Manual

Modbus TCP

LioN-X IO-Link Master Multiprotocol: 0980 XSL 3912-121-007D-00F-BD (8 × IO-Link Class A)

Contents

1 About this manual

1.1 General information

Download the latest version of this document at:
http://doc.beldensolutions.com

Read the assembly and operating instructions on the following pages carefully before starting up the modules. Keep this information where it is accessible to all users.

The texts, figures, diagrams, and examples used in this document are exclusively used to explain how to operate and apply the modules.

Please contact us if you have any detailed questions on installing and starting up the devices.

Belden Deutschland GmbH
Belden
Im Gewerbepark 2
D-58579 Schalksmühle
Germany

https://my.belden.com
https://www.belden.com
https://catalog.belden.com

1.2 Explanation of symbols

1.2.1 Use of danger information

Danger information is denoted as follows:

Danger: Means that death, serious physical injury or substantial damage to property will occur if the required safety measures are not taken.

Warning: Means that death, serious physical injury or substantial damage to property can occur if the required safety measures are not taken.

Caution: Means that minor physical injury or damage to property can occur if the required safety measures are not taken.

1.2.2 Use of general information

General information is denoted as follows:

Attention: Contains important information on the product, on how to manage the product, or on the respective section of the documentation to which your special attention is being drawn.

1.3 Version information

Version	Created	Changes
1.0	07/2026	

Table 1: Overview of manual revisions

2 Safety instructions

2.1 Intended use

The products described in this manual are decentralized IO-Link Masters on an Industrial Ethernet Network.

We adhere to all safety standards when developing, producing, testing, and documenting our products. When you adhere to the handling specifications and safety instructions described for the configuration, assembly, and correct operation, there should not normally be any risks for people or equipment.

The modules fulfill the requirements of the EMC guidelines (2014/30/EU) and the low voltage guideline (2014/35/EU).

The IO-Link Masters are designed to be used in the industrial sector. The industrial environment is distinguished by the fact that the consumer is not connected directly to the public low voltage network. Additional measures are required for use in residential areas or in business and commercial sectors.

Attention: This equipment may cause radio interference in residential areas. In this case the operator may be requested to carry out appropriate measures.

The proper and safe operation of this product depends on proper transportation, storage, assembly, and installation, and careful operation.

A completely assembled device housing is required for the proper operation of the IO-Link Masters. Only connect devices that fulfill the requirements of EN 61558-2-4 and EN 61558-2-6 to the IO-Link Masters.

During the configuration, installation, start-up, maintenance, and testing of the devices, adhere to the safety and accident-prevention guidelines for the specific application.

Only install cables and accessories that fulfill the requirements and regulations for safety, electromagnetic compatibility, and, where applicable, telecommunication end devices, as well as the specification information. Information on which cables and accessories are permitted for the installation

can be obtained from Belden Deutschland GmbH – Belden or is contained in this manual.

2.2 Qualified personnel

The configuration, installation, start-up, maintenance, and testing of the devices may only be performed by a qualified electrician who is familiar with the safety standards of the automation technology.

The personnel requirements are based on the requirement profiles described by ZVEI, VDMA, or equivalent organizations.

Only electricians who are familiar with the content of all provided device documentation are authorized to install and maintain the devices described. These are persons who

- based on their technical training, knowledge, and experience, and their knowledge of the pertinent standards, can evaluate the work to be carried out and identify any potential risks or
- based on working for several years in a related sector, have the same level of knowledge as they would have from the relevant technical training.

Only Belden Deutschland GmbH – Belden is permitted to make changes to the hardware or software of the products that go beyond the scope of this manual.

Warning: Making unqualified changes to the hardware or software, or non-adherence to the warning information contained in this document, can result in serious personal injury or damage to equipment.

Attention: Belden Deutschland GmbH accepts no liability for any damage caused by unqualified personnel or improper use. This automatically voids the warranty.

3 Designations and synonyms

AI	Analog Input
AO	Analog Output
AOI	Add-On Instruction
API	Application Programming Interface
BF	Bus Fault LED
Big Endian	Data format with High-B on first place (PROFINET and IO-Link)
BUI	Back-Up Inconsistency (EIP diagnostics)
CC	CC-Link IE Field
C/Q	I/O port pin 4 mode, IO-Link communication/switching signal
Ch. A	Channel A (Pin 4) of I/O port
Ch. B	Channel B (Pin 2) of I/O port
CIP	Common Industrial Protocol (media independent protocol)
CIP Safety™	Common Industrial Protocol for Safety applications, CIP Safety™ is a registered trademark of ODVA
Class A	IO-Link port specification (Class A)
Class B	IO-Link port specification (Class B)
CoAP	Constrained Application Protocol
CSP+	Control & Communication System Profile Plus
DAT	Device Acknowledgement Time
DCP	Discovery and Configuration Protocol
DevCom	Device Comunicating (EIP diagnostics)
DevErr	Device Error (EIP diagnostics)
DI	Digital Input
DIA	Diagnostic LED
DO	Digital Output
DIO	Digital Input/Output

DTO	Device Temperature Overrun (EIP diagnostics)
DUT	Device under test
EIP	EtherNet/IP™ is a registered trademark of ODVA
ERP	Enterprise Resource Planning system
ETH	ETHERNET
FE	Functional Earth
FME	Force Mode Enabled (EIP diagnostics)
FS	Functional Safety
FSU	Fast Start-Up
GSDML	General Station Description Markup Language
High-B	High-Byte
HTTPS	Hyper Text Transfer Protocol Secure
ICE	IO-Link port COM Error (EIP diagnostics)
ICT	Invalid Cycle Time (EIP diagnostics)
IE	IO-Link port Error (EIP diagnostics)
IN	IO-Link port Notification (EIP diagnostics)
IW	IO-Link port Warning (EIP diagnostics)
IIoT	Industrial Internet of Things
ILE	Input process data Length Error (EIP diagnostics)
IME	Internal Module Error (EIP diagnostics)
I/O	Input / Output
I/O port	X1 .. X8
I/O port pin 2	Channel B of I/O ports
I/O port pin 4 (C/Q)	Channel A of I/O ports
IODD	I/O Device Description
IOL or IO-L	IO-Link
I/Q	I/O port pin 2 mode, Digital Input/switching signal
ISDU	Indexed Service Data Unit
IVE	IO-Link port Validation Error (EIP diagnostics)
I&M	Identification & Maintenance

JSON	JavaScript Object Notation (platform independent data format)
L+	I/O port pin 1, sensor power supply
LioN-X 60	LioN-X variants with a width of 60mm
Little Endian	Data format with Low-B on first place (EtherNet/IP)
LLDP	Link Layer Discovery Protocol
Low-B	Low-Byte
LSB	Least Significant Bit
LVA	Low Voltage Actuator Supply (EIP diagnostics)
LVS	Low Voltage System/Sensor Supply (EIP diagnostics)
MIB	Management Information Base
MP	Multi-protocol: PROFINET + EtherNet/IP + EtherCAT® + Modbus TCP (+ CC-Link IE Field Basic)
MQTT	Message Queuing Telemetry Transport (open networking protocol)
MSB	Most Significant Bit
M12	Metric thread according to DIN 13-1 with 12 mm diameter
n.a.	not applicable
n.c.	no connection / not connected
NTP	Network Time Protocol
OFDT	One Fault Delay Time
OLE	Output process data Length Error (EIP diagnostics)
OPC UA	Open Platform Communications Unified Architecture (platform independent, service-oriented architecture)
PFH	Probability of dangerous Failure per Hour [h -1]
PD	Process Data
PDCT	Port and Device Configuration Tool
PLC	Programmable Logic Controller
PN	PROFINET
PWR	Power
Qualifier	Validity on a process value. Valid = "1"
REST	REpresentational State Transfer
RFC	Request for Comments

RO	Read Only
RPI	Requested Packet Interval
RTD	Resistance Temperature Detector
RW	Read/Write
RWr	Word data input as seen from the master station (CC-Link)
RWw	Word data output as seen from the master station (CC-Link)
RX	Bit data input as seen from the master station (CC-Link)
RY	Bit data output as seen from the master station (CC-Link)
SCA	Short Circuit Actuator/U_L/U_{AUX} (EIP diagnostics)
SCS	Short Circuit Sensor (EIP diagnostics)
SFRT	Safety Function Response Time
SIO mode	Standard Input Output mode
SLMP	Seamless Message Protocol
SNMP	Simple Network Management Protocol
SP	Single Protocol (PROFINET, EtherNet/IP, EtherCAT®, Modbus TCP or CC-Link IE Field Basic)
SPE	Startup Parameterization Error (EIP diagnostics)
T-B	Test Channel B
T-A	Test Channel A
U_{AUX}	$U_{Auxiliary}$, supply voltage for the load circuit (Actuator supply on Class B ports of Class A/B IO-Link Master)
UDP	User Datagram Protocol
UDT	User-Defined Data Types
UINT8	Byte in PLC (IB, QB)
UINT16	Unsigned integer with 16 bits or word in PLC (IW, QW)
U_L	U_{Load}, supply voltage for the load circuit (Actuator supply on Class A IO-Link-Master)
UL	Underwriters Laboratories Inc. (certification company)
UTC	Coordinated Universal Time (Temps Universel Coordonné)
WCDT	Worst Case Delay Time

Table 2: Designations and synonyms

4 System description

The LioN-Modules function as the interface in an industrial Ethernet system: A central controller on the management level is able to communicate with the decentralized sensors and actuators on the field level. The line or ring topologies for which LioN modules can be used ensure not only reliable data communication but also significantly reduce the number of cables required and thus also the costs for installation and maintenance. They additionally enable easy and quick extension.

4.1 About LioN-X

The LioN-X device variants convert standard input, standard output or IO-Link signals from sensors & actuators into an industrial Ethernet protocol (PROFINET, EtherNet/IP, EtherCAT®, Modbus TCP, CC-Link IE Field Basic) and/or into a cloud protocol (REST API, CoAP, OPC UA, MQTT). For the first time, there is now Syslog on board. The robust 8 port housing design allows the use even in harsh environments where e.g. weld field immunity, high temperature ranges or protection class IP67 & IP69K are needed.

Use all benefits of the Belden product solution by additionally downloading the configuration tool *LioN-Management Suite* from https://www.belden.com to enable e.g. a fast and easy parameterization of the connected IO-Link devices via the embedded IODD interpreter.

4.2 Device variants

The following variants are available in the LioN-X Multi-protocol family:

Article number	Product designation	Description	I/O port functionality
935720001	0980 XSL 3912-121-007D-00F-BD	LioN-X M12-60 mm, IO-Link Master Multi-protocol (PN, EIP, EC, MB) Security	8 x IO-Link Class A

Table 3: Overview of LioN-X Multi-protocol variants

4.3 I/O port overview

The following tables show the main I/O port differences of the LioN-X IO-Link-Master family. Pin 4 and Pin 2 of the I/O ports can be configured partly to IO-Link, Digital Input or Digital Output.

LioN-X Class A IO-Link ports

Device variant	Port	Pin 1 U_S	Pin 4 / Ch. A (C/Q)				Pin 2 / Ch. B (I/Q)	
0980 XSL 3x12...	**Info:**	–	Class A	Type 1	Supply by U_S [1)]	Supply by U_L [2)]	Type 1	Supply by U_L [2)]
	X8:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)
	X7:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)
	X6:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)
	X5:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)
	X4:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)
	X3:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)
	X2:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)
	X1:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)

Table 4: Port configuration of 0980 XSL 3x12... variants

[1)] DO switch mode configured as "Push-Pull" (description in the configuration chapters).

[2)] DO switch mode configured as "High-Side" (description in the configuration chapters).

5 Overview of product features

5.1 Modbus TCP product features

Modbus Mode

Features MODBUS server via a standard TCP network. The number of allowed operations for holding registers is dependent on the device configuration. The device supports 4 up to 8 TCP sockets for communication.

Data connection

The connection option provided by LioN-X is the widely-used M12 connector with D-coding for the Modbus TCP network.

The connectors are also color-coded to prevent the ports from being mixed up.

Data transmission rates

Featuring a transmission rate of up to 100 MBit/s, the Modbus TCP devices can handle both fast transmission of I/O data and transmission of larger volumes of data.

Diagnostic data

The devices support diagnosis flags and extended diagnostic data that can be appended to the I/O data.

5.2 I/O port features

IO-Link specification

LioN-X fulfills IO-Link specification v1.1.3.

8 x IO-Link Master ports

Warning: If modules with electric isolation and modules without electric isolation are used within the same system, the electric isolation of all connected modules is annulled.

IO-Link port connections

The IO-Link port connection option provided by the module series is the 5-pin M12 connector.

Validation & Backup

The Validation & Backup function checks if the right device is connected and stores / monitors the parameters of the IO-Link Device. The function thus gives you an easy option for replacing the IO-Link Device.

This is possible as of IO-Link specification V1.1 and only if the IO-Link Device **and** the IO-Link Master support the function.

LED

You can see the status of a port by the color of the matching LEDs and their flash pattern. For details on the meanings of the LED colors, please see section LEDs on page 165.

5.3 Integrated Web server

Network parameter display

Get an overview of network parameters such as the IP address, subnet mask and gateway.

Displaying diagnostics

View diagnostics via the integrated Web server.

User management

Use the integrated Web server for convenient management of all users.

IO-Link Device parameters

Reading and writing of IO-Link Device parameters is supported. The system command Store parameters is needed after parameter writing, to take over the changed parameter into the IO-Link Master backup memory when enabled.

5.4 Security features

Firmware signature

The official firmware update packages contain a signature which helps prevent the system against manipulated firmware updates.

Syslog

The LioN-X multi-protocol variants support the traceability of messages centrally managed and logged via Syslog.

User manager

The Web server provides a user manager to help protect the Web interface against unauthorized access. You can manage the users by groups with different access levels “Admin” or “Write”.

Default user settings:

User: admin

Password: private

Attention: Change the default settings to help protect the device against unauthorized access.

5.5 Other features

Interface protection

The devices have reverse polarity, short-circuit and overload protection for each interface.

For more details, see section Port assignments on page 27.

Failsafe

The devices support a failsafe function. This allows you to define the behavior of every single channel configured as an output in the case of invalid PLC data (e.g. PLC in STOP) or of lost PLC communication.

Industrial Internet of Things

LioN-X is industry 4.0 ready and supports the integration in IIoT networks via REST API and the IIoT-relevant protocols MQTT, OPC UA and CoAP.

Color-coded connectors

The colored connectors help you avoid confusion in your cabling.

IP protection classes: IP65 / IP67 / IP69K

The IP protection class describes environmental influences that the devices can be exposed to without risk and without suffering damage or causing a risk for the user.

The whole LioN-X family offers IP65, IP67 and IP69K.

6 Assembly and wiring

6.1 General information

Mount the device on a flat surface using 2 screws (M4x 25/30). The torque required here is 1 Nm. Use washers for all fastening methods as per DIN 125.

Attention: The devices have a ground connection with an M4 thread for the conduction of interference currents and the EMC immunity. This is labeled with the symbol for the ground and the designation "FE".

Attention: Use a low-impedance connection to connect the device to the reference ground. When using a grounded mounting surface, you can make the connection directly via the fixing screws.

Attention: If the mounting surface is ground-free, use a ground strap or a suitable FE line (FE = Functional Earth). Use an M4 screw to connect the ground strap or the FE line to the ground point and if possible put a washer and a toothed washer below the fixing screw.

6.2 Outer dimensions

6.2.1 LioN-X multi-protocol

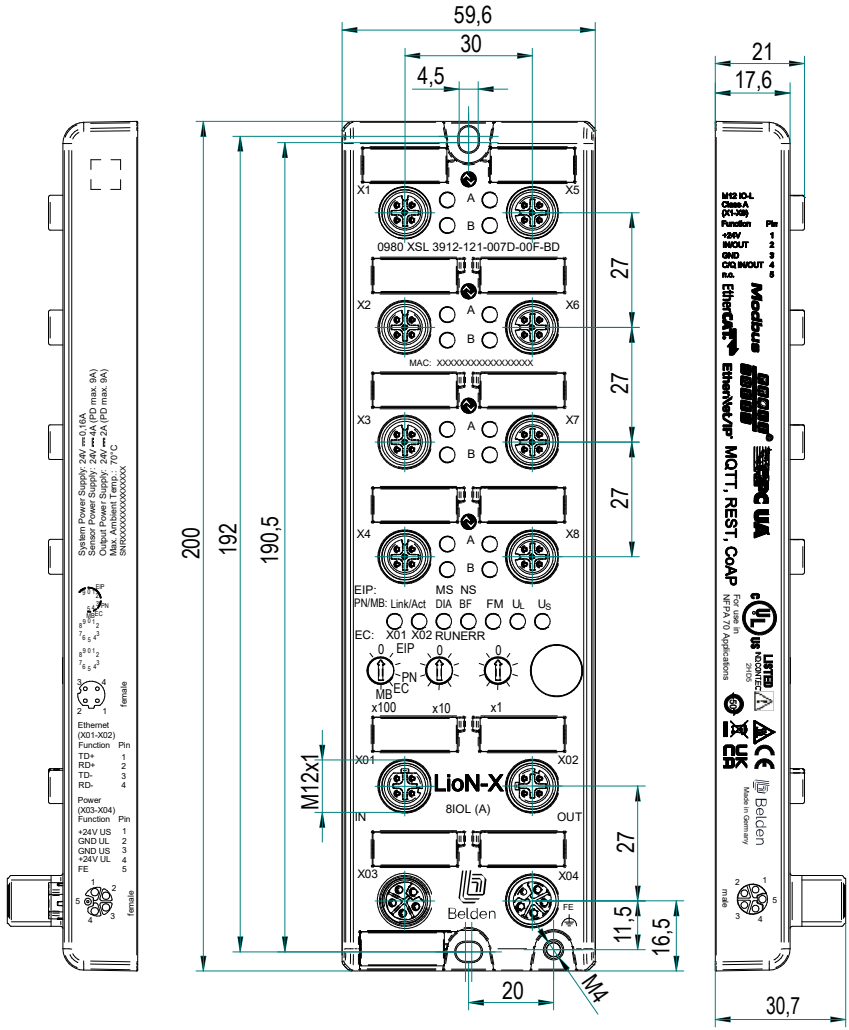


Figure 1: 0980 XSL 3912-121-007D-00F-BD

6.2.2 Notifications

Attention:

For **UL applications**, be sure to use a UL-certified cable with a suitable evaluation to connect the devices (CYJV or PVVA). To program the control, please refer to the OEM information, and only use suitable accessories.

Only approved for interior use. Please note the maximum elevation of 2000 meters. Approved up to a maximum soiling level of 2.

Warning: Terminals, housings field-wired terminal boxes or components can exceed temperatures of +60 °C (140 °F).

Warning: For **UL applications** at a maximum ambient temperature of +70 °C (158 °F):

Use temperature-resistant cables with heat resistance up to at least +125 °C (257 °F) for all device variants.

Warning: Observe the following maximum output power for the sensor supply of Class A devices:

Max. 4.0 A per port; for **UL applications** max. 5 A for every port pair X1/X2, X3/X4, X5/X6, X7/X8; max. 9.0 A in total (with derating) for the whole port group X1 .. X8.

6.3 Port assignments

All the contact arrangements shown in this chapter show the frontal view of the connection area for the connectors.

6.3.1 Ethernet ports, M12 socket, 4-pin, D-coded

Color coding: green

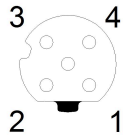


Figure 2: Schematic drawing, ports X01, X02

Port	Pin	Signal	Function
Ethernet Ports X01, X02	1	TD+	Transmit data plus
	2	RD+	Receive data plus
	3	TD-	Transmit data minus
	4	RD-	Receive data minus

Table 5: Assignment of ports X01, X02

Caution: Risk of destruction! Never connect the power supply to the data cables.

6.3.2 Power supply with M12 power L-coded

Color coding: gray

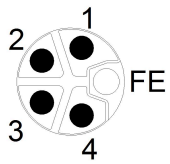


Figure 3: Schematic diagram of the M12 L-coding (connector X03 for Power In)

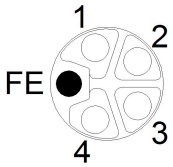


Figure 4: Schematic diagram of the M12 L-coding (socket X04 for Power Out)

6.3.2.1 IO-Link Master with Class A ports

Power supply	Pin	Signal	Function
	1	U_S (+24 V)	Sensor/system power supply
	2	GND_U_L	Ground/reference potential U_L
	3	GND_U_S	Ground/reference potential U_S[1]
	4	U_L (+24 V)	Load supply (**NOT** electrically isolated to U_S internally in device)
	5	FE	Functional ground

Table 6: Power supply with M12-Power Class A

Attention: Only use power supply units for the system/sensor and actuator supply that correspond to PELV (Protective Extra Low Voltage) or SELV (Safety Extra Low Voltage). Power supplies according to EN 61558-2-6 (transformers) or EN 60950-1 (switching power supply units) fulfill these requirements.

