Device manual

IO-Link master with EtherNet/IP interface
CabinetLine
8 ports
IP 20

AL1920

Firmware: 3.1.x

English

80284064 / 03 10 / 2021

Contents

1 Preliminary note

Content

33203

1.1 Legal and copyright information

33117

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

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

1.2 Purpose of the document

34227

This document is only for device types "IO-Link master - EtherNet/IP gateway (CabinetLine) 8 port IP 20" (art. no.: AL1920).

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

► Read this document before using the device.

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

1.3 Explanation of Symbols

34171

WARNING

Warning of serious personal injury.

Death or serious irreversible injuries may result.

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

NOTICE
Warning of damage to property

Important note
Non-compliance can result in malfunction or interference

Information
Supplementary note

► ... Request for action

> ... Reaction, result

→ ... "see"

abc Cross-reference

123 Decimal number
0x123 Hexadecimal number
0b010 Binary number

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

1.4 Change history

61118

Version	Topic	Date
00	New creation of the document	04 / 2019
01	Correction: Technical data - current rating per output	09 / 2019
02	▪ Added: New IoT core functions ▪ Added: IoT Core Visualizer ▪ Correction: Description of the IoT Core Service getsubscriptioninfo	10 / 2020
03	Deleted: ifm IoT Core – DNS support	10 / 2021

2 Safety instructions

Content

28333

2.1 General

58525

- The device described is a subcomponent for integration into a system. The manufacturer is responsible for the safety of the system. The system manufacturer undertakes to perform a risk assessment and to create documentation in accordance with legal and normative requirements to be provided to the operator and user of the system. This documentation must contain all necessary information and safety instructions for the operator, the user and, if applicable, for any service personnel authorised by the manufacturer of the system.
- Read this document before setting up the product and keep it during the entire service life.
- The product must be suitable for the corresponding applications and environmental conditions without any restrictions.
- Only use the product for its intended purpose (→ **Intended use** (→ p. 9)).
- If the operating instructions or the technical data are not adhered to, personal injury and/or damage to property may occur.
- The manufacturer assumes no liability or warranty for any consequences caused by tampering with the product or incorrect use by the operator.
- Installation, electrical connection, set-up, programming, configuration, operation and maintenance of the product must be carried out by personnel qualified and authorised for the respective activity.
- Protect units and cables against damage.

2.2 Required background knowledge

34185

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

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

2.3 Safety symbols on the device

34199

General warning
Observe instructions in chapter "Electrical connection" (→ **Electrical connection** (→ p. 14))!

2.4 IT security

54678

NOTICE!

If the device is operated in an unprotected network environment.

> Unauthorised read or write access to data is possible.

> Unauthorised manipulation of the device function is possible.

► Check and restrict access options to the device:

- Restrict access to authorised persons.
- Do not connect the device to open networks or the internet.

If access from the internet is inevitable:

► choose a safe method to connect with the device (e. g. VPN).

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

3 Intended use

34594

The IO-Link master serves as a gateway between intelligent IO-Link devices and the EtherNet/IP network. The device is designed for use as cabinet module in plant construction.

► Use the device only within the limits of the technical data (→ **Technical data** (→ p. 94)).

4 Function

Content

33836

4.1 Communication, parameter setting, evaluation

Content

33860

4.1.1 IO-Link

34084

The device offers the following IO-Link functions:

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

4.1.2 EtherNet/IP

52585

The device offers the following EtherNet/IP functions:

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

4.1.3 Internet of Things (IoT)

54679

The device offers the following IoT functions:

- Gateway for the transmission of process, parameter and monitoring data between IO-Linkmaster / IO-Link devices and the IT network level
- REST-API to access process and parameter data
- Supported protocols: TCP/IP JSON, MQTT

4.1.4 Security mode

54697

The IoT interface offers the following optional sercurity functions:

- Secure data transfer via encrypted connection (Secure Layer Transport - TLS)
- Access protection via authentification

4.1.5 Parameter setting

34210

The device provides the following configuration options:

- Parameter setting of the IO-Link master of the AL1920 with LR DEVICE parameter setting software, EtherNet/IP projection software or ifm IoT-Core services.
- Parameter setting of the connected IO-Link devices (sensors, actuators) with LR DEVICE parameter setting software, EtherNet/IP projection software or ifm IoT-Core services
- Storage of parameter sets of the connected IO-Link devices for automatic recovery (data storage)

4.1.6 Visual indication

34192

The device has the following visual indicators:

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

4.2 Digital inputs

33817

The device has 8 additional digital inputs (type 2 according to EN 61131-2).

The digital inputs are on clamp 2 of the ports X01...X04.

All inputs refer to the potential of the device supply (clamp 3).

4.3 IO-Link supply

34077

The device has 8 supplies for IO-Link devices.

The IO-Link ports X01...X08 are ports class A.

Every supply provides short circuit monitoring.

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

5 Mounting

Content

34058

5.1 Install the device

34070

► Disconnect power before installation.

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

► When handling the device, observe the necessary safety precautions against electrostatic discharge (ESD).

► Only operate the device when mounted on a grounded DIN rail.

► Install the device in a control cabinet of protection rating IP 54 or higher. The control cabinet has to be installed in accordance with local and national regulations.

► Fix the device vertically onto a 35 mm raised rail.

► Leave enough space between the unit and the top or bottom of the control cabinet as well as to adjacent devices to enable air circulation and to avoid inadmissible heating.

6 Electrical connection

Content

33805

6.1 Notes

34181

The unit must be connected by a qualified electrician.

► The national and international regulations for the installation of electrical equipment must be adhered to.

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

► Observe the information concerning IO-Link circuits!

The IP rating of the overall system depends on the protection ratings of the individual devices and the applied connection elements.

For UL applications:

► To connect the IO-Link master, only use cables with AWG 26 to 12 and a minimum temperature range of 75 °C.

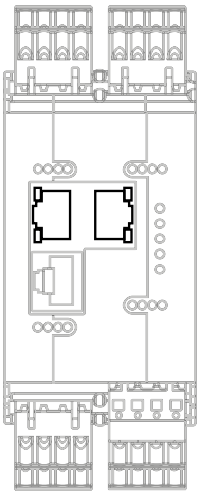
Wiring: → **Technical data** (→ p. 94)

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

The communication interfaces are separated from each other and from device surfaces that could be touched by means of basic insulation according to EN61010-1 (secondary circuit with 30 V DC maximum, supplied from mains circuit up to 300 V of overvoltage category II). They are designed for network environment 0 according to IEC TR62102.

6.2 Connecting the EtherNet/IP ports

33678

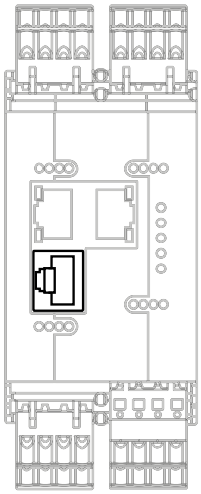


► Connect the unit via the sockets X21 and/or X22 to the EtherNet/IP network.

► To connect the devices, use connectors with protection rating IP 20 or higher (→ **Accessories** (→ p. 92)).

6.3 IoT port

34045



- ► Connect the device via the socket X23 to the IT network (e.g. laptop/PC with LR DEVICE parameter setting software, laptop/PC with LR SMARTOBSERVER monitoring software, laptop/PC with http request enabled software).
- ► To connect the devices, use connectors with protection rating IP 20 or higher (→ **Accessories** (→ p. 92)).

6.4 IO-Link ports

52232

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

► Please note the information concerning IO-Link wiring!

WARNING

Supply of energy to the IO-Link ports of the IO-Link master

> Risk of fire!

► Prevent supply and feedback of energy to the IO-Link ports.

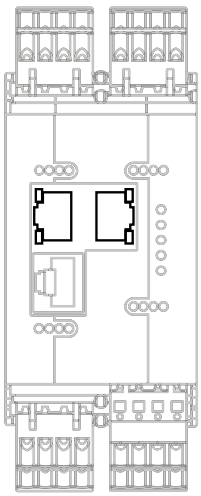
► Before set-up check the correct connection of the supply cables.

6.4.1 Connect IO-Link devices for Class A operation

52233

Wiring information:

- The connected IO-Link devices must be supplied exclusively via the IO-Link master.
- The additional digital inputs of the IO-Link ports X01...X08 (clamp 2) have a type 2 behaviour according to the standard EN61131-2. The connected electronics must be electrically suited for this.



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

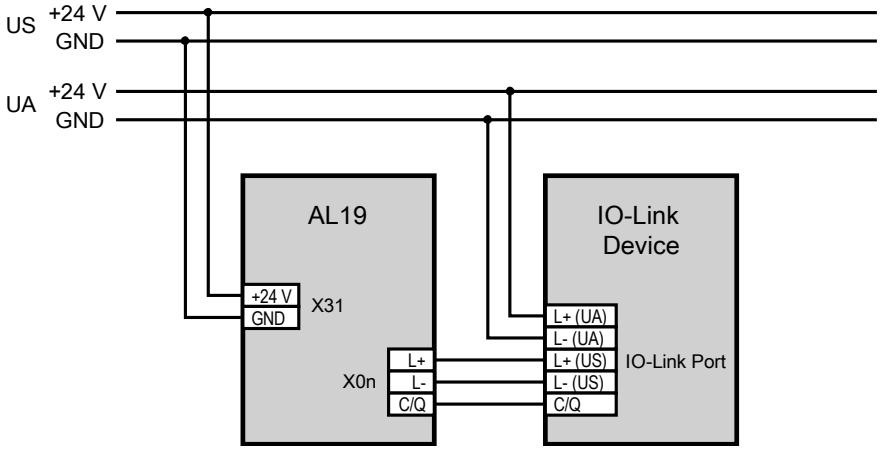
► To connect the devices, only use cables with protection rating IP 20 or higher.

6.4.2 Connect IO-Link devices for Class B operation

52234

Wiring information:

- For the Class B operation, the IO-Link device must be supplied with an additional auxiliary voltage UA.
- Wiring diagram:



- Permitted maximum current intensity for UA: 4A

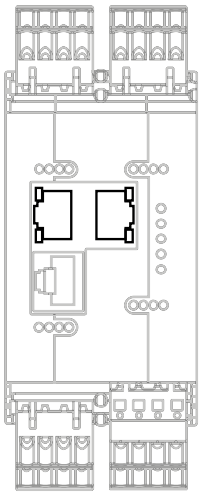
WARNING

Non-compliance with the electrical separation of the circuits

> Risk of fire!

► Ensure that the external supply UA is galvanically separated from the circuit of the IO-Link Master by assuring basic insulation (according to IEC 61010-1, secondary circuit with 30 V DC maximum, supplied from mains circuit up to 300 V of overvoltage category II).

► Ensure that the IO-Link devices and the connection technology support the galvanic separation.



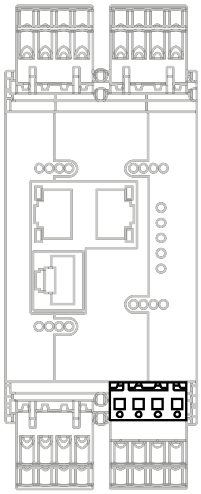
► Connect the IO-Link devices to the ports X01 ... X08.
  - Maximum cable length per IO-Link port: 20 m

► Connect the IO-Link devices to UA with 24 V DC (20...30 V SELV/PELV).

► To connect the IO-Link devices, only use cables with protection rating IP 20 or higher.

6.5 Connect the device

33890



- ► Disconnect power.
- ► Connect the IO-Link master via port X31 to 24 V DC (20...30 V SELV/PELV).
  - Recommended maximum cable length: 25 m
- ► To connect the device, use cables with protection rating IP 20 or higher.

With cable lengths greater than 25 m observe the voltage drop and the necessary minimum supply voltage of 20 V!

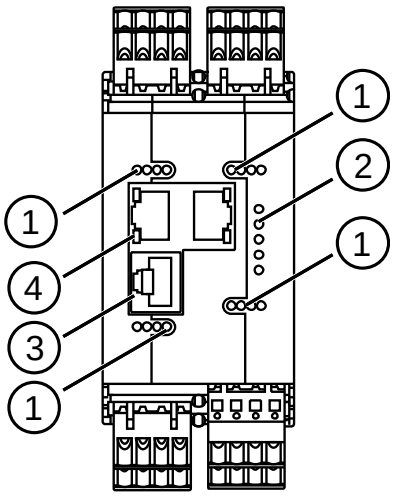
7 Operating and display elements

Content

34063

7.1 Overview

34353



1. IOL and DI status-LEDs of the IO-Link port (X01...X08 (→ **IO-Link Ports (Class A)** (→ p. 21))
2. PWR status LED of the voltage supply (X31) (→ **Power supply** (→ p. 21)
   RDY, NET and MOD Status LEDs (→**Status LEDs** (→ p. 20)
   IoT status LED of the IoT port (X23) (→ **IoT port** (→ p. 21))
3. LNK status LED of the IoT port (X23) (→ **IoT port** (→ p. 21))
4. LNK and ACT status LEDs of the EtherNet/IP ports 1 (X21) and 2 (X22) (→ **Ethernet ports** (→ p. 20))

7.2 LED indicators

34047

The device only has the following LED indicators:

7.2.1 Status LEDs

34408

The RDY LED indicates the status of the gateway.
The NET LED (Network Status) indicates the status of the network.
The MOD LED (Module Status) indicates the status of the EtherNet/IP module.

Status LED			Description
RDY	green	on	Status: OK
		flashes 5 Hz	Status: Error
		flashes (200 ms on, 800 ms off)	Status: Firmware update is running
		off	Status: Gateway not running or gateway booting
NET	green / red	off	Not powered or powered, but IP address not yet configured
		flashes	Device self-testing
	green	flashes	No connection: no CIP connection established and a Exclusive Owner connection has not timed out
		on	Connection: at least one CIP connection established and an Exclusive Owner connection has not timed out
	red	flashes	Connection timeout: an Exclusive Owner connection has timed out
		on	Duplicate IP address: IP address aready in use
MOD	green / red	off	No voltage or voltage too low
		flashes	Device self-testing
	green	flashes	Standby: device not yet configured (no IP address)
		on	Operational
	red	flashes	Major recoverable fault (e.g. incorrect configuration)
		on	Major unrecoverable fault (e.g. module failed)

7.2.2 Ethernet ports

34348

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

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

7.2.3 IoT port

34043

The IoT port has the 3 LNK, ACT and IoT LEDs. The LEDs indicate the status of the Ethernet connection and the device identification.

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

7.2.4 Power supply

34203

The interface for voltage supply (X31) has the PWR LED. The LED indicates the status of the voltage supply.

Status LED			Description
PWR	green	on	Supply voltage Us is applied
		off	No supply voltage is applied or the applied supply voltage is too low

7.2.5 IO-Link Ports (Class A)

34074

Each IO-Link Port Class A has 2 LEDs labelled IOL and DI. The LEDs indicate the status of the IO-Link port.

Status LED			Description
IOL	yellow	Off	Port configured as DI/DO: clamp 4 (C/Q) = OFF
		on	Port configured as DI/DO: clamp 4 (C/Q) =ON
	green	flashing 1 Hz	Port configured asIO-Link: no IO-Link device found
		Flashing with 2 Hz	Port configured asIO-Link: Status PREOPERATE
		on	Port configured asIO-Link: Status OPERATE
	red	Flashing with 2 Hz	Port configuration error or short circuit / overload on US
		on	Transmission Error
DI	yellow	Off	Digital input: clamp 2 = OFF
		on	Digital input: clamp 2 = ON

8 Set-up

52357

When the supply voltage is switched on, the AL1920 starts with the factory settings. The display elements signal the current operating mode (→ **Operating and display elements** (→ p. 19)).

