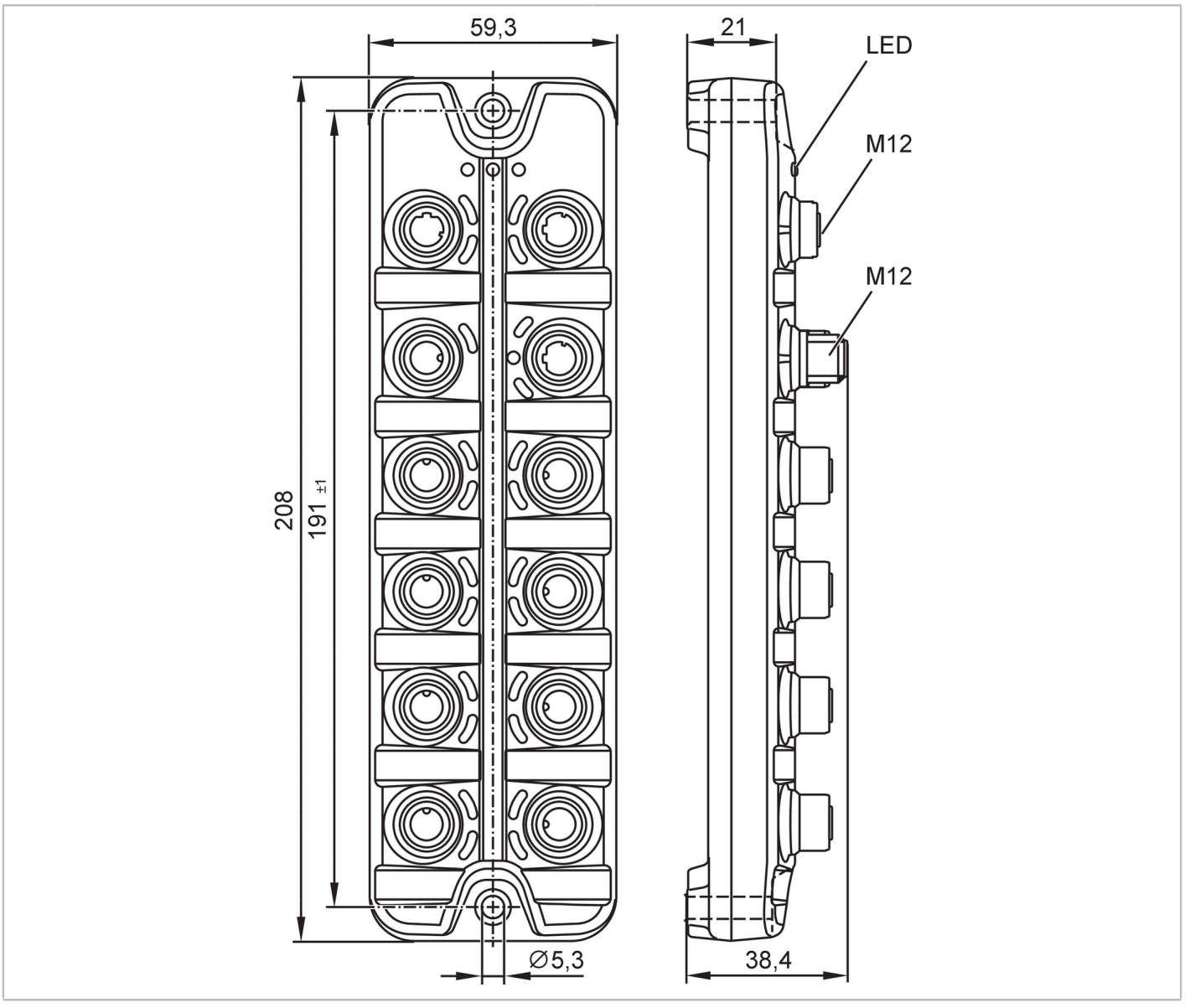
# AL1327

## IO-Link master with EtherNet/IP interface

IO-Link Master DL EIP 8P IP69K




| | | |
|---|---|---|
| **Application** | | |
| Application | | hygienic systems; I/O modules for field applications |
| Daisy-chain function | | fieldbus interface |
| **Electrical data** | | |
| Operating voltage | [V] | 20...28 DC;  (US ; to SELV/PELV) |
| Current consumption | [mA] | 300...3900;  (US) |
| Protection class | | III |
| Sensor supply US | | |
| Max. current load total | [A] | 3.6 |
| **Inputs / outputs** | | |
| Total number of inputs and outputs | | 16;  (configurable) |
| Number of inputs and outputs | | Number of digital inputs: 16; Number of digital outputs: 8 |

# AL1327

## IO-Link master with EtherNet/IP interface

IO-Link Master DL EIP 8P IP69K

| | | |
|---|---|---|
| **Inputs** | | |
| Number of digital inputs | | 16;  (IO-Link Port Class A: 8 x 2) |
| Switching level high | [V] | 11...28 |
| Switching level low | [V] | 0...5 |
| Digital inputs protected against short circuits | | yes |
| **Outputs** | | |
| Number of digital outputs | | 8;  (IO-Link Port Class A: 8 x 1) |
| Max. current load per output | [mA] | 300 |
| Short-circuit proof | | yes |
| **Interfaces** | | |
| Communication interface | | Ethernet; IO-Link |
| Ethernet - EtherNet/IP | | |
| Transmission standard | | 10Base-T; 100Base-TX |
| Transmission rate | | 10 MBit/s; 100 MBit/s |
| Protocol | | EtherNet/IP |
| Factory settings | | IP address: 192.168.1.250 |
| | | subnet mask: 255.255.255.255 |
| | | gateway IP address: 0.0.0.0 |
| | | MAC address: see type label |
| Note on interfaces | | DHCP, BOOTP, static |
| | | DLR (Device Level Ring) |
| | | Supported network topologies: line |
| | | ring |
| Ethernet - Internet of Things | | |
| Transmission standard | | 10Base-T; 100Base-TX |
| Transmission rate | | 10 MBit/s; 100 MBit/s |
| Protocol | | MQTT JSON |
| Factory settings | | IP address: 169.254.x.x |
| | | subnet mask: 255.255.0.0 |
| | | gateway IP address: 0.0.0.0 |
| | | MAC address: see type label |
| Note on interfaces | | Safety protocol: HTTPS |
| IO-Link Master | | |
| Transmission type | | COM1 (4,8 kBaud); COM2 (38,4 kBaud); COM3 (230,4 kBaud) |
| IO-Link revision | | 1.1 |
| Number of ports class A | | 8 |
| **Operating conditions** | | |
| Ambient temperature | [°C] | -25...60 |
| Storage temperature | [°C] | -25...85 |
| Max. relative air humidity | [%] | 90 |
| Protection | | IP 65; IP 66; IP 67; IP 69K;  (operation with stainless steel protective caps: IP 69K) |
| Protection rating (NEMA 250) | | 6P |
| Pollution degree | | 2 |

# AL1327

## IO-Link master with EtherNet/IP interface

IO-Link Master DL EIP 8P IP69K

| | | | |
|---|---|---|---|
| Chemical media | | ISO 16750-5 | HLP, CC, DB, DC, DD, CA |
| | | NEMA 250 5.13.1 | AA |
| **Tests / approvals** | | | |
| 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 | |
| MTTF | [years] | 76 | |
| **Mechanical data** | | | |
| Weight | [g] | 427.95 | |
| Housing | | rectangular | |
| Type of mounting | | Backplane mounting | |
| Dimensions | [mm] | 208 x 59.3 x 38.4 | |
| Material | | housing: PA grey; Socket: stainless steel (1.4404 / 316L) | |
| Sealing material | | EPDM | |
| **Accessories** | | | |
| Items supplied | | Protective cap: 1 x M12, stainless steel, E12542 | |
| **Remarks** | | | |
| Remarks | | Further information is available in the manual. | |
| Pack quantity | | 1 pcs. | |

### Electrical connection - Ethernet

Connector: 3 x M12; coding: D; Contacts: 4; sealing: EPDM




EtherNet/IP X21, X22

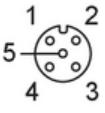
| | |
|---|---|
| 1 | TX + |
| 2 | RX + |
| 3 | TX - |
| 4 | RX - |
| 5 | not used |

IoT X23

| | |
|---|---|
| 1 | TX + |
| 2 | RX + |
| 3 | TX - |
| 4 | RX - |
| 5 | not used |

### Electrical connection - IO-Link

Connector: 8 x M12; coding: A; Contacts: 4; sealing: EPDM



# AL1327

## IO-Link master with EtherNet/IP interface

**IO-Link Master DL EIP 8P IP69K**

| IO-Link Port Class A X01...X08 | |
|---|---|
| 1 | Sensor supply (US) L+ |
| 2 | Digital input |
| 3 | Sensor supply (US) L- |
| 4 | C/Q IO-Link |
| 5 | not used |

## Electrical connection - voltage supply

Connector: 1 x M12; coding: A; Contacts: 4




| X31 | |
|---|---|
| 1 | + 24 V DC (US) |
| 2 | not used |
| 3 | GND (US) |
| 4 | not used |

GB

# Operating instructions

## IO-Link master with EtherNet/IP interface DataLine

**AL1327**

11501196 / 02 07 / 2024

# Contents

1 Preliminary note . . . 6
1.1 Legal and copyright information . . . 6
1.2 Symbols used . . . 6
1.3 Warnings . . . 6
1.4 Change history . . . 6
2 Safety instructions . . . 8
2.1 Cyber security . . . 8
2.2 Safety symbol on the device . . . 8
3 Intended use . . . 9
4 Function . . . 10
4.1 IO-Link . . . 10
4.1.1 IO-Link supply . . . 10
4.2 Parameter setting . . . 10
4.3 Visual indication . . . 10
4.4 EtherNet/IP . . . 10
4.5 ifm IoT Core . . . 11
4.6 Digital inputs . . . 11
5 Installation . . . 12
5.1 Overview . . . 12
6 Electrical connection . . . 13
6.1 General wiring information . . . 13
6.1.1 Connection technology . . . 13
6.2 Ethernet ports . . . 13
6.3 IO-Link ports (class A) . . . 14
6.3.1 Connecting IO-Link devices (class A) . . . 14
6.3.2 Connecting IO-Link devices (Class B) . . . 14
6.4 Ground connection . . . 15
7 Operating and display elements . . . 16
7.1 LEDs . . . 16
7.1.1 Status . . . 16
7.1.2 Ethernet . . . 16
7.1.3 IoT . . . 16
7.1.4 IO-Link ports (Class A) . . . 17
7.1.5 Voltage supply . . . 17
8 Set-up . . . 18
9 Settings . . . 19
9.1 Parameter setting software . . . 19
9.1.1 Supported parameter setting software . . . 19
9.1.2 Getting started . . . 19
9.1.2.1 Online parameter setting . . . 20
9.1.2.2 Offline parameter setting . . . 20
9.1.3 Configuring IP settings . . . 20
9.1.4 IoT interface . . . 20
9.1.4.1 Configuring access rights . . . 21
9.1.4.2 Configuring IP settings . . . 21
9.1.4.3 Configuring the interface to the monitoring software . . . 22
9.1.5 Fieldbus interface . . . 23
9.1.5.1 Configuring the EtherNet/IP interface . . . 23
9.1.5.2 Setting the EtherNet/IP configuration mode . . . 24
9.1.6 IO-Link ports . . . 24
9.1.6.1 Setting data transmission to the monitoring software . . . 24
9.1.6.2 Setting the operating mode Pin 4 (US) . . . 25
9.1.6.3 Setting the cycle time . . . 25
9.1.6.4 Setting the device validation and data storage . . . 26
9.1.6.5 Setting fail-safe values . . . 27
9.1.7 Device information . . . 28

9.1.7.1 Reading identification information . . . 28
9.1.8 Device control . . . 28
9.1.8.1 Resetting the device . . . 28
9.1.8.2 Restarting the device . . . 28
9.1.9 Updating the firmware . . . 29
9.1.10 Setting the parameters of IO-Link devices . . . 29
9.2 ifm IoT Core REST API . . . 31
9.2.1 ifm IoT Core: General information . . . 31
9.2.1.1 Accessing the ifm IoT Core . . . 31
9.2.2 Getting started . . . 31
9.2.3 General functions . . . 31
9.2.3.1 Example: Reading properties of an element . . . 31
9.2.3.2 Example: Outputting the subtree . . . 33
9.2.3.3 Example: Changing a parameter value . . . 35
9.2.3.4 Example: Read multiple elements sequentially . . . 36
9.2.3.5 Example: Browsing device description . . . 37
9.2.4 IoT: Configuring access rights . . . 37
9.2.5 IoT: Configuring IP settings . . . 37
9.2.6 IoT: Configuring security mode . . . 38
9.2.6.1 Example: Activating security mode . . . 39
9.2.6.2 Example: Request with authentication . . . 39
9.2.6.3 Example: Resetting the password . . . 40
9.2.7 IoT: Configuring the interface to the monitoring software . . . 40
9.2.8 Fieldbus: Configuring the EtherNet/IP interface . . . 41
9.2.9 Fieldbus: Setting the EtherNet/IP configuration mode . . . 41
9.2.10 Fieldbus: Setting fail-safe values of the outputs . . . 42
9.2.11 Ports: Setting the operating mode Pin 4 (US) . . . 42
9.2.12 Ports: Configuring device validation and data storage . . . 43
9.2.12.1 Example: Cloning the data memory of an IO-Link port . . . 44
9.2.13 Ports: Setting data transmission to the monitoring software . . . 46
9.2.14 Ports: Reading / writing process data . . . 46
9.2.14.1 Example: Reading an IO-Link process value (operating mode "IO-Link") . . . 46
9.2.14.2 Example: Reading an IO-Link process value (operating mode "IO-Link") . . . 47
9.2.14.3 Example: Reading the digital input value (operating mode "DI") . . . 48
9.2.14.4 Example: Writing the digital output value (operating mode "DO") . . . 48
9.2.15 Ports: Reading port events . . . 49
9.2.16 Timer: Configuring the timers . . . 49
9.2.16.1 Set the counting interval . . . 49
9.2.17 IO-Link devices: Accessing parameters . . . 50
9.2.17.1 Example: Reading a parameter value of an IO-Link device . . . 50
9.2.17.2 Example: Writing a parameter value of an IO-Link device . . . 51
9.2.18 IO-Link devices: Reading and writing device information . . . 51
9.2.19 IO-Link devices: Reading IO-Link events . . . 52
9.2.20 Gateway: Set the application tag . . . 53
9.2.21 Gateway: Update firmware . . . 53
9.2.22 Gateway: Reading status and diagnostic information . . . 53
9.2.23 Gateway: Reading device information . . . 53
9.2.24 Gateway: Resetting, rebooting and localising the device . . . 54
9.2.25 Notifications . . . 54
9.2.25.1 Event-controlled notifications . . . 54
9.2.25.2 Time-controlled notifications . . . 55
9.2.25.3 Example: Subscribing to notifications . . . 55
9.2.25.4 Example: Changing a notification . . . 56
9.2.25.5 Example: Subscribing to a notification in CSV format . . . 56
9.2.25.6 Example: Unsubscribing from a notification . . . 57
9.2.25.7 Example: checking a notification . . . 58
9.2.25.8 MQTT support . . . 58
9.2.25.9 WebSocket support . . . 62
9.3 IoT Core Visualizer . . . 64
9.3.1 Starting the IoT-Core Visualizer . . . 64
9.3.2 Managing notifications . . . 65
9.3.2.1 Creating a new notification . . . 65

9.3.2.2 Deleting a notification . . . 65
9.3.3 Searching for elements of the IoT Core . . . 66
9.3.4 Configuring the IO-Link master . . . 67
9.3.5 Accessing process data . . . 68
9.3.6 Update firmware . . . 69
9.4 EtherNet/IP . . . 70
9.4.1 Note: Startup package . . . 70

10 Maintenance, repair and disposal . . . 71
10.1 Cleaning the housing surface . . . 71
10.2 Update firmware . . . 71

11 Appendix . . . 72
11.1 ifm IoT Core . . . 72
11.1.1 Profiles . . . 72
11.1.2 Types . . . 72
11.1.3 Services . . . 73
11.1.3.1 Service: factoryreset . . . 73
11.1.3.2 Service: getblobdata . . . 73
11.1.3.3 Service: getdata . . . 73
11.1.3.4 Service: getdatamulti . . . 73
11.1.3.5 Service: getelementinfo . . . 74
11.1.3.6 Service: getidentity . . . 74
11.1.3.7 Service: getsubscriberlist . . . 75
11.1.3.8 Service: getsubscriptioninfo . . . 75
11.1.3.9 Service: gettree . . . 75
11.1.3.10 Service: install . . . 76
11.1.3.11 Service: iolreadacyclic . . . 76
11.1.3.12 Service: iolwriteacyclic . . . 76
11.1.3.13 Service: querytree . . . 77
11.1.3.14 Service: reboot . . . 77
11.1.3.15 Service: reset . . . 77
11.1.3.16 Service: setblock . . . 77
11.1.3.17 Service: setdata . . . 78
11.1.3.18 Service: signal . . . 78
11.1.3.19 Service: start . . . 78
11.1.3.20 Service: start_stream_set . . . 78
11.1.3.21 Service: stop . . . 78
11.1.3.22 Service: stream_set . . . 79
11.1.3.23 Service: validation_useconnecteddevice . . . 79
11.2 EtherNet/IP . . . 80
11.2.1 Parameters . . . 80
11.2.1.1 Configuration Assembly (Instance 199) . . . 80
11.2.2 Cyclic data . . . 83
11.2.2.1 Input Assembly (Instance 100) . . . 83
11.2.2.2 Input Assembly (Instance 101) . . . 84
11.2.2.3 Input Assembly (Instance 102) . . . 85
11.2.2.4 Output Assembly (Instance 150) . . . 90
11.2.2.5 Output Assembly (Instance 151) . . . 90
11.2.3 Acyclical data . . . 92
11.2.3.1 Acyclic command channel . . . 92
11.2.3.2 Acyclic commands . . . 97
11.2.4 Object directory . . . 102
11.2.4.1 CIP object classes . . . 102
11.2.4.2 CIP class and instance services . . . 102
11.2.4.3 Identity Object (Class Code: 0x01) . . . 103
11.2.4.4 Message Router Object (Class Code: 0x02) . . . 104
11.2.4.5 Assembly Object (Class Code: 0x04) . . . 105
11.2.4.6 Connection Manager Object (Class Code: 0x06) . . . 106
11.2.4.7 Device Level Ring Object (Class Code: 0x47) . . . 107
11.2.4.8 Quality Of Service Object (Class Code: 0x48) . . . 108
11.2.4.9 IO-Link Request Object (Class Code: 0x80) . . . 109
11.2.4.10 TCP/IP Object (Class Code: 0xF5) . . . 113

11.2.4.11 Ethernet Link Object (Class Code: 0xF6) . . . . . .116

# 1 Preliminary note

You will find instructions, technical data, approvals and further information using the QR code on the unit / packaging or at documentation.ifm.com.

## 1.1 Legal and copyright information

© All rights reserved by ifm electronic gmbh. No part of these instructions may be reproduced and used without the consent of ifm electronic gmbh.

All product names, pictures, companies or other brands used are the property of the respective rights owners.

## 1.2 Symbols used

| | |
|---|---|
| ✓ | Requirement |
| ▶ | Instructions |
| ▷ | Reaction, result |
| [...] | Designation of keys, buttons or indications |
| → | Cross-reference |
| | Important note<br>Non-compliance may result in malfunction or interference. |
| | Information<br>Supplementary note |

## 1.3 Warnings

Warnings indicate the possibility of personal injury and damage to property. This enables safe product handling. Warnings are graded as follows:

**WARNING**

Warning of serious personal injury

▷ If the warning is not observed, fatal and serious injuries are possible.

**CAUTION**

Warning of minor to moderate personal injury

▷ If the warning is not observed, minor to moderate injuries are possible.

**ATTENTION**

Warning of damage to property

▷ If the warning is not observed, damage to property is possible.

## 1.4 Change history

| Issue | Subject | Date |
|---|---|---|
| 00 | New creation of the document | 02 / 2023 |

| Issue | Subject | Date |
|---|---|---|
| 01 | • Changed: Identity Object (Class Code: 0x01) (→ 🗋 103)<br>• Removed: REST API – DNS support<br>• Added: WebSocket support (→ 🗋 62) | 07 / 2023 |
| 02 | • Changed: Chapter “Ground connection”, Ground connection (→ 🗋 15)<br>• Changed: Description and example MQTT command channel, MQTT support (→ 🗋 58)<br>• Added: Note regarding UL applications, Connection technology (→ 🗋 13) | 07 / 2024 |

# 2 Safety instructions

- The unit described is a subcomponent for integration into a system.
  - The system architect is responsible for the safety of the system.
  - The system architect 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 architect 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 (→ Intended use).
- 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, operation and maintenance of the product must be carried out by qualified personnel authorised by the machine operator.
- Protect units and cables against damage.