6.3.3 I/O ports as M12 sockets

Color coding: black

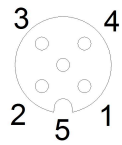


Figure 5: Schematic drawing I/O port as M12 socket IO-Link

[1] U_L and U_S ground connected in device

6.3.3.1 IO-Link ports (Class A)

0980 XSL 3x12...	Pin	Signal	Function
IO-Link Class A, ports X1 .. X8	1	+24 V	power supply +24 V
	2	IN/OUT	Ch. B: Digital input or digital output
	3	GND	Ground/reference potential
	4	C/Q	Ch. A: IO-Link data communication, digital input or digital output
	5	n.c.	not connected

Table 7: I/O ports as IO-Link Class A

Used signal names compared to the IO-Link specification conventions:

Pin	LioN-X	IO-Link specification	Comment
1	+24 V	L+	Supplied by U_S
2	IN/OUT	I/Q	
3	GND	L-	
4	C/Q IN/OUT	C/Q	

7 Starting operation

All listed register addresses are zero-based. This must be taken into account for read and write access by the Modbus master. If the master is one-based, the register addresses must be added by '1'.

The data is assigned to Holding Registers starting from 40001/400001 with the offset '0'.

7.1 Device identification

With any MODBUS client, the server running on LioN-X can be reached to get identification data including vendor name, product code and revision.

The following table represents the device identification data.

Register address[2]	Name	Access	Register length	Size (bytes)	Description
1024	Firmware Version minor	RO (Read Only)	1	2	MBTCP Digital device Firmware Version minor
1025	Firmware Version major	RO	1	2	MBTCP Digital device Firmware Version major
1026 .. 1041	Model Number/ Device name	RO	16	32	Name of device
1042	IP address Source	RO	1	2	0: DHCP 1: Static
1043 .. 1044	IP address	RO	2	4	IP Address of Device
1045 .. 1047	MAC address	RO	3	6	MAC Address of device
1048	Current TCP connections	RO	1	2	Active TCP connections

7.2 MAC addresses

Every device has three unique assigned MAC addresses that cannot be changed by the user. The first assigned MAC address is printed onto the device.

[2] Given numbers are addresses starting from "0".

7.3 Modbus function codes

LioN-X devices support the following Modbus function codes:

- Function code 03 (0x03, Read Holding Registers)
- Function code 04 (0x04, Read Input Registers)
- Function code 06 (0x06, Write Single Register)
- Function code 16 (0x10, Write Multiple Registers)

Write access to holding registers is subject to the device feature set and the configuration of the holding register.

7.4 State on delivery

Modbus TCP parameters in state on delivery or after a factory reset:

Network mode:	DHCP
Static IP address:	192.168.1.XXX (XXX = rotary switch position or last stored data)
Subnet mask:	255.255.255.0
Gateway address	0.0.0.0
Device designations:	0980 XSL 3912-121-007D-00F-BD
Product type:	Modbus TCP server

7.5 Setting the rotary encoding switches

The following LioN-X IO-Link Master variants support multi-protocol application for the protocols EtherNet/IP (E/IP), PROFINET (P), EtherCAT® (EC) and Modbus TCP (MB):

- 0980 XSL 3912-121-007D-00F-BD

Caution: Risk of device damage due to corrupt device memory

Any interruption of the power supply to the device during and after protocol selection can lead to a corrupt device memory.

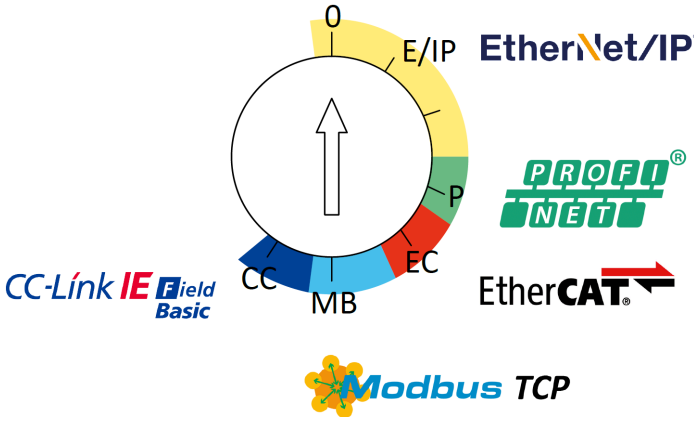
After selecting a protocol followed by a restart of the device, the new protocol is initialized. This can take up to 15 seconds. During this time the device is not usable and the LED indicators are out of function. When the protocol change is complete, the LED indicators return to normal operation and the device can be used again.

- Make sure that the power supply is maintained during the entire process.

The LioN-X multi-protocol variants allow you to select different protocols for communication within an industrial Ethernet system. In this way the IO-Link Masters with multi-protocol function can be integrated into different networks without it being necessary to purchase products specific for each protocol. This technology also gives you the option to use the same IOL-Master in different environments.

Using rotary encoding switches at the lower front of the devices, you can easily and conveniently set both the protocol and the address of the device, if the protocol to be used supports this. Once you have made a protocol selection and started the cyclical communication, the device stores this setting permanently and uses the selected protocol from this point on. To use another supported protocol with this device, perform a factory reset.

The multi-protocol devices have a total of three rotary encoding switches. With the first rotary encoding switch (x100) you set the protocol by using the corresponding switch position. Additionally, x100 is used to set the third last digit of the IP address for EIP.



With the other rotary encoding switches (x10 / x1), you set the last two digits of the IP address when you are using EtherNet/IP, Modbus TCP or CC-Link IE Field Basic.

Protocol	x100	x10	x1
EtherNet/IP	0-2	0-9	0-9
PROFINET	P	–	–
EtherCAT®	EC	–	–
Modbus TCP	MB	0-9	0-9
CC-Link IE Field	CC	0-9	0-9

Table 8: Assignment of the rotary encoding switches for each protocol

The setting you make to select a protocol is described detailed in the protocol-specific sections.

In delivery state no protocol settings are stored in the device. In this case only the desired protocol has to be chosen. To take over a changed rotary encoding switch setting (protocol setting), a power cycle or “Reset” from the Web interface is necessary.

Once you have set the protocol using the rotary encoding switches, the device stores this setting when it starts in cyclic communication. Changing the protocol using the rotary encoding switch is no longer possible after this

point. The device will always start using the stored protocol from that point on. The IP address can be changed depending on the selected protocol.

To change the protocol, carry out a factory reset. In this way you restore the factory settings of the respective device. How you perform the factory reset for your device is described in chapter Factory reset on page 37.

If you position the rotary encoding switch in a manner that is invalid, the device signals this to you with a solid red lighting BF LED.

7.5.1 Modbus TCP

If you decide to use Modbus TCP as a protocol, use the first rotary encoding switch to select the protocol. The second rotary encoding switch (x10) can be used to configure the 10 position of the last octet of the IP address, and the third rotary encoding switch (x1) allows you to configure the 1 position. Values between 0 and 9 can be selected for the second and third switches. The first three octets of the IP address are set by default to 192.168.1.

For example, the rotary encoding switch setting 5(x100), 1(x10) and 0(x1) gives you an IP address of 192.168.1.10 for Modbus TCP. It is only possible to assign IP addresses between 192.168.1.1 and 192.168.1.99 for Modbus TCP via the rotary switches.

Rotary switch setting	Function
500 (network parameters already saved)	The network parameters last saved are used (IP address, subnet mask, gateway address, DHCP on/off, BOOTP on/off).
501 ... 599	The last 2 digits of the saved or preset IP address are overwritten by the setting of the rotary switch.
979	The device performs a reset to the factory settings. The network parameters are also reset to the default values. Communication is not possible in this operation mode.

Table 9: Setting options of the rotary encoding switches for Modbus TCP

7.5.2 Factory reset

A factory reset restores the original factory settings and thus resets the changes and settings you have made up to that point. It also resets the protocol selection. To perform a factory reset, set the first rotary encoding switch (x100) to 9, the second (x10) to 7, and the third (x1) also to 9.

Afterwards perform a power cycle and wait 10 seconds due to internal memory write processes.

During the factory reset, the U_S LED lights up red. As soon as the factory settings have been restored, the U_S LED is flashing green.

	x100	x10	x1
Factory Reset	9	7	9

Follow the steps from section Setting the rotary encoding switches again to select a new protocol.

For performing a factory reset via software configuration, see chapter OPC UA configuration on page 97 and the configuration section.

7.6 Setting network parameters

Use the two right-hand rotary switches (x10 and x1) on the front of the device to set the last octet of the static IP address. Each rotary switch in the range of Modbus TCP is assigned to one decimal digit, so that you can configure a number between **0 – 99**. During startup, the position of the rotary switches is typically read within one time cycle.

The complete IP address, the subnet mask, the gateway address and the network mode (DHCP or BOOTP) can be configured and stored via the Web server or any other available configuration interfaces. New configuration interfaces can only be applied to after a restart of the device.

For additional information, see chapter Setting the rotary encoding switches.

8 Configuration Modbus TCP

The LioN-X devices support Modbus over a standard TCP network. It is possible to create 4 to 8 socket connections with devices.

All listed register addresses are zero-based. This must be taken into account for read and write access by the Modbus master. If the master is one-based, the register addresses must be added by '1'.

The data is assigned to Holding Registers starting from 40001/400001 with the offset '0'.

8.1 Configuration parameters

Parameters of the LioN-X device can be configured via function code 06 and 16, depending on the features and the designated functionality of the holding register. To avoid malfunctioning of the device, there is a register which acts as latch switch.

Register address	Name	Access	Register length	Size (bytes)	Description
511	Configuration Latch Register	RW (Read/ Write)	1	2	This is a latch register which acts as switch. If "1" is written, then and only then the configuration registers described below can be written. If "0" is written, the updated configuration will be taken by the device It detects transition: "0" to "1": Below described registers will be RW "1" to "0": Configuration apply Default Value: "0"

8.2 General settings

Register address	Name	Access	Register length	Size (bytes)	Description
592	Report U_L/ U_{AUX} supply voltage fault	RW	1	2	0 = Report U_L/U_{AUX} supply voltage fault disabled 1 = Report U_L/U_{AUX} supply voltage fault enabled 2 = Auto Default Value: "1"
593	Report actuator fault without U_L/U_{AUX} voltage	RW	1	2	0 = Report actuator fault without U_L/U_{AUX} voltage enabled 1 = Report actuator fault without U_L/U_{AUX} voltage disabled Default Value: "0"
594	Report U_S voltage fault	RW	1	2	0 = Report U_S voltage fault enabled 1 = Report U_S voltage fault disabled Default Value: "0"
595	Reserved	RW	1	2	
596	Reserved	RW	1	2	
597	Web interface lock	RW	1	2	0 = Web interface lock disabled 1 = Web interface lock enabled Default Value: "0"
598	Force mode lock	RW	1	2	0 = Force lock disabled 1 = Force lock enabled Default Value: "0"
599	External configuration lock	RW	1	2	0 = External configuration lock enabled 1 = External configuration lock disabled Default Value: "1"

8.2.1 Force mode lock

The input and output process data can be forced via different interfaces (e.g. Web interface, REST, OPC UA, MQTT). The support of interfaces depends on the available software features. If the *Force mode lock* is enabled, it is no longer possible to force input and output process data through these interfaces.

Danger: Risk of physical injury or death! Unattended forcing can lead to unexpected signals and uncontrolled machine movements.

8.2.2 Web interface lock

The Web interface access can be configured. If *Web interface lock* is enabled, the Web pages are no longer reachable.

8.2.3 Report U_L/U_{AUX} supply voltage fault

During commissioning, it is possible that no power supply is connected to the U_L/U_{AUX} pins. Therefore it can be helpful to suppress and disable the *U_L/U_{AUX} supply voltage fault* diagnosis.

8.2.4 Report actuator fault without U_L/U_{AUX} voltage

During commissioning, it is possible that no power supply is connected to the U_L/U_{AUX} pins. Therefore it can be helpful to suppress and disable the *Report actuator fault without U_L/U_{AUX} voltage* diagnosis.

8.2.5 Report U_S voltage fault

During commissioning, it is possible that no power supply is connected to the U_S pins. Therefore it can be helpful to suppress and disable the *Report U_S voltage fault* diagnosis.

8.2.6 External configuration lock

Configuration parameters can be set via different alternative interfaces (e.g. Web interface, REST, OPC UA, MQTT). An external configuration can only be done, if no cyclic PLC connection is active. Every new PLC configuration overwrites the external configuration settings.

8.3 Channel settings

Register address	Name	Access	Register length	Size (bytes)	Description
512 .. 527	DO Surveillance Timeout	RW ("Read/ Write")	1	32	DO Surveillance Timeout Port X1 Ch A .. Port X8 Ch B Valid values: 0 .. 255 Default value: 80
528 .. 543	DO Failsafe	RW	1	32	DO Failsafe Port X1 Ch A .. Port X8 Ch B 0: Set Low 1: Set High 2: Hold Last Default value: 0
544 .. 559	DI Filter	RW	1	32	DI Filter Port X1 Ch A .. Port X8 Ch B 0: Disabled 1: 1 ms 2: 2 ms 3: 3 ms 4: 6 ms 5: 10 ms 6: 15 ms Default value: 0
560 .. 575	DI Logic	RW	1	32	DI Logic Port X1 Ch A .. Port X8 Ch B 0: Normally Open 1: Normally Close Default value: 0

Register address	Name	Access	Register length	Size (bytes)	Description
576 .. 591	DO Restart	RW	1	32	DO Restart Port X1 Ch A .. Port X8 Ch B 0: Disable 1: Enable Default value: 1
600 .. 615	DI Latch	RW	1	32	DI Latch 0: DI Latch disabled 1: DI Latch enabled Default value: 0
616 .. 631	DI Extension	RW	1	32	DI Extension Valid values: 0 .. 255 (0 = 0 ms/Off, 16 = 16 ms) Default value: 0

8.3.1 DO Surveillance Timeout

The digital output channels are monitored during runtime. The error states are detected and reported as a diagnosis. To avoid error states during the switching of output channels, the surveillance timeout can be configured as a delay with deactivated monitoring.

The delay time begins with a rising edge of the output control bit. After delay time has elapsed, the output is monitored and error states are reported by diagnosis. When the channel is permanently switched on or off, the typical filter value (not changeable) is 5 ms.

8.3.2 DO Failsafe

The LioN-X devices support a failsafe function for the channels used as digital outputs. In case of an internal device error, the PLC is in STOP state and cannot provide valid process data. The connection is interrupted or the communication is lost. The outputs are controlled according to the configured failsafe values.

Set Low:

If failsafe is active, the physical output pin of the channel is set to low ("0").

Set High:

If failsafe is active, the physical output pin of the channel is set to high ("1").

Hold Last:

If failsafe is active, the physical output pin of the channel holds the last valid process data state ("0" or "1").

8.3.3 DO Restart Mode

In case of a short circuit or overload at an output channel, a diagnosis is reported and the output is switched to "off".

If *DO Restart Mode* is disabled, the output channel is not automatically turned on again. It can be turned on after a logical reset of the process output data of the channel.

If *DO Restart Mode* for this channel is enabled, the output will automatically be turned on again after a fix time delay for checking if the overload or short

circuit condition is still active. When it is active, the channel is switched off again.

8.3.4 DI Logic

The logical state of an input channel can be configured via these parameters. If a channel is set to "Normally Open", a low signal ("0") is transferred to the process input data (e.g. if a non-damped sensor has an open switching output).

If a channel is set to "Normally Close", a high signal ("0") is transferred to the process input data (e.g. if a non-damped sensor has a closed switching output).

The channel LED shows, independent of these settings, the physical input state of the port pin.

8.3.5 DI Filter

A filter time for every digital input channel can be configured by these parameters. When there is no need for a filter it can be disabled. Given values must be divided by '10' to get the actual filter time in ms.

Example: Value "100" = 10 ms

8.3.6 DI Latch

Note: Only applicable for firmware version 11.2 or higher in combination with the latest device description file.

With this parameter enabled, a rising edge at digital input is held high in the input status data (latched) as this is acknowledged by the PLC.

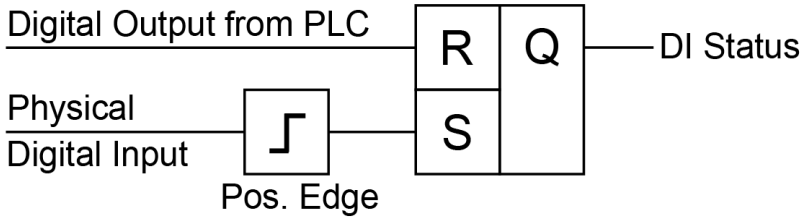


Figure 6: Input latch

R	S	Q
0	0	x (hold last)
0	1	1
1	0	0
1	1	0

Table 10: Truth table for input latch

When the input latch is enabled for a specific channel:

- The corresponding physical input of the channel is logically connected to the set input of a latch via an edge detector.
- The corresponding PLC output control bit (consuming data of I/O device) is logically connected to the reset input of the latch.
- The latch output is connected to the PLC input status data (producing data of the I/O device) of the corresponding channel.
- The latch will work according to the truth table above.
- It is not possible to directly read the physical input status data of that input channel, as the latch output is mapped to the PLC input status data.

The behavior in detail:

- A rising edge on the digital input will trigger the latch and set the latch output to '1'.
- The output will remain '1' until it is reset by the PLC program.
- A logical '1' on the corresponding PLC output control bit for that channel will reset the latch and set the latch output to '0', regardless of the set input or physical input state.
- If the input logic is inverted in the channel configuration, the inverted input logic is connected to the latch. Therefore, it will trigger on the falling edge regarding to a physical input signal.
- If the input is already high during enabling the latch, the latch will set to '1' (Q).

These DI Latch settings only work for channels that have been set to 'digital input mode'. It is recommended to always reset the latch before usage.

Default: Disabled

8.3.7 DI Extension

Note: Only applicable for firmware version 11.2 or higher in combination with the latest device description file.

This parameter extends the duration of the digital input status after a state change at the physical input, when the input state change is faster than the extension time set.

The extension time will be applied on 'high' to 'low' and 'low' to 'high' input transitions. This setting only works for channels that have been set to 'digital input'.

Example:

The DI extension parameter is set to 16 ms, the physical input signal has low status => a high signal is detected for 8 ms.

In this case, the DI channel reports a high-status signal for 16 ms, regardless of other physical input signal transitions during this time.

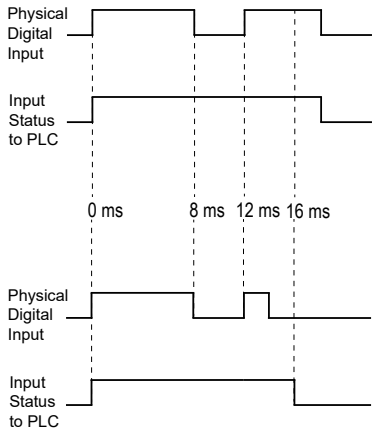


Figure 7: DI Extension

Default: Off (0)

8.4 IO-Link Port 1 .. 8 settings

Register address	Name	Access	Register length	Size (bytes)	Description
1512	IO-Link Port 1 Settings	RW ("Read/ Write")	64	128	For more information, see Detailed channel settings.
1576	IO-Link Port 2 Settings	RW	64	128	
1640	IO-Link Port 3 Settings	RW	64	128	
1704	IO-Link Port 4 Settings	RW	64	128	
1768	IO-Link Port 5 Settings	RW	64	128	
1832	IO-Link Port 6 Settings	RW	64	128	
1896	IO-Link Port 7 Settings	RW	64	128	
1960	IO-Link Port 8 Settings	RW	64	128	

Register address	Name	Access	Register length	Size (bytes)	Description
2024	IO-Link Port 1 Serial Nr.	RO ("Read Only")	16	32	Default value: 0 For more information, see Serial number on page 59.
2040	IO-Link Port 2 Serial Nr.	RO	16	32	
2056	IO-Link Port 3 Serial Nr.	RO	16	32	
2072	IO-Link Port 4 Serial Nr.	RO	16	32	
2088	IO-Link Port 5 Serial Nr.	RO	16	32	
2104	IO-Link Port 6 Serial Nr.	RO	16	32	
2120	IO-Link Port 7 Serial Nr.	RO	16	32	
2136	IO-Link Port 8 Serial Nr.	RO	16	32	

Detailed channel settings

The following parameters for IO-Link Port 1 represent all possible port settings for IO-Link. The values are identical for all 8 ports.

