCAT® is a registered trade mark and patented technology, licensed by Beckhoff Automation GmbH, Germany
- IO-Link® is the property of the → PROFIBUS Nutzerorganisation e.V., Germany (→ www.io-link.com)
- ISOBUS is the property of the AEF – Agricultural Industry Electronics Foundation e.V., Deutschland (→ www.aef-online.org)
- Microsoft® is the property of the Microsoft Corporation, USA (→ www.microsoft.com)
- Modbus® is the property of the Schneider Electric SE, France (→ www.schneider-electric.com)
- PROFIBUS® is the property of the PROFIBUS Nutzerorganisation e.V., Germany (→ www.profibus.com)
- PROFINET® is the property of the → PROFIBUS Nutzerorganisation e.V., Germany
- Windows® is the property of the → Microsoft Corporation, USA

1.2 Purpose of the document

34227

This document is only for device types "IO-Link master - EtherNet/IP gateway (PerformanceLine) 4 port IP 65 / IP 66 / IP 67 / IP 69K" (art. no.: AL1421).

It is part of the device and contains information about the correct handling of the product.

► Read this document before using the device.

► Keep this document during the service life of the device.

1.3 Explanation of Symbols

34171

WARNING
Warning of serious personal injury.
Death or serious irreversible injuries may result.

CAUTION
Warning of personaly injury.
Slight reversible injuries may result.

NOTICE
Warning of damage to property

Important note
Non-compliance can result in malfunction or interference

Information
Supplementary note

► ... Request for action

> ... Reaction, result

→ ... "see"

abc Cross-reference

123 Decimal number
0x123 Hexadecimal number
0b010 Binary number

[...] Designation of pushbuttons, buttons or indications

1.4 Modification history

34492

Version	Topic	Date
00	New creation of document	10 / 2020
01	Added: Derating UL	04 / 2021
02	Deleted: ifm IoT Core – DNS support	10 / 2021

2 Safety instructions

Content

General 7
Required background knowledge 7
Safety symbols on the device 7
IT security 8

28333

2.1 General

58525

- The device described is a subcomponent for integration into a system. The manufacturer is responsible for the safety of the system. The system manufacturer undertakes to perform a risk assessment and to create documentation in accordance with legal and normative requirements to be provided to the operator and user of the system. This documentation must contain all necessary information and safety instructions for the operator, the user and, if applicable, for any service personnel authorised by the manufacturer of the system.
- Read this document before setting up the product and keep it during the entire service life.
- The product must be suitable for the corresponding applications and environmental conditions without any restrictions.
- Only use the product for its intended purpose (→ **Bestimmungsgemäße Verwendung**).
- If the operating instructions or the technical data are not adhered to, personal injury and/or damage to property may occur.
- The manufacturer assumes no liability or warranty for any consequences caused by tampering with the product or incorrect use by the operator.
- Installation, electrical connection, set-up, programming, configuration, operation and maintenance of the product must be carried out by personnel qualified and authorised for the respective activity.
- Protect units and cables against damage.

2.2 Required background knowledge

34185

This document is intended for specialists. Specialists are people who, based on their relevant training and experience, are capable of identifying risks and avoiding potential hazards that may be caused during operation or maintenance of the product.

The document contains information about the correct handling of the product.

2.3 Safety symbols on the device

34199

General warning
Observe instructions in chapter "Electrical connection" (→ **Electrical connection** (→ S. 13))!

2.4 IT security

58235

NOTICE!

If the device is operated in an unprotected network environment:

> Unauthorised read or write access to data is possible.

> Unauthorised manipulation of the device function is possible.

► Check and restrict access options to the device.

3 Intended use

Content

Permitted use 9
Prohibited use.....9

34079

3.1 Permitted use

34211

The IO-Link master serves as a gateway between intelligent IO-Link devices and the EtherNet/IP network. The device is designed for use without a control cabinet in the food industry.

3.2 Prohibited use

34228

The device may not be used beyond the limits of the technical data (→ **Technical data** (→ S. 86))!

4 Function

Content

Parameter setting 10
Visual indication....10
EtherNet/IP 10
Internet of Things (IoT) 11
IO-Link 11
Voltage output 11

33836

4.1 Parameter setting

34583

The device provides the following configuration options:

- Parameter setting of the IO-Link master of the AL1421 with parameter setting software LR DEVICE and/or EtherNet/IP projection software
- Parameter setting of the connected IO-Link devices (sensors, actuators) with parameter setting software LR DEVICE and/or EtherNet/IP projection software
- Storage of parameter sets of the connected IO-Link devices for automatic recovery (data storage)

4.2 Visual indication

34192

The device has the following visual indicators:

- Status and error indication of the gateway, of the EtherNet/IP connection and of the system
- Status display of the voltage supply
- Status and activity display of the Ethernet connection
- Status, error and short circuit/overload indication of the IO-Link ports

4.3 EtherNet/IP

52585

The device offers the following EtherNet/IP functions:

- EtherNet/IP Device
- 2 port switch for access to the EtherNet/IP interface
- Gateway for transmission of the process and parameter data between the connected IO-Link devices and the higher-level EtherNet/IP controller
- Min. cycle time: 1 ms (RPI)
- Connection classes: 1, 3
- Connection Application types: Exclusive Owner, Input Only, Listen Only Connections
- UCMM supported
- Predefined standard objects:
  - Identity Object (0x01)
  - Message Router Object (0x02)
  - Assembly Object (0x04)
  - Connection Manager (0x06)
  - DLR Object (0x47)
  - QoS Object (0x48)
  - TCP/IP Interface Object (0xF5)
  - Ethernet Link Object (0xF6)

- Supported protocols: DHCP, BOOTP, ACD, DLR
- Device description: EDS file

4.4 Internet of Things (IoT)

58240

The device offers the following IoT functions:

- 2-port switch for access to the IoT interface (XF1 / XF2)
- Gateway for the transmission of process, parameter and monitoring data between IO-Linkmaster / IO-Link devices and the IT network level
- REST-API to access process and parameter data
- Supported protocols: TCP/IP JSON

4.5 IO-Link

34084

The device offers the following IO-Link functions:

- IO-Link master (IO-Link revision 1.0 and 1.1)
- 4 IO-Link ports for connection of IO-Link devices
- Provision of process data of the connected IO-Link devices for LR SMARTOBSERVER monitoring software (→ www.ifm.com)

4.5.1 IO-Link supply

57561

The device has 4 supplies for IO-Link devices (sensors, actuators).

The X1...X4 ports are class B ports.

The pin 2 (UA) of the X1...X4 ports supports different operating modes (disconnected from power, supply voltage UA, digital output).

The current intensity of the supply voltages US and UA of the X1...X4 ports can be adjusted.

Every supply provides short circuit monitoring.

The device ensures fire protection for connected IO-Link devices by providing an energy-limited circuit at the ports (according to IEC61010-1 and Class 2 to UL1310).

4.5.2 Digital outputs

57562

The device has 4 digital outputs that can be activated as an option (switching output DC-13 according to IEC 60947-5-1, 20 W).

The digital outputs are connected to pin 2 of the X1...X4 ports.

The digital outputs are supplied with the UA voltage. They refer to the potential of UA (pin 5).

4.6 Voltage output

57563

The device has a voltage output (XD2) to supply an additional device. This makes it possible to supply several "PerformanceLine"-type devices with one voltage source (daisy chain).

5 Mounting

Content

Installing the device 12

34058

5.1 Installing the device

57564

- ► Disconnect power before installation.
- ► Please observe the maximum tightening torque.
- ► Use a plane mounting surface for installation.

► Fix the unit to the mounting surface using 2 M5 mounting screws and washers.
  - Tightening torque: 1.8 Nm

6 Electrical connection

Content

Remarks 13
Connect EtherNet/IP ports....14
Connect IO-Link ports....15
Ground the device 16
Connect the device....17

33805

6.1 Remarks

34179

A qualified electrician must connect the unit.

► Observe the national and international regulations for the installation of electrical equipment.

Device is only suitable for operation on SELV/PELV voltages.

► Observe the information concerning IO-Link circuits!

The device contains components that can be damaged or destroyed by electrostatic discharge (ESD).

► Observe the required safety measures against electrostatic discharge!

The IP rating depends on the individual protection ratings of the unit, the applied connection elements and the corresponding protective covers.

The M12 connection parts in the device comply with the ingress resistance requirements of the standard EN 61076-2-101. To adhere to the protection rating, only cables certified to this standard must be used. The system creators undertake to ensure ingress resistance for cables which they cut to length themselves.

► Carry out the fitting according to the indications of the cable manufacturer. A maximum of 0.8 Nm is permitted.

► During installation, place the M12 connector vertically so that the coupling nut will not damage the thread.

► Depending on the mounting conditions, cables must be provided with a strain relief to avoid unacceptable loads on the mounting points and M12 connections.

► Make sure that the M12 connection parts are correctly seated and mounted correctly. The specified protection rating can not be guaranteed if this is not observed.

For UL applications:

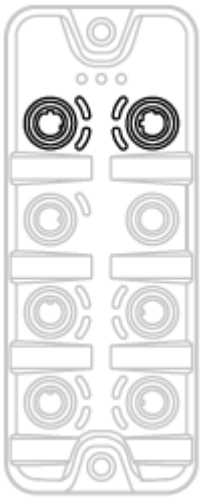
► For connecting the device and the IO-Link devices use UL certificated cables of category CYJV or PVVA with a minimum temperature rating of 80°C (75 °C for max. ambient temperature of 40 °C).

Wiring: → **Technical data** (→ S. 86)

The communication interfaces are separated from each other and from device surfaces that could be touched by means of basic insulation according to EN61010-1 (secondary circuit with 28 V DC maximum, supplied from mains circuit up to 300 V of overvoltage category II). They are designed for network environment 0 according to IEC TR62102.

6.2 Connect EtherNet/IP ports

57565



- ► Connect the device via the M12 socket XF1 and/or XF2 with the EtherNet/IP network (e.g. EtherNet/IP PLC, additional EtherNet/IP device)
- ► Connect the device via the M12 socket XF1 and/or XF2 with the IT network (e.g. PC with parameter setting software LR DEVICE, PC with monitoring software LR SMARTOBSERVER, PC with software capable of http requests)
- ► To connect the devices, use M12 connectors with protection rating IP 65 / IP 66 / IP 67 / IP 69K or higher (→ **Accessories** (→ S. 84)).
- ► Cover the unused sockets with M12 protective caps (art. no.: E12542).
  - Tightening torque 0.6...0.8 Nm

6.3 Connect IO-Link ports

57566

Wiring information:

- The connected IO-Link devices may only be supplied via the AL1421.
- The ports of the device meet the requirements of the IO-Link specifications 1.0 to 1.1.2.
- The additional digital outputs of the ports X1...X4 (pin 2) meet the requirements of the utilisation category DC-13 according to the standard IEC 60947-5-1: 20 W. The connected electronics must be electrically suited for this.

CAUTION

Connection of IO-Link Class A devices with Class B ports using 4/5 pole connectors

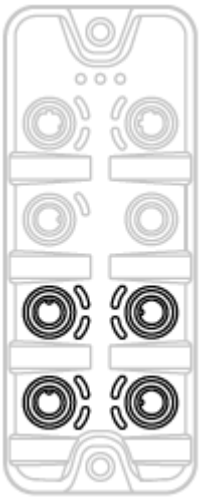
> Risk of fire

> Impairment of electrical safety

> Malfunctions

► To connect IO-Link Class A devices to Class B ports, use 3-pole connectors only!

Detailed information: → IO-Link planning directive www.io-link.com



► Connect the IO-Link devices with the M12 sockets X1...X4.
  - Maximum cable length per port: 20 m

► To connect the devices, use M12 connectors with protection rating IP 65 / IP 66 / IP 67 / IP 69K or higher (→ **Accessories** (→ S. 84)).

► Cover the unused sockets with M12 protective caps (art. no.: E12542).
  - Tightening torque: 0.6...0.8 Nm

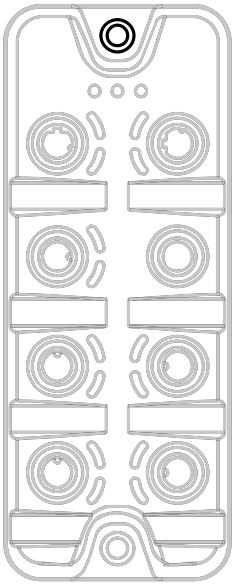
6.4 Ground the device

57568

The FE potential is connected to the following points of the device:

- Upper mounting lug of the housing
- Ports XD1 and XD2: Pin 5 (FE)
- Ports XF1 and XF2

To ensure the protection of the device against electrical interference and to ensure the safe function of the device, the housing has to be connected to the GND of the installation using the shortest possible route.



► Ground the unit via the mounting screw of the upper mounting lugs.
  - Tightening torque: 1.8 Nm

► Optional: Connect pin 5 of the port XD1 or XD2 via an L-coded M12 connector with the FE socket of the power supply.

6.5 Connect the device

52464

CAUTION

Exceeding the maximum input current of 16 A

> Fire hazard

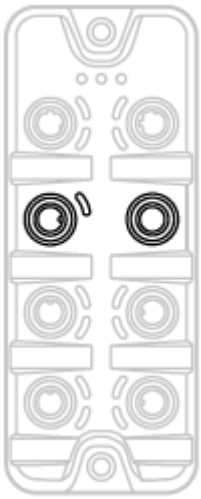
► Select I_U and I_A of the power supplies US and UA taking into account the derating characteristics of the AL1421 (→ **Derating behaviour** (→ S. 87))!

NOTICE

Missing reverse polarity protection in case of cross polarity between US and UA: The supply voltages US and UA are individually protected against reverse polarity. The supply voltages US and UA are not protected against cross polarity reversal.

> Damage to the device

► Make sure that the supply voltages US and UA are connected correctly.



► Disconnect power.

► Connect the device via M12 socket X31 to US and UA supplying 24 V DC each (20...28 V SELV/PELV; according to IEC 61010-1, secondary circuit with maximum 28 V DC, supplied from mains circuit up to 300 V of overvoltage category II).

- Recommended maximum cable length: 25 m

► To connect the device, use L-coded M12 connectors with at least protection rating IP 65 / IP 66 / IP 67 / IP 69K (→ **Accessories** (→ S. 84)).

Optional: Supply of an additional PerformanceLine devices (Daisy chain):

► Connect additional PerformanceLine Master to M12 socket X32 of the AL1421 (Daisy-chain function).

- Recommended maximum cable length: 25 m

► To connect the device, use L-coded M12 connectors with at least protection rating IP 65 / IP 66 / IP 67 / IP 69K (→ **Accessories** (→ S. 84)).

In case of cables that are longer than 25 m, take the voltage drop and the necessary minimum supply voltage of 20 V into consideration!

7 Operating and display elements

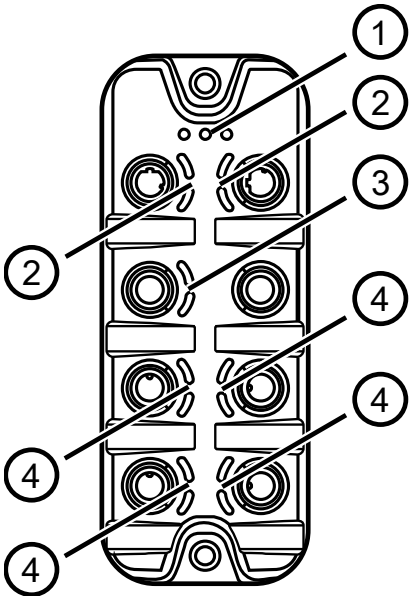
Content

Overview....18
LED indicators....19

34063

7.1 Overview

57569



1. Status LEDs RDY, NET and MOD (→ **Status LEDs** (→ S. 19))
2. Status LEDs LNK and ACT of the EtherNet/IP interfaces 1 (XF1) and 2 (XF2) (→ **Ethernet ports** (→ S. 19))
3. Status LEDs US and UA of the voltage supply (XD1/XD2) (→ **Voltage supply** (→ S. 20))
4. Status LEDs IOL and UA/DO of the IO-Link Class B ports (X1...X4) (→ **IO-Link Ports (Class B)** (→ S. 20))

7.2 LED indicators

34047

The device only has the following LED indicators:

7.2.1 Status LEDs

34549

The RDY LED shows the status of the gateway.
The BF LED (Bus Failure) shows the status of the EtherNet/IP connection.
The SF LED (System Failure) shows the status of the system.

Status LED			Description
RDY	green	on	Status: OK
		flashes 5 Hz	Status: Error
		flashes (200 ms on, 800 ms off)	Status: Firmware update is running
		off	Status: Gateway not running or gateway booting
BF	red	on	Bus error
		flashes 1 Hz	No connection to the EtherNet/IP controller
		off	error-free
SF	red	on	▪ Error in gateway ▪ At least 1 IO-Link device sends warning / alarm (temperature, over/under current, over/under voltage, shortcut)
		off	error-free

7.2.2 Ethernet ports

34348

Each Ethernet port has 2 LEDs (LNK and ACT). The LEDs indicate the status of the Ethernet connection.

Status LED			Description
LNK	green	on	Ethernet connection established
		off	No Ethernet connection
ACT	yellow	flashes	Data is transmitted via the Ethernet interface.
		off	No data transmission

7.2.3 Voltage supply

57570

The port for the voltage supply (XD1) has the LEDs with the designation US and UA. The LEDs indicate the status of the supply voltage.

Status LED			Description
US	green	on	Supply voltage US is connected
		off	either there is no supply voltage or the connected supply voltage is too low
UA	green	on	Supply voltage UA is connected
		Off	either there is no supply voltage or the connected supply voltage is too low

7.2.4 IO-Link Ports (Class B)

57571

Each IO-Link Port Class B has 2 LEDs with the designation IOL and UA/DO. The LEDs show the status of the IO-Link ports.

Status LED			Description
IOL	yellow	Off	Port configured as DI/DO: pin 4 (C/Q) = OFF
		on	Port configured as DI/DO: pin 4 (C/Q) =ON
	green	flashing 1 Hz	Port configured asIO-Link: no IO-Link device found
		Flashing with 2 Hz	Port configured asIO-Link: Status PREOPERATE
		on	Port configured asIO-Link: Status OPERATE
	red	Flashing with 2 Hz	Port configuration error or short circuit / overload on US
		on	Transmission error
UA/DO	yellow	Off	Digital output: Pin 2 (UA) = OFF
		on	Digital output: Pin 2 (UA) = ON
	green	Off	IO-Link type A supply: Pin 2 (UA) = OFF
		on	IO-Link type B supply: Pin 2 (UA) = ON (not switchable)
	red	on	Error: Overcurrent or undervoltage

8 Setup

57572

When the supply voltages have been switched on, the AL1421 starts with the factory settings. The LEDs signal the current operating status of the device and the ports (→ **LED indicators** (→ S. 19)).

8.1 Parameter setting options

59779

The device can be configured withe the following options:

- Software LR DEVICE (→ **LR DEVICE** (→ S. 23)
- REST API for IoT Core (→ **ifm IoT Core** (→ S. 147, → S. 37)
- EtherNet/IP projecting software (→ **EtherNet/IP** (→ S. 73)

9 Configuration

Content

LR DEVICE....23
ifm IoT Core....37
EtherNet/IP....73

33858

9.1 LR DEVICE

Content

Remarks 24
IoT: Configure IP settings 25
IoT: Configuring access rights 26
IoT: Configure the interface to LR AGENT or LR SMARTOBSERVER 27
Fieldbus: Configure IP settings 28
Fieldbus: set the configuration mode 29
IO-Link ports: Activate data transfer to LR AGENT or LR SMARTOBSERVER 30
IO-Link ports: Set the operating mode Pin 2 (UA) 30
IO-Link ports: setting the operating mode Pin 4 (US) 31
IO-Link ports: Limit the current intensity 32
IO-Link ports: Set the device validation and data storage 33
IO-Link ports: Setting fail-safe values 34
Info: Show device information 34
Firmware: Reset device to factory settings 35
Firmware: Reboot the device 35
Configure IO-Link devices 36

33692

On delivery, the AL1421 is configured with the factory settings (→ **Factory Settings** (→ S. 83)).

Required software: LR DEVICE (1.6.x or higher) (art.-no.: QA0011/QA0012)

9.1.1 Remarks

Content

Online parameter setting 24
Offline parameter setting 24
VPN connection 24

34180

Online parameter setting

57575

The parameters of the IO-Link master and the connected sensors and actuators can be set before installation and set-up or during operation.

If parameters are changed during operation, this will influence the function of the plant.

► Ensure that there will be no malfunctions in your plant.

During the parameter setting process, the IO-Link master and the sensor stay in operating mode. They continue their monitoring functions with the existing parameters until the parameter setting has been completed.

Offline parameter setting

34060

The AL1421 supports the offline parameter setting. In this context, the user creates and stores a configuration for the IO-Link master and the connected IO-Link devices without being connected to the AL1421 (OFFLINE mode). The configuration created in this way can be stored as a file (*.lrp) and loaded to the AL1421 and activated at a later date.

Further information about offline parameter setting: → Operating instructions LR DEVICE

VPN connection

34382

An active VPN connection blocks the access of the parameter setting software LR DEVICE to the EtherNet/IP interface of the AL1421.

► Deactivate the VPN connection in order to be able to access the AL1421 with the LR DEVICE.

9.1.2 IoT: Configure IP settings

34049

For access to the IO-Link master via the IT infrastructure the user has to set the IP settings of the IoT port.

To configure the IP settings with DHCP, a DHCP server has to be active in the IT network. If no DHCP server can be reached in the IT network, an IP address is automatically assigned to the IoT port with the Zeroconfig protocol (address range: → **Factory Settings** (→ S. 83)).

To configure the IP settings of the IoT interface:

► Select [IoT] menu.

> The menu page shows the current settings.

► Set the following parameters as required:

Name	Description	Possible values	
[DHCP]	Activate/deactivate the DHCP client of the device	[Static IP]	IP settings were set by the user
		[DHCP]	IP settings are set by a DHCP server in the network.
[IP address]*	IP address of the IoT port	Factory setting: 169.254.X.X	
[Subnet mask]*	Subnet mask of the Ethernet network	Factory setting: 255.255.0.0	
[Default gateway IP address]*	IP address of the network gateway	Factory setting: 0.0.0.0	
[MAC address]	MAC address of the IoT port	The value is firmly set.	

* ... can only be edited if parameter [DHCP] = [Static IP]

► Save changed values on the device.

9.1.3 IoT: Configuring access rights

34046

The access rights define which instance may read and / or write the parameter data, process data and event/diagnostic messages.

In order to configure the access rights to the IO-Link master:

► Select [IoT] menu.

> The menu page shows the current settings.

► Set the following parameters as required:

Name	Description	Possible values	
[Access Rights]	The access rights to the parameter data, process data and the event/diagnostic messages of the IO-Link master as well as the connected IO-Link devices	[EtherNet/IP + IoT]*	▪ EtherNet/IP and IoT Core have read and write access rights to parameters and process data ▪ EtherNet/IP and IoT Core have read access rights to events/alarms
		[EtherNet/IP + IoT (read-only)]	▪ EtherNet/IP has read and write access rights to parameters and process data ▪ EtherNet/IP has read access rights to events/alarms ▪ IoT Core only has read access rights to parameters, process data and events/alarms
		[IoT only]	▪ IoT Core has read and write access rights to parameters and process data ▪ IoT has read access rights to events/alarms ▪ EtherNet/IP has no access rights

* ... Factory setting

► Save changed values on the device.

If in LR DEVICE and EtherNet/IP projection software the parameter [Access Rights] is = [EtherNet/IP + IoT], the parameter values set in the EtherNet/IP projection software will always apply.

If the parameter [Access Rights] in LR DEVICE is = [IoT only], set the parameter [Access Rights] = [Keep settings] in the EtherNet/IP projection software.

If the parameter [Access Rigts] in LR DEVICE is = [<Fieldbus> + IoT (read-only)], write access to the device configuration via LR DEVICE and IoT core services is blocked. To enable write access again, set the parameter to [<Fieldbus> + IoT] via fieldbus configuration software.

Changes of the parameter [Access Rights] will only be effective after restarting the IO-Link master (→ **Firmware: Reboot the device** (→ S. 35)).

9.1.4 IoT: Configure the interface to LR AGENT or LR SMARTOBSERVER

34048

To enable transfer of process data from the IO-Link master to LR AGENT or LR SMARTOBSERVER, the interface has to be configured accordingly.

► Select [IoT] menu.

> The menu page shows the current settings.

► Set the following parameters as required:

Name	Description	Possible values	
[IP address LR Agent or SMARTOBSERVER]	IP address of LR AGENT or LR SMARTOBSERVER	Factory setting: 255.255.255.255	
[Port LR Agent or SMARTOBSERVER]	Port number that is used to send process data to LR AGENT or LR SMARTOBSERVER	0 ... 65535	Factory setting:: 35100
[Interval LR Agent or SMARTOBSERVER]	Cycle time for the transfer of the process data to LR AGENT or LR SMARTOBSERVER (value in milliseconds)	[Off]	no transfer
		500 ... 2147483647	500 ms ... 2147483647 ms
[Application Tag]	Source identifier of the IO-Link master in the structure of LR AGENT or LR SMARTOBSERVER (String32)	Factory setting: AL1421	

After changing the parameter [Port LR Agent or SMARTOBSERVER] or [Application Tag], it may take 120 seconds before the device establishes a new TCP connection.

To prevent the delay:

► Reboot the device after changing the the parameter.

► Save changed values on the device.

9.1.5 Fieldbus: Configure IP settings

54692

For communication with the EtherNet/IP network, the EtherNet/IP interface must be configured.

► Select [Fieldbus] menu.

> The menu page shows the current settings.

► Set the following parameters as required:

Name	Description	Possible values	
[DHCP]	Enable / disable the DHCP client of the device	[Static IP]	IP parameters are set by the user
		[DHCP]	IP parameters are set by a DHCP server in the network.
		[BOOTP]	IP parameters are set via the Bootstrap Protocol (BOOTP)
[IP address]*	IP address of the EtherNet/IP port	Factory setting: 192.168.1.250	
[Subnet mask]*	Subnet mask of the IP network	Factory setting: 255.255.255.0	
[Default gateway IP address]*	IP address of the gateway	Factory setting: 0.0.0.0	
[Host name]	Name of the device in the EtherNet/IP network	e.g. al1xxx	
[MAC address]	MAC address of the device	The value is firmly set.	
[Fieldbus firmware]		e.g. 3.4.04 (EtherNet/IP Adapter)	

* ... Parameter can only be edited if parameter [DHCP] = [Static IP]

► Save changed values on the device.

9.1.6 Fieldbus: set the configuration mode

57640

The AL1421 supports the EtherNet/IP configuration modes "top-down" and "independent". Additionally, the user can configure the length of the transmitted process data and select the required connection types.

► Select [Fieldbus] menu.

> The menu page shows the current settings.

► Set the following parameters as required:

Name	Description	Possible values	
[Configuration]*	EtherNet/IP configuration mode	Independent mode off	Configuration via fieldbus PLC
		Independent mode on	Configuration via AL1421
[Process data length]*	Length of process data per IO-Link port	2 bytes input 2 bytes output	2 bytes input data, 2 bytes output data
		4 bytes input 4 bytes output	4 bytes input data, 4 bytes output data
		8 bytes input 8 bytes output	8 bytes input data, 8 bytes output data
		16 bytes input 16 bytes output	16 bytes input data, 16 bytes output data
		32 bytes input 32 bytes output	32 bytes input data, 32 bytes output data
[Swap]*	Sequence of bytes in the data word	off	as Array of Bytes
		on	as Integer16 value; during an update of the process data, the bytes are exchanged
[Explicitpdmode]**	Enable / disable explicit PD mode and select the process data to be transmitted (connection types)	Explicit process data mode off	Explicit PD mode disabled
		Explicit process data mode with IO-Link I/O + Acyclic + Diag	Explicit PD mode enabled: IO-Link inputs / outputs, acyclic data and diagnostic data are transmitted
		Explicit process data mode with IO-Link I/O + Acyclic + Diag + EnMo	Explicit PD mode enabled: IO-Link inputs / outputs and acyclic data, diagnostic data and data for energy monitoring are transmitted
		Explicit process data mode with IO-Link I/O	Explicit PD mode enabled: IO-Link inputs/outputs are transmitted

* ... Parameter can only be changed if the EtherNet/IP controller is disconnected
** ... Parameter only valid if [Configuration] = [Independent mode on]

► Save changed values on the device.