To enable parameter setting of the AL1920, the IoT interface and / or the fieldbus interface must be configured according to the network environment.

► Configure IoT interface (LR DEVICE: → **IoT: Configure IP settings** (→ p. 26) or → **IoT: Configuring IP settings** (→ p. 46)).

► Configure fieldbus interface (LR DEVICE: → **Fieldbus: Configure IP settings** (→ p. 29) or IoT: → **Feldbus-Schnittstelle konfigurieren**).

> IoT / fieldbus interface has valid IP settings.

> User can set the parameters of the AL1920.

Further steps:

- Optional: Update firmware of AL1920 (→ **Updating the firmware** (→ p. 90)).
- Set the parameters of the AL1920 (→ **Configuration** (→ p. 23)).

9 Configuration

Content

33858

9.1 LR DEVICE

Content

33692

On delivery, the AL1920 is configured with the factory settings (→ **Factory settings** (→ p. 91)).

Required software: LR DEVICE (1.5.0.x or higher) (art.-no.: QA0011/QA0012)

9.1.1 Remarks

Content

34180

Offline parameter setting

34060

The AL1920 supports the offline parameter setting. In this context, the user creates and stores a configuration for the IO-Link master and the connected IO-Link devices without being connected to the AL1920 (OFFLINE mode). The configuration created in this way can be stored as a file (*.lrp) and loaded to the AL1920 and activated at a later date.

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

VPN connection

34382

An active VPN connection blocks the access of the parameter setting software LR DEVICE to the EtherNet/IP interface of the AL1920.

► Deactivate the VPN connection in order to be able to access the AL1920 with the LR DEVICE.

9.1.2 IoT: Configure IP settings

34049

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

To configure the IP settings with DHCP, a DHCP server has to be active in the IT network. If no DHCP server can be reached in the IT network, an IP address is automatically assigned to the IoT port with the Zeroconfig protocol (address range: → **Factory settings** (→ p. 91)).

To configure the IP settings of the IoT interface:

► Select [IoT] menu.

> The menu page shows the current settings.

► Set the following parameters as required:

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

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

► Save changed values on the device.

9.1.3 IoT: Configure security mode

54680

The IoT interface of the IO-Link offers a security mode. It enables secure data transmission via transport encryption and restriction of the access to IO-Link masters and IO-Link devices via user authentication.

To configure the security mode:

► Select [IoT] menu.

> The menu page shows the current settings.

► Set the following parameters as required:

Name	Description	Possible values	
[Security mode HTTPS]	Set the security mode	[Disabled]	Security mode disabled
		[Enabled]	Security mode enabled
[Security password]	Password Note: The set password is not displayed.		

► Save changed values on the device.

The security mode only protects the access to the device via the IoT interface.
The user name "administrator" cannot be changed.

The security mode can be enabled without setting the password. During the attempt to write to the device, LR DEVICE requires to enter and confirm the password.

After entering the password, the user has unrestricted access to IO-Link masters and connected IO-Link devices. The password will only be requested again if the current LR DEVICE session is over (e. g. after restarting the LR DEVICE).

To change the set password:

► Sign in with a valid password.

► Enter the new password in the field [Security password].

► Write changes to the device.

> The new password is set.

9.1.4 IoT: Configuring access rights

34046

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

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

► Select [IoT] menu.

> The menu page shows the current settings.

► Set the following parameters as required:

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

* ... Factory setting

► Save changed values on the device.

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

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

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

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

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

34048

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

► Select [IoT] menu.

> The menu page shows the current settings.

► Set the following parameters as required:

Name	Description	Possible values	
[IP address LR Agent or SMARTOBSERVER]	IP address of LR AGENT or LR SMARTOBSERVER	Factory setting: 255.255.255.255	
[Port LR Agent or SMARTOBSERVER]	Port number that is used to send process data to LR AGENT or LR SMARTOBSERVER	0 ... 65535	Factory setting:: 35100
[Interval LR Agent or SMARTOBSERVER]	Cycle time for the transfer of the process data to LR AGENT or LR SMARTOBSERVER (value in milliseconds)	[Off]	no transfer
		500 ... 2147483647	500 ms ... 2147483647 ms
[Application Tag]	Source identifier of the IO-Link master in the structure of LR AGENT or LR SMARTOBSERVER (String32)	Factory setting: AL1920	

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

To prevent the delay:

► Reboot the device after changing the the parameter.

► Save changed values on the device.

9.1.6 Fieldbus: Configure IP settings

54692

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

► Select [Fieldbus] menu.

> The menu page shows the current settings.

► Set the following parameters as required:

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

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

► Save changed values on the device.

9.1.7 Fieldbus: set the configuration mode

54693

The AL1920 supports the EtherNet/IP configuration modes "top-down" and "independent". Additionally, the user can configure the length of the transmitted process data and select the required connection types.

► Select [Fieldbus] menu.

> The menu page shows the current settings.

► Set the following parameters as required:

Name	Description	Possible values	
[Configuration]*	EtherNet/IP configuration mode	Independent mode off	Configuration via fieldbus PLC
		Independent mode on	Configuration via AL1920
[Process data length]*	Length of process data per IO-Link port	2 bytes input 2 bytes output	2 bytes input data, 2 bytes output data
		4 bytes input 4 bytes output	4 bytes input data, 4 bytes output data
		8 bytes input 8 bytes output	8 bytes input data, 8 bytes output data
		16 bytes input 16 bytes output	16 bytes input data, 16 bytes output data
		32 bytes input 32 bytes output	32 bytes input data, 32 bytes output data
[Swap]*	Sequence of bytes in the data word	off	as Array of Bytes
		on	as Integer16 value; during an update of the process data, the bytes are exchanged
[Explicitpdmode]**	Enable / disable explicit PD mode and select the process data to be transmitted (connection types)	Explicit process data mode off	Explicit PD mode disabled
		Explicit process data mode with IO-Link I/O + Acyclic + Diag	Explicit PD mode enabled: IO-Link inputs /outputs, acyclic data and diagnostic data are transmitted
		Explicit process data mode with IO-Link I/O + Acyclic	Explicit PD mode enabled: IO-Link inputs/outputs and acyclic data are transmitted
		Explicit process data mode with IO-Link I/O	Explicit PD mode enabled: IO-Link inputs/outputs are transmitted

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

► Save changed values on the device.

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

33690

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

To transfer process data the interface to the LR AGENT or LR SMARTOBSERVER has to be correctly configured (→ **IoT: Configure the interface to LR AGENT or LR SMARTOBSERVER** (→ p. 28)).

To activate / deactivate data transfer:

► Select [Port x] menu (x = 1...8).

> The menu page shows the current settings.

► Set the following parameters as required:

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

► Save changed values on the device.

9.1.9 IO-Link ports: Configure operating mode

33694

The IO-Link ports X01...X08 of the device support the following operating modes:

- Disabled: no data transfer at clamp 4 (C/Q) of the IO-Link port
- Digital input (DI): binary input signal at clamp 4 (C/Q) of the IO-Link port
- Digital output (DO): binary output signal at clamp 4 (C/Q) of the IO-Link port
- IO-Link: IO-Link data transfer via clamp 4 (C/Q) of the IO-Link port

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

To set the operating mode of an IO-Link port:

► Select [Port x] menu (x = 1...8).

> The menu page shows the current settings.

► Set the following parameters as required:

Name	Description	Possible values	
[Mode Pin4 US]	Operating mode of clamp 4 of the IO-Link port	[Disabled]	Port deactivated
		[DI]	Operation as digital input
		[DO]	Operation as digital output
		[IO-Link]	Operation as IO-Link interface
[Cycle time actual]**	Current cycle time of the data transfer between IO-Link master and IO-Link device on the port (value in microseconds)	Parameter can only be read	
[Cycle time preset]*	Cycle time of the data transfer between the IO-Link master and the IO-Link device at the port (value in microseconds)	0	The device automatically sets the fastest possible cycle time.
		1 ... 132800	1 microsecond ... 132800 microseconds
[Bitrate]**	Current transmission rate of the data transfer between the IO-Link master and the IO-Link device on the port	Parameter can only be read	

* ... Parameter only available if [Mode] = [IO-Link]
** ... Parameter only visible if the IO-Link device is connected to the IO-Link port.

► Save changed values on the device.

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

33697

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

The following options are available:

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

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

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

To configure the device validation and the data storage:

► select [Port x] menu (x = 1...8).

> The menu page shows the current settings.

► Set the following parameters as required:

Name	Description	Possible values	
[Validation / Data Storage]	Supported IO-Link standard and behaviour of the IO-Link master when connecting a new IO-Link device at port x (x = 1...8)	[No check and clear]	
		[Type compatible V1.0 device]	
		[Type compatible V1.1 device]	
		[Type compatible V1.1 device with Backup + Restore]	
		[Type compatible V1.1 device with Restore]	
[Vendor ID]	ID of the manufacturer that is to be validated	0...65535	Factory setting: 0# ifm electronic: 310
[Device ID]	ID of the IO-Link device that is to be validated	0...16777215	Factory setting: 0

► Save changed values on the device.

9.1.11 IO-Link ports: Setting fail-safe values

34329

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

To set the fail-safe values:

► Select [Port x] menu (x = 1...8).

> The menu page shows the current settings.

► Set the following parameters as required:

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

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

► Save changed values on the device.

9.1.12 Info: Show device information

34065

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

► Select [Info] menu.

> The menu page shows the current settings.

Name	Description	Possible values
[Product code]	Article number of the IO-Link master	AL1920
[Device family]	Device family of the IO-Link master	IO-Link master
[Vendor]	Vendor	ifm electronic gmbh
[SW-Revision]	Firmware of the IO-Link master	
[HW revision]	Hardware version of the IO-Link master	
[Bootloader revision]	Bootloader version of the IO-Link master	
[Serial number]	Serial number	

9.1.13 Firmware: Reset device to factory settings

33838

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

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

9.1.14 Firmware: Reboot the device

33832

When rebooting the device, all settings are kept.
To restart the AL1920:

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

9.1.15 Configure IO-Link devices

33856

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

Requirements:

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

> The IO-Link device is connected correctly with the AL1920.

> Operating mode of the IO-Link port is "IO-Link" (→ **IO-Link ports: Configure operating mode** (→ p. 32)).

> IoT has write access rights to the IO-Link master (→ **IoT: Configuring access rights** (→ p. 27)).

1 Select IO-Link master

► Start LR DEVICE.

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

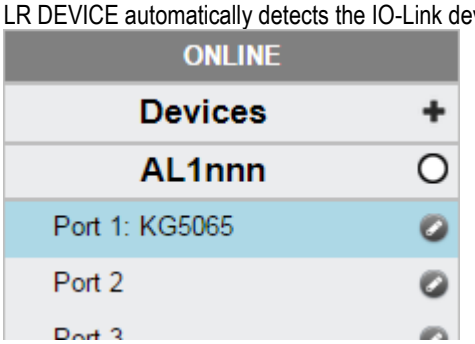
► Scan network for devices.

> LR DEVICE detects IO-Link master.

2 Add IO-Link device

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

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



3 Configure IO-Link device

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

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

► Configure IO-Link device.

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

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

9.2 ifm IoT Core

Content

52244

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

9.2.1 Programmers' notes

Content

34229

IoT Core: General information

52256

The CabinetLine device family has an IoT Core. The IoT Core allows the user to address the AL1920 from IT networks via a REST API and to integrate it into Internet-of-Things applications.

A device description is stored on the AL1920. This device description is a structured, machine-readable data object in JSON format. All current values of parameters, process data, diagnostic data and device information are mapped in this data object. These data values can be read and changed by means of services.

Access the ifm IoT Core

52257

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

GET request

33804

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

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

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

The syntax of the return of the IoT Core is:

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

Parameter	Description
id	Correlation ID for the assignment of request and return
resp_data	Value of the data point; depending on the data type of the data point
diag_code	Diagnostic code (→ **IoT Core: Diagnostic codes** (→ p. 41))

Example: GET request

54033

Request (via browser):

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

Response:

```
{
"cid":-1,
"data":{"value":"AL1920"},
"code":200
}
```

POST request

54700

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

```
{
"code":"code_id",
"cid":id,
"adr":"data_point/service",
"data":{req_data},
"auth":{"user":"usr_id","passwd":"password"}
}
```

Field	Parameter	Description	
code	code_id	Service class	
		▪ request	Request
		▪ transaction	Transaction
		▪ event	Event
cid	id	Correlation ID for the assignment of request and response; ID freely assignable by the user	
adr	data_point	Data point of the element tree which is to be accessed	
	service	Service to be performed (→ **Overview: IoT services** (→ p. 158))	
data*	req_data	Data to be transferred to the IoT Core (e.g. new values); syntax depending on the service	
auth**	usr_id	user name (base64 coded); default value: administrator	
	password	password (base64 coded)	

* = optional; only required for services, that submit data to the IoT core (e. g. setdata)
** = optional; only required, if security mode is activated

The syntax of the return of the IoT Core is:

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

Field	Parameter	Description
cid	id	Correlation ID for the assignment of request and response (see request)
data*	resp_data	Value of the data point; syntax depending on the service
code	diag_code	Diagnostic code (→ **IoT Core: Diagnostic codes** (→ p. 41))

* = optional; only required for services, that receive data from the IoT core (e.g. getdata)

Example: POST request

54035

Request:

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

Response:


```
{
"cid":4711,
"data":{"value":"AL1920"},
"code":200
}
```


IoT Core: Diagnostic codes

54688

Code	Text	Description
200	OK	Request successfully processed
230	OK but needs reboot	Request successfully processed; IO-Link master must be restarted
231	OK but block request not finished	Request successfully processed; blockwise request, but not yet finished
232	Data has been accepted, but internally modified	New values have been accepted, but were adjusted by the IO-Link master (Master cycle time)
233	IP settings (of IoT-Port) have been updated. Application needs to reload device. Wait at least 1 second before reloading device.	IP settings have been successfully changed, IO-Link master will be reloaded; wait for at least 1 second
400	Bad request	Invalid request
401	Unauthorized	Non authorised request
403	Forbidden	Forbidden request
500	Internal Server Error	Internal fault
503	Service Unavailable	The service is not available (e. g. IO-Link port in wrong operating mode; no IO-Link device at IO-Link port)
530	The requested data is invalid	Invalid process data
531	IO-Link error	Error in IO-Link Master / device
532	PLC connected Error	Error while setting data, because IO-Link master is still connected to fieldbus PLC

9.2.2 First steps

52245

To read the device description of the AL1920:

► Send the following POST request to the AL1920:
`{"code":"request","cid":-1,"adr":"gettree"}`

> AL1920 returns the device description as structured JSON object.

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

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

9.2.3 General functions

61148

The AL1920 has the type `device` (→ **Overview: IoT types** (→ p. 157)).

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

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

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

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

Example: Reading properties of an element

59782

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

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

- Request:

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

- Response:

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

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


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

Example: output subtree

61149

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

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

- Request:


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


- Response:


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

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

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

33840

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

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

- Request:

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

- Response:

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

Example: Browsing device description

61150

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

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

- Request:

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

- Response:

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

Setting the storage duration

61153

The IoT Core offers the possibility to set the storage duration of data and notifications. The Services **Service: setdata** (→ p. 168) and **Service: subscribe** (→ p. 170) therefore have the parameter "duration".