Register address	Name	Access	Register length	Size (bytes)	Description
1512	Port Mode	RW	1	2	0 = Deactivated 1 = IO-Link Manual 2 = IO-Link Auto 3 = Digital Input 4 = Digital Output Default value: 3
1513	Validation and Backup	RW	1	2	0: No device check (no data storage) 1: Type compatible V1.0 device (no data storage) 2: Type compatible V1.1 device (no data storage) 3: Type compatible V1.1 device with Backup + Restore (download + upload) 4 Type compatible V1.1 device with Restore (download master to device) Default value: 0
1514	IQ Mode	RW	1	2	0 = Deactivated 1 = Digital Input 2 = Digital Output 5 = AUX (Class A/B variants only) Default value: 1
1515	Cycle Time	RW	1	2	0: As fast as possible 1: 1.6 ms 2: 3.2 ms 3: 4.8 ms 4: 8.0 ms 5: 20.8 ms 6: 40.0 ms 7: 80.0 ms 8: 120.0 ms Default value: 0
1516	Vendor ID	RW	1	2	0 .. 65535 ("0") Default value: 0

Register address	Name	Access	Register length	Size (bytes)	Description
1517	Device ID	RW	2	4	0 .. 16777215 ("0") Default value: 0
1519	Swapping Mode (Producing)	RW	1	2	0: Raw IO-Link 1 .. 16: 1 .. 16 WORD 17 .. 24: 1 .. 8 DWORD 0 .. 30 Bytes ("0") Default value: 0
1520	Swapping Mode (Consuming)	RW	1	2	0: Raw IO-Link 1 .. 16: 1 .. 16 WORD 17.. 24: 1 .. 8 DWORD 0 .. 30 Bytes ("0") Default value: 0
1521	Swapping Offset (Producing)	RW	1	2	0 .. 30 Bytes ("0") Default value: 0
1522	Swapping Offset (Consuming)	RW	1	2	0 .. 30 Bytes ("0") Default value: 1
1523	Pin2 LED enabled	RW	1	2	0: Disable LED on Channel B 1: Enable LED on Channel B Default value: 1
1524	Suppress All Diagnosis	RW	1	2	0: Generate diagnosis on this channel 1: Do not generate any diagnosis on this channel Default value: 0
1525	Failsafe mode	RW	1	2	0: Set Low 1: Set High 2: Hold Last 3: Replacement Value (transferred via offset 14-46) Default value: 2
1526	IOL Failsafe replacement values	RW	32	64	Default value: 0

Register address	Name	Access	Register length	Size (bytes)	Description
1558	Use Push Pull for Pin 4	RW	1	2	0: Use High-Side switches 1: Use Push Pull Default value: 0
1559	Reserved	RW	1	2	
1560	Pin 4 current limit	RW	1	2	0: U_L, 0.5 A 1: U_L, 1.0 A 2: U_L, 1.5 A 3: U_L, 2.0 A 4: U_L, Max. Default value: 4
1561	Pin 2 current limit	RW	1	2	0: U_L, 0.5 A 1: U_L, 1.0 A 2: U_L, 1.5 A 3: U_L, 2.0 A 4: U_L, Max. Default value: 4
1562 .. 1575	Reserved	RW	14	28	

8.4.1 Port Mode

The *Port Mode* describes how the IO-Link Master handles the presence of an IO-Link device at the port.

Deactivated:

The IO-Link port is deactivated but can be configured for later use. No diagnostics are generated if the IO-Link device is not connected.

IO-Link Manual:

The IO-Link port is activated and explicit port configuration can be done for the parameters *Validation and Backup* (Inspection Level), *Vendor ID*, *Device ID* and *Cycle Time*.

IO-Link Auto:

The IO-Link port is activated and no explicit port configuration is needed. Configurations such as *Validation and Backup* (Inspection Level), *Vendor ID*, *Device ID* and *Cycle Time* are not required.

Digital Input:

In this mode, the channel operates as digital input. The channel state can be seen in the *Digital Input Channel* status of the cyclic process data.

Digital Output:

In this mode, the channel operates as digital output. The channel can be controlled by the *Digital Output Channel Control* (first two bytes of the output data) or by the *IO-Link Output Data* (first byte of each IO-Link device output data) of the cyclic process data. This depends on the *Digital Output Control* parameter in the general settings.

8.4.2 Validation and Backup

With this parameter, the user can set the behavior of the IO-Link ports regarding the type compatibility and data storage mechanism of the connected IO-Link Device.

The precondition for using *Validation and Backup* is that you configure the *Port Mode* to "IO-Link Manual".

The IO-Link Master has a backup memory which can be used for storing the device parameters and for restoring them on the device. This backup memory can be deleted by the following events:

- IO-Link Master factory reset
- *Channel Mode* reconfiguration, e.g. from “Digital-Input” to “IO-Link”
- *Validation and Backup* reconfiguration, e.g. from “No device check” to “Type compatible V1.1 device with Backup & Restore”

For further information refer to the ‘IO-Link Interface and System Specification’ version 1.1.3 which can be downloaded from https://io-link.com/

No device check (no data storage):

No check of connected Vendor ID or Device ID and no "Backup and Restore" support of the IO-Link Master parameter server.

Type compatible V1.0 device (no data storage):

Type compatible according IO-Link specification V1.0 which includes validation of Vendor ID and Device ID. The IO-Link specification V1.0 does not support IO-Link Master parameter server.

Type compatible V1.1 device (no data storage):

Type compatible according IO-Link specification V1.1 which includes validation of Vendor ID and Device ID. "Backup and Restore" is disabled.

Type compatible V1.1 device with Backup + Restore

(upload + download):

Type compatible according IO-Link specification V1.1 which includes validation of Vendor ID and Device ID. "Backup and Restore" is enabled.

Pay attention to the following explanations regarding *Backup and Restore* conditions:

- Backup (Device to Master):

  A Backup (upload from IOL-Device to IOL-Master) is performed when an IO-Link Device is connected and the Master does not have any valid parameter data. The read parameter data are permanently stored on the IO-Link Master.

  An upload will also be performed, when the IO-Link Device has set the DS_UPLOAD_FLAG (Data Storage Upload Flag). This IOL-Device flag can be set in two ways:

  - Parameters written to IOL-Device in *Block Parameter* mode: An IO-Link Device sets the DS_UPLOAD_FLAG self-dependent, if the parameters were written in block parameter mode to the IO-Link Device with the last system command ParamDownloadStore (e.g. by a third party USB IO-Link Master for commissioning).
  - Parameters written to IOL-Device in *Single Parameter* mode: If single parameter data is changed on the IOL-Device during runtime, the stored device parameters on the IOL-Master can be updated using the ParamDownloadStore (index 0x0002, subindex 0x00, value 0x05) command. This command sets the DS_UPLOAD_REQ flag on the IOL-Device and thus the IO-Link Master executes an upload procedure from the IO-Link Device.

- Restore (Master to Device):

  A Restore (download from IOL-Master to IOL-Device) is performed when an IO-Link Device is connected and the IO-Link Master has valid parameter data stored which are usable for the IOL-Device and not equal compared to the device parameters.

  The restore procedure can be blocked by the IO-Link Device via the *Device Access Locks* parameter when supported by the IO-Link Device (Index 0x000C, refer to vendor specific IO-Link Device documentation).

Type compatible V1.1 device with Restore (download Master to Device):

Type compatible according IO-Link specification V1.1 which includes validation of Vendor ID and Device ID. Only "Restore" is enabled.

Pay attention to the following explanations regarding *Restore* conditions:

- Restore (Download / IOL-Master to IOL-Device):

  A Restore (download from IOL-Master to IOL-Device) is performed when an IO-Link Device is connected and the IO-Link Master has valid parameter data stored which are usable for the IOL-Device and not equal compared to the device parameters.

  In the *Restore* mode no change of the IOL-Device parameters will be stored permanently on the IOL-Master. When the IOL-Device sets the DS_UPLOAD_FLAG in this mode, the device parameters will be restored by the IOL-Master.

  The restore procedure can be blocked by the IO-Link Device via the *Device Access Locks* parameter when supported by the IO-Link Device (Index 0x000C, refer to vendor specific IO-Link Device documentation).

8.4.3 IQ Mode

The operating mode of Pin 2 (Channel B) of the respective IO-Link channel can be configured via this parameter.

Digital Output:

In this mode, the channel operates as digital output. The channel can be controlled by the *Digital Output Channel Control* (first two bytes of the output data) or by the *IO-Link Output Data* (first byte of each IO-Link device output data) of the cyclic process data. This depends on the *Digital Output Control* parameter in the general settings.

Digital Input:

In this mode, the channel operates as digital input. The channel state can be seen in the *Digital Input Channel* status of the cyclic process data.

8.4.4 Cycle Time

The IO-Link cycle time can be configured by this parameter.

As fast as possible:

The IO-Link port uses the max. supported IO-Link Device and Master update cycle time for the cyclic I/O data update between IO-Link Master and IO-Link Device.

1.6 ms, 3.2 ms, 4.8 ms, 8.0 ms, 20.8 ms, 40.0 ms, 80.0 ms, 120.0 ms:

The cycle time can be set manually to the provided options. This option can be used e.g. for IO-Link devices which are connected over inductive couplers. Inductive couplers are normally the bottleneck in the update cycle time between IO-Link Master and IO-Link Device. In this case, please refer to the data sheet of the inductive coupler.

8.4.5 Vendor ID

The *Vendor ID* is needed for the validation of the IO-Link device and can be configured with this parameter.

Precondition for using the *Vendor ID* is that you configure *Port Mode* to "IO-Link Manual". *Validation and Backup* must be set to a type compatible V1.X device.

8.4.6 Device ID

The *Device ID* is needed for the validation of the IO-Link device and can be configured with this parameter.

Precondition for using the *Device ID* is that you configure *Port Mode* to "IO-Link Manual". *Validation and Backup* must be set to a type compatible V1.X device.

8.4.7 Swapping Mode

The byte order of IO-Link is big endian. For setting output data in the correct format, the parameters *Swapping Mode* and *Swapping Offset* support the user. There can be selected up to 16 words or up to 8 double words for converting the output data.

Raw IO-Link Data:

No byte swap

Data type WORD:

Data byte order: Byte 1, Byte 2

Order after Swap: Byte 2, Byte 1

Data type DWORD:

Data byte order: Byte 1, Byte 2, Byte 3, Byte 4

Order after Swap: Byte 4, Byte 3, Byte 2, Byte 1

8.4.8 Swapping Offset

The *Swapping Offset* describes the start point in the process data for using the configured *Swapping Mode*.

8.4.9 IOL Failsafe

The LioN-X devices support a failsafe function for the output data of the IO-Link channels. In case of an internal device error, the PLC is in STOP state and cannot provide valid process data, the connection is interrupted or the communication is lost: The output data of the IO-Link channels is controlled by the configured failsafe values.

Set Low:

If failsafe is active, all bits of the IO-Link output data are set to low ("0").

Set High:

If failsafe is active, all bits of the IO-Link output data are set to high ("1").

Hold Last:

If failsafe is active, all bits of the IO-Link output data arel hold the last valid process data state ("0" or "1").

Replacement Value:

A replacement value can be set via the *IO-Link Failsafe* parameter object for every IO-Link device. If failsafe is active, these replacement values are transmitted to the IO-Link device. Take into account that in the case of an error the replacement values are sent instead of the output process data so that a configured *Swapping Mode* has influence on the byte order.

IO-Link Master Command:

If failsafe is active, an IO-Link-specific mechanism for valid/invalid output process data is used and the IO-Link device determines the behavior itself.

8.4.10 IOL Failsafe replacement values

32 holding registers represent byte-wise 32 replacement values. If failsafe is active, these values are transmitted to IO-Link device.

8.4.11 Serial number

The serial number of a connected IO-Link Device can be read from the holding registers. Each serial number has a maximum size of 32 bytes, meaning 16 registers are assigned to each IO-Link port.

9 Process data assignment

The LioN-X devices in general support process data communication in both directions. The consuming data in this context is defined as the process output data which controls physical outputs and IO-Link output data. The producing data in this context is defined as the process input data which contains the physical inputs, diagnostics and IO-Link input data with optional extended status and event data.

The following chapters describe the data images for the consuming and producing data direction which are assigned to the Holding Registers.

All listed register addresses are zero-based. This must be taken into account for read and write access by the Modbus master. If the master is one-based, the register addresses must be added by '1'.

The data is assigned to Holding Registers starting from 40001/400001 with the offset '0'.

9.1 Consuming data (output)

Register address	Name	Access	Register length	Size (byte)	Description
0	Process Data Output Port 1	RW ("Read/ Write")	16	32	Each holding register holds two bytes of output data. Valid process data values depend on the port configuration.
16	Process Data Output Port 2	RW	16	32	
32	Process Data Output Port 3	RW	16	32	
48	Process Data Output Port 4	RW	16	32	
64	Process Data Output Port 5	RW	16	32	
80	Process Data Output Port 6	RW	16	32	
96	Process Data Output Port 7	RW	16	32	
112	Process Data Output Port 8	RW	16	32	

Examples:

Port	Mode	Valid output data to be written
1	IO-Link 1 byte output	Least significant 1 byte of register 0
2	Digital output	Least significant bit (bit 0) of register 16
3	IO-Link 4 bytes output	All 2 bytes on registers 32 to 33
4	IO-Link 16 bytes output	All 2 bytes on registers 48 to 56
5	IO-Link 32 bytes output	All 2 bytes on registers 64 to 79

Table 11: Examples for consuming data

9.2 Producing data (input)

Register address	Name	Access	Register length	Size (byte)	Description
256	Process Data Input Port 1	RO ("Read Only")	16	32	Each holding register holds two bytes of input data. Valid process data values depend on the port configuration.
272	Process Data Input Port 2	RO	16	32	
288	Process Data Input Port 3	RO	16	32	
304	Process Data Input Port 4	RO	16	32	
320	Process Data Input Port 5	RO	16	32	
336	Process Data Input Port 6	RO	16	32	
352	Process Data Input Port 7	RO	16	32	
368	Process Data Input Port 8	RO	16	32	

Examples:

Port	Mode	Valid input data to be read
1	IO-Link 1 byte input	Least significant 1 byte of register 256
2	Digital input	Least significant bit (bit 0) of register 272
3	IO-Link 4 bytes input	All 2 bytes on registers 288 to 289
4	IO-Link 16 bytes input	All 2 bytes on registers 304 to 311
5	IO-Link 32 bytes input	All 2 bytes on registers 320 to 335

Table 12: Examples for producing data

9.3 Channel B as digital output

Register address	Name	Access	Register length	Size (byte)	Description
435	PIN 2 (Channel B) as DO (Bit wise)	RW ("Read/ Write")	1	2	This holding register represents bitwise the values of Channel B (Pin 2) of IO-Link class-A port. Values written on this register will only have an effect if Channel B is configured as Digital Output.

400436	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Ch. B Digital Output	NA	NA	NA	NA	NA	NA	NA	NA	X8	X7	X6	X5	X4	X3	X2	X1

9.4 Channel B as digital input

Register address	Description	Access	Register-Länge	Size (byte)	Description
434	PIN 2 (Channel B) as DI (Bit wise)	RO ("Read Only")	1	2	This holding register represents bitwise the values of Channel B (Pin 2) of IO-Link class-A port. Values written on this register will only have an effect if Channel B is configured as Digital Input.

400435	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Ch. B Digital Input	NA	NA	NA	NA	NA	NA	NA	NA	X8	X7	X6	X5	X4	X3	X2	X1

10 Diagnostics processing

All listed register addresses are zero-based. This must be taken into account for read and write access by the Modbus master. If the master is one-based, the register addresses must be added by '1'.

The data is assigned to Holding Registers starting from 40001/400001 with the offset '0'.

Register address	Name	Access	Register length	Size (byte)	Description
400	System/ sensor power supply present	RO ("Read Only")	1	2	0: System/sensor power not present 1: System/sensor power supply present
401	Error of the system/ sensor power supply	RO	1	2	0: System/sensor power supply fault not present 1: System/sensor power supply fault occurred
402	Auxiliary/ Actuator power supply present	RO	1	2	0: Auxiliary/Actuator power supply not present 1: Auxiliary/Actuator power supply present
403	Error of the auxiliary/ actuator power supply	RO	1	2	0: Auxiliary/actuator power supply fault not present 1: Auxiliary/actuator power supply fault occurred
404	Internal Module Error	RO	1	2	
405	Force mode diagnosis	RO	1	2	
406	Overload/ short-circuit of the I/O port sensor supply outputs	RO	1	2	This register represent bitwise status of Overload/ short-circuit.
407	Overload/ short circuit of the digital outputs Channel A (CQ)	RO	1	2	This register represent bitwise status of overload/ short-circuit on digital outputs Channel A (CQ).

Register address	Name	Access	Register length	Size (byte)	Description
408	IQ short fault Overload/ short circuit of the digital outputs Channel B (IQ).	RO	1	2	This register represent bitwise status of overload/ short-circuit on digital outputs Channel B (IQ).
409	Overload/ short circuit between Actuator voltage supply (Class B)	RO	1	2	This register represent bitwise status of Actuator voltage supply (Class B).

Register address	Name	Access	Register length	Size (byte)	Description
426	Status of IO-LinkPort 1	RO	1	2	
427	Status of IO-LinkPort 2	RO	1	2	
428	Status of IO-LinkPort 3	RO	1	2	
429	Status of IO-LinkPort 4	RO	1	2	
430	Status of IO-LinkPort 5	RO	1	2	
431	Status of IO-LinkPort 6	RO	1	2	
432	Status of IO-LinkPort 7	RO	1	2	
433	Status of IO-LinkPort 8	RO	1	2	

Register address	Name	Access	Register length	Size (byte)	Description
2501	IO-Link eventsPort 1	RO	200	400	

Register address	Name	Access	Register length	Size (byte)	Description
2701	IO-Link eventsPort 2	RO	200	400	
2901	IO-Link eventsPort 3	RO	200	400	
3101	IO-Link eventsPort 4	RO	200	400	
3301	IO-Link eventsPort 5	RO	200	400	
3501	IO-Link eventsPort 6	RO	200	400	
3701	IO-Link eventsPort 7	RO	200	400	
3901	IO-Link eventsPort 8	RO	200	400	

10.1 Error of the system/sensor power supply

The voltage value for the incoming system/sensor power supply is also monitored globally. If the voltage drops below approx. 18 V, or exceeds approx. 30 V, an error diagnosis is generated. The IO-Link specification requires at least 20 V at the L+ (pin1) output supply of the I/O ports. At least 21 V of U_S supply voltage for the IO-Link Master are required to minimize the risk of internal voltage drops in the IO-Link Master.

The green U_S indicator is off.

The error diagnosis has no effect on the outputs.

Caution: It must definitely be ensured that the supply voltage, measured at the most remote participant is not below 21 V DC from the perspective of the system power supply.

The diagnostics in the holding register 400 will be updated.

10.2 Error of the auxiliary/actuator power supply

The voltage value for the incoming auxiliary/actuator power supply is also monitored globally. If *Report U_L/U_{AUX} Supply Voltage Fault* is enabled, an error message is generated when the voltage drops below approx. 18 V or exceeds approx. 30 V. The U_L/U_{AUX} indicator shows red.

Holding register 401 can be read to get the present status of the auxiliary/ actuator power supply. The diagnostic data on holding register 402 will be updated.

If output channels are set to *High State* and *Report DO Fault without U_L/U_{AUX}*, additional error diagnostics, caused by the voltage failure, are generated on the channels.

If *Report U_L/U_{AUX} Supply Voltage Fault* is disabled, no U_L/U_{AUX} or channel diagnostics appear.

10.3 Overload/short-circuit of the I/O port sensor supply outputs

In case of an overload or a short circuit between pin 1 and pin 3 on the ports (X1 .. X8), holding register 406 will be updated with the following data:

406	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
sensor supply Short circuit	NA	NA	NA	NA	NA	NA	NA	NA	X8	X7	X6	X5	X4	X3	X2	X1

10.4 Overload/short circuit of the digital outputs

In case of an overload or a short circuit between pin 4 and pin 3 on the ports (X1 .. X8), holding register 407 will be updated with the following data. This diagnosis is valid if the respective port's channel A is in *Digital Output Mode*.

407	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Overload/ short circuit of Channel A (CQ)	NA	NA	NA	NA	NA	NA	NA	NA	X8	X7	X6	X5	X4	X3	X2	X1

In case of an overload or a short circuit between pin 2 (Channel B) and pin 3 on the ports (X1 .. X8), holding register 408 will be updated with the following data. This diagnosis is valid if the respective port's channel B is in *Digital Output Mode*.

408	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Overload/ short circuit of Channel B (IQ)	NA	NA	NA	NA	NA	NA	NA	NA	X8	X7	X6	X5	X4	X3	X2	X1

10.5 Overload/short circuit of Actuator voltage supply P24 (Class B)

In case of an overload or a short circuit of the Actuator voltage supply P24 (Class B) on the ports X5 .. X8, the holding register 409 will be updated with the following data.

409	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Overload/ short circuit of Actuator voltage supply P24 (Class B)	NA	NA	NA	NA	NA	NA	NA	NA	X8	X7	X6	X5	NA	NA	NA	NA

10.6 IO-Link channel status

IO-Link contains standard data for the channel status. The states of an individual IO-Link channel can be read on holding registers 425 .. 432.

States of IO-Link channel

0:	No device present
1:	Deactivated
2:	Diagnosis mode
3:	Pre-OP mode
4:	OP mode
5:	DI mode
6:	DO mode
254:	Power OFF
255:	No status available

10.7 IO-Link channel events

During operation, the IO-Link Master generates events for a particular channel. Holding registers 2500 .. 4099 will hold up to 32 events.

Register address	Name	Access	Register length	Size (byte)
2500	IO-Link Port 1 No. of events present	RO ("Read Only")	1	2
2501	IO-Link Port 1 Event 1	RO	5	10
2506	IO-Link Port 1 Event 2	RO	5	10
…	…	…		
2661	IO-Link Port 1 Event 32	RO	5	10
2666	Reserved	RO	33	66
2700	IO-Link Port 2 No. of events present	RO	1	2
2701	IO-Link Port 2 Event 1	RO	5	10
2706	IO-Link Port 2 Event 2	RO	5	10
…	…	…		
2761	IO-Link Port 2 Event 32	RO	5	10
2766	Reserved	RO	33	66
…	…	…		
…	…	…		
3900	IO-Link Port 8 No. of events present	RO	1	2
3901	IO-Link Port 8 Event 1	RO	5	10
3906	IO-Link Port 8 Event 2	RO	5	10
…	…	…		
3961	IO-Link Port 8 Event 32	RO	5	10
3966	Reserved	RO	33	66

The following example shows details of a single event which is applicable for all 32 events on 8 channels:

Register address	Name	Access	Register length	Size (byte)
2501	Timestamp	RO	2	4
2503	Event Type (Classification)	RO	1	2
2504	Event Code	RO	1	2
2505	Reserved	RO	1	2

The 32 event registers act like a ring buffer. Once FIFO buffer is full, the earliest event will be removed from buffer and all other events will be pushed one step up. Latest event details will be available then on event 32.