9.1.7 IO-Link ports: Activate data transfer to LR AGENT or LR SMARTOBSERVER

33690

The user can decide separately for each IO-Link port whether the process data of the connected IO-Link devices should be transferred to LR AGENT or LR SMARTOBSERVER.

To transfer process data the interface to the LR AGENT or LR SMARTOBSERVER has to be correctly configured (→ **IoT: Configure the interface to LR AGENT or LR SMARTOBSERVER** (→ S. 27)).

To activate / deactivate data transfer:

► Select [Port x] menu (x = 1...4).

> The menu page shows the current settings.

► Set the following parameters as required:

Name	Description	Possible values	
[Transmission to LR Agent or SMARTOBSERVER]	Transfer of process data of the connected IO-Link device to LR AGENT oder LR SMARTOBSERVER	[Disabled]	Transfer process data
		[Enabled]	Don't transfer process data

► Save changed values on the device.

9.1.8 IO-Link ports: Set the operating mode Pin 2 (UA)

57578

The pin 2 of the IO-Link ports X1...X4 supports the following operating modes:

- Off: no voltage on pin; port functions as IO-Linkport type A
- On: the voltage UA is applied to pin 2; ports functions as IO-Linkport type B
- Digital output (DO): binary output signal is on pin 2; port functions as digital switching output

The user can set the operating mode separately for each IO-Link port.

To set the operating mode of pin 2 (UA) of an IO-Link port:

► Select menu [Port x] (x = 1...X1...X4)

> The menu page shows the current settings.

► Set the following parameters as required:

Name	Description	Possible values	
[Mode Pin2 UA]	Operating mode of pin 2 of the port	[Off (IO-Link Type A Supply)]	IO-Link Port type A
		[On (IO-Link Type B Supply)]	IO-Link Port type B
		[Digital Output]	Digital switching output

► Save changed values on the device.

9.1.9 IO-Link ports: setting the operating mode Pin 4 (US)

57580

The pin 4 of the IO-Link-Ports X1...X4 supports the following operating modes:

- Disabled: no data transmission on pin 4 (C/Q) of the IO-Link ports
- Digital input (DI) binary input signal on pin 4 (C/Q) of the IO-Link ports
- Digital output (DO): binary output signal on pin 4 (C/Q) of the IO-Link ports
- IO-Link: IO-Link data transfer via pin 4 (C/Q) of the IO-Link ports

The user can set the operating mode separately for each IO-Link port.
To set the operating type of pin 4 (US) of an IO-Link port:

► Select [Port x] menu (x = 1...4).

> The menu page shows the current settings.

► Set the following parameters as required:

Name	Description	Possible values	
[Mode Pin4 US]	Operating mode of the pin 4 of the port	[Disabled]	no data transmission
		[DI]	Digital input
		[DO]	Digital output
		[IO-Link]	IO-Link data
[Cycle time actual]**	Current cycle time of the data transfer between IO-Link master and IO-Link device on the port (value in microseconds)	Parameter can only be read	
[Cycle time preset]*	Cycle time of the data transfer between the IO-Link master and the IO-Link device at the port (value in microseconds)	0	The device automatically sets the fastest possible cycle time.
		1 ... 132800	1 microsecond ... 132800 microseconds
[Bitrate]**	Current transmission rate of the data transfer between the IO-Link master and the IO-Link device on the port	Parameter can only be read	

* ... Parameter only available if [Mode] = [IO-Link]
** ... Parameter only visible if IO-Link Device am IO-Link port is connected.

► Save changed values on the device.

9.1.10 IO-Link ports: Limit the current intensity

57579

For the IO-Link ports X1...X4, the following features can be set:

- Max. current intensity of the supply voltage US
- Max. current intensity of the supply voltage UA

To set the max. current intensity of the supply voltages US and UA of an IO-Link port:

► Select [Port x] menu (x = 1...4).

> The menu page shows the current settings.

► Set the following parameters as required:

Name	Description	Possible values	
[Current Limit Pin2 UA]	Max. current intensity of the supply voltage UA on the port (value in milliamps)	0 ... 2000*	0 mA* 2000 mA
[Current Limit Pin1 + Pin4 US]	Max. current intensity of the supply voltage US on the port (value in milliamps)	0 ... 450* ... 2000	0 mA 450 mA 2000 mA

* ... Factory setting

► Save changed values on the device.

9.1.11 IO-Link ports: Set the device validation and data storage

33697

The user can choose how the IO-Link ports are to behave with regard to the device validation and the storage / recovery of parameter data of the connected IO-Link device.

The following options are available:

Option	Validation of the IO-Link device	Storage of the parameter values	Recovery of the parameter values
[No check and clear]	no	no	no
[Type compatible V1.0 device]	yes, test the compatibility with IO-Link standard V1.0	no	no
[Type compatible V1.1 device]	yes, test the compatibility with IO-Link standard V1.1	no	no
[Type compatible V1.1 device with Backup + Restore]	yes, test the compatibility with IO-Link standard V1.1 and identity of design (vendor ID and device ID)	yes, automatic storage of the parameter values; changes of the current parameter values will be stored	yes, recovery of the parameter values when connecting an identical IO-Link device with factory settings
[Type compatible V1.1 device with Restore]	yes, test the compatibility with IO-Link standard V1.1 and identity of design (vendor ID and device ID)	no, there is no automatic storage changes of the current parameter values will not be stored	yes, recovery of the parameter values when connecting an identical IO-Link device with factory settings

The options only apply if the IO-Link port is in the operating mode "IO-Link".

For options [Type compatible V1.1 device with Backup + Restore] and [Type compatible V1.1 device with Restore]: If the vendor ID and device ID are changed in the online mode, the data memory will be deleted and a new backup of the parameter values of the connected IO-Link device will be created in the IO-Link master.

To configure the device validation and the data storage:

► select [Port x] menu (x = 1...4).

> The menu page shows the current settings.

► Set the following parameters as required:

Name	Description	Possible values	
[Validation / Data Storage]	Supported IO-Link standard and behaviour of the IO-Link master when connecting a new IO-Link device at port x (x = 1...4)	[No check and clear]	
		[Type compatible V1.0 device]	
		[Type compatible V1.1 device]	
		[Type compatible V1.1 device with Backup + Restore]	
		[Type compatible V1.1 device with Restore]	
[Vendor ID]	ID of the manufacturer that is to be validated	0...65535	Factory setting: 0# ifm electronic: 310
[Device ID]	ID of the IO-Link device that is to be validated	0...16777215	Factory setting: 0

► Save changed values on the device.

9.1.12 IO-Link ports: Setting fail-safe values

34329

For the configuration mode "Independent" the user can set fail-safe values for the outputs of IO-Link ports. The fail-safe values will be activated in case of an interuption of the EtherNet/IP conection.

To set the fail-safe values:

► Select [Port x] menu (x = 1...4).

> The menu page shows the current settings.

► Set the following parameters as required:

Name	Description	Possible values	
[Fail-safe digital out]*	Fail-safe value of the output for operating mode "Digital Output (DO)"	[Reset]	OFF
		[Old]	old value
		[Set]	ON
[Fail-safe IO-Link]*	Fail-safe value of the output for operating mode "IO-Link"	[Off]	no Fail-safe
		[Reset]	Fail-safe: OFF
		[Old	Fail-safe: old value
		[Pattern]	Fail-safe: byte pattern

* ... Parameter only changeable, if the connection to the EtherNet/IP controller is closed

► Save changed values on the device.

9.1.13 Info: Show device information

34065

To read the general information of the ifm IO-Link master:

► Select [Info] menu.

> The menu page shows the current settings.

Name	Description	Possible values
[Product code]	Article number of the IO-Link master	AL1421
[Device family]	Device family of the IO-Link master	IO-Link master
[Vendor]	Vendor	ifm electronic gmbh
[SW-Revision]	Firmware of the IO-Link master	
[HW revision]	Hardware version of the IO-Link master	
[Bootloader revision]	Bootloader version of the IO-Link master	
[Serial number]	Serial number	

9.1.14 Firmware: Reset device to factory settings

33838

When the IO-Link master is reset, all parameters are set to the factory settings:
To reset the device to factory settings:

► Select [Firmware] menu.
> The menu page shows the current settings.
► Click on [Factory Reset] to reset the device.
> LR DEVICE sets the device to the factory settings.

9.1.15 Firmware: Reboot the device

33832

When rebooting the device, all settings are kept.
To restart the AL1421:

► Select [Firmware] menu.
> The menu page shows the current settings.
► Click on [Reboot] to reboot the device.
> LR DEVICE reboots the ifm IO-Link master.

9.1.16 Configure IO-Link devices

33856

To configure the IO-Link devices connected to the device with the LR DEVICEparameter setting software:

Requirements:

> IO-Link master is correctly installed and connected to the LR DEVICE parameter setting software.

> The IO-Link device is connected correctly with the AL1421.

> Operating mode of the IO-Link port is "IO-Link" (→ **IO-Link ports: setting the operating mode Pin 4 (US)** (→ S. 31)).

> IoT has write access rights to the IO-Link master (→ **IoT: Configuring access rights** (→ S. 26)).

1 Select IO-Link master

► Start LR DEVICE.

► Update IODD file library
OR:
Import IODD file of the IO-Link device manually.

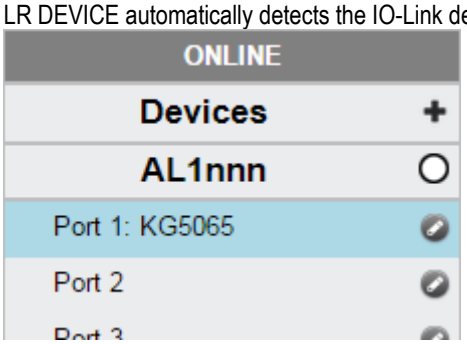
► Scan network for devices.

> LR DEVICE detects IO-Link master.

2 Add IO-Link device

► Under [ONLINE]: Click on the required IO-Link master.

> LR DEVICE automatically detects the IO-Link devices connected to the IO-Link master (e.g. ifm sensor KG5065).



3 Configure IO-Link device

► Mouse click on the port to which the IO-Link device is connected.

> LR DEVICE reads and shows the current parameter values of the IO-Link device.

► Configure IO-Link device.

Information about the available parameters of the IO-Link device: → IO Device Description (IODD) des IO-Link Devices

► Save the changed configuration on the IO-Link device.

9.2 ifm IoT Core

Content

Programmers' notes 38
First steps 43
General functions 43
Fieldbus: Configuring IP settings....47
Fieldbus: Selecting the configuration mode 47
Fieldbus: Setting failsafe values....48
IoT: Configuring access rights....48
IoT: Configuring the LR AGENT or LR SMARTOBSERVER interface 49
IO-Link ports: Limiting current values....49
IO-Link ports: Setting the operating mode of pin 4 (US) 49
IO-Link ports: Configuring device validation and data storage....50
IO-Link ports: Configuring data transfer to LR AGENT or LR SMARTOBSERVER 52
IO-Link ports: Reading and writing process data 52
IO-Link ports: Indicating port events....55
IO-Link devices: Accessing parameters....55
IO-Link devices: Reading an writing device information 57
IO-Link devices: Indicating IO-Link events....57
Gateway: Resetting, rebooting and localising the device....57
Gateway: Reading device information....58
Gateway: Reading status and diagnostic information 58
Gateway: Updating the firmware 59
Gateway: Setting the application tag 60
Subscribing to notifications....61
Using Web Socket....64
Using the IoT-Core Visualizer 66

52244

General notes on the ifm IoT Core: → **Programmers' notes** (→ S. 38)

9.2.1 Programmers' notes

Content

IoT Core: General information 38
Access the ifm IoT Core 39

34229

IoT Core: General information

52256

The PerformanceLine device family has an IoT Core. The IoT Core allows the user to address the AL1421 from IT networks via a REST API and to integrate it into Internet-of-Things applications.

A device description is stored on the AL1421. This device description is a structured, machine-readable data object in JSON format. All current values of parameters, process data, diagnostic data and device information are mapped in this data object. These data values can be read and changed by means of services.

Access the ifm IoT Core

52257

The user can access the ifm IoT Core via HTTP requests. The following request methods are available.

GET request

33804

Using the GET method the user has read access to a data point.

The syntax of the request to the IoT Core is:

```
http://ip/datapoint/service
```

Parameter	Description
ip	IP address of the IoT interface
data_point	Data point which is to be accessed
service	Service

The syntax of the return of the IoT Core is:

```
{
"cid":id,
"data":{"value":resp_data},
"code":diag_code
}
```

Parameter	Description
id	Correlation ID for the assignment of request and return
resp_data	Value of the data point; depending on the data type of the data point
diag_code	Diagnostic code (→ **IoT core: Diagnostic codes** (→ S. 42))

Example: GET request

54033

Request (via browser):

```
http://192.168.0.250/devicetag/applicationtag/getdata
```

Response:

```
{
"cid":-1,
"data":{"value":"AL1421"},
"code":200
}
```

POST request

34212

Using the POST method the user has read and write access to a data point.
The syntax of the request to the IoT Core is:

```
{
"code":"code_id",
"cid":id,
"adr":"data_point/service",
"data":{req_data}
}
```

Field	Parameter	Description	
code	code_id	service class	
		▪ Request	Request
		▪ Transaction	Transaction
		▪ Event	Event
cid	id	Correlation ID for the assignment of request and response; id freely setable by the user	
adr	data_point	Data point of element tree which is to be accessed	
	service	Service to be called (→ **Overview: IoT services** (→ S. 155))	
data*	req_data	Data to be transferred to the IoT Core (e.g. new values); syntax depending on the service	

* = optional: only required for services which send data to the IoT core (e.g. setdata)

The syntax of the return of the IoT Core is:

```
{
"cid":id,
"data":{"value":resp_data},
"code":diag_code
}
```

Field	Parameter	Description
cid	id	Correlation ID for the assignment of request and return
data*	resp_data	Value of the data point; depending on the data type of the data point
code	diag_code	Diagnostic code (→ **IoT core: Diagnostic codes** (→ S. 42))

* = optional: only required for services which receive data from the IoT core (e.g. getdata)

Example: POST request

54035

Request:

```
{
"code":"request",
"cid":4711,
"adr":"devicetag/applicationtag/getdata"
}
```

Response:

```
{
"cid":4711,
"data":{"value":"AL1421"},
```

"code":200
}

IoT core: Diagnostic codes

58223

Code	Text	Description
200	OK	Request successfully processed
230	OK but needs reboot	Request successfully processed; IO-Link master must be restarted
231	OK but block request not finished	Request successfully processed; blockwise request, but not yet finished
232	Data has been accepted, but internally modified	New values have been accepted, but were adjusted by the IO-Link master (Master cycle time)
400	Bad request	Invalid request
403	Forbidden	Forbidden request
500	Internal Server Error	Internal fault
503	Service Unavailable	The service is not available (e. g. IO-Link port in wrong operating mode; no IO-Link device at IO-Link port)
530	The requested data is invalid	Invalid process data
531	IO-Link Error	Error in IO-Link Master / device
532	PLC connected Error	Error while setting data, because IO-Link master is still connected to fieldbus PLC

9.2.2 First steps

52245

To read the device description of the AL1421:

► Send the following POST request to the AL1421:
`{"code":"request","cid":-1,"adr":"gettree"}`

> AL1421 returns the device description as structured JSON object.

► Identify all substructures and the data points contained therein in the tree structure of the JSON object.

► Identify the applicable services for the access to substructures and the data points contained therein.

9.2.3 General functions

61148

The AL1421 has the type `device` (→ **Overview: IoT types** (→ S. 154)).

The following services can be used on the root element of the type `device`:

Service	Description
../gettree	Provide the complete tree or subtree of the device description (JSON)
../getidentity	Reading device information
../getdatamulti	Reading several parameter values sequentially
../getelementinfo	Reading detailed information of an element
../getsubscriberlist	Print a list of all active notification subscriptions
../querytree	Search device description for specific elements

Depending on the read and write access rights, the following services can be applied to elements of type `data`:

Service	Description
../getdata	Reading the value of the element
../setdata	Write the value of the element

Example: Reading properties of an element

59782

Task: Determine the data type and value range of the accessrights parameter.

Solution: Read the properties of the element `iotsetup/accessrights` of the getelementinfo service. The fields type (data type) and valuation (range of values) contain the required information.

- Request:

```
{
"code":"request",
"cid":4711,
"adr":"getelementinfo",
"data":{"adr":"iotsetup/accessrights"}
}
```

- Response:

```
{
"cid":4711,
"data":{
"identifier":"accessrights",
```

```
"type":"data",
"uid":null,
"profiles":["parameter"],
"format":{
"type":"enum",
"namespace":"json",
"encoding":"integer",
"valuation":{
"valuelist":{
"0":"Fieldbus + IoT",
"1":"Fieldbus + IoT (read-only)",
"3":"IoT only"}}}},
"code":200
}
```


The accessrights parameter has the data type ENUM with the valid values "Fieldbus + IoT", "Fieldbus + IoT (read only)" and "IoT only".

Example: output subtree

61149

Task: Output all direct sub-elements of the node `firmware`.

Solution: Use the service gettree to output the required subtree (root node: firmware, sub-levels to be shown: 1)

- Request:


```
{
"code":"request",
"cid":4711,
"adr":"gettree",
"data":{
"adr":"firmware",
"level":1}
}
```


- Response:


```
{
"cid":4711,
"data":{
"identifier":"firmware",
"type":"structure",
"profiles":[
"software","software/uploadablesoftware"],
"subs":[
{
"identifier":"version","type":"data","profiles":["parameter"],
"format":{"type":"string","namespace":"json","encoding":"UTF-8"}},
{
"identifier":"type","type":"data",
"format":{"type":"string","namespace":"json","encoding":"UTF-8"}},
{
"identifier":"install","type":"service"},
{
"identifier":"factoryreset","type":"service"},
{
"identifier":"signal","type":"service"},
{
"identifier":"container","type":"data",
"format":{"type":"binary","namespace":"json","encoding":"base64"}},
```

```
{
"identifier":"reboot","type":"service"}]
},
"code":200
}
```

Example: Read several parameter values of the IO-Link master simultaneously

33840

Task: The following current values are to be read by the IO-Link master: temperature, serial number

Solution: Read the current parameter values using the getdatamulti service (data point temperature: /processdatamaster/temperature; data point serial number: /deviceinfo/serialnumber)

- Request:

```
{
"code":"request",
"cid":4711,
"adr":"/getdatamulti",
"data":{"datatosend":["/processdatamaster/temperature","/deviceinfo/serialnumber"]
}
}
```

- Response:

```
{
"cid":4711,
"data":{"processdatamaster/temperature":{"code":200,"data":44},
"deviceinfo/serialnumber":{"code":200,"data":"000174210147"}},
"code":200
}
```

Example: Browsing device description

61150

Task: List all elements with the designation "status" and the profile "runcontrol".

Solution: Use the service querytree to browse the device description with the parameters "status" (name) and "runcorntrol" (profile)

- Request:

```
{
"cid":4711,
"adr":"querytree",
"code":"request",
"data":{
"profile":"runcontrol",
"name":"status"}
}
```

- Response:

```
{
"cid":4711,
"data":{
"adrList":[
"device/connections/mqttConnection/status",
"device/connections/mqttConnection/mqttCmdChannel/status"]},
"code":200
}
```

Setting the storage duration

61153

The IoT Core offers the possibility to set the storage duration of data and notifications. The Services **Service: setdata** (→ S. 165) and **Service: subscribe** (→ S. 166) therefore have the parameter "duration".

Example: Subscribing to notifications

61154

Task: The current values of the following parameters are to be sent regularly to a network server with IP address 192.168.0.4:

- Product name of the IO-Link Devices an IO-Link port X02
- Cyclic input data of the IO-Link Devices an IO-Link port X02
- Operating temperature of the IO-Link master.

The subscription is only to be active until the next restart of the IO-Link master.

Solution: Subscribe to the required data using the subscribe service.

- Request:

```
{
"code":"request",
"cid":4711,
"adr":"/timer[1]/counter/datachanged/subscribe",
"data":{
"callback":"http://192.168.0.4:80/temp",
"datatosend":[
"/iolinkmaster/port[2]/iolinkdevice/productname",
"/iolinkmaster/port[2]/iolinkdevice/pdin",
"/processdatamaster/temperature"],
"duration":"uptime"}
}
```

- Response:

```
{
"cid":4711,
"code":200
}
```

9.2.4 Fieldbus: Configuring IP settings

59783

Substructure: `fieldbussetup`

Available data points:

Name	Description	Access
../hostname	Name of the IO-Link master in the fieldbus project	rw
../fieldbusfirmware	Firmware version of the IO-Link master	r
../network/macaddress	MAC address of the fieldbus port	r
../network/ipaddress	IP address of the fieldbus port	rw*
../network/subnetmask	Subnet mask of the network segment	rw*
../network/ipdefaultgateway	IP address of the network gateway	rw*
../network/dhcp	Configuration of the IP settings of the fieldbus interface	rw
../connectionstatus	Status of the connection to the EtherNet/IP network	r

r ... read only
rw ... read and write
* ... only changeable, if the EtherNet/IP controller is not in RUNNING state

Applicable services:

Name	Description
../network/setblock	Write all values of the substructure at once

Change the IP parameters in the substructure `network` only blockwise by using the service setblock!

9.2.5 Fieldbus: Selecting the configuration mode

52486

Substructure: `fieldbussetup/configuration`

Available data points:

Name	Description	Access
../independentmode	Set the configuration mode (Top-down, Independent)	r/w*
../explicitpdmode	Connection types	r/w*
../processdataconfiguration	Length of the process input data and process output data	rw*
../configuration/swap	Byte order of process data	r/w*

rw ... read and write
* ... only changeable, if the EtherNet/IP controller is not in RUNNING stated

9.2.6 Fieldbus: Setting failsafe values

SYS_OBJECTID>

Substructure: `fieldbussetup/configuration/port[n] (n = 1...4)`

Available data points:

Name	Description	Access
../failsafedigitalout	Failsafe value of the digital output - pin 4 (DO)	rw*
../failsafeiolink	Failsafe value of the IO-Link output data - pin 4 (IO-Link)	rw*
../failsafepin2	Failsafe value of digital output - pin 2 (only available ports class B)	rw*

rw ... read and write
* ... only changeable, if the EtherNet/IP controller is not in RUNNING stated

9.2.7 IoT: Configuring access rights

59785

Substructure: `iotsetup`

Available data points:

Name	Description	Access
../accessrights	Access rights to the IO-Link master	rw

rw ... read and write

If in IoT and EtherNet/IP projection software the parameter [Access Rights] is = [EtherNet/IP + IoT], the parameter values set in the EtherNet/IP projection software will always apply.

If in IoT the parameter [Access Rights] is = [IoT only], set the parameter [Access Rights] = [Keep settings] in the EtherNet/IP projection software.

If in LR DEVICE the parameter [Access Rigts] is = [EtherCAT + IoT (read-only)], write access to the device configuration via LR DEVICE and IoT core services is blocked. To enable write access again, set the parameter to [EtherCAT + IoT] via fieldbus configuration software.

Changes of the parameter [Access Rights] will only be effective after restarting the IO-Link master (→ **Firmware: Reboot the device** (→ S. 35)).

9.2.8 IoT: Configuring the LR AGENT or LR SMARTOBSERVER interface

59786

Substructure: `iotsetup`

Avalable data points:

Name	Description	Access
../smobip	IP address of the LR SMARTOBSERVER	rw
../smobport	Port number of the LR SMARTOBSERVER	rw
../smobinterval	Cycle time for data transmission to LR SMARTOBERVER (value in milliseconds)	rw

rw ... read and write

9.2.9 IO-Link ports: Limiting current values

59805

Substructure: `iolinkmaster/port[n]/powercontrol` (n = 1...4).

Available data points:

Name	Description	Access
../mode_ua	Operating mode of supply UA (pin 2) of the port	rw*
../current_us_max	Max. current value of the supply voltage US (pin 1 and 4) at the port (value in mA)	rw*
../current_us	Present current value of supply US (pin 1 and 4) at the port (value in mA)	r
../current_ua_max	Max. current value of the supply voltage UA (pin 2) at the port (value in mA)**	rw*
../current_ua	Present current value of supply voltage UA (pin 2) at the port (value in mA)**	r
../status	Status of the supply voltages US and UA at the port	r

r ... read only
rw ... read and write
* ... only changeable, if the EtherNet/IP plc is not in RUNNING state
** ... only available for IO-Link ports class B

9.2.10 IO-Link ports: Setting the operating mode of pin 4 (US)

59793

Substructure: `iolinkmaster/port[n]` (n = 1...4).

Available data points:

Name	Description	Access
../mode	Operating mode of the IO-Link port	rw*
../mastercycletime_preset	Cycle time of the data transfer at the IO-Link port (value in ms)	rw*
../mastercycletime_actual	Current cycle time of the data transfer at the IO-Link port (value in ms)	r
../comspeed	Data transfer rate of the IO-Link port	r

r ... read only
rw ... read and write
* ... only changeable, if the <Feldsbus> plc is not in RUNNING state

9.2.11 IO-Link ports: Configuring device validation and data storage

59792

Substructure: `iolinkmaster/port[n]` (n = 1...4).

Available data points:

Name	Description	Access
../validation_datastorage_mode	Response of the IO-Link port when a new IO-Link device is connected	rw*
../validation_vendorid	IO-Link ID of the manufacturer that is to be validated	rw*
../validation_deviceid	IO-Link ID of the device that is to be validated	rw*
../datastorage	Structure for port data storage	rw
../datastorage/maxsize	Maximum size of the data storage content (in bytes)	r
../datastorage/chunksize	Size of a data segment (in bytes)	r
../datastorage/size	Size of the data storage content (in bytes)	r

r ... read only
rw ... read and write
* ... can only be changed if the EtherNet/IP PLC is not in RUNNING state

Applicable services:

Service	Description
../validation_useconnecteddevice	Validate the IO-Link device connected to the IO-Link port*
../datastorage/getblobdata	Reading the content of the data storage area
../datastorage/stream_set	Transfer an individual data segment*
../datastorage/start_stream_set	Start sequential transmission of several data segments*

* ... can only be changed if the EtherNet/IP PLC is not in the RUNNING state

Example: Clone the Data Storage of an IO-Link port

52344

Task: Save the Data Storage of IO-Link port X02 of IO-Link master 1 and restore the data at IO-Link master 2.

Solution: The cloning process consists of 2 steps. In the first step, the Data Storage of the IO-Link port of IO-Link master 1 is saved. In the second step, the saved data is restored at the Data Storage of port IO-Link port of IO-Link master 2.

Save Data Storage:

1 Preparations

- ► Read size of segments of Data Storage (h = number of bytes):
  {"code":"request", "cid": -1,"adr":"/iolinkmaster/port[2]/datastorage/chunksize/getdata"}
  Example: h = 256
- ► Read total size of Data Storage area (g = number of bytes):
  {"code": "request", "cid": -1, "adr": "/iolinkmaster/port[2]/datastorage/size/getdata"}
  Example: g = 550
- ► Calcuate the number of reading steps n: n = first integer value to which the following applies: g < n*h
  Example: n= 3, because 550 < 3*256

2 Read Data Storage of IO-Link port

- ► Read Data Storage segment by segment ("pos" is the byte offset, at which the reading process with length "length" starts).
  {"code": "request", "cid": -1, "adr": "/iolinkmaster/port[2]/datastorage/getblobdata", "data": {"pos": 0, "length": h}}
  {"code": "request", "cid": -1, "adr": "/iolinkmaster/port[2]/datastorage/getblobdata", "data": {"pos": h, "length": h}}
  {"code": "request", "cid": -1, "adr": "/iolinkmaster/port[2]/datastorage/getblobdata", "data": {"pos": 2*h, "length": h}}
  ...

{"code": "request", "cid": -1, "adr": "/iolinkmaster/port[2]/datastorage/getblobdata", "data": {"pos": n*h, "length": h}}
Example:
1st read request: pos = 0, length = 256
2nd read resquest: pos = 256, length =256
3rd read request: pos = 512, length = 256

> Each segment value will be returned as BASE64 coded string.

► Join segments.

