lumbergautomation
A BELDEN BRAND

Manual

EtherCAT®

LioN-X Digital-I/O Multiprotocol:
0980 XSL 3900-121-007D-01F (16 x Input/Output)
0980 XSL 3901-121-007D-01F (16 x Input)
0980 XSL 3903-121-007D-01F (8 x Input, 8 x Output isolated)
0980 XSL 3923-121-007D-01F (8 x Input, 8 x Output)

Contents

1 About this manual

1.1 General information

Please read the assembly and operating instructions in this manual carefully before starting up the devices. Keep the manual where it is accessible to all users.

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

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

Belden Deutschland GmbH
– Lumberg Automation™ –
Im Gewerbepark 2
D-58579 Schalksmühle
Germany
lumberg-automation-support.belden.com
www.lumberg-automation.com
catalog.belden.com

Belden Deutschland GmbH – Lumberg Automation™ – reserves the right to make technical changes or changes to this manual at any time without notice.

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.

1.2.3 EtherCAT® trademark information

EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.

1.3 Version information

Version	Created	Changes
1.0	03/2023	
1.1	07/2023	Warning in ch. Setting the rotary encoding switches on page 36

Table 1: Overview of manual revisions

2 Safety instructions

2.1 Intended use

The products described in this manual are decentralized I/O Devices 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 (89/336/EEC, 93/68/EEC and 93/44/EEC) and the low voltage guideline (73/23/EEC).

The devices 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.

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 Lumberg Automation™ 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 this manual 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 – Lumberg Automation™ – 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 manual, can result in serious personal injury or damage to equipment.

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

3 Designations and synonyms

AOI	Add-On Instruction
API	Application Programming Interface
BF	Bus Fault LED
Big Endian	Data format with High-B on first place (PROFINET)
BUI	Back-Up Inconsistency (EIP diagnostics)
CC	CC-Link IE Field
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)
CoAP	Constrained Application Protocol
CSP+	Control & Communication System Profile Plus
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)
DTU	Device Temperature Underrun (EIP diagnostics)
DUT	Device under test
EIP	EtherNet/IP
ERP	Enterprise Resource Planning system
ETH	ETHERNET
FE	Functional Earth
FME	Force Mode Enabled (EIP diagnostics)
FSU	Fast Start-Up

GSDML	General Station Description Markup Language
High-B	High-Byte
ICT	Invalid Cycle Time (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 X1 .. X8
I/O port pin 4 (C/Q)	Channel A of X1 .. X8
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	Multiprotocol: 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
NTP	Network Time Protocol
OLE	Output process data Length Error (EIP diagnostics)
OPC UA	Open Platform Communications Unified Architecture (platform independent, service-oriented architecture)

PLC	Programmable Logic Controller
PN	PROFINET
PWR	Power
REST	REpresentational State Transfer
RFC	Request for Comments
RPI	Requested Packet Interval
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)
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)
U_{AUX}	$U_{Auxiliary}$, supply voltage for the load circuit (Actuator supply on Class B ports)
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)
UL	Underwriters Laboratories Inc. (certification company)
UTC	Coordinated Universal Time (Temps Universel Coordonné)

Table 2: Designations and synonyms

4 System description

The LioN modules (**L**umberg Automation™ **I**nput/**O**utput **N**etwork) 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 Device variants

The following Digital I/O device variants are available in the LioN-X family:

Article number	Product designation	Description	I/O port functionality
935705001	0980 XSL 3900-121-007D-01F	LioN-X M12-60 mm, I/O Device Multiprotocol (PN, EIP, EC, MB, CC) Security	16 x Input/Output universal
935706002	0980 XSL 3901-121-007D-01F	LioN-X M12-60 mm, I/O Device Multiprotocol (PN, EIP, EC, MB, CC) Security	16 x Input
935707001	0980 XSL 3903-121-007D-01F	LioN-X M12-60 mm, I/O Device Multiprotocol (PN, EIP, EC, MB, CC) Security	8 x Input, 8 x Output Mixmodule, galvanic isolated
935708001	0980 XSL 3923-121-007D-01F	LioN-X M12-60 mm, I/O Device Multiprotocol (PN, EIP, EC, MB, CC) Security	8 x Input, 8 x Output Mixmodule, without galvanic isolation of the outputs

Table 3: Overview of LioN-X Digital-I/O variants

4.2 I/O port overview

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

LioN-X 16DIO ports

Device variant	Port	Pin 1 U_S	Pin 4 / Ch. A (In/Out)		Pin 2 / Ch. B (In/Out)	
0980 XSL 3900...	**Info:**	–	Type 3	Supply by U_L	Type 3	Supply by U_L
	X8:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)
	X7:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)
	X6:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)
	X5:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)
	X4:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)
	X3:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)
	X2:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)
	X1:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)

Table 4: Port configuration of 0980 XSL 3900... variants

LioN-X 16DI ports

Device variant	Port	Pin 1 U_S	Pin 4 / Ch. A (Input)	Pin 2 / Ch. B (Input)
0980 XSL 3901...	**Info:**	–	Type 3	Type 3
	X8:	U_S (4 A)	DI	DI
	X7:	U_S (4 A)	DI	DI
	X6:	U_S (4 A)	DI	DI
	X5:	U_S (4 A)	DI	DI
	X4:	U_S (4 A)	DI	DI
	X3:	U_S (4 A)	DI	DI
	X2:	U_S (4 A)	DI	DI
	X1:	U_S (4 A)	DI	DI