## 2.1 Cyber security

**ATTENTION**

Device operation in an unprotected network environment.

▷ Unauthorised read or write access to data is possible.

▷ Unauthorised manipulation of the device function is possible.

▶ Restrict access to authorised users (e.g. password-protected access).

▶ Choose a safe method to connect with the device (e. g. VPN).

▶ Use encrypted data transmission (e. g. https / TLS).

## 2.2 Safety symbol on the device



Safety symbol on the device:

▶ Adhere to the operating instructions for the safe operation of the device.

# 3 Intended use

The device may only be used for the following purposes:

- IO-Link master for configuration, management and operation of IO-Link devices
- Gateway between IO-Link devices and a higher-level EtherNet/IP control system

The device is designed for use without a control cabinet in the food industry.

# 4 Function

## 4.1 IO-Link

The device offers the following IO-Link functions:

- IO-Link master (IO-Link revision 1.0 and 1.1)

### 4.1.1 IO-Link supply

The device offers supplies for 8 IO-Link devices (sensors, actuators).

Ports X01...X08 are class A ports.

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).

## 4.2 Parameter setting

The device can be configured with the following options:

- Parameter setting software
  - ifm moneo OS + ifm moneo|configure
  - ifm moneo|configure free
- ifm IoT Core
  - ifm IoT Core Visualizer
- EtherNet/IP
  - Projection software

## 4.3 Visual indication

The device displays the following indications:

- Status and activity indication of the Ethernet connections

## 4.4 EtherNet/IP

The device supports supports the following EtherNet/IP functions:

- Device profile: EtherNet/IP device
- 2-port switch for access to the fieldbus interface
- Min. cycle time: 1 ms (RPI)
- Supported connection classes: 1, 3
- Supported connection types:
  - Exclusive Owner
  - Input Only
  - Listen Only
- UCMM support
- Predefined fieldbus objects
  - Identity Object (0x01)

  - Message Router Object (0x02)
  - Assembly Object (0x04)
  - Connection Manager (0x06)
  - DLR Object (0x47)
  - QoS Object (0x48)
  - IO-Link Request Object (0x80)
  - TCP/IP Interface Object (0xF5)
  - Ethernet Link Object (0xF6)
- Supported protocols: DHCP, BOOTP, ACD, DLR
- Device description: EDS file

## 4.5 ifm IoT Core

- Mapping of the data, events and services of the device to an object model (ifm IoT Core)
- Integration of the device in IIoT applications
- Access to the ifm IoT Core:
  - Browser-based interface (IoT Core Visualizer)

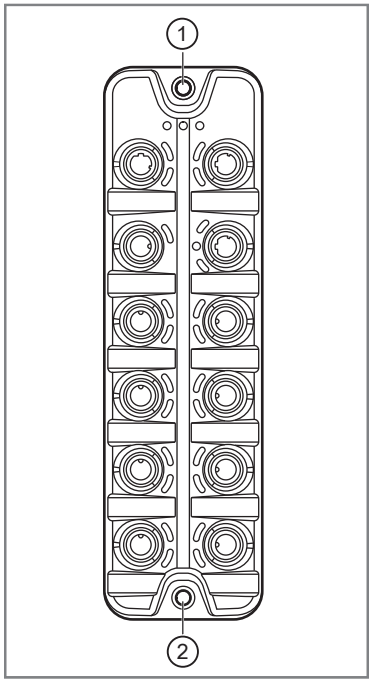
## 4.6 Digital inputs

The ports X01...X08 each have an additional digital input (type 2 according to EN 61131-2). The digital inputs are on pin 2.

The digital inputs are supplied via US. They refer to the potential of US (pin 3).

# 5 Installation

## 5.1 Overview




1: Upper mounting lug
2: Lower mounting lug

# 6 Electrical connection

## 6.1 General wiring information

The unit must be connected by a qualified electrician.

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

The unit is only suitable for operation using SELV/PELV voltages.

▶ Take information about IO-Link configuration into consideration!

This device contains components that may be damaged or destroyed by electrostatic discharge (ESD).

▶ Please observe the required precautions against electrostatic discharge!

By means of basic insulation according to EN61010-1, the circuits are isolated from each other and from device surfaces that could be touched (secondary circuit with max. 30 V DC, supplied from mains circuit up to 300 V overvoltage category II).

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

### 6.1.1 Connection technology

The threaded connections in the device correspond to the M12 standard. To ensure compliance with the specified protection rating, only cables that comply with this standard may be used. In the case of self-assembled cables, the system manufacturer is responsible for the protection rating.

▶ Use connectors with gold-plated contacts.

▶ Check the screw connection 2 weeks after installation and retighten to the correct torque if necessary.

▶ During installation, place the connectors vertically so that the coupling nut will not damage the thread.

▶ Before connecting the connector, check that there is an undamaged O-ring in the socket.

▶ Observe the coding of the connectors during installation.

▶ Cover unused connections with protective covers. Tightening torque: 1.3 ± 0.1 Nm

For UL applications:



**CAUTION**

Short circuit bewtween the sensor inputs of the IO-Link ports

▷ Fire Hazard / Danger of electric shock

▶ Do not interconnect terminations.

▶ To connect the device and the IO-Link devices, only use UL-certified cables belonging to category CYJV or PVVA with a minimum temperature of 80 °C (75 °C in case of a maximum ambient temperature of 40 °C).

## 6.2 Ethernet ports

The device is connected to the EtherNet/IP network via the Ethernet ports (e. g. EtherNet/IP controller, additional EtherNet/IP device).

Wiring:




1: TX +
2: RX +
3: TX -
4: RX -
5: not used

▶ Connect the device to the EtherNet/IP network via the X21, X22 ports.

▶ For connection, use M12 connectors (minimum protection rating: IP 65 / IP 66 / IP 67 / IP 69K).

▶ Tighten the cable plug using 1.3 ± 0.1 Nm.

## 6.3 IO-Link ports (class A)

IO-Link devices (sensors, actuators) are connected to the IO-Link ports (class B).

The IO-Link ports meet the requirements of the IO-Link specifications 1.0 to 1.1.2.

The ports are equipped with short-circuit detection (US).

The ports have an additional digital input (type 2, according to IEC 61131-2).

Wiring:




1: Sensor supply (US) L+
2: Digital input
3: Sensor supply (US) L-
4: C/Q IO-Link
5: not used

### 6.3.1 Connecting IO-Link devices (class A)

Wiring information:

- The connected IO-Link devices must be supplied exclusively via the IO-Link master.

▶ Connect the IO-Link devices to ports X01...X08. Maximum cable length per port: 20 m.

▶ For connection, use M12 connectors (minimum protection rating: IP 65 / IP 66 / IP 67 / IP 69K).

▶ Tighten the cable plug using 1.3 ± 0.1 Nm.

### 6.3.2 Connecting IO-Link devices (Class B)

Wiring information:

- The connection of IO-Link devices (Class B) requires the supply with an additional auxiliary voltage UA. Feeding is done with the help of a Y-connection cable (→ accessories).

**CAUTION**

Non-compliance with the electrical separation of the circuits

▷ Risk of fire

▶ Ensure that the external supply UA is electrically separated from the power circuit of the unit by basic insulation (according to IEC 61010-1, secondary current circuit with max. 30 V DC, derived from mains circuit up to 300 V of overvoltage category II).

▶ Ensure that the IO-Link devices and the connection technology support the electrical separation.

- When connecting an IO-Link device (Class B), the additional digital input at pin 2 of the port is not available.

▶ Connect IO-Link devices via a Y-connection cable with ports X01...X08.

▶ Connect Y-connection cable to the supply voltage 24 V DC (20…30 V SELV/PELV).

▶ For connection, use M12 connectors (minimum protection rating: IP 65 / IP 66 / IP 67 / IP 69K).

▶ Tighten the cable plug using 1.3 ± 0.1 Nm.

## 6.4 Ground connection

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

- Upper mounting lug of the housing

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 device via the mounting screw of the upper mounting lug.

# 7 Operating and display elements

## 7.1 LEDs

### 7.1.1 Status

| LED | Description | Colour | State | Description |
|---|---|---|---|---|
| RDY | Gateway status | -- | Off | Not active or reboots |
| | | green | flashes (5 Hz) | Error |
| | | | flashes (200 ms on, 800 ms off) | Firmware update running |
| | | | On | OK |
| NET | Network status | -- | off | Not powered, no IP address: device has no IP address or no voltage applied or voltage applied is too low |
| | | green / red | flashes | Self test: device in self-test after start (→ MOD-LED) |
| | | green | flashes (1 Hz) | No connection: no CIP connection established and no Exclusive Owner connection timeout |
| | | | on | Connected: device has IP address and at least one CIP connection established and no Exclusive Owner connection timeout |
| | | red | flashes (1 Hz) | Connection timeout: device has IP address and Exclusive Owner connection timeout |
| | | | on | Duplicate: IP address used twice |
| MOD | EtherNet/IP module status | -- | off | No power: no voltage is applied or the applied voltage is too low |
| | | green / red | flashes | Self test: device in self-test |
| | | green | flashes (2 Hz) | Standby: device not yet configured (no IP address) |
| | | | on | Device operational: device functions reliably (normal operation) |
| | | red | flashes (1 Hz) | Major recoverable fault: serious, recoverable error (e.g. incorrect configuration) |
| | | | on | Major unrecoverable fault: serious, unrecoverable error (e.g. module failed) |

### 7.1.2 Ethernet

| LED | Description | Colour | State | Description |
|---|---|---|---|---|
| LNK | Status of the connection | Green | Off | No Ethernet connection |
| | | | On | Ethernet connection established |
| ACT | Status of the transmission | Yellow | Off | No data transmission |
| | | | Flashes | Data transmission |

### 7.1.3 IoT

| LED | Description | Colour | State | Description |
|---|---|---|---|---|
| LNK | Status of the connection | Green | Off | No Ethernet connection |
| | | | On | Ethernet connection established |

| LED | Description | Colour | State | Description |
|---|---|---|---|---|
| IoT | Device identification | Green | Flashes | Device identification active |
| ACT | Status of the data transmission | Yellow | Off | No data transmission |
| | | | Flashes | Data transmission |

### 7.1.4 IO-Link ports (Class A)

| LED | Description | Colour | State | Description |
|---|---|---|---|---|
| | Status of the IO-Link port (pin 4) | Yellow | Off | Port mode DI / DO: Pin 4 (C/Q) = OFF |
| | | | On | Port mode DI / DO: Pin 4 (C/Q) = ON |
| | | Green | Flashes (1 Hz) | Port mode IO-Link: No IO-Link device found |
| | | | Flashes (5 Hz) | Port mode IO-Link: “PREOPERATE” state |
| | | | On | Port mode IO-Link: “OPERATE” state |
| | | Red | Flashes (1.2 Hz) | IO-Link communication error |
| | | | On | Port configuration error or short circuit / overload at US |
| DI | Status of the digital input (pin 2) | Yellow | Off | Digital input: Pin 2 = OFF |
| | | | Flashes (5 Hz) | Digital input: Pin 2 = ON |

### 7.1.5 Voltage supply

| LED | Description | Colour | State | Description |
|---|---|---|---|---|
| US | Status of the supply voltage US | Green | Off | No supply voltage is applied or the applied voltage is too low |
| | | | On | Supply voltage applied |

# 8 Set-up

▶ Install the unit correctly.

▶ Establish a correct electrical connection with the device.

▷ Once connected to the supply voltage, the unit will start.

▷ The LEDs show status and error conditions.

▷ The unit is ready for operation.

▷ The device can be configured.

# 9 Settings

## 9.1 Parameter setting software

### 9.1.1 Supported parameter setting software

For parameterising the IO-Link master and the connected IO-Link devices a parameter setting software is required.

Supported parameter setting software

- ifm moneo OS (art no. QMS001) + module licence ifm moneo|configure (art. no. QMP020)
- ifm moneo|configure free

▶ For a correct installation: observe the installation instruction for ifm moneo.

▶ Install the desired parameter setting software.

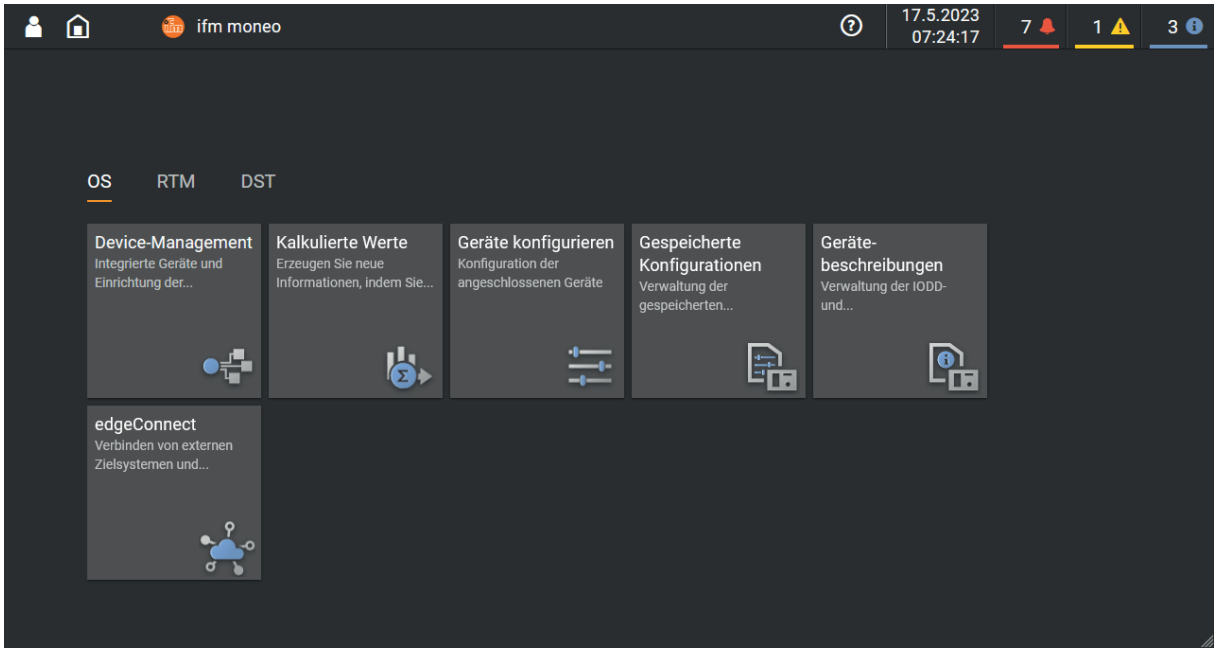
▶ Activate the licences required for operation.

▷ The installed parameter setting software can be used for parameter setting of the device.

### 9.1.2 Getting started

Requirements:

✓ The laptop / PC is connected directly or via a suitable network coupling element (e.g. switch) to an IoT port (X23) of the device.




▷ View of [connected devices] appears.

▶ Scan the network for devices.

▷ ifm moneo detects the IO-Link master.

▶ Optional: Adapt the IP settings of the IO-Link master.

#### 9.1.2.1 Online parameter setting

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 you change parameters 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 connected IO-Link devices stay in operating mode. They continue to perform their functions with the existing parameters until the parameter setting has been completed.

#### 9.1.2.2 Offline parameter setting

The device supports offline parameter setting. The user creates a configuration for the IO-Link master and the connected IO-Link devices without being connected to the device. The configuration created can be stored as a file (*.lrp) and loaded to the device and activated at a later point.

### 9.1.3 Configuring IP settings

For communication with the IT network, the IP parameters of the IoT interface must be configured.

Available parameters:

| Parameter | Description | Value range | Access |
|---|---|---|---|
| [DHCP] | Enable / disable the DHCP client | • [Static IP]: DHCP client disabled; IP parameters are set by the user<br>• [DHCP]: DHCP client enabled; IP parameters are set by the network's DHCP server. | rw [1] |
| [IP address] | IP address of the IoT interface | e.g. 192.168.1.100<br>• 169.254.x.x(default) | rw [1] [2] |
| [Subnet mask] | Subnet mask of the Ethernet network | e.g. 255.255.255.0<br>• 255.255.0.0 (Default) | rw [1] [2] |
| [Default gateway IP address] | IP address of the network gateway | e.g. 192.169.1.1<br>• 0.0.0.0 (Default) | rw [1] [2] |
| [MAC address] | MAC address of the IoT interface | e.g. 00:02:01:0E:66:C1 | ro [3] |

[1] Read and write

[2] can only be changed if DHCP is disabled

[3] Read only

To configure the IoT interface:

Requirements:

✓ The parameter setting software has been started.

✓ The connection to the device has been established.

✓ The detailed view of the device is active.

▶ Select [IoT] menu.

▷ The menu page shows the current settings.

▶ Set IP parameters.

▶ Write the changed values to the device.

▷ IP settings of the IoT interface are configured.

### 9.1.4 IoT interface

The [IoT] menu provides access to the parameters of the IoT interface.

#### 9.1.4.1 Configuring access rights

The device can be integrated in parallel into a fieldbus environment and into an IT network structure for IIoT applications (“Y-path”). To avoid competing access to parameters and process data, the rights of the instances must be set.

Available parameters:

| Parameter | Description | Value range | Access |
|---|---|---|---|
| [Access rights] | Access rights to parameters, process data and alarms / events | • [Fieldbus + IoT]:<br>– Fieldbus and ifm IoT Core have read and write access rights to parameters and process data<br>– Fieldbus and ifm IoT Core have read access rights to events / alarms<br>• [Fieldbus + IoT (read only)]:<br>– Fieldbus has read and write access rights to parameters and process data<br>– Fieldbus has read access rights to events / alarms<br>– ifm IoT Core has read access rights to parameters, process data and events / alarms<br>• [IoT only]:<br>– ifm IoT Core has read and write access rights to parameters and process data<br>– ifm IoT Core has read access rights to events / alarms<br>– Fieldbus has no access rights | rw [1] |

[1] Read and write

When setting the parameters of the access rights in the parameter setting software, observe the following notes:

- If the access rights in the fieldbus projection software and ifm moneo|configure are set to the [Fieldbus + IoT] value, the parameter values set on the fieldbus side always apply.
- For exclusive access to the device via ifm moneo|configure, set the access rights to the [IoT only] value and set the access rights to [Keep settings] in the fieldbus projection software.
- If the access rights in have been set to the [Fieldbus + IoT (read only)] value, the access rights in ifm moneo|configure can no longer be changed. To regain write access with ifm moneo|configure, set the access rights in the fieldbus projection software to the [Fieldbus + IoT] value.

To configure the access rights:

Requirements:

✓ The parameter setting software has been started.

✓ The connection to the device has been established.

▶ Select the [IoT] menu.

▷ The menu page shows the current settings.

▶ Set the access rights.

▶ Write the changed values to the device.

▶ Restart the device.

▷ The changed access rights are active.

#### 9.1.4.2 Configuring IP settings

For communication with the IT network, the IP parameters of the IoT interface must be configured.

Available parameters:

| Parameter | Description | Value range | Access |
|---|---|---|---|
| [DHCP] | Enable / disable the DHCP client | • [Static IP]: DHCP client disabled; IP parameters are set by the user<br>• [DHCP]: DHCP client enabled; IP parameters are set by the network's DHCP server. | rw [1] |
| [IP address] | IP address of the IoT interface | e.g. 192.168.1.100<br>• 169.254.x.x(default) | rw [1] [2] |
| [Subnet mask] | Subnet mask of the Ethernet network | e.g. 255.255.255.0<br>• 255.255.0.0 (Default) | rw [1] [2] |
| [Default gateway IP address] | IP address of the network gateway | e.g. 192.169.1.1<br>• 0.0.0.0 (Default) | rw [1] [2] |
| [MAC address] | MAC address of the IoT interface | e.g. 00:02:01:0E:66:C1 | ro [3] |

[1] Read and write

[2] can only be changed if DHCP is disabled

[3] Read only

To configure the IoT interface:

Requirements:

✓ The parameter setting software has been started.

✓ The connection to the device has been established.

✓ The detailed view of the device is active.

▶ Select [IoT] menu.

▷ The menu page shows the current settings.

▶ Set IP parameters.

▶ Write the changed values to the device.

▷ IP settings of the IoT interface are configured.

#### 9.1.4.3 Configuring the interface to the monitoring software

To enable transmission of the process data of the IO-Link master to the monitoring software (e.g. ifm moneo suite), the interface must be configured accordingly.