Restore Data Storage:

1 Preparations

► Determine the size of the saved Data Storage value (n = number of bytes).
Example: n = 550

► Read size of segments (s = number of bytes):
{"code":"request", "cid": -1,"adr":"/iolinkmaster/port[1]/datastorage/chunksize/getdata"}
Example: s = 256

2 Transfer Data Storage strings

► Start transfer of Data Storage string ("size" = size of Data Storage string):
{"code":"request", "cid": -1, "adr":"/iolinkmaster/port[1]/datastorage/start_stream_set", "data": {"size": n}}
Example: size = 550

► Transfer Data Storage string segment by segment ("value" = string value of length s):
{"code": "request", "cid": -1, "adr": "/iolinkmaster/port[1]/datastorage/stream_set", "data": {"value": "aWZtfgIAAABBTDF4NXhfY25faXRfdDIuMi43Nw..."}

9.2.12 IO-Link ports: Configuring data transfer to LR AGENT or LR SMARTOBSERVER

59795

Substructure: `iolinkmaster/port[n]` (n = 1...4).

Available data points:

Name	Description	Access
../senddatatosmob	Process data to LR AGENT or LR SMARTOBSERVER	rw

rw ... read and write

9.2.13 IO-Link ports: Reading and writing process data

59801

Substructure: `iolinkmaster/port[n]` (n = 1...4)

Available data points:

Name	Description	Access
../pin2out	Value of the digital output on pin 2 of the IO-Link port	rw*
../iolinkdevice/pdin	Value of the IO-Link input on pin 4 of the IO-Link port	r
../iolinkdevice/pdout	Value of the IO-Link output on pin 4 of the IO-Link port	rw*

r = only read
rw = read and write
* = only changeable, if the EtherNet/IP plc is not in RUNNING state

Example: Read IO-Link process data (operating mode "IO-Link")

33842

Task: Read the current measured value of the ifm temperature sensor TN2531 at IO-Link port X02

Solution: Read the data point for the process input data with the `getdata` service.

- Request:

```
{
"code":"request",
"cid":4711,
"adr":"/iolinkmaster/port[2]/iolinkdevice/pdin/getdata"
}
```

- Response:

```
{
"cid":4711,
"data":{"value": "03C9"},
"code":200
}
```

The return value is given in hexadecimal format. Besides the temperature value the return value comprises additional information (→ IO Device Description (IODD) of the sensor). The temperature value is shown in bits 2 to 15.

0x03C9 = 0b**1111001001**

Temperature value: 0b11110010 = 242

Therefore: The current temperature value is 24.2 °C.

Example: Writing IO-Link value (operating mode "IO-Link")

59804

Task: Switch on the buzzer of DV2500 at IO-Link Port X2. The DV2500 operates in On/Off mode.

Solution: The IODD of the DV2500 shows the structure of the IO-Link process value (→ e.g. LED activity). The buzzer will be switched using bit 40 of the process value (OFF = 0, ON = 1).

To switch the buzzer:

1. Read the current process value (→ **Example: Read IO-Link process data (operating mode "IO-Link")** (→ S. 52)).
2. Set bit 40 of the read value to 1.
3. Write the process value to the IO-Link device.

Example:

Read process value:

0x0000 0000 004D = 0b0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0100 1101

New process value:

0b0000 000**1** 0000 0000 0000 0000 0000 0000 0000 0000 0100 1101 = 0x0100 0000 004D

- Request:

```
{
"code":"request",
"cid":10,
"adr":"iolinkmaster/port[2]/iolinkdevice/pdout/setdata",
"data":{"newvalue":"01000000004D"}
}
```

- Response:

```
{
"cid":10,
"code":200
}
```

Example: Writing digital output (operating mode "DO")

59803

Task: Set the output value of the IO-Link devices at IO-Link Port X1 to "ON". The operating mode of the IO-Link port is "Digital Output (DO)".

Solution: Write the value 1 to data point pdout. The value has to be written as hexadecimal value with a length of 1 byte (OFF = "00", ON = "01").

- Request:

```
{
"code":"request",
"cid":10,
"adr":"iolinkmaster/port[1]/iolinkdevice/pdout/setdata",
"data":{"newvalue":"01"}
}
```

- Response:

```
{
"cid":10,
"code":200
}
```

Example: Reading digital input (operating mode "DI")

59802

Task: Read the current input value of the IO-Link device at IO-Link port X5. The operating mode of the IO-Link port is "Digital Intput (DI)".

Solution: Read the value of data point pdin. The value will be returned as hexadecimal value with a length of 1 byte (OFF = "00", ON = "01"").

- Request:

```
{
"code":"request",
"cid":10,
"adr":"iolinkmaster/port[5]/iolinkdevice/pdin/getdata"
}
```

- Response:

```
{
"cid":10,
"data":{"value":"00"},
"code":200
}
```

9.2.14 IO-Link ports: Indicating port events

59796

Substructure: iolinkmaster/port[n] (n = 1...4).

Available data points:

Name	Description	Access
../portevent	Indication of the following events at IO-Link port n: ▪ plugging IO-Link device ▪ pulling IO-Link device ▪ changing operating mode of IO-Link port	r

r ... read only

Subscribing events: → **Subscribing to notifications** (→ S. 61)

9.2.15 IO-Link devices: Accessing parameters

59800

The ifm IoT Core supports the configuration of the connected IO-Link devices. A parameter is accessed via IO-Link index and subindex (→ IO Device Description (IODD) of the device).

Substructure: `iolinkmaster/port[n]/iolinkdevice` (n = 1...4)

Applicable services:

Service	Description
../iolreadacyclic	Read a parameter of an IO-Link device (acyclic)
../iolwriteacyclic	Write a parameter of an IO-Link device (acyclic)

Example: Read the parameter value of an IO-Link device

33847

Task: Read the serial number of the ifm temperature sensor TN2531 at IO-Link port X02

Solution: Read the serial number with the `iolreadacyclic` service from the IO-Link device (index: 21, subindex: 0)

- **Request:**

```
{
"code":"request",
"cid":4711,
"adr":"/iolinkmaster/port[2]/iolinkdevice/iolreadacyclic",
"data":{"index":21,"subindex":0}
}
```

- **Return:**

```
{
"cid":4711,
"data":{"value":"4730323134323830373130"},
"code":200
}
```

The returned value is given in hexadecimal format. The conversion of the HEX value in a STRING value is: G0214280710

Example: Change the parameter value of an IO-Link device

33844

Task: Set the output configuration OUT1 of the ifm temperature sensor TN2531 at IO-Link port X02 to the value "Hnc / hysteresis function, normally closed".

Solution: Change the parameter [ou1] of the sensor to the value 4 using the iolwriteacyclicdata service. The parameter can be accessed via IO-Link index 580, subindex 0 (→ IO-Link description of the sensor).

- Request:

```
{
"code":"request",
"cid":4711,
"adr":"/iolinkmaster/port[2]/iolinkdevice/iolwriteacyclic",
"data":{"index":580,"subindex":0,"value":"34"}
}
```

The value has to be given in hexadecimal format. The conversion of the STRING value in a HEX value is: 34.

- Response:

```
{
"cid":4711,
"code":200
}
```

9.2.16 IO-Link devices: Reading an writing device information

59797

Substructure: `iolinkmaster/port[n]/iolinkdevice` (n = 1...4)
Available data points:

Name	Description	Access
../status	Status of the connected IO-Link device	r
../vendorid	IO-Link ID of the vendor	r
../deviceid	IO-Link ID of the IO-Link device	r
../productname	Product name of the IO-Link device	r
../serial	Serial number of the IO-Link device	r
../applicationspecifictag	Device-specific identification (application tag)	rw

r ... read only
rw ... read and write

9.2.17 IO-Link devices: Indicating IO-Link events

59798

Substructure: `iolinkmaster/port[n]/iolinkdevice` (n = 1...4).
Available data points:

Name	Description	Access
../iolinkevent	Indication of IO-Link events	r

r ... read only

Subscribing events: → **Subscribing to notifications** (→ S. 61)

9.2.18 Gateway: Resetting, rebooting and localising the device

59790

Substructure: `firmware`
Applicable services:

Name	Description
../factoryreset	Reset IO-Link master to factory settings
../reboot	Reboot IO-Link master
../signal	Trigger the flashing of the status LED

9.2.19 Gateway: Reading device information

59787

Substructure: `deviceinfo`

Available data points:

Name	Description	Access
../productcode	Article number	r
../vendor	Producer	r
../devicefamily	Device family	r
../hwrevision	Hardware revision	r
../serialnumber	Serial number	r
../swrevision	Firmware version	r
../bootloaderrevision	Bootloader version	r
../extensionrevisions	Firmware and bootloader version	r
../fieldbustype	Fieldbus	r

r ... read only

Additional information about the AL1421 can be read with the `getidentity` service (→ **Service: getidentity** (→ S. 158)).

9.2.20 Gateway: Reading status and diagnostic information

59788

Substructure: `processdatamaster`

Available data points:

Name	Description	Access
../temperature	Temperature of the IO-Link master (value in °C)	r
../voltage	Voltage applied (value in mV)	r
../current	Current (value in mA)	r
../supervisionstatus	Diagnostic information of the device supply	r
../voltage_ua	Voltage value of power supply UA (value in mV)	r
../current_ua	Current value of power supply UA (value in mA)	r
../supervisionstatus_ua	Status of power supply UA	r

r ... read only

9.2.21 Gateway: Updating the firmware

59789

Substructure: `firmware`

Available data points:

Name	Description	Access
../version	Software version	r
../type	Software type	r
../container	Structure for updating the firmware	w
../container/maxsize	Maximum size of the container structure (in bytes)	r
../container/chunksize	Size of a data segment (in bytes)	r
../container/size	Size of the container content (in bytes)	r

r = only read
w = write only

Applicable services:

Name	Description
../install	Install firmware transferred to the IO-Link master
../container/stream_set	Transfer an individual data segment
../container/start_stream_set	Start sequential transmission of several data segments

Example: Update firmware

52252

Task:

Update the firmware of the device; size of the firmware file: 356676 bytes

Solution:

The firmware is transferred to the device in fragments (chunks). The size of the fragments depends on the size of the flash memory of the IO-Link master. To transfer the firmware, the firmware file must be converted into a character string using BASE64.

1 Preparations

► Determine the size of the fragments (g = number of bytes):
{"code":"request", "cid": -1, "adr":"/firmware/container/chunksize/getdata"}

► Convert the firmware file into a BASE64 string.

2 Start the transfer of the firmware

► Start the transfer of the firmware via the service start_stream_set (parameter "size": size of the firmware file):
{"code":"request", "cid": -1, "adr":"/firmware/container/start_stream_set", "data":{"size":356676}}

3 Load the firmware into the flash memory of the IO-Link master

► Send the BASE64 string of the firmware file to the IO-Link master fragment by fragment (value = string value with length g).
{"code": "request", "cid": -1, "adr": "/firmware/container/stream_set", "cid": -1, "data": {"value": "aWZtfgIAAABBTDF4NXhfY25faXRfdDIuMi43Nw..."}

► Repeat step 3 until all fragments of the firmware file have been sent to the IO-Link master.

> IO-Link master stores the segments received in the container area.

4 Install firmware

► Start the installation of the transmitted firmware.
{"code": "request", "cid": -1, "adr": "/firmware/install", "data": {}}

9.2.22 Gateway: Setting the application tag

59791

Substructure: `devicetag`

Available data points:

Name	Description	Access
../applicationtag	Name of the IO-Link master (application tag)	rw

rw ... read and write

For the storage of the applicationtag 32 bytes are available on the IO-Link master. If the memory area is exceeded during writing with setdata, the IoT core aborts the write process and returns the diagnostics code 400.

When writing the application tag, note the different memory requirements of the individual UTF-8 characters:

- characters 0-127: 1 byte per character
- characters >127: more than 1 byte per character

Example: Change name of the IO-Link master

a33823

Task: Set the name of the IO-Link master to AL1421 for the representation in the LR SMARTOBSERVER.

Solution: Change the parameter [Application Tag] with the setdata service to the value [AL1421].

The data point of the parameter [Application Tag] in the device description object is /devicetag/applicationtag.

- Request:

```
{
"code":"request",
"cid":4711,
"adr":"/devicetag/applicationtag/setdata",
"data":{"newvalue":"AL1421"}
}
```

- Response:

```
{"cid":4711,"code":200}
```

9.2.23 Subscribing to notifications

61159

If a data point has the sub-element `datachanged`, the user can subscribe to notifications on value and condition changes. Notifications can be triggered by the expiration of a timer or an event. The IoT Core supports the output of notifications in CSV or JSON format.

Available data points:

Name	Description	Access
timer[x]/counter	Timer for triggering a notification	rw
timer[x]/interval	Cycle time of the update of the subscribed values	rw
iolinkmaster/port[n]/portevent	Display of the following events on IO-Link port n: ▪ IO-Link device connected ▪ IO-Link device disconnected ▪ Operating mode of the IO-Link port changed	rw
iolinkmaster/port[n]/iolinkdevice/iolinkevent	Display of IO-Link events	rw

r ... read only
rw ... read and write
x = [1,2]
n = 1...4

Applicable services:

Name	Description
../datachanged/subscribe	Subsrscibe to notification
../datachanged/unsubsribe	Unsubscribe notification
../datachanged/getsubscriptioninfo	Show information about notifications

Additionally, the user can use **Service: getsubscriberlist** (→ S. 159) show all active subscriptions.

Example: Subscribing to notifications

33853

Task: The current values of the following parameters are to be sent regularly to a network server with IP address 192.168.0.4:

- cyclic input data of the IO-Link Devices an IO-Link port X02
- Operating temperature of the IO-Link master.

Solution: Subscribe to the required data using the subscribe service.

The following options are additionally available:

- via WebSockets (ws://): **Example: Subscribing notifications via WebSocket** (→ S. 64)

- Request:

```
{
"code":"request",
"cid":4711,
"adr":"/timer[1]/counter/datachanged/subscribe",
"data":
{
"callback":"http://192.168.0.4:80/temp",
```

```
"datatosend":[
"/iolinkmaster/port[2]/iolinkdevice/pdin",
"/processdatamaster/temperature"]
}
}
```

In addition, the time interval of the timer[1] must be set to a value between 500 ms and 2147483647 ms.

- Request:

```
{
"code":"request",
"cid":4712,
"adr":"/timer[1]/interval/setdata",
"data":{"newvalue":500}
}
```

- Response:

```
{
"cid":4712,
"code":200
}
```

- Notification (JSON)

```
{
"code":"event",
"cid":4711,
"adr":"",
"data":{
"eventno":"6317",
"srcurl":"/timer[1]/counter/datachanged",
"payload":{
"/timer[1]/counter":{"code":200,"data":1},
"/processdatamaster/temperature":{"code":200,"data":39},
"/iolinkmaster/port[2]/iolinkdevice/pdin":{"code":200,"data":"03B0"}}}
}
```

Example: Changing a subscription

61161

Task: The existing subscription (**Example: Subscribing to notifications** (→ S. 61)) is to be changed. Instead of the temperature of the IO-Link master, the operating voltage applied is to be transmitted.

Solution: Overwrite the existing subscription. For this purpose, the parameter values for "cid" and "callback" in the request must be the same as those of the existing subscription.

- Request:

```
{
"code":"request",
"cid":4711,
"adr":"/timer[1]/counter/datachanged/subscribe",
"data":{
"callback":"http://192.168.0.4:80/temp",
"datatosend":[
"/iolinkmaster/port[2]/iolinkdevice/pdin",
"/processdatamaster/voltage"]}
}
```

Example: Unsubscribing from notifications

61163

Task: The existing subscription (**Example: Subscribing to notifications** (→ S. 61)) is to be deleted.

Solution: Use the unsubscribe service to delete the subscription. For this purpose, the value of the parameter "callback" in the request must be equal to the value of the existing subscription.

```
{
"code":"request",
"cid":4711,
"adr":"/timer[1]/counter/datachanged/unsubscribe",
"data":{
"callback":"http://192.168.0.4:80/temp"
}
```

Example: Checking subscriptions

61164

Task: Information about the existing subscription (**Example: Subscribing to notifications** (→ S. 61)Show **Example: Subscribing to notifications** (→ S. 61)).

Solution: Use the service getsubscriptioninfo and the parameter values cid, "adr" and "callback" of the existing subscription to retrieve the information.

- Request:

```
{
"code":"request",
"cid":4711,
"adr":"/timer[1]/counter/datachanged/getsubscriptioninfo",
"data":{
"callback":"http://192.168.0.4:80/temp"}
}
```

- Response:

```
{
"cid." 4711,
"data":{
"callback":"http://192.168.0.4:80/temp",
"datatosend":[
"/iolinkmaster/port[2]/iolinkdevice/productname",
"/iolinkmaster/port[2]/iolinkdevice/pdin",
"/processdatamaster/temperature"]},
"code":200
}
```

9.2.24 Using Web Socket

61349

The IoT Core supports communication via WebSocket protocol. With Web Sockets, the user can establish a full-duplex communication channel via a TCP connection.

WebSockets can be used for the following services:

- subscribe / unsubsribe

Maximum number of WebSocket connections: 1
Fail-safe WebSocket connections (wss://) are not supported.

To transmit notifications via a WebSockets connection:

► Establish the WebSocket connection (e.g. "`ws://192.168.0.55:80/websocket`")

- Option 1. without parameter "callback"

► make subscribe/unsusbribe request without parameter "callback".

> IoT-Core sends notifications about existing WebSocket connections.

- Option 2. with parameter "callback"

► make subscribe/unsubscribe requests with parameter "callback" ("`ws:///myTopic`").

> IoT-Core sends notifications about existing WebSocket connections to the topic `myTopic`.

Example: Subscribing notifications via WebSocket

61166

Task: The current values of the following parameters are to be sent regularly to the data sink `myTopic` via an existing WebSocket connection:

- Product name of the IO-Link Devices an IO-Link port X02
- cyclic input data of the IO-Link Devices an IO-Link port X02
- Operating temperature of the IO-Link master.

Solution: Subscribe to the required data using the subscribe service.

- Request:

```
{
"code":"request",
"cid":4711,
"adr":"/timer[1]/counter/datachanged/subscribe",
"data":{
"callback":"ws:///myTopic",
"datatosend":[
"/iolinkmaster/port[2]/iolinkdevice/productname",
"/iolinkmaster/port[2]/iolinkdevice/pdin",
"/processdatamaster/temperature"]}
}
```

If the notifications are to be transmitted via the existing WebSocket connection, but without a special data sink, the callback parameter is not required.

- Request:

```
{
"code":"request",
"cid":4711,
"adr":"/timer[1]/counter/datachanged/subscribe",
"data":{
"datatosend":[
```

```
"/iolinkmaster/port[2]/iolinkdevice/productname",
"/iolinkmaster/port[2]/iolinkdevice/pdin",
"/processdatamaster/temperature"]}
}
```

9.2.25 Using the IoT-Core Visualizer

Content

Managing notifications 67
Searching for elements in the device tree 69
Configuring IO-Link the master 70
Reading and writing process data 71
Updating the firmware 72

61173

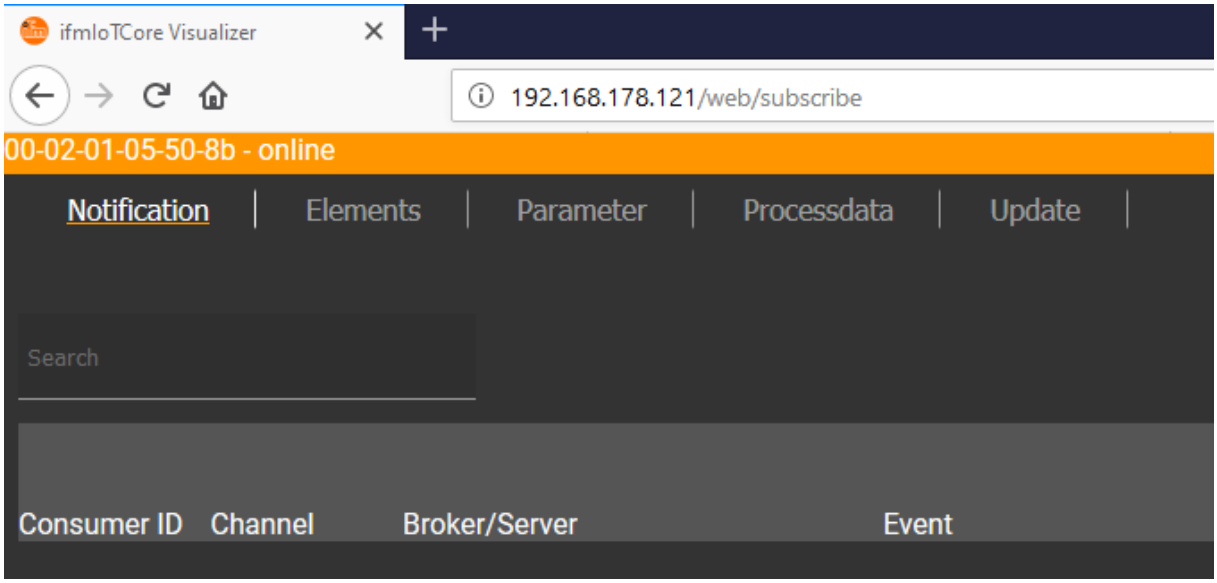
The ifm-IoT Core Visualizer of the IO-Link master provides a graphical user interface for accessing functions of the ifm-IoT Core.

To start the IoT Core Visualizer:

► Start web browser.

► Call the following address: http://ipaddress/web/subscribe

> Browser shows IoT Core Visualizer:



The navigation menu gives the user access to the following functions:

- [Notification]: Creating and managing notifications (subscribe / unsubscribe)
- [Elements]: Searching for elements in device description
- [Parameter]: Configuring IO-Link master
- [Processdata]: Reading and writing process data
- [Update]: Updating the firmware of the IO-Link master

Managing notifications

61174

The menu page allows you to perform the following functions

- Creating notifications
- Showing active notifications
- Deleting notifications (single, all)

Requirements:

- Iot-Core Visualizer has been started.

► Click on [Notification].

> The menu page for managing notifications appears.

> The menu page shows all registered notifications in a table

Creating a new notification

61175

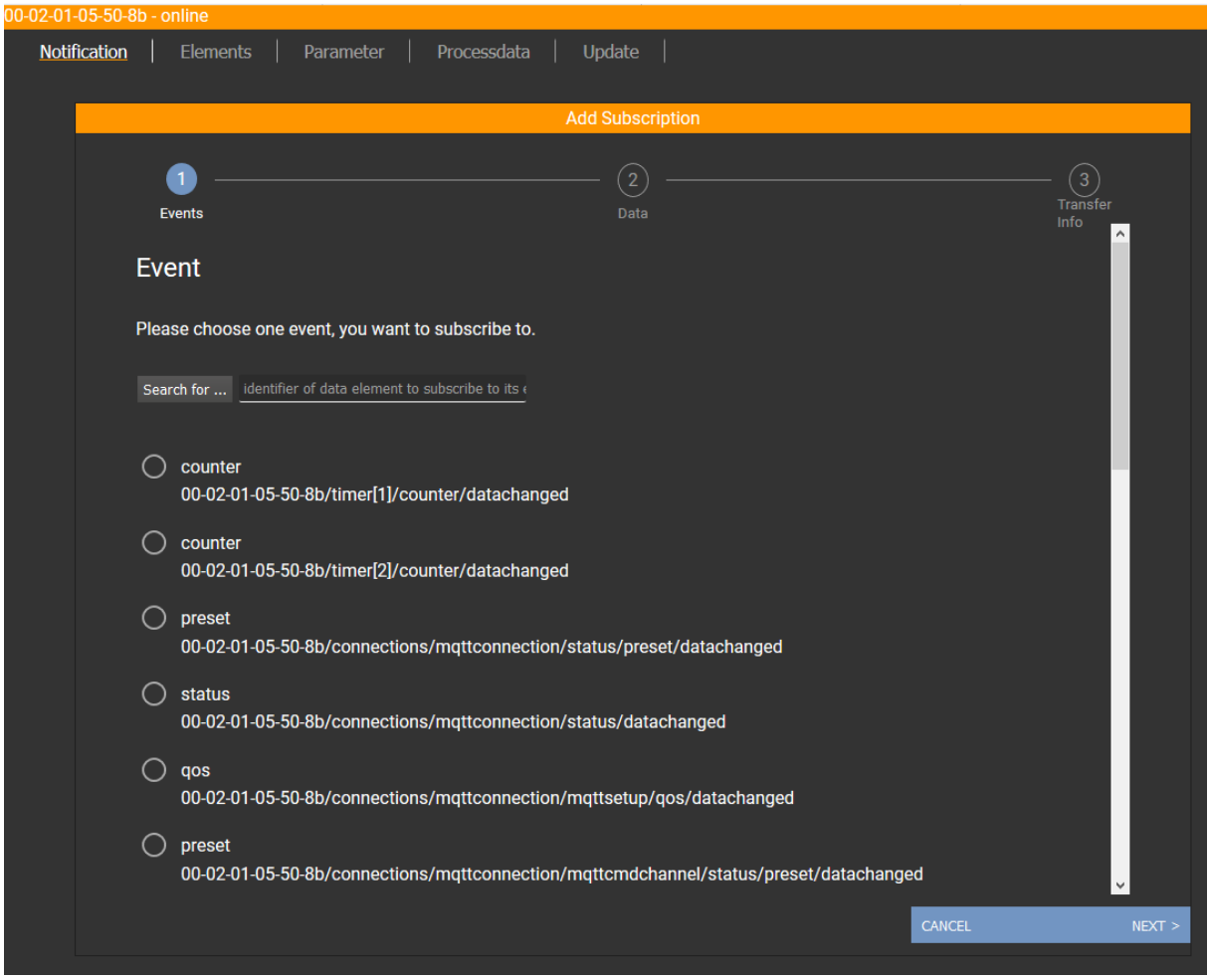
A wizard is used to register new notifications.

Requirements:

- The [Notification] menu page is open.

► Click on [+] on the right side of the table.

> The wizard for the creation of notifications appears.



► Use the wizard to enter the required notification parameters step by step.
> Created notification subscription is displayed in the table.

For cyclical notifications via timer[1] or timer[2], the user also needs to set the interval time of the timer in question.

Deleting a notification

61176

Requirements:

- The [Notification] menu page is open.
- At least one notification is active.

► Click on [x] in the column [Unsubscribe].
> The selected notification will be deleted (unsubscribe).

Searching for elements in the device tree

61177

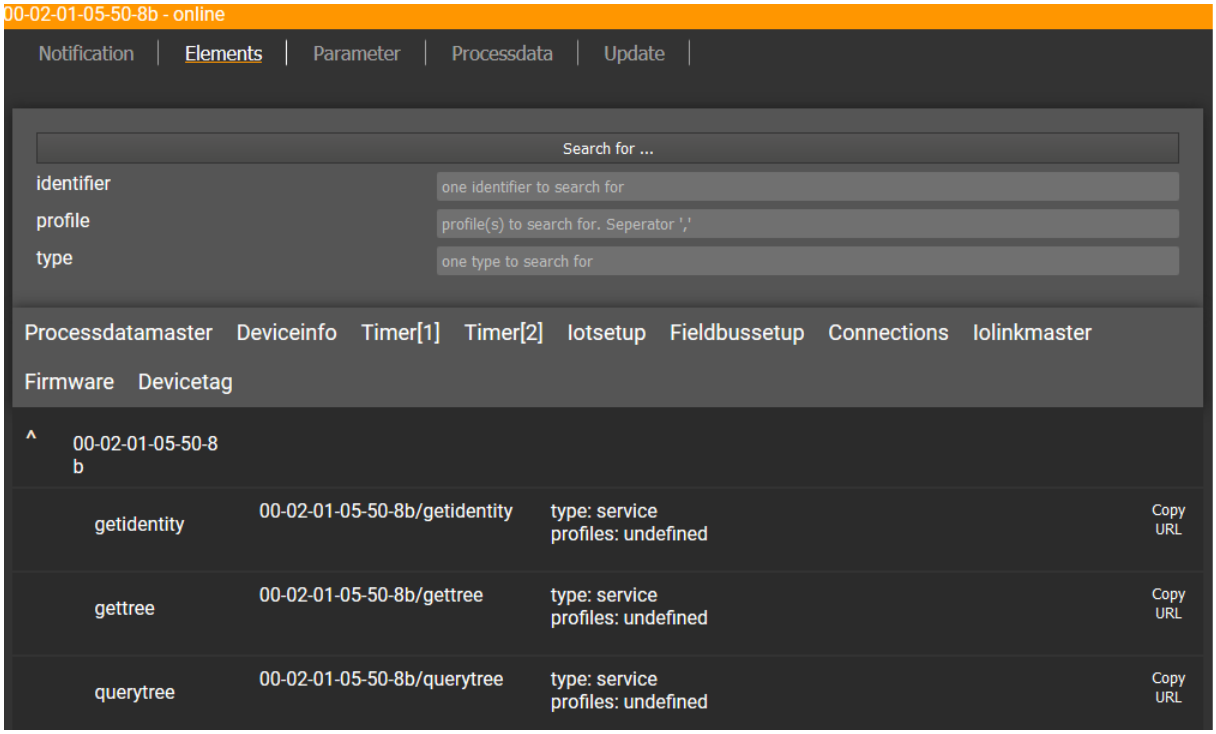
The [Elements] menu page allows you to search the device description for elements with specific properties (status, profile, name) and to output the results.