Table 5: Port configuration of 0980 XSL 3901... variants

LioN-X 8DI8DO ports with galvanic isolation of the outputs

Device variant	Port	Pin 1 U_S	Pin 4 / Ch. A (In/Out)		Pin 2 / Ch. B (In/Out)	
0980 XSL 3903...	**Info:**	–	Type 3	Supply by U_L	Type 3	Supply by U_L
	X8:	–	–	DO (2 A)	–	DO (2 A)
	X7:	–	–	DO (2 A)	–	DO (2 A)
	X6:	–	–	DO (2 A)	–	DO (2 A)
	X5:	–	–	DO (2 A)	–	DO (2 A)
	X4:	U_S (4 A)	DI	–	DI	–
	X3:	U_S (4 A)	DI	–	DI	–
	X2:	U_S (4 A)	DI	–	DI	–
	X1:	U_S (4 A)	DI	–	DI	–

Table 6: Port configuration of 0980 XSL 3903... variants

LioN-X 8DI8DO ports without galvanic isolation of the outputs

Device variant	Port	Pin 1 U_S	Pin 4 / Ch. A (In/Out)		Pin 2 / Ch. B (In/Out)	
0980 XSL 3923...	**Info:**	–	Type 3	Supply by U_L	Type 3	Supply by U_L
	X8:	–	–	DO (2 A)	–	DO (2 A)
	X7:	–	–	DO (2 A)	–	DO (2 A)
	X6:	–	–	DO (2 A)	–	DO (2 A)
	X5:	–	–	DO (2 A)	–	DO (2 A)
	X4:	U_S (4 A)	DI	–	DI	–
	X3:	U_S (4 A)	DI	–	DI	–
	X2:	U_S (4 A)	DI	–	DI	–
	X1:	U_S (4 A)	DI	–	DI	–

Table 7: Port configuration of 0980 XSL 3923... variants

5 Overview of product features

5.1 EtherCAT® product features

Data connection

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

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

Data transmission rates

Support of 100Mbit/s with auto crossover and auto negotiation corresponding to IEEE 802.3.

Integrated switch

The integrated Ethernet switch has two EtherCAT® ports and thus supports the establishment of a line or ring topology for the EtherCAT® network.

Alarm and diagnostic messages

The devices support extended EtherCAT® diagnostic emergency messages.

ESI-based configuration and parameterization of the I/O ports

The ESI offers the option to configure and parametrize I/O ports on the Master modules via an engineering tool of a PLC.

5.2 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.

5.3 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 multiprotocol 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.

5.4 Other features

Interface protection

The devices have reverse polarity, short-circuit and overload protection for all interfaces.

For more details, see section Port assignments on page 31.

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 a loss of the 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 Digital-I/O multiprotocol variants