Available parameters:

| Name | Description | Value range | Access |
|---|---|---|---|
| [IP address LR Agent or SMARTOBSERVER] | IP address of the monitoring software | e.g. 255.255.255.255 (default) | rw [1] |
| [Port LR Agent or SMARTOBSERVER] | Port number of the monitoring software | • 0: 0<br>…<br>• 35100: 35100 (default)<br>…<br>• 65535: 65535 | rw [1] |
| [Interval LR Agent or SMARTOBSERVER] | Interval for data transmission to the monitoring software (value in milliseconds) | • [Disabled]: No transmission (default)<br>• 500: 500 ms<br>…<br>• 2147483647: 2147483647 ms | rw [1] |
| [Application tag] | Source identifier of the device in the monitoring software | e.g., factory 2 plant 1 | rw [1] |

[1] Read and write

To set the interface to the monitoring software:

After changing the parameters [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:

▶ When the parameter has been changed, restart the IO-Link master.

Requirements:

✓ The parameter setting software has been started.

✓ The connection to the device has been established.

✓ The detailed view of the device is active.

▶ Select the [IoT] menu.

▷ The menu page shows the current settings.

▶ Set the parameters of the interface.

▶ Write the changed values to the device.

▷ Interface to the monitoring software has been set.

## 9.1.5 Fieldbus interface

The [Fieldbus] menu provides access to the parameters of the EtherNet/IP interface.

### 9.1.5.1 Configuring the EtherNet/IP interface

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

Available parameters:

| Name | Description | Value range | Access |
|---|---|---|---|
| [DHCP] | Enable / disable the DHCP client of the device | • [Static IP]: IP parameters are set by the user<br>• [DHCP]: IP parameters are set by a DHCP server in the network<br>• [BOOTP]: IP parameters are set via Bootstrap Protocol (BOOTP) | rw [1] |
| [IP address] | IP address of the EtherNet/IP interface | e.g. 192.168.82.100<br>192.168.1.250 (default) | rw [1] |
| [Subnet mask] | Subnet mask of the EtherNet/IP network | e.g. 255.255.0.0<br>• 255.255.255.0 (default) | rw [1] |
| [Default gateway IP address] | IP address of the network gateway | e.g. 192.168.82.1<br>• 0.0.0.0 (default) | rw [1] |
| [Host name] | EtherNet/IP name of the device (max. 240 characters) | e.g. iol12.plant1 | rw [1] |
| [MAC address] | MAC address of the device's EtherNet/IP interface | e.g. 00:02:01:0f:c8:8f | ro [2] |
| [Fieldbus firmware] | Version of the fieldbus firmware | e.g. 5.3.0.10 (EtherNet/IP Adapter) | ro [2] |

[1] Read and write

[2] Read only

Requirements:

✓ The parameter setting software has been started.

✓ The connection to the device has been established.

✓ The detailed view of the device is active.

▶ Select the [Fieldbus] menu.

▷ The menu page shows the current settings.

▶ Set the parameters of the EtherNet/IP interface.

▶ Save the changes on the device.

▷ IP settings of the interface are configured.

#### 9.1.5.2 Setting the EtherNet/IP configuration mode

The device supports the fieldbus configuration modes "top-down" and "independent". For the configuration mode "Independent", the user can additionally select the connection type.

Available parameters:

| Parameter | Description | Value range | Access |
|---|---|---|---|
| [Independent mode] | EtherNet/IP configuration mode | • [Independent mode off]: Configuration via fieldbus PLC<br>• [Independent mode on]: Configuration via the device | rw [1] [2] |
| [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<br>• [4 bytes input 4 bytes output]: 4 bytes input data, 4 bytes output data<br>• [8 bytes input 8 bytes output]: 8 bytes input data, 8 bytes output data<br>• [16 bytes input 16 bytes output]: 16 bytes input data, 16 bytes output data<br>• [32 bytes input 32 bytes output]: 32 bytes input data, 32 bytes output data | rw [1] [2] |
| [Swap] | Swap the order of the bytes in the data word of the process data | • [off]: as Array of Bytes<br>• [on]: as Integer16 value; during an update of the process data, the bytes are exchanged | rw [1] [2] |

[1] Read and write

[2] Read and write; parameter can only be changed if no active connection to fieldbus controller

Requirements:

✓ The parameter setting software has been started.

✓ The connection to the device has been established.

✓ The detailed view of the device is active.

▶ Select the [Fieldbus] menu.

▷ The menu page shows the current settings.

▶ Set the configuration mode.

▶ Optional: Select connection type.

▶ Set arrangement of the bytes in the process data.

▶ Save the changes on the device.

▷ EtherNet/IP configuration mode is set.

### 9.1.6 IO-Link ports

The [Port x](x:1...8) menues provide access to the parameters of the IO-Link portsX01...X08.

#### 9.1.6.1 Setting data transmission to the monitoring software

Available parameters:

| Name | Description | Value range | Access |
|---|---|---|---|
| [Transmission to LR Agent or SMARTOBSERVER] | Enable / Disable the transmission of process data of the port to the monitoring software | • [Disabled]: Process data is not transmitted<br>• [Enabled]: Process data is transmitted | rw [1] |

[1] Read and write

To set the data transmission:

Requirements:

✓ The parameter setting software has been started.

✓ The connection to the device has been established.

✓ The detailed view of the device is active.

✓ Interface to the monitoring software has been configured.

▶ Select the [Port x] (x = 1…8) menu.

▷ The menu page shows the current settings.

▶ Set the parameters.

▶ Write the changed values to the device.

▷ Data transmission to the monitoring software has been set.

#### 9.1.6.2 Setting the operating mode Pin 4 (US)

Available parameters:

| Parameter | Description | Value range | Access |
|---|---|---|---|
| [Mode Pin4 US] | Operating mode of Pin 4 (C/Q) of the port | • [Disabled]: Port deactivated<br>• [DI]: Digital input<br>• [DO]: Digital output<br>• [IO-Link]: IO-Link process data | rw [1] |

[1] Read and write

Requirements:

✓ The parameter setting software has been started.

✓ The connection to the device has been established.

✓ The detailed view of the device is active.

▶ Select the [Port x] (x = 1…8) menu.

▷ The menu page shows the current settings.

▶ Set the operating mode.

▶ Save the changes on the device.

▷ Operating mode of pin 4 (US) of the port has been set.

#### 9.1.6.3 Setting the cycle time

In the IO-Link operating mode, the cycle time of the port can also be set.

Available parameters:

| Parameter | Description | Value range | Access |
|---|---|---|---|
| [Cycle time actual] | Current cycle time between the IO-Link master and the IO-Link device (value in µs) | • 0: best possible cycle time<br>…<br>• 132800: 132800 µs | ro [1] [2] |

| Parameter | Description | Value range | Access |
|---|---|---|---|
| [Cycle time preset] | Cycle time between the IO-Link master and the IO-Link device (value in µs) | • 0: best possible cycle time<br>• 1: 1 µs<br>…<br>• 132800: 132800 µs | rw [3] [2] |
| [Bit rate] | Transmission rate between the IO-Link master and the IO-Link device | • COM1: 4.8 kbaud<br>• COM2: 38.4 kbaud<br>• COM3: 230.4kbaud | ro [1] [2] |

[1] Read only

[2] Parameter only available if operating mode pin 4 (US) = IO-Link and IO-Link device is connected to the port

[3] Read and write

Requirements:

✓ The parameter setting software has been started.

✓ The connection to the device has been established.

✓ The detailed view of the device is active.

✓ Operating mode of pin 4 of the port is "IO-Link".

▶ Select the [Port x] > [IO-Link] menu (x = 1…8).

▶ Set the cycle time.

▶ Write the changed values to the device.

▷ Cycle time of the IO-Link port is set.

#### 9.1.6.4 Setting the device validation and data storage

The device supports the functions device validation and backup / restore of parameter data of the connected IO-Link devices.

The following options are available:

| Option | Validation of the connected IO-Link device | Backup of parameter values | Restore of parameter values |
|---|---|---|---|
| [No check and clear] | No | No | No |
| [Type compatible V1.0 device] | Yes, test compatibility with IO-Link standard V1.0 | No | No |
| [Type compatible V1.1 device] | Yes, test compatibility with IO-Link standard V1.1 | No | No |
| [Type compatible V1.1 device with Backup + Restore] | Yes, test compatibility with IO-Link standard V1.1 and identity of design (vendor ID, device ID) | Yes, automatic backup of parameter values; Changes to the current parameter values will be stored | Yes, restore of parameter values when connecting an identical IO-Link device with factory settings |
| [Type compatible V1.1 device with Restore] | Yes, test compatibility with IO-Link standard V1.1 and identity of design (vendor ID, device ID) | No, there is no automatic backup; Changes to the parameter values are not stored | Yes, restore of parameter values when connecting an identical IO-Link device with factory settings |

Device validation and backup / restore of parameter values are only available if the “IO-Link” operating mode of the port has been activated.

For options [Type compatible V1.1 device with Backup + Restore] and [Type compatible V1.1 device with Restore]: If the parameters [Vendor ID] or [Device ID] are changed in online mode, the data memory is deleted and the parameter values of the connected IO-Link device are backed-up again in the IO-Link master.

Available parameters:

| Parameter | Description | Value range | Access |
|---|---|---|---|
| Validation / Data Storage | Validation of the connected IO-Link devices and automatic restore of parameter sets of the IO-Link device | • No check and clear<br>• Type compatible V1.0 device<br>• Type compatible V1.1 device<br>• Type compatible V1.1 device with Backup + Restore<br>• Type compatible V1.1 device with Restore | rw [1] [2] |
| Vendor ID | Vendor ID of the IO-Link device to be validated against | e.g., 310: ifm electronic gmbh | rw [1] [2] |
| Device ID | Device ID of the IO-Link device to be validated against | e.g., 1129: TCC501 (ifm temperature sensor) | rw [1] [2] |

[1] Read and write

[2] Parameter only available if operating mode pin 4 (US) = IO-Link and IO-Link device is connected to the port

To configure the device validation and the backup / restore of parameter values:

Requirements:

✓ The parameter setting software has been started.

✓ The connection to the device has been established.

✓ The detailed view of the device is active.

✓ Operating mode of pin 4 of the port is "IO-Link".

▶ Select the [Port x] > [IO-Link] menu (x = 1…8).

▷ The menu page shows the current settings.

▶ Set the parameters for device validation and backup / restore of parameter values.

▶ Optional: Enter the vendor ID and device ID.

▶ Save the changes on the device.

▷ Device validation and backup / restore of parameter values have been configured.

#### 9.1.6.5 Setting fail-safe values

The user can set the fail-safe values of the outputs of the IO-Link ports for the "Independent" configuration mode. The fail-safe values will only be enabled if the EtherNet/IP connection is interrupted.

Available parameters:

| Parameter | Description | Value range | Access |
|---|---|---|---|
| [Fail-safe digital out] | Fail-safe value of the output for operating mode "digital output (DO)" | • [Reset]: default value<br>• [Old]: most recent process value<br>• [Set]: HIGH level | rw [1] [2] |
| [Fail-safe IO-Link] | Fail-safe value of the output for "IO-Link" operating mode | • [Off]: Fail-safe deactivated<br>• [Reset]: Fail-safe activated – default value<br>• [Old]: Fail-safe activated – most recent process value<br>• [Pattern]: Fail-safe activated - byte sequence | rw [1] [2] |

[1] Read and write

[2] Read and write; parameter can only be changed if no active connection to fieldbus controller

Requirements:

✓ The parameter setting software has been started.

✓ The connection to the device has been established.

✓ The detailed view of the device is active.

✓ Configuration mode “Independent” is enabled.

▶ Select the [Port x] (x = 1…8) menu.
  ▷ The menu page shows the current settings.
▶ Setting fail-safe values
▶ Write the changed values to the device.
▷ Fail-safe values are configured.

## 9.1.7 Device information

The [Info] menu provides access to the identification information of the device.

### 9.1.7.1 Reading identification information

Available information:

To read the device information:

Requirements:

✓ The parameter setting software has been started.
✓ The connection to the device has been established.
✓ The detailed view of the device is active.
▶ Select [Info] menu.
▷ The menu page shows the device information.

## 9.1.8 Device control

The [Firmware] menu provides access to functions for controlling the device.

### 9.1.8.1 Resetting the device

Requirements:

✓ The parameter setting software has been started.
✓ The connection to the device has been established.
✓ The detailed view of the device is active.
▶ Select the [Firmware] menu.
▶ Click on [Factory reset].
▷ The device is reset to factory settings.
▷ All parameters are set to their default values.
▷ The device is read again.

### 9.1.8.2 Restarting the device

Requirements:

✓ The parameter setting software has been started.
✓ The connection to the device has been established.
✓ The detailed view of the device is active.
▶ Select the [Firmware] menu.
▶ Click on [Reboot].
▷ The device is restarted.

▷ All set parameter values will be retained.

### 9.1.9 Updating the firmware

To update the firmware of the device:

Requirements:

✓ New firmware has been downloaded. documentation.ifm.com

✓ The parameter setting software has been started.

✓ The connection to the device has been established.

✓ The detailed view of the device is active.

▶ In the header: Click on .

▷ A dialogue window appears.

▶ Click on [Import firmware file].

▷ The file explorer appears.

▶ Select the firmware file (.bin) and click on [Open].

▷ The new firmware file is selected.

▶ Click on [OK].

▷ The firmware of the device is updated.

▷ The status of the update process is displayed.

▷ The firmware of the device has been updated.

### 9.1.10 Setting the parameters of IO-Link devices

The device supports access to connected IO-Link devices (sensors, actuators).

Information about the parameters of the IO-Link device:

→ Operating instructions IO-Link device

→ IODD description of the IO-Link device

Requirements:

✓ ifm moneo|configure has been started.

✓ Operating mode pin 4 (US) of the port with the IO-Link device is „IO-Link“.

✓ IoT has read and write access rights for the IO-Link master.

✓ The IO-Link device is properly connected to the IO-Link port of the device.

✓ The IODD library of ifm moeno|configure contains the IODD of the IO-Link device.

▶ Scan the network for devices.

▷ ifm moneo|configure detects the IO-Link master.

▶ Estblish connection to the IO-Link master.

▷ ifm moneo|configure detects the connected IO-Link device.

▶ In the row of the IO-Link device: Click on .

▷ The detailed view of the IO-Link device appears.

▷ The detailed view shows the current parameter values of the IO-Link device.

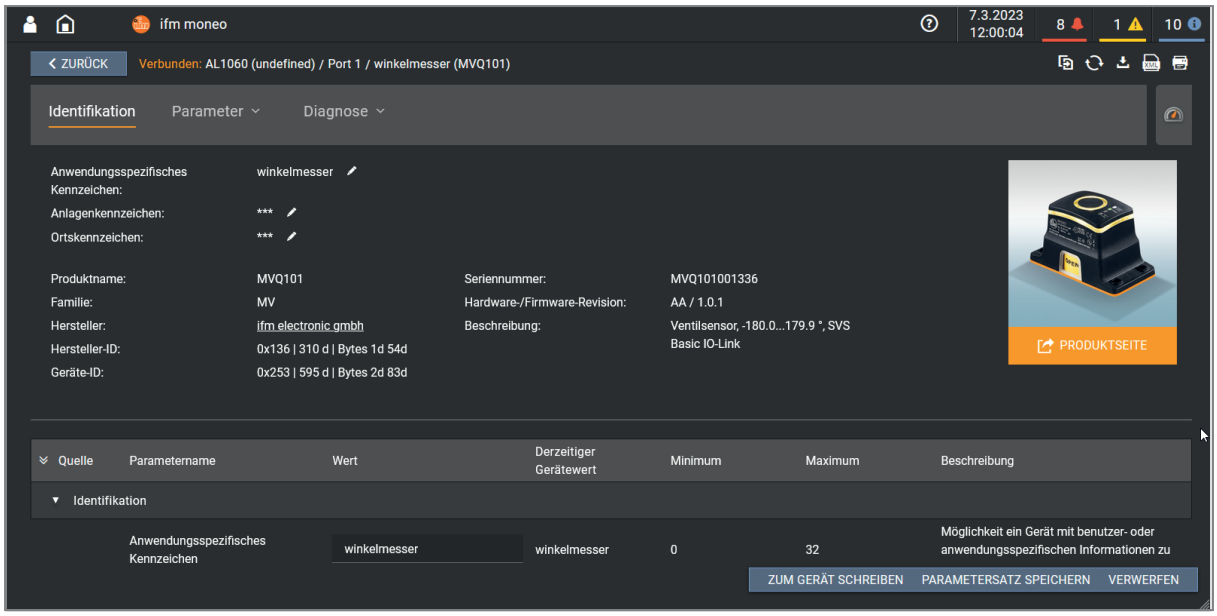



Fig. 1: Example: Detailed view of the MVQ101 sensor

▶ Configure the IO-Link device.

▶ Store the chenged configuration on the IO-Link device.

▷ The parameters of the IO-Link device have been set.

# 9.2 ifm IoT Core REST API

## 9.2.1 ifm IoT Core: General information

The device has the ifm IoT Core. The ifm IoT Core represents the functionality of the device. The functionality consists of a set of data, services and events. The elements of the ifm IoT Core are arranged in a JSON object in a hierarchical tree structure. The ifm IoT Core makes these elements available to the outside world via standard interfaces (e.g. HTTP) . This allows the user and other devices to access data (parameters, process data), events and functions (services) of the device.

### 9.2.1.1 Accessing the ifm IoT Core

Any element of the ifm IoT Core is accessed via its address (e.g. `root/port1/pin2`). The address is composed of the path leading to the element ( `root/port1` ) and the identifier of the element ( `pin2` ).

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

## 9.2.2 Getting started

To read the device description:

▶ Send the following POST request to the ifm IoT Core:
`{"code":"request","cid":-1,"adr":"gettree"}`

▷ The ifm IoT Core returns the device description as a 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

The device has the type “device” (→ Types 🗋 72). The following services can be applied to the root element of the device tree:

| Service | Description |
|---|---|
| ../gettree | Provide the complete tree or subtree of the device description (JSON) |
| ../getidentity | Read identification information of the unit |
| ../getdatamulti | Read multiple elements sequentially |
| ../getelementinfo | Read detailed information of an element |
| ../getsubscriberlist | Provide a list of all active notification subscriptions |
| ../querytree | Search device description for specific elements |

The following services can be applied to elements of the type `data` depending on the configured access rights:

| Service | Description |
|---|---|
| ../getdata | Read the value of the element |
| ../setdata | Write the value of the element |

### 9.2.3.1 Example: Reading properties of an element

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",
    "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".

#### 9.2.3.2 Example: Outputting the subtree

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 display: `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
}
```

#### 9.2.3.3 Example: Changing a parameter value

Task: The Application Tag parameter of the device is to be written with the value "Do not use". The new value is only supposed to be valid until the next reboot of the device.

Solution: Write the new value of the `/devicetag/applicationtag` element with the `setdata` service. To keep the new value only until the next restart of the unit, pass on the duration option with the uptime value.

Request:

```
{
  "code":"request",
  "cid":4711,
  "adr":"/devicetag/applicationtag/setdata",
  "data":{
    "duration":"uptime",
    "newvalue":"Do not use"
  }
}
```

Response:

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

#### 9.2.3.4 Example: Read multiple elements sequentially

Task: The following current values of the device are to be read consecutively: temperature, serial number

Solution: Read the current parameter values using the service `getdatamulti` (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
}
```

#### 9.2.3.5 Example: Browsing device description

Task: All elements with the "timer" profile are to be listed.

Solution: Use the `querytree` service to search the device description with parameter "timer" (profile).

Request:

```
{
  "cid":4711,
  "code":"request",
  "adr":"querytree",
  "data":{
    "profile":"timer"
  }
}
```

Response:

```
{
  "cid":4711,
  "data":{
    "adrList":[
      "device/timer[1]",
      "device/timer[2]"
    ]
  },
  "code": 200
}
```

### 9.2.4 IoT: Configuring access rights

The device can be integrated in parallel into a fieldbus environment and into an IT network structure for IIoT applications (“Y-path”).

Changes to the Gateway: Resetting, rebooting and localising the device (→ 🗋 54) data point will only take effect after the device has been restarted.

Substructure: iotsetup

Available data points:

When parameterising the access rights with the ifm IoT Core, observe the following notes:

- If the access rights in the fieldbus projection software and the parameter setting software are set to the [Fieldbus + IoT] value, the parameter values set on the fieldbus side always apply.
- For exclusive access to the device via the parameter setting software, set the access rights to the [IoT only] value and set the access rights to [Keep settings] in the fieldbus projection software.
- If the access rights in the ifm IoT Core are set to the value [Fieldbus + IoT (read only)], the access rights can no longer be changed via the ifm IoT Core. To regain write access with the parameter setting software, set the access rights in the fieldbus projection software to the [Fieldbus + IoT] value.