Timestamp:

Local timestamp of a device. Takes 4 bytes.

Event Type:

Classification of events. There are 3 different types of events. The type is classified via the values 1 .. 3 on the respective holding register.

1:	Notification
2:	Warning
3:	Error

Event Code:

The event code represents a unique code for individual events as per IO-Link specifications. The following event code(s) will be generated on the respective holding register.

0x1000:	General Malfunction
0x1800:	No Device
0x1801:	Startup parametrization error
0x1802:	Incorrect Vendor ID - Inspection Level mismatch
0x1803:	Incorrect Device ID - Inspection Level mismatch
0x1804:	Short circuit at C/Q
0x1805:	PHY over temperature
0x1806:	Short circuit at L+
0x1807:	Overcurrent at L+
0x1808:	Device Event overflow
0x1809:	Backup inconsistency - memory out of range
0x180A:	Backup inconsistency - identity fault
0x180B:	Backup inconsistency - Data Storage unspecific error
0x180C:	Backup inconsistency - upload fault
0x180D:	Parameter inconsistency - download fault
0x180E:	P24 (Class B) missing or under voltage
0x180F:	Short circuit at P24 (Class B)
0x1810:	Short circuit at I/Q
0x1811:	Short circuit at C/Q (Digital)

0x1812:	Overcurrent at I/Q
0x1813:	Overcurrent at C/Q (Digital)
0x6000:	Invalid cycle time
0x6001:	Revision fault
0x6002:	ISDU batch failed
0xFF21:	DL: Device plugged in
0xFF22:	Device communication lost
0xFF23:	Data Storage identification mismatch
0xFF24:	Data Storage buffer overflow
0xFF25:	Data Storage parameter access denied
0xFF26:	Port status changed
0xFF27:	Data Storage upload completed
0xFF31:	DL: Incorrect Event signaling

11 IIoT functionality

The LioN-X variants offer a number of new interfaces and functions for the optimal integration into existing or future IIoT (Industrial Internet of Things) networks. The devices continue to work as field bus devices which communicate with and are controlled by a PLC (Programmable Logic Controller).

In addition, the devices offer common IIoT interfaces, which enable new communication channels besides the PLC. The communication is performed via IIoT-relevant protocols MQTT and OPC UA. With the help of these interfaces not only all information in a LioN-X device can be read. They also enable its configuration and control, if the user wishes. All interfaces can be configured extensively and offer read-only functionality.

All LioN-X variants provide user administration, which is also applicable for accessing and configuring the IIoT protocols. This allows you to manage all modification options for the device settings via personalized user authorizations.

All IIoT protocols can be used and configured independently of the field bus. It is also possible to use the devices completely without the help of a PLC and control them via IIoT protocols.

Attention: When using the IIoT functionality, a protected local network environment without direct access to the Internet is recommended.

Attention: Only activate one of the IIoT protocols at a time. Exclusively use MQTT **or** OPC UA.

11.1 MQTT

MQTT functions are **only** applicable for the following LioN-X variants:

- 0980 XSL 3912-121-007D-00F-BD

The MQTT (Message Queuing Telemetry Transport) protocol is an open network protocol for machine-to-machine communication, which provides the transmission of telemetric data messages between devices. The integrated MQTT client allows the device to publish a specific set of information to an MQTT broker.

The publishing of messages can either occur periodically or be triggered manually.

Attention: When using MQTT, the OPC UA protocol must be disabled.

11.1.1 MQTT configuration

In **delivery state**, MQTT functions are **disabled**. The MQTT client can be configured either using the Web interface or directly via a JSON object sent in an HTTP/HTTPS request. For more information see chapter MQTT configuration - Quick start guide on page 94.

The configuration URL is:

`http://[ip-address]/w/config/mqtt.json`

The configuration can also read back as a JSON file:

`http://[ip-address]/r/config/mqtt.json`

The configuration is a JSON object. Each JSON member is a configuration element. The object must not contain all elements. Only the provided elements will be changed. The configuration changes apply only after a device restart.

The following configuration elements are available (default values in bold):

Element	Data type	Description	Example data
mqtt-enable	boolean	Master switch for the MQTT client.	true / **false**
broker	string	IP address of the MQTT Broker	"**192.168.1.1**"
login	string	Username for MQTT Broker	"admin" (Default: **null**)
password	string	Password for MQTT Broker	"private" (Default: **null**)
port	number	Broker port	**1883**
base-topic	string	Base topic	"iomodule_[mac]" (Default: "**lionx**")
will-enable	boolean	If true, the device provides a last will message to the broker	true / **false**
will-topic	string	The topic for the last will message.	(Default: **null**)
auto-publish	boolean	If true, all enabled domains will be published automatically in the specified interval.	**true** / false
publish-interval	number	The publish interval in ms if auto-publish is enabled. Minimum is 250 ms.	**2000**
publish-identity	boolean	If true, all identity domain data will be published	**true** / false
publish-config	boolean	If true, all config domain data will be published	**true** / false
publish-status	boolean	If true, all status domain data will be published	**true** / false
publish-process	boolean	If true, all process domain data will be published	**true** / false
publish-devices	boolean	If true, all IO-Link Device domain data will be published	true / **false**
commands-allowed	boolean	Master switch for MQTT commands. If false, the device will not subscribe to any command topic, even if specific command topics are activated below.	true / **false**
force-allowed	boolean	If true, the device accepts force commands via MQTT.	true / **false**
reset-allowed	boolean	If true, the device accepts restart and factory reset commands via MQTT.	true / **false**
config-allowed	boolean	If true, the device accepts configuration changes via MQTT.	true / **false**

Element	Data type	Description	Example data
qos	number	Selects the "Quality of Service" status for all published messages.	**0 = At most once** 1 = At least once 2 = Exactly once

Table 13: MQTT configuration

MQTT response:

The resulting response is a JSON object with a "status" field. Status should be "0" if no error occurred and "-1" if there is an error.

In case of an error, the response contains an error array.

The error array contains an error object for each error occurred. The object consists of a field "Element" which names the config element which caused the error, and a field "Message" for the error message.

- A malformed JSON object produces an error.
- Not existing parameters produce an error.
- Parameters with a wrong data type produce an error.

It is not allowed to write all available parameters at once. You may write only one or a limited number of parameters.

Examples:

```
{"status": -1, "error": [{"Element": "publish-interval", "Message": "Integer
 expected"}]}

{"status": 0}

{"status": -1, "error": [{"Element": "root", "Message": "Not a JSON
 object"}]}
```

For more information see chapter MQTT topics on page 82.

11.1.2 MQTT topics

MQTT mainly relates to topics. All messages are attached to a topic which adds context to the message itself. Topics may consist of any string and they are allowed to contain slashes (`/`) as well as wildcard symbols (`*, #`).

11.1.2.1 Base topic

For all LioN-X variants there is a configurable Base topic which is the prefix for all topics. The Base topic can be chosen freely by the user. The Base topic can also contain selected variables as shown in Table 14: Base topic variables on page 82.

Variables in the Base topic have to be written in brackets ("[ ]"). The following variables are possible:

Variable	Description
mac	The MAC address of the device
name	The name of the device
order	The ordering number of the device
serial	The serial number of the device

Table 14: Base topic variables

Example:

The Base topic `"io_[mac]"` translates to `"io_A3B6F3F0F2F1"`.

All data is organized in domains. The domain name is the first level in the topic after the Base topic. Note the following notation:

`Base-Topic/domain/….`

There are the following domains:

Domain name	Definition	Example content
identity	All fixed data which is defined by the used hardware and which cannot be changed by configuration or at runtime.	Device name, ordering number, MAC address, port types, port capabilites and more.
config	Configuration data which is commonly loaded once at startup, mostly by a PLC.	IP address, port modes, input logic, failsafe values and more.
status	All (non-process) data which changes quite often in normal operation.	Bus state, diagnostic information, IO-Link Device status and data.
process	All process data which is produced and consumed by the device itself or by attached devices.	Digital inputs, digital outputs, cyclic IO-Link data.
iold	IO-Link Device parameters according to the IO-Link specification.	Vendor name, product name, serial number, hardware revision, software revision and more.

Table 15: Data domains

There is often one topic used for all gateway related information and topics for each port. All identity topics are published just once at start-up, because this information should never change. All other topics are published either in a fixed interval or just triggered manually, according to the configuration.

Topic	Content examples	Total publish count	Publish interval
[base-topic]/identity/gateway	Name, ordering number, MAC, vendor, I&M etc.	1	Startup
[base-topic]/identity/port/n	Port name, port type	8	Startup
[base-topic]/config/gateway	Configuration parameters, ip address etc.	1	Interval
[base-topic]/config/port/n	Port mode, data storage, mapping, direction	8	Interval
[base-topic]/status/gateway	Bus state, device diagnosis, master events	1	Interval
[base-topic]/status/port/n	Port or channel diagnosis, IO-Link state, IO-Link Device events	8	Interval
[base-topic]/process/gateway	All Digital IN/OUT	1	Interval
[base-topic]/process/port/n	Digital IN/OUT per port, IOL-data, pdValid	8	Interval
[base-topic]/iold/port/n	IO-Link Device parameter	8	Interval

Table 16: Data model

An MQTT client which wants to subscribe to one or more of these topics can also use wildcards.

Full topic	Description
[base-topic]/identity/gateway	Receive only indentity objects for the gateway
[base-topic]/identity/#	Receive all data related to the identity domain
[base-topic]/status/port/5	Receive only status information for port number 5
[base-topic]/+/port/2	Receive information of all domains for port number 2
[base-topic]/process/port/#	Receive only process data for all ports
[base-topic]/config/#	Receive config data for the gateway and all ports.

Table 17: Use case examples

11.1.2.2 Publish topic

Overview of all publish JSON data for the defined topics:

Key	Data type
product_name	json_string
ordering_number	json_string
device_type	json_string
serial_number	json_string
mac_address	json_string
production_date	json_string
fw_name	json_string
fw_date	json_string
fw_version	json_string
hw_version	json_string
vendor_name	json_string
vendor_address	json_string
vendor_phone	json_string
vendor_email	json_string
vendor_techn_support	json_string
vendor_url	json_string
vendor_id	json_integer
device_id	json_integer

Table 18: Identity/gateway

Key	Data type	Range	Default value	Remarks
fieldbus_protocol	json_string	PROFINET, EtherNet/IP, EtherCAT®		
ip_address	json_string		192.168.1.1	
subnet_mask	json_string		255.255.255.0	
report_alarms	json_boolean		0.0.0.0	
report_ul_alarm	json_boolean	true / false	true	
report_do_fault_without_ul	json_boolean	true / false	false	
force_mode_lock	json_boolean	true / false	false	
web_interface_lock	json_boolean	true / false	false	
do_auto_restart	json_boolean	true / false	true	
fast_startup	json_boolean	true / false	false	PROFINET and EIP only

Table 19: Config/gateway

Key	Data type	Range	Default value	Remarks
protocol	json_string	wait_for_io_system wait_for_io_Connection failsafe connected error		
ethernet_port1	json_string	100_mbit/s_full 100_mbit/s 10_mbit/s_full 100_mbit/s		
ethernet_port2	json_string	100_mbit/s_full 100_mbit/s 10_mbit/s_full 100_mbit/s		
module_restarts	json_integer	0 .. 4294967295		
channel_diagnosis	json_boolean	true / false		
failsafe_active	json_boolean	true / false		
system_voltage_fault	json_boolean	true / false		
actuator_voltage_fault	json_boolean	true / false		
internal_module_error	json_boolean	true / false		
simulation_active_diag	json_boolean	true / false		
us_voltage	json_integer	0 .. 32		in Volts
ul_voltage	json_integer	0 .. 32		in Volts
forcemode_enabled	json_boolean	true / false		

Table 20: Status/gateway

Key	Data type	Range	Default value	Remarks
Input_data	json_integer[]			
output_data	json_integer[]			

Table 21: Process/gateway

Key	Data type	Range	Default value	Remarks
port	json_integer	1 .. 8		
type	json_string	digital_universal digital_input digital_Output io_link		
max_output_power_cha	json_string	2.0_mA 0.5_mA		
max_output_power_chb	json_string	2.0_mA 0.5_mA		
channel_cha	json_string	input/output input output io_link aux		
channel_chb	json_string	input/output input output io_link aux		

Table 22: Identity/port/1 .. 8

Key	Data type	Range	Default value	Remarks
port	json_integer	1 .. 8		
direction_cha	json_string	input/output input output		
restart_mode_cha	json_string	Manual Auto		
restart_mode_chb	json_string	Manual Auto		
input_polarity_cha	json_string	NO NC		
input_polarity_chb	json_string	NO NC		
input_filter_cha	json_integer			ms
input_filter_chb	json_integer			ms
do_auto_restart_cha	json_boolean	true / false		
do_auto_restart_chb	json_boolean	true / false		

Table 23: Config/port/1 .. 8

Key	Data type	Range	Default value	Remarks
port	json_integer	1 .. 8		
physical_state_cha	json_integer	0 .. 1		
physical_state_chb	json_integer	0 .. 1		
actuator_short_circuit_cha	json_boolean	true / false		
actuator_short_circuit_chb	json_boolean	true / false		
sensor_short_circuit	json_boolean	true / false		
current_cha	json_integer			mA
current_chb	json_integer			mA
current_pin1	json_integer			mA

Table 24: Status/port/1 .. 8

11.1.2.3 Command topic (MQTT Subscribe)

The main purpose of MQTT is to publish data from the device to a broker. This data can then be received by any subscriber who is interested in this data. But also the other way round is possible. The device can subscribe to a topic on the broker and is then able to receive data. This data can contain configuration or forcing data. This allows the user to fully control a device via MQTT only, without using other ways of communication like Web or REST.

If the configuration allows commands in general, the device subscribes to special Command topics on which it can receive commands from other MQTT clients. The Command topic is based upon the Base topic. It always has the following form:

`[base-topic]/command`

After the Command topic, there are fixed topics for different writeable objects. The data format of the MQTT payload is always JSON. It is possible to set only a subset of the possible objects and fields.

[…]/forcing

Use the Command topic `[base-topic]/command/forcing` for *Force object* data. The *Force object* can contain any of the following properties:

Property	Data type	Example values	Remarks
forcemode	boolean	true / false	Forcing Authority: on/off
digital	array (Table 26: Force object: Digital on page 91)		
iol	array (Table 27: Force object: IOL (IO-Link devices only) on page 91)		

Table 25: Force object properties

For the *Force object* properties `digital` and `iol`, there are several value specifications arrayed:

Property	Data type	Example values	Remarks
port	integer	1, 2, 5	
channel	string	"a", "b"	
force_dir	string	"out", "in", "clear"	
force_value	integer	0, 1	

Table 26: Force object: Digital

Property	Data type	Example values	Remarks
port	integer	0, 1, 5	
output	array[integer]	[55, 88, 120]	
input	array[integer]		Input-Simulation

Table 27: Force object: IOL (IO-Link devices only)

[…]/config

Use the Command topic `[base-topic]/command/config` for *Config object* data. The *Config object* can contain any of the following properties:

Property	Data type	Example values	Remarks
portmode	array (Table 29: Config object: Portmode on page 92)		
ip_address	string	"192.168.1.5"	
subnet_mask	string	"255.255.255.0"	
gateway	string	"192.168.1.100"	

Table 28: Config object properties

For the *Config object* property `portmode`, there are several value specifications arrayed:

Property	Data type	Example values	Remarks
port	integer	2	
channelA*	string	"dio", "di", "do", "iol", "off"	
channelB*	string	"dio", "di", "do", "iol", "off", "aux"	
inlogicA	string	"no", "nc"	
inlogicB	string	"no", "nc"	
filterA	integer	3	input filter in ms
filterB	integer	3	input filter in ms
autorestartA	boolean		
autorestartB	boolean		
iolValidation	integer	0 = NoCheck 1 = Type 1.0 2 = Type 1.1 3 = Type 1.1 BR 4 = Type 1.1 RES	
iolDeviceID	integer		for validation
iolVendorID	integer		for validation

Table 29: Config object: Portmode

*channelA = Pin 4, channelB = Pin 2

[…]/reset

Use the Command topic `[base-topic]/command/reset` for *Reset object* data about restart and factory reset issues. The *Reset object* can contain any of the following properties:

Property	Data type	Example values	Remarks
factory_reset	boolean	true / false	
system_reset	boolean	true / false	

Table 30: Reset object properties

[…]/publish

Use the Command topic `[base-topic]/command/publish` for *Publish object* data.

Trigger publish of all topics manually (can be used when auto publish is off or long interval is set).

11.1.3 MQTT configuration - Quick start guide

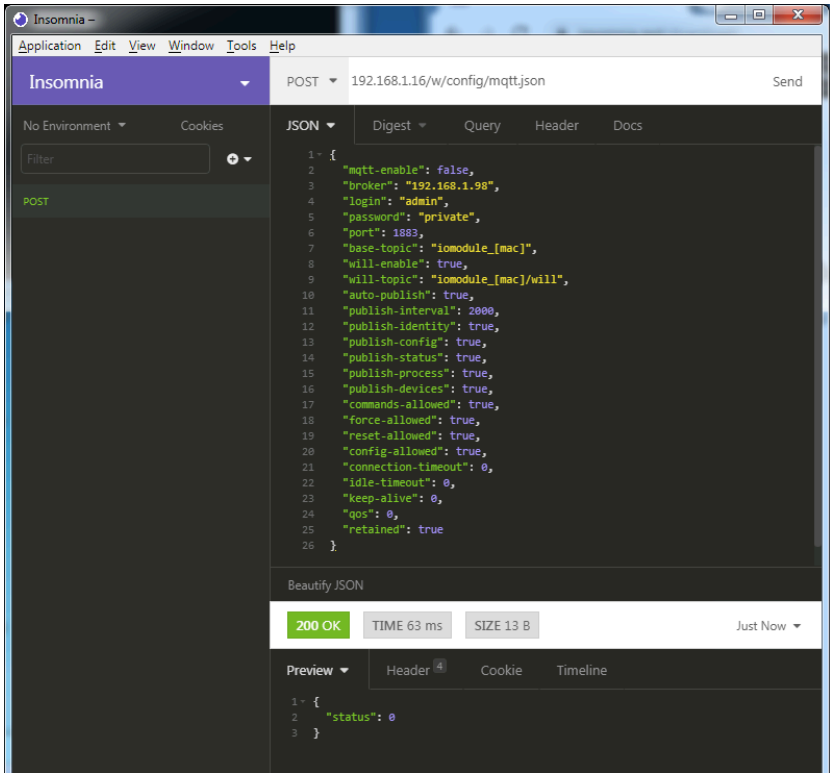
Attention: Belden Deutschland GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

11.1.3.1 MQTT configuration via JSON

1. Depending on your application case, download and install *Insomnia* or a comparable application: https://insomnia.rest/download/

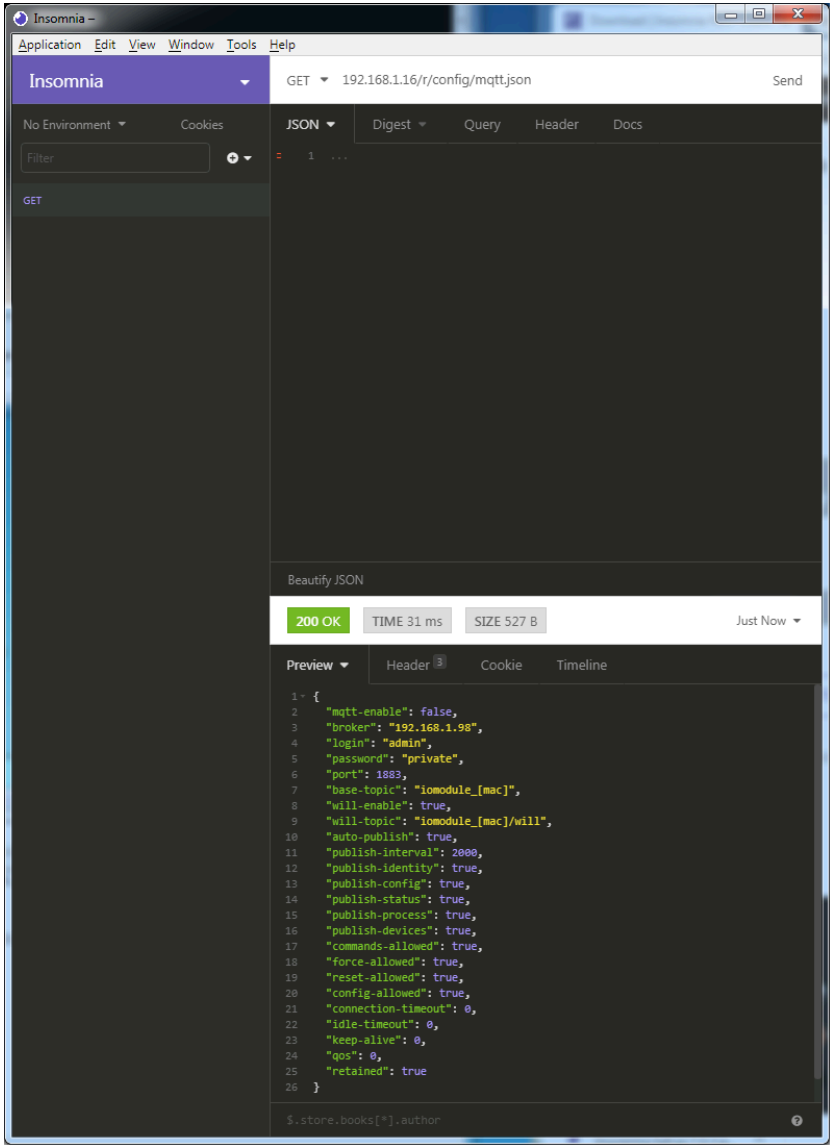
2. Configure MQTT:

POST: `[IP-address]/w/config/mqtt.json`



3. Read MQTT:

GET: `[IP-address]/r/config/mqtt.json`



11.2 OPC UA

OPC UA functions are **only** applicable for the following LioN-X variants:

- 0980 XSL 3912-121-007D-00F-BD

OPC Unified Architecture (OPC UA) is a platform-independent standard with a service-oriented architecture for communication in and with industrial automation systems.