Example: Subscribing to notifications

61154

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

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

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

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

- Request:

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

- Response:

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

9.2.4 IoT: Configuring access rights

59785

Substructure: `iotsetup`

Available data points:

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

rw ... read and write

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

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

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

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

9.2.5 IoT: Configuring IP settings

61155

Substructure: `iotsetup`

Available data points:

Name	Description	Access
../network/dhcp	Configuration of the IP settings of the IoT port	rw
../network/ipaddress	IP address of the IoT port	rw
../network/subnetmask	Subnet mask of the network segment	rw
../network/ipdefaultgateway	IP address of the network gateway	rw

rw ... read and write

Applicable services:

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

Change the IP parameters in the substructure network only blockwise with the setblock service!

9.2.6 IoT: Configuring the LR AGENT or LR SMARTOBSERVER interface

59786

Substructure: `iotsetup`

Avalable data points:

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

rw ... read and write

9.2.7 IoT: Configuring security mode

54683

The access to the IoT interface of the IO-Link master can be protected with a security mode:

Substructure: `iotsetup`

Available data points:

Name	Description	Access
../security/securitymode	active security mode	rw
../security/password	Password for authentication (Base64 coded)	w

rw ... read and write
w ... write only

Valid character set for the Base64 coding / decoding of the password: UTF-8
Online tool for coding / decoding: → www.base64encode.org)

Note: Security mode

54684

The security mode enables restricting access to the IO-Link master and the connected IO-Link devices from the IT network. In the activated security mode, the following restrictions apply:

- Access only with authentication (password-protected user account)
- Access only via secure https connection (Transport Layer Security - TLS)

The security mode only protects the access to the device via the IoT interface.
The standard value for users is: administrator
The set password cannot be read with getdata.

The current status of the security function can be read with the getidentity service (→ **Servicet: getidentity** (→ p. 161)).

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 will be shown as Base64-coded character strings (→ **Example: Request with authentication** (→ p. 48)).

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

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

Example: Activate security mode

54701

Task: Activate the security mode of the IO-Link interface of the IO-Link master. Set the password "password" (Base64 coded: cGFzc3dvcmQ=)

Solution: The activation sonsists of 2 steps:

1 Activate security mode

Use service setdata with datapoint iotsetup/security/securitymode to activate the security mode.

- Request:

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

- Response:

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

2 Set required password

Use service setdata with data point iotsetup/security/password to set the required password.

- Request:

```
{
"code":"request",
"cid":-1,
"adr":"/iotsetup/security/password/setdata",
"data":{"newvalue":"cGFzc3dvcmQ="}
}
```

- Response:

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

Example: Request with authentication

54685

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 with the getdata service. 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
}
```


Example: reset password

54686

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
}
```

9.2.8 Fieldbus: Configuring IP settings

59783

Substructure: `fieldbussetup`

Available data points:

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

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

Applicable services:

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

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

9.2.9 Fieldbus: Selecting the configuration mode

52486

Substructure: `fieldbussetup/configuration`

Available data points:

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

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

9.2.10 Fieldbus: Setting fail-safe values

61120

Substructure: `fieldbussetup/configuration/port[n]` (n = 1...8)

Available data points:

Name	Description	Access
../failsafedigitalout	Fail-safe value for the digital output - pin 4 (DO)	rw*
../failsafeiolink	Fail-safe value for IO-Link output data - pin 4 (IO-Link)	rw*

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

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

59793

Substructure: `iolinkmaster/port[n]` (n = 1...8).

Available data points:

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

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

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

59792

Substructure: `iolinkmaster/port[n]` (n = 1...8).

Available data points:

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

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

Applicable services:

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

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

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

52344

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

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

Save Data Storage:

1 Preparations

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

2 Read Data Storage of IO-Link port

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

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

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

► Join segments.

Restore Data Storage:

1 Preparations

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

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

2 Transfer Data Storage strings

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

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

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

59795

Substructure: `iolinkmaster/port[n]` (n = 1...8).

Available data points:

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

rw ... read and write

9.2.14 IO-Link ports: Reading / writing process data

61156

Substructure: `iolinkmaster/port[n]` (n = 1...8)

Available data points:

Name	Description	Access
../pin2in	Value of the digital input on clamp 2 of the IO-Link port	r
../iolinkdevice/pdin	Value of the IO-Link input on clamp 4 of the IO-Link port	r
../iolinkdevice/pdout	Value of the IO-Link output on clamp 4 of the IO-Link port	rw*

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

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

33842

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

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

- Request:

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

- Response:

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

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

0x03C9 = 0b**1111001001**

Temperature value: 0b11110010 = 242

Therefore: The current temperature value is 24.2 °C.

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

59804

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

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

To switch the buzzer:

1. Read the current process value (→ **Example: Read IO-Link process data (operating mode "IO-Link")** (→ p. 54)).
2. Set bit 40 of the read value to 1.
3. Write the process value to the IO-Link device.

Example:

Read process value:

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

New process value:

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

- Request:

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

- Response:

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

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

59803

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

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

- Request:

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

- Response:

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

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

59802

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

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

- Request:


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


- Response:


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

9.2.15 IO-Link ports: Indicating port events

59796

Substructure: iolinkmaster/port[n] (n = 1...8).

Available data points:

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

r ... read only

Subscribing events: → **Subscribing to notifications** (→ p. 63)

9.2.16 IO-Link devices: Accessing parameters

59800

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

Substructure: `iolinkmaster/port[n]/iolinkdevice` (n = 1...8)

Applicable services:

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

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

33847

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

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

- **Request:**

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

- **Return:**

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

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

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

33844

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

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

- Request:

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

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

- Response:

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

9.2.17 IO-Link devices: Reading an writing device information

59797

Substructure: `iolinkmaster/port[n]/iolinkdevice` (n = 1...8)
Available data points:

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

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

9.2.18 IO-Link devices: Indicating IO-Link events

59798

Substructure: `iolinkmaster/port[n]/iolinkdevice` (n = 1...8).
Available data points:

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

r ... read only

Subscribing events: → **Subscribing to notifications** (→ p. 63)

9.2.19 Gateway: Resetting, rebooting and localising the device

59790

Substructure: `firmware`
Applicable services:

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

9.2.20 Gateway: Reading device information

52254

Substructure: `deviceinfo`

Available data points:

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

r ... read only

Additional information about the AL1920 can be read with the service `getidentity` (→ **Servicet: getidentity** (→ p. 161)).

9.2.21 Gateway: Reading status and diagnostic information

61157

Substructure: `processdatamaster`

Available data points:

Name	Description	Access
../temperature	Temperature of the IO-Link master (value in °C)	r
../voltage	Present voltage value of the supply voltage US (value in mV)	r
../current	Present current value of the sensor supply US (value in mA)	r
../supervisionstatus	Status of the device supply US	r

r ... read only

9.2.22 Gateway: Updating the firmware

59789

Substructure: `firmware`

Available data points:

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

r = only read
w = write only

Applicable services:

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

Example: Update firmware

52252

Task:

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

Solution:

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

1 Preparations

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

► Convert the firmware file into a BASE64 string.

2 Start the transfer of the firmware

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

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

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

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

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

4 Install firmware

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

9.2.23 Gateway: Setting the application tag

59791

Substructure: `devicetag`

Available data points:

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

rw ... read and write

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

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

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

Example: Change name of the IO-Link master

a33823

Task: Set the name of the IO-Link master to AL1920 for the representation in the LR SMARTOBSERVER.

Solution: Change the parameter [Application Tag] with the setdata service to the value [AL1920].

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

- Request:

```
{
"code":"request",
"cid":4711,
"adr":"/devicetag/applicationtag/setdata",
"data":{"newvalue":"AL1920"}
}
```

- Response:

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

9.2.24 Subscribing to notifications

61159

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

Available data points:

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

r ... read only
rw ... read and write
x = [1,2]
n = 1...8

Applicable services:

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

Additionally, the user can use **Service: getsubscriberlist** (→ p. 162) show all active subscriptions.

Example: Subscribing to notifications

61160

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

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

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

The following options are additionally available:

- via WebSockets (ws://): **Example: Subscribing notifications via WebSocket** (→ p. 67)
- via MQTT (mqtt://): **Example: Configuring the MQTT command channel** (→ p. 70)

- Request:

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

```
"callback":"http://192.168.0.4:80/temp",
"datatosend":[
"/iolinkmaster/port[2]/iolinkdevice/pdin",
"/processdatamaster/temperature"]
}
}
```

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

- Request:

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

- Response:

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

- Notification (JSON)

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

Example: Changing a subscription

61161

Task: The existing subscription (**Example: Subscribing to notifications** (→ p. 63)) is to be changed. Instead of the temperature of the IO-Link master, the operating voltage applied is to be transmitted.

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

- Request:

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

Example: Subscribing to notifications in CSV format

61162

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 X02
- 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 protocol.

- Request:

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

Example: Unsubscribing from notifications

61163

Task: The existing subscription (**Example: Subscribing to notifications** (→ p. 63)) is to be deleted.

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

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

Example: Checking subscriptions

61164

Task: Information about the existing subscription (**Example: Subscribing to notifications** (→ p. 63)Show **Example: Subscribing to notifications** (→ p. 63)).

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

- Request:

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

- Response:

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

9.2.25 Using Web Socket

61165

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

WebSockets can be used for the following services:

- subscribe / unsubscribe

Maximum number of WebSocket connections: 8
Fail-safe WebSocket connections (wss://) are not supported.

To transmit notifications via a WebSockets connection:

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

- Option 1: without parameter "callback"

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

> IoT-Core sends notifications about existing WebSocket connections.

- Option 2. with parameter "callback"

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

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

Example: Subscribing notifications via WebSocket

61166

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

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

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

- Request:

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

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

- Request:

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

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

9.2.26 MQTT support

61168

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

Configuring the MQTT command channel

61169

To enable MQTT communication, the user needs to activate and configure an MQTT command channel.

Substructure: `connections/mqttConnection`

Name	Description	Access
../type	Type of the connection (MQTT)	r
../status	Global MQTT status	r
../status/preset	Presetting of the MQTT status; Basic settings: running	r
../MQTTSetup	Substructure for general MQTT settings	w
../MQTTSetup/QoS	Quality of Service of the MQTT communication ▪ 0: QoS Level 0 - PUBLISH (without confirmation) ▪ 1: QoS Level 1 - PUBLISH > PUBREC (one-time confirmation) ▪ 2: QoS Level 2 - PUBLISH > PUBREC > PUBREL > PUBCOMP (double confirmation)	rw
../MQTTSetup/version	MQTT version	r
../mqttCmdChannel	Substructure of the MQTT command channel	w
../mqttCmdChannel/type	Type of the MQTT command channel	r
../mqttCmdChannel/status	Status of the MQTT command channel	r
../mqttCmdChannel/status/preset	Presetting of the MQTT status; Basic setting: stopped	r
../mqttCmdChannel/mqttCmdChannelSetup	Structure for settings of the command channel	w
../mqttCmdChannel/mqttCmdChannelSetup/brokerIP	IP address of the MQTT broker	rw
../mqttCmdChannel/mqttCmdChannelSetup/brokerPort	Port number of the MQTT broker	rw
../mqttCmdChannel/mqttCmdChannelSetup/cmdTopic	Designation of the MQTT topic	rw
../mqttCmdChannel/mqttCmdChannelSetup/defaultReplyTopic	Standard response topic	rw

Applicable services:

Name	Description
../status/start	Enable MQTT
../status/stop	Deactivate MQTT
../status/reset	Reset MQTT
../mqttCmdChannel/status/start	Activate MQTT command channel
../mqttCmdChannel/status/stop	Deactivate MQTT command channel
../mqttCmdChannel/status/reset	Reset MQTT command channel

Notes on the states of an MQTT connection: **Note: Connection states** (→ p. 70)

To create an MQTT connection, perform the following steps in sequence:

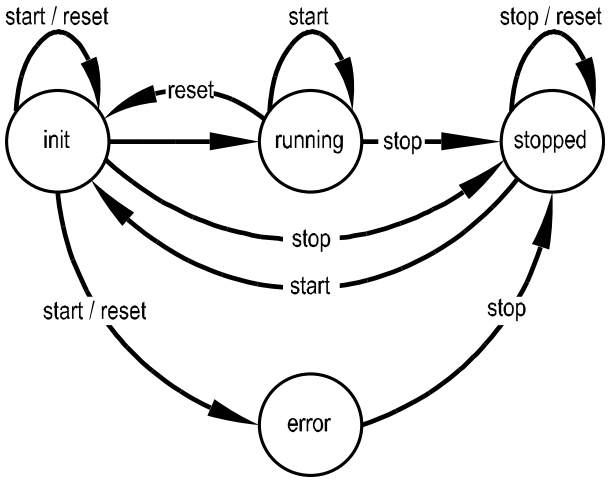
Ensure that the MQTT broker can be reached and that the selected port of the MQTT broker is enabled for data transmission.
Max. number of simultaneous MQTT connections: 10
Wildcards "+" and "#" in topics are not supported.

- ► Activate MQTT command channel.
- ► Set the IP address of the MQTT.
- ► Set the port number of the MQTT broker.
- ► Set topic.
- ► Set standard response topic.
- > The command channel is created with the selected properties.
- > The user can publish on the topic with the IoT Core.
- > MQTT clients can subscribe to the topic.

Note: Connection states

61170

The following status diagram shows the influence of the services "start", "stop" and "reset" on the status of an MQTT connection:



After the initialisation in the "init" state has been completed, the connection automatically changes to the "running" state.

The connection automatically switches to the "error" state if at least one of the following events occurs:

- no MQTT broker available

Example: Configuring the MQTT command channel

61171

Task: Configuring and activating the MQTT command channel (IP address MQTT broker: 192.168.82.100, port: 1883, topic: abc).

Solution:

- ► Check whether MQTT broker can be reached and the port has been released.
- ► Activate command channel

- Request:

```
{
"code":"request",
"cid":4711,
"adr":"/connections/mqttConnection/MQTTSetup/mqttCmdChannel/status/start"
}
```

► Set the IP address of the MQTT broker/server.

- Request:

```
{
"code":"request",
"cid":4712,
"adr":"/connections/mqttConnection/mqttCmdChannel/mqttCmdChannelSetup/brokerIP/set
data"
"data":{"192.168.82.100"}
}
```

► Set the port number of the MQTT broker/server.

- Request:

```
{
"code":"request",
"cid":4713,
"adr":"/connections/mqttConnection/mqttCmdChannel/mqttCmdChannelSetup/brokerPort/s
etdata"
"data":{"1883"}
}
```

► Set topic.

- Request:

```
{
"code":"request",
"cid":4714,
"adr":"/connections/mqttConnection/mqttCmdChannel/mqttCmdChannelSetup/cmdTopic/set
data"
"data":{"abc"}
}
```

► Set standard response topic.

- Request:

```
{
"code":"request",
"cid":4715,
"adr":"/connections/mqttConnection/mqttCmdChannel/mqttCmdChannelSetup/defaultReply
Topic/setdata"
"data":{"xyz"}
}
```

► Set the QoS.