### 9.2.5 IoT: Configuring IP settings

Substructure: iotsetup/network

Available data points:

| Parameter | Description | Value range | Access |
|---|---|---|---|
| ../dhcp | Activate / deactivate DHCP mode | • static ip: DHCP client disabled; IP parameters are set by the user<br>• dhcp: DHCP client enabled; IP parameters are set by the network's DHCP server. | rw [1] |
| ../ipaddress | IP address of the IoT interface | e.g. 192.168.1.100<br>• 169.254.x.x (default) | rw [1] |
| ../subnetmask | Subnet mask of the network segment | e.g. 255.255.255.0<br>• 255.255.0.0 (default) | rw [1] |
| ../ipdefaultgateway | IP address of the network gateway | e.g. 192.169.1.1<br>• 0.0.0.0 (default) | rw [1] |

[1] Read and write

Applicable services:

| Name | Description |
|---|---|
| ../setblock | write all data points of the substructure blockwise |

Write the data points of the substructure only with the `setblock` service!

### 9.2.6 IoT: Configuring security mode

The unit provides a security mode. The security mode makes it possible to protect access to IO-Link masters and connected IO-Link devices from the IT network. When security mode is activated, the following restrictions apply:

- Access only by authentication (password protection)
- Access only via encrypted connection (TLS - Transport Layer Security)

The security mode only protects access and data transmission via the IoT interface.

▷ The user name `administrator` cannot be changed.

▷ The set password cannot be read with getdata.

The current status of the security mode can be read with the getidentity service (→ Service: getidentity 🗋 74).

For the authentication, the user must additionally provide the POST requests with a valid user name and password in the field "auth". The user name and the password are shown as Base64-coded character strings.

The following requests can be done if the security mode is enabled, also without authentication:

- `getidentity`
- `deviceinfo/vendor/getdata`
- `deviceinfo/productcode/getdata`

Substructure: `iotsetup`

Available data points:

| Name | Description | Value range | Access |
|---|---|---|---|
| ../security/securitymode | Enabling / disabling security mode | • disabled: Security mode deactivated<br>• enabled: Security mode activated | rw [1] |
| ../security/password | Password for authentication (Base64 coded) | | wo [2] |

[1] Read and write

[2] write only

Valid character set for the Base64 coding / decoding of the password: UTF-8

Online tool for coding / decoding: www.base64encode.org

#### 9.2.6.1 Example: Activating security mode

Task: The security mode of the IoT interface is to be activated. The valid password is to be set to "password" (BASE64-coded: cGFzc3dvcmQ=).

Solution: Activating the security mode with data point `iotsetup/security/securitymode`. Writing the password to the data point `iotsetup/security/password`.

**Activating security mode**

Request:

```
{
  "code":"request",
  "cid":1,
  "adr":"/iotsetup/security/securitymode/setdata",
  "data":{"newvalue":1}
}
```

Response:

```
{
  "cid":-1,
  "code":200
}
```

**Setting the password**

Request:

```
{
  "code":"request",
  "cid":2,
  "adr":"/iotsetup/security/password/setdata",
  "data":{"newvalue":"cGFzc3dvcmQ="}
}
```

Response:

```
{
  "cid":2,
  "code":200
}
```

#### 9.2.6.2 Example: Request with authentication

Task: The temperature of the IO-Link master is to be read. The security function is enabled (current password: password).

Solution: Read the data point processdatamaster/temperature. The request must be sent using https. The user name and the password are transferred as a Base64-coded character string ("administrator" = "YWRtaW5pc3RyYXRvcg==", "password" = "cGFzc3dvcmQ=")

Request:

```
{
  "code":"request",
  "cid":1,
  "adr":"processdatamaster/temperature/getdata",
  "auth":{
    "user":"YWRtaW5pc3RyYXRvcg==",
    "passwd":"cGFzc3dvcmQ="
  }
}
```

Response:

```
{
  "cid":1,
  "data":{"value":37},
  "code":200
}
```

#### 9.2.6.3 Example: Resetting the password

Task: The existing password is to be reset.

Solution: To reset a password, disable the security mode. To disable it, enter the user name and the password (the fields "user" and "passwd").

Request:

```
{
  "code":"request",
  "cid":-1,
  "adr":"iotsetup/security/securitymode/setdata",
  "data":{"newvalue":0},
  "auth":{
    "user":"YWRtaW5pc3RyYXRvcg==",
    "passwd":"SW9UNGlmbQ=="
  }
}
```

Response:

```
{
  "cid":-1,
  "code":200
}
```

### 9.2.7 IoT: Configuring the interface to the monitoring software

To enable transmission of the process data of the IO-Link master to the monitoring software (e.g. ifm moneo suite), the interface must be configured accordingly.

Substructure: iotsetup

Available data points:

| Name | Description | Value range | Access |
|---|---|---|---|
| ../smobip | IP address of the monitoring software | e.g. 255.255.255.255 (default) | rw [1] |

| Name | Description | Value range | Access |
|---|---|---|---|
| ../smobport | Port number of the monitoring software | • 0: 0<br>…<br>• 35100: 35100 (default)<br>…<br>• 65535: 65535 | rw [1] |
| ../smobinterval | Interval for data transmission to the monitoring software (value in milliseconds) | • 0: Off – no transmission (default)<br>• 500: 500 ms<br>…<br>• 2147483647: 2147483647 ms | rw [1] |

[1] Read and write

### 9.2.8 Fieldbus: Configuring the EtherNet/IP interface

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

Substructure: fieldbussetup

Available data points:

| Name | Description | Values | Access |
|---|---|---|---|
| ../network/dhcp | Status of the DHCP client | • 0: Static IP address:<br>• 1: DHCP (Dynamic Host Configuration Protocol)<br>• 2: BOOTP (Bootstrap Protocol) | rw [1] |
| ../network/ipaddress | IP address of the Ethernet interface: | e.g. 192.200.0.100<br>• 192.168.1.250 (Default) | rw [1] [2] |
| ../network/subnetmask | Subnet mask of the network segment | e.g. 255.255.192.0<br>• 255.255.255.0 (Default) | rw [1] [2] |
| ../network/ipdefaultgateway | IP address of the network gateway | e.g. 192.200.63.1<br>• 0.0.0.0 (Default) | rw [1] [2] |
| ../network/macaddress | MAC address of the Ethernet interface | e.g. 00:02:01:0E:10:7C | ro [3] |
| ../hostname | EtherNet/IP hostname | e.g. alxx2x | rw [1] |
| ../fieldbusfirmware | Firmware of the fieldbus | e.g. 5.3.0.10 (EtherNet/IP Adapter) | ro [3] |
| ../connectionstatus | Status of the fieldbus connection | • 0: Not connected<br>• 1: Connected | ro [3] |

[1] Read and write

[2] Read and write; parameter can only be changed if no active connection to fieldbus controller

[3] Read only

Applicable services:

| Name | Description |
|---|---|
| ../network/setblock | Write all values of the substructure blockwise |

▶ Change the parameters in the substructure network only block by block with the service setblock!

### 9.2.9 Fieldbus: Setting the EtherNet/IP configuration mode

The device supports the EtherNet/IP configuration modes "Top-Down Mode" and "Independent Mode".

Substructure: fieldbussetup/configuration

Available data points:

| Name | Description | Value range | Access |
|---|---|---|---|
| ../independentmode | Configuration mode | • 0: Independent mode off<br>• 1: Independent mode on | rw [1] [2] |
| ../explicitmodepd | Connection types (Input Assembly, Output Assembly) | • 0: Explicit Process Data Mode off<br>• 1: Explicit Process Data Mode with IO-Link I/O + Acyclic + Diag<br>• 2: Explicit Process Data Mode with IO-Link I/O + Acyclic<br>• 3: Explicit Process Data Mode with IO-Link I/O | rw [1] [2] |
| ../processdataconfiguration | Length of the process data | • 0: 2 bytes input 2 bytes output<br>• 1: 4 bytes input 4 bytes output<br>• 2: 8 bytes input 8 bytes output<br>• 3: 16 bytes input 16 bytes output<br>• 4: 32 bytes input 32 bytes output | rw [1] [2] |
| ../swap | Swap of the bytes in process data | • 0: Off<br>• 1: On | rw [1] [2] |

[1] Read and write

[2] Read and write; parameter can only be changed if no active connection to fieldbus controller

### 9.2.10 Fieldbus: Setting fail-safe values of the outputs

If the EtherNet/IP connection is interrupted, fail-safe values can be set for the configured outputs of the ports.

Substructure: fieldbussetup/configuration/port[x] (x:1…8)

Available data points:

| Name | Description | Value range | Access |
|---|---|---|---|
| ../failsafedigitalout | Fail-safe value for the digital output - pin 4 (DO) | • 0: Failsafe – Reset value: default value<br>• 1: Failsafe – Old value: most recent process value<br>• 2: Failsafe – Set value: HIGH level | rw [1] [2] |
| ../failsafeiolink | Fail-safe value for IO-Link output data - Pin 4 (IO-Link) | • 0: No failsafe: Fail-safe deactivated<br>• 1: Failsafe – Reset value: Fail-safe activated – default value<br>• 2: Failsafe – Old value: Fail-safe – most recent process value<br>• 3: Failsafe with Pattern: Fail-safe – byte sequence | rw [1] [2] |

[1] Read and write

[2] Read and write; parameter can only be changed if no active connection to fieldbus controller

### 9.2.11 Ports: Setting the operating mode Pin 4 (US)

Pin 4 of ports X01...X08 supports the following operating modes:

- Disabled: No data transmission at pin 4 (C/Q) of the port
- Digital input: Binary input signal at pin 4 (C/Q) of the port
- Digital output Binary output signal at pin 4 (C/Q) of the port
- IO-Link: IO-Link data transmission via pin 4 (C/Q) of the port

The set operating mode must match the operating mode of the device connected to the port (sensor, actuator, IO-Link device).

Substructure: iolinkmaster/port[x] (x: 1…8)

Available data points:

| Name | Description | Value range | Access |
|---|---|---|---|
| …/mode | Operating mode of pin 4 (US) of the port | • 0: Disabled - port deactivated<br>• 1: DI - digital input<br>• 2: DO - digital output<br>• 3: IO-Link - IO-Link data | rw [1] [2] |
| ../mastercycletime_actual | Current cycle time of the data transfer between port and IO-Link device (value in µs) | • 0: best possible cycle time<br>• 1: 1 µs<br>…<br>• 132800: 132800 µs | ro [3] |
| ../comspeed | Current transfer rate between port and IO-Link device | • 0: COM1 - 4.8 kBaud<br>• 1: COM2 - 38.4 kBaud<br>• 2: COM3 - 230.4 kBaud | ro [3] |
| ../mastercycletime_preset | Cycle time of the data transfer between port and IO-Link device (value in µs); only in IO-Link operating mode | • 0: Automatically set the best possible cycle time<br>• 1: 1 µs<br>…<br>• 132800: 132800 µs | rw [1] [2] |

[1] Read and write

[2] Read and write; parameter can only be changed if no active connection to fieldbus controller

[3] Read only

### 9.2.12 Ports: Configuring device validation and data storage

The device supports the functions device validation and backup / restore of parameter data of the connected IO-Link devices.

The following options are available:

| Option | Validation of the connected IO-Link device | Backup of parameter values | Restore of parameter values |
|---|---|---|---|
| [No check and clear] | No | No | No |
| [Type compatible V1.0 device] | Yes, test compatibility with IO-Link standard V1.0 | No | No |
| [Type compatible V1.1 device] | Yes, test compatibility with IO-Link standard V1.1 | No | No |
| [Type compatible V1.1 device with Backup + Restore] | Yes, test compatibility with IO-Link standard V1.1 and identity of design (vendor ID, device ID) | Yes, automatic backup of parameter values; Changes to the current parameter values will be stored | Yes, restore of parameter values when connecting an identical IO-Link device with factory settings |
| [Type compatible V1.1 device with Restore] | Yes, test compatibility with IO-Link standard V1.1 and identity of design (vendor ID, device ID) | No, there is no automatic backup; Changes to the parameter values are not stored | Yes, restore of parameter values when connecting an identical IO-Link device with factory settings |

Device validation and backup / restore of parameter values are only available if the “IO-Link” operating mode of the port has been activated.

For options [Type compatible V1.1 device with Backup + Restore] and [Type compatible V1.1 device with Restore]: If the parameters [Vendor ID] or [Device ID] are changed in online mode, the data memory is deleted and the parameter values of the connected IO-Link device are backed-up again in the IO-Link master.

Substructure: iolinkmaster/port[x] (x: 1…8)

Available data points:

| Name | Description | Value range | Access |
|---|---|---|---|
| ../validation_datastor-age_mode | Behaviour of the port when connecting a new IO-Link device | • 0: No check and clear<br>• 1: Type compatible V1.0 device<br>• 2: Type compatible V1.1 device<br>• 3: Type compatible V1.1 device with Backup + Restore<br>• 4: Type compatible V1.1 device with Restore | rw [1] [2] |
| ../validation_vendorid | IO-Link ID of the manufacturer that is to be validated | 0…655535 | rw [1] [2] |
| ../validation_deviceid | IO-Link ID of the device to be validated | 0…16777215 | rw [1] [2] |
| ../datastorage | Structure for port data storage | - | rw [1] [2] |
| ../datastorage/maxsize | Maximum size of the data storage content (in bytes) | e.g. 2066: 2066 bytes | ro [3] |
| ../datastorage/chunksize | Size of a data segment (in bytes) | e.g. 256: 256 bytes | ro [3] |
| ../datastorage/size | Size of the data storage content (in bytes) | e.g. 1024: 2 bytes | ro [3] |

[1] Read and write

[2] Read and write; parameter can only be changed if no active connection to fieldbus controller

[3] Read only

Applicable services:

| Name | Description |
|---|---|
| ../validation_useconnecteddevice | Validate the IO-Link device |
| ../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 |

#### 9.2.12.1 Example: Cloning the data memory of an IO-Link port

Task: The data memory of port 2 of IO-Link master 1 is to be cloned to IO-Link master 2.

Solution: The cloning consists of 2 steps. In the first step, the data memory of the port is read by IO-Link master 1. In the second step, the read data is stored in the data memory of the port of IO-Link master 2.

**Read data memory**

▶ Read segment size of the data storage (h = number of bytes)

```
{
  "code":"request",
  "cid": -1,
  "adr":"/iolinkmaster/port[2]/datastorage/chunksize/getdata"
}
```

Example: h = 256

▶ Read total size of the data storage (g = number of bytes)

```
{
  "code":"request",
  "cid":-1,
  "adr": "/iolinkmaster/port[2]/datastorage/size/getdata"
}
```

Example: g = 550

▶ Calculate the number of reading steps n: n = first integer value for which applies: g < n*h

Example: n= 3, da 550 < 3*256

▶ Read the data storage segment by segment ("pos" indicates the byte offset at which the read process starts with the length "length").

```
{
  "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 command: pos = 0, length = 256

2nd read command: pos = 256, length =256

3rd read command: pos = 512, length = 256

Each segment is returned as a BASE64-encoded string.

▶ Connect segments (concatenate).

**Restore data memory**

▶ Determine the size of the backed-up data storage content (n = number of bytes).

Example: n = 550

▶ Read segment size (s = number of bytes)

```
{
  "code":"request",
  "cid":-1,
  "adr":"/iolinkmaster/port[1]/datastorage/chunksize/getdata"
}
```

Example: s = 256

▶ Start writing the data storage string segment by segment ("size" = size of the read data storage)

```
{
  "code":"request",
  "cid":-1,
  "adr":"/iolinkmaster/port[1]/datastorage/start_stream_set",
  "data": {"size": n}
}
```

Example: size = 550

▶ Transmit data storage string segment by segment ("value" = string value with length s)

```
{
  "code":"request",
  "cid":-1,
  "adr":"/iolinkmaster/port[1]/datastorage/stream_set",
  "data": {"value": "aWZtfgIAAABBTDF4NXhfY25faXRfdDIuMi43Nw..."
}
```

### 9.2.13 Ports: Setting data transmission to the monitoring software

Substructure: iolinkmaster/port[x] (x: 1…8)

Available data points:

| Name | Description | Value range | Access |
|---|---|---|---|
| ../senddatatosmob | Enable / Disable the transmission of process data of the port to the monitoring software | • 0: Disabled – process data will not be transmitted<br>• 1: Enabled – process data will be transmitted | rw [1] |

[1] Read and write

### 9.2.14 Ports: Reading / writing process data

Substructure: iolinkmaster/port[x] (x: 1…8)

Available data points:

| Name | Description | Value range | Access |
|---|---|---|---|
| ../pin2in | Digital input (pin 2) | • 0: OFF<br>• 1: ON | ro [1] |
| ../iolinkdevice/pdin | Input data (pin 4) | "DI" operating mode:<br>• 0x00: OFF<br>• 0x01: ON<br>"IO-Link" operating mode:<br>• Process value in hexadecimal representation | ro [1] |
| ../iolinkdevice/pdout | Output data (pin 4) | "DO" operating mode:<br>• 0x00: OFF<br>• 0x01: ON<br>"IO-Link" operating mode:<br>• Process value in hexadecimal representation | rw [2] [3] |

[1] Read only

[2] Read and write

[3] Read and write; parameter can only be changed if no active connection to fieldbus controller

#### 9.2.14.1 Example: Reading an IO-Link process value (operating mode "IO-Link")

Task: Read the current measured value of the ifm temperature sensor TN2531 at port 2.

Solution: Read the value of the pdin data point.

Request:

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

Response:

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

The return value is displayed in hexadecimal format. 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**11110010**01

Temperature value: 0b11110010 = 242

This means that The current temperature value is 24.2 °C.

#### 9.2.14.2 Example: Reading an IO-Link process value (operating mode "IO-Link")

Task: The buzzer of the DV2500 at port 2 is to be switched on. The DV2500 operates in On/Off mode.

Solution: Write the value of the pdout data point. The IODD of the DV2500 shows the structure of the process value (e.g. LED activity). The buzzer is switched via bit 40 of the process value (OFF = 0, ON = 1).

Course:

1. Read the current process value Example: Reading an IO-Link process value (operating mode "IO-Link") (→ 🗋 46).
2. Set bit 40 of the read value to 1.
3. Write the new process value.

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
}
```

#### 9.2.14.3 Example: Reading the digital input value (operating mode "DI")

Task: The digital input value of the IO-Link device at port 2 (pin 4) is to be read. The operating mode of the IO-Link port is "Digital Intput (DI)".

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

Request:

```
{
  "code":"request",
  "cid":10,
  "adr":"iolinkmaster/port[2]/iolinkdevice/pdin/getdata"
}
```

Response:

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

#### 9.2.14.4 Example: Writing the digital output value (operating mode "DO")

Task: The digital output value of the IO-Link device at port 2 (pin 4) is to be set. The operating mode of the IO-Link port is "Digital Output (DO)".

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

Request:

```
{
  "code":"request",
  "cid":10,
  "adr":"iolinkmaster/port[2]/iolinkdevice/pdout/setdata",
  "data":{
    "newvalue":"01"
  }
}
```

Response:

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

### 9.2.15 Ports: Reading port events

A port event contains information about events on the IO-Link port (e.g. IO-Link device connected or disconnected, change of the operating mode of the port).

Substructure: iolinkmaster/port[x] (x: 1…8)

Available data points:

| Name | Description | Value range | Access |
|---|---|---|---|
| ../portevent | Port event<br>Structure: 0xYYZZ00<br>• 0xYY: Device Status - Status of the IO-Link devices<br>• 0xZZ: Connection Status - Status of the connection | Device status:<br>• 0x00: No IO-Link device connected<br>• 0x40: Wrong IO-Link device connected<br>• 0x80: IO-Link device connected and in PREOPERATE state<br>• 0xFF: IO-Link device connected and in OPERATE state<br>Connection Status:<br>• 0x00: Port deactivated<br>• 0x01: Port activated, but no device connected<br>• 0x02: Port activated and in “IO-Link” operating mode<br>• 0x03: Port activated and in “DI” operating mode<br>• 0x04: Port activated and in “DO” operating mode<br>• 0x1B: Repetitions detected<br>• 0x1E: Short circuit detected at pin 4<br>• 0x42: Wrong revision of the IO-Link device<br>• 0x43: Wrong vendor ID (V1.1 Revision)<br>• 0x44: Wrong device ID (V1.1 Revision)<br>• 0x45: Wrong vendor ID (V1.0 Revision)<br>• 0x46: Wrong device ID (V1.0 Revision)<br>• 0x48: Wrong cycle time | ro [1] |

[1] Read only

### 9.2.16 Timer: Configuring the timers

The unit has 2 timers. The timers can be used for time-triggered transmission of notifications.

The timers are active by default.

Substructure: timer[1] | timer[2]

Available data points:

| Name | Description | Value range | Access |
|---|---|---|---|
| ../counter | Counter reading | • 0…4294967295 | rw [1] |
| ../interval | Counting interval (value in ms) | • 1000: 1000 ms<br>…<br>• 86400000: 86400000 ms | rw [1] |
| ../interval/unit | Unit of the counting interval | • ms: milliseconds | ro [2] |

[1] Read and write

[2] Read only

#### 9.2.16.1 Set the counting interval

Task: The counting interval of timer 2 is to be set to 2.5 s.