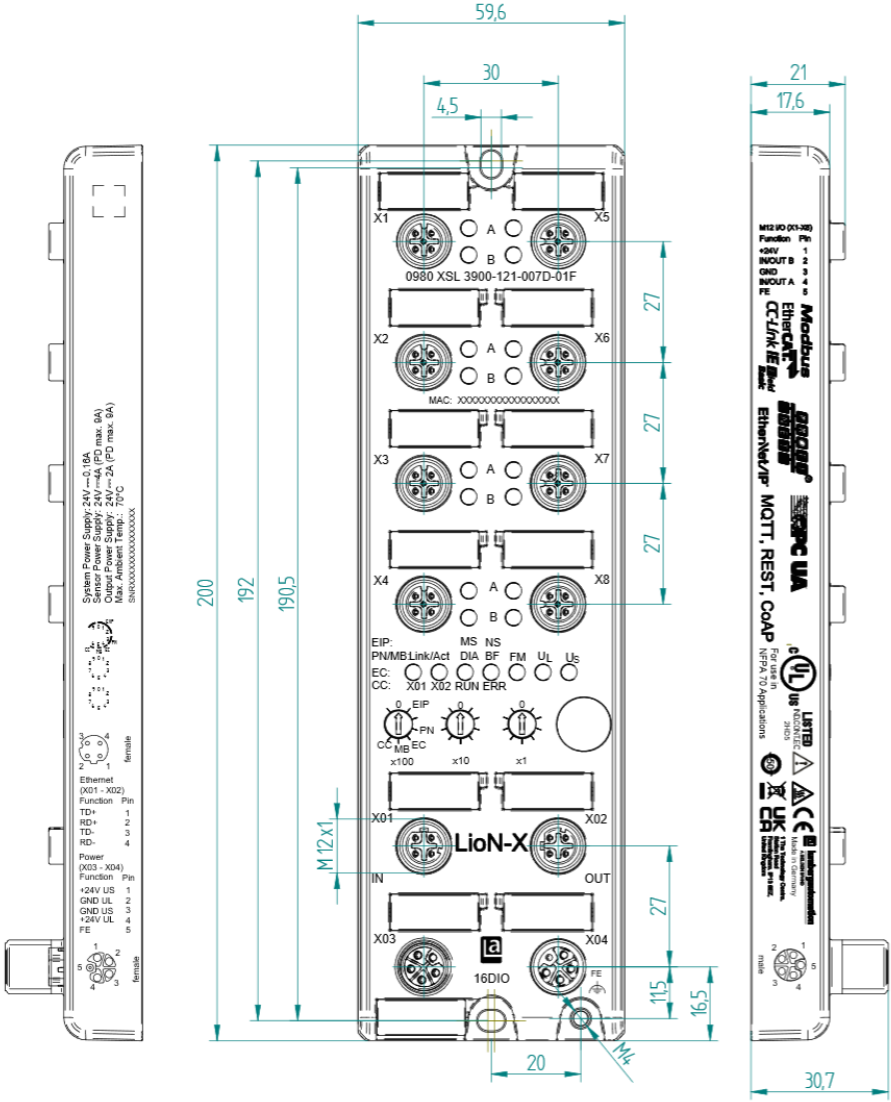


Figure 1: 0980 XSL 3900-121-007D-01F

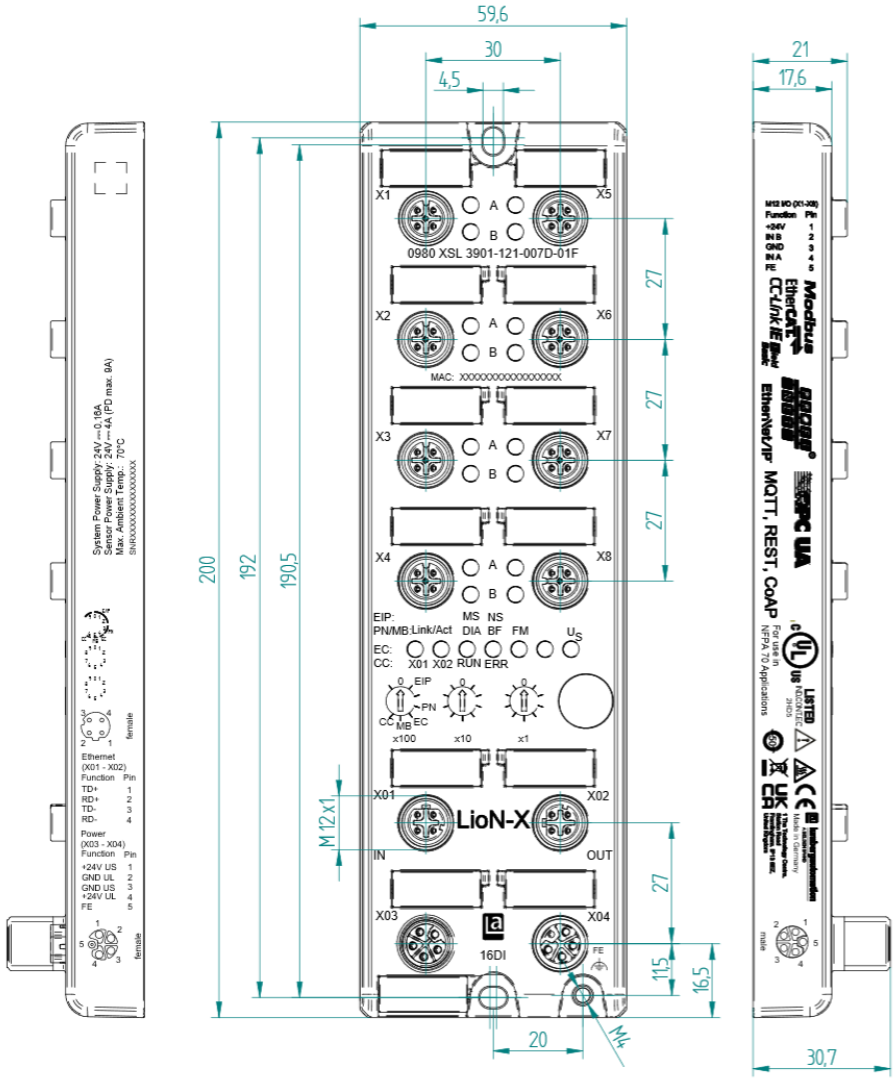


Figure 2: 0980 XSL 3901-121-007D-01F

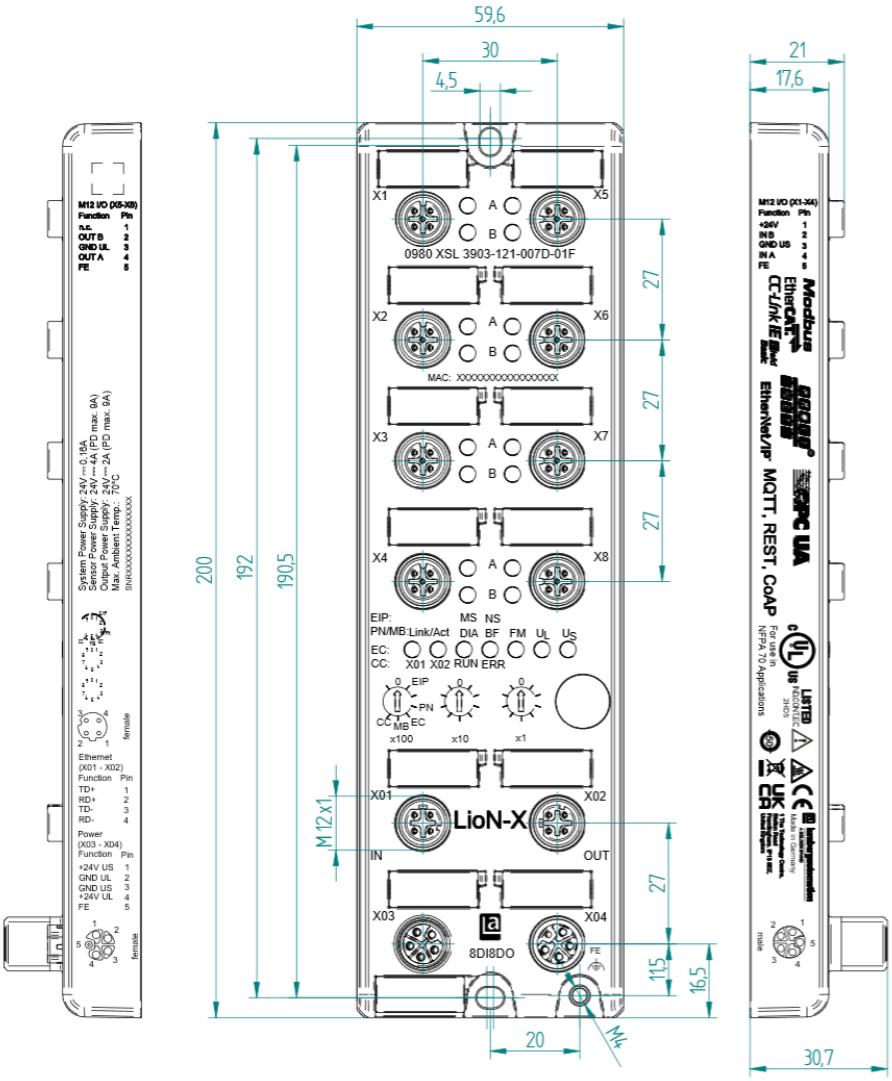


Figure 3: 0980 XSL 3903-121-007D-01F