The OPC UA standard is based on the client-server principle and lets machines and devices, regardless of any preferred field bus, communicate horizontally among each other as well as vertically to the ERP system or the cloud. LioN-X provides an OPC UA server on field device level, with which an OPC UA client can connect for information exchange secure in transmission.

For OPC UA, we comply (apart from the exceptions listed below) with the IO-Link Companion Specification, which can be downloaded from https://catalog.belden.com or directly from io-link.com.

Attention: When using OPC UA, the MQTT protocol must be disabled.

Feature	Support
Managing IODDs (chapter 6.1.6 in the specification)	Not supported
Mapping IODD information to OPC UA ObjectTypes (chapter 6.3 in the specification)	Not supported
IOLinkIODDDeviceType (chapters 7.2 ff. in the specification)	Not supported
ObjectTypes generated based on IODDs (chapters 7.3 ff. in the specification)	Not supported
Creation of Instances based on ObjectTypes generated out of IODDs (chapter 7.4 in the specification)	Not supported
IODDManagement Object (chapter 8.2 in the specification)	Not supported
RemoveIODD Method (chapter 8.3 in the specification)	Not supported

Table 31: Non-supported OPC UA features according to the IO-Link Companion Specification

11.2.1 OPC UA configuration

In **delivery state**, OPC UA functions are **disabled**. The OPC UA Server can be configured either using the Web interface or directly via a JSON Object sent in an HTTP/HTTPS request. For more information see OPC UA configuration - Quick start guide on page 100.

The configuration URL is:

`http://[ip-address]/w/config/opcua.json`

The configuration can also read back as a JSON file:

`http://[ip-address]/r/config/opcua.json`

The configuration is a JSON object. Each JSON member is a configuration element. The object must not contain all elements. Only the provided elements will be changed. All configuration changed applies only after a device restart.

There are the following configuration elements (default values in bold):

Element	Data type	Description	Example data
port	integer	Server port for the OPC UA server.	0, **4840**, 0xFFFF
opcua-enable	boolean	Master switch for the OPC UA server.	true / **false**
anon-allowed	boolean	If true, anonymous login is allowed.	**true** / false
commands-allowed	boolean	Master switch for OPC UA commands. If false there will be no writeable OPC UA objects.	true / **false**
force-allowed	boolean	If true, the device accepts force commands via OPC UA.	true / **false**
reset-allowed	boolean	If true, the device accepts restart and factory reset commands via OPC UA.	true / **false**
config-allowed	boolean	If true, the device accepts configuration changes via OPC UA.	true / **false**

Table 32: OPC UA Configuration

All configuration elements are optional and do not need a specific order. Not every element is required to be sent. This means that only configuration changes will be taken over.

Optional: The configuration parameters of OPC UA can be set directly via the Web interface. It is possible to download the Web interface for sharing with other devices.

Response:

The resulting response is a JSON object with a status field. Status should be "0" if no error occurred and "-1" if there is an error.

In case of an error, the response contains an error array.

The error array contains an error object for each error occurred. The object consists of a field "Element" which names the config element which caused the error, and a field "Message" for the error message.

Examples:

```
{"status": -1, "error": [{"Element": "upcua-enable", "Message": "Boolean
 expected"}]}

{"status": 0}

{"status": -1, "error": [{"Element": "root", "Message": "Not a JSON
 object"}]}
```

11.2.2 OPC UA address space

OPC UA provides different services on the LioN-X devices with which a client can navigate through the hierarchy of the address space and read or write variables. In addition, the client can monitor up to 10 attributes from the address space for value changes.

A connection to an OPC UA server is established via the endpoint URL:

```
opc.tcp://[ip-address]:[port]
```

Various device data such as MAC address, device settings, diagnostics or status information can be read via *Identity objects*, *Config objects*, *Status objects* and *Process objects*.

Command objects can be read and written. This makes it possible, for example, to transfer new network parameters to the device, to use Force Mode or to reset the entire device to its factory settings.

11.2.3 OPC UA configuration - Quick start guide

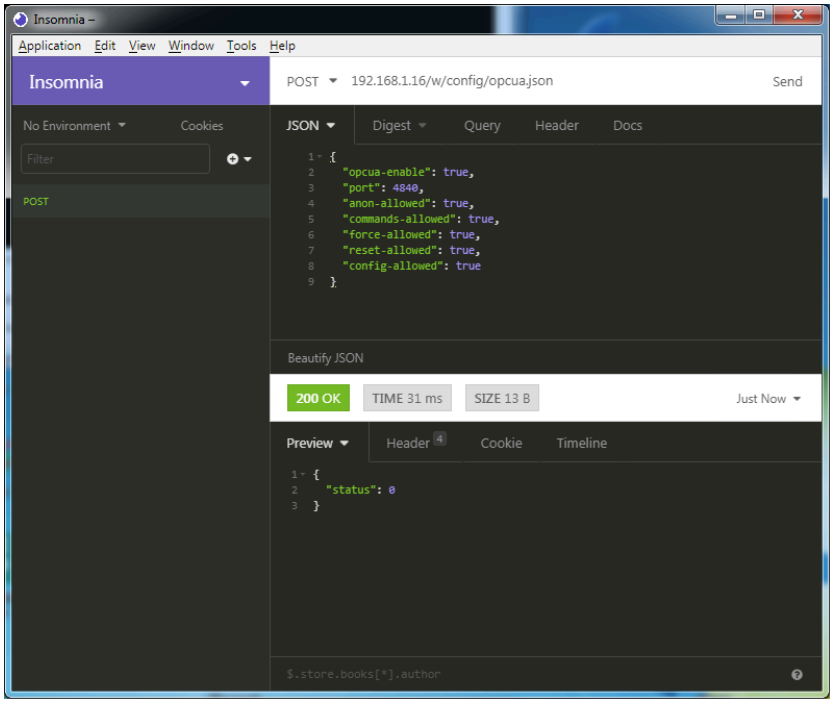
Attention: Belden Deutschland GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

11.2.3.1 OPC UA configuration via JSON

1. Depending on your application case, download and install *Insomnia* or a comparable application: https://insomnia.rest/download/

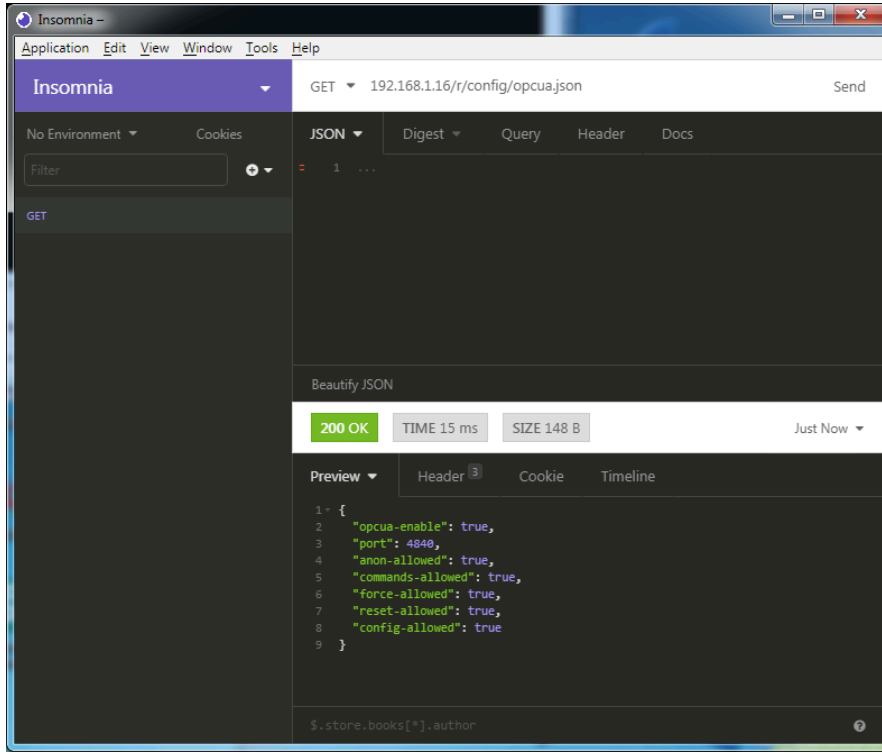
2. Configure OPC UA:

POST: `[IP-address]/w/config/opcua.json`



3. Read OPC UA:

GET: `[IP-address]/r/config/opcua.json`



11.3 REST API

The Representational State Transfer – Application Programming Interface (REST API) is a programmable interface which uses HTTP/HTTPS requests to GET and POST data. This enables the access to detailed device information.

For all LioN-X variants, the REST API can be used to read the device status. For the LioN-X multi-protocol variants, the REST API can also be used to write configuration and forcing data.

There are two different REST API standards you can use for the requests:

1. A standardized REST API that has been specified by the IO-Link Community and is described separately:

   `JSON_Integration_10222_V100_Mar20.pdf`

   Please download the file from https://catalog.belden.com or directly from io-link.com.

   Attention: Consider the following table to get an overview of the supported features of the IO-Link specification:

Feature		Supported
Gateway	GET /identification	Yes
	GET /capabilities	Yes
	GET /configuration	Yes
	POST /configuration	Yes
	POST /reset	Yes
	POST /reboot	Yes
	GET /events	Yes
Master	GET /masters	Yes
	GET /capabilities	Yes
	GET /identification	Yes
	POST /identification	Yes

Feature		Supported
Port	GET /ports	Yes
	GET /capabilities	Yes
	GET /status	Yes
	GET /configuration	Yes
	POST /configuration	Yes
	GET /datastorage	Yes
	POST /datastorage	Yes
Devices	GET /devices	Yes
	GET /capabilities	Yes
	GET /identification	Yes
	POST /identification	Yes
	GET /processdata/value	Yes
	GET /processdata/getdata/value	Yes
	GET /processdata/setdata/value	Yes
	POST /processdata/value	Yes
	GET /parameters	Not supported
	GET /parameters/{index}/subindices	Not supported
	GET /parameters/{parameterName}/subindices	Not supported
	GET /parameters/{index}/value	Not supported
	GET /parameters/{index}/subindices/{subindex}/value	Not supported
	GET /parameters/{parameterName}/value	Not supported
	GET /parameters/{parameterName}/subindices/{subParameterName}/value	Not supported
	POST /parameters/{index}/value	Not supported
	POST /parameters/{parameterName}/value	Not supported
	POST /parameters/{index}/subindices/{subindex}/value	Not supported
	POST /parameters/{parameterName}/subindices/{subParameterName}/value	Not supported
	POST /blockparametrization	Yes
	GET /events	Yes

Feature		Supported
IODD	GET /iodds	Not supported
	POST /iodds/file	Not supported
	DELETE /iodds	Not supported
	GET /iodds/file	Not supported

Table 33: Support of REST API features according to the IO-Link specification

2. A customized Belden REST API that is described in the following chapters.

11.3.1 Standard device information

Request method:	http GET
Request URL:	`<ip>/info.json`
Parameters	n.a.
Response format	JSON

The goal of the "Standard device information" request is to get a complete snapshot of the current device status. The format is JSON. For IO-Link devices, all ports with connected IO-Link device information are included.

11.3.2 Structure

Name	Data type	Description	Example
name	string	Device name	"0980 XSL 3912-121-007D-01F "
order-id	string	Ordering number	"935700002"
fw-version	string	Firmware version	"V.11.2.0.0 - 08.08.2024"
hw-version	string	Hardware version	"V.1.00"
mac	string	MAC address of the device	"3C B9 A6 F3 F6 05"
bus	number	0 = No connection 1 = Connection with PLC	1
failsafe	number	0 = Normal operation 1 = Outputs are in failsafe	0
ip	string	IP address of the device	
snMask	string	Subnet Mask	
gw	string	Default gateway	
rotarys	array of numbers (3)	Current position of the rotary switches: Array element 0 = x1 Array element 1 = x10 Array element 2 = x100	
ulPresent	boolean	True, if there is a UL voltage supply detected within valid range	
usVoltage_mv	number	US voltage supply in mV	
ulVoltage_mv	number	UL voltage supply in mV (only available for devices with UL supply)	
inputs	array of numbers (2)	Real state of digital inputs. **Element 0 = 1 Byte:** Port X1 Channel A to Port X4 Channel B **Element 0 = 1 Byte:** Port X5 Channel A to Port X8 Channel B	\[128,3\]
output	array of numbers (2)	Real State of digital outputs. **Element 0 =1 Byte:** Port X1 Channel A to port X4 Channel B **Element 0 = 1 Byte:** Port X5 Channel A to port X8 Channel B	\[55,8\]

Name	Data type	Description		Example
consuming	array of numbers (2)	Cyclic data from PLC to device		
producing	array of numbers (2)	Cyclic data from device to PLC		
diag	array of numbers (4)	Diagnostic information	**Element 0 = 1 Byte:** Bit 7: Internal module error (IME) Bit 6: Forcemode active Bit 3: Actuator short Bit 2: Sensor short Bit 1: U_L fault Bit 0: U_S fault	
			Element 1 = 1 Byte: Sensor short circuit ports X1 .. X8.	
			Element 2 =1 Byte: Actuator short circuit ports X1 Channel A to X4 Channel B	
			Element 3 = 1 Byte: Actuator short circuit ports X5 Channel A to X8 Channel B	
fieldbus	FIELDBUS Object			
FIELDBUS Object				
fieldbus_name	string	Currently used fieldbus		
state	number	Fieldbus state		
state_text	number	Textual representation of fieldbus state: 0 = Unknown 1 = Bus disconnected 2 = Preop 3 = Connected 4 = Error 5 = Stateless		
forcing	FORCING Object	Information about the forcing state of the device		

Name	Data type	Description	Example
channels	Array of CHANNEL (16)	Basic information about all input/output channels	
iol	IOL Object	Contains all IO-Link related information such as events, port states, device parameters.	
iol/diagGateway	array of DIAG	Array of currently active device/ gateway related events	
iol/diagMaster	array of DIAG	Array of currently active IOL-Master related events	
iol/ports	array of PORT (8)	Contains one element for each IO-Link port	
CHANNEL Object			
name	string	Name of channel	
type	number	Hardware channel type as number: 0 = DIO 1 = Input 2 = Output 3 = Input/Output 4 = IO-Link 5 = IOL AUX 6 = IOL AUX with DO 7 = IOL AUX with DO. Can be deactivated. 8 = Channel not available	
type_text	string	Textual representation of the channel type	
config	number	Current configuration of the channel: 0 = DIO 1 = Input 2 = Output 3 = IO-Link 4 = Deactivated 5 = IOL AUX	
config_text	string	Textual representation of the current config	
inputState	boolean	Input data (producing data) bit to the PLC	

Name	Data type	Description	Example
outputState	boolean	Output data bit to the physical output pin	
forced	boolean	True, if the output pin of this channel is forced	
simulated	boolean	True, if the input value to the PLC of this channel is simulated	
actuatorDiag	boolean	True, if the output is in short circuit / overload condition	
sensorDiag	boolean	True, if the sensor supply (Pin 1) is in short circuit / overload condition	
maxOutputCurrent_mA	number	Maximum output current of the output in mA	
current_mA	number	Measured current of the output in mA (if current measurement is available)	
voltage_mV	number	Measured voltage of this output in mV (if voltage measurement is available)	
PORT Object			
port_type	string	Textual representation of the IO-Link port type	
iolink_mode	number	**Current port mode:** 0 = Inactive 1 = Digital output 2= Digital input 3 = SIO 4 = IO-Link	
iolink_text	string	Textual representation of the current port mode	"Digital Input"
aux_mode	number	Indicates the configured mode for the Pin 2: 0 = No AUX 1 = AUX output (always on) 2 = Digital output (can be controlled by cyclic data) 3 = Digital input	
aux_text	string	Textual representation of the current aux mode	"AUX Output"
cq_mode	number	Port mode according to IOL specification	

Name	Data type	Description	Example
iq_mode	number	Pin2 mode according to IOL specification	
port_status	number	Port status according to IOL specification	
ds_fault	number	Data storage error number	
ds_fault_text	string	Textual data storage error.	
device	DEVICE Object	IO-Link device parameters. → Null if no IO-Link communication active	
diag	array of DIAG (n)	Array of port related events	
DIAG Object			
error	number	Error code	
source	string	Source of the current error.	"device" "master"
eventcode	number	Event code according to IO-Link specification	
eventqualifier	number	Event qualifier according to IO-Link specification	
message	string	Error message	"Supply Voltage fault"
DEVICE Object		Standard parameters of the IOL-Device	
device_id	number		
vendor_id	number		
serial	string		
baudrate	string	Baudrate (COM1,2,3)	
cycle_time	number	Cycle time in microseconds	
input_len	array of numbers (n)	IOL input length in bytes	
output_len	array of numbers (n)	IOL output length in bytes	
input_data	array of numbers (n)	IOL input data	
output_data	array of numbers (n)	IOL output data	

Name	Data type	Description	Example
pd_valid	number	"1", if IOL input data is valid	
pdout_valid	number	"1", if IOL output data is valid	
FORCING Object		Forcing information of the device	
forcingActive	boolean	Force mode is currently active	
forcingPossible	boolean	True, if forcing is possible and force mode can be activated	
ownForcing	boolean	True, if forcing is performed by REST API at the moment	
forcingClient	string	Current forcing client identifier	
digitalOutForced	array of numbers (2)	The force values of all 16 digital output channels.	
digitalOutMask	array of numbers (2)	The forcing mask of all 16 digital output channels.	
digitalInForced	array of numbers (2)	The force values of all 16 digital input channels.	
digitalInMask	array of numbers (2)	The forcing mask of all 16 digital input channels.	

11.3.3 Configuration and forcing

Method: POST

URL: `<ip>/w/force.json`

Parameters: None

Post-Body: JSON Object

Property	Data type	Example values	Description
forcemode	boolean	true / false	Forcing authority on/off
portmode	array (Port mode object)		
digital	array (Digital object)		
iol	array (IOL object)		

Table 34: Root object

Property	Data type	Example values	Remarks
port	integer	0..7	
channel	string	"a","b"	optional default is "a"
direction	string	"dio","di","do","iol", "off", "aux"	
aux	string	"dio","di","do","iol", "off", "aux"	IOL only, but optional
inlogica	string	"no","nc"	
inlogicb	string	"no","nc"	
inputlatch	bool	true / false	enable/disable input latch, optional
inputext	integer	Depends on the fieldbus: ▶ eip: 0 (off) - 255 (ms) ▶ ethercat: 0 (off) - 255 (ms) ▶ pns: 0 (off) - 255 (ms) ▶ cclink: 0 (off) - 255 (ms) ▶ mbtcp: 0 (off) - 255 (ms)	set input extension, optional
inputfilter	integer	0 .. 255	set input filter, optional

Table 35: Port mode object

Property	Data type	Example values	Remarks
port	integer	0..7	
channel	string	"a","b"	
force_dir	string	"phys_out","plc_in","clear"	optional default is "phys_out"
force_value	integer	0,1	

Table 36: Digital object

Property	Data type	Example values	Remarks
port	integer	0..7	
output	array[integer] or null to clear forcing	[55,88,120]	Output forcing
input	array[integer] or null to clear forcing	[20,0,88]	Input simulation to PLC

Table 37: IOL object

11.3.4 Reading and writing ISDU parameters

The *Indexed Service Data Unit* (ISDU) provides a highly flexible message format, which can contain single or multiple commands.

LioN-X IOL-Masters with IIoT support reading and writing ISDU parameters from connected IOL-Devices. It is possible to do this as a bulk transfer by reading and writing of multiple ISDU parameters via a single request.