Solution: Using the `setdata` service to write the new value 2500 to the data point `/timer[2]/interval` (2500 ms = 2.5 s).

Request:

```
{
  "code": "request",
  "cid": 1,
  "adr": "/timer[2]/interval/setdata",
  "data": {
  "newvalue": 2500
  }
}
```

Response:

```
{
  "cid": 1,
  "code": 200
}
```

## 9.2.17 IO-Link devices: Accessing parameters

The ifm IoT Core supports access to the parameters of connected IO-Link devices via the IT network using ISDU (Index Service Data Unit). Each parameter can be addressed via its ISDU index and ISDU subindex (→ IODD of the IO-Link device). If access to the IO-Link device is faulty, the ISDU error code will also be output in the response (→ IODD of the IO-Link device).

Substructure: iolinkmaster/port[x]/iolinkdevice (x: 1…8)

Applicable services:

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

When using the validation level "Type compatible V1.1 Device with Backup + Restore":

After changing a parameter via ISDU write access, the user must end the parameter setting process with the "ParamDownloadStore" system command and activate the data storage mechanism on the IO-Link device.

▶ Set the SystemCommand object (ISDU index: 0x0002) to the value 0x05 (command "ParamDownloadStore") via acyclic ISDU write access.

▷ The parameter setting process is finished.

▷ The data storage mechanism on IO-Link device will be activated.

▷ The IO-Link device synchronises changed parameter values with data storage of the IO-Link master.

### 9.2.17.1 Example: Reading a parameter value of an IO-Link device

Task: Reading the serial number of the ifm temperature sensor TN2531 at port 2.

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
  }
}
```

Response:

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

The returned value is displayed in hexadecimal format. Converting the HEX value to a STRING value results in: G0214280710.

#### 9.2.17.2 Example: Writing a parameter value of an IO-Link device

Task: Setting the output configuration OUT1 of the ifm temperature sensor TN2531 at port 2 to the value "Hnc / hysteresis function, normally closed".

Solution: Set the parameter [ou1] of the sensor with the service `iolwriteacyclic` to the value 4. 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 must be passed in hexadecimal format. Converting the STRING value to a HEX value results in: 34.

Response:

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

### 9.2.18 IO-Link devices: Reading and writing device information

Substructure: iolinkmaster/port[x]/iolinkdevice (x: 1…8)

Available data points:

| Name | Description | Value range | Access |
|---|---|---|---|
| ../status | Status of the connected IO-Link device | • 0: State not connected - State not connected<br>• 1: State preoperate - State PREOPERATE<br>• 2: State operate - State OPERATE<br>• 3: State communication error - State communication error | ro [1] |
| ../vendorid | IO-Link ID of the manufacturer | e.g. 310: ifm electronic | ro [1] |
| ../deviceid | IO-Link ID of the IO-Link device | e.g. 712: ifm pressure sensor, PV7003 | ro [1] |
| ../productname | Product name of the IO-Link device | e.g. PV7003 | ro [1] |
| ../serial | Serial number of the IO-Link device | e.g. 000008500706 | ro [1] |
| ../applicationspecifictag | Application specific identification (application tag) | e.g. sensor1_machine3 | rw [2] |

[1] Read only

[2] Read and write

### 9.2.19 IO-Link devices: Reading IO-Link events

The device supports IO-Link events. IO-Link events are event and error messages. IO-Link events can be generated in the IO-Link master and in the connected IO-Link devices. IO-Link events generated in the IO-Link devices are forwarded to the IO-Link master and stored there.

An IO-Link event message has the following structure:

| Byte 0…1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 |
|---|---|---|---|---|---|
| Event Code | Source | Validity | Type | Mode | Instance |

Legend

| | | | |
|---|---|---|---|
| • Instance | IO-Link Event Qualifier: Trigger | 1 byte | • 0x00: unknown<br>• 0x01: PL (Physical Layer)<br>• 0x02: DL (Data Layer)<br>• 0x03: AL (Application Layer)<br>• 0x04: APPL (Application) |
| • Mode | IO-Link Event Qualifier: Event trigger | 1 byte | • 0x40: One-time event or warning (single shot)<br>• 0xC0: Error disappeared<br>• 0x80: Error appeared |
| • Type | IO-Link Event Qualifier: Event category | 1 byte | • 0x10: Notification<br>• 0x20: Warning<br>• 0x30: Error |
| • Validity | Validity of the process data | 1 byte | • 0x00: Valid<br>• 0x40: Invalid |
| • Source | IO-Link Event Qualifier: Event source | 1 byte | • 0x00: IO-Link device<br>• 0xFF: IO-Link master |
| • Event Code | IO-Link event code (bytes are swapped!) | 2 bytes | → IO-Link specification |

Substructure: iolinkmaster/port[x]/iolinkdevice (x: 1…8)

Available data points:

| Name | Description | Value range | Access |
|---|---|---|---|
| ../iolinkevent | IO-Link event | See note | ro [1] [2] |

[1] Read only

[2] Parameter only available if operating mode pin 4 (US) = IO-Link and IO-Link device is connected to the port

### 9.2.20 Gateway: Set the application tag

Substructure: devicetag

Available data points:

| Name | Description | Value range | Access |
|---|---|---|---|
| ../applicationtag | Bezeichnung des Geräts (Application Tag) | e.g. “factory 2 plant 1” | rw [1] |

[1] Read and write

32 bytes are available on the device for storing the applicationtag parameter. If the memory range is exceeded, the device will abort the write process (diagnostic code 400).

▶ When writing the applicationtag parameter, 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).

### 9.2.21 Gateway: Update firmware

Substructure: firmware

Available data points:

| Name | Description | Value range | Access |
|---|---|---|---|
| ../version | Firmware version | AL1x2x_cn_ei_v3.1.95 | ro [1] |
| ../type | Software type | firmware | ro [1] |
| ../container | Structure for firmware (BLOB) | - | wo |
| ../container/maxsize | Container size (in bytes) | e.g. 4194304 | ro [1] |
| ../container/chunksize | Size of a data segment (in bytes) | e.g. 4096 | ro [1] |
| ../container/size | Size of firmware file in container (in bytes) | e.g. 634523 | ro [1] |

[1] Read only

Applicable services:

| Name | Description |
|---|---|
| ../install | Install firmware |
| ../container/stream_set | Transfer an individual data segment |
| ../container/start_stream_set | Start sequential transmission of several data segments |

ifm recommends using the IoT Core Visualizer (→ IoT Core Visualizer 64) to update the firmware.

### 9.2.22 Gateway: Reading status and diagnostic information

Substructure: processdatamaster

Available data points:

### 9.2.23 Gateway: Reading device information

Substructure: deviceinfo

Available data points:

Additional information about the can be read with the service getidentity (→ Service: getidentity 74).

### 9.2.24 Gateway: Resetting, rebooting and localising the device

Substructure: firmware

Applicable services:

| Name | Description |
|---|---|
| ../factoryreset | Reset device to factory settings |
| ../reboot | Restart the device. |
| ../signal | Trigger the flashing of the status LEDs |

### 9.2.25 Notifications

The ifm IoT-Core offers the possibility to send notifications. A notification can contain the current values of individual or multiple process data.

Notifications can be triggered either time-controlled or event-controlled.

Notifications can be transmitted using the following protocols:

- HTTP (Hypertext Transfer Protocol)
- MQTT (Message Queuing Telemetry Transport)
- TCP (Transmission Control Protocol)
- WS (WebSocket)

Notifications can be transmitted in the following formats:

- JSON
- CSV

The activity duration of a notification can be set:

- once: Notification is only active once and is then deleted
- uptime: Notification is active until the next restart of the unit and is then deleted
- lifetime: Notification is permanently active

Notifications are set up with the subscribe service.

Active notifications can be changed with the subscribe service.

All active notifications can be displayed with the getsubscriberlist service (→ Service: getsubscriberlist 🗋 75).

Information about an active notification can be displayed with the getsubscriptioninfo service (→ Service: getsubscriptioninfo 🗋 75).

Active notifications can be unsubscribed using the unsubscribe service.

#### 9.2.25.1 Event-controlled notifications

All elements of the ifm IoT Core with a sub-element of the type `event` can be used as triggers for event-triggered notifications.

| Name | Description |
|---|---|
| ../datachanged | Node for displaying value changes of the parent data node |
| ../eventcollection | Node for displaying value changes of multiple data nodes of the IoT Core |
| ../treechanged | Node for displaying structural changes in the IoT Core |

Applicable services:

| Name | Description |
|---|---|
| ../subscribe | Subscribe to notifications |

| Name | Description |
|---|---|
| ../unsubscribe | Unsubscribe from notifications |
| ../getsubscriptioninfo | Display information about notifications |

#### 9.2.25.2 Time-controlled notifications

The ifm IoT Core offers 2 timers that can be used as triggers for time-triggered notifications.

| Name | Description |
|---|---|
| ../timer[1]/counter/datachanged | Counter of timer 1 |
| ../timer[2]/counter/datachanged | Counter of timer 2 |

The interval time of the timers can be set. Each counting pulse triggers a notification.

Applicable services:

| Name | Description |
|---|---|
| ../subscribe | Subscribe to notifications |
| ../unsubscribe | Unsubscribe from notifications |
| ../getsubscriptioninfo | Display information about notifications |

#### 9.2.25.3 Example: Subscribing to notifications

Task: Every 0.5 seconds, the current values of the following parameters are to be sent regularly to a network server with the IP address 192.168.0.4/temp.

- Cyclic IO-Link input data of the IO-Link device at IO-Link port 2
- Operating temperature of the IO-Link master.



The following options are additionally available:

via WebSockets (ws://): Example: Using WebSockets (→ 🗋 63)

via MQTT (mqtt://): Example: Configure and activate MQTT command channel (→ 🗋 60)

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/pdin",
      "/processdatamaster/temperature"]
  }
}
```

▶ Set the timer to 500 ms.

- Request:

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

- Response:

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

- Notification (Format: 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"}
    }
  }
}
```

#### 9.2.25.4 Example: Changing a notification

Task: The existing sbscription is to be changed (→ Example: Subscribing to notifications 55). 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"]
  }
}
```

#### 9.2.25.5 Example: Subscribing to a notification in CSV format

Task: Every 2 seconds, the current values of the following parameters are to be sent to a network server with the IP address 192.168.0.4

- Cyclic IO-Link input data of the IO-Link device at port 2
- Operating temperature of the IO-Link master.

The data should be transmitted in CSV format (comma separator).

Solution:

▶ Use the subscribe service to subscribe to the required data and set the output format to "csv0".

Data in CSV format can only be sent via TCP connection.

- Request:

```
{
  "code":"request",
  "cid": 1,
  "adr":"/timer[1]/counter/datachanged/subscribe",
  "callback":"tcp://192.168.50.59:1883/topic",
  "codec":"csv0",
  "data":{
    "datatosend":[
      "/iolinkmaster/port[2]/iolinkdevice/pdin",
      "/processdatamaster/temperature"]
  }
}
```

▶ Set the interval of the timer to 2 seconds:

- Request:

```
{
  "code":"request",
  "cid":4712,
  "adr":"/timer[1]/interval/setdata",
  "data":{"newvalue":2000}
}
```

The cyclically sent notification has the following structure:

```
/timer[1]/counter/datachanged,6317,200,1,200,39,200,03B0
```

#### 9.2.25.6 Example: Unsubscribing from a notification

Task: The existing subscription Example: Subscribing to notifications (→ 🗋 55) 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.

- Request:

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

#### 9.2.25.7 Example: checking a notification

Task: Display information about the existing subscription Example: Subscribing to notifications (→ 🗋 55).

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:

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

#### 9.2.25.8 MQTT support

The ifm IoT Core supports the MQTT protocol. MQTT lets a client communicate with the ifm IoT Core via a broker to request and receive data. The ifm IoT Core can publish data to an MQTT broker.

▶ Establish MQTT connections only on IoT port X23 of the device.

To enable MQTT communication, the user needs to configure the MQTT command channel. The configuration consists of the following sub-steps:

▶ Check that the MQTT broker can be reached.

▶ Check that the requested port of the MQTT broker is open.

▶ Activate the MQTT command channel.

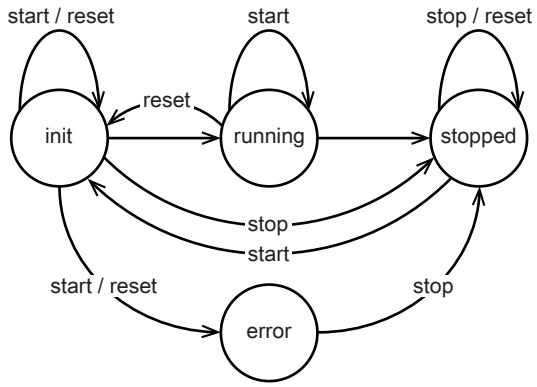
▶ Configure the MQTT command channel.

▶ Configure the MQTT connection.

▷ The MQTT connection is configured and activated.

**Configuring the MQTT connection**

The following status diagram shows how the `start`, `stop` and `reset` services influence the current status of an MQTT connection:




After successful initialisation in the [init] state, the connection automatically changes to the [running] state.

The connection automatically assumes the [error] state if at least one of the following events occurs:

- no MQTT broker can be reached

Maximum number of MQTT broker: 1

Maximum number of simultaneous MQTT connections: 3

Substructure: connections/mqttConnection

Available data points:

| Name | Description | Value range | Access |
|---|---|---|---|
| ../type | Type of the connection (MQTT) | mqtt: MQTT connection | ro [1] |
| ../status | Status of the MQTT connection | • init: [init] state – initialisation<br>• stopped: [stopped] state – connection stopped<br>• running: [running] state – connection running<br>• error: [error] state | ro [1] |
| ../status/preset | Presetting of the status of the MQTT connection | • running: [running] state (default) | ro [1] |
| ../MQTTSetup | Substructure for general MQTT settings | | |
| ../MQTTSetup/QoS | Quality of Service (QoS) of the MQTT connection | • 0: QoS level 0 - PUBLISH (without confirmation) (default)<br>• 1: QoS level 1 - PUBLISH > PUBREC (single confirmation)<br>• 2: QoS level 2 - PUBLISH > PUBREC > PUBREL > PUBCOMP (double confirmation) | rw [2] |
| ../MQTTSetup/version | MQTT version | e.g. 3.1.1 | ro [1] |

[1] Read only

[2] Read and write

Applicable services:

| Name | Description |
|---|---|
| ../status/start | Activate MQTT. |
| ../status/stop | Deactivate MQTT. |
| ../status/reset | Reset MQTT. |

### Configuring the MQTT command channel

Make sure that the MQTT broker can be reached and that the selected port of the MQTT broker is released for data transfer.

The wildcards “+” and “#” are not supported in topic names.

Substructure: connections/mqttConnection/mqttCmdChannel

Available data points:

| Name | Description | Value range | Access |
|---|---|---|---|
| ../type | Type of the command channel | • mqtt: MQTT command channel (default) | ro [1] |
| ../status | Status of the command channel | • init: [init] state<br>• stopped: [stopped] state<br>• running: [running] state (default)<br>• error: [error] state | ro [1] |
| ../status/preset | Preset channel state | • running: RUN state (default) | ro [1] |
| ../mqttCmdChannelSetup | Substructure for settings of the command channel | -- | |
| ../mqttCmdChannelSetup/brokerIP | IP address of the MQTT broker | e.g. 3.69.1.9 | rw [2] |
| ../mqttCmdChannelSetup/brokerPort | Port number of the MQTT broker | e.g. 1883 | rw [2] |
| ../mqttCmdChannelSetup/cmdTopic | Designation of the MQTT command topic | e.g. reqTopic | rw [2] |
| ../mqttCmdChannelSetup/defaultReplyTopic | Designation of the MQTT standard response topic | e.g. respTopic | rw [2] |

[1] Read only

[2] Read and write

Applicable services:

| Name | Description |
|---|---|
| ../status/start | Start the MQTT command channel |
| ../status/stop | Stop the MQTT command channel |
| ../status/reset | Reseit the MQTT command channel |

### Example: Configure and activate MQTT command channel

Task: Configuring and activating the MQTT command channel (IP address of the MQTT broker): 192.168.82.100, port: 1883, command topic: reqTopic, standard response topic: respTopic, Quality of Service: Level 2)

Solution:

▶ Check whether MQTT broker can be reached and the port has been released.

▶ Set QoS level.

Request:

```
{
  "code":"request",
  "cid":1,
  "adr":"/connections/mqttConnection/MQTTSetup/cmdChannel/QoS/setdata"
  "data":{"newvalue":2}
```

}

▶ Activate MQTT connection

Request:

```
{
  "code":"request",
  "cid":1,
  "adr":"connections/mqttConnection/MQTTSetup/cmdChannel/Status/start"
}
```

▶ Set the IP address of the MQTT broker.

Request:

```
{
  "code":"request",
  "cid":2,
  "adr":"connections/mqttConnection/mqttCmdChannel/mqttCmdChannelSetup/
brokerIP/setdata",
  "data":{
    "newvalue":"192.168.82.100"
  }
}
```

▶ Set the port number of the MQTT broker.

Request:

```
{
  "code":"request",
  "cid":3,
  "adr":"connections/mqttConnection/mqttCmdChannel/mqttCmdChannelSetup/
brokerPort/setdata",
  "data":{
    "newvalue":"1883"
  }
}
```

▶ Set the command topic.

Request:

```
{
  "code":"request",
  "cid":4,
  "adr":"connections/mqttConnection/mqttCmdChannel/mqttCmdChannelSetup/
cmdTopic/setdata",
  "data":{
    "newvalue":"reqTopic"
  }
}
```

▶ Set standard response topic.

Request:

```
{
  "code":"request",
  "cid":5,
  "adr":"/connections/mqttConnection/mqttCmdChannel/mqttCmdChannelSetup/
defaultReplyTopic/setdata",
  "data":{
    "newvalue":"respTopic"
  }
}
```

**Example: Publishing the temperature to an MQTT broker**

Task: The temperature of the unit is to be published to an MQTT broker every 5 seconds (MQTT broker: 192.168.82.100, port: 1883, topic: deviceTemp).

Solution:

▶ Setting the interval of the timer

Request:

```
{
  "code":"request",
  "cid":1,
  "adr":"/timer[1]/interval/setdata",
  "data":{
    "newvalue":5000
  }
}
```

▶ Publishing the temperature to an MQTT broker

Request:

```
{
  "code":"request",
  "cid":2,
  "adr":"/timer[1]/counter/datachanged/subscribe",
  "data":{
    "callback":"mqtt://192.168.82.100:1883/deviceTemp",
    "datatosend":["processdatamaster/temperature"]
  }
}
```

Response:

```
{
  "cid":2,
  "code":200
}
```

#### 9.2.25.9 WebSocket support

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

WebSocket can be used for the following services:

- subscribe
- unsubscribe

Maximum number of simultaneous WebSocket connections: 8

Secure WebSocket connections (wss://) are not supported.

To transmit notifications via a WebSocket connection:

▶ Establish a WebSocket connection (e.g. "ws://192.168.0.55:80/mywebsocket")

- Option 1: without parameter "callback”

▶ Send a subscribe/unsubscribe request without "callback” parameter.

▷ ifm IoT-Core sends notifications about existing WebSocket connections.

- Option 2: with parameter "callback”

▶ Send subscribe/unsubscribe requests with "callback” parameter ("ws:///myTopic").