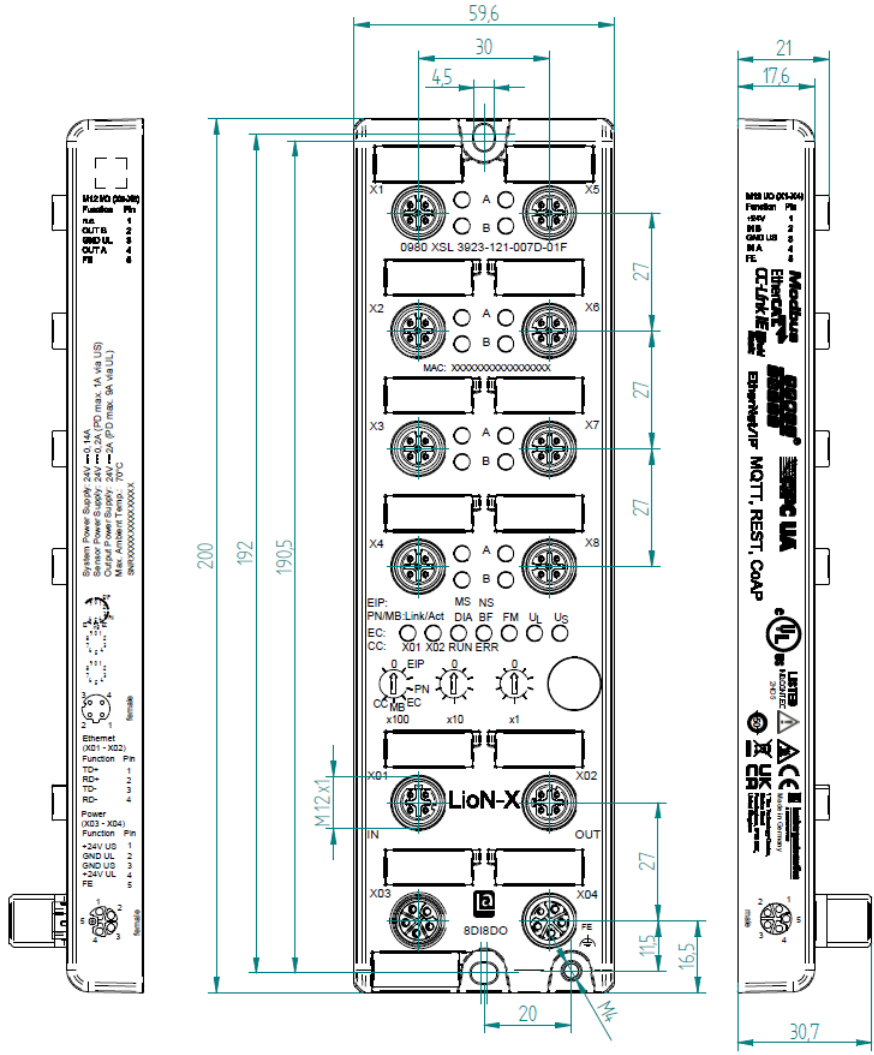


Figure 4: 0980 XSL 3923-121-007D-01F

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 LioN-X and LioN-Xlight 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.