Requirements:

- Iot-Core Visualizer has been started.

► Click on [Elements].

> The input mask appears.



► Enter the search criteria of the required item in the [identifier], [profile] and [type] boxes.

► Click on [Search for ...].

> IoT-Core Visualizer searches device description for elements with selected search criteria.

> The result list shows all elements found.

Configuring IO-Link the master

61178

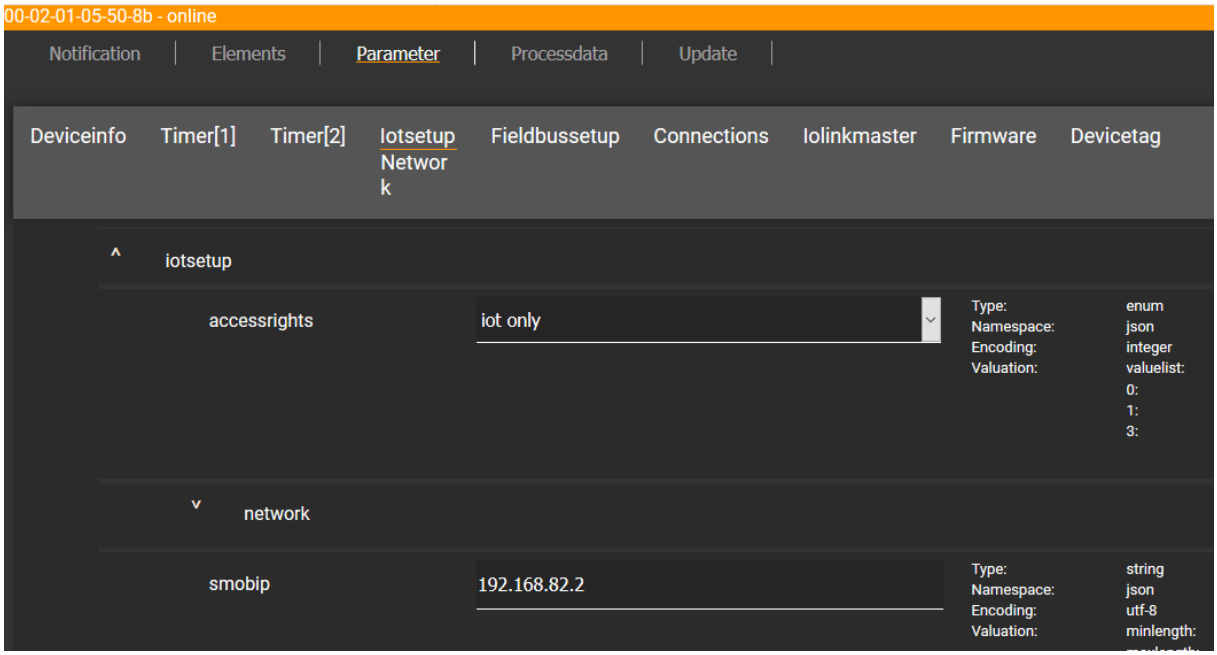
The [Parameter] menu page allows you to configure the IO-Link master.
Available options:

- Reading and writing individual parameters
- Backup and restore the current configuration of the machine.

Requirements:

- Iot-Core Visualizer has been started.

► Click on [Parameter].
> The menu page shows the available parameters of the IO-Link master.
> Current parameter values are displayed.
> Editable parameters can be changed.



To change a parameter:

► Navigate to the desired parameter in the device description.
► Changing the parameter value
► Click on the pencil icon to save the change on the IO-Link master.
> The changed parameter value is active.
► Optional: Repeat the procedure to change further parameter values.

Reading and writing process data

61179

The menu page allows the process data of the IO-Link master and the connected IO-Link devices to be read and written.

Requirements:

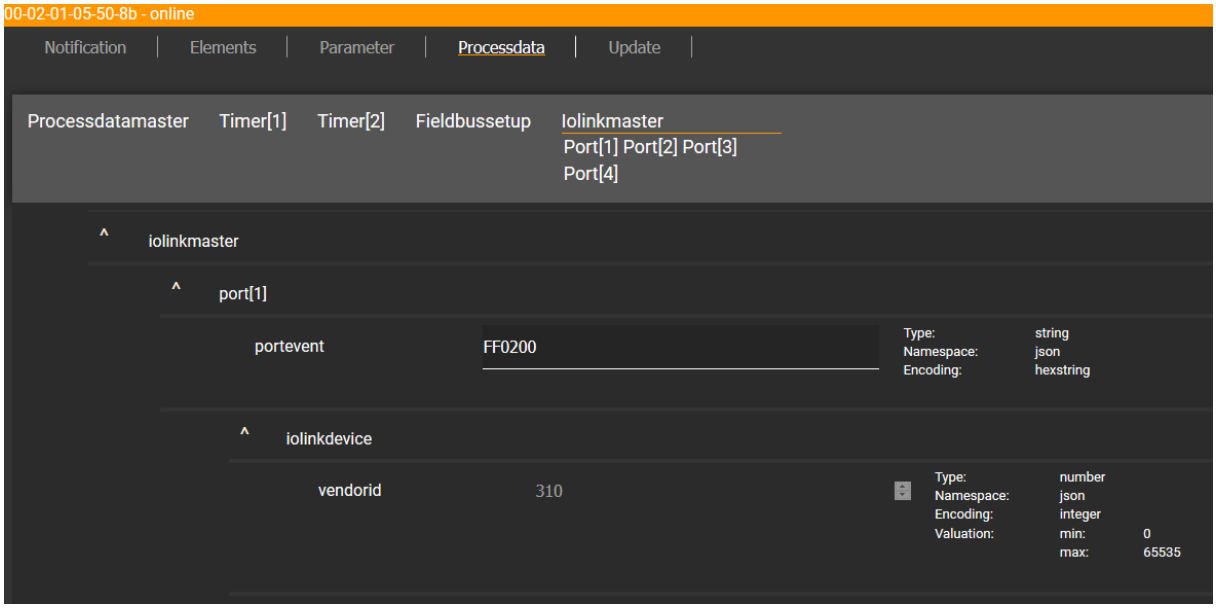
- Iot-Core Visualizer has been started.

► Click on [Processdata].

> Menu page shows the substructures of the device description that contain process data and events.

> The current process values are displayed.

> Editable process data can be changed.



To change the value of a process date:

► Navigate to the required process date in the device description.

► Change the process value.

► Click on the pencil icon to save the change on the IO-Link master.

> The changed process value is active.

► Optional: Repeat the procedure to change further process values.

Updating the firmware

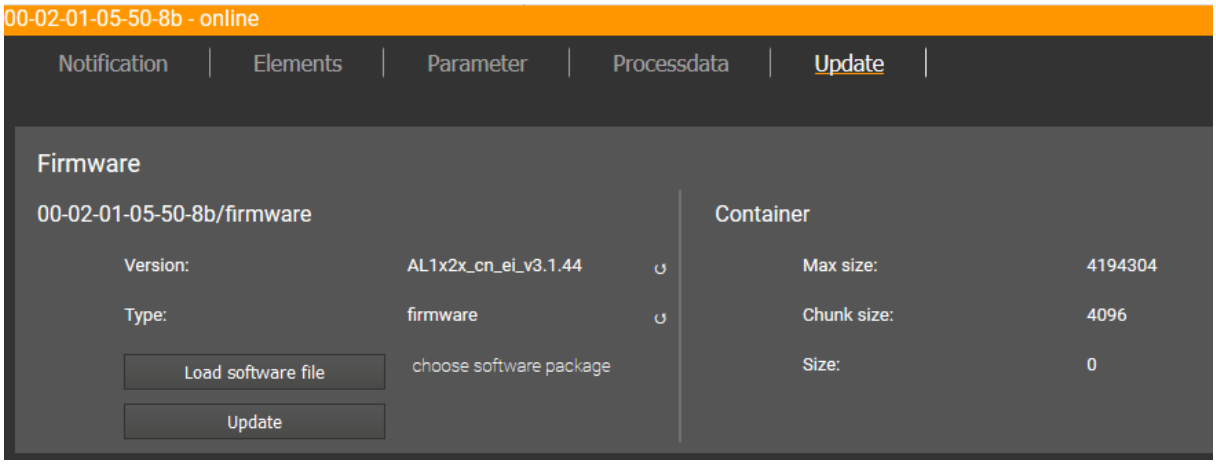
61180

The [Update] menu page allows you to update the firmware of the IO-Link master:
Requirements:

- Iot-Core Visualizer has been started.

► Click on [Update].

> Menu page displays information about the current firmware version.



► Click on [Load software file] and select a new firmware file (*.bin).

► Click on [Update] to start the update process.

> The firmware of the IO-Link master will be updated.

> The area shows the progress bar.

> If the update process has been successful, the IO-Link master will restart automatically.

9.3 EtherNet/IP

Content

EtherNet/IP: Programmers' notes....73
Registration of the EDS file....74
Integrate the IO-Link Master into the EtherNet/IP project....74
Configure connection types....75
Configure the IO-Link master....76
Configure IO-Link ports....76
Configure IO-Link devices....77
Read process data....77
Write process data....78
Read diagnostic information and events....78
Use acyclic services....79

34391

On the fieldbus side, the device can be configured with any EtherNet/IP compatible projection software.

The information in the following sections refers to the EtherNet/IP projection software RSLogix 5000.

9.3.1 EtherNet/IP: Programmers' notes

34397

The programmer can access on the following data from the PLC application:

- Read device information of the AL1421
- Read diagnostics and alarms
- Set parameters of the connected IO-Link devices

The following sections show the available options.

Further information about the functional/operational blocks: → Help function of the EtherNet/IP projection software

Supported configuration options

42783

The AL1421 supports the following EtherNet/IP configuration modes:

- **Top-Down**
  - Configuration of the EtherNet/IP slave (IO-Link master) via EtherNet/IP projecting software (Configuration Assembly)
  - The created configuration is transferred to the EtherNet/IP slave (IO-Link master) via the EtherNet/IP controller.
- **Independent**
  - Configuration of the EtherNet/IP slave (IO-Link master) via LR DEVICE
  - Configuration Assembly in the EtherNet/IP project will not be evaluated.

9.3.2 Registration of the EDS file

34324

ifm provides an EDS file to integrate the AL1421 in a EtherNet/IP projection software. The user can download the EDS file from the ifm website (→ www.ifm.com). In the EDS file, all parameters, process data, and their valid value ranges are defined.

To add the AL1421 to the device catalogue of RSLogix5000:

► Download the EDS file of the AL1421 from the ifm website.
► Start RSLogix5000.
► Select [Tools] > [EDS Hardware Installation Tool].
> EDS Wizard appears.
► Register the downloaded EDS file of the AL1421 with the EDS Wizard.
> EDS Wizard installs the EDS file and adds the AL1421 to the device catalogue.

9.3.3 Integrate the IO-Link Master into the EtherNet/IP project

34392

The AL1421 is integrated as module of an I/O scanner in the EtherNet/IP project.

Requirements:

> The EDS file of the AL1421 is installed (→ **Registration of the EDS file** (→ S. 74)).

1 Create/open EtherNet/IP project

► Start RSLogix 5000.
► Create new EtherNet/IP project.
OR
Open an existing EtherNet/IP project.

2 Configure EtherNet/IP PLC and IO scanner

► Select and configure EtherNet/IP controller and IO scanner.
> EtherNet/IP project includes a EtherNet/IP controller and an IO scanner.

3 Integrate AL1421 in project

► In the Controller Organizer: Right mouse click on the IO scanner.
> Context menu appears.
► In the context menu: Select [New Module...].
> The window [Select Module Type] appears.
► Select AL1421 and click on [Create].
> The [New Module] window appears.
► Enter name and IP address of the AL1421.
► Click on [OK] to adopt the entered values.
> RSLogix 5000 adds AL1421 as sub-element of the IO scanner to the project.

4 Save the project

► Save EtherNet/IP project

9.3.4 Configure connection types

57641

With the connection types, the user defines which process data is transferred in Input Assembly and Output Assembly.

	Process data	
Connection type	**Inputs**	**Outputs**
Exclusive Owner IO + Acyclic + Diag + EnMo	▪ cyclic input data ▪ Port Qualifier Information (PQI) ▪ Acyclic command channel (response) ▪ Diagnostic data (events, status information) ▪ Energy monitoring	▪ cyclic output data ▪ Acyclic command channel (request)
Exclusive Owner IO + Acyclic + Diag	▪ cyclic input data ▪ Port Qualifier Information (PQI) ▪ Acyclic command channel (response) ▪ Diagnostic data (events, status information)	▪ cyclic output data ▪ Acyclic command channel (request)
Exclusive Owner IO	▪ cyclic input data ▪ Port Qualifier Information (PQI)	▪ cyclic output data
Input only	→ IO + Acyclic + Diag	-
Listen only	→ IO + Acyclic + Diag	-

Additionally, the user can set the Request Package Interval (RPI).

To set the connection type:

Requirements:

> AL1421 is correctly integrated in the EtherNet/IP project (→ **Integrate the IO-Link Master into the EtherNet/IP project** (→ S. 74, "**Example: Reading properties of an element**" → S. 43)).

1 Open module settings

► In the Controller Organizer: Double-click on the IO-Link master node

> A dialogue window appears.

2 Set the connection type

► Click on [Change...].

> The [Module Definition] dialogue window appears.

► Select the required connection type from the list [Connections].

► Click on [OK] to adopt the changes.

3 Set the RPI

► Click on the [Connection] tab.

> The connection settings are shown.

► Select the required value in the [RPI] list.

► Click on [OK] to adopt the changes.

9.3.5 Configure the IO-Link master

57642

The user can set the following parameters for the IO-Link master:

- Communication profile
- Length of the process data

Details about the parameters of the IO-Link master: → **Configuration Assembly (Instance 199)** (→ S. 95)

Use the Controller Tag to configure the IO-Link master.

To configure the IO-Link master:

Requirements:

> The IO-Link master is integrated correctly in the EtherNet/IP project (→ **Integrate the IO-Link Master into the EtherNet/IP project** (→ S. 74)).

1 Open controller tags

► In the Controller Organizer: Double click on [Controller Name_of_Project] > [Controller Tags]

> [Controller Tags] window appears.

► In the tree view: Click on [AL1421:C].

> Controller tags for the configuration are displayed.

2 Configure the IO-Link master

► Set the Controller Tags as required.

> Save the project.

9.3.6 Configure IO-Link ports

42788

The user can set the following parameters separately for each IO-Link port:

- Port configuration (operating mode pin 4, cycle time, process data alignment (Swap), device validation and data storage, fail-safe values operating modes "Digital Output" and "IO-Link")
- Operating mode supply voltage UA (pin 2)
- Fail-safe value for operating mode "Digital Output" of the supply voltage UA (Pin 2)
- max. current rating of the supply voltage UA (pin 2) (class B ports)
- max. current rating of the supply voltage US (pin 1 + pin 4)

Details about the parameters of the IO-Link ports: → **Configuration Assembly (Instance 199)** (→ S. 95)

Use the Controller Tags to configure the IO-Link ports.

To configure the IO-Link ports:

Requirements

> The IO-Link master is integrated correctly in the EtherNet/IP project (→ **Integrate the IO-Link Master into the EtherNet/IP project** (→ S. 74)).

1 Open controller tags

► In the Controller Organizer: Double click on [Controller Name_of_Project] > [Controller Tags]

> [Controller Tags] window appears.

► In the tree view: Click on [AL1421:C].

> Controller tags for the configuration are displayed.

2 Configure IO-Link ports

► Set the Controller Tags as required.

► Save the project.

9.3.7 Configure IO-Link devices

34359

The AL1421 supports the configuration of the connected IO-Link devices from the EtherNet/IP projection software. For this, ifm offers the EtherNet/IP object "IO-Link Request" (→ **IO-Link requests (object class: 0x80)** (→ S. 133)). The object enables direct read and write access to IO-Link objects of the IO-Link device (Indexed Service Data Unit (ISDU)). The extent of the configurable parameters depends on the IO-Link device.

The following services are available:

Name	Description	Reference
Read request	Send a request to read an IO-Link object	→ **Read_ISDU** (→ S. 134)
Write request	Send a request to write an IO-Link object	→ **Write_ISDU** (→ S. 137)

Information for the execution of acyclic commands: → **Use acyclic services** (→ S. 79)
Available parameters of the IO-Link devices: → Operating instructions of the IO-Link device

9.3.8 Read process data

34360

The user can access the cyclic input data of the connected sensors and IO-Link devices via the controller tags of the AL1421.

To check the validity of the cyclic process data, evaluate the PQI byte (→ **Mapping: Port Qualifier Information (PQI)** (→ S. 105)).

Even if the fieldbus connection is interrupted, the PQI byte indicates that the process data is valid. This may have an unintended impact on the control process.

► Take suitable measures to detect an interruption of the fieldbus connection.

To access the input data:

► Start RSLogix5000.
► Open a EtherNet/IP project.
► In the project tree: Mouse click on [Controller Tags] > [AL1421.I]
> The window shows the data structure with cyclic input data ([AL1421.I:Data]).
► Link process data to variables.

The mapping of the process data to the data structure [AL1421.I:Data] depends on the configured instance of the input assembly object (→ **Cyclic data** (→ S. 100)).

9.3.9 Write process data

34386

The user can access the cyclic output data of the connected actuators and IO-Link devices via the controller tags of the AL1421.

To check the validity of the cyclic process data, evaluate the PQI byte (→ **Mapping: Port Qualifier Information (PQI)** (→ S. 105)).

Even if the fieldbus connection is interrupted, the PQI byte indicates that the process data is valid. This may have an unintended impact on the control process.

► Take suitable measures to detect an interruption of the fieldbus connection.

To access the cyclic output data:

► start RSLogix5000.

► Open a EtherNet/IP project.

► In the project tree: Mouse click on [Controller Tags] > [AL1421.O]

> The window shows the data structure with cyclic output data ([AL1421.O:Data]).

► Link process data to variables.

The mapping of the process data to the data structure [AL1421:O.Data] depends on the configured instance of the input assembly object (→ **Cyclic data** (→ S. 100)).

9.3.10 Read diagnostic information and events

57643

Diagnostic and status information is a part of the cyclically transmitted process data.

The following information is available:

- Short circuit / overload of the IO-Link ports
- Status of the supply voltages US and UA
- Device ID and vendor ID of the connected IO-Link devices
- Events of the IO-Link ports
- present current and voltage values of the supply voltages US and UA
- present current values of the supply voltages US and UA of the IO-Link ports

To access the diagnostic and status information:

► start RSLogix5000.

► Open the EtherNet/IP project.

► In the project tree: Click on [Controller Tags] > [AL1421.I]

> The window shows the data structure with cyclic input data ([AL1421.I:Data])

► Link diagnostic and status data with variables.

The mapping of the diagnostic and status data to the data structure [AL1421.I:Data] depends on the configured instance of the input assembly object (→ **Cyclic data** (→ S. 100)).

9.3.11 Use acyclic services

34381

The AL1421 offers the following options to execute acyclic commands:

Command channels in cyclic process data

34318

Within the cyclic input and output data, special areas are available for the acyclic data transmission. Both read and write access can be implemented via the areas.

Principle of the command channels

34343

General process of an acyclic communication:

1 Write command request

- ► In the request channel: write requested command data (without [Trigger] bit)
- ► Set [Trigger] = 1.
- > Change of [Trigger] = 1 indicates a new command.
- > In the response channel: all bytes are set to 0.
- > Command processing is started.

2 Check status

- ► In the response channel: check [Handshake] bit.
- If [Handshake] <> 0: command processing completed, continue with step 3.
- If [Handshake] == 0: command is processed, repeat step 2.

3 Read command response

- ► In the response channel: read responsed user data.
- ► In the request channel: set [Trigger] = 0.

Acyclic port commands

34336

For the acyclic access to the configuration of the IO-Link ports of the AL1421, the following commands are available:

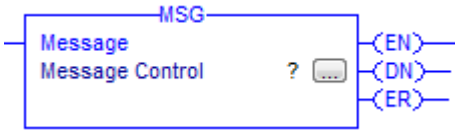
Command	Description	Reference
Set mode	Set the operating type of the IO-Link port	→ **Command 0x10 – Set mode** (→ S. 118)
Set Validation ID / Data Storage	Adjust the supported IO-Link standard and the behaviour of the IO-Link master when connecting a new IO-Link device to the IO-Link port	→ **Command 0x20 – Set validation ID / data storage** (→ S. 120)
Set fail-safe data pattern	Behaviour of the outputs when the EtherNet/IP connection is interrupted and setting of the corresponding fail-safe values	→ **Command 0x30 – Set fail-safe data pattern** (→ S. 122)

The port commands use the same mechanisms as the acyclic command channel (→ **Acyclic command channel** (→ S. 112)).

EtherNet/IP mechanisms for acyclic commands

34404

Acyclic commands can be executed with the EtherNet/IP command `Message (MSG)`.



Parameters of the available field bus objects: → **Field bus objects** (→ S. 124)
For detailed information about the `Message (MSG)` command: → Operating instructions RSLogix 5000

10 Operation

Content

Use web-based management 81

34061

10.1 Use web-based management

57592

The device has an integrated web server The web server generates a website with the following data:

- Status information of the ports
- Diagnostic information of the device
- Version information of the installed firmware components

To access the web interface of the IO-Link master:

► Connect the IO-Link master via port XF1 or XF2 with the laptop / PC.

► Optional: Check the IP settings of the EtherNet/IP interface.

► Start web browser.

► In the address field of the web browser, enter the IP address of the EtherNet/IP interface and confirm with [ENTER].

> The web browser shows the website with the status and diagnostic information of the device.

11 Maintenance, repair and disposal

Content

Cleaning process....82
Updating the firmware....82
Exchanging the IO-Link device....82

51990

The operation of the unit is maintenance-free.

► Dispose of the unit in an environmentally friendly way in accordance with the applicable national regulations when it is no longer used.

11.1 Cleaning process

51991

► Clean the surface of the unit when necessary.
► Do not use any caustic cleaning agents for this!
► In case of severe soiling, use a damp cloth.
► Do not use any caustic cleaning agents for this!

11.2 Updating the firmware

57593

The firmware of the IO-Link master can be updated via the IoT Core Visualizer↘→ **Updating the firmware** (→ S. 82).

► During the firmware update, ensure that the device is connected to the supply voltages US and UA.

11.3 Exchanging the IO-Link device

57594

To exchange an IO-Link device:

Requirements:

> The new IO-Link device is in the default state, as on delivery (factory settings).
> The new IO-Link device supports the IO-Link standard 1.1 or higher.

1 Store the parameter set of the old IO-Link device

► Create the following parameter of the IO-Link port
Device validation and data storage = [Type compatible V1.1 device with Restore]
► Save the changes on the device.
> The IO-Link master saves the parameter values of the connected IO-Link device in the data memory.

2 Exchange IO-Link device

► Disconnect the old IO-Link device from the IO-Link master.
► Connect the new IO-Link device with the same IO-Link port of the AL1421.
> The IO-Link master transfers the parameter values from the data memory to the new IO-Link device.
> The new IO-Link device works with the restored parameter set.

12 Factory Settings

34591

In the factory settings, the device has the following parameter settings:

Parameter	Factory setting
[IP address]	192.168.1.250
[Subnet mask]	255.255.255.0
[IP gateway address]	0.0.0.0
[Host name]	blank
Data Storage	empty

13 Accessories

33870

List of accessories of AL1421: → www.ifm.com > Product page > Accessories

14 Appendix

Content

Technical data 86
EtherNet/IP 94
ifm IoT Core 147

33879

14.1 Technical data

Content

Application 86
Electrical data 86
Inputs/outputs 91
Inputs 91
Outputs 91
Interfaces 92
Environmental conditions 92
Approvals / tests 92
Mechanical data 93
Electrical connection 93

34188

14.1.1 Application

33878

Application	
Application	Hygienic systems; I/O modules for field applications
Daisy-chain function	Voltage supply; Fieldbus interface

14.1.2 Electrical data

57595

Electrical data	
Operating voltage [V]	20...28 DC; (US; according to SELV/PELV)
Current consumption [mA]	300...3900; (US)
protection class	III
Additional voltage supply[V]	20...28 DC; (UA)
Sensor supply US	
Total current rating [A]	3,6
Current rating per port [A]	2; (configurable: 0...2; factory setting: 0.45)
Actuator supply UA	
Total current rating [A]	8
Current rating per port [A]	2; (configurable: 0...2; factory setting: 2)

Derating behaviour

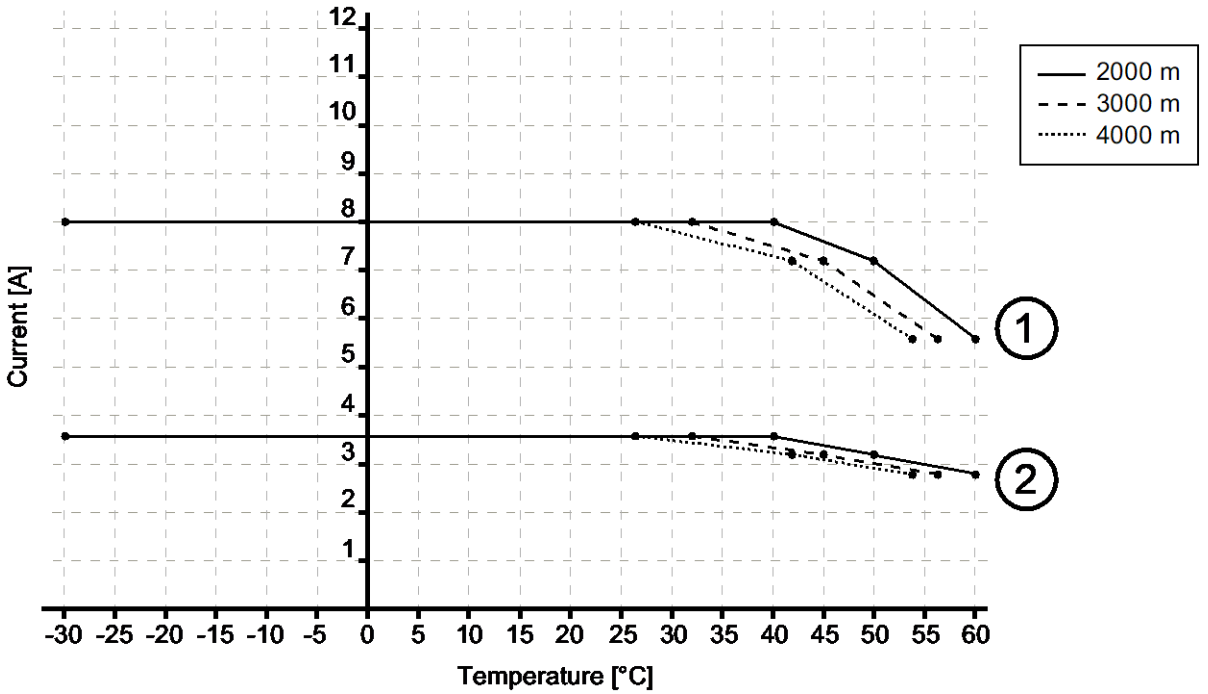
28.06.2020 22:29:00

The derating measurements were performed under the following conditions:

- Supply voltages US / UA: 24 V DC

Derating witout Daisy Chain

59780



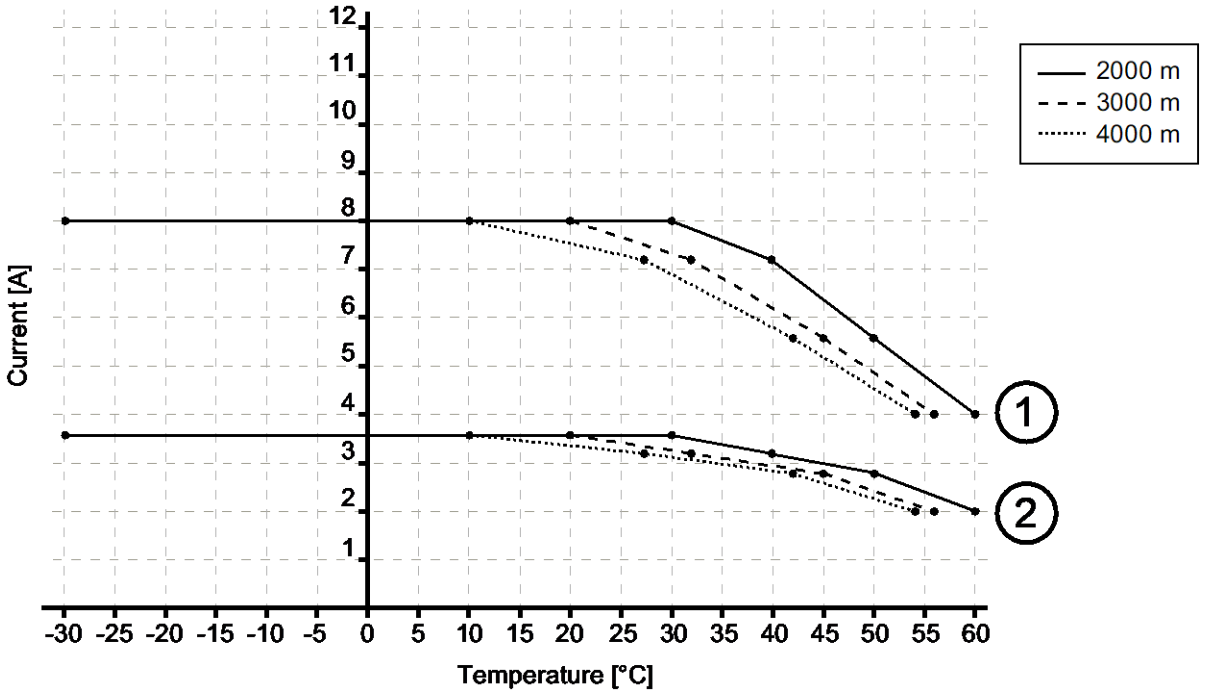
(1) I_{UA} at port X1...X4

(2) I_{US} at port X1...X4

Derating with Daisy Chain

59781

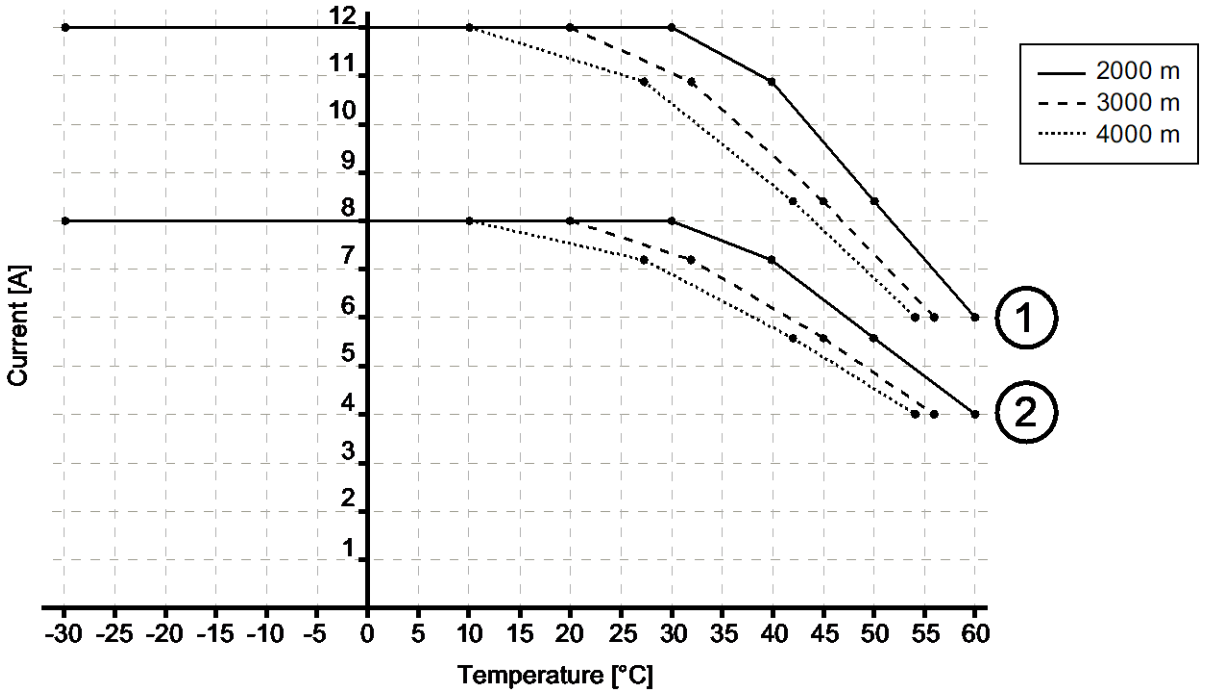
Port X1…X4:



(1) I_{UA} at port X1...X4

(2) I_{US} at Port X1...X4

Port XD2:



① $I_{US\text{-}daisy\text{-}chain}$

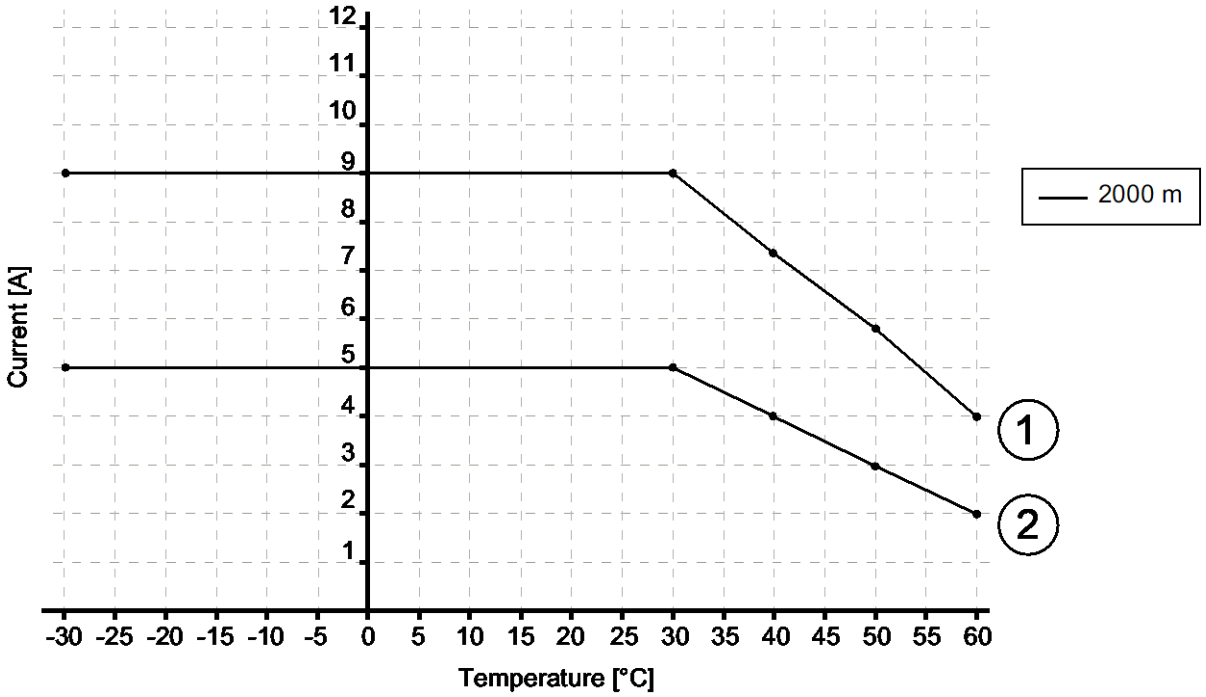
② $I_{UA\text{-}daisy\text{-}chain}$

Example:

Temperature	I_{UA} (X1…X4)	I_{US} (X1…X4)	$I_{UA\text{-}daisy\text{-}chain}$ (XD2)	$I_{US\text{-}daisy\text{-}chain}$ (XD2)
30°C	8 A	3,6 A	8 A	12 A

For UL applications:

Port XD2:



① $I_{US\text{-}daisy\text{-}chain}$

② $I_{UA\text{-}daisy\text{-}chain}$

Example:

Temperature	**I_{UA} (X1...X4)**	**I_{US} (X1...X4)**	**$I_{UA\text{-}daisy\text{-}chain}$ (XD2)**	**$I_{US\text{-}daisy\text{-}chain}$ (XD2)**
30°C	8 A	3,6 A	5 A	9 A

14.1.3 Inputs/outputs

57599

Inputs/outputs	
total number of inputs and outputs	12; (configurable)
Number of inputs and outputs	Number of digital inputs: 4; Number of outputs: 8

14.1.4 Inputs

34069

Inputs	
Number of digital inputs	4; (IO-Link Port Class B: 4x1)
Switching level high [V]	11...28
Switching level low [V]	0...5
Digital inputs protected against short circuits	yes

14.1.5 Outputs

57600

Outputs	
Number of digital outputs	8; (IO-Link Port Class B: 4 x 2)
Short-circuit protection	yes
Actuator supply UA	
Max.current load per output [mA]	2000
Sensor supply US	
Max. current load per output [mA]	2000

14.1.6 Interfaces

57644

Interfaces	
Communication interface	Ethernet; IO-Link
Ethernet	
Transmission standard	10Base-T; 100Base-TX
Transmission rate [MBit/s]	10; 100
Protocol	EtherNet/IP
Factory settings	▪ IP address: 192.168.1.250 ▪ Subnet mask: 255.255.255.0 ▪ Gateway IP address: 0.0.0.0 ▪ MAC address: see type label
Addition	Supported network topologies: Line; Ring; DLR
IO-Link master	
Type of transmission	COM 1 / COM 2 / COM 3
IO-Link revision	1.1
Number of class B ports	4

14.1.7 Environmental conditions

57602

Environmental conditions	
Applications	Indoor use
Ambient temperature [°C]	-25...60 (up to 2000 m) -25...50 (up to 3000 m) -25...40 (up to 4000 m)
Storage temperature [°C]	-25...85
Max. perm. relative air humidity [%]	90
Protection rating	IP 65; IP 66; IP 67; IP 69K; (operation with stainless steel protective caps: IP 69K)
Degree of soiling	2

14.1.8 Approvals / tests

57603

Approvals / tests	
EMC	▪ EN 61000-6-2 ▪ EN 61000-6-4
Shock resistance	DIN EN 60068-2-27
Vibration resistance	▪ DIN EN 60068-2-64 2009-04 ▪ DIN EN 60068-2-6 2008-10

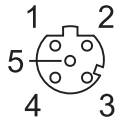
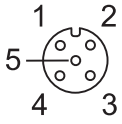
14.1.9 Mechanical data

34050

Mechanical data	
Weight [g]	291.1
Materials	Housing: PA grey; socket: 1.4404 (stainless steel / 316L)

14.1.10 Electrical connection

57604

Voltage supply IN XD1		
Connector	M12 (L-coded)	
Wiring		1: + 24 V DC (US) 2: GND (UA) 3: GND (US) 4: +24 V DC (UA) 5: FE
Voltage supply OUT XD2		
Connector	M12 (L-coded)	
Wiring		1: + 24 V DC (US) 2: GND (UA) 3: GND (US) 4: +24 V DC (UA) 5: FE
Ethernet IN / OUT XF1, XF2		
Connector	M12	
Wiring		1: TX + 2: RX + 3: TX - 4: RX - 5: -
Process connection Ports Class B X1...X4		
Connector	M12	
Wiring		1: Sensor supply (US) L+ 2: Actuator supply (UA) L+ / DO 3: Sensor supply (US) L- 4: C/Q IO-Link 5: Actuator supply (UA) L-

14.2 EtherNet/IP

Content

Supported connection types 94
Parameter data 95
Cyclic data 100
Acyclic data 112
Field bus objects 124

33674

14.2.1 Supported connection types

57645

Name	Configuration Assembly	Input Assembly - Instance	Output Assembly - Instance
Exclusive Owner IO-Acyc-Diag	199	100	150
Exclusive Owner IO-Acyc-Diag-EnMo	199	103	150
Exclusive Owner IO	199	102	151
Input only	199	100	193 (empty)
Listen only	199	100	192 (empty)

14.2.2 Parameter data

Content

Configuration Assembly (Instance 199) 95

34170

Configuration Assembly (Instance 199)

57646

Bytes	Contents
0	Communication Profile
1	Port Process Data Size
2...13	Port X1: Port Configuration (→ **Mapping: Port configuration** (→ S. 97))
14...25	Port X2: Port Configuration (→ **Mapping: Port configuration** (→ S. 97))
26...37	Port X3: Port Configuration (→ **Mapping: Port configuration** (→ S. 97))
38...49	Port X4: Port Configuration (→ **Mapping: Port configuration** (→ S. 97))
50	Port X1: Mode - Pin 2 UA
51	Port X2: Mode - Pin 2 UA
52	Port X3: Mode - Pin 2 UA
53	Port X4: Mode - Pin 2 UA
54	Port X1: Failsafe DO Mode - Pin 2 UA
55	Port X2: Failsafe DO Mode - Pin 2 UA
56	Port X3: Failsafe DO Mode - Pin 2 UA
57	Port X4: Failsafe DO Mode - Pin 2 UA
58...59	Port X1: Current Limit - Pin 2 UA (→ **Mapping: Current Limit - Pin 2 UA** (→ S. 99))
60...61	Port X2: Current Limit - Pin 2 UA (→ **Mapping: Current Limit - Pin 2 UA** (→ S. 99))
62...63	Port X3: Current Limit - Pin 2 UA (→ **Mapping: Current Limit - Pin 2 UA** (→ S. 99))
64...65	Port X4: Current Limit - Pin 2 UA (→ **Mapping: Current Limit - Pin 2 UA** (→ S. 99))
66...67	Port X1: Current Limit - Pin 1 + Pin 4 US (→ **Mapping: Current Limit - Pin 1 + Pin 4 US** (→ S. 99))
68...69	Port X2: Current Limit - Pin 1 + Pin 4 US (→ **Mapping: Current Limit - Pin 1 + Pin 4 US** (→ S. 99))
70...71	Port X3: Current Limit - Pin 1 + Pin 4 US (→ **Mapping: Current Limit - Pin 1 + Pin 4 US** (→ S. 99))
72...73	Port X4: Current Limit - Pin 1 + Pin 4 US (→ **Mapping: Current Limit - Pin 1 + Pin 4 US** (→ S. 99))

Legend:

▪ [Communication Profile]	The access rights to the parameter data, process data and the events/diagnostic messages of the IO-Link master as well as the connected IO-Link devices.	1 byte	0x00	EtherNet/IP + IoT
			0x01	EtherNet/IP + IoT (read-only)
			0x02	EtherNet/IP only
			0x03	Keep current Communication Profile
▪ [Port Process Data Size]	Length of the process input data and process output data per IO-Link port	1 byte	0x00	2 bytes input / 2 bytes output data
			0x01	4 bytes input / 4 bytes output data
			0x02	8 bytes input / 8 bytes output data

				0x03	16 bytes input / 16 bytes output data
				0x04	32 bytes input / 32 bytes output data
▪	[Mode Pin 2 UA]	Operating mode of the supply voltage UA of the IO-Link port	1 byte	0x00	Off: IO-Link Type A Supply
				0x01	On: IO-Link Type B Supply
				0x02	Digital Output
▪	[Failsafe DO Mode Pin 2 UA]	Fail-safe mode for output data of the port if the EtherNet/IP connection is interrupted (only valid for operating mode: Digital Output)	1 byte	0x00	Failsafe Reset Value
				0x01	Failsafe Old Value
				0x02	Failsafe Set Value

Mapping: Port configuration

34394

Byte (offset)	Contents
n	Port Mode
n+1	Port Cycle Time
n+2	Swap
n+3	Validation / Data Storage
n+4	Vendor ID (LSB)
n+5	Vendor ID (MSB)
n+6	Device ID (LSB)
n+7	Device ID
n+8	Device ID (MSB)
n+9	reserved
n+10	Failsafe Mode -- Pin 4 (IO-Link)
n+10	Failsafe Mode -- Pin 4 (DO)

Legend:

▪ [Port Mode]	Operating mode of the port	1 byte	0x00	Disabled
			0x01	Digital Input (Pin 4)
			0x02	Digital Output (Pin 4)
			0x03	IO-Link (Pin 4)
▪ [Port Cycle Time]	Cycle time of the data transmission between the IO-Link master and the IO-Link device	1 byte	0x00	As fast as possible
			0x01	2 ms
			0x02	4 ms
			0x03	8 ms
			0x04	16 ms
			0x05	32 ms
			0x06	64 ms
			0x07	128 ms
▪ [Swap]	Visualisation of the process data (EtherNet/IP uses Little Endian Format (Intel), IO-Link uses Big Endian Format (Motorola))	1 byte	0x00	Disabled
			0x01	Enabled
▪ [Validation / Data Storage]	Supported IO-Link standard and behaviour of the IO-Link master if new IO-Link devices are connected to the port (only valid for Port Mode: IO-Link)	1 byte	0x00	No device check and clear
			0x01	Type compatible V1.0 Device
			0x02	Type compatible V1.1 Device
			0x03	Type compatible V1.1 Device with Backup + Restore
			0x04	Type compatible V1.1 Device with Backup

▪ [Vendor ID]	Vendor ID of the manufacturer of the device at the port (only valid for Port Mode: IO-Link) Vendor ID = 0x1234 ▪ Vendor ID (MSB) = 0x12 ▪ Vendor ID (LSB) = 0x34	2 bytes	0x0000...0xFFFF	
▪ [Device ID]	Device ID of the device at the port (only valid for Port Mode: IO-Link) Device ID = 0x123456 ▪ Device ID (MSB) = 0x12 ▪ Device ID = 0x34 ▪ Device ID (LSB) = 0x56	3 bytes	0x000000...0xFFFFFF	
▪ [Failsafe Mode -- Pin 4 (IO-Link)]	Fail-safe mode for output data of the port if the EtherNet/IP connection is interrupted (only valid for port mode: IO-Link)	1 byte	0x00 0x01 0x02 0x03	No Failsafe Failsafe Reset Value Failsafe Old Value Failsafe with Pattern
▪ [Failsafe Mode -- Pin 4 (DO)]	Fail-safe value for output data of the port if the EtherNet/IP connection is interrupted (only valid for port mode: Digital Output (DO)	1 byte	0x00 0x01 0x02	Failsafe Reset Value Failsafe Old Value Failsafe Set Value

Mapping: Current Limit - Pin 2 UA

42753

Byte (offset)	Contents
n	Current Limit - Pin 2 UA (LSB)
n+1	Current Limit - Pin 2 UA (MSB)

Legend:

▪ [Current Limit - Pin 2 UA]	Current limitation of the supply voltage UA of the port (value in mA) Current Limit - Pin 2 UA = 0x1234 ▪ Current Limit - Pin 2 UA (MSB) = 0x12 ▪ Current Limit - Pin 2 UA (LSB) = 0x34	2 bytes	0x0000 ... **0x012C** ... 0x07D0	0 mA ... **300 mA** ... 2000 mA

Mapping: Current Limit - Pin 1 + Pin 4 US

42784

Byte (offset)	Contents
n	Current Limit - Pin 1 + Pin 4 US (LSB)
n+1	Current Limit - Pin 1 + Pin 4 US (MSB)

Legend:

▪ [Port Xm: Current Limit - Pin 1 + Pin 4 US]	Current limitation of the supply voltage US on the IO-Link port (value in mA) Current Limit - Pin 1 + Pin 4 US = 0x1234 ▪ Current Limit - Pin 1 + Pin 4 US (MSB) = 0x12 ▪ Current Limit - Pin 1 + Pin 4 US (LSB) = 0x34	2 bytes	0x0000 ... **0x012C** ... 0x07D0	0 mA ... **300 mA** ... 2000 mA

14.2.3 Cyclic data

Content

Input Assembly (Instance 100): I/O + Acyclic + Diag 100
Input Assembly (Instance 102): I/O 102
Input Assembly (Instance 103): I/O + Acyclic + Diag + EnMo 103
Output Assembly (Instance 150): I/O + Acyclic 110
Output Assembly (Instance 151): I/O 111

33814

Input Assembly (Instance 100): I/O + Acyclic + Diag

57647

Byte (offset)	Contents							
	Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	reserved	reserved	reserved	reserved	Port X4: Pin 4	Port X3: Pin 4	Port X2: Pin 4	Port X1: Pin 4
1	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
2	reserved	reserved	reserved	reserved	Port X4: Short / OL	Port X3: Short / OL	Port X2: Short / OL	Port X1: Short / OL
3	reserved	reserved	reserved	reserved	reserved	reserved	Status Sensor Supply UA	Status Sensor Supply US
4...45	Acyclic Command Channel: Response (→ **Response channel** (→ S. 114))							
46...47	Port X1: Port Qualifier Information (→ **Mapping: Port Qualifier Information (PQI)** (→ S. 105))							
48...63	Port X1: Vendor ID, Device ID, Events (→ **Mapping: IO-Link device information + events** (→ S. 106))							
64...65	Port X2: Port Qualifier Information (→ **Mapping: Port Qualifier Information (PQI)** (→ S. 105))							
66...81	Port X2: Vendor ID, Device ID, Events (→ **Mapping: IO-Link device information + events** (→ S. 106))							
82...83	Port X3: Port Qualifier Information (→ **Mapping: Port Qualifier Information (PQI)** (→ S. 105))							
84...99	Port X3: Vendor ID, Device ID, Events (→ **Mapping: IO-Link device information + events** (→ S. 106))							
100...101	Port X4: Port Qualifier Information (→ **Mapping: Port Qualifier Information (PQI)** (→ S. 105))							
102...117	Port X4: Vendor ID, Device ID, Events (→ **Mapping: IO-Link device information + events** (→ S. 106))							
118	Port X1: Input Data IO-Link (n Bytes)							
118+n	Port X2: Input Data IO-Link (n Bytes)							
118+2n	Port X3: Input Data IO-Link (n Bytes)							
118+3n	Port X4: Input Data IO-Link (n Bytes)							

Legend:

▪ [Pin 4]	Signal level on pin 4 of the port (port mode: DI)	1 bit	0x0	LOW
			0x1	HIGH
▪ [Short / OL]	Occurrence of a short circuit or of an overvoltage on the IO-Link port	1 bit	0x0	no error
			0x1	Short circuit or overvoltage detected
▪ [Status Sensor Supply US]	Status of the supply voltage US	1 bit	0x0	no error
			0x1	Overvoltage on US (> 3,6 A)
▪ [Status Sensor Supply UA]	Status of the supply voltage UA	1 bit	0x0	no error
			0x1	no UA connected

▪	[Input Data IO-Link (n Bytes)]	IO-Link process data on pin 4 of the port (port mode: IO-Link) n = [2, 4, 8, 16, 32]; is determined by the parameter [Port Process Data Size] (→ **Configuration Assembly (Instance 199)** (→ S. 95))	n bytes	Pro byte: 0x00...0xFF

Input Assembly (Instance 102): I/O

57648

Byte (offset)	Contents							
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	reserved	reserved	reserved	reserved	Port X4: Pin 4	Port X3: Pin 4	Port X2: Pin 4	Port X1: Pin 4
1	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
2	reserved	reserved	reserved	reserved	Port X4: Short / OL	Port X3: Short / OL	Port X2: Short / OL	Port X1: Short / OL
3	reserved	reserved	reserved	reserved	reserved	reserved	Status Sensor Supply UA	Status Sensor Supply US
4...5	Port X1: Port Qualifier Information (→ **Mapping: Port Qualifier Information (PQI)** (→ S. 105))							
6...7	Port X2: Port Qualifier Information (→ **Mapping: Port Qualifier Information (PQI)** (→ S. 105))							
8...9	Port X3: Port Qualifier Information (→ **Mapping: Port Qualifier Information (PQI)** (→ S. 105))							
10...11	Port X4: Port Qualifier Information (→ **Mapping: Port Qualifier Information (PQI)** (→ S. 105))							
12	Port X1: Input Data IO-Link (n Bytes)							
12+n	Port X2: Input Data IO-Link (n Bytes)							
12+2n	Port X3: Input Data IO-Link (n Bytes)							
12+3n	Port X4: Input Data IO-Link (n Bytes)							

Legend:

▪ [Pin 4]	Signal level on pin 4 of the port (port mode: DI)	1 bit	0x0	LOW
			0x1	HIGH
▪ [Short / OL]	Occurrence of a short circuit or of an overvoltage on the IO-Link port	1 bit	0x0	no error
			0x1	Short circuit or overvoltage detected
▪ [Status Sensor Supply US]	Status of the supply voltage US	1 bit	0x0	no error
			0x1	Overvoltage on US (> 3,6 A)
▪ [Status Sensor Supply UA]	Status of the supply voltage UA	1 bit	0x0	no error
			0x1	no UA connected
▪ [Input Data IO-Link (n Bytes)]	IO-Link process data on pin 4 of the port (port mode: IO-Link) n = [2, 4, 8, 16, 32]; is determined by the parameter [Port Process Data Size] (→ **Configuration Assembly (Instance 199)** (→ S. 95))	n bytes	Pro byte: 0x00...0xFF	

Input Assembly (Instance 103): I/O + Acyclic + Diag + EnMo

57649

Byte (offset)	Contents							
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	reserved	reserved	reserved	reserved	Port X4: Pin 4	Port X3: Pin 4	Port X2: Pin 4	Port X1: Pin 4
1	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
2	reserved	reserved	reserved	reserved	Port X4: Short / OL	Port X3: Short / OL	Port X2: Short / OL	Port X1: Short / OL
3	reserved	reserved	reserved	reserved	reserved	reserved	Status Sensor Supply UA	Status Sensor Supply US
4...45	Acyclic Command Area: Response Channel (→ **Response channel** (→ S. 114))							
46...47	Port X1: Port Qualifier Information (→ **Mapping: Port Qualifier Information (PQI)** (→ S. 105))							
48...63	Port X1: Vendor-ID, Device ID, Events (→ **Mapping: IO-Link device information + events** (→ S. 106))							
64...65	Port X2: Port Qualifier Information (→ **Mapping: Port Qualifier Information (PQI)** (→ S. 105))							
66...81	Port X2: Vendor-ID, Device ID, Events (→ **Mapping: IO-Link device information + events** (→ S. 106))							
82...83	Port X3: Port Qualifier Information (→ **Mapping: Port Qualifier Information (PQI)** (→ S. 105))							
84...99	Port X3: Vendor-ID, Device ID, Events (→ **Mapping: IO-Link device information + events** (→ S. 106))							
100...101	Port X4: Port Qualifier Information (→ **Mapping: Port Qualifier Information (PQI)** (→ S. 105))							
102...117	Port X4: Vendor-ID, Device ID, Events (→ **Mapping: IO-Link device information + events** (→ S. 106))							
118...121	Voltage + Current Actuator Supply UA (→ **Mapping: Voltage + Current Actuator Supply UA** (→ S. 108))							
122...125	Voltage + Current Sensor Supply US (→ **Mapping: Voltage + Current Sensor Supply US** (→ S. 108))							
126...127	Port X1: Current Pin 2 (Actuator UA) (→ **Mapping: Current Pin 2 (Actuator UA)** (→ S. 109))							
128...129	Port X2: Current Pin 2 (Actuator UA) (→ **Mapping: Current Pin 2 (Actuator UA)** (→ S. 109))							
130...131	Port X3: Current Pin 2 (Actuator UA) (→ **Mapping: Current Pin 2 (Actuator UA)** (→ S. 109))							
132...133	Port X4: Current Pin 2 (Actuator UA) (→ **Mapping: Current Pin 2 (Actuator UA)** (→ S. 109))							
134...135	Port X1: Current Pin 1 + Pin 4 (Sensor US) (→ **Mapping: Current Pin 1 + Pin 4 (Sensor US)** (→ S. 109))							
136...137	Port X2: Current Pin 1 + Pin 4 (Sensor US) (→ **Mapping: Current Pin 1 + Pin 4 (Sensor US)** (→ S. 109))							
138...139	Port X3: Current Pin 1 + Pin 4 (Sensor US) (→ **Mapping: Current Pin 1 + Pin 4 (Sensor US)** (→ S. 109))							
140...141	Port X4: Current Pin 1 + Pin 4 (Sensor US) (→ **Mapping: Current Pin 1 + Pin 4 (Sensor US)** (→ S. 109))							
142	Port X1: Input Data IO-Link (n Bytes)							
142+n	Port X2: Input Data IO-Link (n Bytes)							
142+2n	Port X3: Input Data IO-Link (n Bytes)							
142+3n	Port X4: Input Data IO-Link (n Bytes)							