11.3.4.1 Reading ISDU

Method: POST

URL: `<ip>/r/isdu.json`

Parameters: port (0-7)

Example:

```
192.168.1.20/r/isdu.json?port=5
```

Post-Body: JSON array of read ISDU object

Property	Data type	Example values	Remarks
ix	integer	0-INT16	Index to be read
subix	integer	0-INT8	Sub-index to be read

Table 38: Read ISDU object

Property	Data type	Example values	Remarks
status	integer	0, -1	0 = no error, -1= an error occured
message	string		Error Message if error occured
data	array (Read ISDU data object)		data, if no error occured. otherweise null

Table 39: Read ISDU response object

Property	Data type	Example values	Remarks
ix	integer	0-INT16	Index that was read
subix	integer	0-INT8	Sub-index that was read
status	integer	0, -1	0 = no error, -1= an error occured
eventcode	integer		IOL eventcode if status is -1
data	array[integer]		data, if no error occured. otherweise null

Table 40: Read ISDU data object

11.3.4.2 Writing ISDU

Method:	POST
URL:	`<ip>/w/isdu.json`
Parameters:	port (0-7)
Post-Body:	JSON array of write ISDU object

Property	Data type	Example values	Remarks
ix	integer	0-INT16	Index to be read
subix	integer	0-INT8	Sub-index to be read
data	array[integer]		Data to be written

Table 41: Write ISDU object

Response: Write ISDU response object

Property	Data type	Example values	Remarks
status	integer	0, -1	0 = no error, -1= an error occured
message	string		Error Message if error occured
data	array (Write ISDU data object)		data, if no error occured. otherweise null

Table 42: Write ISDU response object

Property	Data type	Example values	Remarks
ix	integer	0-INT16	Index that was written
subix	integer	0-INT8	Sub-index that was written
status	integer	0, -1	0 = no error, -1= an error occured
eventcode	integer		IOL eventcode if status is -1

Table 43: Write ISDU data object

11.3.5 Upload and process an IODD file

The REST API supports IODD file upload to the IO-Link Master.

Perform the following work steps:

1. Check file upload status

   Send request: GET file_upload

   Purpose: Get file upload status to check if there is another upload in progress.

   Expected response:

```
{
"status": 0,
"progress": 0,
"name": "",
"action": 0,
"upid": 0,
"errid": 0,
"errstr": "",
"pschr": 0
}
```

   Check the status ID. If status is '0', you can start a new iodd upload process. For reference, see tables Table 44: Status ID and meaning on page 120 and Table 45: Error ID and meaning on page 121. Proceed with the next step.
2. Initiate file upload

   Send request: POST file_upload

   Content-Type: application/json

   Purpose: Send details about the file to be uploaded.

   Expected response:

```
                        {"size":<total size>,"name":"<file name>",
"action":"iodd", "upid":
                       <upload id>}
                       {
                       "size": <total size>,
                       "name": "<file name>",
                       "action": "iodd",
                       "upid": <upload id>
                       }
```

The upload id (upid) is a number used by the backend to identify a specific upload and parsing process. It has to be used as a query parameter in the following steps.

The action will allways be iodd.

The size is the total size of the file in bytes.

The correct content type has to be set.

Attention: Remember the upload ID (upid) for subsequent steps

3. Upload file content

   Send request: POST file_upload?upid=<value> → Use the upid value from Step 2.

   Content-Type: application/octet-stream → The correct content type has to be set.

   Purpose: Send file or file chunks (max chunk size: 64KB).

   Attention: Sending file chunks bigger than 64KB will result in unresponsive behavior.

4. Monitor upload status

   Send request: GET file_upload?upid=<value> → Use the upid value from Step 2.

   Purpose: Get the current file upload status.

   Expected response:

```
{
"status": <status id value>,
"progress": <percentage>,
"name": "<file name given in step 2>",
"action": "iodd",
"upid": <upload id chosen in step 2>,
"errid": <error id>,
"errstr": "",
"pschr": <count of parsed characters>
}
```

   Repeat this step until the status becomes 'idle'. For some states this request triggers the necessary transitions in the internal state machine.

Only after the backend is sure that the correct client identified by its upid received the action finished or error state it will transition to the next one, the idle state.

The fields now show values depending on what was sent in step 2 and on the current process status.

Status ID	Status
0	File upload idle. New upload can be triggered.
1	File upload started.
2	File upload in progress.
3	File upload finished.
4	Error during file upload.
5	File upload timeout.
6	IODD parsing started.
7	IODD parsing finished.
8	IODD parsing error.
9	IODD parsing canceled.

Table 44: Status ID and meaning

ID	Error
0	No error.
1	Json parsing error.
2	Json type error.
4	Upload error.
5	File opening error.
6	File writing error.
7	Thread creating error.
8	Error during file copy.
9	Upload timeout.
10	Upload size exceeded.
11	Unknown action.
12	No upload id.
13	IODD paasing error.
14	Internal error.
15	IODD store full. Delete an IODD before uploading a new one.
16	Internal error.
17	IODD file CRC error.
18	Standard IODD file crc error.
19	No available space for parsing.

Table 45: Error ID and meaning

11.3.6 Example: Reading ISDU

ISDU read request

```
[
 {"ix":5,"subix":0},
 {"ix":18,"subix":0},
 {"ix":19,"subix":0},
 {"ix":20,"subix":0}
]
```

Response

```
{
 "message":"OK",
 "data":
[
 {"ix":5,"subix":0,"status":-1,"eventcode":32785},
 {"ix":18,"subix":0,"data":[79,68,83,49,48,76,49,46,56,47,76,65,54,44,50,
        48,48,45,77,49,50],"status":0},
 {"ix":19,"subix":0,"data":[53,48,49,50,57,53,51,53],"status":0},
 {"ix":20,"subix":0,"data":[100,105,115,116,97,110,99,101,32,115,101,110,
        115,111,114],"status":0}
],
"status":0}
```

11.3.7 Example: Writing ISDU

ISDU write request

```
[
 {"ix":24,"subix":0,"data":[97,98,99,100,101,102]},
 {"ix":9,"subix":0,"data":[97,97,97,97,97,98]}
]
```

Response

```
{
 "message":"OK",
 "data":[
 {"ix":24,"subix":0,"status":0},
 {"ix":9,"subix":0,"eventcode":32785,"status":-1}
],
"status":0}
```

11.4 CoAP server

The CoAP server functions are **only** applicable for the following LioN-X variants:

- 0980 XSL 3912-121-007D-00F-BD

The **Co**nstrained **A**pplication **P**rotocol (CoAP) is a specialized Internet application protocol for constrained networks such as lossy or low power networks. CoAP is useful especially in M2M (Machine to Machine) communication and can be used to translate simplified HTTP/HTTPS requests of low speed networks.

CoAP is based on the Server-Client principle and a service layer protocol that lets nodes and machines communicate with each other. The LioN-X multi-protocol variants provide CoAP server functionalities via a REST API interface over UDP.

11.4.1 CoAP configuration

In delivery state, CoAP functions are *disabled*. The CoAP server can be configured either using the Web interface or directly via a JSON object sent in an HTTP/HTTPS request. For more information see chapter CoAP configuration - Quick start guide on page 127.

The configuration URL is:

`http://[ip-address]/w/config/coapd.json`

The configuration can also read back as a JSON file:

`http://[ip-address]/r/config/coapd.json`

The configuration is a JSON object. Each JSON member is a configuration element. The object must not contain all elements. Only the provided elements will be changed. The configuration changes apply only after a device restart.

The following configuration elements are available (default values in bold):

Element	Data type	Description	Example data
enable	boolean	Master switch for the CoAP server	true / **false**
port	integer (0 to 65535)	Port of the CoAP server	**5683**

Table 46: CoAP configuration

CoAP response:

The resulting response is a JSON object with a "status" field. Status should be "0" if no error occurred, and "-1" if there is an error.

In case of an error, the response contains an error array.

The error array contains an error object for each error occurred. The object consists of a field "Element" which names the config element that caused the error, and of a field "Message" for the error message.

Examples:

```
{"status": -1, "error": [{"Element": "upcua-enable", "Message": "Boolean
 expected"}]}

{"status": 0}

{"status": -1, "error": [{"Element": "root", "Message": "Not a JSON
 object"}]}
```

11.4.2 REST API access via CoAP

A connection to the CoAP server running on the LioN-X multi-protocol variants can be established via the following URL:

`coap://[ip-address]:[port]/[api]`

For LioN-X, the following REST API Requests (JSON format) can be accessed via a CoAP endpoint:

Type	API	Note
GET	/r/status.lr	
GET	/r/system.lr	
GET	/info.json"	
GET	/r/config/net.json	
GET	/r/config/mqtt.json	
GET	/r/config/opcua.json	
GET	/r/config/coapd.json	
GET	/r/config/syslog.json	
GET	/contact.json	
GET	/fwup_status	
GET	/iolink/v1/gateway/identification	
GET	/iolink/v1/gateway/capabilities	
GET	/iolink/v1/gateway/configuration	
GET	/iolink/v1/gateway/events	
GET	/iolink/v1/masters	
GET	/iolink/v1/masters/1/capabilities	
GET	/iolink/v1/masters/1/identification	
GET	/iolink/v1/masters/1/ports	
GET	/iolink/v1/masters/1/ports/{port_number}/capabilities	This API is available for all 8 ports. {port_number} should be between "1" and "8".
GET	/iolink/v1/masters/1/ports/{port_number}/status	This API is available for all 8 ports. {port_number} should be between "1" and "8".
GET	/iolink/v1/masters/1/ports/{port_number}/configuration	This API is available for all 8 ports. {port_number} should be between "1" and "8".
GET	/iolink/v1/devices/master1port{port_number}/identification	This API is available for all 8 ports. {port_number} should be between "1" and "8".

Type	API	Note
GET	/iolink/v1/devices/master1port{port_number}/capabilities	This API is available for all 8 ports. {port_number} should be between "1" and "8".
GET	/iolink/v1/devices/master1port{port_number}/processdata/getdata/value	This API is available for all 8 ports. {port_number} should be between "1" and "8".
GET	/iolink/v1/devices/master1port{port_number}/events	This API is available for all 8 ports. {port_number} should be between "1" and "8".

Table 47: REST API access via CoAP

11.4.3 CoAP configuration - Quick start guide

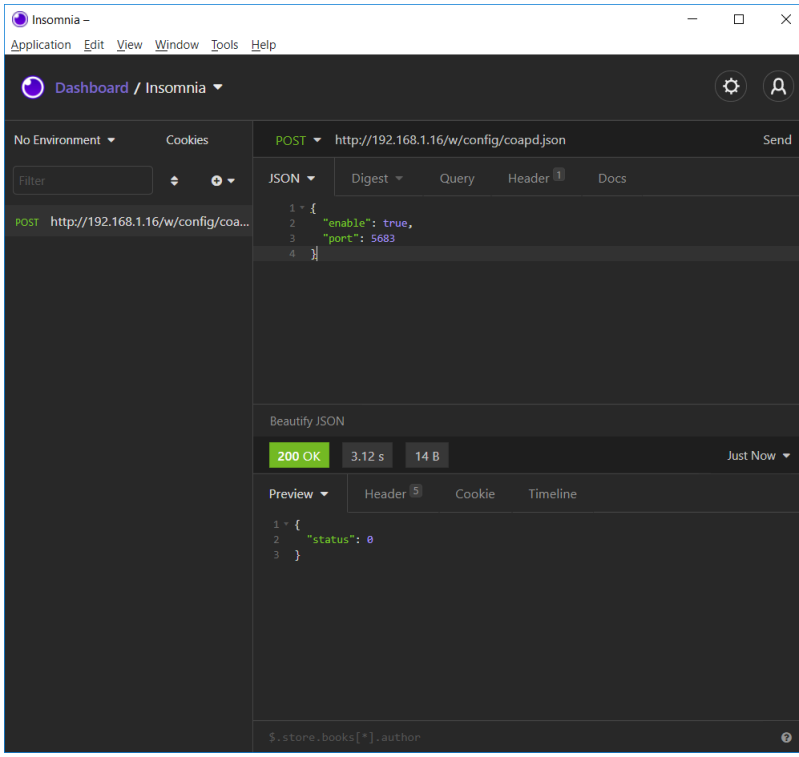
Attention: Belden Deutschland GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

11.4.3.1 CoAP configuration via JSON

1. Depending on your application case, download and install *Insomnia* or a comparable application: https://insomnia.rest/download/

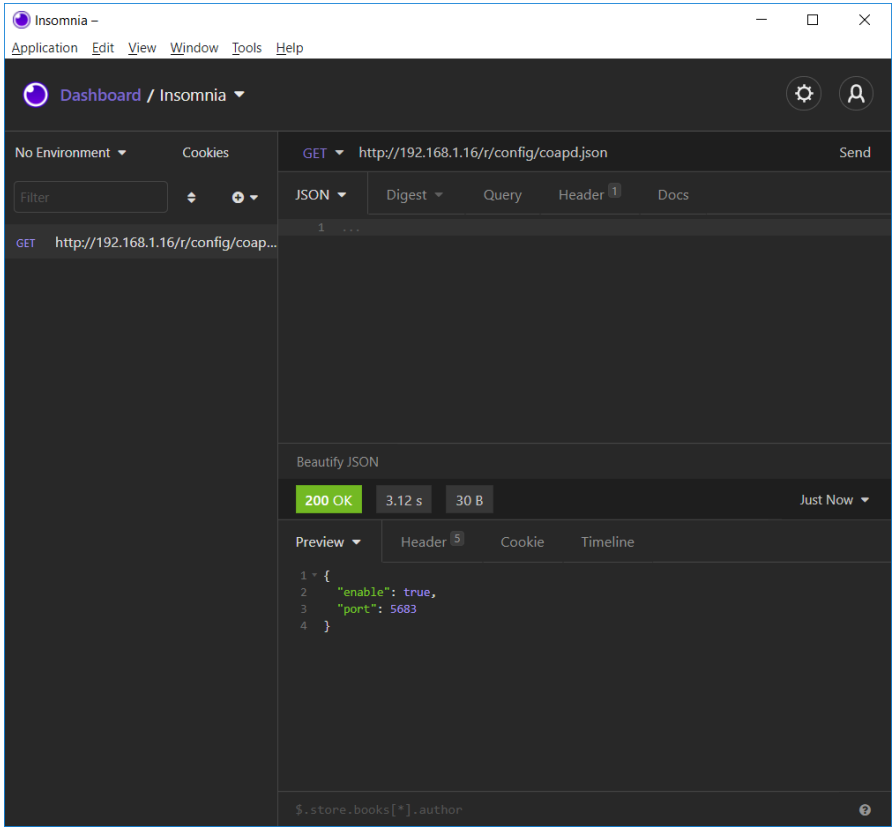
2. Configure CoAP:

POST: `[IP-address]/w/config/coapd.json`



3. Read CoAP configuration:

GET: `[IP-address]/r/config/coapd.json`



11.5 Syslog

Syslog functions are **only** applicable for the following LioN-X variants:

- 0980 XSL 3912-121-007D-00F-BD

The LioN-X multi-protocol variants provide a Syslog client which can connect with a configured Syslog server and is able to log messages.

Syslog is a platform-independent standard for logging messages. Each message contains a timestamp as well as information about the severity level and the subsystem. The Syslog protocol RFC5424 is based on the Server-Client principle and lets machines and devices send messages in the network and collect them centrally. (For more details on the used syslog standard, please refer to https://datatracker.ietf.org/doc/html/rfc5424.)

LioN-X supports the storage of 256 messages in a ring buffer which are sent to the configured Syslog server. When the ring is full with 256 messages, the oldest message is always replaced by the newly arriving messages. All messages can be saved on the Syslog server. The Syslog client of the IO-Link Master will not store any message permanently.

11.5.1 Syslog configuration

In **delivery state**, Syslog functions are **disabled**. The Syslog client can be configured either using the Web interface or directly via a JSON object sent in an HTTP/HTTPS request. For more information see chapter Syslog configuration - Quick start guide on page 132.

The configuration URL is:

`http://[ip-address]/w/config/syslog.json`

The configuration can also read back as a JSON file:

`http://[ip-address]/r/config/syslog.json`

The configuration is a JSON object. Each JSON member is a configuration element. The object must not contain all elements. Only the provided elements will be changed. The configuration changes apply only after a device restart.

The following configuration elements are available (default values in bold):

Element	Data type	Description	Example data
syslog-enable	boolean	Master switch for the Syslog client	true / **false**
global-severity	integer	Severity level of Syslog client 0 – Emergency 1 – Alert 2 – Critical **3 – Error** 4 – Warning 5 – Notice 6 – Info 7 – Debug The client will log all messages of severity according to the setting, including all below levels.	0/1/2/**3**/4/5/6/7
server-address	string (IP address)	IP address of the Syslog server	192.168.0.51 (Default: **null**)
server-port	integer (0 to 65535)	Server port of the Syslog server	**514**
server-severity	integer (0 to 7)	Severity level of Syslog server 0 – Emergency 1 – Alert 2 – Critical **3 – Error** 4 – Warning 5 – Notice 6 – Info 7 – Debug	0/1/2/**3**/4/5/6/7

Table 48: Syslog configuration

Syslog response:

The resulting response is a JSON object with a "status" field. Status should be "0" if no error occurred, and "-1" if there is an error.

In case of an error, the response contains an error array.

The error array contains an error object for each error occurred. The object consists of a field "Element" which names the config element that caused the error, and of a field "Message" for the error message.

Examples:

```
{"status": -1, "error": [{"Element": "upcua-enable", "Message": "Boolean
 expected"}]}

{"status": 0}

{"status": -1, "error": [{"Element": "root", "Message": "Not a JSON
 object"}]}
```

11.5.2 Syslog configuration - Quick start guide

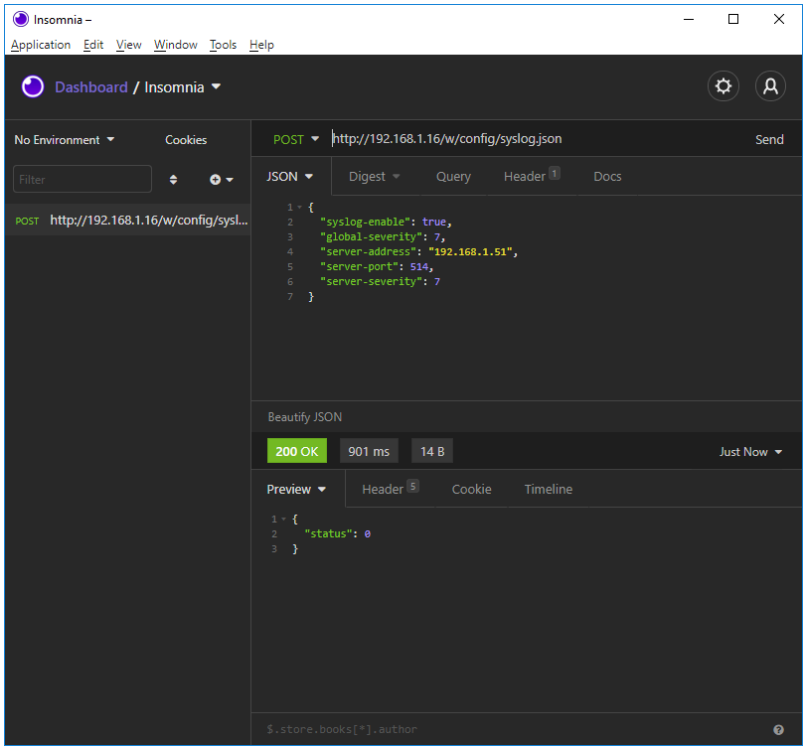
Attention: Belden Deutschland GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

11.5.2.1 Syslog configuration via JSON

1. Depending on your application case, download and install *Insomnia* or a comparable application: https://insomnia.rest/download/

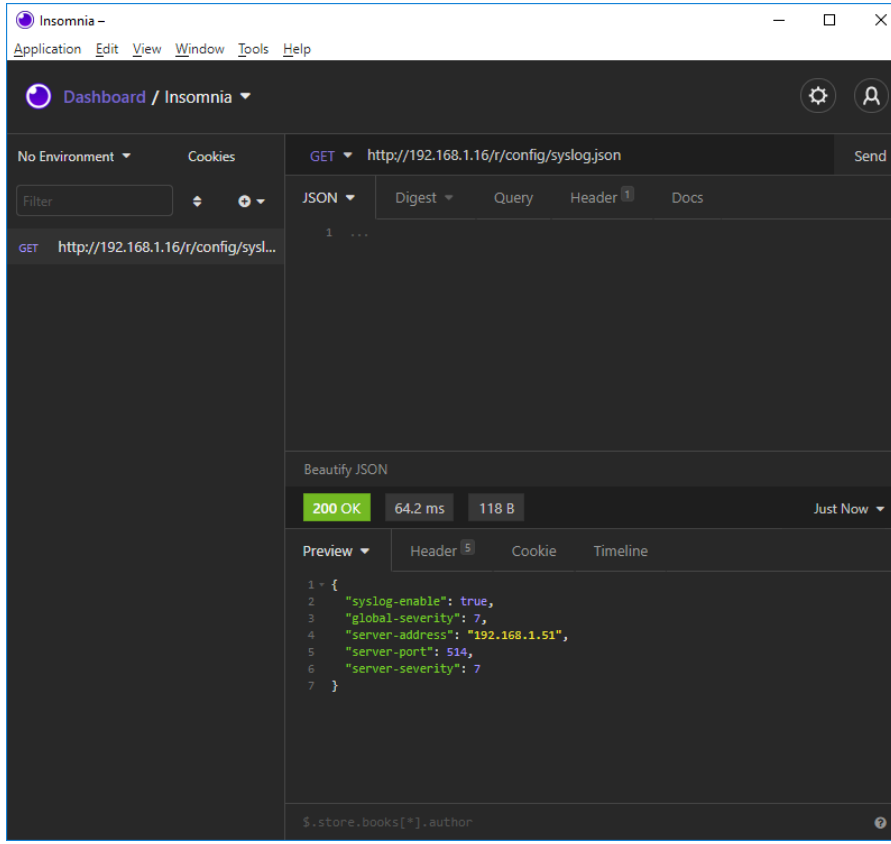
2. Configure Syslog:

POST: `[IP-address]/w/config/syslog.json`



3. Read Syslog configuration:

GET: `[IP-address]/r/config/syslog.json`



11.6 Network Time Protocol (NTP)

The NTP function is **only** applicable for the following LioN-X variant:

- 0980 XSL 3912-121-007D-00F-BD

The LioN-X multi-protocol variants provide an NTP client (version 3) which can connect with a configured NTP server and is able to synchronize the network time at a configurable interval.