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

Max. 4.0 A per port; for **UL applications** max. 5.0 A from U_S power supply for every port pair X1/X2, X3/X4, X5/X6, X7/X8 and max. 5.0 A from U_{AUX} power supply in total for port group 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

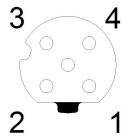


Figure 5: Schematic drawing, ports X01, X02

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 8: 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

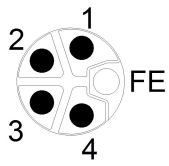


Figure 6: Schematic diagram of the M12 L-coding (connector X03 for Power In)

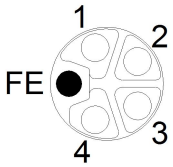


Figure 7: Schematic diagram of the M12 L-coding (socket X04 for Power Out)

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
	4	U_L (+24 V)	Load supply Actuator supply
	5	FE	Functional ground

Table 9: Pin assignments ports X03 and X04

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.

Attention: For the input module 0980 XSL 3901-xxx, the two contacts 1 and 5 are not required for the voltage supply of the actuator. Nevertheless, these two contacts are bridged together on the plug and socket side to enable a 5-pole forwarding of the voltage supply to a subsequent module.

6.3.3 I/O ports as M12 sockets

Color coding: black

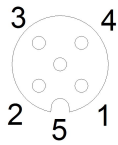


Figure 8: Schematic drawing I/O port as M12 socket

6.3.3.1 I/O ports

0980 XSL 3900-121...	Pin	Signal	Function
16DIO 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	IN/OUT	Ch. A: Digital input or digital output
	5	FE	Functional ground

0980 XSL 3901-121...	Pin	Signal	Function
16DI X1 .. X8	1	+24 V	power supply +24 V
	2	IN	Ch. B: Digital input
	3	GND U_S	Ground/reference potential
	4	IN	Ch. A: Digital input
	5	FE	Functional ground

0980 XSL 39x3-121...	Pin	Signal	Function
8DI8DO X1 .. X4	1	+24 V	power supply +24 V
	2	IN	Ch. B: Digital input
	3	GND U_S	Ground/reference potential
	4	IN	Ch. A: Digital input
	5	FE	Functional ground
8DI8DO X5 .. X8	1	n.c.	–
	2	OUT	Ch. B: Digital output
	3	GND U_L	Ground/reference potential
	4	OUT	Ch. A: Digital output
	5	FE	Functional ground

Table 10: Pin assignments I/O ports

7 Starting operation

7.1 ESI file

An ESI file in XML format is required to configure the LioN-X EtherCAT® devices. All device variants are grouped in a single ESI file. The file can be downloaded from the product pages on our online catalog: catalog.belden.com

On request, the ESI file is also sent by the support team.

The ESI file is named **LumbergAutomation-LioN-X-Digital.xml**.

Install the ESI file for the device variant used with the aid of the hardware or network configuration tool of your controller manufacturer.

For TwinCAT®, the ESI file normally has to be copied into the installation folder, e.g.: `C:\TwinCAT\3.1\Config\Io\EtherCAT`

After installation, TwinCAT® needs a system restart. Alternatively, use the menu bars in TwinCAT® to reload the application:

TWINCAT > **EtherCAT Devices** > **Reload Device Descriptions**.

As a result, the EtherCAT® devices are now available in the hardware catalog.

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.

For EtherCAT®, the MAC address has no function. For EoE (Ethernet over EtherCAT®), a virtual MAC address will be assigned to the I/O module.

7.3 Setting the rotary encoding switches

The following LioN-X variants support multiprotocol application for the protocols EtherNet/IP (E/IP), PROFINET (P), EtherCAT® (EC), Modbus TCP (MB) and CC-Link IE Field Basic (CC):

- 0980 XSL 3900-121-007D-01F
- 0980 XSL 3901-121-007D-01F
- 0980 XSL 3903-121-007D-01F
- 0980 XSL 3923-121-007D-01F

Caution: Risk of device damage due to memory malfunction

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.

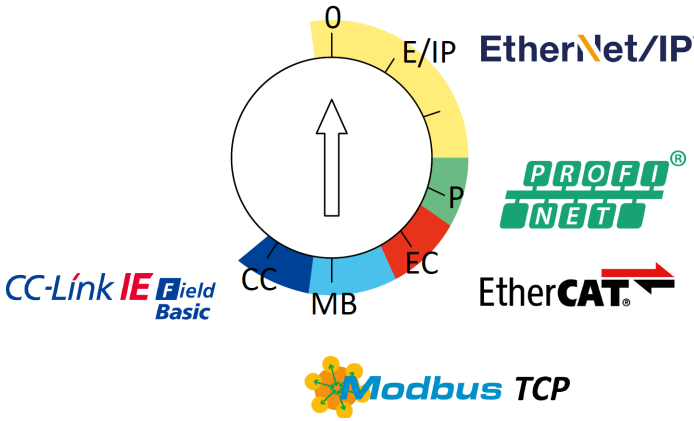
- Make sure that the power supply is maintained during the entire process.

The LioN-X multiprotocol variants allow you to select different protocols for communication within an industrial Ethernet system. In this way Digital I/O devices with multiprotocol 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 I/O Device 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 multiprotocol 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 11: 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 39.

If you position the rotary encoding switch in a manner that is invalid, the device signals this to you with a blink code (the BF/MS LED blinks in red three times).

7.3.1 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 is blinking red. After the internal memory write processes have finished, the U_S LED returns to display static green or red light, in dependency of the actual U_S voltage.

	x100	x10	x1
Factory Reset	9	7	9

Follow the steps from section Setting the rotary encoding switches on page 36 again to select a new protocol.