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

**Example: Using WebSockets**

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

- Product name of the IO-Link device at IO-Link port 2
- Cyclic IO-Link input data of the IO-Link device at IO-Link port 2
- 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"]
  }
}
```

## 9.3 IoT Core Visualizer

The IoT Core Visualizer provides a graphical user interface to access the functions of the ifm IoT Core.

### 9.3.1 Starting the IoT-Core Visualizer

To start the IoT-Core Visualizer:

Requirements:

✓ The IoT interface is configured.

▶ Start the web browser.

▶ Go to the following URL: http://<ip-address>/web/subscribe (e. g. `http://192.168.82.100/web/subscribe`)

▷ The web browser displays the start page of the IoT-Core Visualizer.

The navigation menu gives access to the following functions:

- [Notification]: Managing notifications (→ 🗋 65)
- [Elements]: Searching for elements of the IoT Core (→ 🗋 66)
- [Parameter]: Configuring the IO-Link master (→ 🗋 67)
- [Processdata]: Accessing process data (→ 🗋 68)
- [Update]: Update firmware (→ 🗋 69)

### 9.3.2 Managing notifications

The [Notification] menu page gives access to the following functions:

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

Requirements:

✓ IoT Core Visualizer has been started.

▶ Select the [Notification] menu.

▷ Menu page for managing notifications appears.

▷ Table shows all active notifications.

#### 9.3.2.1 Creating a new notification

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 creating notifications appears.

▶ Use the wizard to enter the required notification parameters step by step.

▷ The wizard generates a notification.

▷ The created notification is displayed in the table.

▶ For cyclical notifications via timer[1] or timer[2], set the interval time of the corresponding timer.

#### 9.3.2.2 Deleting a notification

To delete an active IODD:

Requirements:

✓ The [Notification] menu page is open

✓ At least one notification is active.

▶ Click on the x symbol in the [Unsubscribe] column.

▷ The selected notification will be deleted (unsubscribe).

### 9.3.3 Searching for elements of the IoT Core

The [Elements] menu page allows you to search the IoT Core tree for elements with specific properties and output the results.

The following properties can be searched for:

- [identifier]: Name of the element
- [profile]: Profile of the element
- [type]: Type of the element

Requirements:

✓ IoT Core Visualizer has been started.

✓ The [Elements] menu is active.

▶ Select the search criteria for the required element in the selection lists [identifier], [profile] and [type].

▶ Click on [Search for…].

▷ IoT Core Visualizer searches the device description for elements with selected search criteria.

▷ The result list shows all the elements found.

### 9.3.4 Configuring the IO-Link master

The [Parameter] menu page allows 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.

▶ Select the [Parameter] menu.

▷ The menu page shows the available parameters of the IO-Link master.

▷ Current parameter values are displayed.

▶ Optional: Click on ↻ next to an element to manually update the process value.

To change a parameter:

▶ Navigate to the desired parameter in the device description.

▶ Changing the parameter value

▶ Click on ✎ to save the change on the IO-Link master.

▷ The changed parameter value is active.

▶ Optional: Repeat the procedure to change further parameter values.

### 9.3.5 Accessing process data

The [Processdata] menu page makes it possible to read and write the process data of the IO-Link master and the connected IO-Link devices

Requirements:

✓ IoT Core Visualizer has been started.

▶ Select the [Processdata] menu.

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

▷ The current process values are displayed.

▶ Optional: Activate the [Polling] option and change the update interval.

▷ The process values will be updated with the set interval.

▶ Optional: Click on ↻ next to an element to manually update the process value.

To change the value of a process date:

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

▶ Change the process value.

▶ Click on ✎ to save the change on the IO-Link master.

▷ Changes are saved on the IO-Link master.

▷ The changed process value is active.

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

### 9.3.6 Update firmware

The [Update] menu page allows you to update the firmware of the device:

Requirements:

✓ IoT Core Visualizer has been started.

✓ New firmware has been downloaded. documentation.ifm.com

▶ Select the [Update] menu.

▷ Menu page shows information about the current firmware version.

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

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

▷ The firmware of the device is updated.

▷ The progress of the update process is displayed.

▷ After successful update: The device reboots automatically.

## 9.4 EtherNet/IP

### 9.4.1 Note: Startup package

ifm electronic provides Startup Packages for the integration of the device in EtherNet/IP environments with controllers from different manufacturers.

The Startup Packages can be downloaded from the ifm website: documentation.ifm.com

# 10 Maintenance, repair and disposal

The operation of the unit is maintenance-free.

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

## 10.1 Cleaning the housing surface

Clean the surface of the device when necessary.

▶ Disconnect the device.

▶ Clean the device from dirt using a soft, chemically untreated and dry cloth.

▶ In case of severe soiling, use a cloth moistened with water.

▶ Do not use any caustic cleaning agents for this!

## 10.2 Update firmware

The firmware of the device can be updated using the following methods:

- ifm IoT Core Visualizer: Update firmware (→ 🗋 69)
- ifm moneo: Updating the firmware (→ 🗋 29)

# 11 Appendix

## 11.1 ifm IoT Core

### 11.1.1 Profiles

| Profile | Description |
|---|---|
| blob | Binary Large Object |
| deviceinfo | Identification information of a device |
| devicetag | Device-specific identification |
| iolinkdevice_full | IO-Link device |
| iolinkmaster | IO-Link master |
| network | Network |
| parameter | Parameter |
| processdata | Process data |
| service | Service |
| software | Software |
| software/uploadable-software | upgradeable software |
| timer | Counter |

### 11.1.2 Types

| Type | Description |
|---|---|
| data | Data point |
| device | Root element representing a device |
| event | Event that can be triggered by the firmware and sends notifications |
| service | Service that can be addressed from the network |
| structure | Structural element (e.g. a folder in the file system) |

### 11.1.3 Services

#### 11.1.3.1 Service: factoryreset

Name: `factoryreset`

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

Request (“data” field): none

Response (“data” field): none

#### 11.1.3.2 Service: getblobdata

Name: `getblobdata`

Description: The service reads a Binary Large Object (blob).

Request (“data” field):

| Parameter | Mandatory field | Data type | Description |
|---|---|---|---|
| Pos | Mandatory | NUMBER | Byte position |
| length | Mandatory | NUMBER | Size of the object (number of bytes) |

Response (“data” field):

| Parameter | Mandatory field | Data type | Description |
|---|---|---|---|
| data | Mandatory | STRING | Data to be decoded (BASE64-encoded) |
| crc | Optional | HEX STRING | CRC of the data after decoding |
| md5 | Optional | HEX STRING | MD5 checksum of the data after decoding |

#### 11.1.3.3 Service: getdata

Name: `getdata`

Description: The service reads the value of a data point and provides it.

Request (“data” field): none

Response (“data” field):

| Parameter | Mandatory field | Data type | Description |
|---|---|---|---|
| value | Mandatory | STRING | Value of the data point |

#### 11.1.3.4 Service: getdatamulti

Name: `getdatamulti`

Description: The service sequentially reads the values of multiple data points and outputs them. The value and a diagnostic code are output for each data point.

Request (“data” field):

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

Response (“data” field):

| Parameter | Mandatory field | Data type | Description |
|---|---|---|---|
| url | Mandatory | STRING | Data point that was queried |
| code | Mandatory | INT | Diagnostic code of the request |
| data | Mandatory | STRING | Value of the data point |

#### 11.1.3.5 Service: getelementinfo

Name: `getelementinfo`

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

Request ("data" field):

| Parameter | Mandatory field | Data type | Description |
|---|---|---|---|
| adr | mandatory | STRING | URL of the element whose proerties are to be changed |

Return ("data" field):

| Parameter | Mandatory field | Data type | Description |
|---|---|---|---|
| identifier | mandatory | STRING | Identifier of the element |
| type | mandatory | STRING | Type of the element |
| format | optional | JSON object | Format of the data or of the service content |
| uid | optional | STRING | |
| profiles | optional | JSON-ARRAY | Element profiles |
| hash | optional | STRING | |

#### 11.1.3.6 Service: getidentity

Name: `getidentity`

Description: The service reads device information and outputs it.

Request (field „data“): keine

Response (field „data“):

| Parameter | Mandatory field | Data type | Description |
|---|---|---|---|
| iot | | device | Device description as JSON object |
| iot.name | mandatory | STRING | Type of the element |
| iot.uid | optional | STRING | |
| iot.version | mandatory | STRING | |
| iot.catalogue | optional | ARRAY OF OBJECTS | |
| iot.deviceclass | optional | ARRAY OF STRING | Device class |
| iot.serverlist | optional | ARRAY OF OBJECTS | |
| device | optional | | Article nummer |
| device.serialnumber | optional | | Serial number |
| device.hwrevision | optional | | Hardware version |
| device.swrevision | optional | | Software version |
| device.custom | optional | | |

#### 11.1.3.7 Service: getsubscriberlist

Name: `getsubscriberlist`

Description: The service provides a list of all active subscriptions.

Request (“data” field): none

Response (“data” field):

| Parameter | Mandatory field | Data type | Description |
|---|---|---|---|
| adr | Mandatory | STRING | Notification trigger |
| datatosend | Mandatory | ARRAY OF STRINGS | List with URLs of the data elements; URLs are comma separated |
| cid | Mandatory | STRING | ID of the active notification |
| callbackurl | Mandatory | STRING | Destination address for the notifications |
| duration | Mandatory | STRING | Activity duration |

#### 11.1.3.8 Service: getsubscriptioninfo

Name: `getsubscriptioninfo`

Description: The service provides information about an active notification (subscribe). The following parameters of the active notification are to be used for the query:

- `cid` (e.g. 4711)
- `adr` (e.g. timer[1]/counter/datachanged)
- `callback` (e.g. http://192.168.82.121:8080/topic)

Request ("data" field):

| Parameter | Mandatory field | Data type | Description |
|---|---|---|---|
| callback | mandatory | STRING | Destination address of the notifications; complete URL (e.g. http://ipaddress:port/path) |

Return ("data" field):

| Parameter | Mandatory field | Data type | Description |
|---|---|---|---|
| subscription | mandatory | BOOL | Status of the transferred parameters of the notification<br>• FALSE: Parameter incorrect<br>• TRUE: Parameter correct; Notification found |
| datatosend | mandatory | STRING | List with URLs of the data elements; URLs are comma separated |
| cid | mandatory | STRING | ID of the active notification |
| callbackurl | mandatory | STRING | Destination address of the notifications |

#### 11.1.3.9 Service: gettree

Name: `gettree`

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

Request (“data” field):

| Parameter | Mandatory field | Data type | Description |
|---|---|---|---|
| adr | Optional | STRING | Root element of the subtree |

| Parameter | Mandatory field | Data type | Description |
|---|---|---|---|
| level | Optional | STRING | Max. level up to which the subtree is output<br>• no entry: all levels will be displayed<br>• 0: do not display sub-elements (“subs”)<br>• 1: display sub-elements<br>• 2: display sub-elements up to the 2nd level<br>• 3: display sub-elements up to the 3rd level<br>...<br>• 20: display sub-elements up to the 20th level |

Response (“data” field):

| Parameter | 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 | |

#### 11.1.3.10 Service: install

Name: `install`

Description: The service installs the firmware stored in a memory area of the unit.

Request ("data" field): none

Return ("data" field): none

#### 11.1.3.11 Service: iolreadacyclic

Service: `iolreadacyclic`

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

Request ("data" field):

| Parameter | Mandatory field | Data type | Description |
|---|---|---|---|
| index | mandatory | NUMBER | IO-Link index of the parameter |
| subindex | mandatory | NUMBER | IO-Link subindex of the parameter |

Return ("data" field):

| Parameter | Mandatory field | Data type | Description |
|---|---|---|---|
| value | mandatory | STRING | Parameter value (value in hexadecimal format) |

#### 11.1.3.12 Service: iolwriteacyclic

Name: `iolwriteacyclic`

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

Request ("data" field):

| Parameter | Mandatory 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 | Parameter value (value in hexadecimal format) |

Return ("data" field): none

#### 11.1.3.13 Service: querytree

Name: `querytree`

Description: The service searches a device tree for the criteria `profile`, `type` and `identifier` 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 unit.

Request (“data” field):

| Parameter | Mandatory field | Data type | Description |
|---|---|---|---|
| profile | Optional | STRING | Profile of the requested element |
| type | Optional | STRING | Type of the requested element |
| identifier | Optional | STRING | Name of the requested element |

Response (“data” field): none

| Parameter | Mandatory field | Data type | Description |
|---|---|---|---|
| urlList | Mandatory | ARRAY | Array with URLs of the found elements; URLs are separated by commas |

#### 11.1.3.14 Service: reboot

Name: `reboot`

Description: The service reboots the device.

Request (“data” field): none

Response (“data” field): none

#### 11.1.3.15 Service: reset

Name: `reset`

Description: The service resets a connection to the initialisation state.

Request (“data” field): none

Response (“data” field): none

#### 11.1.3.16 Service: setblock

Name: `setblock`

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

Request (“data” field):

| Parameter | Mandatory field | Data type | Description |
|---|---|---|---|
| datatoset | Mandatory | ARRAY OF OBJECTS | List of data points and their new values; Data points must support the `setdata` service |

| Parameter | Mandatory field | Data type | Description |
|---|---|---|---|
| consistent | Optional | BOOL | Response within a certain time |

Response (“data” field): none

#### 11.1.3.17 Service: setdata

Name: `setdata`

Description: The service sets the value of a data point.

Request (“data” field):

| Parameter | Mandatory field | Data type | Description |
|---|---|---|---|
| newvalue | Mandatory | STRING | New value of the data point |
| duration | Optional | STRING | Duration of value storage<br>• lifetime: value is saved with IoT Core; value remains valid even after restart of the device<br>• uptime: value is saved until the next restart of the device |

Response (“data” field): none

#### 11.1.3.18 Service: signal

Name: `signal`

Description: The service triggers the flashing of the status LEDs of the unit.

Request (“data” field): none

Response (“data” field): none

#### 11.1.3.19 Service: start

Name: `start`

Description: The service starts a function (e.g. connection, timer).

Request (“data” field): none

Response (“data” field): none

#### 11.1.3.20 Service: start_stream_set

Name: `start_stream_set`

Description: The service starts the sequential transmission of several data fragments.

Request (“data” field):

| Parameter | Mandatory field | Data type | Description |
|---|---|---|---|
| size | Mandatory | STRING | Overall length of the data to be transmitted (number of bytes) |

Response (“data” field): none

#### 11.1.3.21 Service: stop

Name: `stop`

Description: The service stops a function (e.g. connection, timer).

Request (“data” field): none

Response (“data” field): none

#### 11.1.3.22 Service: stream_set

Name: `stream_set`

Description: The service transfers a data segment.

Request (“data” field):

| Parameter | Mandatory field | Data type | Description |
|---|---|---|---|
| value | Mandatory | BIN (BASE64) | Segment of the binary data (BASE64-encoded) |

Response (“data” field): none

#### 11.1.3.23 Service: validation_useconnecteddevice

Name: `validation_useconnecteddevice`

Description: The service checks whether the device ID and the vendor ID of the connected IO-Link device match the data points `../validation_vendorid` and `../validation_deviceid` .

Request ("data" field): none

Return ("data" field): none

# 11.2 EtherNet/IP

## 11.2.1 Parameters

### 11.2.1.1 Configuration Assembly (Instance 199)

| Byte (Offset) | Bit | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | Communication profile | | | | | | | |
| 1 | Port Process Data Size | | | | | | | |
| 2…13 | Port X01: Port Configuration (→ Mapping: Port Configuration 81) | | | | | | | |
| 14…25 | Port X02: Port Configuration (→ Mapping: Port Configuration 81) | | | | | | | |
| 26…37 | Port X03: Port Configuration (→ Mapping: Port Configuration 81) | | | | | | | |
| 38…49 | Port X04: Port Configuration (→ Mapping: Port Configuration 81) | | | | | | | |
| 50…61 | Port X05: Port Configuration (→ Mapping: Port Configuration 81) | | | | | | | |
| 62…73 | Port X06: Port Configuration (→ Mapping: Port Configuration 81) | | | | | | | |
| 74…85 | Port X07: Port Configuration (→ Mapping: Port Configuration 81) | | | | | | | |
| 86…97 | Port X08: Port Configuration (→ Mapping: Port Configuration 81) | | | | | | | |

Legend:

| | | | |
|---|---|---|---|
| • Communication profile | The access rights to the parameters, process data and events/diagnostic messages of the IO-Link master and the connected IO-Link devices | 1 BYTE | • 0x00: EtherNet/IP + LineRecorder<br>• 0x01: EtherNet/IP + LineRecorder (ro)<br>• 0x02: EtherNet/IP only<br>• 0x03: Keep setting (Default) |
| • Port Process Data Size | Length of the process data (inputs and outputs) | 1 BYTE | • 0x00: 2 bytes<br>– Input Assemby: 206 bytes<br>– Output Assembly: 62 bytes<br>• 0x01: 4 bytes<br>– Input Assemby: 222 bytes<br>– Output Assembly: 78 bytes<br>• 0x02: 8 bytes<br>– Input Assemby: 254 bytes<br>– Output Assembly: 110 bytes<br>• 0x03: 16 bytes<br>– Input Assemby: 318 bytes<br>– Output Assembly: 174 bytes<br>• 0x04: 32 bytes<br>– Input Assemby: 446 bytes<br>– Output Assembly: 302 bytes |

### Mapping: Port Configuration

| Byte (Offset) | Content |
|---|---|
| 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+11 | Failsafe Mode – Pin 4 (DO) |

Legend:

| | | | |
|---|---|---|---|
| • Mode | Operating mode of the port (pin 4) | 1 BYTE | • 0x00: Disabled<br>• 0x01: digital input (pin 4)<br>• 0x02: digital output (pin 4)]<br>• 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<br>• 0x01: 2 ms<br>• 0x02: 4 ms<br>• 0x03: 8 ms<br>• 0x04: 16 ms<br>• 0x05: 32 ms<br>• 0x06: 64 ms<br>• 0x07: 128 ms |
| • Swap | Swap arrangement of bytes in process data (EtherNet/IP uses little-endian format (Intel), IO-Link uses big-endian format (Motorola)) | 1 BYTE | • 0x00: Enabled<br>• 0x01: Disabled |