Legend:

▪ [Pin 4]	Signal level on pin 4 of the port (port mode: DI)	1 bit	0x0	LOW
			0x1	HIGH
▪ [Short / OL]	Occurrence of a short circuit or of an overvoltage on the IO-Link port	1 bit	0x0	no error
			0x1	Short circuit or overvoltage detected
▪ [Status Sensor Supply US]	Status of the supply voltage US	1 bit	0x0	no error
			0x1	Overvoltage on US (> 3,6 A)

▪ [Status Sensor Supply UA]	Status of the supply voltage UA	1 bit	0x0	no error
			0x1	no UA connected
▪ [Input Data IO-Link (n Bytes)]	IO-Link process data on pin 4 of the port (port mode: IO-Link) n = [2, 4, 8, 16, 32]; is determined by the parameter [Port Process Data Size] (→ **Configuration Assembly (Instance 199)** (→ S. 95))	n bytes	Pro byte: 0x00...0xFF	

Mapping: Port Qualifier Information (PQI)

57650

Byte (offset)	Contents							
	Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
n	Diagnosis present	Wrong Length PD OUT	Wrong Length PD IN	Wrong Cycle Time	Wrong VID / DID	Invalid Data	Dev Not Conn	IOL Mode
n+1	res.	res.	res.	res.	res.	res.	Status Sensor Supply US	Status Actuator Supply UA

Legend:

▪ [IOL Mode]	Operating mode of the IO-Link port	1 bit	0x0	Other
			0x1	IO-Link
▪ [Dev Not Conn]	Connection between IO-Link device and IO-Link port	1 bit	0x0	connected
			0x1	not connected
▪ [Invalid Data]	Status of the process input data on the IO-Link port	1 bit	0x0	valid data
			0x1	invalid data
▪ [Wrong VID/DID]	Verification whether the current and configured vendor ID and device ID are identical	1 bit	0x0	OK
			0x1	No match
▪ [Wrong Cycle Time]	Verification whether the current and configured cycle time are identical	1 bit	0x0	OK
			0x1	No match
▪ [Wrong PD Input Length]	Verification whether the size of the received input data are identical with the configured size	1 bit	0x0	OK
			0x1	Configured size too small
▪ [Wrong PD Output Length]	Verification whether the size of the sent output data is identical with the size expected by the IO-Link device	1 bit	0x0	OK
			0x1	Configured size too small
▪ [Diagnosis present]	Indication of a new event (Coming Event, Single Shot Event). ▪ Coming Events disappear when the corresponding Disappearing Event occurs ▪ Single Shot Events disappear automatically	1 bit	0x0	no event
			0x1	New event is there
▪ [Status Actuator Supply UA]	Status of the supply voltage UA on pin 2 of the IO-Link port	1 bit	0x0	no error
			0x1	no UA connected
▪ [Status Sensor Supply US]	Status of the supply voltage US on pin 1 and pin 4 of the IO-Link port	1 bit	0x0	no error
			0x1	US is not connected or current too low (< 3.6 A)

Mapping: IO-Link device information + events

34413

Byte (offset)	Contents							
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n	VID (LSB)							
n+1	VID (MSB)							
n+2	DID (LSB)							
n+3	DID							
n+4	DID (MSB)							
n+5	reserved							
n+6	Event 1: Mode		Event 1: Type		Event 1: Src	Event 1: Instance		
n+7	Event 1: Code (LSB)							
n+8	Event 1: Code (MSB)							
n+9	Event 2: Mode		Event 2: Type		Event 2: Src	Event 2: Instance		
n+10	Event 2: Code (LSB)							
n+11	Event 2: Code (MSB)							
n+12	Event 3: Mode		Event 3: Type		Event 3: Src	Event 3: Instance		
n+13	Event 3: Code (LSB)							
n+14	Event 3: Code (MSB)							
n+15	reserved							

Legend:

- [VID] Vendor ID of the connected IO-Link device
  VID = 0x1234
  - DID (MSB) = 0x12
  - DID (LSB) = 0x34

  2 bytes — 0x0000...0xFFFF
- [DID] Device ID of the connected IO-Link device
  DID = 0x123456
  - DID (MSB) = 0x12
  - DID = 0x34
  - DID (LSB) = 0x56

  3 bytes — 0x000000...0xFFFFFF
- [Event m: Mode] Mode: Mode of the event — 2 bits
  - 0x0 reserved
  - 0x1 One-time event
  - 0x2 Event has disappeared
  - 0x3 Event has appeared
- [Event m: Type] Type: category of the event — 2 bits
  - 0x0 reserved
  - 0x1 Notification
  - 0x2 Warning
  - 0x3 Error
- [Event m: Src] Source: Source of the event — 1 bit
  - 0x0 IO-Link Device
  - 0x1 IO-Link master
- [Event m: Instance] Instance: Trigger of the event — 3 bits
  - 0x0 Unknown
  - 0x1 ... 0x3 reserved
  - 0x4 Application
  - 0x5 ... 0x7 reserved

▪ [Event m: Code]	Code: Event code Code = 0x1234 ▪ Code (MSB) = 0x12 ▪ Code (LSB) = 0x34	2 bytes	depends on the device (→ IODD description of the IO-Link device)

Mapping: Voltage + Current Actuator Supply UA

57651

Byte (offset)	Contents
n	Voltage Actuator Supply UA (LSB)
n+1	Voltage Actuator Supply UA (MSB)
n+2	Current Actuator Supply UA (LSB)
n+3	Current Actuator Supply UA (MSB)

Legend:

▪ [Voltage Actuator Supply UA]	voltage value of the supply voltage UA (value in mV) Voltage Actuator Supply UA = 0x1234 ▪ Voltage Actuator Supply UA (MSB) = 0x12 ▪ Voltage Actuator Supply UA (LSB) = 0x34	2 bytes	0x4E20 ... 0x7530	20000 mV 30000 mV
▪ [Current Actuator Supply UA]	Current intensity value of the supply voltage UA (value in mA) Current Actuator Supply UA = 0x1234 ▪ Current Actuator Supply UA (MSB) = 0x12 ▪ Current Actuator Supply UA (LSB) = 0x34	2 bytes	0x0000 ... 0x1F40	0 mA 8000 mA

Mapping: Voltage + Current Sensor Supply US

57652

Byte (offset)	Contents
n	Voltage Sensor Supply US (LSB)
n+1	Voltage Sensor Supply US (MSB)
n+2	Current Sensor Supply US (LSB)
n+3	Current Sensor Supply US (MSB)

Legend:

▪ [Voltage Sensor Supply US]	Voltage value of the sensor supply US (value in mV) Voltage Sensor Supply US = 0x1234 ▪ Voltage Sensor Supply US (MSB) = 0x12 ▪ Voltage Sensor Supply US (LSB) = 0x34	2 bytes	0x4E20 ... 0x7530	20000 mV 30000 mV
▪ [Current Sensor Supply US]	Current value of the sensor supply US (value in mA) Current Sensor Supply US = 0x1234 ▪ Current Sensor Supply US (MSB) = 0x12 ▪ Current Sensor Supply US (LSB) = 0x34	2 bytes	0x0000 ... 0x0E10	0 mA 3600 mA

Mapping: Current Pin 2 (Actuator UA)

57653

Byte (offset)	Contents
n	Current Pin 2 (Actuator UA) (LSB)
n+1	Current Pin 2 (Actuator UA) (MSB)

Legend:

▪ [Current Pin 2 (Actuator UA)]	Current value of the supply voltage UA on pin 2 of the IO-Link port (value in mA) Current Pin 2 (Actuator UA) = 0x1234 ▪ Current Pin 2 (Actuator UA) (MSB) = 0x12 ▪ Current Pin 2 (Actuator UA) (LSB) = 0x34	2 bytes	0x0000 ... 0x07D0	0 mA 2000 mA

Mapping: Current Pin 1 + Pin 4 (Sensor US)

57654

Byte (offset)	Contents
n	Current Pin 1 + Pin 4 (Sensor US) (LSB)
n+1	Current Pin 1 + Pin 4 (Sensor US) (MSB)

Legend:

▪ [Current Pin 1 + Pin 4 (Sensor US)]	Current value of the supply voltage US on pin 1 and pin 4 and the IO-Link port (value in mA) Current Pin 1 + Pin 4 (Sensor US) = 0x1234 ▪ Current Pin 1 + Pin 4 (Sensor US) (MSB) = 0x12 ▪ Current Pin 1 + Pin 4 (Sensor US) (LSB) = 0x34	2 bytes	0x0000 ... 0x07D0	0 mA 2000 mA

Output Assembly (Instance 150): I/O + Acyclic

57655

Byte (offset)	Contents							
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	reserved	reserved	reserved	reserved	Port X4: Pin 4	Port X3: Pin 4	Port X2: Pin 4	Port X1: Pin 4
1	reserved	reserved	reserved	reserved	Port X4: Pin 2	Port X3: Pin 2	Port X2: Pin 2	Port X1: Pin 2
2	reserved							
3	reserved							
4...45	Acyclic Command Area: Request Channel (→ **Request channel** (→ S. 113))							
46	Port X1: Output Data IO-Link (n Bytes)							
46+n	Port X2: Output Data IO-Link (n Bytes)							
46+2n	Port X3: Output Data IO-Link (n Bytes)							
46+3n	Port X4: Output Data IO-Link (n Bytes)							

Legend:

▪ [Pin 4]	Signal level on pin 4 of the port (port mode: DO)	1 bit	0x0	LOW
			0x1	HIGH
▪ [Pin 2]	Signal level on pin 2 of the port	1 bit	0x0	LOW
			0x1	HIGH
▪ [Output Data IO-Link (n Bytes)]	IO-Link process data on pin 4 of the port (port mode: IO-Link) n = [2, 4, 8, 16, 32]; is determined by the parameter [Port Process Data Size] (→ **Configuration Assembly (Instance 199)** (→ S. 95))	n bytes	Pro byte: 0x00...0xFF	

Output Assembly (Instance 151): I/O

57656

Byte (offset)	Contents							
	Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	reserved	reserved	reserved	reserved	Port X4: Pin 4	Port X3: Pin 4	Port X2: Pin 4	Port X1: Pin 4
1	reserved	reserved	reserved	reserved	Port X4: Pin 2	Port X3: Pin 2	Port X2: Pin 2	Port X1: Pin 2
2	Port X1: Output Data IO-Link (n Bytes)							
2+n	Port X2: Output Data IO-Link (n Bytes)							
2+2n	Port X3: Output Data IO-Link (n Bytes)							
2+3n	Port X4: Output Data IO-Link (n Bytes)							

Legend:

▪ [Pin 4]	Signal level on pin 4 of the port (port mode: DO)	1 bit	0x0	LOW
			0x1	HIGH
▪ [Pin 2]	Signal level on pin 2 of the port	1 bit	0x0	LOW
			0x1	HIGH
▪ [Output Data IO-Link (n Bytes)]	IO-Link process data on pin 4 of the port (port mode: IO-Link) n = [2, 4, 8, 16, 32]; is determined by the parameter [Port Process Data Size] (→ **Configuration Assembly (Instance 199)** (→ S. 95))	n bytes	Pro byte: 0x00...0xFF	

14.2.4 Acyclic data

Content

Acyclic command channel 112
Acyclic commands 117

33868

Acyclic command channel

34325

In the cyclic process data, command channels for the transmission of acyclic data is available.

Object	Contents	Bytes	Access
Output assembly	Request channel (field bus PLC >>> IO-Link master) → **Request channel** (→ S. 113)	4...45	r/w
Input assembly	Response channel (IO-Link master >>> fieldbus PLC) → **Response channel** (→ S. 114)	4...45	r

Legend:

r = only read access rights
r/w = read and write access rights

Request channel

34398

Byte	Content							
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
4	Port No. (LSB)							
5	Port No. (MSB)							
6	Index (LSB)							
7	Index (MSB)							
8	Sub-index (LSB)							
9	Sub-index (MSB)							
10	Trigger	Command id						
11	Length of the user data (number of bytes)							
12	Data (byte 0)							
13	Data (byte 1)							
..	...							
43	Data (byte 31)							
44	reserved							
45	reserved							

Legend:

▪ [Port No.]	Number of the IO-Link port Port No. = 0x1234 ▪ Port No. (MSB) = 0x12 ▪ Port No. (LSB) = 0x34	1 Word	0x0001 0x0002 ... 0x0004	Port X01 Port X02 Port X04
▪ [Index]	Index of the IO-Link object Index = 0x1234 ▪ Index (MSB) = 0x12 ▪ Index (LSB) = 0x34	1 Word	0x0000...0xFFFF	
▪ [Subindex]	Subindex of the IO-Link object Subindex = 0x1234 ▪ Subindex (MSB) = 0x12 ▪ Subindex (LSB) = 0x34	1 Word	0x0000...0xFFFF	
▪ [Trigger]	Control of the command execution	1 Bit	0x0 0x1	do not process command execute command
▪ [Command ID]	Command number	7 Bit	0x01 0x02	read write
▪ [Length of user data (number of bytes)]	Number of bytes that contain relevant user data	1 Byte	0x00 ... 0x20	0 bytes 32 bytes
▪ [Data (byte n)]	User data	1 Byte	per byte: 0x00...0xFF	

Response channel

34401

Byte	Content							
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
4	Port No. (LSB)							
5	Port No. (MSB)							
6	Index (LSB)							
7	Index (MSB)							
8	Sub-index (LSB)							
9	Sub-index (MSB)							
10	Handshake	Command ID						
11	Result							
12	Length of response data (number of bytes)							
13	Data (byte 0) or Error Code							
14	Data (byte 1) or Additional Code							
...	...							
44	Data (byte 31)							
45	reserved							

Legend:

▪ [Port No.]	Number of the IO-Link port Port No. = 0x1234 ▪ Port No. (MSB) = 0x12 ▪ Port No. (LSB) = 0x34	1 word	0x0001 0x0002 ... 0x0004	Port X01 Port X02 Port X04
▪ [Index]	Index of the IO-Link object Index = 0x1234 ▪ Index (MSB) = 0x12 ▪ Index (LSB) = 0x34	1 word	0x0000...0xFFFF	
▪ [Subindex]	Subindex of the IO-Link object Subindex = 0x1234 ▪ Subindex (MSB) = 0x12 ▪ Subindex (LSB) = 0x34	1 word	0x0000...0xFFFF	
▪ [Handshake]	Validity of the IO-Link response data	1 bit	0x0 0x1	Data invalid Data valid
▪ [Command ID]	Command number	7 bits	0x01 0x02	Read Write
▪ [Result]	Status of the command processing	8 bits	0x00 0x0F 0xFF	OK OK, data read >32 bytes Error occurred
▪ [Length of response data (number of bytes)]	Number of bytes that contain relevant user data	1 byte	0x00 ... 0x20	0 bytes 32 bytes

▪ [Data (byte 0) or Error Code]	User data (byte 0) or error codes	1 byte	User data: 0x00...0xFF Error Code: → **Error codes** (→ S. 116)
▪ [Data (byte 1) or Additional Code]	User data (byte1) or additional error codes	1 byte	User data: 0x00...0xFF Additional Code: → **Additional Codes** (→ S. 116)
▪ [Data (byte n)]	User data (byte n)	1 byte	0x00...0xFF

Error codes

34342

Error code	Description
0x71	Service not available (unknown command has been sent to the IO-Link port)
0x72	Port blocked (another cyclic process accesses the IO-Link port)
0x73	Forbidden (access rights don't allow command processing)
0x74	Invalid data (wrong parameter has been sent in the command)
0x76	Wrong port (wrong port number)
0x77	Wrong port function (wrong port function or wrong parameter has been sent to the device)
0x78	Invalid length (set length is > 0x20)
0x80	Error in the device application; observe additional code (→ **Additional Codes** (→ S. 116))

Additional Codes

54584

Code	Name	Description
0x00	APP_DEV	Device application error - no details
0x11	IDX_NOTAVAIL	Index not available
0x12	SUBIDX_NOTAVAIL	Subindex not available
0x20	SERV_NOTAVAIL	Service temporarily not available
0x21	SERV_NOTAVAIL_LOCCTRL	Service temporarily not available - local control
0x22	SERV_NOTAVAIL_DEVCTRL	Service temporarily not available - device control
0x23	IDX_NOT_WRITEABLE	Access denied
0x30	PAR_VALOUTOFRNG	Parameter value out of range
0x31	PAR_VALGTLIM	Parameter value above limit
0x32	PAR_VALLTLIM	Parameter value below limit
0x33	VAL_LENOVRRUN	Parameter length overrun
0x34	VAL_LENUNDRUN	Parameter length underrun
0x35	FUNC_NOTAVAIL	Function not available
0x36	FUNC_UNAVAILTEMP	Function temporarily not available
0x40	PAR_SETINVALID	Invalid parameter set
0x41	PAR_SETINCONSIST	Inconsistent parameter set
0x82	APP_DEVNOTRDY	Application not ready

Additional Codes are only available, if Error Code = 0x80 (→ **Error codes** (→ S. 116))

Acyclic commands

Content

Command 0x10 – Set mode....118
Command 0x20 – Set validation ID / data storage....120
Command 0x30 – Set fail-safe data pattern....122

34331

Command 0x10 – Set mode

34322

The command changes the operating mode of an IO-Link port of the AL1421.

Corresponding parameter: [Port Mode] (→ **Mapping: Port configuration** (→ S. 97))

Command request

34314

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
4	Port No. (LSB)							
5	Port No. (MSB)							
6	reserved							
7	reserved							
8	reserved							
9	reserved							
10	Trigger	0x10						
11	Target Mode							
12...45	reserved							

Legend:

▪ [Port No.]	Number of the IO-Link port Port No. = 0x1234 ▪ Port No. (MSB) = 0x12 ▪ Port No. (LSB) = 0x34	1 word	0x0001 0x0002 ... 0x0004	Port X01 Port X02 Port X04
▪ [Trigger]	Control of the command execution	1 bit	0x0 0x1	do not process command execute command
▪ [Target Mode]	Operating type of the IO-Link port	1 byte	0x00 0x01 0x02 0x03	deactivated operation as digital input (DI) operation as digital output (DO) operation as IO-Link intervals

Command response

34316

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
4	Port No. (LSB)							
5	Port No. (MSB)							
6	reserved							
7	reserved							
8	reserved							
9	reserved							
10	Handshake	0x10						
11	Result							
12	Target Mode							
13...45	reserved							

Legend:

▪ [Port No.]	Number of the IO-Link port Port No. = 0x1234 ▪ Port No. (MSB) = 0x12 ▪ Port No. (LSB) = 0x34	1 word	0x0001 0x0002 ... 0x0004	Port X01 Port X02 Port X04
▪ [Handshake]	Status of the execution of the command	1 bit	0x0 0x1	command is executed execution of the command was successful
▪ [Result]	Error indication	1 byte	0x00 0x01	no error error occured
▪ [Target Mode]	Operating type of the IO-Link port	1 byte	0x00 0x01 0x02 0x03	deactivated operation as digital input (DI) operation as digital output (DO) operation as IO-Link intervals

Command 0x20 – Set validation ID / data storage

34321

The command sets the behaviour of the IO-Link master when connecting a new IO-Link device to an IO-Linkport of the device.

Corresponding parameter: [Validation ID] (→ **Mapping: Port configuration** (→ S. 97))

Command request

34315

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
4	Port No. (LSB)							
5	Port No. (MSB)							
6	reserved							
7	reserved							
8	reserved							
9	reserved							
10	Trigger	0x20						
11	Validation ID							
12...42	reserved							

Legend:

▪ [Port No.]	Number of the IO-Link port Port No. = 0x1234 ▪ Port No. (MSB) = 0x12 ▪ Port No. (LSB) = 0x34	1 word	0x0001 0x0002 ... 0x0004	Port X01 Port X02 Port X04
▪ [Trigger]	Control command execution	1 Bit	0x0 0x1	do not process command execute command
▪ [Validation ID]	Behaviour of the IO-Link master when connecting an IO-Link device to the IO-Link port	1 byte	0x00 0x01 0x02 0x03 0x04	No check Type compatible V1.0 Device Type compatible V1.1 Device Type compatible V1.1 Device with Backup + Restore Type compatible V1.1 Device with Restore

Command response

34362

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
4	Port No. (LSB)							
5	Port No. (MSB)							
6	reserved							
7	reserved							
8	reserved							
9	reserved							
10	Handshake	0x20						
11	Result							
12	Validation ID							
13..45	reserved							

Legend:

▪ [Port No.]	Number of the IO-Link port Port No. = 0x1234 ▪ Port No. (MSB) = 0x12 ▪ Port No. (LSB) = 0x34	1 word	0x0001 0x0002 ... 0x0004	Port X01 Port X02 Port X04
▪ [Handshake]	Status of the execution of the command	1 bit	0x0 0x1	command is executed execution of the command was successful
▪ [Result]	Error indication	1 byte	0x00 0x01	no error errors occurred
▪ [Validation ID]	Behaviour of the IO-Link master when connecting an IO-Link device to the IO-Link port	1 byte	0x00 0x01 0x02 0x03 0x04	No check Type compatible V1.0 Device Type compatible V1.1 Device Type compatible V1.1 Device with Backup + Restore Type compatible V1.1 Device with Restore

Command 0x30 – Set fail-safe data pattern

34379

The command sets the behaviour of the outputs when the EtherNet/IP connection and the corresponding fail-safe values are interrupted.

Corresponding parameter: [Fail-safe Mode] (→ **Mapping: Port configuration** (→ S. 97)
The number of the required fail-safe values results from the size of the output data (→ **Configuration Assembly (Instance 199)** (→ S. 95)).

Command request

34317

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
4	Port No. (LSB)							
5	Port No. (MSB)							
6	reserved							
7	reserved							
8	reserved							
9	reserved							
10	Trigger	0x30						
11	Fail-safe mode							
12	Byte Length N							
13	Fail-safe data (byte 0)							
...	...							
44	Fail-safe data (byte 31)							
45	reserved							

Legend:

▪ [Port No.]	Number of the IO-Link port Port No. = 0x1234 ▪ Port No. (MSB) = 0x12 ▪ Port No. (LSB) = 0x34	1 word	0x0001 0x0002 ... 0x0004	Port X01 Port X02 Port X04
▪ [Trigger]	Control command execution	1 bit	0x0 0x1	do not process command execute command
▪ [Fail-safe Mode]	Behaviour of the outputs when the EtherNet/IP connection is interrupted and setting of the corresponding fail-safe values	1 byte	0x00 0x01 0x02 0x03	No Fail-safe Fail-safe Reset Value Fail-safe Old Value Fail-safe with Pattern
▪ [Byte Length N]	Number of the bytes that contain fail-safe values	1 byte	0x00 ... 0x20	0 Bytes 32 Bytes
▪ [Fail-safe Data (Byte n)]	Fail-Safe value n (n = 0...31)	1 bytet	per byte: 0x00...0xFF	

Command response

34361

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
4	Port No. (LSB)							
5	Port No. (MSB)							
6	reserved							
7	reserved							
8	reserved							
9	reserved							
10	Handshake	0x30						
11	Result							
12	Fail-safe mode							
13...45	reserved							

Legend:

- [Port No.] Number of the IO-Link port
  Port No. = 0x1234
  - Port No. (MSB) = 0x12
  - Port No. (LSB) = 0x34

  1 word

0x0001	Port X01
0x0002	Port X02
...	
0x0004	Port X04

- [Handshake] Status of the execution of the command

  1 bit

0x0	command is executed
0x1	execution of the command was successful

- [Result] Error indication

  1 byte

0x00	no error
0x01	error occured

- [Fail-safe Mode] Behaviour of the outputs when the EtherNet/IP connection is interrupted

  1 byte

0x00	No Fail-safe
0x01	Fail-safe Reset Value
0x02	Fail-safe Old Value
0x03	Fail-safe with Pattern

14.2.5 Field bus objects

Content

CIP class services 124
CIP object classes 125
Identity Object (object class: 0x01) 126
Message Router Object (object class: 0x02) 128
Assembly Object (object class: 0x04) 129
Connection Manager Object (object class: 0x06) 130
Device Level Ring Object (object class: 0x47) 131
Quality of Service (object class: 0x48) 132
IO-Link requests (object class: 0x80) 133
TCP/IP object (object class: 0xF5) 143
Ethernet Link Object (object class: 0xF6) 145

34352

CIP class services

34335

The device supports the following class and instance services:

Class code		Service	Description
dec	hex		
01	01	Get Attribute All	Read all attribute values of the class or instance
02	02	Set Attribute All	Change all attribute values of the class or instance
05	05	Reset	Reset
09	09	Delete	Delete
14	0E	Get Attribute Single	Read single attribute value of the class or instance
16	10	Set Attribute Single	Change single attribute value of the class or instance
75	4B	Read ISDU	Read ISDU
76	4C	Write ISDU	Write ISDU
77	4D	Write Failsafe Pattern	Write failsafe pattern
78	4E	Forward Close	Close connection
84	54	Forward Open	Open new connection

CIP object classes

34334

The device supports the following CIP object classes:

Class code		Object type	Reference
dec	hex		
01	01	Identity Object	→ **Identity Object (object class: 0x01)** (→ S. 126)
02	02	Message Router Object	→ **Message Router Object (object class: 0x02)** (→ S. 128)
04	04	Assembly Object	→ **Assembly Object (object class: 0x04)** (→ S. 129)
06	06	Connection Manager Object	→ **Connection Manager Object (object class: 0x06)** (→ S. 130)
71	47	Device Level Ring Object	→ **Device Level Ring Object (object class: 0x47)** (→ S. 131)
72	48	Quality of Service	→ **Quality of Service (object class: 0x48)** (→ S. 132)
128	80	IO-Link Requests	→ **IO-Link requests (object class: 0x80)** (→ S. 133)
245	F5	TCP/IP Object	→ **TCP/IP object (object class: 0xF5)** (→ S. 143)
246	F6	Ethernet Link Object	→ **Ethernet Link Object (object class: 0xF6)** (→ S. 145)

Identity Object (object class: 0x01)

34340

The Identity Object contains the general information about the device.

Class attributes

34310

Attr. ID	Access	Name	Data type	Description	Value
1	Get	Revision	UINT	Revision of the object	1
2	Get	Max instance	UINT	Max. number of instances of the object	1
6	Get	Maximum ID Number Class Attributes	UINT	ID of the last class attribute	7
7	Get	Maximum ID Number Instance Attributes	UINT	ID of the last instance attribute	9

Instance attributes

34339

Attr. ID	Access	Name	Data type	Description	Preset
1	Get	Vendor ID	UINT	Manufacturer ID	322
2	Get	Device type	UINT	Type of unit	12
3	Get	Product code	UINT	Identification of a particular product of a vendor	1421
4	Get	Revision	STRUCT	Revision of the article that is represented by the Identity Object	1.1
		▪ Major revision	USINT	Main revision (1...127)	1
		▪ Minor revision	USINT	Side revision (3 digits, if necessary with zeros in the beginning)	1
5	Get	Status	WORD	Status of the device	
6	Get	Serial number	UDINT	Serial number of the device	
7	Get	Product Name	SHORT STRING	Readable device designation (max. 32 ASCII characters)	IO-Link Master PFL EIP 4P IP69K
8	Get	State	USINT	Current status of the device (according to status transition diagram)	
				0 Nonexistent	
				1 Device Self Testing	
				2 Standby	
				3 Operational	
				4 Major Recoverable Fault	
				5 Major Unrecoverable Fault	
				6...254 Reserved	
				255 Default for Get_Attributes_All service	
9	Get	Configuration Consistency Value	UINT	The content shows the configuration of the device	0

Supported services

34377

Service code		Name	Class	Attribute	Description
dec	hex				
01	01	Get_Attribute_All	yes	yes	Read all attributes
05	05	Reset	yes	yes	Reset
14	0E	Get_Attribute_Single	yes	yes	Read single attribute
16	10	Set_Attribute_Single	yes	yes	Change single attribute

If an Identity Object receives a reset request, it carries out the following actions:

- It checks if it supports the requested reset type.
- It responds to the request.
- It tries to execute the requested reset type.

Supported reset types:

- 0 Reboot the device (obligatory for all EtherNet/IP devices).
- 1 Restore factory settings and reboot the device.

Message Router Object (object class: 0x02)

34390

The Message Router Object provides an access with which an EtherNet/IP client can address a service to any object class or instance in the physical device.