For performing a factory reset via software configuration, see chapter OPC UA configuration on page 95 and the configuration section.

8 Configuration and operation in TwinCAT® 3

8.1 PDO assignments

Only applicable for variant 0980 XSL 3900-121-007D-00F (16 x Input/Output).

The device supports different PDO (Process Data Object) assignments for input and output data.

By selecting the relevant PDO, you can choose your preferred I/O data content. The device features a dynamic, slot-based PDO assignment. The following PDO assignments are provided:

8.1.1 Input data

PDO 0x1A00 (TxPDO Mapping 2 Byte)

PDO		PDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x1A00	2	0x6000:1	1	UINT32	SubIndex 001
		0x6000:2	1	UINT32	SubIndex 002

PDO 0x1A01 (TxPDO Mapping 16 Bits)

PDO		PDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x1A01	16	0x6020:1	1	UINT32	SubIndex 001
		0x6020:2	1	UINT32	SubIndex 002
		0x6020:3	1	UINT32	SubIndex 003
		0x6020:4	1	UINT32	SubIndex 004
		0x6020:5	1	UINT32	SubIndex 005
		0x6020:6	1	UINT32	SubIndex 006
		0x6020:7	1	UINT32	SubIndex 007
		0x6020:8	1	UINT32	SubIndex 008
		0x6020:9	1	UINT32	SubIndex 009
		0x6020:10	1	UINT32	SubIndex 010
		0x6020:11	1	UINT32	SubIndex 011
		0x6020:12	1	UINT32	SubIndex 012
		0x6020:13	1	UINT32	SubIndex 013
		0x6020:14	1	UINT32	SubIndex 014
		0x6020:15	1	UINT32	SubIndex 015
		0x6020:16	1	UINT32	SubIndex 016

PDO 0x1A10 (TxPDO Mapping 1 Byte)

PDO		PDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x1A10	1	0x6000:1	1	UINT32	SubIndex 001

PDO 0x1A11 (TxPDO Mapping 8 Bits)

PDO		PDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x1A11	8	0x6020:1	1	UINT32	SubIndex 001
		0x6020:2	1	UINT32	SubIndex 002
		0x6020:3	1	UINT32	SubIndex 003
		0x6020:4	1	UINT32	SubIndex 004
		0x6020:5	1	UINT32	SubIndex 005
		0x6020:6	1	UINT32	SubIndex 006
		0x6020:7	1	UINT32	SubIndex 007
		0x6020:8	1	UINT32	SubIndex 008

PDO 0x1A04 (TxPDO Error Register)

PDO		PDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x1A04	1	0x1000:1	1	UINT32	SubIndex 001

PDO 0x1A05 (TxPDO Diagnostic Register)

PDO		PDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x1A05	1	0x2001:1	1	UINT32	SubIndex 001

PDO 0x1A81 (TxPDO U_S/U_L measurements)

PDO		PDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x1A81	2	0x2002:1	1	UINT32	SubIndex 001
		0x2002:2	1	UINT32	SubIndex 002

PDO 0x1A82 (TxPDO current measurements)

PDO		PDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x1A82	8	0x2003:1	1	UINT32	SubIndex 001
		0x2003:2	1	UINT32	SubIndex 002
		0x2003:3	1	UINT32	SubIndex 003
		0x2003:4	1	UINT32	SubIndex 004
		0x2003:5	1	UINT32	SubIndex 005
		0x2003:6	1	UINT32	SubIndex 006
		0x2003:7	1	UINT32	SubIndex 007
		0x2003:8	1	UINT32	SubIndex 008

8.1.2 Output data

PDO 0x1600 (RxPDO Mapping 2 Byte)

PDO		PDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x1600	2	0x7000:1	1	UINT32	SubIndex 001
		0x7000:2	1	UINT32	SubIndex 002

PDO 0x1601 (RxPDO Mapping 16 Bits)

PDO		PDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x1601	16	0x7020:1	1	UINT32	SubIndex 001
		0x7020:2	1	UINT32	SubIndex 002
		0x7020:3	1	UINT32	SubIndex 003
		0x7020:4	1	UINT32	SubIndex 004
		0x7020:5	1	UINT32	SubIndex 005
		0x7020:6	1	UINT32	SubIndex 006
		0x7020:7	1	UINT32	SubIndex 007
		0x7020:8	1	UINT32	SubIndex 008
		0x7020:9	1	UINT32	SubIndex 009
		0x7020:10	1	UINT32	SubIndex 010
		0x7020:11	1	UINT32	SubIndex 011
		0x7020:12	1	UINT32	SubIndex 012
		0x7020:13	1	UINT32	SubIndex 013
		0x7020:14	1	UINT32	SubIndex 014
		0x7020:15	1	UINT32	SubIndex 015
		0x7020:16	1	UINT32	SubIndex 016

PDO 0x1610 (RxPDO Mapping 1 Byte)

PDO		PDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x1610	1	0x7000:1	1	UINT32	SubIndex 001

PDO 0x1611 (RxPDO Mapping 8 Bits)

PDO		PDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x1611	8	0x7020:1	1	UINT32	SubIndex 001
		0x7020:2	1	UINT32	SubIndex 002
		0x7020:3	1	UINT32	SubIndex 003
		0x7020:4	1	UINT32	SubIndex 004
		0x7020:5	1	UINT32	SubIndex 005
		0x7020:6	1	UINT32	SubIndex 006
		0x7020:7	1	UINT32	SubIndex 007
		0x7020:8	1	UINT32	SubIndex 008

8.1.3 Modular slots

Only applicable for variant 0980 XSL 3900-121-007D-00F (16 x Input/Output).