| • Validation / Data Storage | Validation mode / data storage mode | 1 BYTE | • 0x00: No device check and clear<br>• 0x01: Type compatible V1.0 device<br>• 0x02: Type compatible V1.1 device<br>• 0x03: Type compatible V1.1 device with Backup + Restore<br>• 0x04: Type compatible V1.1 device with Restore |
| • Vendor ID | Vendor ID of the manufacturer of the IO-Link device<br>Vendor ID = 0x1234<br>• Vendor ID (MSB): 0x12<br>• Vendor ID (LSB): 0x34 | 2 BYTE | • 0x00…0x7FFF |
| • Device ID | Device ID of the IO-Link device<br>Device ID = 0x123456<br>• Device ID (MSB): 0x12<br>• Device ID 0x34<br>• Device ID (LSB): 0x56 | 3 BYTE | • 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: No failsafe<br>• 0x01: Failsafe Reset Value<br>• 0x02: Failsafe Old Value<br>• 0x03: 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: Failsafe Reset Value<br>• 0x01: Failsafe Old Value<br>• 0x02: Failsafe Set Value |

## 11.2.2 Cyclic data

### 11.2.2.1 Input Assembly (Instance 100)

| Byte (Offset) | Bit | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0…1 | Digital Input (→ Mapping: Digital Input 86) | | | | | | | |
| 2…3 | Status Information (→ Mapping: Status Information 87) | | | | | | | |
| 4…45 | Acyclic Command Channel: Response (→ Response channel 94) | | | | | | | |
| 46…47 | Port X01: Mapping: Port Qualifier Information (PQI) (→ 88) | | | | | | | |
| 48…63 | Port X01: Mapping: IO-Link information + events (→ 88) | | | | | | | |
| 64…65 | Port X02: Mapping: Port Qualifier Information (PQI) (→ 88) | | | | | | | |
| 66…81 | Port X02: Mapping: IO-Link information + events (→ 88) | | | | | | | |
| 82…83 | Port X03: Mapping: Port Qualifier Information (PQI) (→ 88) | | | | | | | |
| 84…99 | Port X03: Mapping: IO-Link information + events (→ 88) | | | | | | | |
| 100…101 | Port X04: Mapping: Port Qualifier Information (PQI) (→ 88) | | | | | | | |
| 102…117 | Port X04: Mapping: IO-Link information + events (→ 88) | | | | | | | |
| 118…119 | Port X05: Mapping: Port Qualifier Information (PQI) (→ 88) | | | | | | | |
| 120…135 | Port X05: Mapping: IO-Link information + events (→ 88) | | | | | | | |
| 136…137 | Port X06: Mapping: Port Qualifier Information (PQI) (→ 88) | | | | | | | |
| 138…153 | Port X06: Mapping: IO-Link information + events (→ 88) | | | | | | | |
| 154…155 | Port X07: Mapping: Port Qualifier Information (PQI) (→ 88) | | | | | | | |
| 156…171 | Port X07: Mapping: IO-Link information + events (→ 88) | | | | | | | |
| 172…173 | Port X08: Mapping: Port Qualifier Information (PQI) (→ 88) | | | | | | | |
| 174…189 | Port X08: Mapping: IO-Link information + events (→ 88) | | | | | | | |
| 190 | Port X01: Input data IO-Link (n bytes) [1] | | | | | | | |
| 190+n | Port X02: Input data IO-Link (n bytes) [1] | | | | | | | |
| 190+2n | Port X03: Input data IO-Link (n bytes) [1] | | | | | | | |
| 190+3n | Port X04: Input data IO-Link (n bytes) [1] | | | | | | | |
| 190+4n | Port X05: Input data IO-Link (n bytes) [1] | | | | | | | |
| 190+5n | Port X06: Input data IO-Link (n bytes) [1] | | | | | | | |
| 190+6n | Port X07: Input data IO-Link (n bytes) [1] | | | | | | | |
| 190+7n | Port X08: Input data IO-Link (n bytes) [1] | | | | | | | |

[1] n: {2, 4, 8, 16, 32}; n is determined by the parameter [Port Process Data Size] in the configuration assembly (→ Configuration Assembly (Instance 199) 80)

#### 11.2.2.2 Input Assembly (Instance 101)

| Byte (Offset) | Bit | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0…1 | Digital Input (→ Mapping: Digital Input 86) | | | | | | | |
| 2…3 | Status Information (→ Mapping: Status Information 87) | | | | | | | |
| 4…45 | Acyclic Command Channel: Response (→ Response channel 94) | | | | | | | |
| 46…47 | Port X01: Mapping: Port Qualifier Information (PQI) (→ 88) | | | | | | | |
| 48…49 | Port X02: Mapping: Port Qualifier Information (PQI) (→ 88) | | | | | | | |
| 50…51 | Port X03: Mapping: Port Qualifier Information (PQI) (→ 88) | | | | | | | |
| 52…55 | Port X04: Mapping: Port Qualifier Information (PQI) (→ 88) | | | | | | | |
| 54…53 | Port X05: Mapping: Port Qualifier Information (PQI) (→ 88) | | | | | | | |
| 56…57 | Port X06: Mapping: Port Qualifier Information (PQI) (→ 88) | | | | | | | |
| 58…59 | Port X07: Mapping: Port Qualifier Information (PQI) (→ 88) | | | | | | | |
| 60…61 | Port X08: Mapping: Port Qualifier Information (PQI) (→ 88) | | | | | | | |
| 62 | Port X01: Input data IO-Link (n bytes) [1] | | | | | | | |
| 62+n | Port X02: Input data IO-Link (n bytes) [1] | | | | | | | |
| 62+2n | Port X03: Input data IO-Link (n bytes) [1] | | | | | | | |
| 62+3n | Port X04: Input data IO-Link (n bytes) [1] | | | | | | | |
| 62+4n | Port X05: Input data IO-Link (n bytes) [1] | | | | | | | |
| 62+5n | Port X06: Input data IO-Link (n bytes) [1] | | | | | | | |
| 62+6n | Port X07: Input data IO-Link (n bytes) [1] | | | | | | | |
| 62+7n | Port X08: Input data IO-Link (n bytes) [1] | | | | | | | |

[1] n: {2, 4, 8, 16, 32}; n is determined by the parameter [Port Process Data Size] in the configuration assembly (→ Configuration Assembly (Instance 199) 80)

#### 11.2.2.3 Input Assembly (Instance 102)

| Byte (Offset) | Bit | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0…1 | Digital Input (→ Mapping: Digital Input 🗋 86) | | | | | | | |
| 2…3 | Status Information (→ Mapping: Status Information 🗋 87) | | | | | | | |
| 4…5 | Port X01: Port Qualifier Information (→ Mapping: Port Qualifier Information (PQI) 🗋 88) | | | | | | | |
| 6…7 | Port X02: Port Qualifier Information (→ Mapping: Port Qualifier Information (PQI) 🗋 88) | | | | | | | |
| 8…9 | Port X03: Port Qualifier Information (→ Mapping: Port Qualifier Information (PQI) 🗋 88) | | | | | | | |
| 10…11 | Port X04: Port Qualifier Information (→ Mapping: Port Qualifier Information (PQI) 🗋 88) | | | | | | | |
| 4…5 | Port X01: Port Qualifier Information (→ Mapping: Port Qualifier Information (PQI) 🗋 88) | | | | | | | |
| 6…7 | Port X02: Port Qualifier Information (→ Mapping: Port Qualifier Information (PQI) 🗋 88) | | | | | | | |
| 8…9 | Port X03: Port Qualifier Information (→ Mapping: Port Qualifier Information (PQI) 🗋 88) | | | | | | | |
| 10…11 | Port X04: Port Qualifier Information (→ Mapping: Port Qualifier Information (PQI) 🗋 88) | | | | | | | |
| 12…13 | Port X05: Port Qualifier Information (→ Mapping: Port Qualifier Information (PQI) 🗋 88) | | | | | | | |
| 14…15 | Port X06: Port Qualifier Information (→ Mapping: Port Qualifier Information (PQI) 🗋 88) | | | | | | | |
| 16…17 | Port X07: Port Qualifier Information (→ Mapping: Port Qualifier Information (PQI) 🗋 88) | | | | | | | |
| 18…19 | Port X08: Port Qualifier Information (→ Mapping: Port Qualifier Information (PQI) 🗋 88) | | | | | | | |
| 20 | Port X01: Input data IO-Link (n bites) | | | | | | | |
| 20+n | Port X02: Input data IO-Link (n bites) | | | | | | | |
| 20+2n | Port X03: Input data IO-Link (n bites) | | | | | | | |
| 20+3n | Port X04: Input data IO-Link (n bites) | | | | | | | |
| 20+4n | Port X05: Input data IO-Link (n bites) | | | | | | | |
| 20+5n | Port X06: Input data IO-Link (n bites) | | | | | | | |
| 20+6n | Port X07: Input data IO-Link (n bites) | | | | | | | |
| 20+7n | Port X08: Input data IO-Link (n bites) | | | | | | | |

## Mapping: Digital Input

Legend:

| | | | |
|---|---|---|---|
| • DI Pin 4 | Signal status of the digital input on Pin 4 of the port | 1 BIT | • 0x0: LOW<br>• 0x1: HIGH |
| • DI Pin 2 | Signal status of the digital input on Pin 2 of the port | 1 BIT | • 0x0: LOW<br>• 0x1: HIGH |

## Mapping: Status Information

Legend:

| | | | |
|---|---|---|---|
| • SC / OL | Short Circuit / Overload: Indication of a short circuit or of an overvoltage in the IO-Link port | 1 BIT | • 0x0: No error<br>• 0x1: Short circuit or overvoltage |
| • SENS PWR | Status of the voltage supply US | 1 BIT | • 0x0: No error<br>• 0x1: Excessive current on US (> 3.6 V) |

### Mapping: Port Qualifier Information (PQI)

### Mapping: IO-Link information + events

| Byte (Offset) | Bit | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| n | Vendor ID (LSB) | | | | | | | |
| n+1 | Vendor ID (MSB) | | | | | | | |
| n+2 | Device ID (LSB) | | | | | | | |
| n+3 | Device ID | | | | | | | |
| n+4 | Device ID (MSB) | | | | | | | |
| n+5 | reserved | | | | | | | |
| n+6 | Event 1: Mode | | Event 1: Type | | Event 1: Src | Event 1: Instance | | |
| n+7 | Event Code 1 (LSB) | | | | | | | |
| n+8 | Event Code 1 (MSB) | | | | | | | |
| n+9 | Event 2: Mode | | Event 2: Type | | Event 2: Src | Event 2: Instance | | |
| n+10 | Event 2 (LSB) | | | | | | | |
| n+11 | Event 2 (MSB) | | | | | | | |
| n+12 | Event 3: Mode | | Event 3: Type | | Event 3: Src | Event 3: Instance | | |
| n+13 | Event 3 (LSB) | | | | | | | |
| n+14 | Event 3 (MSB) | | | | | | | |
| n+15 | reserved | | | | | | | |

Legend:

| | | | |
|---|---|---|---|
| • VID | Vendor ID of the connected IO-Link device<br>VID = 0x1234<br>• DID (MSB): 0x12<br>• DID (LSB): 0x34 | 2 BYTE | 0x0000...0x7FFF |
| • DID | Device ID of the connected IO-Link device<br>DID = 0x123456<br>• DID (MSB): 0x12<br>• DID: 0x34<br>• DID (LSB): 0x56 | 3 BYTE | 0x000000...0xFFFFFF |
| • Event m: Mode | Mode Mode of the event | 2 BIT | • 0x0: reserved<br>• 0x1: One-time event<br>• 0x2: Event has disappeared<br>• 0x3: Event has appeared |
| • Event m: Type | Type: category of the event | 2 BIT | • 0x0: reserved<br>• 0x1: Notification<br>• 0x2: Warning<br>• 0x3: Error |
| • Event m: Src | Src: Source of the event | 1 BIT | • 0x0: IO-Link device<br>• 0x1: IO-Link master |
| • Event m: Instance | Instance: Trigger of the event | 3 BIT | • 0x0: Unknown<br>• 0x1...0x3: reserved<br>• 0x4: Application<br>• 0x5...0x7: reserved |

| | | | |
|---|---|---|---|
| • Event m: Code | Code Event code<br>Code = 0x1234<br>• Code (MSB): 0x12<br>• Code (LSB): 0x34 | 2 BYTE | depending on the device (→ IODD description of the IO-Link device) |

#### 11.2.2.4 Output Assembly (Instance 150)

| Byte (Offset) | Bit | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 2…3 | reserved | | | | | | | |
| 4…45 | Acyclic Command Channel: Request (→ Request channel 93) | | | | | | | |

#### 11.2.2.5 Output Assembly (Instance 151)

## Mapping: Digital output

Legend:

| | | | |
|---|---|---|---|
| • DO Pin 4 | Signal status of the digital output on pin 4 of the port | 1 BIT | • 0x0: LOW<br>• 0x1: HIGH |

## 11.2.3 Acyclical data

### 11.2.3.1 Acyclic command channel

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

| Object | Content | Byte | Access |
|---|---|---|---|
| Output assembly | Request channel (EtherNet/IP controller ➞ IO-Link master)<br>Request channel (→ 🗋 93) | 4…45 | rw [1] |
| Input assembly | Response channel (IO-Link master ➞ EtherNet/IP controller)<br>Response channel (→ 🗋 94) | 4…45 | ro [2] |

[1] Read and write

[2] Read only

### Request channel

| Byte (offset) | Bit | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 4 | Port No. (LSB) | | | | | | | |
| 5 | Port No. (MSB) | | | | | | | |
| 6 | Index (LSB) | | | | | | | |
| 7 | Index (MSB) | | | | | | | |
| 8 | Subindex (LSB) | | | | | | | |
| 9 | Subindex (MSB) | | | | | | | |
| 10 | Trigger | Command ID | | | | | | |
| 11 | Data Length | | | | | | | |
| 12 | Data (Byte 0) | | | | | | | |
| 13 | Data (Byte 1) | | | | | | | |
| … | … | | | | | | | |
| 43 | Data (Byte 31) | | | | | | | |
| 44 | reserved | | | | | | | |
| 45 | reserved | | | | | | | |

Legend:

| | | | |
|---|---|---|---|
| • Port No. | IO-Link Port<br>Port No. 0x1234<br>• Port No. (MSB): 0x12<br>• Port No. (LSB): 0x34 | 1 WORD | • 0x0001: Port 1<br>• 0x0002: Port 2<br>• 0x0003: Port 3<br>• 0x0004: Port 4<br>• 0x0005: Port 5<br>• 0x0006: Port 6<br>• 0x0007: Port 7<br>• 0x0008: Port 8 |
| • Index | Index of the ISDU object<br>Index: 0x1234<br>• Index (MSB): 0x12<br>• Index (LSB): 0x34 | 1 WORD | 0x0000…0xFFFF |
| • Subindex | Subindex of the ISDU object<br>Subindex 0x1234<br>• Subindex (MSB): 0x12<br>• Subindex (LSB): 0x34 | 1 WORD | 0x0000…0xFFFF |
| • Trigger | Control of the command execution | 1 BIT | • 0x0: no action<br>• 0x1: Execute command |
| • Command ID | Command number | 7 BIT | • 0x01: Get<br>• 0x02: Write |
| • Data Length | Number of bytes that contain relevant user data | 1 BYTE | • 0x00: 0 bytes<br>…<br>• 0x20: 32 bytes |
| • Data (byte n) | User data (byte n) | 1 BYTE | Pro byte:<br>• 0x00…0xFF |

### Response channel

| Byte (offset) | Bit | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 4 | Port No. (LSB) | | | | | | | |
| 5 | Port No. (MSB) | | | | | | | |
| 6 | Index (LSB) | | | | | | | |
| 7 | Index (MSB) | | | | | | | |
| 8 | Subindex (LSB) | | | | | | | |
| 9 | Subindex (MSB) | | | | | | | |
| 10 | Handshake | Command ID | | | | | | |
| 11 | Result | | | | | | | |
| 12 | Data Length | | | | | | | |
| 13 | Data (Byte 0) \| Error Code | | | | | | | |
| 14 | Data (Byte 1) \| Additional Code | | | | | | | |
| | … | | | | | | | |
| 44 | Data (Byte 31) | | | | | | | |
| 45 | reserved | | | | | | | |

Legend:

| | | | |
|---|---|---|---|
| • Port No. | IO-Link Port<br>Port No. 0x1234<br>• Port No. (MSB): 0x12<br>• Port No. (LSB): 0x34 | 1 WORD | • 0x0001: Port 1<br>• 0x0002: Port 2<br>• 0x0003: Port 3<br>• 0x0004: Port 4<br>• 0x0005: Port 5<br>• 0x0006: Port 6<br>• 0x0007: Port 7<br>• 0x0008: Port 8 |
| • Index | Index of the ISDU object<br>Index: 0x1234<br>• Index (MSB): 0x12<br>• Index (LSB): 0x34 | 1 WORD | 0x0000…0xFFFF |
| • Subindex | Subindex of the ISDU object<br>Subindex 0x1234<br>• Subindex (MSB): 0x12<br>• Subindex (LSB): 0x34 | 1 WORD | 0x0000…0xFFFF |
| • Handshake | Validity of the IO-Link response data | 1 BIT | • 0x0: data invalid<br>• 0x1: data valid |
| • Command ID | Command number | 7 BIT | • 0x01: Get<br>• 0x02: Write |
| • Result | Status of the command processing | 1 BYTE | • 0x00: OK<br>• 0x0F: OK, data user read > 32 bytes<br>• 0xFF: Error |
| • Data Length | Number of bytes that contain relevant user data | 1 BYTE | • 0x00: 0 bytes<br>…<br>• 0x20: 32 bytes |
| • Data (Byte 0) \| Error Code | User data (byte 0) or error codes | 1 BYTE | • User data: 0x00…0xFF<br>• Error Code: Error Code (→ 96) |
| • Data (Byte 1) \| Additional Code | User data (byte 1) or additional error codes | 1 BYTE | • User data: 0x00…0xFF<br>• Additional Code: Additional Code (→ 96) |

| | | | |
|---|---|---|---|
| • Data (byte n) | User data (byte n) | 1 BYTE | Pro Byte:<br>0x00…0xFF |

### Error Code

| 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 prohibit command execution) |
| 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 Consider additional code (Additional Code (→ 96)) |

### Additional Code

Additional Code only valid if Error Code = 0x80 (→ Error Code 96).

| Code | Name | Description |
|---|---|---|
| 0x00 | APP_DEV | Error in the device application - no details |
| 0x11 | IDX_NOTAVAIL | Index not available |
| 0x12 | SUBIDX_NOTAVAIL | Subindex not available |
| 0x20 | SERV_NOTAVAIL | Service temporarily unavailable |
| 0x21 | SERV_NOTAVAIL_LOCCTRL | Service temporarily unavailable - local |
| 0x22 | SERV_NOTAVAIL_DEVCTRL | Service temporarily unavailable - device |
| 0x23 | IDX_NOT_WRITEABLE | Access denied |
| 0x30 | PAR_VALOUTOFRNG | Parameter value outside the valid range |
| 0x31 | PAR_VALGTLIM | Parameter value above the threshold |
| 0x32 | PAR_VALLTLIM | Parameter value below the threshold |
| 0x33 | VAL_LENOVRRUN | Parameter length exceeded |
| 0x34 | VAL_LENUNDRUN | Parameter length not reached |
| 0x35 | FUNC_NOTAVAIL | Function not available |
| 0x36 | FUNC_UNAVAILTEMP | Function temporarily unavailable |
| 0x40 | PAR_SETINVALID | Invalid parameter set |
| 0x41 | PAR_SETINCONSIST | Inconsistent parameter set |
| 0x82 | APP_DEVNOTRDY | Application not ready |

#### 11.2.3.2 Acyclic commands

**Set Mode (0x10)**

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

**Command request**

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

Legend:

| | | | |
|---|---|---|---|
| • Port No. | IO-Link Port<br>Port No. 0x1234<br>• Port No. (MSB): 0x12<br>• Port No. (LSB): 0x34 | 1 WORD | • 0x0001: Port 1<br>• 0x0002: Port 2<br>• 0x0003: Port 3<br>• 0x0004: Port 4<br>• 0x0005: Port 5<br>• 0x0006: Port 6<br>• 0x0007: Port 7<br>• 0x0008: Port 8 |
| • Trigger | Control of the command execution | 1 BIT | • 0x0: no action<br>• 0x1: Execute command |
| • Target Mode | Operating mode of the IO-Link port | 1 BYTE | • 0x00: deactivated<br>• 0x01: Digital input (DI)<br>• 0x02: Digital output (DO)<br>• 0x03: IO-Link |

### Command response

| Byte (offset) | Bit | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 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 | reserved | | | | | | | |
| … | … | | | | | | | |
| 45 | reserved | | | | | | | |

Legend:

| | | | |
|---|---|---|---|
| • Port No. | IO-Link Port<br>Port No. 0x1234<br>• Port No. (MSB): 0x12<br>• Port No. (LSB): 0x34 | 1 WORD | • 0x0001: Port 1<br>• 0x0002: Port 2<br>• 0x0003: Port 3<br>• 0x0004: Port 4<br>• 0x0005: Port 5<br>• 0x0006: Port 6<br>• 0x0007: Port 7<br>• 0x0008: Port 8 |
| • Handshake | Validity of the IO-Link response data | 1 BIT | • 0x0: data invalid<br>• 0x1: data valid |
| • Result | Status of the command processing | 1 BYTE | • 0x00: OK<br>• 0x0F: OK, data user read > 32 bytes<br>• 0xFF: Error |
| • Target Mode | Operating mode of the IO-Link port | 1 BYTE | • 0x00: deactivated<br>• 0x01: Digital input (DI)<br>• 0x02: Digital output (DO)<br>• 0x03: IO-Link |

### Set validation ID / data storage (0x20)

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

### Command request

| Byte (offset) | Bit | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 4 | Port No. (LSB) | | | | | | | |
| 5 | Port No. (MSB) | | | | | | | |
| 6 | reserved | | | | | | | |
| 7 | reserved | | | | | | | |
| 8 | reserved | | | | | | | |
| 9 | reserved | | | | | | | |
| 10 | Trigger | 0x20 | | | | | | |
| 11 | Validation ID | | | | | | | |
| 12 | reserved | | | | | | | |
| … | … | | | | | | | |
| 45 | reserved | | | | | | | |

Legend:

| | | | |
|---|---|---|---|
| • Port No. | IO-Link Port<br>Port No. 0x1234<br>• Port No. (MSB): 0x12<br>• Port No. (LSB): 0x34 | 1 WORD | • 0x0001: Port 1<br>• 0x0002: Port 2<br>• 0x0003: Port 3<br>• 0x0004: Port 4<br>• 0x0005: Port 5<br>• 0x0006: Port 6<br>• 0x0007: Port 7<br>• 0x0008: Port 8 |
| • Trigger | Control of the command execution | 1 BIT | • 0x0: no action<br>• 0x1: Execute command |
| • Validation ID | Validation type and data storage behaviour of the port | 1 BYTE | • 0x00: No check<br>• 0x01: Type compatible V1.0 device<br>• 0x02: Type compatible V1.1 device<br>• 0x03: Type compatible V1.1 device with Backup + Restore<br>• 0x04: Type compatible V1.1 device with Restore |

### Command response

| Byte (offset) | Bit | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 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 | reserved | | | | | | | |
| … | … | | | | | | | |
| 45 | reserved | | | | | | | |

Legend:

| | | | |
|---|---|---|---|
| • Port No. | IO-Link Port<br>Port No. 0x1234<br>• Port No. (MSB): 0x12<br>• Port No. (LSB): 0x34 | 1 WORD | • 0x0001: Port 1<br>• 0x0002: Port 2<br>• 0x0003: Port 3<br>• 0x0004: Port 4<br>• 0x0005: Port 5<br>• 0x0006: Port 6<br>• 0x0007: Port 7<br>• 0x0008: Port 8 |
| • Handshake | Validity of the IO-Link response data | 1 BIT | • 0x0: data invalid<br>• 0x1: data valid |
| • Result | Status of the command processing | 1 BYTE | • 0x00: OK<br>• 0x0F: OK, data user read > 32 bytes<br>• 0xFF: Error |
| • Validation ID | Validation type and data storage behaviour of the port | 1 BYTE | • 0x00: No check<br>• 0x01: Type compatible V1.0 device<br>• 0x02: Type compatible V1.1 device<br>• 0x03: Type compatible V1.1 device with Backup + Restore<br>• 0x04: Type compatible V1.1 device with Restore |