NTP is a network protocol which uses UDP datagrams to send and receive timestamps in order to synchronize with a local clock. The NTP protocol RFC1305 is based on the Server-Client principle and exclusively supplies the synchronization with Coordinated Universal Time (UTC). (For more details on the used NTP standard, please refer to https://datatracker.ietf.org/doc/html/rfc1305.)

11.6.1 NTP configuration

In **delivery state**, the NTP client is **disabled**. The NTP client can be configured either using the Web interface or directly via a JSON object sent in an HTTP/HTTPS request. For more information see chapter NTP configuration - Quick start guide on page 136.

The configuration URL is:

`http://[ip-address]/w/config/ntpc.json`

The configuration can also read back as a JSON file:

`http://[ip-address]/r/config/ntpc.json`

The configuration is a JSON object. Each JSON member is a configuration element. The object must not contain all elements. Only the provided elements will be changed. The configuration changes apply only after a device restart.

The following configuration elements are available (default values in bold):

Element	Data type	Description	Example data
NTP client state	boolean	Master switch for the NTP client	true / **false**
Server address	string	IP address of the NTP server	192.168.1.50
Server port	integer	Port of the NTP server	**123**
Update interval	integer	Interval at which the client will connect with the configured NTP server (see table row "Server address"). **Note: This value is in seconds.**	1/2/10/**60**

Table 49: NTP configuration

NTP response:

The resulting response is a JSON object with a "status" field. Status should be "0" if no error occurred, and "-1" if there is an error.

In case of an error, the response contains an error array.

The error array contains an error object for each error occurred. The object consists of a field "Element" which names the configuration element that caused the error, and of a field "Message" for the error message.

Examples:

```
{"status": -1, "error": [{"Element": "ntpc-enable", "Message": "Boolean
 expected"}]}

{"status": 0}

{"status": -1, "error": [{"Element": "root", "Message": "Not a JSON
 object"}]}
```

11.6.2 NTP configuration - Quick start guide

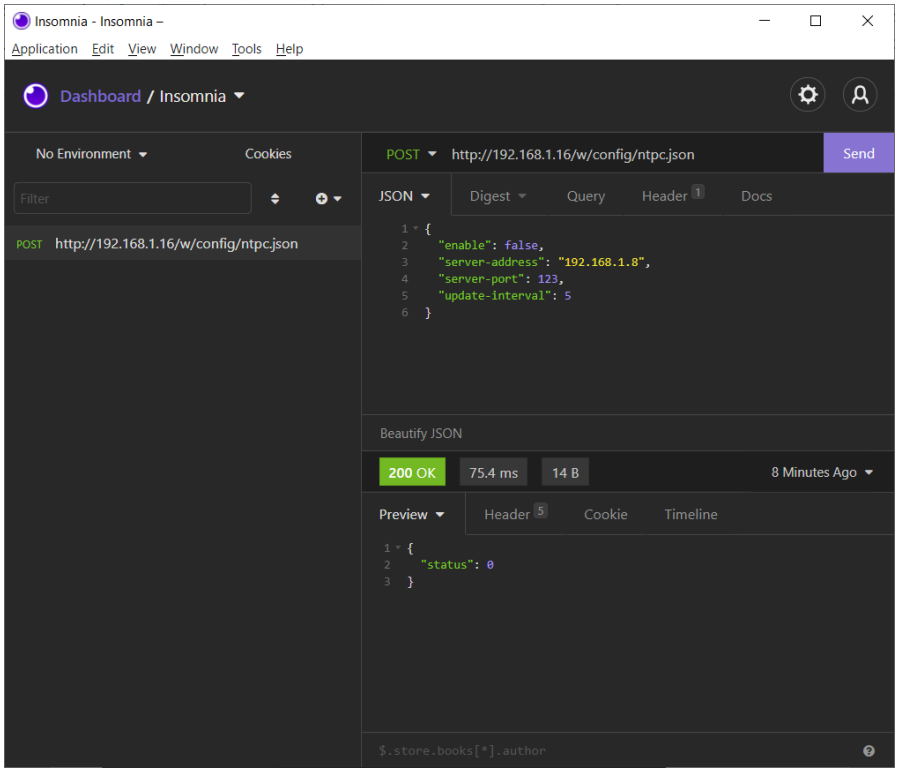
Attention: Belden Deutschland GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

11.6.2.1 NTP configuration via JSON

1. Depending on your application case, download and install *Insomnia* or a comparable application: https://insomnia.rest/download/

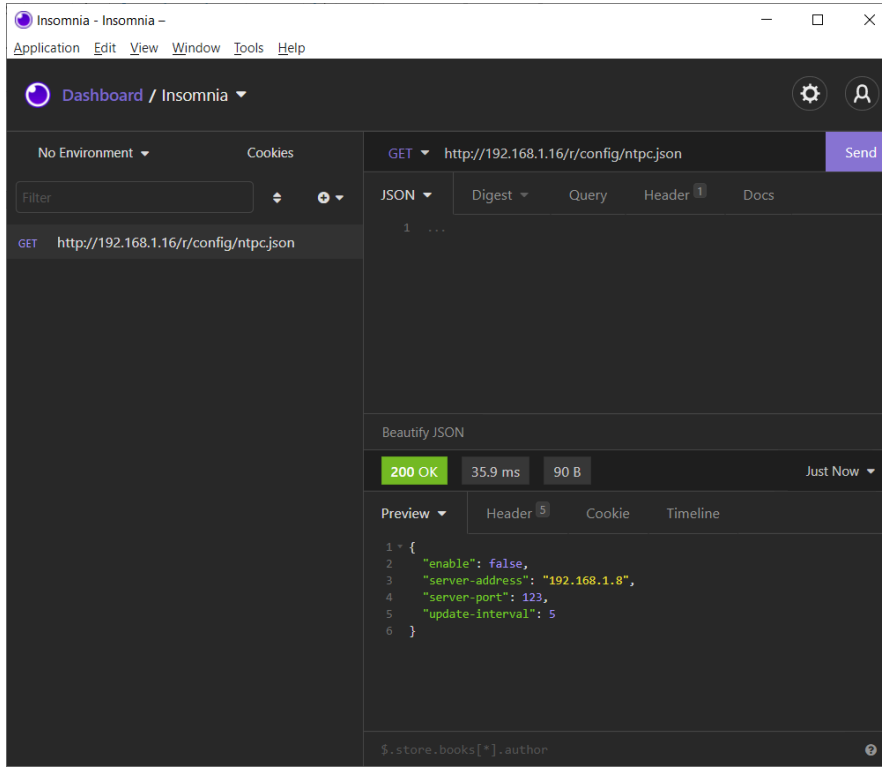
2. Configure NTP:

POST: `[IP-address]/w/config/ntpc.json`



3. Read NTP configuration:

GET: `[IP-address]/r/config/ntpc.json`



12 The integrated Web server

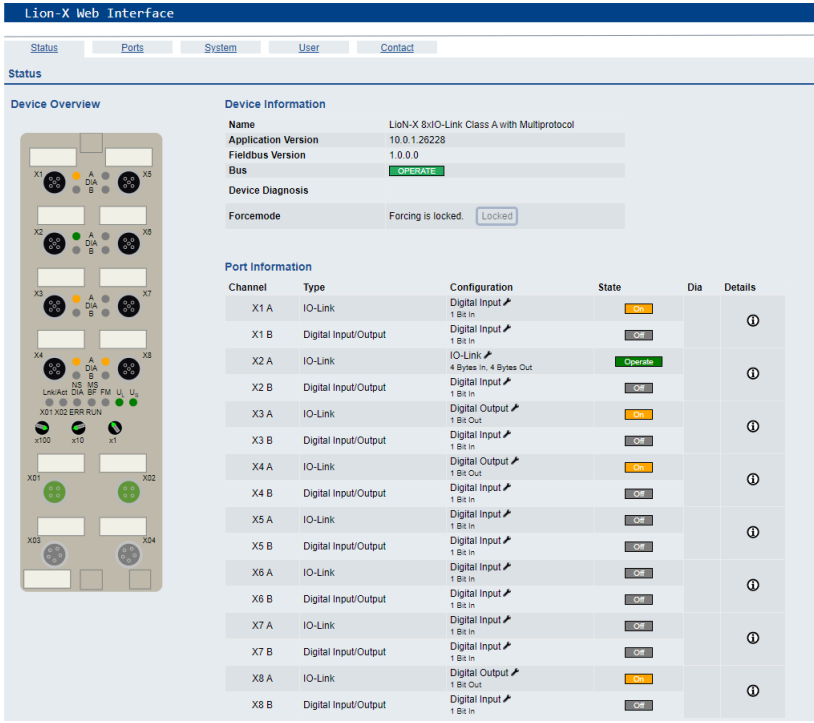
All device variants are equipped with an integrated Web server which makes functions for the device configuration and the display of status and diagnostic information available via a Web interface.

The Web interface provides an overview of the configuration and status of the device. It is also possible to use the Web interface to trigger a reboot, reset to the factory defaults, or perform a firmware update.

Enter "http://" or "https://" followed by the IP address, such as "http://192.168.1.5", in your Web browser’s address bar. If the status page of the device is not displayed, check your browser and firewall settings.

12.1 LioN-X 0980 XSL... variants

12.1.1 The Status page



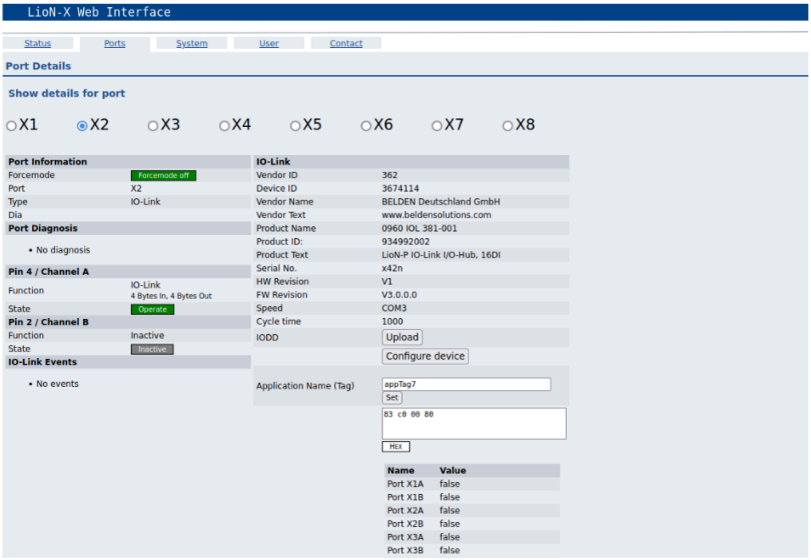
The status page provides a quick overview of the current state of the device.

The left side shows a graphical representation of the module with all its LEDs and the positions of the rotary encoding switches.

The right side shows the "Device Information" table with some basic data for the module; for example, the variant, the cyclic communication status and a diagnostic indicator. The indicator shows whether diagnostics for the module exist.

The "Port Information" table shows the configuration and state of the I/O ports.

12.1.2 The Ports page



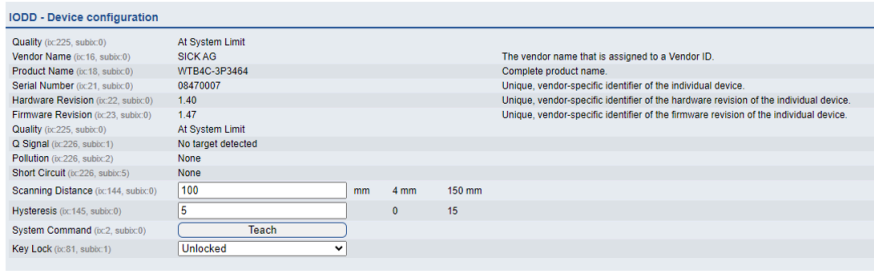
The page shows detailed port information. In the field **Port Diagnosis**, incoming and outgoing diagnostics are displayed as clear text. **Pin 2** and **Pin 4** contain information about the configuration and state of the port. For IO-Link ports, additional information relating to the connected sensor and the process data is displayed.

12.1.2.1 IODD upload

The button **UPLOAD** allows uploading an IODD file into the module, regardless of the device for which the IODD is designed.

The maximum number of IODDs is limited due to storage space. If there is no more space for a new IODD left, there will be a message of the detected error.

With the help of the IODD management page ("System" page), not used IODDs can be deleted. If there is already a matching IODD stored in the system for the connected IO-Link device, the button **CONFIGURE** is shown. By clicking this button, the page "IODD - Device configuration" will open, where the IO-Link device can be configured.



IODD - Device configuration

Quality (ix:225, subix:0)	At System Limit				
Vendor Name (ix:16, subix:0)	SICK AG				The vendor name that is assigned to a Vendor ID.
Product Name (ix:18, subix:0)	WTB4C-3P3464				Complete product name.
Serial Number (ix:21, subix:0)	08470007				Unique, vendor-specific identifier of the individual device.
Hardware Revision (ix:22, subix:0)	1.40				Unique, vendor-specific identifier of the hardware revision of the individual device.
Firmware Revision (ix:23, subix:0)	1.47				Unique, vendor-specific identifier of the firmware revision of the individual device.
Quality (ix:225, subix:0)	At System Limit				
Q Signal (ix:226, subix:1)	No target detected				
Pollution (ix:226, subix:2)	None				
Short Circuit (ix:226, subix:5)	None				
Scanning Distance (ix:144, subix:0)	100	mm	4 mm	150 mm	
Hysteresis (ix:145, subix:0)	5		0	15	
System Command (ix:2, subix:0)	Teach				
Key Lock (ix:81, subix:1)	Unlocked				

12.1.3 The System page

The System page shows the basic information for the module like Firmware version, Device information, Ethernet, Network and Fieldbus information.

IP Settings

Use this parameter to change the current IP address of the module.

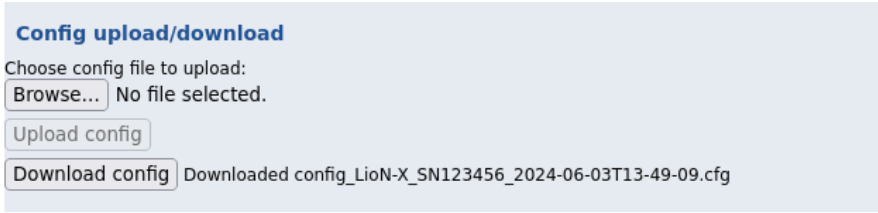
For PROFINET, this is only useful during commissioning. Normally, the PLC sets the IP address at start-up by detecting the PROFINET module via its device name.

12.1.3.1 License

This button opens a new window with Open Source Software information used in this product.

12.1.3.2 Config upload/download

With this feature, settings configured via the Web interface can be stored outside the I/O-Device (Download) for later Upload, e.g. after an I/O-Device change.



The following settings will be stored inside this file:

Scope	Type	Setting	Options	Details
Gateway		deviceID		To check device identity.
	iol	applicationSpecificTag		
	iol	functionTag		
	iol	locationTag		
		forcing		Enable/disable forcing
		channel_count		
		network configuration	ip	
			snMask	
			gw	
			source	1 - manual 2 - dhcp 3 - rotary 4 - dcp
Channel		index		channel index starting from 0
		channel configuration		0 - DIO 1 - IN 2 - OUT 3 - IOL 4 - AUX 5 - SAFIN 6 - SAFOUT
	iol	forced		
	iol	simulated		
	iol	force values		array

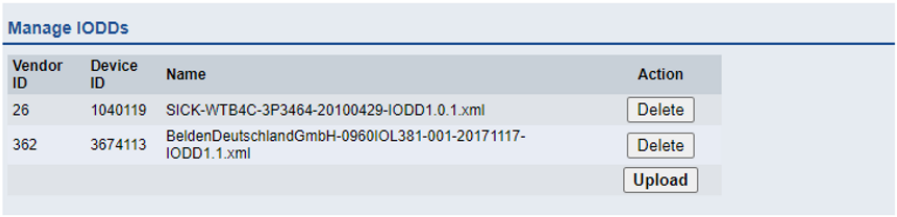
Scope	Type	Setting	Options	Details
	iol	simulated		
	iol	sim values		array
	iol	validation	option	validation and backup
			vendorId	
			deviceId	
	digital	force		
	digital	force value		
	digital	simulate		
	digital	sim value		
	digital	inputPolarity		
	digital	autorestart mode		
	digital	inputFilter100us		
	digital	currentLimit		
	digital	outputRestartMode		
	digital	failsafeMode		
	digital	surveillanceTimeouMs		
OPC UA		opcua	opcua-enable	
			port	
			anon-allowed	
			commands-allowed	
			force-allowed	
			reset-allowed	
			config-allowed	

Scope	Type	Setting	Options	Details
	digital		dcu-allowed	
MQTT		mqtt	mqtt-enable	
			broker	
			login	
			password	
			port	
			base-topic	
			will-enable	
			will-topic	
			auto-publish	
			publish-interval	
			publish-identity	
			publish-config	
			publish-status	
			publish-process	
	iol		publish-devices	
			commands-allowed	
			force-allowed	
			reset-allowed	
			config-allowed	
			qos	
SYSLOG		syslog	syslog-enable	
			global-severity	

Scope	Type	Setting	Options	Details
			server-address	
			server-port	
			server-severity	
COAP		coap	enable	
			port	
NTP		ntpc	enable	
			server-address	
			server-port	
			update-interval	

12.1.3.3 IODD

The button **Manage IODDs** opens a new page for the IODD management on the I/O-Device. IODDs can be uploaded or deleted on this page, and all uploaded IODDs are listed here. For configuring connected IO-Link devices, open the related "Ports" page.



12.1.3.4 Restart device

The module initializes a software reset.

12.1.3.5 Reset configuration to factory defaults

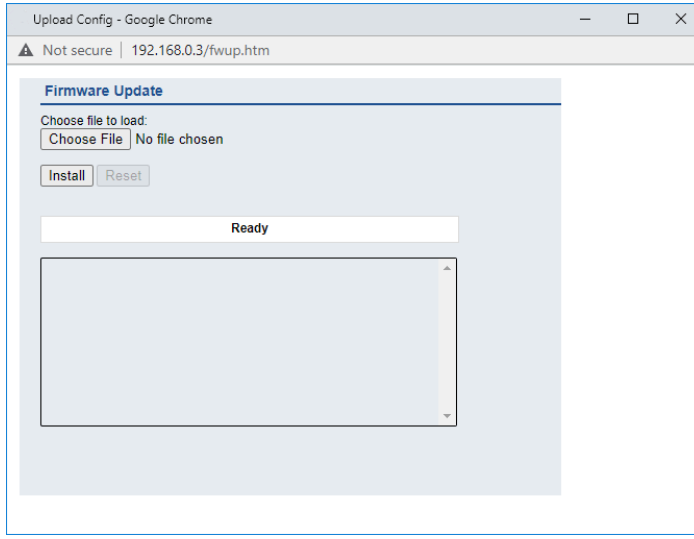
The module restores to the default factory settings.

12.1.3.6 Firmware update

The module initializes a Firmware update.

Attention: Firmware updates may only be performed in a non-production environment! The I/O data is not updated during the firmware update. As a minimum requirement, the controller must be in operating state “Stop”, or the connection to the controller must be disconnected before the update. The device automatically performs a reset during the update.

For a firmware update choose the **.ZIP* container available on our website or ask our support team. Afterwards follow the instructions shown on your screen.



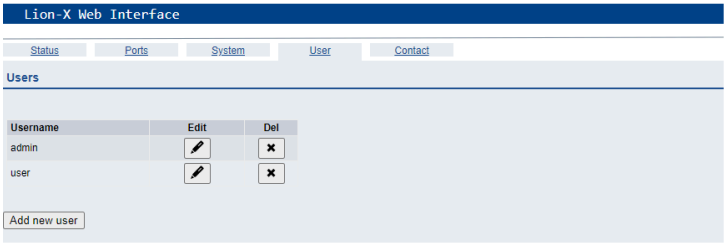
12.1.3.7 System diagnosis

All Syslog messages will be displayed in a ring buffer with 256 entries. With the 'Store timer', the nonvolatile storing interval can be set.

The default value is 'Off' (no nonvolatile storage of system diagnosis ring buffer).

The configuration is activated by clicking the button *Set configuration*.