- Request:

```
{
"code":"request",
"cid":4716,
"adr":"/connections/mqttConnection/MQTTSetup/QoS/setdata",
"data":{"QoS2"}
}
```

Example: Publish the temperature to an MQTT broker

54687

Task: Publish the temperature of the IO-Link master to an MQTT broker (IP address MQTT broker: 192.168.82.100, port: 1883, topic: abc

Solution:

- Request:

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

- Response:

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

9.2.27 Using the IoT-Core Visualizer

Content

61173

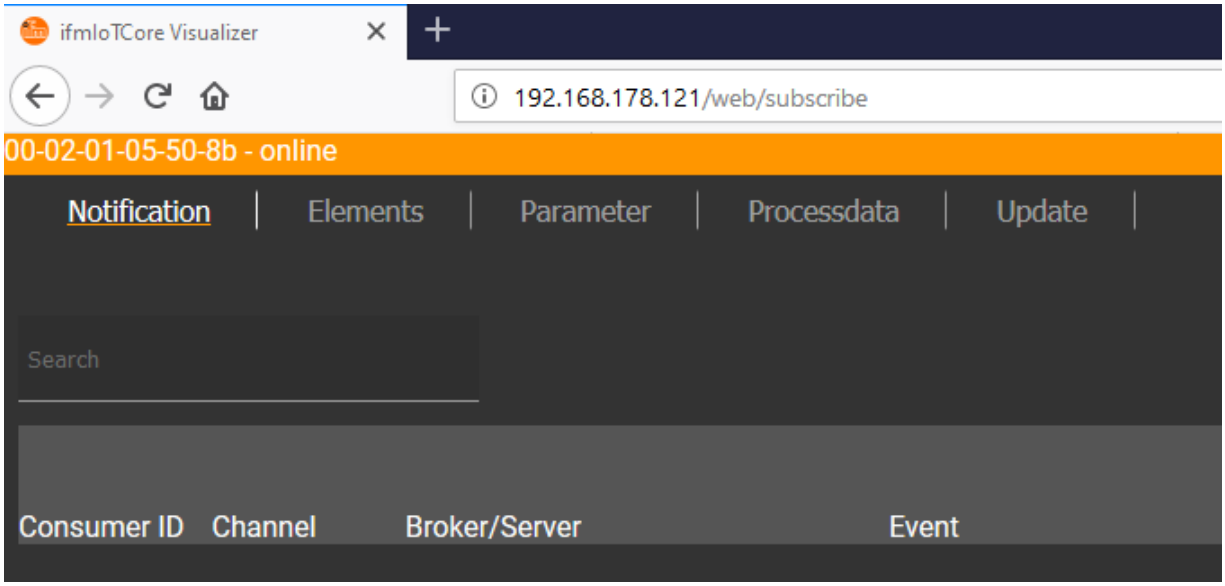
The ifm-IoT Core Visualizer of the IO-Link master provides a graphical user interface for accessing functions of the ifm-IoT Core.

To start the IoT Core Visualizer:

► Start web browser.

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

> Browser shows IoT Core Visualizer:



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

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

Managing notifications

61174

The menu page allows you to perform the following functions

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

Requirements:

- Iot-Core Visualizer has been started.

► Click on [Notification].

> The menu page for managing notifications appears.

> The menu page shows all registered notifications in a table

Creating a new notification

61175

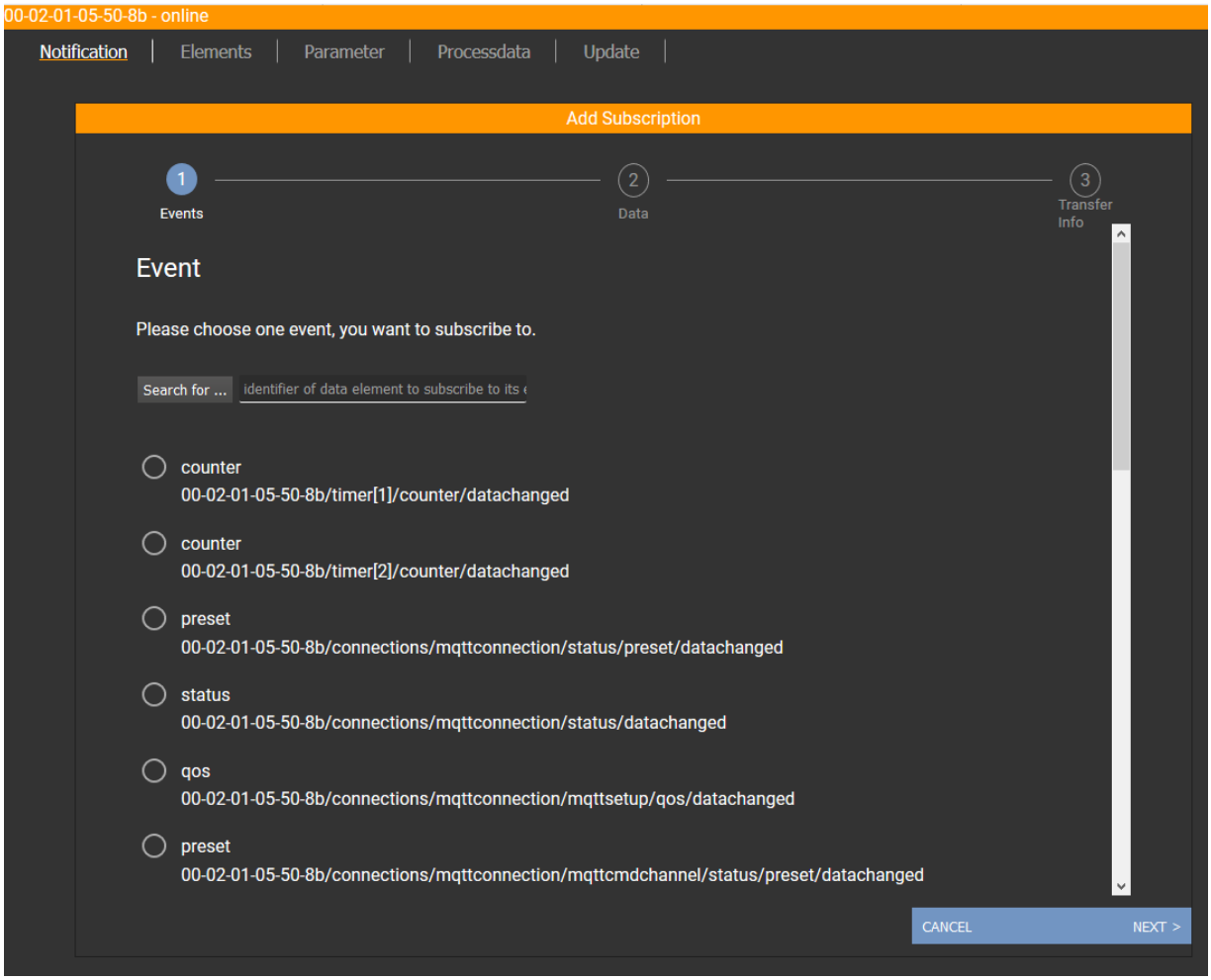
A wizard is used to register new notifications.

Requirements:

- The [Notification] menu page is open.

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

> The wizard for the creation of notifications appears.



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

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

Deleting a notification

61176

Requirements:

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

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

Searching for elements in the device tree

61177

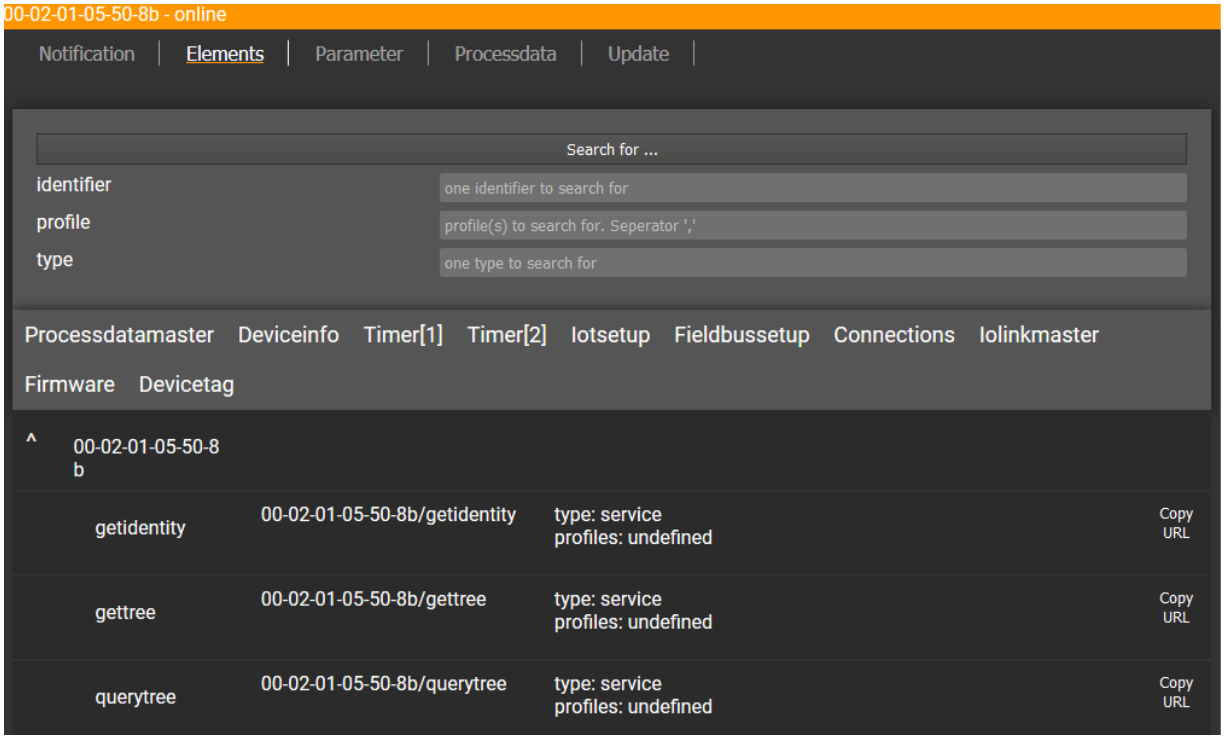
The [Elements] menu page allows you to search the device description for elements with specific properties (status, profile, name) and to output the results.

Requirements:

- Iot-Core Visualizer has been started.

► Click on [Elements].

> The input mask appears.



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

► Click on [Search for ...].

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

> The result list shows all elements found.

Configuring IO-Link the master

61178

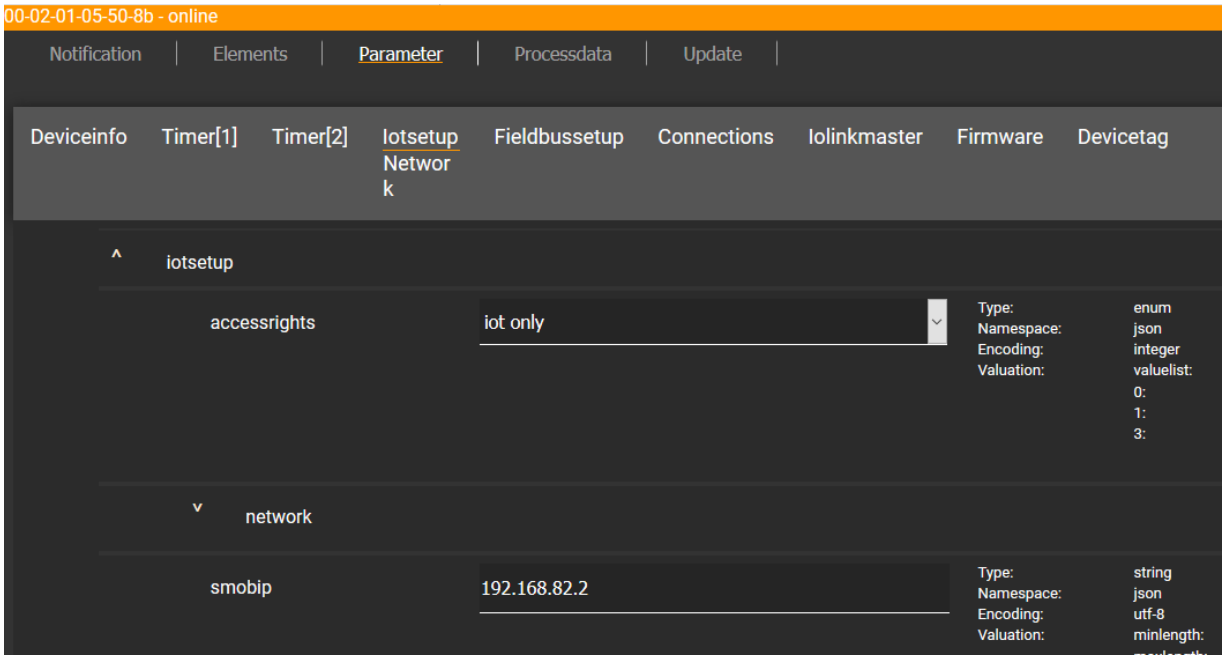
The [Parameter] menu page allows you to configure the IO-Link master.
Available options:

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

Requirements:

- Iot-Core Visualizer has been started.

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



To change a parameter:

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

Reading and writing process data

61179

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

Requirements:

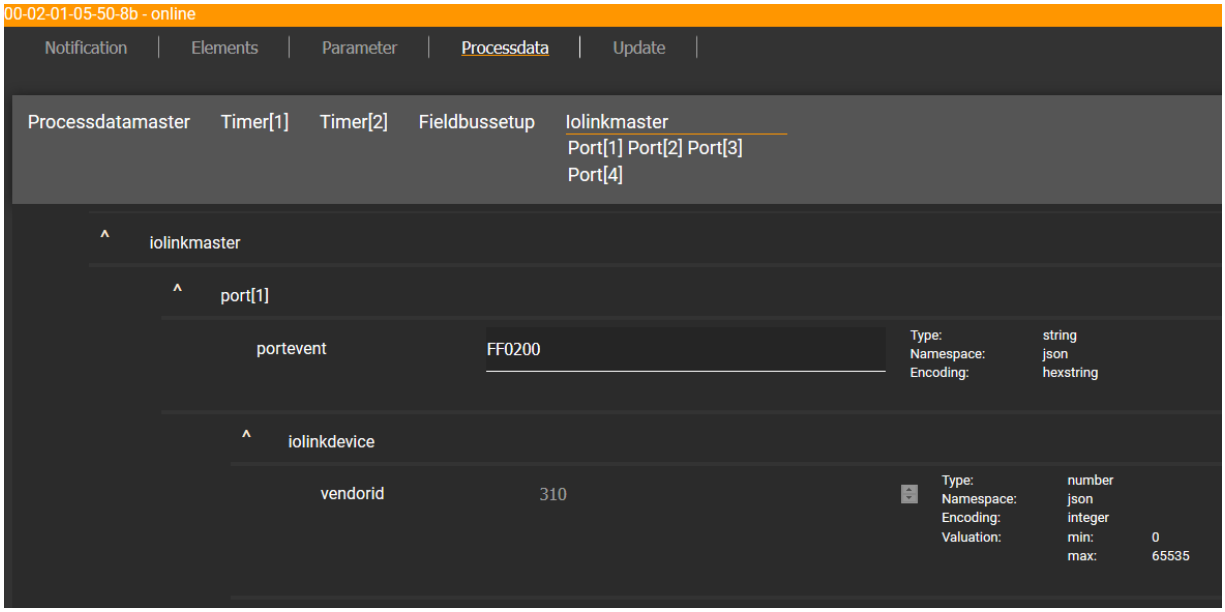
- Iot-Core Visualizer has been started.

► Click on [Processdata].

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

> The current process values are displayed.

> Editable process data can be changed.



To change the value of a process date:

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

► Change the process value.

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

> The changed process value is active.