The ESI file features a modular slot-based PDO configuration. The following slots are available for the I/O configuration:

Slot name	Description
16DI/DO (byte)	16DI / 16DO, byte-wise
16DI (byte)	16DI, byte-wise
16DO (byte)	16DO, byte-wise
8DI/8DO (byte)	8DI/8DO, byte-wise
8DI (byte)	8DI, byte-wise
8DO (byte)	8DO, byte-wise
16DI/DO (bit)	16DI / 16DO, bit-wise
16DI (bit)	16DI bit-wise
16DO (bit)	16DO, bit-wise
8DI/8DO (bit)	8DI/8DO, bit-wise
8DI (bit)	8DI, bit-wise
8DO (bit)	8DO, bit-wise

8.2 Device parameters

The device supports different parameters. The parameters must be transferred from the controller to the device during startup. The following blocks of parameters can be adjusted.

Certain configuration parameters apply only to Digital Outputs or only to Digital Inputs. For these to be effective, the corresponding channel must have output or input functionality and must also be configured accordingly.

Configuration parameter	Applicable for channel configuration
Surveillance Timeout	DIO, Output
Failsafe	DIO, Output
Auto Restart	DIO, Output
Current Limit	DIO, Output
Input Filter Time	DIO, Input
Input Logic	DIO, Input

8.2.1 Failsafe mode for the digital output

The firmware of the devices provides a fail-safe function for ports in *Digital Output* mode. During device configuration, you have the option to define the status of channels A and B for ports in *Digital Output* mode in the case of an interruption or the loss of communication.

The following table represents possible failsafe replacement values of ports in *Digital Output* mode.

SDO		SDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x2380	16	0x2380:1	1	UINT8	Port X1 A 0 = Set Low 1 = Set High 2 = Hold Last Others: reserved
		0x238 :2	1	UINT8	Port X1 B 0 = Set Low 1 = Set High 2 = Hold Last Others: reserved
		0x2380:3	1	UINT8	Port X2 A 0 = Set Low 1 = Set High 2 = Hold Last Others: reserved
		0x2380:4	1	UINT8	Port X2 B 0 = Set Low 1 = Set High 2 = Hold Last Others: reserved
		0x2380:5	1	UINT8	Port X3 A 0 = Set Low 1 = Set High 2 = Hold Last Others: reserved
		0x2380:6	1	UINT8	Port X3 B 0 = Set Low 1 = Set High 2 = Hold Last Others: reserved

SDO		SDO content			
		0x2380:7	1	UINT8	Port X4 A 0 = Set Low 1 = Set High 2 = Hold Last Others: reserved
		0x2380:8	1	UINT8	Port X4 B 0 = Set Low 1 = Set High 2 = Hold Last Others: reserved
		0x2380:9	1	UINT8	Port X5 A 0 = Set Low 1 = Set High 2 = Hold Last Others: reserved
		0x2380:10	1	UINT8	Port X5 B 0 = Set Low 1 = Set High 2 = Hold Last Others: reserved
		0x2380:11	1	UINT8	Port X6 A 0 = Set Low 1 = Set High 2 = Hold Last Others: reserved
		0x2380:12	1	UINT8	Port X6 B 0 = Set Low 1 = Set High 2 = Hold Last Others: reserved
		0x2380:13	1	UINT8	Port X7 A 0 = Set Low 1 = Set High 2 = Hold Last Others: reserved

SDO		SDO content			
		0x2380:14	1	UINT8	Port X7 B 0 = Set Low 1 = Set High 2 = Hold Last Others: reserved
		0x2380:15	1	UINT8	Port X8 A 0 = Set Low 1 = Set High 2 = Hold Last Others: reserved
		0x2380:16	1	UINT8	Port X8 B 0 = Set Low 1 = Set High 2 = Hold Last Others: reserved

These values are only applicable when the port is in *Digital Output* mode.

8.2.2 General device settings

The device supports the setting of different parameters. The following blocks of parameters can be adjusted:

SDO		SDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x2381	7	0x2381:1	1	BOOL	Web Interface Locked 0 = false, 1 = true
		0x2381:2	1	BOOL	Force Mode Locked 0 = false, 1 = true
		0x2381:3	1	BOOL	Disable U_S Emergency Messages 0 = false, 1 = true
		0x2381:4	1	BOOL	Disable U_L Emergency Messages 0 = false, 1 = true
		0x2381:5	1	BOOL	Disable Actuator Emergency Messages without U_L 0 = false, 1 = true
		0x2381:6	1	BOOL	Enable External Configuration 0 = false, 1 = true
		0x2381:7	1	BOOL	Automatic Output Restart after failure 0 = false, 1 = true

8.2.3 Surveillance timeout

The firmware of the devices allows you to define a delay time before the automatic monitoring of the output currents. This is known as the surveillance timeout.

You can define the surveillance timeout for every individual output channel.

The delay time begins after the output channel has been activated (after a rising edge) or deactivated (after a falling edge). After the surveillance timeout has elapsed, the monitoring of the output begins and the diagnosis will report error states.

The value of the surveillance timeout is 0 to 255 ms. The default value is 80 ms.