12.1.4 The User page



The User page provides the user management of the Web interface. New users with access rights "Admin" or "Write" can be added here. For security reasons please change the default admin password immediately after configuring the device.

Default user login data:

- User: admin
- Password: private

13 IODD

IODD functions are **only** applicable for the following device variants:

- 0980 XSL 3912-121-007D-00F-BD

The **IO D**evice **D**escription (IODD) is a set of files formally describing an IO-Link Device. The IODD is created by the vendor and is mandatory for each IO-Link Device.

Belden LioN-X IO-Link Masters with the "IODD on Module" functionality are ready to use IODDs in order to make the IO-Link Device configuration much more easier and the process data human readable in a better way. IODDs can be uploaded via the Web Interface and remanently stored on the IO-Link Master afterwards.

If a corresponding IO-Link Device is connected, the stored IODD is used to provide a user friendly configuration page, where all parameters of the device can be viewed and edited. Additionally, according to the IODD, the process data will also be formatted and displayed to the user.

13.1 IO-Link Device parameters and ISDU requests

Every IO-Link Device provides parameters that can be read and written via the special IO-Link service ISDU (**I**ndexed **S**ervice **D**ata **U**nit).

Every parameter is addressed by an index. Sub-indices are possible but optional. Some parameters (most of them read-only) are mandatory for IO-Link devices an can be found always on the same indices (See *Table B.8* in the *IO-Link Interface and System Specification*: https://io-link.com/share/Downloads/Package-2020/IOL-Interface-Spec_10002_V113_Jun19.pdf).

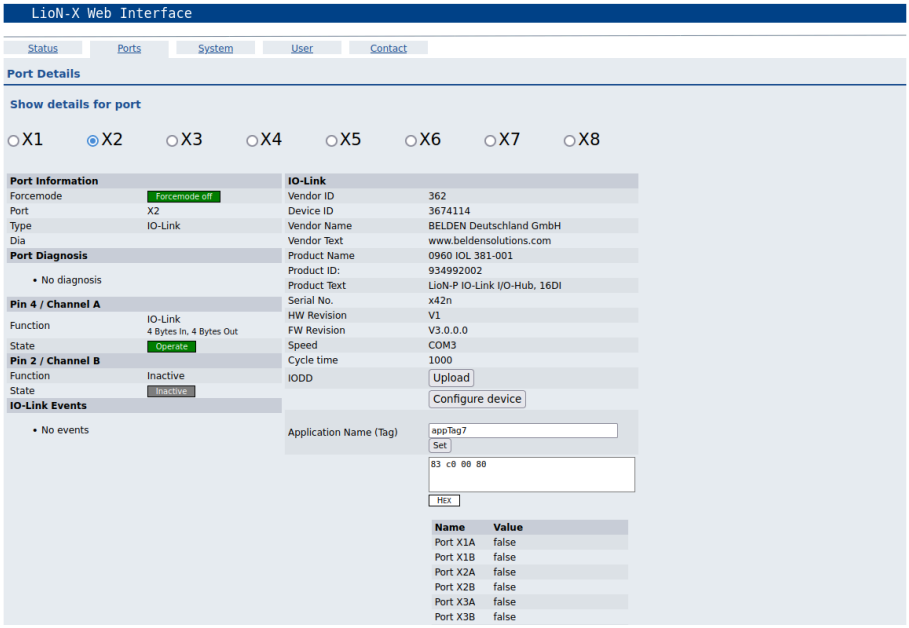
A vendor can use additional parameters and therefore more indices for their devices in order to provide additional configuration options. These vendor specific parameters can be described in an IODD. The "IODD on Module" feature of the LioN-X IO-Link Masters can read and parse this information out of an IODD and use it to provide the user viewing and editing options

for vendor specific parameters without any additional knowledge about the vendor specific device features.

13.2 Web GUI functionality

All of the "IODD on Module" features are accessible via the LioN-X Web interface.

13.2.1 Port Details page



The Port Details Page shows all information about the selected port. In the left column, all port and channel specific information is displayed. If the port is configured as IO-Link and there is an IO-Link Device connected, all IO-Link information for the connected device is displayed in the right column.

IODD buttons

The row called *IODD* provides access to the "IODD on Module" features. The button *UPLOAD* will let the user upload an IODD file into the module, regardless of the original device the IODD has been designed for.

The maximum number of IODDs is limited due to storage space. If there is no more space left for new IODDs, there will be an error message. In this case, navigate to the IODD Management page to delete IODDs which are no longer used.

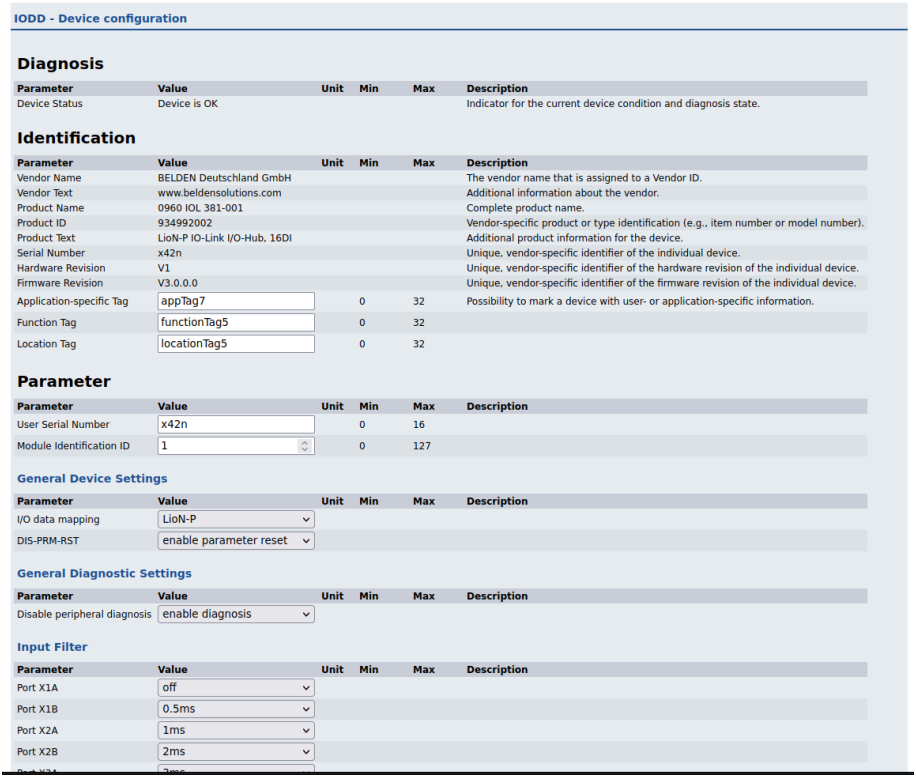
If there is a matching IODD for the currently connected device already stored in the system, the button *CONFIGURE* is shown in the interface. By clicking this button, the Parameter Page will open to configure the device.

Process data

For every connected IO-Link Device, raw process data for input and output direction (set of bytes) is on display.

If a matching IODD providing information about process data is already stored in the system, this data will also be displayed in a user-friendly format according to the IODD.

13.2.2 Parameters page



The parameters page "IODD – Device configuration" shows all parameters which are provided by the IODD of the device. That means the parameter set is variable and depends on the connected IO-Link Device.

The stored IODD reads the parameter meta data, such as names, units, min/max values, descriptions etc. The values will be obtained directly from the connected device. For that reason it may take several seconds until the page is updated.

If not already saved into the browser, you will be asked for your credentials to continue. A valid user access with Web Interface group membership is needed in order to edit the device parameters. After the registration, enabled values can be changed. Disabled values cannot be changed and may be

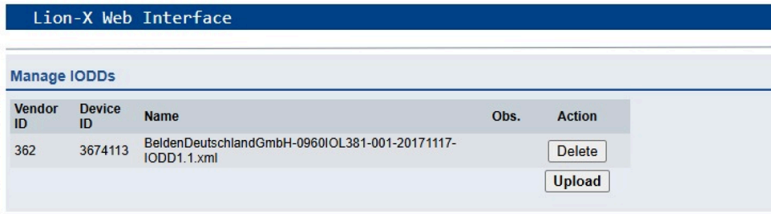
marked as "read-only" in the IODD. All values are directly written back to the device after any change.

Limitations

- Editing parameter values will directly change them inside the connected device. No parameter server action is triggered by that.
- There is a maximum size of the IODD in order to be uploaded into the system. This depends on several values, such as file size, parameter count, nesting levels etc.

13.2.3 IODD Management page

The IODD Management Page can be accessed via the System page displaying all IODDs that are currently stored in the system. All IODDs matching connected devices are marked. On the IODD Management page, you can manually delete any IODD in the system.



14 Technical data

The following sections give an overview of the most important functional data needed to operate the device. For further information and detailed technical data, see the respective **Data Sheet** of your required product in the product specific download area on https://catalog.belden.com.

14.1 General

Protection class (Only applies if the connectors are screwed together or if protective caps are used.)[3]	IP65 IP67 IP69K	
Ambient temperature (during operation and storage)	0980 XSL 3x12...	-40 °C .. +70 °C (-40 °F .. +158 °F)
Weight	LioN-X 60 mm	approx. 500 gr. (17.6 oz)
Ambient moisture	Max. 98% RH (For UL applications: Max. 80% RH)	
Housing material	Die-cast zinc	
Surface finish	Frosted nickel	
Flammability class	UL 94 (IEC 61010)	
Vibration resistance (oscillation) DIN EN 60068-2-6 (2008-11)	15 g / 5-500 Hz	
Shock resistance DIN EN 60068-2-27 (2010-02)	50 g / 11 ms +/- X,Y,Z	
Fastening torques	M4 fixing screws	1 Nm
	M4 ground connection	1 Nm
	M12 connector	0.5 Nm
Permitted cables	Ethernet cables according to IEEE 802.3, min. CAT 5 (shielded) Max. length of 100 m, not routed out of facility (= local network)	

Table 50: General information

3 Not under UL investigation.

14.2 Modbus TCP protocol

Protocol	Modbus TCP
Update cycle	1 ms
Transmission rate	100 Mbit/s, full duplex
Transmission procedure Autonegotiation	100BASE-TX supported
Product type	Modbus TCP server
Product code	0980 XSL 3912-121-007D-00F-BD, 935720001
Supported Ethernet protocols	Ping ARP- HTTP TCP/IP DHCP/BOOTP
Switch functionality	Integrated
Modbus TCP interface	2 M12 sockets, 4-pin, D-coded (see pin assignments)
Connections	2 M12 Hybrid male/female, 8-pin
Autocrossing	supported
Electrically isolated Ethernet ports -> FE	2000 V DC

Table 51: Modbus TCP protocol

14.3 Power supply of the module electronics/ sensors

Port X03, X04	M12-L-coded Power, connector/socket, 5-pole Pin 1 / Pin 3		
Nominal voltage U_S	24 V DC (SELV/PELV)		
Current U_S	Max. 16 A		
Voltage range	21 .. 30 V DC		
Power consumption of module electronics	Typically 160 mA (+/-20 % at U_S nominal voltage)		
Power supply interruption	Max. 10 ms		
Voltage ripple U_S	Max. 5 %		
Current consumption sensor system (L+ / Pin 1)	0980 XSL 3x12...	Port X1 .. X8 (Pin 1)	max. 4 A per port (at $T_{ambient}$ = 30° C)
Voltage level of the sensor power supply	Min. (U_S – 1.5 V)		
Short circuit/overload protection of sensor supply	Yes, per port		
Reverse polarity protection	Yes		
Operational indicator (U_S)	LED green:	18 V (+/- 1 V) < U_S	
	LED red:	U_S < 18 V (+/- 1 V)	

Table 52: Information on the power supply of the module electronics/ sensors

14.4 Power supply of the actuators

14.4.1 IO-Link Class A devices (U_L)

Nominal voltage U_L	24 V DC (SELV/PELV)
Voltage range	18 .. 30 V DC
Current U_L	Max. 16 A
Voltage ripple U_L	Max. 5 %
Reverse polarity protection	Yes
Operational indicator (U_L)	LED green: 18 V (+/- 1 V) < U_L LED red: U_L < 18 V (+/- 1 V) or U_L > 30 V (+/- 1 V) * if "Report U_L supply voltage fault" is enabled.
Port X03, X04	M12_L-coded Power, connector/socket, 5-pole Pin 2 / Pin 4

Table 53: Information on the power supply of the actuators

14.5 I/O ports Channel A (Pin 4)

0980 XSL 3x12...	Port X1 .. X8	Class A	IOL, DI, DO	M12 socket, 5-pin, Pin 4

Table 54: IO-Link Master ports: Functional overview for Ch. A (Pin 4)

14.5.1 Configured as digital input, Ch. A (Pin 4)

Input connection	0980 XSL 3x12...		Type 1 as per IEC 61131-2
Nominal input voltage	24 V DC		
Input current	Typically 3 mA		
Channel type	Normally open, p-switching		
Number of digital inputs	0980 XSL 3x12...	X1 .. X8	8
Status indicator	yellow LED		
Diagnostic indicator	red LED per channel		

Table 55: I/O ports Ch. A (Pin 4) configured as digital inputs

14.5.2 Configured as digital output, Ch. A (Pin 4)

Attention: For variant 0980 XSL 3912-121-007D-00F-BD,, the digital outputs of channel A are **supplied by the U_L power** when parameterized to "High-Side Switch" mode.

Output type	normally open, p-switching (parameterized to "High-Side Switch" mode)	
Nominal output voltage per channel Signal status “1” Signal status “0”	 min. (U_S -1 V) **or** min. (U_L -1 V) depending on the device variant max. 2 V	
Max. output current per device	0980 XSL 3x12...	9 A (power supplied via U_L)
Max. output current per channel[4]	0980 XSL 3x12... (X1 .. X8)	2 A (power supplied via U_S)
Short-circuit/overload protected	yes/yes	
Behavior in case of short circuit or overload	deactivation with automatic power-on (parameterized)	
Number of digital outputs	0980 XSL 3x12... (X1 .. X8)	8
Status indicator	yellow LED per output	
Diagnostic indicator	red LED per channel	

Table 56: I/O ports Ch. A (Pin 4) configured as digital outputs

4 Max. 2.0 A per channel; for every port pair X1/X2, X3/X4, X5/X6, X7/X8 max. 6.5 A (for **UL applications** max. 5.0 A); for the whole port group X1 .. X8 max. 9.0 A in total (with derating).

14.5.3 Configured as IO-Link port in COM mode, Ch. A

IO-Link Master specification	v1.1.3, IEC 61131-9	
Communication rates	4.8 kbaud (COM 1) 38.4 kbaud (COM 2) 230.4 kbaud (COM 3)	
Line lengths in the IO-Link Device	max. 20 m	
Number of IO-Link ports	0980 XSL 3x12... (X1 .. X8)	8
Min. IO-Link cycle time	400 µs	

Table 57: Configured as IO-Link port in COM mode

14.6 I/O ports Channel B (Pin 2)

0980 XSL 3x12...	Port X1 .. X8	Class A	DI, DO	M12 socket, 5-pin, Pin 2

Table 58: IO-Link Master ports: Functional overview for Ch. B (Pin 2)

14.6.1 Configured as a digital input, Ch. B (Pin 2)

Input connection	0980 XSL 3x12...		Type 1 as per IEC 61131-2
Nominal input voltage	24 V DC		
Input current	Typically 3 mA		
Channel type	Normally open, p-switching		
Number of digital inputs	0980 XSL 3x12...	X1 .. X8	8
Status indicator	white LED		
Diagnostic indicator	red LED per channel		

Table 59: I/O ports Ch. B (Pin 2) configured as digital inputs

14.6.2 Configured as a digital output, Ch. B (Pin 2)

Attention: For variant 0980 XSL 3912-121-007D-00F-BD, the digital outputs of Channel B are **supplied by the U_L power**.

Output type	normally open, p-switching	
Nominal output voltage per channel Signal status “1” Signal status “0”	min. (U_S -1 V) **or** min. (U_L -1 V)depending on the device variant max. 2 V	
Max. output current per device	0980 XSL 3x12...	9 A (power supplied via U_L)
Max. output current per channel [5]	0980 XSL 3x12...	2 A (power supplied via U_S)
Short-circuit/overload protected	yes/yes	
Behavior in case of short circuit or overload	deactivation with automatic power-on (parameterized)	
Number of digital outputs	0980 XSL 3x12...	8
Status indicator	white LED per output	
Diagnostic indicator	red LED per channel	

Table 60: I/O ports Ch. B (Pin 2) configured as digital outputs

[5] For Class A devices: Max. 2.0 A per channel; for every port pair X1/X2, X3/X4, X5/X6, X7/X8 max. 6.5 A (for **UL applications** max. 5.0 A); for the whole port group X1 .. X8 max. 9.0 A in total (with derating).

14.7 LEDs

LED	Color	Description
U_L	Green	Auxiliary sensor/actuator voltage OK 18 V (+/- 1 V) < U_L < 30 V (+/- 1 V)
	Red*	Auxiliary sensor/actuator voltage LOW U_L < 18 V (+/- 1 V) or U_L > 30 V (+/- 1 V) * if "Report U_L supply voltage fault" is enabled.
	OFF	None of the above conditions.
U_S	Green	System/sensor voltage OK 18 V (+/- 1 V) < U_S < 30 V (+/- 1 V)
	Red	System/sensor voltage LOW U_S < 18 V (+/- 1 V) or U_S > 30 V (+/- 1 V)
	Red flashing	Device performs a factory reset (position of rotary encoding switches: 9-7-9)
	OFF	None of the above conditions.
X1 .. X8 A	Green	IO-Link COM Mode: IO-Link communication exists.
	Green flashing	IO-Link COM Mode: No IO-Link communication.
	Yellow	Standard-I/O Mode: Status of digital input or output on C/Q (pin 4) line "on".
	OFF	None of the above conditions
X1 .. X8 B	White	Status of digital input or digital output on pin 2 line "on".
	Red	Short circuit on pin 4 and pin 2 line. / All modes: Overload or short circuit on L+ (pin 1) line / communication error
	OFF	None of the above conditions.
P1 Lnk/Act P2 Lnk/Act	Green	Ethernet connection to another subscriber exists. Link detected.
	Yellow flashing	Data exchange with another subscriber.
	OFF	No connection to another subscriber. No link, no data exchange.

LED	Color	Description
BF	Red	Bus fault. No configuration, no or slow physical connection.
	Red flashing at 2 Hz	Link exists but no communication link to the Modbus TCP controller.
	OFF	Modbus TCP controller has established an active connection to the device.
DIA	Red	Modbus TCP module diagnostic alarm active.
	Red flashing at 1 Hz	Watchdog time-out; fail safe mode is active.
	Red flashing at 2 Hz, 3 sec	DCP signal service is initiated via the bus.
	Red double flash	Firmware update
	OFF	None of the above conditions.
FM	Blue and red flashing alternately	Forcemode active.
	OFF	Forcemode off.

Table 61: Information on the LED colors

14.8 Data transfer times

The following tables give an overview of the internal data transfer times of the LioN-X IO-Link Master with a connected IO-Link Device as digital I/O extension (Belden article 0960 IOL 3816-001 16DIO Hub with a minimum cycle time of 1 ms).

There are three measured data direction values for each use case:

- **PLC to DO:** Transfer of a changed PLC output data to IO-Link Device digital output.
- **DI to PLC:** Transfer of a changed digital input signal on IO-Link Device to PLC.
- **Round-trip time (RTT):** Transfer of a changed PLC output data to IO-Link Device digital output. The digital output is connected to an IO-Link Device digital input. Transfer of the changed digital input signal on IO-Link Device to PLC. RTT = [PLC to DO] + [DI to PLC].

The measured values are taken from the ethernet data transmission line. The values are therefore without PLC processing times and PLC cycle time.

The configurable digital input filter value on 0960 IOL 3816-001 was set to "off" (0 ms).

For calculation of user specific data transfer and round-trip times of possible input filters, PLC processing and cycles times must be taken into calculation.

The measured values are valid for a maximum of 48 bytes of IO-Link data for the IO-Link Master in each direction (Input/Output).

Use case 1:

IO-Link Master configuration with enabled Web interface and *disabled* IIoT protocols

Data direction	Data transfer time in ms		
	Minimum	Average	Maximum
PLC to DO	3.7	6.0	7.7
DI to PLC	1.1	3.0	4.3
RTT	6.1	8.9	11.1

Use case 2:

IO-Link Master configuration with enabled Web interface and *enabled* IIoT protocols

Data direction	Data transfer time in ms		
	Minimum	Average	Maximum
PLC to DO	7.7	10.0	13.4
DI to PLC	3.3	4.4	5.6
RTT	12.1	14.3	17.0

15 Recycling note

The symbol of a crossed-out wheeled bin shown on the device indicates that the device MUST NOT be disposed of with household waste at the end of its service life.

After its service life, the used device must be disposed of properly as electronic waste in accordance with the locally applicable disposal regulations.

End users are responsible for deleting personal data from the used device prior to disposal.

End users are obliged to separate used batteries and accumulators that are not enclosed by the used device from the used device in a nondestructive manner before disposing of the used device. The used batteries and accumulators must be handed in for separate collection. This does not apply if the used device is handed in for reuse.

16 Accessories

In order to get access to various types of accessories, please visit our Web page:

https://www.belden.com