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

Updating the firmware

61180

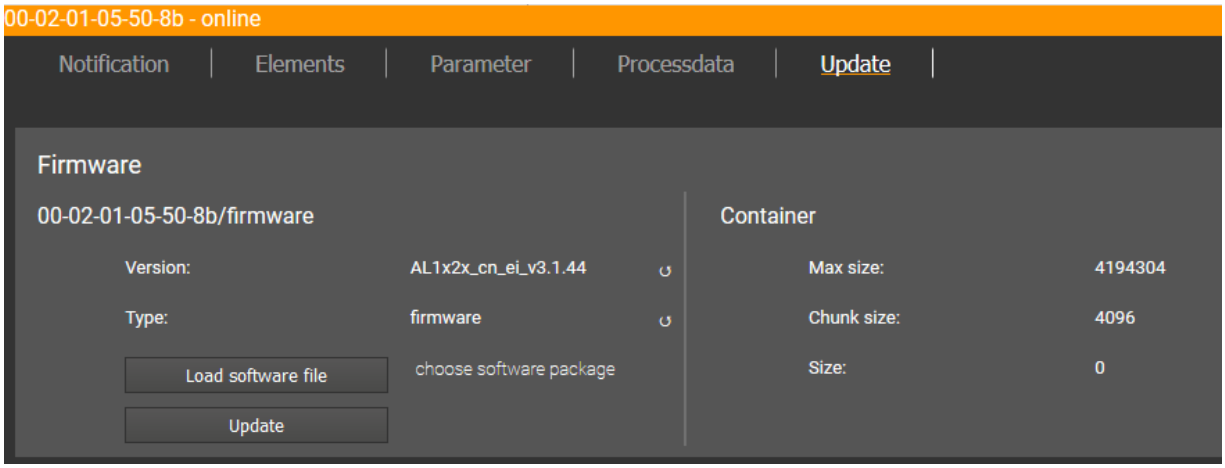
The [Update] menu page allows you to update the firmware of the IO-Link master:

Requirements:

- Iot-Core Visualizer has been started.

► Click on [Update].

> Menu page displays information about the current firmware version.



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

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

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

> The area shows the progress bar.

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

9.3 EtherNet/IP

Content

34391

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

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

9.3.1 Registration of the EDS file

34324

ifm provides an EDS file to integrate the AL1920 in a EtherNet/IP projection software. The user can download the EDS file from the ifm website (→ www.ifm.com). In the EDS file, all parameters, process data, and their valid value ranges are defined.

To add the AL1920 to the device catalogue of RSLogix5000:

► Download the EDS file of the AL1920 from the ifm website.

► Start RSLogix5000.

► Select [Tools] > [EDS Hardware Installation Tool].

> EDS Wizard appears.

► Register the downloaded EDS file of the AL1920 with the EDS Wizard.

> EDS Wizard installs the EDS file and adds the AL1920 to the device catalogue.

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

34392

The AL1920 is integrated as module of an I/O scanner in the EtherNet/IP project.

Requirements:

> The EDS file of the AL1920 is installed (→ **Registration of the EDS file** (→ p. 80)).

1 Create/open EtherNet/IP project

► Start RSLogix 5000.

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

2 Configure EtherNet/IP PLC and IO scanner

► Select and configure EtherNet/IP controller and IO scanner.

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

3 Integrate AL1920 in project

► In the Controller Organizer: Right mouse click on the IO scanner.

> Context menu appears.

► In the context menu: Select [New Module...].

> The window [Select Module Type] appears.

► Select AL1920 and click on [Create].

> The [New Module] window appears.

► Enter name and IP address of the AL1920.

► Click on [OK] to adopt the entered values.

> RSLogix 5000 adds AL1920 as sub-element of the IO scanner to the project.

4 Save the project

► Save EtherNet/IP project

9.3.3 Set connection types and RPI

34407

The IO-Link master supports different connection types (→ **Supported connection types** (→ p. 98)). The user can choose which object instances of the input assembly and the output assembly are used. This makes it possible to adapt the size of the transmitted and received data. Additionaly the Request Package Interval (RPI) can be selected.

To set the connection type:

Requirements:

> AL1920 is correctly integrated into the EtherNet/IP project (→ **Integrate the IO-Link Master into the EtherNet/IP project** (→ p. 80, "**Example: Reading properties of an element**" → p. 42)).

1 Open the module properties

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

> Dialogue window appears.

2 Set connection type

► Click on [Change...].

> The [Module Definition] dialogue window appears.

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

► Click on [OK] to apply the changes.

3 Change RPI

► Click on [Connection] tab.

> The connection settings appear.

► Select required time value from [RPI] list.

► Click on [OK] to apply the changes.

9.3.4 Configure AL1920

34357

The AL1920 is configured via the controller tags.

Requirements:

> AL1920 is correctly integrated in the EtherNet/IP project (→ **Integrate the IO-Link Master into the EtherNet/IP project** (→ p. 80, "**Example: Reading properties of an element**" → p. 42)).

1 Open controller tags

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

> [Controller Tags] window appears.

► In the tree view: Click on [AL1920:C].

> Controller tags for the configuration of the device appear.

2 Configure AL1920

► Set the following controller tags as required:

Name	Description	Possible values	
[AL1920:C.Communication_Profile]	The access rights to the parameter data, process data and events/diagnostic messages of the IO-Link master and the connected IO-Link devices	0x00	EtherNet/IP + LineRecorder ▪ EtherNet/IP and LR DEVICE have read and write access rights to parameters and process data ▪ EtherNet/IP and LR DEVICE have read access rights to events/alarms
		0x01	EtherNet/IP + LineRecorder (ro) ▪ EtherNet/IP has read and write access rights to parameters and process data ▪ EtherNet/IP has read access rights to events/alarms ▪ LR DEVICE only has read access rights to parameters, process data and events/alarms
		0x02	EtherNet/IP only ▪ EtherNet/IP has read and write access rights to parameters and process data ▪ EtherNet/IP has read access rights to events/alarms ▪ LR DEVICE has no access rights (parameters, process data, events/alarms, web interface, firmware update)
		0x03	Continue in Use Case previous setting is valid
[AL1920:C.Port_Process_Data_Size]	Length of the process input data and process output data	0x00	2 bytes input, 2 bytes output
		0x01	4 bytes input, 4 bytes output
		0x02	8 bytes input, 8 bytes output
		0x03	16 bytes input, 16 bytes output
		0x04	32 bytes input, 32 bytes output

► Save EtherNet/IP project

9.3.5 Configure IO-Link ports

34365

The IO-Link ports are configured via the controller tags. The user can configure each IO-Link port separately.

To configure the IO-Link ports:

Requirements:

> AL1920 is correctly integrated in the EtherNet/IP project (→ **Integrate the IO-Link Master into the EtherNet/IP project** (→ p. 80, "**Example: Reading properties of an element**" → p. 42)).

1 Open controller tags

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

> [Controller Tags] window appears.

► In the tree view: Click on [AL1920:C].

> Controller tags for the configuration of the device appear.

2 Configure IO-Link ports

► Configure the following tags for each IO-Link port at will:

Name	Description	Possible values	
[AL1920:C.Port_Mode_Port_x]	Operating mode of the IO-Link port	0x00	Interface deactivated
		0x01	Operation as digital input (DI)
		0x02	Operation as digital output (DO)
		0x03	Operation as IO-Link interface
[AL1920:C.Port_Cycle_Time_Port_x]	Cycle time of the data transmission between the IO-Link master and the IO-Link device	0x00	The device automatically sets the fastest possible cycle time
		0x01	2 milliseconds
		0x02	4 milliseconds
		0x03	8 milliseconds
		0x04	16 milliseconds
		0x05	32 milliseconds
		0x06	64 milliseconds
		0x07	128 milliseconds
[AL1920:C.Swap_Port_x]	Visualisation of the process data (EtherNet/IP uses Little Endian Format (Intel), IO-Link uses Big Endian Format (Motorola))	0x00	Byte swapping for IO-Linkdata deactivated
		0x01	Byte swapping for IO-Linkdata activated
[AL1920:C.Validation_Data_Storage_Port_x]	Supported IO-Link standard and behaviour of the IO-Link master when connecting new IO-Link devices to the IO-Link port	0x00	No validation
		0x01	Type compatible V1.0 device
		0x02	Type compatible V1.1 device
		0x03	Type compatible V1.1 device with Backup + Restore
		0x04	Type compatible V1.1 device with Restore
[AL1920:C.Vendor_ID_Port_x]	Vendor ID of the manufacturer of the device on the IO-Link port	0x0000...0xFFFF ifm electronic: 0x136	
[AL1920:C.Device_ID_Port_x]	Device ID of the device on the IO-Link port	0x000000...0xFFFFFF	

Name	Description	Possible values	
[AL1920:C.Fail_Safe_Mode_Port_x]	Fail-safe mode for output data when the EtherNet/IP connection is interrupted	0x00	No Failsafe
		0x01	Failsafe Reset Value
		0x02	Failsafe Old Value
		0x03	Failsafe with Pattern
[AL1920:C.Fail_Safe_Value_DO_Port_x]	Fail-safe value for the operating mode "digital output (DO)"	0x00	Failsafe Reset Value
		0x01	Failsafe Old Value
		0x02	Failsafe Set Value

x = 1...8

► Save EtherNet/IP project.

9.3.6 Configure IO-Link devices

34359

The AL1920 supports the configuration of the connected IO-Link devices from the EtherNet/IP projection software. For this, ifm offers the EtherNet/IP object "IO-Link Request" (→ **IO-Link requests (object class: 0x80)** (→ p. 135)). The object enables direct read and write access to IO-Link objects of the IO-Link device (Indexed Service Data Unit (ISDU)). The extent of the configurable parameters depends on the IO-Link device.

The following services are available:

Name	Description	Reference
Read request	Send a request to read an IO-Link object	→ **Read_ISDU** (→ p. 136)
Write request	Send a request to write an IO-Link object	→ **Write_ISDU** (→ p. 139)

Information for the execution of acyclic commands: → **Use acyclic services** (→ p. 87)
Available parameters of the IO-Link devices: → Operating instructions of the IO-Link device

9.3.7 Read process data

34360

The user can access the cyclic input data of the connected sensors and IO-Link devices via the controller tags of the AL1920.

To check the validity of the cyclic process data, evaluate the PQI byte (→ **Mapping: PQI** (→ p. 107)).
Even if the fieldbus connection is interrupted, the PQI byte indicates that the process data is valid. This may have an unintended impact on the control process.
► Take suitable measures to detect an interruption of the fieldbus connection.

To access the input data:
► Start RSLogix5000.
► Open a EtherNet/IP project.
► In the project tree: Mouse click on [Controller Tags] > [AL1920.I]
> The window shows the data structure with cyclic input data ([AL1920.I:Data]).
► Link process data to variables.

The mapping of the process data to the data structure [AL1920.I:Data] depends on the configured instance of the input assembly object (→ **Cyclic data** (→ p. 102)).

9.3.8 Write process data

34386

The user can access the cyclic output data of the connected actuators and IO-Link devices via the controller tags of the AL1920.

To check the validity of the cyclic process data, evaluate the PQI byte (→ **Mapping: PQI** (→ p. 107)).
Even if the fieldbus connection is interrupted, the PQI byte indicates that the process data is valid. This may have an unintended impact on the control process.
► Take suitable measures to detect an interruption of the fieldbus connection.

To access the cyclic output data:
► start RSLogix5000.
► Open a EtherNet/IP project.
► In the project tree: Mouse click on [Controller Tags] > [AL1920.O]
> The window shows the data structure with cyclic output data ([AL1920.O:Data]).
► Link process data to variables.

The mapping of the process data to the data structure [AL1920:O.Data] depends on the configured instance of the input assembly object (→ **Cyclic data** (→ p. 102)).

9.3.9 Read diagnostic and status information

34364

Diagnostic and status information is a part of the cyclically transmitted process data. The input assembly includes the following information:

Byte	Content
2	Indication of short circuit/overload of the IO-Link ports X01...X08
3	Status indication of the voltage supply of the device
43	Port X01: Status information + events
58	Port X02: Status information + events
73	Port X03: Status information + events
88	Port X04: Status information + events
103	Port X05: Status information + events
118	Port X06: Status information + events
133	Port X07: Status information + events
148	Port X08: Status information + events

To access the cyclically transmitted diagnostic and status information:

► Starting RSLogix5000.

► Open a EtherNet/IP project.

► In the project tree: Mouse click on [Controller Tags] > [AL1920.I]

> The window shows cyclic input data (Input Assembly).

► Link diagnostic and status information with variables.

Mapping of the diagnostic and status information on the data structure [AL1920.C:I]: → **Cyclic data** (→ p. 102).

9.3.10 EtherNet/IP: Programmers' notes

34400

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

- Read cyclic input and output data of the IO-Link devices
- Read diagnostic and status information
- Change parameters of the IO-Link port of the AL1920
- Read and change parameters of the connected IO-Link devices

The following sections show the available options.

Supported configuration modes

34383

The AL1920 supports the following EtherNet/IP configuration modes:

- **Top down**
  - Configuration of the EtherNet/IP slave with the EtherNet/IP projection software (Configuration Assembly)
  - EtherNet/IP plc transmitts the created configuration to the EtherNet/IP slave, where it is stored
- **Independent**
  - Configuration of the EtherNet/IP slave with LR DEVICE oder IoT core
  - Confiuration Assembly in EtherNet/IP project is not evaluated

Use acyclic services

34381

The AL1920 offers the following options to execute acyclic commands:

Command channels in cyclic process data

34318

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

Principle of the command channels

34343

General process of an acyclic communication:

1 Write command request

► In the request channel: write requested command data (without [Trigger] bit)

► Set [Trigger] = 1.

> Change of [Trigger] = 1 indicates a new command.

> In the response channel: all bytes are set to 0.

> Command processing is started.

2 Check status

► In the response channel: check [Handshake] bit.

- If [Handshake] <> 0: command processing completed, continue with step 3.
- If [Handshake] == 0: command is processed, repeat step 2.

3 Read command response

► In the response channel: read responsed user data.

► In the request channel: set [Trigger] = 0.

Acyclic port commands

34336

For the acyclic access to the configuration of the IO-Link ports of the AL1920, the following commands are available:

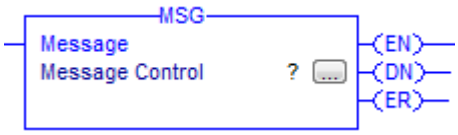
Command	Description	Reference
Set mode	Set the operating type of the IO-Link port	→ **Command 0x10 – Set mode** (→ p. 119)
Set Validation ID / Data Storage	Adjust the supported IO-Link standard and the behaviour of the IO-Link master when connecting a new IO-Link device to the IO-Link port	→ **Command 0x20 – Set validation ID / data storage** (→ p. 121)
Set fail-safe data pattern	Behaviour of the outputs when the EtherNet/IP connection is interrupted and setting of the corresponding fail-safe values	→ **Command 0x30 – Set fail-safe data pattern** (→ p. 123)

The port commands use the same mechanisms as the acyclic command channel (→ **Acyclic command channel** (→ p. 113)).

EtherNet/IP mechanisms for acyclic commands

34404

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



Parameters of the available field bus objects: → **Field bus objects** (→ p. 125)

For detailed information about the `Message (MSG)` command: → Operating instructions RSLogix 5000

10 Operation

Content

34061

10.1 Using web-based management

61181

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

- Status information of the ports
- Access to product page of connected IO-Link devices (only ifm devices)
- Diagnostic information of the device
- Version information of the installed firmware components

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

► Connect the IO-Link master to the laptop / PC via the IoT port.

► Optional: Check the IP settings of the IoT interface.

► Start web browser.

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

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

11 Maintenance, repair and disposal

Content

51990

The operation of the unit is maintenance-free.

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

11.1 Cleaning process

51991

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

11.2 Updating the firmware

61183

The firmware of the IO-Link master can be updated via the IoT Core Visualizer↘→ **Updating the firmware** (→ p. 79, → p. 90)).

11.3 Replacing IO-Link device

34182

To replace an IO-Link device:

Requirement:

> New IO-Link device is with factory settings.
> New IO-Link device supports IO-Link standard 1.1 or higher.