Class attributes

34320

Attr. ID	Access	Name	Data type	Description	Value
1	Get	Revision	UINT	Revision of the object	1
2	Get	Max instance	UINT	Max. number of instances of the object	1
3	Get	Number of Instances	UINT	Number of instances	1
6	Get	Maximum ID Number Class Attributes	UINT	ID of the last class attribute	7
7	Get	Maximum ID Number Instance Attributes	UINT	ID of the last instance attribute	0

Instance attributes

34402

The object has no instance attributes.

Supported services

34374

Service code		Name	Class	Attribute	Description
dec	hex				
14	0E	Get_Attribute_Single	yes	no	Read single attribute value

Assembly Object (object class: 0x04)

34332

The Assembly Object combines attributes of several objects to allow data to be sent to or received from each object via one connection.

Class attributes

34309

Attr. ID	Access	Name	Data type	Description	Value
1	Get	Revision	UINT	Revision of the object	2
2	Get	Max instance	UINT	Max. number of instances of the object	0x00C7
3	Get	Number of Instances	UINT	Number of instances	3
6	Get	Maximum ID Number Class Attributes	UINT	ID of the last class attribute	7
7	Get	Maximum ID Number Instance Attributes	UINT	ID of the last instance attribute	4

Instances

57658

Attr. ID	Access	Name	Data type	Description
100	Get	Input Assembly	STRUCT	Cyclic input data (→ **Input Assembly (Instance 100): I/O + Acyclic + Diag** (→ S. 100))
102	Get	Input Assembly	STRUCT	Cyclic input data (→ **Input Assembly (Instance 102): I/O** (→ S. 102))
103	Get	Input Assembly	STRUCT	Cyclic input data (→ **Input Assembly (Instance 103): I/O + Acyclic + Diag + EnMo** (→ S. 103))
150	Get, Set	Output Assembly	STRUCT	Cyclic output data (→ **Output Assembly (Instance 150): I/O + Acyclic** (→ S. 110))
151	Get, Set	Output Assembly	STRUCT	Cyclic output data (→ **Output Assembly (Instance 151): I/O** (→ S. 111))
199	Set	Configuration Assembly	STRUCT	Configuration data (→ **Configuration Assembly (Instance 199)** (→ S. 95))

Supported services

34376

Service code		Name	Class	Attribute	Description
dec	hex				
14	0E	Get_Attribute_Single	yes	yes	Read attribute value
16	10	Set_Attribute_Single	no	yes	Change attribute value

Connection Manager Object (object class: 0x06)

34367

The Connection Manager Object structures and manages the internal resources that are used for the connection.

Class attributes

34319

Attr ID	Access	Name	Data type	Description	Value
1	Get	Revision	UINT	Revision of the object	1
2	Get	Max instance	UINT	Max. number of instances of the object	1
3	Get	Number of Instances	UINT	Number of instances	3
6	Get	Maximum ID Number Class Attributes	UINT	ID of the last class attribute	7
7	Get	Maximum ID Number Instance Attributes	UINT	ID of the last instance attribute	0

Instance attributes

34402

The object has no instance attributes.

Supported services

34375

Service code		Name	Class	Attribute	Description
dec	hex				
14	0E	Get_Attribute_Single	yes	yes	Read single attribute
16	10	Set_Attribute_Single	no	yes	Change single attribute
78	4E	Forward_Close	yes	no	Close connection
84	54	Forward_Open	yes	no	Open new connection

Device Level Ring Object (object class: 0x47)

34345

The Device Level Ring (DLR) Object represents the interface for configuration and status information.

Class attributes

34313

Attr ID	Access	Name	Data type	Description	Value
1	Get	Revision	UINT	Revision of the object	3
2	Get	Max instance	UINT	Max. number of instances of the object	1
6	Get	Maximum ID Number Class Attributes	UINT	ID of the last class attribute	7
7	Get	Maximum ID Number Instance Attributes	UINT	ID of the last instance attribute	12

Instance attributes

34327

Attr. ID	Access	Name	Data type	Description		Preset
1	Get	Network Topology	USINT	current network topology		0
2	Get	Network status	USINT	current network status		0
10	Get	Active Supervisor	STRUCT of	Identification of the supervisor		0
			▪ UDINT	IP address of the supervisor		
			▪ ARRAY of 6 USINTs	MAC address of the supervisor		
12	Get	Capability Flags	DWORD	DLR functions of the device		0x82
				Bit 0	Announced-based ring node	0
				Bit 1	Beacon-based ring node	1
				Bit 2...4	reserved	--
				Bit 5	Supervisor capable	0
				Bit 6	Redundant Gateway capable	0
				Bit 7	Flush_Table frame capable	1
				Bit 8..31	reserved	--

I

Supported services

34409

Service code		Name	Class	Attribute	Description
dec	hex				
1	01	Get_Attribute_All	no	yes	Read all attribute values
14	0E	Get_Attribute_Single	yes	yes	Read single attribute value

Quality of Service (object class: 0x48)

34371

Quality of Service (QoS) enables prioritising of Ethernet frames. The priorities of the Ethernet frames can be influenced with the attributes "Differentiate Service Code Points" (DSCP) or "802.1Q Tag".

Class attributes

34307

Attr ID	Access	Name	Data type	Description	Value
1	Get	Revision	UINT	Revision of the object	3
2	Get	Max instance	UINT	Max. number of instances of the object	1
6	Get	Maximum ID Number Class Attributes	UINT	ID of the last class attribute	7
7	Get	Maximum ID Number Instance Attributes	UINT	ID of the last instance attribute	8

Instance attributes

34328

Attr ID	Access	Name	Data type	Description	Value
1	Get	802.1Q tagRevision	USINT	Current network topology	0
2	Get, Set	DSCP PTP Event	USINT	DSCP value for PTP event frames	59
3	Get, Set	DSCP PTP general	USINT	DSCP value for PTP general frames	47
4	Get, Set	DSCP PTP Urgent	USINT	DSCP value for implicit messages with "urgent" priority	55
5	Get, Set	DSCP Scheduled	USINT	DSCP value for implicit messages with "scheduled" priority	47
6	Get, Set	DSCP High	USINT	DSCP value for implicit messages with "high" priority	43
7	Get, Set	DSCP Low	USINT	DSCP value for implicit messages with "low" priority	31
8	Get, Set	DSCP explicit	USINT	DSCP value for explicit messages with "scheduled" priority	27

Supported services

34406

Service code		Name	Class	Attribute	Description
dec	hex				
01	01	Get_Attribute_All	yes	yes	Read all attribute values
14	0E	Get_Attribute_Single	no	yes	Read single attribute value

IO-Link requests (object class: 0x80)

34412

The manufacturer-specific object "IO-Link Requests" enables read and write access to the IO-Link objects of an IO-Link device connected to a AL1421 via ISDU (Index Service Data Unit). The object projects the mechanisms of the CIP addressing on the IO-Link protocol.

Class attributes

34308

Attr ID	Access	Name	Data type	Description	Value
1	Get	Revision	UINT	Revision of the object	4
2	Get	Max instance	UINT	Max. number of instances of the object	2
6	Get	Maximum ID Number Class Attributes	UINT	Number of instances of the object	8

Instance attributes

34399

The required IO-Link port of the device is addressed via the instance attribute.

Supported services

34378

Service code		Name	Class	Attribute	Description
dec	hex				
75	4B	→ **Read_ISDU** (→ S. 134)	no	yes	Read ISDU
76	4C	→ **Write_ISDU** (→ S. 137)	no	yes	Read ISDU
77	4D	→ **Write Failsafe Pattern** (→ S. 140)	no	yes	Write failsafe values of IO-Link port

Read_ISDU

34323

With Read_ISDU, parameters of a connected IO-Link device can be read.

Request

34337

CIP Attribute determines the IO-Link port to which the IO-Link device is connected. The area CIP User Specific Service Data contains the IO-Link index and the IO-Link sub-index of the IO-Link object whose value is to be read:

CIP format	Data type	MSG Config	IO-Link mapping
CIP Class ID	UINT	0x80	IO-Link acyclic access
CIP Instance ID	UINT	0x01	IO-Link master
CIP Attributes	USINT	0x01...0x04	Port number
CIP Service code ID	USINT	0x4B	Request "Read_ISDU"
CIP User specific service data	UINT	0x0000...0xFFFF	IO-Link ISDU object index
	USINT	0x00...0xFF	IO-Link ISDU object sub-index

Response

34326

- **Positive response**

If the service has been executed successfully (CIP Error Code = 0), the read data are returned bit by bit (CIP User Specific Service Data). The answer has the following format:

CIP format	Data type	MSG Config	IO-Link mapping
CIP Class ID	UINT	0x80	IO-Link acyclic access
CIP Instance ID	UINT	0x01	IO-Link master
CIP Attributes	USINT	0x01...0x04	Port number
CIP Service Code ID	USINT	0x4C	Response "Read_ISDU"
CIP Error Code	USINT	0x00	--
CIP Extended Error Code	USINT	0x00	--
CIP User Specific Service Data	USINT	0x00...0xFF	Data (byte 0)
	USINT	0x00...0xFF	Data (byte 1)
	...		...
	USINT	0x00...0xFF	Data (byte n)

The read data is in the IO-Link format. If necessary, the user needs to adapt the byte arrangement of the read data to the CIP format.

- **Negative response**

If an error occurs while executing the service (CIP Error Code <> 0), an extended error code is transmitted. If the CIP Error Code = 0x1E, then the CIP Extended Error Code = 0x00 and the CIP User Specific Service Data area contains the IO-Link Error Code as well as IO-Link Additional Code. The answer has the following format:

CIP format	Data type	MSG Config	IO-Link mapping
CIP Class ID	UINT	0x80	IO-Link acyclic access
CIP Instance ID	UINT	0x01	IO-Link master
CIP Attributes	USINT	0x01...0x04	Port number
CIP Service Code ID	USINT	0x4B	Response "Read_ISDU"
CIP Error Code	USINT	<> 0x00	Error code: see table below
CIP Extended Error Code	USINT	0x00	Extended error code
CIP User Specific Service Data	USINT	<> 0x00	IO-Link Error Code: → **Error codes** (→ S. 116) (only if CIP Error Code = 0x1E)
	USINT	<> 0x00	IO-Link Additional Code: → **Additional Codes** (→ S. 116) (only if CIP Error Code = 0x1E)

CIP Error Code:

Code	Description
0x02	Resource not available: The IO-Link port is busy processing another acyclic service.
0x05	Invalid class ID or instance ID
0x08	Wrong service ID: only service code 0x4B or 0x4C is permitted
0x09	Wrong attribute ID: wrong port number
0x20	Invalid parameter value (e.g. invalid length)
0x1E	Embedded service, error: Error occurred during an IO-Link service. Byte 0 and byte 1 of the User Specific Service Data contain the IO-Link error code and an additional code that are returned by the IO-Link master.

Example: reading the parameter value of anIO-Link device

34350

Task: reading the value of the parameter X of an IO-Link device

- IO-Link device at the port: 0x02
- Parameter X in the object directory of an IO-Link device: Index: 90, sub-index 3

From this, the following results for the configuration of the EtherNet/IP command Message (MSG):

CIP format	Data type	MSG Config	Description
CIP Class ID	UINT	0x80	IO-Link acyclic access
CIP Instance ID	UINT	0x01	IO-Link master
CIP Attributes	USINT	0x02	Port number
CIP Service Code ID	USINT	0x4B	Request "ISDU_Read"
CIP User Specific Service Data	UINT	0x005A	IO-Link ISDU object index
	USINT	0x03	IO-Link ISDU object sub-index

After successful execution of the request, the response area has the following content:

CIP format	Data type	MSG Config	Description
CIP Class ID	UINT	0x80	Object class "IO-Link requests"
CIP Instance ID	UINT	0x01	IO-Link master
CIP Attributes	USINT	0x02	Port number
CIP Service Code ID	USINT	0x4B	Response "ISDU_Read"
CIP Error Code	USINT	0x00	Request processed successfully
CIP Extended Error Code	USINT	0x00	--
CIP User Specific Service Data	USINT	e.g. 0x12	Parameter value that has been read (byte 0)
	USINT	e.g. 0x34	Parameter value that has been read (byte 1)

If an error occurs while the request is executed, the response area has the following content:

CIP format	Data type	MSG Config	Description
CIP Class ID	UINT	0x80	Object class "IO-Link requests"
CIP Instance ID	UINT	0x01	IO-Link master
CIP Attributes	USINT	0x02	Port number
CIP Service code ID	USINT	0x4B	Response "ISDU_Read"
CIP Error code	USINT	0x1E	Error code: Embedded service error
CIP Extended error code	USINT	0x00	--
CIP User specific service data	USINT	e.g. 0x80	IO-Link Error Code: Error in device application
	USINT	e.g. 0x20	IO-Link Additional Code: Service temporarily unavailable

Write_ISDU

34385

With Write_ISDU, the parameters of a connected IO-Link device can be changed.

Request

34387

CIP Attribute determines the IO-Link port to which the IO-Link device is connected. The area CIP User Specific Service Data contains the IO-Link index, the IO-Link sub-index of the IO-Link object whose value is to be changed. It is followed, bit by bit, by the value that is to be assigned to the parameter.

CIP format	Data type	MSG Config	IO-Link mapping
CIP Class ID	UINT	0x80	IO-Link acyclic access
CIP Instance ID	UINT	0x1	IO-Link master
CIP Attribute	USINT	0x01...0x04	Port number
CIP Service Code ID	USINT	0x4C	Request "Write_ISDU"
CIP User Specific Service Data	UINT	0x0000...0xFFFF	IO-Link ISDU object index
	USINT	0x00...0xFF	IO-Link ISDU object sub-index
	USINT	0x00...0xFF	IO-Link ISDU data (byte 0)
	USINT	0x00...0xFF	IO-Link ISDU data (byte 1)

Response

34384

- **Positive response**

If the service has been executed successfully (CIP Error Code = 0), the area CIP User Specific Service Data stays empty The answer has the following format:

CIP format	Data type	MSG Config	IO-Link mapping
CIP Class ID	UINT	0x80	IO-Link acyclic access
CIP Instance ID	UINT	0x01	IO-Link master
CIP Attribute	USINT	0x01...0x04	Port number
CIP Service Code ID	USINT	0x4C	Response "Write_ISDU"
CIP Error Code	USINT	0x00	--
CIP Extended Error Code	USINT	0x00	--

- **Negative response**

If an error occurs while executing the service (CIP Error Code <> 0), an extended error code is transmitted. If the CIP Error Code = 0x1E, then the CIP Extended Error Code = 0x00 and the CIP User Specific Service Data area contains the IO-Link Error Code as well as IO-Link Additional Code. The answer has the following format:

CIP format	Data type	MSG Config	IO-Link mapping
CIP Class ID	UINT	0x80	IO-Link acyclic access
CIP Instance ID	UINT	0x01	IO-Link master
CIP Attributes	USINT	0x01...0x04	Port number
CIP Service Code ID	USINT	0x4C	Response "Write_ISDU"
CIP Error Code	USINT	<> 0x00	Error code. see table below
CIP Extended Error Code	USINT	0x00	Extended error code
CIP User Specific Service Data	USINT	<> 0x00	IO-Link Error Code: → **Error codes** (→ S. 116) (only if CIP Error Code = 0x1E)
	USINT	<> 0x00	IO-Link Additional Code: → **Additional Codes** (→ S. 116) (only if CIP Error Code = 0x1E)

CIP Error Code:

Code	description
0x02	Resource not available: The IO-Link port is busy processing another acyclic service.
0x05	Invalid class ID or instance ID
0x08	Wrong service ID: only service code 0x4B or 0x4C is permitted
0x09	Wrong attribute ID: wrong port number
0x20	Invalid parameter value (e.g. invalid length)
0x1E	Embedded service, error: Error occurred during an IO-Link service. Byte 0 and byte 1 of the User Specific Service Data contain the IO-Link error code and an additional code that are returned by the IO-Link master (see below).

Example: changing the parameter value of an IO-Link device

34355

Task: changing the parameter X of an IO-Link device

- IO-Link device at the port: 0x03
- Parameter X in the object directory of an IO-Link device: Index: 91, sub-index 5
- new parameter value: 0xABCD

From this, the following results for the configuration of the EtherNet/IP command Message (MSG):

CIP format	Data type	MSG Config	Description
CIP Class ID	UINT	0x80	IO-Link acyclic access
CIP Instance ID	UINT	0x01	IO-Link master
CIP Attributes	USINT	0x03	Port number
CIP Service code ID	USINT	0x4C	Service "Write_ISDU"
CIP User specific service data	UINT	0x005B	IO-Link ISDU object index
	USINT	0x05	IO-Link ISDU object sub-index
	USINT	0xAB	New parameter value (MSB)
	USINT	0xCD	New parameter value (LSB)

After successful execution of the request, the response area has the following content:

CIP format	Data type	MSG Config	Description
CIP Class ID	UINT	0x80	Object class "IO-Link Requests"
CIP Instance ID	UINT	0x01	IO-Link master
CIP Attributes	USINT	0x03	Port number
CIP Service code ID	USINT	0x4B	Service "Write_ISDU"
CIP Error code	USINT	0x00	Request processed successfully
CIP Extended error code	USINT	0x00	--

If an error occurs while the request is executed, the response area has the following content:

CIP format	Data type	MSG Config	Description
CIP Class ID	UINT	0x80	Object class "IO-Link Requests"
CIP Instance ID	UINT	0x01	IO-Link master
CIP Attributes	USINT	0x03	Port number
CIP Service code ID	USINT	0x4B	Service "Write_ISDU"
CIP Error code	USINT	0x1E	Error code: Embedded Service Error
CIP Extended error code	USINT	0x00	--
CIP User specific service data	USINT	0x80	IO-Link Error Code: Error in device application
	USINT	0x23	IO-Link Additional Code: Access denied

Write Failsafe Pattern

54597

By using Write Failsafe Pattern the fail-safe value of a IO-Link port can be written.

Request

54694

CIP Attribute determines the IO-Link port. The area CIP User Specific Service Data includes the fail-safe mode and the fail-safe value (Failsafe Pattern).

CIP Format	Data type	MSG Config	Description
CIP Class ID	UINT	0x80	IO-Link acyclic access
CIP Instance ID	UINT	0x01	IO-Link master
CIP Attribute	USINT	0x01...0x04	Port number
CIP Service Code ID	USINT	0x4D	Request "Write Failsafe Pattern"
CIP User Specific Service Data	USINT	0x00 = No Fail-safe 0x01 = Fail-safe Reset Value 0x02 = Fail-safe Old Value 0x03 = Fail-safe with Pattern	Failsafe Mode
	USINT	0x00...0xFF	Failsafe Pattern (MSB)
	USINT	0x00...0xFF	Failsafe Pattern (LSB)

Response

54695

- **Positive response**

If the service was executed successfully (CIP Error Code = 0), the area "User Specific Data" will remain empty. The response has the following format:

CIP Format	Data type	MSG Config	Description
CIP Class ID	UINT	0x80	IO-Link acyclic access
CIP Instance ID	UINT	0x01	IO-Link master
CIP Attribute	USINT	0x01...0x04	Port number
CIP Service Code ID	USINT	0x4D	Response "Write Failsafe Pattern"
CIP Error Code	USINT	0x00	--
CIP Extended Error Code	USINT	0x00	--

- **Negative response**

If an error occurs while executing the service (CIP Error Code <> 0), an extended error code is transmitted (CIP Extended Error Code). The answer has the following format:

CIP-Format	Data type	MSG Config	Description
CIP Class ID	UINT	0x80	IO-Link acyclic access
CIP Instance ID	UINT	0x01	IO-Link master
CIP Attribute	USINT	0x01...0x04	Port number
CIP Service Code ID	USINT	0x4D	Response "Write Failsafe Pattern"
CIP Error Code	USINT	<> 0x00	Error code: see below
CIP Extended Error Code	USINT	<> 0x00	Extended error code

CIP Error Code:

Code	Description
0x02	Resource not available: The IO-Link port is busy processing another acyclic service.
0x05	Invalid class ID or instance ID
0x08	Wrong service ID: only service 0x4B, 0x4C or 0x4D is permitted
0x09	Wrong attribute ID: wrong port number
0x20	Invalid parameter value (e.g. invalid length)
0x1E	Embedded service, error: Error occurred during an IO-Link service. Byte 0 and byte 1 of the User Specific Service Data contain the IO-Link error code and an additional code that are returned by the IO-Link master (see below).
0x0F	Insufficient access rights

Example: Write fail-safe value

54696

Task: Write fail-safe mode for IO-Link port X02 to "Fail-safe with pattern" and fail-safe value to 0x1234

- IO-Link device in the port: 0x02
- Fail-safe mode: 0x03
- Fail-safe value: 0x1234

From this, the following results for the configuration of the EtherNet/IP command message (MSG):

CIP Format	Data type	MSG Config	Description
CIP Class ID	UINT	0x80	IO-Link acyclic access
CIP Instance ID	UINT	0x01	IO-Link master
CIP Attribute	USINT	0x02	Port number
CIP Service Code ID	USINT	0x4D	Request "Write Failsafe Pattern"
CIP User Specific Service Data	USINT	0x03	"Fail-safe with Pattern" mode
	USINT	0x12	Failsafe Pattern (MSB)
	USINT	0x34	Failsafe Pattern (LSB)

After successful execution of the request, the response area has the following content:

CIP-Format	Data type	MSG Config	Description
CIP Class ID	UINT	0x80	IO-Link acyclic access
CIP Instance ID	UINT	0x01	IO-Link master
CIP Attribute	USINT	0x02	Port number
CIP Service Code ID	USINT	0x4D	Response "Write Failsafe Pattern"
CIP Error Code	USINT	0x00	Request processed successfully
CIP Exended Error Code	USINT	0x00	--

If an error occurs while the request is executed, the response area will have the following content:

CIP-Format	Data type	MSG Config	Description
CIP Class ID	UINT	0x80	IO-Link acyclic access
CIP Instance ID	UINT	0x01	IO-Link master
CIP Attribute	USINT	0x02	Port number
CIP Service Code ID	USINT	0x4D	Response "Write Failsafe Pattern"
CIP Error Code	USINT	e. g. 0x0F	Error code: Insufficient access rights
CIP Exended Error Code	USINT	0x00	no additional information

TCP/IP object (object class: 0xF5)

34388

TCP/IP Interface Object enables the configuration of the physical network interface of the device.

Class attributes

34311

Attr ID	Access	Name	Data type	Description	Value
1	Get	Revision	UINT	Revision of the object	4
2	Get	Max instance	UINT	Max. number of instances of the object	1

Instance attributes

34330

Attr. ID	Access	Name	Data type	Description	Preset
1	Get	Status	DWORD	Status of the TCP/IP interface	
				Bit 0...3: Configuration status of the interface	
				Bit 4: Mcast pending (always 0)	
				Bit 5: Interface configuration pending	
				Bit 6: ACD Status	
				Bit 7: ACD Fault	
				Bit 8...31: reserved	
2	Get	Configuration Capability	DWORD	Functions of the interface (flags)	0x95 (BOOTP,DHCP Client,TCP/IP configurable, ACD capable)
				Bit 0: BOOTP Client	
				Bit 1: reserved	
				Bit 2: DHCP Client	
				Bit 3: reserved	
				Bit 4: TCP/IP configurable via EtherNet/IP	
				Bit 5: reserved	
				Bit 6: reserved	
				Bit 7: ACD Capable	
				Bit 8...31: reserved	
3	Get, Set	Configuration Control	DWORD	Interface control (control flags):	0
				Bit 0...3: Start-up configuration	
				0: Static IP configuration	
				1: Configuration via BOOTP	
				2: Configuration via DHCP	
				Bit 4: reserved	
				Bit 5...31: reserved	
4	Get	Physical Link Object path	STRUCT:	Logical path to the physical communication interface: the Ethernet Link object	
		▪ Path Size	▪ UINT	Length (in Little Endian Format as WORD)	02 00

Attr. ID	Access	Name	Data type	Description	Preset
		▪ Path	▪ Padded EPATH	Path	20 F6 24 01
				Class ID = 0xF6 Ethernet Link Object	
				Instance ID = 1	
5	Get, Set	Interface Configuration	STRUCT:	TCP/IP configuration	
		▪ IP Address	▪ UDINT	IP adress	192.168.1.250
		▪ Network mask	▪ UDINT	Subnet mask	255.255.255.0
		▪ Gateway address	▪ UDINT	Default gateway address	0.0.0.0
		▪ Name Server	▪ UDINT	1. Name Server	0.0.0.0
		▪ Name Server 2	▪ UDINT	2. Name Server	0.0.0.0
		▪ Domain Name	▪ STRING	Default domain name	0
6	Get, Set	Host name	STRING	Host name	0
				0 no name configured	
8	Get	TTL value		TTL value	1
9	Get	Mcast Config			0
10	Get, Set	SelectAcd	BOOL	activate ACD	1
				0 deactivate	
				1 activate	
11	Get, Set	Last Conflict Detected	STRUCT:	Structure with information via the latest detected conflict	0
			▪ USINT	Condition of the ACD activity with the latest detected conflict	
				0 Noconflictdetetcted	
				1 Probelpv4Address	
				2 OngoingDetection	
				3 SemiActiveProbe	
			▪ ARRAY of 6 USINT	MAC address	
			▪ ARRAY of 28 USINT	Copy of the data of the ARP PDU in which the conflict was detected	
13	Get, Set	Encapsulation Inactivity Timeout	UINT	Inactivity before the TCP connection is deactivated (in seconds)	120

Supported services

34416

Service code		Name	Class	Attribute	Description
dec	hex				
01	01	Get_Attribute_All	no	yes	Read all attributes
14	0E	Get_Attribute_Single	yes	yes	Read single attribute
16	10	Set_Attribute_Single	no	yes	Change single attribute

Ethernet Link Object (object class: 0xF6)

34354

The Ethernet Link Object contains status information of the Ethernet interface.