### Set fail-safe data pattern (0x30)

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

### Command response

| Byte (Offset) | Bit | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 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 | reserved | | | | | | | |
| … | … | | | | | | | |
| 45 | reserved | | | | | | | |

| | | | |
|---|---|---|---|
| • Handshake | Validity of the IO-Link response data | 1 BIT | • 0x0: data invalid<br>• 0x1: data valid |
| • Result | Status of the command processing | 1 BYTE | • 0x00: OK<br>• 0x0F: OK, data user read > 32 bytes<br>• 0xFF: Error |
| • Fail-safe mode | Fail-safe mode for the outputs of the IO-Link ports in case of an interruption of the EtherNet/IP connection | 1 BYTE | • 0x00: No Fail-safe<br>• 0x01: Fail-safe - Reset Value<br>• 0x02: Fail-safe - Old Value<br>• 0x03: Fail-safe - Pattern |

## 11.2.4 Object directory

### 11.2.4.1 CIP object classes

Supported object classes:

| Class Code | Name | Description |
|---|---|---|
| 0x01 | Identity Object | Identity Object (Class Code: 0x01) (→ 🗋 103) |
| 0x02 | Message Router Object | Message Router Object (Class Code: 0x02) (→ 🗋 104) |
| 0x04 | Assembly Object | Assembly Object (Class Code: 0x04) (→ 🗋 105) |
| 0x06 | Connection Manager Object | Connection Manager Object (Class Code: 0x06) (→ 🗋 106) |
| 0x47 | Device Level Ring Object | Device Level Ring Object (Class Code: 0x47) (→ 🗋 107) |
| 0x48 | Quality Of Service Object | Quality Of Service Object (Class Code: 0x48) (→ 🗋 108) |
| 0x80 | IO-Link Request Object | IO-Link Request Object (Class Code: 0x80) (→ 🗋 109) |
| 0xF5 | TCP/IP Object | TCP/IP Object (Class Code: 0xF5) (→ 🗋 113) |
| 0xF6 | Ethernet Link Object | Ethernet Link Object (Class Code: 0xF6) (→ 🗋 116) |

### 11.2.4.2 CIP class and instance services

Supported class and instance services:

| Service code | Name | Description |
|---|---|---|
| 0x01 | Get Attribute All | Read all attributes |
| 0x02 | Set Attribute All | Write all attributes |
| 0x05 | Reset | Reset |
| 0x09 | Delete | Delete |
| 0x0E | Get Attribute Single | Read single attribute |
| 0x10 | Set Attribute Single | Write single attribute |
| 0x4B | Read ISDU | Read ISDU |
| 0x4C | Write ISDU | Write ISDU |
| 0x4D | Write Failsafe Pattern | Write fail-safe values |
| 0x4E | Forward Close | Close connection |
| 0x54 | Forward Open | Open new connection |

#### 11.2.4.3 Identity Object (Class Code: 0x01)

**Class attributes**

| Attr. ID | Access | Name | Data type | Description | Value |
|---|---|---|---|---|---|
| 1 | Get | Revision | UINT | Revision of the object | 2.003 |
| 2 | Get | Max. Instance | UINT | Max. number of instances of the object | 1 |
| 6 | Get | Max. ID Number Class Attributes | UINT | Max. ID number of a class attribute | 7 |
| 7 | Get | Max. ID Number Instance Attributes | UINT | Max. ID number of the instance attribute | 9 |

**Instance attributes**

| Attr. ID | Access | Name | Data type | Description | Default |
|---|---|---|---|---|---|
| 1 | Get | Vendor ID | UINT | Manufacturer ID | 322 |
| 2 | Get | Device Type | UINT | Device type | 12 |
| 3 | Get | Product Code | UINT | Product code of the device | 1008 |
| 4 | Get | Revision | STRUCT | Revision of the device | 2.003 |
| | | • Major Revision | USINT | Major revision (1...127) | 2 |
| | | • Minor Revision | USINT | Minor revision (3 digits) | 003 |
| 5 | Get | Status | WORD | Overall status of the device | |
| 6 | Get | Serial Number | UDINT | Serial number of the device | -- |
| 7 | Get | Product Name | SHORT STRING | Product name of the device | IO-Link Master DL EIP 8P IP69K |
| 8 | Get | State | USINT | State of the device (State machine)<br>• 0: Nonexistent<br>• 1: Device Self Testing<br>• 2: Standby<br>• 3: Operational<br>• 4: Major Recoverable Fault<br>• 5: Major Unrecoverable Fault<br>• 6-254: Reserved<br>• 255: Default for “Get_Attributes_All” service | |

**Services**

| Code | Service | Class | Instance | Description |
|---|---|---|---|---|
| 0x01 | Get Attribute All | Yes | Yes | Read all attributes |
| 0x05 | Reset | Yes | Yes | Reset |
| 0x0E | Get Attribute Single | Yes | Yes | Read 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 (mandatory for all EtherNet/IP devices).
- 1: restore factory settings and reboot the device.

#### 11.2.4.4 Message Router Object (Class Code: 0x02)

**Class attributes**

| 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 | Max. ID Number Class Attributes | UINT | Max. ID number of a class attribute | 7 |
| 7 | Get | Max. ID Number Instance Attributes | UINT | Max. ID number of the instance attribute | 0 |

**Instance attributes**

The object class has no instance attributes.

**Services**

| Code | Service | Class | Instance | Description |
|---|---|---|---|---|
| 0x0E | Get Attribute Single | Yes | Yes | Read single attribute |

#### 11.2.4.5 Assembly Object (Class Code: 0x04)

**Class attributes**

| 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 | Max. ID Number Class Attributes | UINT | Max. ID number of the class attribute | 7 |
| 7 | Get | Max. ID Number Instance Attributes | UINT | Max. ID number of the instance attribute | 4 |

**Instance attributes**

The following object instances are available:

**Services**

| Code | Service | Class | Instance | Description |
|---|---|---|---|---|
| 0x0E | Get Attribute Single | Yes | Yes | Read single attribute |
| 0x10 | Set Attribute Single | No | Yes | Write single attribute |

#### 11.2.4.6 Connection Manager Object (Class Code: 0x06)

**Class attributes**

| 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 | Max. ID Number Class Attributes | UINT | Max. ID number of a class attribute | 7 |
| 7 | Get | Max. ID Number Instance Attributes | UINT | Max. ID number of the instance attribute | 0 |

**Instance attributes**

The object class has no instance attributes.

**Services**

| Code | Service | Class | Instance | Description |
|---|---|---|---|---|
| 0x0E | Get Attribute Single | Yes | Yes | Read single attribute |
| 0x10 | Set Attribute Single | No | Yes | Write single attribute |
| 0x4E | Forward Close | Yes | No | Close connection |
| 0x54 | Forward Open | Yes | No | Open new connection |

### 11.2.4.7 Device Level Ring Object (Class Code: 0x47)

#### Class attributes

| 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 | Max. ID Number Class Attributes | UINT | Max. ID number of the class attribute | 7 |
| 7 | Get | Max. ID Number Instance Attributes | UINT | Max. ID number of the instance attribute | 12 |

#### Instance attributes

| Attr. ID | Access | Name | Data type | Description | Default |
|---|---|---|---|---|---|
| 1 | Get | Network topology | USINT | Current network topology<br>• 0: linear | 0 |
| 2 | Get | Network status | USINT | Current network status<br>• 0: OK | 0 |
| 10 | Get | Active Supervisor | STRUCT | Identification of the supervisor | 0 |
| | | | • UDINT | IP address | |
| | | | • ARRAY(6) of USINT | MAC address of the active supervisor | |
| 12 | Get | Capability Flags | DWORD | DLR capability of the device<br>• 0x82: Beacon based Ring Node, Flush Table Frame support | 0x82 |

#### Services

| Code | Service | Class | Instance | Description |
|---|---|---|---|---|
| 0x01 | Get Attribute All | No | Yes | Read all attributes |
| 0x0E | Get Attribute Single | Yes | Yes | Read single attribute |

#### 11.2.4.8 Quality Of Service Object (Class Code: 0x48)

### Class attributes

| 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 of the object | 1 |
| 6 | Get | Max. ID Number Class Attributes | UINT | Max. ID number of a class attribute | 7 |
| 7 | Get | Max. ID Number Instance Attributes | UINT | Max. ID number of the instance attribute | 8 |

### Instance attributes

| Attr. ID | Access | Name | Data type | Description | Default |
|---|---|---|---|---|---|
| 1 | Get | 802.1Q Tag Enable | USINT | Current network topology | 0: deactivated |
| 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 |

### Services

| Code | Service | Class | Instance | Description |
|---|---|---|---|---|
| 0x01 | Get Attribute All | Yes | Yes | Read all attributes |

#### 11.2.4.9 IO-Link Request Object (Class Code: 0x80)

**Class attributes**

| 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 | Max. ID Number Class Attributes | UINT | Max. ID number of a class attribute | 8 |

**Instance attributes**

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

**Services**

### Read ISDU

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

### Request

CIP Attributes 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 | Description |
|---|---|---|---|
| CIP Class ID | UINT | 0x80 | IO-Link acyclic access |
| CIP Instance ID | UINT | 0x01 | IO-Link master |
| CIP attributes | USINT | 0x01…0x08 | Port number |
| CIP Service Code ID | USINT | 0x4B | “Read ISDU” service |
| CIP User Specific Service Data | UINT | 0x0000…0xFFFF | ISDU object index |
| | USINT | 0x00…0xFF | ISDU object subindex |

### Response

#### *Fault-free processing*

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 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 attributes | USINT | 0x01…0x08 | Port number |
| CIP Service Code ID | USINT | 0x4B | “Read ISDU” service |
| CIP Error Code | USINT | 0x00 | Fault-free processing |
| 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.

#### *Fault*

If an error occurs while executing the service (CIP Error Code <> 0), an extended error code will be transmitted (CIP Extended Error Code). If the CIP Error Code = 0x1E, the CIP Extended Error Code is = 0x00 and the IO-Link Error Code and the IO-Link Additional Code will be transmitted in the CIP User Specific Service Data area. 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 attributes | USINT | 0x01…0x08 | Port number |
| CIP Service Code ID | USINT | 0x4B | “Read ISDU” service |
| CIP Error Code | USINT | <> 0x00 | Error code (see below) |
| CIP Extended Error Code | USINT | 0x00 | Extended error code |
| CIP User Specific Service Data | USINT | <> 0x00 | IO-Link Error Code: Error Code (→ 🗋 96) |

| CIP Format | Data type | MSG Config | Description |
|---|---|---|---|
| CIP User Specific Service Data | USINT | <> 0x00 | IO-Link Additional Code: Additional Code (→ 96) |

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 CIP User Specific Service Data contain the IO-Link error code and an additional code returned by the IO-Link master. |

### Write ISDU

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

### Request

CIP Attributes determines the IO-Link port to which the IO-Link device is connected. The area "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. The value to be assigned to the parameter follows byte by byte.

| 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 | 0x01…0x08 | Port number |
| CIP Service Code ID | USINT | 0x4C | “Write ISDU” service |
| CIP User Specific Service Data | UINT | 0x0000…0xFFFF | ISDU object index |
| | USINT | 0x00…0xFF | ISDU object subindex |
| | USINT | 0x00…0xFF | ISDU data (byte 0) |
| | USINT | 0x00…0xFF | ISDU data (byte 1) |

### Response

#### *Fault-free processing*

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 attributes | USINT | 0x01…0x08 | Port number |
| CIP Service Code ID | USINT | 0x4C | “Write ISDU” service |
| CIP Error Code | USINT | 0x00 | Fault-free processing |
| CIP Extended Error Code | USINT | 0x00 | -- |

#### *Fault*

If an error occurs while executing the service (CIP Error Code <> 0), an extended error code is transmitted (CIP Extended Error Code). If the CIP Error Code = 0x1E, the CIP Extended Error Code is = 0x00 and the IO-Link Error Code and the IO-Link Additional Code will be transmitted in the CIP User Specific Service Data area. 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 attributes | USINT | 0x01…0x08 | Port number |
| CIP Service Code ID | USINT | 0x4C | “Write ISDU” service |
| CIP Error Code | USINT | <> 0x00 | Error code (see below) |
| CIP Extended Error Code | USINT | 0x00 | Extended error code |
| CIP User Specific Service Data | USINT | <> 0x00 | IO-Link Error Code: Error Code (→ 🗋 96) |
| | USINT | <> 0x00 | IO-Link Additional Code: Additional Code (→ 🗋 96) |

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 CIP User Specific Service Data contain the IO-Link error code and an additional code returned by the IO-Link master. |

### Write Failsafe Pattern

Write Failsafe Pattern can be used to write the failsafe value of an IO-Link port.

### Request

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 attributes | USINT | 0x01…0x08 | Port number |
| CIP Service Code ID | USINT | 0x4D | “Write Failsafe Pattern” service |
| CIP User Specific Service Data | USINT | • 0x00: No Fail-safe<br>• 0x01: Fail-safe Reset Value<br>• 0x02: Fail-safe Old Value<br>• 0x03: Fail-safe with Pattern | Fail-safe Mode |
| | USINT | 0x00…0xFF | Fail-safe Pattern (MSB) |
| | USINT | 0x00…0xFF | Fail-safe Pattern (LSB) |

### Response

#### *Fault-free processing*

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 attributes | USINT | 0x01…0x08 | Port number |

| CIP Format | Data type | MSG Config | Description |
|---|---|---|---|
| CIP Service Code ID | USINT | 0x4D | “Write Failsafe Pattern” service |
| CIP Error Code | USINT | 0x00 | Fault-free processing |
| CIP Extended Error Code | USINT | 0x00 | -- |

***Fault***

If an error occurs while executing the service (CIP Error Code <> 0), an extended error code will be transmitted (CIP Extended Error Code). 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 attributes | USINT | 0x01…0x08 | Port number |
| 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 CIP User Specific Service Data contain the IO-Link error code and an additional code returned by the IO-Link master. |
| 0x0F | Insufficient access rights |

#### 11.2.4.10 TCP/IP Object (Class Code: 0xF5)

**Class attributes**

| 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 |
| 3 | Get | Number of instances | UINT | Number of instances of the object | 1 |
| 6 | Get | Max. ID Number Class Attributes | UINT | Max. ID number of a class attribute | 7 |
| 7 | Get | Max. ID Number Instance Attribute | UINT | Max. ID number of the instance attribute | 13 |

**Instance attributes**

| Attr. ID | Access | Name | Data type | Description | Default |
|---|---|---|---|---|---|
| 1 | Get | Status | UINT | Status of the TCP/IP interface<br>• 0: interface not configured<br>• 1: interface configured with DHCP, BOOTP or non-volatile memory | |

| Attr. ID | Access | Name | Data type | Description | Default |
|---|---|---|---|---|---|
| 2 | Get | Configuration Capability | DWORD | Configuration options for interface<br>• Bit 0: BOOTP Client<br>• Bit 2: DHCP Client<br>• Bit 4: configuration adjustable<br>• Bit 7: ACD-capable | 0x95 (BOOTP, DHCP, TCP/IP configur-able, ACD capable) |
| 3 | Get / Set | Configuration Control | DWORD | Configuration control<br>Bit 0..3: startup configuration<br>• 0: statically assigned IP address<br>• 1: configuration via BOOTP<br>• 2: configuration via DHCP | 0 |
| 4 | Get | Physical Link Object Path | STRUCT of | Logical path to the physical communica-tion interface (Ethernet link object) | |
| | | • Path Size | UINT | Length (number of words, little endian for-mat) | 0x02 00 |
| | | • Path | Padded EPATH | Class ID: 0xF6 (Ethernet Link Object)<br>• Instance ID: 0x1 | 0x20 F6 2 4 01 |
| 5 | Get / Set | Interface configuration | STRUCT of | TCP/IP interface configuration | |
| | | • IP Address | UDINT | IP address | 192.168.1. 250 |
| | | • Network Mask | UDINT | Network mask | 255.255.2 55.0 |
| | | • Gateway Address | UDINT | Gateway address | 0.0.0.0 |
| | | • Name Server | UDINT | Primary name server | 0.0.0.0 |
| | | • Name Server 2 | UDINT | Secondary name server | 0.0.0.0 |
| | | • Domain Name | STRING | Default domain name | 0 |
| 6 | Get / Set | Host Name | STRING | Host Name<br>• 0: not configured | 0 |
| 8 | Get | TTL Value | UINT | TTL value | 1 |
| 9 | Get / Set | Mcast Config | UINT | Mcast configuration | 0 |
| 10 | Get / Set | SelectAcd | BOOL | ACD activation / deactivation<br>• 0: deactivated<br>• 1: activated | 1 |
| 11 | Get / Set | Last conflict detected | STRUCT of | Last conflict detected | 0 |
| | | • AcdActivity | USINT | Condition of ACD activity at last conflict detected<br>• 0: Noconflictdetected<br>• 1: ProbeIpv4Address<br>• 2: Ongoing Detection<br>• 3: SemiActiveprobe | |
| | | • Remote MAC | ARRAY(6) of USINT | MAC address of the remote node of the ARP PDU in which the conflict was de-tected | |
| | | • ArpPdu | ARRAY(28) of USINT | Copy of the data of the ARP PDU in which the conflict was detected | |
| 13 | Get / Set | Encapsulation Inactivity Time-out | UINT | Inactivity before the TCP connection is deactivated (in seconds) | 120 |

### Services

| Code | Service | Class | Instance | Description |
|---|---|---|---|---|
| 0x01 | Get Attribute All | No | Yes | Read all attributes |
| 0x0E | Get Attribute Single | Yes | Yes | Read single attribute |

| Code | Service | Class | Instance | Description |
|---|---|---|---|---|
| 0x10 | Set Attribute Single | No | Yes | Write single attribute |

### 11.2.4.11 Ethernet Link Object (Class Code: 0xF6)

#### Class attributes

| 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 | 2 |
| 6 | Get | Max. ID Number Class Attributes | UINT | Max. ID number of a class attribute | 7 |
| 7 | Get | Max. ID Number Instance Attribute | UINT | Max. ID number of the instance attribute | 768 |

#### Instance attributes

| Attr. ID | Access | Name | Data type | Description | Default |
|---|---|---|---|---|---|
| 1 | Get | Interface Speed | UDINT | Current transmission rate<br>• 10: 10 Mbit/s<br>• 100: 100 Mbit/s | 100 |
| 2 | Get | Interface Status Flag | DWORD | Interface status flags<br>• Bit 0: link status<br>• Bit 1: half/full duplex<br>• Bit 2…4: negotiation status<br>• Bit 5: manual change requires reset<br>• Bit 6: local hardware error<br>• Bit 7…31: reserved | 0x20 |
| 3 | Get | Physical Address | ARRAY(6) of USINT | MAC address | |
| 4 | Get | Interface Counters | STRUCT(11) of UDINT | Interface-specific counter | |
| 5 | Get | Media Counters | STRUCT(12) of UDINT | Medium-specific counter | |
| 6 | Get / Set | Interface Control | STRUCT of | Interface control | |
| | | • Interface Settings | DWORD | Settings<br>Bit 0:<br>• 0: auto-negotiation on<br>• 1: auto-negotiation off<br>Bit 1:<br>• 0: half duplex<br>• 1: full duplex | 0 |
| | | • Interface Speed | UINT | Transmission rate<br>• 10: 10 Mbit/s<br>• 100: 100 Mbit/s | |
| 7 | Get | Interface type | USINT | • 0: unknown<br>• 1: internal interface<br>• 2: twisted pair<br>• 3: optical fibre | 2 |
| 8 | Get | Interface State | USINT | Current status of the interface<br>• 0: unknown<br>• 1: active; ready for transmission and reception<br>• 2: not active<br>• 3: test mode | 0 |

| Attr. ID | Access | Name | Data type | Description | Default |
|---|---|---|---|---|---|
| 9 | Get / Set | Admin State | USINT | Control of the access to the interface<br>• 1: activate<br>• 2: deactivate | 1 |
| 10 | Get | Interface Label | SHORT_STRING | Interface identifier | • X21 (instance 1)<br>• X22 (instance 2) |
| 11 | Get | Interface Capability | STRUCT of | Interface capability | |
| | | • Interface Speed | DWORD | Transmission rate<br>• 10: 10 Mbit/s<br>• 100: 100 Mbit/s | |
| | | • Interface Duplex Mode | DWORD | Duplex mode<br>• HD: half duplex<br>• FD: full duplex | |
| 768 | Get / Set | MDIX | USINT | MDIX configuration<br>• 1: MDIX_AUTO<br>• 2: MDIX_MDI<br>• 3: MDIX_MDIX | 1 |

## Services

| Code | Service | Class | Instance | Description |
|---|---|---|---|---|
| 0x01 | Get Attribute All | No | Yes | Read all attributes |
| 0x0E | Get Attribute Single | Yes | Yes | Read single attribute |
| 0x10 | Set Attribute Single | No | Yes | Write single attribute |