1 Set data storage

► Set the following parameters of the IO-Link port
▪ Set Validation and Data Storage to [Type compatible V1.1 device with Restore] or [Type compatible V1.1. device with Backup + Restore]
▪ Set correct values to [Vendor ID] and [Device ID] according to properties of the IO-Link device.
► Save changes.

2 Replace IO-Link device

► Disconnect old IO-Link device from IO-Link master.
► Connect new IO-Link device with the same IO-Link port of the AL1920.
> IO-Link master copies parameter values from the data memory to the new IO-Link device.

12 Factory settings

34509

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

Parameter	Factory setting
[IP address] (EtherNet/IP)	192.168.1.250
[Subnet mask] (EtherNet/IP)	255.255.255.0
[IP gateway address] (EtherNet/IP)	0.0.0.0
[IP address] (IoT interface)	169.254.X.X
[Subnet mask] (IoT interface)	255.255.0.0
[IP gateway address] (IoT interface)	0.0.0.0
[EtherNet/IP name]	blank
Data memory (Data Storage)	empty

13 Accessories

33870

List of accessories of AL1920: → www.ifm.com > Product page > Accessories

14 Appendix

Content

33879

14.1 Technical data

Content

34188

14.1.1 Application

33878

Application	
Application	I/O modules for control cabinet
Daisy-chain function	Fieldbus interface

14.1.2 Electrical data

33808

Electrical data	
Operating voltage [V]	20...30 DC; (US; to SELV/PELV)
Current Consumption [mA]	300...3900; (US)
Protection class	III
Sensor supply US	
Max. current load total [A]	3.6

14.1.3 Inputs / outputs

34068

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

14.1.4 Inputs

34069

Inputs	
Number of digital inputs	16; (IO-Link Port Class A)
Switching level high [V]	11...30
Switching level low [V]	0...5
Digital inputs protected against short circuits	yes

14.1.5 Outputs

34053

Outputs	
Number of digital outputs	8; (IO-Link Port Class A)
Max. current load per output [mA]	300
Short-circuit protection	yes

14.1.6 Interfaces

34389

Interfaces	
Communication interface	Ethernet; IO-Link
Communication interface	IO-Link; TCP/IP; EtherNet/IP
Ethernet	
Transmission standard	10Base-T; 100Base-TX
Transmission rate [MBit/s]	10; 100
Protocol	TCP/IP; EtherNet/IP
Factory settings	▪ IP address: 192.168.1.250 ▪ Subnet mask: 255.255.255.0 ▪ Gateway IP address: 0.0.0.0 ▪ MAC address: see type label
IO-Link master	
Type of transmission	COM 1 / COM 2 / COM 3
IO-Link revision	V1.1
Number of ports Class A	8
IoT interface	
Transmission standard	10Base-T; 100Base-TX
Transmission rate [Mbits/s]	10; 100
Protocol	DCP, DCHP, Auto IP
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

14.1.7 Environmental conditions

33811

Environmental conditions	
Applications	Control cabinet
Ambient temperature [°C]	-25...65
Storage temperature [°C]	-25...85
Max. perm. relative air humidity [%]	90, linearly decreasing to 50 % (40 °C)
Max. height above sea level [m]	2000
Protection	IP 20
Degree of soiling	2

14.1.8 Approvals / tests

33877

Approval / tests	
EMC	▪ EN 61000-6-2 ▪ EN 61000-6-4
MTTF [Years]	90

14.1.9 Mechanical data

34050

Mechanical data	
Weight [g]	329
Materials	Housing: PA

14.1.10 Electrical connection

33807

Voltage supply IN X31	
Plug and socket connection	COMBICON
Wiring	1: GND (US) 2: GND (US) 3: + 24 V DC (US) 4: + 24 V DC (US)
Process connection IO-Link ports Class A X01...X08	
Plug and socket connection	COMBICON
Wiring	1: Sensor supply (US) L+ 2: DI 3: Sensor supply (US) L- 4: C/Q IO-Link
Ethernet IN / OUT X21, X22	
Plug and socket connection	RJ-45
IoT X32	
Plug and socket connection	RJ-45

14.2 EtherNet/IP

Content

33674

14.2.1 Supported connection types

34410

Name	Configuration Assemby	Input Assembly - Instance	Output Assembly - Instance
Exclusive Owner IO-Acyc-Diag	199	100	150
Exclusive Owner IO-Acyc	199	101	150
Exclusive Owner IO	199	102	151
Input only	199	100	193 (empty)
Listen only	199	100	192 (empty)

14.2.2 Parameter data

Content

34170

Configuration Assembly (Instance 199)

34358

The values of the Configuration Assembly are set in RSLogix 5000 via the controller tags of the EtherNet/IP project.

Bytes	Contents
0	Access Rights
1	Process Data Length
2...13	Port X01: Port Configuration (→ **Mapping: Port configuration** (→ p. 100))
14...25	Port X02: Port Configuration (→ **Mapping: Port configuration** (→ p. 100))
26...37	Port X03: Port Configuration (→ **Mapping: Port configuration** (→ p. 100))
38...49	Port X04: Port Configuration (→ **Mapping: Port configuration** (→ p. 100))
50...61	Port X05: Port Configuration (→ **Mapping: Port configuration** (→ p. 100))
62...73	Port X06: Port Configuration (→ **Mapping: Port configuration** (→ p. 100))
74...85	Port X07: Port Configuration (→ **Mapping: Port configuration** (→ p. 100))
86...97	Port X08: Port Configuration (→ **Mapping: Port configuration** (→ p. 100))

Legend:

▪ [Access Rights]	Access rights to parameter, process data and events / diagnostics data of the IO-Link master and the connected IO-Link devices	1 Byte	0x00	EtherNet/IP + IoT
			0x01	EtherNet/IP + IoT (ro)
			0x02	EtherNet/IP only
			0x03	Keep setting (default)
▪ [Process Data Length]	Length of the process input data and process output data	1 Byte	0x00	2 Bytes Input / 2 Bytes Output Data ▪ Input Assembly: 206 Bytes ▪ Output Assembly: 62 Bytes
			0x01	4 Bytes Input / 4 Bytes Output Data ▪ Input Assembly: 222 Bytes ▪ Output Assembly: 78 Bytes
			0x02	8 Bytes Input / 8 Bytes Output Data ▪ Input Assembly: 254 Bytes ▪ Output Assembly: 110 Bytes
			0x03	16 Bytes Input / 16 Bytes Output Data ▪ Input Assembly: 318 Bytes ▪ Output Assembly: 174 Bytes
			0x04	32 Bytes Input / 32 Bytes Output Data ▪ Input Assembly: 446 Bytes ▪ Output Assembly: 302 Bytes

Mapping: Port configuration

34394

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

Legend:

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

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

14.2.3 Cyclic data

Content

33814

Input assembly (Instance 100): I/O data + acyclic data + diagnosis data

34344

Byte	Content
0...1	Port X01...X08: Digital input - clamp 2 / 4 (DI) (→ **Mapping: digital input data (DI)** (→ p. 106))
2...3	Status information (→ **Mapping: Status information** (→ p. 106))
4...45	Acyclic command area: Response channel (→ **Response channel** (→ p. 115))
46...47	Port X01: PQI (→ **Mapping: PQI** (→ p. 107))
48...63	Port X01: Diagnostic, vendor ID, device ID, events (→ **Mapping: IO-Link device information + events** (→ p. 108))
64...65	Port X02: PQI (→ **Mapping: PQI** (→ p. 107))
66...81	Port X02: Diagnostic, vendor ID, device ID, results (→ **Mapping: IO-Link device information + events** (→ p. 108))
82...83	Port X03: PQI (→ **Mapping: PQI** (→ p. 107))
84...99	Port X03: Diagnostic, vendor ID, device ID, events (→ **Mapping: IO-Link device information + events** (→ p. 108))
100...101	Port X04: PQI (→ **Mapping: PQI** (→ p. 107))
102...117	Port X04: Diagnostic, vendor ID, device ID, events (→ **Mapping: IO-Link device information + events** (→ p. 108))
118...119	Port X05: PQI (→ **Mapping: PQI** (→ p. 107))
120...135	Port X05: Diagnostic, vendor ID, device ID, events (→ **Mapping: IO-Link device information + events** (→ p. 108))
136...137	Port X06: PQI (→ **Mapping: PQI** (→ p. 107))
138...153	Port X06: Diagnostic, vendor ID, device ID, events (→ **Mapping: IO-Link device information + events** (→ p. 108))
154...155	Port X07: PQI (→ **Mapping: PQI** (→ p. 107))
156...171	Port X07: Diagnostic, vendor ID, device ID, events (→ **Mapping: IO-Link device information + events** (→ p. 108))
172...173	Port X08: PQI (→ **Mapping: PQI** (→ p. 107))
174...189	Port X08: Diagnostic, vendor ID, device ID, events (→ **Mapping: IO-Link device information + events** (→ p. 108))
190	Port X01: Input data IO-Link (n bytes)
190+n	Port X02: Input data IO-Link (n bytes)
190+2n	Port X03: Input data IO-Link (n bytes)
190+3n	Port X04: Input data IO-Link (n bytes)
190+4n	Port X05: Input data IO-Link (n bytes)
190+5n	Port X06: Input data IO-Link (n bytes)
190+6n	Port X07: Input data IO-Link (n bytes)
190+7n	Port X08: Input data IO-Link (n bytes)

Legend:
n = [2,4,8,16,32]; is determined by the parameter [Prozess_Data_Length] (→ **Configuration Assembly (Instance 199)** (→ p. 99)

Input Assembly (Instance 101): I/O data + acyclic data

34369

Byte	Content
0...1	Port X01...X08: Digital Input clamp 2 / 4 (DI) (→ **Mapping: digital input data (DI)** (→ p. 106))
2...3	Status information (→ **Mapping: Status information** (→ p. 106))
4...45	Acyclic commando area: Response channel (→ **Response channel** (→ p. 115))
46...47	Port X01: PQI (→ **Mapping: PQI** (→ p. 107))
48...49	Port X02: PQI (→ **Mapping: PQI** (→ p. 107))
50...51	Port X03: PQI (→ **Mapping: PQI** (→ p. 107))
52...53	Port X04: PQI (→ **Mapping: PQI** (→ p. 107))
54...55	Port X05: PQI (→ **Mapping: PQI** (→ p. 107))
56...57	Port X06: PQI (→ **Mapping: PQI** (→ p. 107))
58...59	Port X07: PQI (→ **Mapping: PQI** (→ p. 107))
60...61	Port X08: PQI (→ **Mapping: PQI** (→ p. 107))
62	Port X01: Input data IO-Link (n bytes)
62+n	Port X02: Input data IO-Link (n bytes)
62+2n	Port X03: Input data IO-Link (n bytes)
62+3n	Port X04: Input data IO-Link (n bytes)
62+4n	Port X05: Input data IO-Link (n bytes)
62+5n	Port X06: Input data IO-Link (n bytes)
62+6n	Port X07: Input data IO-Link (n bytes)
62+7n	Port X08: Input data IO-Link (n bytes)

Legend:

n = [2,4,8,16,32]; is determined by the parameter [Prozess Data Length] (→ **Configuration Assembly (Instance 199)** (→ p. 99))

Input Assembly (Instance 102): I/O data

34372

Byte	Content
0...1	Digital inputs of the IO-Link ports in DI operating mode (→ **Mapping: digital input data (DI)** (→ p. 106))
2...3	Status information (→ **Mapping: Status information** (→ p. 106))
4...5	Port X01: PQI (→ **Mapping: PQI** (→ p. 107))
6...7	Port X02: PQI (→ **Mapping: PQI** (→ p. 107))
8...9	Port X03: PQI (→ **Mapping: PQI** (→ p. 107))
10...11	Port X04: PQI (→ **Mapping: PQI** (→ p. 107))
12...13	Port X05: PQI (→ **Mapping: PQI** (→ p. 107))
14...15	Port X06: PQI (→ **Mapping: PQI** (→ p. 107))
16...17	Port X07: PQI (→ **Mapping: PQI** (→ p. 107))
18...19	Port X08: PQI (→ **Mapping: PQI** (→ p. 107))
20	Port X01: Cyclic input data (n bytes)
20+n	Port X02: Cyclic input data (n bytes)
20+2n	Port X03: Cyclic input data (n bytes)
20+3n	Port X04: Cyclic input data (n bytes)
20+4n	Port X05: Cyclic input data (n bytes)
20+5n	Port X06: Cyclic input data (n bytes)
20+6n	Port X07: Cyclic input data (n bytes)
20+7n	Port X08: cyclic input data (n bytes)

Legend:

n = [2,4,8,16,32]; is determined by the parameter [Prozess Data Length] (→ **Configuration Assembly (Instance 199)** (→ p. 99))

Mapping: digital input data (DI)

34380

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
X08: clamp 4	X07: clamp 4	X06: clamp4	X05: clamp 4	X04: clamp 4	X03: clamp 4	X02: clamp 4	X01: clamp 4
X08: clamp 2	X05: clamp 2	X06: clamp 2	X05: clamp 2	X04: clamp 2	X03: clamp 2	X02: clamp 2	X01: clamp 2

Legend:

▪ [clamp 4]	Signal level on clamp 4 of the IO-Link port	1 bit	0x0	LOW
			0x1	HIGH
▪ [clamp 2]	Signal level on clamp 2 of the IO-Link port	1 bit	0x0	LOW
			0x1	HIGH

Mapping: Status information

34396

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
X08: Short / OL	X07: Short / OL	X06: Short / OL	X05: Short /OL	X04: Short / OL	X03: Short / OL	X02: Short / OL	X01: Short / OL
reserved	reserved	reserved	reserved	reserved	reserved	Sensor PWR	AUX PWR

Legend:

▪ [Short / OL]	Occurrence of a short circuit or of an overvoltage on the IO-Link port	1 bit	0x0	no error
			0x1	short circuit or overvoltage detected
▪ [Sensor PWR]	Status of the supply voltage US	1 bit	0x0	no error
			0x1	error
▪ [AUX PWR]	Status of the supply voltage UA	1 bit	0x0	UA present and no error
			0x1	no error

Mapping: PQI

34393

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Diagnosis present	Wrong PD Output Length	Wrong PD Input Length	Wrong Cycle Time	Wrong VID / DID	Invalid Data Bit	Dev Not Conn	IOL Mode
reserved							

Legend:

▪ [IOL Mode]	Operating mode of the IO-Link port	1 bit	0x0	not IO-Link
			0x1	IO-Link
▪ [Dev Not Conn]	Connection between IO-Link Device and IO-Link port	1 bit	0x0	connected
			0x1	not connected
▪ [Invalid Data]	Status of the process input data on the IO-Link port	1 bit	0x0	valid Data
			0x1	invalid Data
▪ [Wrong VID/DID]	Evaluation, whether actual and projected Vendor ID and Device ID match	1 bit	0x0	OK
			0x1	wrong VID and/or DID
▪ [Wrong Cycle Time]	Evaluation, whether actual and projected cycle time match	1 bit	0x0	OK
			0x1	wrong cycle time
▪ [Wrong PD Input Length]	Evaluation, whether actual and projected input process data length match	1 bit	0x0	OK
			0x1	projected length too small
▪ [Wrong PD Output Length]	Evaluation, whether actal and projected output process data length match	1 bit	0x0	OK
			0x1	projected length too small
▪ [Diagnosis present]	Signals a new diagnosis event (Coming Event, Single Shot Event) ▪ Coming Events are removed when if the related Disappearing Event appears ▪ Single Shot Events are removed automatically	1 Bit	0x0	no event
			0x1	New event present

Mapping: IO-Link device information + events

34413

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

Legend:

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

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

Output assembly (Instance 150): I/O data + acyclic data

34370

Byte	Content
0	Port X01...X08: Digital output - clamp 4 (DO) (→ **Mapping: Digital output data (DO)** (→ p. 112))
1	reserved
2	reserved
3	reserved
4...45	Acyclic command area: Request channel (→ **Request channel** (→ p. 114))
46	Port X01: Output data IO-Link (n bytes)
46+n	Port X02: Output data IO-Link (n bytes)
46+2n	Port X03: Output data IO-Link (n bytes)
46+3n	Port X04: Output data IO-Link (n bytes)
46+4n	Port X05: Output data IO-Link (n bytes)
46+5n	Port X06: Output data IO-Link (n bytes)
46+6n	Port X07: Output data IO-Link (n bytes)
46+7n	Port X08: Output data IO-Link (n bytes)

Legend:

n = [2,4,8,16,32]; is determined by the parameter [Prozess_Data_Length] (→ **Configuration Assembly (Instance 199)** (→ p. 99)

Output Assembly (Instance 151): I/O data

34341

Byte	Content
0	Port X01...X08: Digital output - clamp 4 (DO) (→ **Mapping: Digital output data (DO)** (→ p. 112))
1	reserved
2	Port X01: Output data IO-Link (n bytes)
2+n	Port X02: Output data IO-Link (n bytes)
2+2n	Port X03: Output data IO-Link (n bytes)
2+3n	Port X04: Output data IO-Link (n bytes)
2+4n	Port X05: Output data IO-Link (n bytes)
2+5n	Port X06: Output data IO-Link (n bytes)
2+6n	Port X07: Output data IO-Link (n bytes)
2+7n	Port X08: Output data IO-Link (n bytes)

Legend:

n = [2,4,8,16,32]; is determined by the parameter [Process_Data_Length] (→ **Configuration Assembly (Instance 199)** (→ p. 99)

Mapping: Digital output data (DO)

34411

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
X08: clamp 4	X07: clamp 4	X06: clamp4	X05: clamp 4	X04: clamp 4	X03: clamp 4	X02: clamp 4	X01: clamp 4

Legend:

- [clamp 4] Signal level on clamp 4 of the IO-Link port 1 bit 0x0 LOW
  0x1 HIGH

14.2.4 Acyclic data

Content

33868

Acyclic command channel

34325

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

Object	Contents	Bytes	Access
Output assembly	Request channel (field bus PLC >>> IO-Link master) → **Request channel** (→ p. 114)	4...45	r/w
Input assembly	Response channel (IO-Link master >>> fieldbus PLC) → **Response channel** (→ p. 115)	4...45	r

Legend:

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

Request channel

34398

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

Legend:

▪ [Port No.]	Number of the IO-Link port Port No. = 0x1234 ▪ Port No. (MSB) = 0x12 ▪ Port No. (LSB) = 0x34	1 Word	0x0001 0x0002 ... 0x0008	Port X01 Port X02 Port X08
▪ [Index]	Index of the IO-Link object Index = 0x1234 ▪ Index (MSB) = 0x12 ▪ Index (LSB) = 0x34	1 Word	0x0000...0xFFFF	
▪ [Subindex]	Subindex of the IO-Link object Subindex = 0x1234 ▪ Subindex (MSB) = 0x12 ▪ Subindex (LSB) = 0x34	1 Word	0x0000...0xFFFF	
▪ [Trigger]	Control of the command execution	1 Bit	0x0 0x1	do not process command execute command
▪ [Command ID]	Command number	7 Bit	0x01 0x02	read write
▪ [Length of user data (number of bytes)]	Number of bytes that contain relevant user data	1 Byte	0x00 ... 0x20	0 bytes 32 bytes
▪ [Data (byte n)]	User data	1 Byte	per byte: 0x00...0xFF	

Response channel

34401

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

Legend:

▪ [Port No.]	Number of the IO-Link port Port No. = 0x1234 ▪ Port No. (MSB) = 0x12 ▪ Port No. (LSB) = 0x34	1 word	0x0001 0x0002 ... 0x0008	Port X01 Port X02 Port X08
▪ [Index]	Index of the IO-Link object Index = 0x1234 ▪ Index (MSB) = 0x12 ▪ Index (LSB) = 0x34	1 word	0x0000...0xFFFF	
▪ [Subindex]	Subindex of the IO-Link object Subindex = 0x1234 ▪ Subindex (MSB) = 0x12 ▪ Subindex (LSB) = 0x34	1 word	0x0000...0xFFFF	
▪ [Handshake]	Validity of the IO-Link response data	1 bit	0x0 0x1	Data invalid Data valid
▪ [Command ID]	Command number	7 bits	0x01 0x02	Read Write
▪ [Result]	Status of the command processing	8 bits	0x00 0x0F 0xFF	OK OK, data read >32 bytes Error occurred
▪ [Length of response data (number of bytes)]	Number of bytes that contain relevant user data	1 byte	0x00 ... 0x20	0 bytes 32 bytes

▪ [Data (byte 0) or Error Code]	User data (byte 0) or error codes	1 byte	User data: 0x00...0xFF Error Code: → **Error codes** (→ p. 117)
▪ [Data (byte 1) or Additional Code]	User data (byte1) or additional error codes	1 byte	User data: 0x00...0xFF Additional Code: → **Additional Codes** (→ p. 117)
▪ [Data (byte n)]	User data (byte n)	1 byte	0x00...0xFF

Error codes

34342

Error code	Description
0x71	Service not available (unknown command has been sent to the IO-Link port)
0x72	Port blocked (another cyclic process accesses the IO-Link port)
0x73	Forbidden (access rights don't allow command processing)
0x74	Invalid data (wrong parameter has been sent in the command)
0x76	Wrong port (wrong port number)
0x77	Wrong port function (wrong port function or wrong parameter has been sent to the device)
0x78	Invalid length (set length is > 0x20)
0x80	Error in the device application; observe additional code (→ **Additional Codes** (→ p. 117))

Additional Codes

54584

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

Additional Codes are only available, if Error Code = 0x80 (→ **Error codes** (→ p. 117))

Acyclic commands

Content

34331

Command 0x10 – Set mode

34322

The command changes the operating mode of an IO-Link port of the AL1920.

Corresponding parameter: [Port Mode] (→ **Mapping: Port configuration** (→ p. 100))

Command request

34314

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

Legend:

▪ [Port No.]	Number of the IO-Link port Port No. = 0x1234 ▪ Port No. (MSB) = 0x12 ▪ Port No. (LSB) = 0x34	1 word	0x0001 0x0002 ... 0x0008	Port X01 Port X02 Port X08
▪ [Trigger]	Control of the command execution	1 bit	0x0 0x1	do not process command execute command
▪ [Target Mode]	Operating type of the IO-Link port	1 byte	0x00 0x01 0x02 0x03	deactivated operation as digital input (DI) operation as digital output (DO) operation as IO-Link intervals

Command response

34316

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

Legend:

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

  1 word | 0x0001 Port X01 | 0x0002 Port X02 | ... | 0x0008 Port X08
- [Handshake] Status of the execution of the command — 1 bit | 0x0 command is executed | 0x1 execution of the command was successful
- [Result] Error indication — 1 byte | 0x00 no error | 0x01 error occured
- [Target Mode] Operating type of the IO-Link port — 1 byte | 0x00 deactivated | 0x01 operation as digital input (DI) | 0x02 operation as digital output (DO) | 0x03 operation as IO-Link intervals

Command 0x20 – Set validation ID / data storage

34321

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

Corresponding parameter: [Validation ID] (→ **Mapping: Port configuration** (→ p. 100))

Command request

34315

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

Legend:

▪ [Port No.]	Number of the IO-Link port Port No. = 0x1234 ▪ Port No. (MSB) = 0x12 ▪ Port No. (LSB) = 0x34	1 word	0x0001 0x0002 ... 0x0008	Port X01 Port X02 Port X08
▪ [Trigger]	Control command execution	1 Bit	0x0 0x1	do not process command execute command
▪ [Validation ID]	Behaviour of the IO-Link master when connecting an IO-Link device to the IO-Link port	1 byte	0x00 0x01 0x02 0x03 0x04	No check Type compatible V1.0 Device Type compatible V1.1 Device Type compatible V1.1 Device with Backup + Restore Type compatible V1.1 Device with Restore

Command response

34362

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

Legend:

▪ [Port No.]	Number of the IO-Link port Port No. = 0x1234 ▪ Port No. (MSB) = 0x12 ▪ Port No. (LSB) = 0x34	1 word	0x0001 0x0002 ... 0x0008	Port X01 Port X02 Port X08
▪ [Handshake]	Status of the execution of the command	1 bit	0x0 0x1	command is executed execution of the command was successful
▪ [Result]	Error indication	1 byte	0x00 0x01	no error errors occurred
▪ [Validation ID]	Behaviour of the IO-Link master when connecting an IO-Link device to the IO-Link port	1 byte	0x00 0x01 0x02 0x03 0x04	No check Type compatible V1.0 Device Type compatible V1.1 Device Type compatible V1.1 Device with Backup + Restore Type compatible V1.1 Device with Restore

Command 0x30 – Set fail-safe data pattern

34379

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

Corresponding parameter: [Fail-safe Mode] (→ **Mapping: Port configuration** (→ p. 100)
The number of the required fail-safe values results from the size of the output data (→ **Configuration Assembly (Instance 199)** (→ p. 99)).

Command request

34317

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

Legend:

▪ [Port No.]	Number of the IO-Link port Port No. = 0x1234 ▪ Port No. (MSB) = 0x12 ▪ Port No. (LSB) = 0x34	1 word	0x0001 0x0002 ... 0x0008	Port X01 Port X02 Port X08
▪ [Trigger]	Control command execution	1 bit	0x0 0x1	do not process command execute command
▪ [Fail-safe Mode]	Behaviour of the outputs when the EtherNet/IP connection is interrupted and setting of the corresponding fail-safe values	1 byte	0x00 0x01 0x02 0x03	No Fail-safe Fail-safe Reset Value Fail-safe Old Value Fail-safe with Pattern
▪ [Byte Length N]	Number of the bytes that contain fail-safe values	1 byte	0x00 ... 0x20	0 Bytes 32 Bytes
▪ [Fail-safe Data (Byte n)]	Fail-Safe value n (n = 0...31)	1 bytet	per byte: 0x00...0xFF	

Command response

34361

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

Legend:

▪ [Port No.]	Number of the IO-Link port Port No. = 0x1234 ▪ Port No. (MSB) = 0x12 ▪ Port No. (LSB) = 0x34	1 word	0x0001 0x0002 ... 0x0008	Port X01 Port X02 Port X08
▪ [Handshake]	Status of the execution of the command	1 bit	0x0 0x1	command is executed execution of the command was successful
▪ [Result]	Error indication	1 byte	0x00 0x01	no error error occured
▪ [Fail-safe Mode]	Behaviour of the outputs when the EtherNet/IP connection is interrupted	1 byte	0x00 0x01 0x02 0x03	No Fail-safe Fail-safe Reset Value Fail-safe Old Value Fail-safe with Pattern

Field bus objects

Content

34352

CIP class services

34335

The device supports the following class and instance services:

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

CIP object classes

34334

The device supports the following CIP object classes:

Class code		Object type	Reference
dec	hex		
01	01	Identity Object	→ **Identity Object (object class: 0x01)** (→ p. 127)
02	02	Message Router Object	→ **Message Router Object (object class: 0x02)** (→ p. 129)
04	04	Assembly Object	→ **Assembly Object (object class: 0x04)** (→ p. 130)
06	06	Connection Manager Object	→ **Connection Manager Object (object class: 0x06)** (→ p. 132)
71	47	Device Level Ring Object	→ **Device Level Ring Object (object class: 0x47)** (→ p. 133)
72	48	Quality of Service	→ **Quality of Service (object class: 0x48)** (→ p. 134)
128	80	IO-Link Requests	→ **IO-Link requests (object class: 0x80)** (→ p. 135)
245	F5	TCP/IP Object	→ **TCP/IP object (object class: 0xF5)** (→ p. 145)
246	F6	Ethernet Link Object	→ **Ethernet Link Object (object class: 0xF6)** (→ p. 147)

Identity Object (object class: 0x01)

34340

The Identity Object contains the general information about the device.

Class attributes

34310

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

Instance attributes

34339

Attr. ID	Access	Name	Data type	Description	Preset
1	Get	Vendor ID	UINT	Manufacturer ID	322
2	Get	Device type	UINT	Type of unit	12
3	Get	Product code	UINT	Identification of a particular product of a vendor	1920
4	Get	Revision	STRUCT	Revision of the article that is represented by the Identity Object	1.1
		▪ Major revision	USINT	Main revision (1...127)	1
		▪ Minor revision	USINT	Side revision (3 digits, if necessary with zeros in the beginning)	1
5	Get	Status	WORD	Status of the device	
6	Get	Serial number	UDINT	Serial number of the device	
7	Get	Product Name	SHORT STRING	Readable device designation (max. 32 ASCII characters)	IO-Link Master CL EIP 8P IP20
8	Get	State	USINT	Current status of the device (according to status transition diagram)	
				0 Nonexistent	
				1 Device Self Testing	
				2 Standby	
				3 Operational	
				4 Major Recoverable Fault	
				5 Major Unrecoverable Fault	
				6...254 Reserved	
				255 Default for Get_Attributes_All service	
9	Get	Configuration Consistency Value	UINT	The content shows the configuration of the device	0

Supported services

34377

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

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

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

Supported reset types:

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

Message Router Object (object class: 0x02)

34390

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

Class attributes

34320

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

Instance attributes

34402

The object has no instance attributes.

Supported services

34374

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

Assembly Object (object class: 0x04)

34332

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

Class attributes

34309

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

Instances

34403

Attr. ID	Access	Name	Data type	Description
100	Get	Input assembly	STRUCT	Cyclic input data (→ **Input assembly (Instance 100): I/O data + acyclic data + diagnosis data** (→ p. 103))
101	Get	Input assembly	STRUCT	Cyclic input data (→ **Input Assembly (Instance 101): I/O data + acyclic data** (→ p. 104))
102	Get	Input assembly	STRUCT	Cyclic input data (→ **Input Assembly (Instance 102): I/O data** (→ p. 105))
150	Get, Set	Output assembly	STRUCT	Cyclic output data (→ **Output assembly (Instance 150): I/O data + acyclic data** (→ p. 110))
151	Get, Set	Output assembly	STRUCT	Cyclic output data (→ **Output Assembly (Instance 151): I/O data** (→ p. 111))
199	Set	Configuration assembly	STRUCT	Configuration data (→ **Configuration Assembly (Instance 199)** (→ p. 99))

Instance attributes

61121

Attr. ID	Access	Name	Data type	Description	Preset:
1		Number of member	UINT	Manufacturer ID	-
2	Get	Member	UINT	Memeber List	-
3	Get, Set	Data	UINT	Image of the process data	-
4	Get	Size	UINT	Size of the process data (in bytes)	-
300		Member data list	UINT	Data of the Assembly members	-
301		Parameter	UINT	Assembly parameters	-
302		Status	UINT	Status of the assembly	-

Supported services

34376

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

Connection Manager Object (object class: 0x06)

34367

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

Class attributes

34319

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

Instance attributes

34402

The object has no instance attributes.

Supported services

34375

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

Device Level Ring Object (object class: 0x47)

34345

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

Class attributes

34313

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

Instance attributes

34327

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

Supported services

34409

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

Quality of Service (object class: 0x48)

34371

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

Class attributes

34307

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

Instance attributes

34328

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

Supported services

34406

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

IO-Link requests (object class: 0x80)

34412

The manufacturer-specific object "IO-Link Requests" enables read and write access to the IO-Link objects of an IO-Link device connected to a AL1920 via ISDU (Index Service Data Unit). The object projects the mechanisms of the CIP addressing on the IO-Link protocol.