Class attributes

34312

Attr ID	Access	Name	Data type	Description	Value
1	Get	Revision	UINT	Revision of the object	4
2	Get	Max Instance	UINT	Max. number of instances of the object	2
3	Get	Number of Instances	UINT	Number of instances of the object	2

Instance attributes

34333

Attr. ID	Access	Name	Data type	Description		Preset
1	Get	Interface Speed	UDINT	Current data rate (in bytes/s) 10 Mbps,100 Mbps.		100
2	Get	Interface Status Flags	DWORD	Status flag of the interface		0x20
				Bit 0	Link status	
				Bit 1	Half/full duplex	
				Bit 2...4	Auto negotiation status	
				Bit 5	Manual setting requires reset	
				Bit 6	Local Hardware Fault	
				Bit 7...31	reserved	
3	Get	Physical Address	ARRAY of 6 USINTS	MAC address		
4	Get	Interface Counters	STRUCT of 11 UDINTs	Interface-specific counter		
5	Get	Media counters	STRUCT of 12 UDINTs	Medium-specific counter		
6	Get, Set	Interface control	STRUCT of	Control bits: Bit 0: Auto negotiate Bit 1: Forced Duplex Mode (full 1, half 0)		0
			▪ WORD	Control bits of the interface		
				Bit 0	0 = auto-negotiation active	
					1 = auto-negotiation inactive	
				Bit 1	0 = Half duplex	
					1 = Full duplex	
				Bit 2..15	reserved	
			▪ UINT	Data rate of the interface		
				10	10 Mbps	
				100	100 Mpbs	
7	Get	Interface Type	USINT	Physical interface type		2
				0	unknown	
				1	Internal interface	

Attr. ID	Access	Name	Data type	Description		Preset
				2	Twisted pair	
				3	Optical fibre	
				4...255	reserved	
8	Get	Interface state	USINT	Current status of the interface		0
				0	unknown	
				1	active; ready for transmission and reception	
				2	not active	
				3	Test mode	
				4...255	reserved	
9	Get	Admin State	USINT	Control of the access to the interface		1
				0	reserved	
				1	Activate interface	
				2	Deactivate interface	
				3...255	reserved	
10	Get	Interface label	SHORT_ STRING	Designation of the interface		"X21" (instance 1) "X22" (instance 2)
11	Get	Interface capability	STRUCT of	Capabilities of the interface		
			▪ DWORD	Transmission rate		
				10	10 Mbps	
				100	100 Mbps	
			▪ DWORD	Duplex mode		
				HD	Half duplex	
				FD	Full duplex	
300	Get, Set	MDIX	???	MDIX configuration		3
				0		
				1	MDI	
				2	MDIX	
				3	autoMDI	
				4...255	reserved	

Supported services

34414

Service code		Name	Class	Attribute	Description
dec	hex				
01	01	Get_Attribute_All	no	yes	Read all attribute values
14	0E	Get_Attribute_Single	yes	yes	Read single attribute value
16	10	Set_Attribute_Single	no	yes	Change single attribute value

14.3 ifm IoT Core

Content

Overview: IoT profile....148
Overview: IoT types....154
Overview: IoT services....155

33803

14.3.1 Overview: IoT profile

Content

Profile: blob 148
Profile: deviceinfo 149
Profile: devicetag 149
Profile: iolinkdevice_full 150
Profile: iolinkmaster 150
Profile: network 151
Profile: parameter 151
Profile: processdata 151
Profile: service 152
Profile: software 152
Profile: software/uploadedablesoftware 152
Profile: Timer 153

34054

Profile: blob

52264

Element (identifier)	Properties	Mandatory	Comment
blobname	▪ type = data ▪ profiles = blob		labels element as device information
../size	type = data	mandatory	
../chunksize	type = data	mandatory	
../setblobdata	type = service	optional	
../getblobdata	type = service	optional	
../start_stream_set	type = service	optional	
../stream_set	type = service	optional	
../clear	type = service	optional	
../getcrc	type = service	optional	
../getmd5	type = service	optional	
../getdata	type = service	optional	
../setdata	type = service	optional	

Profile: deviceinfo

34207

Element (identifier)	Properties	mandatory	Comments
deviceinfo	▪ type = structure ▪ profile = deviceinfo		characterises the element as device information
../devicename	type = data	optional	
../devicefamiliy	type = data	optional	
../devicevariant	type = data	optional	
../devicesymbol	type = data	optional	
../deviceicon	type = data	optional	
../serialnumber	type = data	mandatory	
../productid	type = data	optional	
../productname	type = data	optional	
../productcode	type = data	mandatory	
../producttext	type = data	optional	
../ordernumber	type = data	optional	
../productiondate	type = data	optional	
../productioncode	type = data	optional	
../hwrevision	type = data	mandatory	
../swrevision	type = data	mandatory	
../bootloaderrevision	type = data	optional	
../vendor	type = data	optional	
../vendortext	type = data	optional	
../vendorurl	type = data	optional	
../vendorlogo	type = data	optional	
../productwebsite	type = data	optional	
../supportcontact	type = data	optional	
../icon	type = data	optional	
../image	type = data	optional	
../standards	type = data	optional	

Profile: devicetag

34206

Element (identifier)	Properties	mandatory	Comments
devicetag	▪ type = structure ▪ profile = devicetag		
../applicationtag	type = data	mandatory	
../applicationgroup	type = data	optional	
../machinecode	type = data	optional	
../tenant	type = data	optional	

Profile: iolinkdevice_full

52265

Element (identifier)	Characteristics	Mandatory	Comments
iolinkdevice	▪ type = structure ▪ profile = iolinkdevice_full		Structure of an IO-Link device
../vendorid	type = data	mandatory	
../deviceid	type = data	mandatory	
../productname	type = data	mandatory	
../serial	type = data	mandatory	
../applicationspecifictag	type = data	mandatory	
../pdin	type = data	mandatory	
../pdout	type = data	mandatory	
../status	type = data	mandatory	
../iolreadacyclic	type = data	mandatory	
../iolwriteacyclic	type = data	mandatory	
../iolinkevent	type = data	mandatory	

Profile: iolinkmaster

34205

Element (identifier)	Properties	Mandatory	Comments
masterport	▪ type = structure ▪ profile = iolinkmaster		Executable service
../mode	▪ type = data ▪ profile = parameter	mandatory	
../comspeed	▪ type = data ▪ profile = parameter	mandatory	
../mastercycletime_actual	▪ type = data ▪ profile = parameter	mandatory	
../mastercycletime_preset	▪ type = data ▪ profile = parameter	mandatory	
../validation_datastorage_mode	▪ type = data ▪ profile = parameter	mandatory	
../validation_vendorid	▪ type = data ▪ profile = parameter	mandatory	
../validation_deviceid	▪ type = data ▪ profile = parameter	mandatory	
../additionalpins_in	▪ type = data ▪ profile = processdata	optional	
../additionalpins_out	▪ type = data ▪ profile = processdata	optional	
../portevent	▪ type = data	mandatory	
../iolinkdevice	▪ type = structure ▪ profile = iolinkdevice_full	mandatory	

Profile: network

52266

Element (identifier)	Characteristics	Mandatory	Comments
network	▪ type = structure ▪ profiles = deviceinfo		Characterises the element as device information
../macaddress	▪ type = data ▪ profile = parameter	mandatory	
../ipaddress	▪ type = data ▪ profile = parameter	optional	
../ipv6address	▪ type = data ▪ profile = parameter	mandatory	
../subnetmask	▪ type = data ▪ profile = parameter	mandatory	
../ipdefaultgateway	▪ type = data ▪ profile = parameter	mandatory	
../dhcp	▪ type = data ▪ profile = parameter	optional	
../ipversion	▪ type = data ▪ profile = parameter	optional	
../hostname	▪ type = data ▪ profile = parameter	optional	
../autonegotiation	▪ type = data ▪ profile = parameter	optional	
../portspeed	▪ type = data ▪ profile = parameter	optional	
../enablenetwork	type = service	optional	
../disablenetwork	type = service	optional	

Profile: parameter

34215

The profile is used to mark the elements of type `data` as parameters (acyclic data). The profile defines no substructure.

Profile: processdata

34225

The profile is used to mark the elements of type `data` as process data (cyclic data). The profile does not define a substructure.

Profile: service

34224

Element (identifier)	Properties	mandatory	Comments
service	▪ type = service ▪ profile = service		Executable service

Profile: software

34223

Element (identifier)	Properties	mandatory	Comments
software	▪ type = structure ▪ profile = software		characterises the element as software
../version	type = data	mandatory	
../type	type = data	mandatory	
../status	type = structure	optional	
../diag	type = structure	optional	

Profile: software/uploadedablesoftware

52267

Element (identifier)	Characteristics	Mandatory	Comments
software	▪ type = structure ▪ profiles = software/uploadablesoftware		Software that can be loaded to the device via the IoT Core
../lastinstall	type = data	optional	
../installhistory	type = data	optional	
../container	▪ type = data ▪ profile = blob	mandatory	
../preinstall	type = service	optional	
../install	type = service	mandatory	
../postinstall	type = service	optional	
../abortinstall	type = service	optional	
../installstatus	type = data	optional	

Profile: Timer

34226

Element (identifier)	Properties	Mandatory	Comment
timer	▪ type = structure ▪ profile = timer		
../counter	▪ type = data ▪ profile = parameter	mandatory	
../interval	▪ type = data ▪ profile = parameter	optional	
../start	type = service	optional	
../stop	type = service	optional	

14.3.2 Overview: IoT types

34055

The ifm IoT Core uses the following element types:

Name	Description
structure	Element is a structure element (like a folder in a file system)
service	Element is a service that can be addressed from the network
event	Element is an event that can be started by the firmware and sends messages.
data	Element is a data point
device	Root element a device represents

14.3.3 Overview: IoT services

Content

Service: factoryreset....155
Service: getblobdata....156
Service: getdata....156
Service: getdatamulti....157
Service: getelementinfo....157
Service: getidentity....158
Service: getsubscriberlist....159
Service: getsubscriptioninfo....160
Service: gettree....161
Service: install....162
Service: iolreadacyclic....162
Service: iolwriteacyclic....162
Service: querytree....163
Service: reboot....163
Service: setblock....164
Service: setdata....165
Service: signal....165
Service: start_stream_set....165
Service: stream_set....166
Service: subscribe....166
Service: unsubscribe....167
Service: validation_useconnecteddevice....167

34056

Service: factoryreset

34184

Name: factoryreset

Description: The service sets the parameters of the device to the factory settings.

Request data (field "data"): none

Response data (field "data"): none

Example:

```
{
"code": "request",
"cid": 4711,
"adr": "/firmware/factoryreset"
}
```

Service: getblobdata

52345

Name: getblobdata
Description: The service reads a binary large object (blob).
Applicable to: datastorage
Request data (field "data"):

Data field	Required field	Data type	Default	Description
pos	mandatory	number	0	Byte position
length	mandatory	number	-	Size of the object (number of bytes)

Return data (field "data"):

Data field	Required field	Data type	Default	Description
data	mandatory	STRING	0	Data to be decoded (BASE64 coded)
crc	optional	HEX STRING		CRC of the data after decoding
md5	optional	HEX STRING		MD5 checksum of the data after decoding

Service: getdata

34183

Name: getdata
Description: Service reads the value of a data point and provides it.
Request data (field "data"): none
Return data (field "data"):

Data field	Required field	Data type	Description
value	mandatory	STRING	Value of the element/data point

Example:

```
{
"code":"request",
"cid":4711,
"adr":"devicetag/applicationtag/getdata"
}
```

Service: getdatamulti

34174

Name: getdatamulti

Description: The service sequentially reads the values of several data points and provides them. The value and the diagnostic code are provided for each data point.

Request data (field "data"):

Data field	Required field	Data type	Description
datatosend	mandatory	ARRAY OF STRINGS	List of data points to be requested; data points must support the service getdata ("datatosend":["url1","url2",...,"urlx"])

Response data (field "data"): for each requested data point

Data field	Required field	Data type	Description
url	mandatory	STRING	Data point request
code	mandatory	INT	Diagnostic code of the request
data	mandatory	STRING	Value of the data point

Service: getelementinfo

52269

Name: getelementinfo

Description: The service reads the properties of an element of the IoT tree.

Applicable to: Objects of the type `device`

Request data (field "data"):

Data field	Required field	Data type	Default	Description
adr	mandatory	STRING		URL of the element, which properties to be changed

Return data (field "data"):

Data field	Required field	Data type	Default	Description
identifier	mandatory	STRING		Identifier of the element
type	mandatory	STRING		Type of the element
format	optional	JSON object	blank	Format of the data or the service content
uid	optional	STRING	blank	
profiles	optional	JSON array	blank	
hash	optional	STRING	--	

Service: getidentity

34173

Name: getidentity

Description: The service reads the complete device description of the AL1421 and provides it as JSON object.

Applicable to: Objects of the `device` type

Request data: none

Return data (data):

Data field	Required field	Data type	Default	Description
iot		device		Device description as JSON object
iot.name	mandatory	STRING		
iot.uid	optional	STRING		
iot.version	mandatory	STRING		
iot.catalogue	optional	ARRAY OF OBJECTS		
iot.deviceclass	optional	ARRAY OF STRING		
iot.serverlist		ARRAY OF OBJECTS		
device	optional			AL1421
device.serialnumber	optional			Serial number
device.hwrevision	optional			Hardware version
device.swrevision	optional			Software version
device.custom	optional			

Service: getsubscriberlist

61191

Name: getsubscriberlist
Description: The service provides a list of all active subscriptions.
Request data ("data" field): none
Return data ("data" field): Array with the following data

Data field	Mandatory field	Data type	Description
adr	mandatory	STRING	Data source
datatosend	mandatory	ARRAY OF STRINGS	List with URLs of the subscribed data points
cid	mandatory	NUMBER	ID of the subscription
callbackurl	mandatory	STRING	Address to which IoT Core event notifications are to be sent;
duration	mandatory	STRING	Storage duration of the value

Example:

- **Request object:**

```
{
"code":"request",
"cid":4711,
"adr":"/getsubscriberlist"
}
```

- **Return object:**

```
{
"cid":4711,
"data":[
{
"adr":"/timer[1]/counter/datachanged/subscribe",
"datatosend":["/iolinkmaster/port[2]/iolinkdevice/pdin"],
"cid":1,
"callbackurl":"http://192.168.0.45:80/temp",
"duration":"lifetime"},
{
"adr":"/timer[1]/counter/datachanged/subscribe",
"datatosend":["/processdatamaster/temperature","/processdatamaster/voltage"],
"cid":2,
"callbackurl":"http://192.168.0.44:80/temp",
"duration":"lifetime"}
]
"code":200
}
```

Service: getsubscriptioninfo

61192

Name: getsubscriptioninfo
Description: The service provides information about an existing subscription (subscribe).

The following parameters of the existing subscription are to be used for the query:

- Value of the identifier cid (e.g. 4711)
- Number of the timer (e.g. timer[1])
- Name of the callback topic (e.g. B. temp)

Request data ("data" field):

Data field	Mandatory field	Data type	Description
callback	mandatory	STRING	Address to which IoT Core event notifications are to be sent; complete URL: http://ipaddress:port/path

Return data ("data" field):

Data field	Mandatory field	Data type	Description
subscription	mandatory	BOOL	Status of the transferred subscription parameter
datatosend	mandatory	ARRAY OF STRINGS	List with subscribed data points
cid	mandatory	NUMBER	ID of the subscribe request
callbackurl	mandatory	STRING	Address to which IoT Core event notifications are to be sent; complete URL: http://ipaddress:port/path

Example:

- **Request object:**

```
{
"code":"request",
"cid":4711,
"adr":"/timer[1]/counter/datachanged/getsubscriptioninfo",
"data":{
"callback":"http://192.168.0.44:80/temp"}
}
```

- **Return object:**

```
{
"cid": 4711,
"data":{
"subscription": true,
"datatosend":[
"/iolinkmaster/port[2]/iolinkdevice/productname",
"/iolinkmaster/port[2]/iolinkdevice/pdin",
"/processdatamaster/temperature"],
"callbackurl":"http://192.168.0.44:80/temp",
"duration":"lifetime"},
"code":200
}
```

Service: gettree

61193

Name: gettree

Description: The service reads the device description of the IO-Link master and outputs it as a JSON object. The output can be limited to a subtree of the device description.

Request data ("data" field):

Data field	Mandatory field	Data type	Description
adr	optional	STRING	Root element of the subtree
level	optional	STRING	max. level up to which the subtree is output ▪ no entry: all levels will be displayed ▪ 0: do not display sub-elements ("subs") ▪ 1: display sub-elements ▪ 2: display sub-elements up to the 2nd level ▪ 3: display sub-elements up to the 3rd level ... ▪ 20: display sub-elements up to the 20th level

Return data ("data" field):

Data field	Mandatory field	Data type	Description
identifier	mandatory	STRING	Identifier of the root element
type	mandatory	STRING	Type of the element
format	optional	JSON Object	Format of the data content
uid	optional	STRING	
profiles	optional	JSON-Array	
subs	mandatory	JSON-Array	Sub-elements
hash	optional	STRING	

Examples:

- output the complete device description

```
{
"code":"request",
"cid":4,
"adr":"/gettree"
}
```

- output the subtree counter[2] of the device description up to the 2nd level

```
{
"code":"request",
"cid":4,
"adr":"/gettree"
"data":{
"adr":"counter[2]",
"level":2}
}
```

Service: install

52343

Name: install

Description: The service installs the firmware stored in the container area of the device.

Applicable to: container

Request data (data): none

Return data (data): none

Service: iolreadacyclic

34178

Name: iolreadacyclic

Description: The service acyclically reads the parameter value of an IO-Link device. The parameter is accessed via IO-Link index and subindex.

Request data (field "data"):

Data field	Required field	Data type	Description
index	mandatory	NUMBER	IO-Link index of the parameter
subindex	mandatory	NUMBER	IO-Link subindex of the parameter

Response data (field "data"):

Data field	Required field	Data type	Description
value	mandatory	STRING	Value of the parameter; Value in hexadecimal format

Service: iolwriteacyclic

34177

Name: iolwriteacyclic

Description: The service acyclically writes the parameter value of an IO-Link device. The parameter is accessed via IO-Link index and subindex.

Request data (field "data"):

Data field	Required field	Data type	Description
index	mandatory	NUMBER	IO-Link index of the parameter
subindex	mandatory	NUMBER	IO-Link subindex of the parameter
value	mandatory	STRING	New value of the parameter; Value in hexadecimal format

Response data (field "data"): none

Service: querytree

61194

Name: querytree

Description: The service searches a device tree for the criteria profile, type and name and outputs a list with the URLs of the elements found. At least one of the search criteria must be specified. The service can only be executed on the root node of the machine.

Return data ("data" field):

Data field	Mandatory field	Data type	Description
profile	optional	STRING	Profile of the searched element
type	optional	STRING	Type of the searched element
name	optional	STRING	Type of the searched element

Return ("data" field):

Data field	Mandatory field	Data type	Description
urlList	mandatory	Array	Array with URLs of the found elements; URLs are separated by commas

Service: reboot

34176

Name: reboot

Description: The service reboots the device.

Request data (field "data"): none

Return data (field "data"): none

Example:

```
{
"code":"request",
"cid":4,
"adr":"firmware/reboot"
}
```

Service: setblock

34186

Name: setblock

Description: The service simultaneously sets the values of several data points of a structure.

Request data (field "data"):

Data field	Required field	Data type	Description
datatoset	mandatory	ARRAY OF OBJECTS	List of data points and their new values; data points must support the service setdata
consistent	optional	BOOL	

Response data (field "data"): none

Example:

Request:

```
{
"code":"request",
"cid":4711,
"adr":"iotsetup/network/setblock",
"data":{
"datatoset":{
"ipaddress":"192.168.0.6",
"subnetmask":"255.255.255.0",
"ipdefaultgateway":"192.168.0.250",
"dhcp":0}
}
}
```

Response:

```
{
"cid":4711,
"code":233
}
```

Service: setdata

34195

Name: setdata
Description: The service sets the value of the data point.
Request data ("data" field):

Data field	Mandatory field	Data type	Description
newvalue	mandatory	STRING	New value of the element/data point
duration	mandatory	STRING	Duration of value storage ▪ lifetime: Value is saved with IoT Core; Value remains valid even after restart of the device ▪ uptime: Value is saved until the next restart of the device

Return data ("data" field): none

Example:

```
{"code": "request",
"cid":4711,
"adr": "devicetag/applicationtag/setdata",
"data":{
"newvalue": "ifm IO-Link master
"duration": "lifetime"}
}
```

Service: signal

33819

Name: signal
Description: The service starts the flashing of the status LEDs of the AL1421.
Request data (field "data"): none
Return data (field "data"): none

Example:

```
{
"code":"request",
"cid":4711,
"adr":"firmware/signal"
}
```

Service: start_stream_set

52342

Name: start_stream_set
Description: The service starts the sequential transfer of multiple data segments.
Applicable to: Objects of type data
Request data (data):

Data field	Required field	Data type	Default	Description
size	mandatory	STRING		Total size of data to be transfered (number of bytes)

Return data (data): none

Service: stream_set

52341

Name: stream_set

Description: The service transfers a data segment.

Applicable to: Objects of type data

Request data (data):

Data field	Required field	Data type	Default	Description
value	mandatory	BIN (BASE64)	*	Segment of binary data (BASE64 coded)

Return data (data): none

Service: subscribe

61365

Name: subscribe

Description: The service subscribes to the values of data points. The data points to be subscribed are transferred as a list. The IoT Core sends changes to the data sink defined in callback.

Request data ("data" field):

Data field	Mandatory field	Data type	Description
callback	mandatory	STRING	Address to which IoT Core event notifications are to be sent; URL format: ▪ JSON: http://ipaddress:port/path ▪ CSV: tcp://ipaddress:port/path
datatosend	mandatory	ARRAY OF STRINGS	List from URLs of data elements; Elements must support getdata
codec	optional	STRING	Format of the returned data ▪ json: JSON formatted ▪ csv: CSV with standard separator (,) ▪ csv0: CSV formatted with comma separator (,) ▪ csv1: CSV formatted with semicolon separator (;)
duration	mandatory	STRING	Duration of value storage ▪ lifetime: Value is saved with IoT Core; Value remains valid even after restart of the device ▪ uptime: Value is saved until the next restart of the device ▪ once: send only one notification. After that, subscription will be cancelled.

Return data ("data" field): none

Service: unsubscribe

34197

Name: unsubscribe

Description: The service deletes an existing subscription. The service unsubcribe is successful if cid and the callback address are registered for an active subscription (subscribe). If the STRING "DELETE" is provided in callback, the IO-Link master deletes all active subscriptions.

Request data (field "data"):

Data field	Required field	Data type	Description
callback	mandatory	STRING	Address to which IoT Core event notifications are to be sent; complete URL: http://ipaddress:port/path

Response data (field "data"): none

Service: validation_useconnecteddevice

52340

Name: validation_connecteddevice

Description: The service checks, whether Device ID and Vendor ID of the connected IO-Link device match with the values of the datapoints `../validation_vendorid` and `../validation_deviceid`.

Applicable to: Objects of type `stucture`

Request data (data): none

Return data (data): none

15 Index

A

Access the ifm IoT Core 39
Accessories 84
Acyclic command channel 112
Acyclic commands 117
Acyclic data 112
Acyclic port commands 80
Additional Codes 116
Appendix 85
Application 86
Approvals / tests 92
Assembly Object (object class 0x04) 129

C

CIP class services 124
CIP object classes 125
Class attributes 126, 128, 129, 130, 131, 132, 133, 143, 145
Cleaning process 82
Command 0x10 – Set mode 118
Command 0x20 – Set validation ID / data storage 120
Command 0x30 – Set fail-safe data pattern 122
Command channels in cyclic process data 79
Command request 118, 120, 122
Command response 119, 121, 123
Configuration 22
Configuration Assembly (Instance 199) 95
Configure connection types 75
Configure IO-Link devices 36, 77
Configure IO-Link ports 76
Configure the IO-Link master 76
Configuring IO-Link the master 70
Connect EtherNet/IP ports 14
Connect IO-Link ports 15
Connect the device 17
Connection Manager Object (object class 0x06) 130
Creating a new notification 67
Cyclic data 100

D

Deleting a notification 68
Derating behaviour 87
Derating with Daisy Chain 88
Derating witout Daisy Chain 87
Device Level Ring Object (object class 0x47) 131
Digital outputs 11

E

Electrical connection 13, 93
Electrical data 86
Environmental conditions 92
Error codes 116
Ethernet Link Object (object class 0xF6) 145
Ethernet ports 19
EtherNet/IP 10, 73, 94
  Programmers' notes 73
EtherNet/IP mechanisms for acyclic commands 80
Example
  Browsing device description 45
  Change name of the IO-Link master 60
  Change the parameter value of an IO-Link device 56
  Changing a subscription 62
  changing the parameter value of an IO-Link device 139
  Checking subscriptions 63
  Clone the Data Storage of an IO-Link port 50
  GET request 39
  output subtree 44
  POST request 40
  Read IO-Link process data (operating mode 52
  Read several parameter values of the IO-Link master simultaneously 45
  Read the parameter value of an IO-Link device 55
  Reading digital input (operating mode 54
  Reading properties of an element 43
  reading the parameter value of anIO-Link device 136
  Subscribing notifications via WebSocket 64
  Subscribing to notifications 46, 61
  Unsubscribing from notifications 63
  Update firmware 59
  Write fail-safe value 142
  Writing digital output (operating mode 53
  Writing IO-Link value (operating mode 53
Exchanging the IO-Link device 82
Explanation of Symbols 6

F

Factory Settings 83
Field bus objects 124
Fieldbus
  Configure IP settings 28
  Configuring IP settings 47
  Selecting the configuration mode 47
  set the configuration mode 29
  Setting failsafe values 48
Firmware
  Reboot the device 35
  Reset device to factory settings 35
First steps 43
Function 10

G

Gateway
  Reading device information 58
  Reading status and diagnostic information 58
  Resetting, rebooting and localising the device 57
  Setting the application tag 60
  Updating the firmware 59
General 7
General functions 43
GET request 39
Ground the device 16

I

Identity Object (object class 0x01) 126
ifm IoT Core 37, 147
Info
  Show device information 34
Input Assembly (Instance 100)

I/O + Acyclic + Diag ... 100
Input Assembly (Instance 102)
I/O ... 102
Input Assembly (Instance 103)
I/O + Acyclic + Diag + EnMo ... 103
Inputs ... 91
Inputs/outputs ... 91
Installing the device ... 12
Instance attributes ... 126, 128, 130, 131, 132, 133, 143, 145
Instances ... 129
Integrate the IO-Link Master into the EtherNet/IP project ... 74
Intended use ... 9
Interfaces ... 92
Internet of Things (IoT) ... 11
IO-Link ... 11
IO-Link devices
Accessing parameters ... 55
Indicating IO-Link events ... 57
Reading an writing device information ... 57
IO-Link ports
Setting fail-safe values ... 34
IO-Link ports
Activate data transfer to LR AGENT or LR SMARTOBSERVER ... 30
Limit the current intensity ... 32
Set the device validation and data storage ... 33
Set the operating mode Pin 2 (UA) ... 30
setting the operating mode Pin 4 (US) ... 31
IO-Link ports
Limiting current values ... 49
IO-Link ports
Setting the operating mode of pin 4 (US) ... 49
IO-Link ports
Configuring device validation and data storage ... 50
IO-Link ports
Configuring data transfer to LR AGENT or LR SMARTOBSERVER ... 52
IO-Link ports
Reading and writing process data ... 52
IO-Link ports
Indicating port events ... 55
IO-Link Ports (Class B) ... 20
IO-Link requests (object class
0x80) ... 133
IO-Link supply ... 11
IoT
Configure IP settings ... 25
Configure the interface to LR AGENT or LR SMARTOBSERVER ... 27
Configuring access rights ... 26, 48
Configuring the LR AGENT or LR SMARTOBSERVER interface ... 49
IoT core
Diagnostic codes ... 42
IoT Core
General information ... 38
IT security ... 8

L

LED indicators ... 19
Legal and copyright information ... 5
LR DEVICE ... 23

M

Maintenance, repair and disposal ... 82
Managing notifications ... 67
Mapping
Current Limit - Pin 1 + Pin 4 US ... 99
Current Limit - Pin 2 UA ... 99
Current Pin 1 + Pin 4 (Sensor US) ... 109
Current Pin 2 (Actuator UA) ... 109
IO-Link device information + events ... 106
Port configuration ... 97
Port Qualifier Information (PQI) ... 105
Voltage + Current Actuator Supply UA ... 108
Voltage + Current Sensor Supply US ... 108
Mechanical data ... 93
Message Router Object (object class
0x02) ... 128
Modification history ... 6
Mounting ... 12

O

Offline parameter setting ... 24
Online parameter setting ... 24
Operating and display elements ... 18
Operation ... 81
Output Assembly (Instance 150)
I/O + Acyclic ... 110
Output Assembly (Instance 151)
I/O ... 111
Outputs ... 91
Overview ... 18
IoT profile ... 148
IoT services ... 155
IoT types ... 154

P

Parameter data ... 95
Parameter setting ... 10
Parameter setting options ... 21
Permitted use ... 9
POST request ... 40
Preliminary note ... 5
Principle of the command channels ... 79
Profile
blob ... 148
deviceinfo ... 149
devicetag ... 149
iolinkdevice_full ... 150
iolinkmaster ... 150
network ... 151
parameter ... 151
processdata ... 151
service ... 152
software ... 152
software/uploadedablesoftware ... 152
Timer ... 153
Programmers' notes ... 38
Prohibited use ... 9
Purpose of the document ... 5

Q

Quality of Service (object class
0x48) ... 132

R

Read diagnostic information and events ... 78
Read process data ... 77
Read_ISDU ... 134

Reading and writing process data 71
Registration of the EDS file 74
Remarks 13, 24
Request 134, 137, 140
Request channel 113
Required background knowledge 7
Response 134, 137, 140
Response channel 114

S

Safety instructions 7
Safety symbols on the device 7
Searching for elements in the device tree 69
Service
- factoryreset 155
- getblobdata 156
- getdata 156
- getdatamulti 157
- getelementinfo 157
- getidentity 158
- getsubscriberlist 159
- getsubscriptioninfo 160
- gettree 161
- install 162
- iolreadacyclic 162
- iolwriteacyclic 162
- querytree 163
- reboot 163
- setblock 164
- setdata 165
- signal 165
- start_stream_set 165
- stream_set 166
- subscribe 166
- unsubscribe 167
- validation_useconnecteddevice 167

Setting the storage duration 46
Setup 21
Status LEDs 19
Subscribing to notifications 61
Supported configuration options 73
Supported connection types 94
Supported services 127, 128, 129, 130, 131, 132, 133, 144, 146

T

TCP/IP object (object class
- 0xF5) 143

Technical data 86

U

Updating the firmware 72, 82
Use acyclic services 79
Use web-based management 81
Using the IoT-Core Visualizer 66
Using Web Socket 64

V

Visual indication 10
Voltage output 11
Voltage supply 20
VPN connection 24

W

Write Failsafe Pattern 140
Write process data 78
Write_ISDU 137