Class attributes

34308

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

Instance attributes

34399

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

Supported services

34378

Service code		Name	Class	Attribute	Description
dec	hex				
75	4B	→ **Read_ISDU** (→ p. 136)	no	yes	Read ISDU
76	4C	→ **Write_ISDU** (→ p. 139)	no	yes	Read ISDU
77	4D	→ **Write Failsafe Pattern** (→ p. 142)	no	yes	Write failsafe values of IO-Link port

Read_ISDU

34323

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

Request

34337

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

CIP format	Data type	MSG Config	IO-Link mapping
CIP Class ID	UINT	0x80	IO-Link acyclic access
CIP Instance ID	UINT	0x01	IO-Link master
CIP Attributes	USINT	0x01...0x08	Port number
CIP Service code ID	USINT	0x4B	Request "Read_ISDU"
CIP User specific service data	UINT	0x0000...0xFFFF	IO-Link ISDU object index
	USINT	0x00...0xFF	IO-Link ISDU object sub-index

Response

34326

- **Positive response**

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

CIP format	Data type	MSG Config	IO-Link mapping
CIP Class ID	UINT	0x80	IO-Link acyclic access
CIP Instance ID	UINT	0x01	IO-Link master
CIP Attributes	USINT	0x01...0x08	Port number
CIP Service Code ID	USINT	0x4C	Response "Read_ISDU"
CIP Error Code	USINT	0x00	--
CIP Extended Error Code	USINT	0x00	--
CIP User Specific Service Data	USINT	0x00...0xFF	Data (byte 0)
	USINT	0x00...0xFF	Data (byte 1)
	...		...
	USINT	0x00...0xFF	Data (byte n)

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

- **Negative response**

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

CIP format	Data type	MSG Config	IO-Link mapping
CIP Class ID	UINT	0x80	IO-Link acyclic access
CIP Instance ID	UINT	0x01	IO-Link master
CIP Attributes	USINT	0x01...0x08	Port number
CIP Service Code ID	USINT	0x4B	Response "Read_ISDU"
CIP Error Code	USINT	<> 0x00	Error code: see table below
CIP Extended Error Code	USINT	0x00	Extended error code
CIP User Specific Service Data	USINT	<> 0x00	IO-Link Error Code: → **Error codes** (→ p. 117) (only if CIP Error Code = 0x1E)
	USINT	<> 0x00	IO-Link Additional Code: → **Additional Codes** (→ p. 117) (only if CIP Error Code = 0x1E)

CIP Error Code:

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

Example: reading the parameter value of anIO-Link device

34350

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

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

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

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

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

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

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

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

Write_ISDU

34385

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

Request

34387

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

CIP format	Data type	MSG Config	IO-Link mapping
CIP Class ID	UINT	0x80	IO-Link acyclic access
CIP Instance ID	UINT	0x1	IO-Link master
CIP Attribute	USINT	0x01...0x08	Port number
CIP Service Code ID	USINT	0x4C	Request "Write_ISDU"
CIP User Specific Service Data	UINT	0x0000...0xFFFF	IO-Link ISDU object index
	USINT	0x00...0xFF	IO-Link ISDU object sub-index
	USINT	0x00...0xFF	IO-Link ISDU data (byte 0)
	USINT	0x00...0xFF	IO-Link ISDU data (byte 1)

Response

34384

- **Positive response**

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

CIP format	Data type	MSG Config	IO-Link mapping
CIP Class ID	UINT	0x80	IO-Link acyclic access
CIP Instance ID	UINT	0x01	IO-Link master
CIP Attribute	USINT	0x01...0x08	Port number
CIP Service Code ID	USINT	0x4C	Response "Write_ISDU"
CIP Error Code	USINT	0x00	--
CIP Extended Error Code	USINT	0x00	--

- **Negative response**

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

CIP format	Data type	MSG Config	IO-Link mapping
CIP Class ID	UINT	0x80	IO-Link acyclic access
CIP Instance ID	UINT	0x01	IO-Link master
CIP Attributes	USINT	0x01...0x08	Port number
CIP Service Code ID	USINT	0x4C	Response "Write_ISDU"
CIP Error Code	USINT	<> 0x00	Error code. see table below
CIP Extended Error Code	USINT	0x00	Extended error code
CIP User Specific Service Data	USINT	<> 0x00	IO-Link Error Code: → **Error codes** (→ p. 117) (only if CIP Error Code = 0x1E)
	USINT	<> 0x00	IO-Link Additional Code: → **Additional Codes** (→ p. 117) (only if CIP Error Code = 0x1E)

CIP Error Code:

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

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

34355

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

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

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

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

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

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

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

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

Write Failsafe Pattern

54597

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

Request

54694

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

CIP Format	Data type	MSG Config	Description
CIP Class ID	UINT	0x80	IO-Link acyclic access
CIP Instance ID	UINT	0x01	IO-Link master
CIP Attribute	USINT	0x01...0x08	Port number
CIP Service Code ID	USINT	0x4D	Request "Write Failsafe Pattern"
CIP User Specific Service Data	USINT	0x00 = No Fail-safe 0x01 = Fail-safe Reset Value 0x02 = Fail-safe Old Value 0x03 = Fail-safe with Pattern	Failsafe Mode
	USINT	0x00...0xFF	Failsafe Pattern (MSB)
	USINT	0x00...0xFF	Failsafe Pattern (LSB)

Response

54695

- **Positive response**

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

CIP Format	Data type	MSG Config	Description
CIP Class ID	UINT	0x80	IO-Link acyclic access
CIP Instance ID	UINT	0x01	IO-Link master
CIP Attribute	USINT	0x01...0x08	Port number
CIP Service Code ID	USINT	0x4D	Response "Write Failsafe Pattern"
CIP Error Code	USINT	0x00	--
CIP Extended Error Code	USINT	0x00	--

- **Negative response**

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

CIP-Format	Data type	MSG Config	Description
CIP Class ID	UINT	0x80	IO-Link acyclic access
CIP Instance ID	UINT	0x01	IO-Link master
CIP Attribute	USINT	0x01...0x08	Port number
CIP Service Code ID	USINT	0x4D	Response "Write Failsafe Pattern"
CIP Error Code	USINT	<> 0x00	Error code: see below
CIP Extended Error Code	USINT	<> 0x00	Extended error code

CIP Error Code:

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

Example: Write fail-safe value

54696

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

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

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

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

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

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

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

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

TCP/IP object (object class: 0xF5)

34388

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

Class attributes

34311

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

Instance attributes

34330

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

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

Supported services

34416

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

Ethernet Link Object (object class: 0xF6)

34354

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

Class attributes

34312

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

Instance attributes

34333

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

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

Supported services

34414

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

14.3 ifm IoT Core

Content

33803

14.3.1 Overview: IoT profile

Content

34054

Profile: blob

52264

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

Profile: deviceinfo

34207

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

Profile: devicetag

34206

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

Profile: iolinkdevice_full

52265

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

Profile: iolinkmaster

34205

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

Profile: mqttCmdChannel

61186

Element (identifier)	Properties	Mandatory	Comment
mqttCmdChannel	▪ type = structure ▪ profile = commChannel		Profile of the MQTT command channel
../type	▪ type = data ▪ data type = STRING	mandatory	Protocol type of the interface
../status	▪ type = data ▪ data type = STRING	mandatory	Status of the MQTT command channel (possible values: init, running, stopped, error)
../mqttCmdChannelSetup	type = profile		Sub-profile: **Profile: mqttCmdChannelSetup** (→ p. 153)

Profile: mqttCmdChannelSetup

61187

Element (identifier)	Properties	Mandatory	Comment
mqttCmdChannelSetup	▪ type = structure ▪ profile = mqttCmdChannelSetup		Settings of the MQTT command channel
../brokerIP	▪ type = datat ▪ data type = STRING	optional	
../brokerPort	▪ type = data ▪ data type = STRING	optional	
../cmdTopic	▪ type = data ▪ data type = STRING	optional	
../defaultReplyTopic	▪ type = data ▪ data type = STRING	optional	

Profile: mqttConnection

61188

Element (identifier)	Properties	Mandatory	Comment
mqttConnection	▪ type = structure ▪ profile = commInterface		MQTT connection in the IoT Core
../type	▪ type = data ▪ data type = STRING	mandatory	Protocol type of the interface
../status	▪ type = data ▪ data type = STRING	mandatory	global status of the MQTT (possible values: init, running, stopped, error)
../mqttSetup	type = profile		Sub-profile: **Profile: mqttSetup** (→ p. 154)
../mqttCmdChannel	type = profile		Sub-profile: **Profile: mqttCmdChannel** (→ p. 153)

Profile: mqttSetup

61189

Element (identifier)	Properties	Mandatory	Comment
mqttSetup	▪ type = structure ▪ profile = mqttSetup		Settings of the MQTT command channel
../QoS	▪ type = data ▪ data type = Number	mandatory	Quality of Service of the MQTT connection
../version	▪ type = data ▪ data type = STRING	mandatory	

Profile: network

52266

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

Profile: parameter

34215

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

Profile: processdata

34225

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

Profile: runcontrol

61190

Element (identifier)	Properties	Mandatory	Comment
runcontrol	▪ type = profile ▪ profile = runcontrol		Control of the MQTT command channel
../start	type = service	mandatory	**Service: start** (→ p. 168)
../stop	type = service	mandatory	**Service: stop** (→ p. 169)
../reset	type = service	mandatory	**Service: Reset** (→ p. 166)

Profile: service

34224

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

Profile: software

34223

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

Profile: software/uploadedablesoftware

52267

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

Profile: Timer

34226

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

14.3.2 Overview: IoT types

34055

The ifm IoT Core uses the following element types:

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

14.3.3 Overview: IoT services

Content

34056

Service: factoryreset

34184

Name: factoryreset

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

Request data (field "data"): none

Response data (field "data"): none

Example:

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

Service: getblobdata

52345

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

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

Return data (field "data"):

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

Service: getdata

34183

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

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

Example:

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

Service: getdatamulti

34174

Name: getdatamulti

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

Request data (field "data"):

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

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

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

Service: getelementinfo

52269

Name: getelementinfo

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

Applicable to: Objects of the type `device`

Request data (field "data"):

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

Return data (field "data"):

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

Servicet: getidentity

54690

Name: getidentity
Description: The service reads the device information of the AL1920 and issues it.
Request data ("data" field): none
Return data ("data" field):

Data field	Required field	Data type	Description	
iot		Device	Device description as JSON object	
iot.name	mandatory	STRING		
iot.uid	optional	STRING		
iot.version	mandatory	STRING		
iot.catalogue	optional	ARRAY OF OBJECTS		
iot.deviceclass	optional	ARRAY OF STRING		
iot.serverlist	optional	ARRAY OF OBJECTS		
device	optional		AL1920	
device.serialnumber	optional		Serial number	
device.hwrevision	optional		Hardware version	
device.swrevision	optional		Software version	
device.custom	optional			
Security	optional		Security options	
security.securitymode	optional	ENUM	shows if the security mode is activated	
security.authscheme	optional	ENUM	shows the active authentication scheme	
security.ispasswordset	optional	BOOL	shows whether a password has been set	
security.activeconnection	optional	ENUM	shows the currently used communication interface	
			▪ tcp_if	unencrypted http connection at the IoT interface, port 80
			▪ tls_if	encrypted https connection at the IoT interface, port 443
			▪ fb_if	unencrypted http connection at the fieldbus interface, port 80

Service: getsubscriberlist

61191

Name: getsubscriberlist

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

Request data ("data" field): none

Return data ("data" field): Array with the following data

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

Example:

- **Request object:**

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

- **Return object:**

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

Service: getsubscriptioninfo

61192

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

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

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

Request data ("data" field):

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

Return data ("data" field):

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

Example:

- **Request object:**

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

- **Return object:**

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

Service: gettree

61193

Name: gettree

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

Request data ("data" field):

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

Return data ("data" field):

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

Examples:

- output the complete device description

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

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

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

Service: install

52343

Name: install

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

Applicable to: container

Request data (data): none

Return data (data): none

Service: iolreadacyclic

34178

Name: iolreadacyclic

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

Request data (field "data"):

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

Response data (field "data"):

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

Service: iolwriteacyclic

34177

Name: iolwriteacyclic

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

Request data (field "data"):

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

Response data (field "data"): none

Service: querytree

61194

Name: querytree

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

Return data ("data" field):

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

Return ("data" field):

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

Service: reboot

34176

Name: reboot

Description: The service reboots the device.

Request data (field "data"): none

Return data (field "data"): none

Example:

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

Service: Reset

61195

Name: Reset

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

Request data ("data" field): none

Return data ("data" field): none

Example:

```
{
"code":"request",
"cid":4711,
"adr":"/connections/mqttConnection/MQTTSetup/mqttCmdChannel/status/reset"
}
```

Service: setblock

34186

Name: setblock

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

Request data (field "data"):

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

Response data (field "data"): none

Example:

Request:

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

Response:

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

Service: setdata

34195

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

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

Return data ("data" field): none

Example:

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

Service: signal

33819

Name: signal
Description: The service starts the flashing of the status LEDs of the AL1920.
Request data (field "data"): none
Return data (field "data"): none

Example:

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

Service: start

61196

Name: start
Description: The service starts a connection.
Request data ("data" field): none
Return data ("data" field): none

Example:

```
{
"code":"request",
"cid":4711,
```

```
"adr":"/connections/mqttConnection/MQTTSetup/mqttCmdChannel/status/start"
}
```

Service: start_stream_set

52342

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

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

Return data (data): none

Service: stop

61197

Name: stop
Description: The servicestops a connection.
Request data ("data" field): none
Return data ("data" field): none

Example:

```
{
"code":"request",
"cid":4711,
"adr":"/connections/mqttConnection/MQTTSetup/mqttCmdChannel/status/stop"
}
```

Service: stream_set

52341

Name: stream_set
Description: The service transfers a data segment.
Applicable to: Objects of type data
Request data (data):

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

Return data (data): none

Service: subscribe

61198

Name: subscribe

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

CSV formatted notifications can only be transmitted using the TCP protocol via an activated and configured MQTT channel.

Request data ("data" field):

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

Return data ("data" field): none

Notification: JSON

```
{
"code":"event",
"cid":4711,
"adr":"",
"data":{
"eventno":"EventNo",
"srcurl":"SrcURL",
"payload":{
"eventurl":{"code":EventStatus,"data":EventData},
"datapointurl_1":{"code":DataStatus_1,"data":DataValue_1},
"datapointurl_2":{"code":DataStatus_2,"data":DataValue_2},
...}}
}
```

Notification: CSV

```
SrcURL,EventNo,EventStatus,EventData,DataStatus_1,DataValue_1,DataStatus_2,DataVal
ue_2,...
```

- `SrcURL`: Source of the event (data point on which subscribe command was listed)
- `EventNo`: Event number

- `EventStatus`: Status code of the event
- `EventData`: Event data
- DataStatus_1: Status code of the 1st element in list datatosend
- `DataValue_1`: Value of the 1st element in list datatosend
- `DataStatus_2`: Status code of the 2nd element in list datatosend
- `DataValue_2`: Value of the 2nd element in list datatosend
- ...

Service: unsubscribe

34197

Name: unsubscribe

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

Request data (field "data"):

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

Response data (field "data"): none

Service: validation_useconnecteddevice

52340

Name: validation_connecteddevice

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

Applicable to: Objects of type `stucture`

Request data (data): none

Return data (data): none

15 Index

A

C

D

E

F

G

I

L

M

N

O

P

Q

R

S

T

U

V

W