SDO		SDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x2382	16	0x2382:1	1	UINT8	Surveillance Timeout Port 1 Channel A
		0x2382:2	1	UINT8	Surveillance Timeout Port 1 Channel B
		0x2382:3	1	UINT8	Surveillance Timeout Port 2 Channel A
		0x2382:4	1	UINT8	Surveillance Timeout Port 2 Channel B
		0x2382:5	1	UINT8	Surveillance Timeout Port 3 Channel A
		0x2382:6	1	UINT8	Surveillance Timeout Port 3 Channel B
		0x2382:7	1	UINT8	Surveillance Timeout Port 4 Channel A
		0x2382:8	1	UINT8	Surveillance Timeout Port 4 Channel B

SDO		SDO content			
		0x2382:9	1	UINT8	Surveillance Timeout Port 5 Channel A
		0x2382:10	1	UINT8	Surveillance Timeout Port 5 Channel B
		0x2382:11	1	UINT8	Surveillance TimeoutPort 6 Channel A
		0x2382:12	1	UINT8	Surveillance Timeout Port 6 Channel B
		0x2382:13	1	UINT8	Surveillance Timeout Port 7 Channel A
		0x2382:14	1	UINT8	Surveillance Timeout Port 7 Channel B
		0x2382:15	1	UINT8	Surveillance Timeout Port 8 Channel A
		0x2382:16	1	UINT8	Surveillance Timeout Port 8 Channel B

8.2.4 Digital input logic

The device supports the configuration of digital input logic for Channel A (Pin 4) and Channel B (Pin 2) of the port.

SDO		SDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x2384	16	0x2384:1	1	UINT8	Digital Input logic Port 1 Channel A 0: NO 1: NC
		0x2384:2	1	UINT8	Digital Input logic Port 1 Channel B 0: NO 1: NC
		0x2384:3	1	UINT8	Digital Input logic Port 2 Channel A 0: NO 1: NC
		0x2384:4	1	UINT8	Digital Input logic Port 2 Channel B 0: NO 1: NC
		0x2384:5	1	UINT8	Digital Input logic Port 3 Channel A 0: NO 1: NC
		0x2384:6	1	UINT8	Digital Input logic Port 3 Channel B 0: NO 1: NC
		0x2384:7	1	UINT8	Digital Input logic Port 4 Channel A 0: NO 1: NC

SDO		SDO content			
		0x2384:8	1	UINT8	Digital Input logic Port 4 Channel B 0: NO 1: NC
		0x2384:9	1	UINT8	Digital Input logic Port 5 Channel A 0: NO 1: NC
		0x2384:10	1	UINT8	Digital Input logic Port 5 Channel B 0: NO 1: NC
		0x2384:11	1	UINT8	Digital Input logic Port 6 Channel A 0: NO 1: NC
		0x2384:12	1	UINT8	Digital Input logic Port 6 Channel B 0: NO 1: NC
		0x2384:13	1	UINT8	Digital Input logic Port 7 Channel A 0: NO 1: NC
		0x2384:14	1	UINT8	Digital Input logic Port 7 Channel B 0: NO 1: NC
		0x2384:15	1	UINT8	Digital Input logic Port 8 Channel A 0: NO 1: NC
		0x2384:16	1	UINT8	Digital Input logic Port 8 Channel B 0: NO 1: NC

These values are only applicable for pins of a port in *Digital Input* mode.

8.2.5 Digital input filter

The device supports the configuration of a digital input filter (in ms) for Channel A (Pin 4) and Channel B (Pin 2) of the port.

Example: Value "100" = 10 ms

SDO		SDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x2385	16	0x2385:1	1	UINT8	Digital Input Filter Port 1 Channel A
		0x2385:2	1	UINT8	Digital Input Filter Port 1 Channel B
		0x2385:3	1	UINT8	Digital Input Filter Port 2 Channel A
		0x2385:4	1	UINT8	Digital Input Filter Port 2 Channel B
		0x2385:5	1	UINT8	Digital Input Filter Port 3 Channel A
		0x2385:6	1	UINT8	Digital Input Filter Port 3 Channel B
		0x2385:7	1	UINT8	Digital Input Filter Port 4 Channel A
		0x2385:8	1	UINT8	Digital Input Filter Port 4 Channel B
		0x2385:9	1	UINT8	Digital Input Filter Port 5 Channel A
		0x2385:10	1	UINT8	Digital Input Filter Port 5 Channel B
		0x2385:11	1	UINT8	Digital Input Filter Port 6 Channel A
		0x2385:12	1	UINT8	Digital Input Filter Port 6 Channel B
		0x2385:13	1	UINT8	Digital Input Filter Port 7 Channel A
		0x2385:14	1	UINT8	Digital Input Filter Port 7 Channel B
		0x2385:15	1	UINT8	Digital Input Filter Port 8 Channel A
		0x2385:16	1	UINT8	Digital Input Filter Port 8 Channel B

These values are only applicable for pins of a port in *Digital Input* mode.

8.2.6 Digital Output restart

The device supports the option to enable or disable a digital output restart for a given Channel A (Pin 4) or Channel B (Pin 2) of the port. Timeout: ~1 s

SDO		SDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x2386	16	0x2386:1	1	BOOL	Enable the Digital Output restart for Port X1 Channel A
		0x2386:2	1	BOOL	Enable the Digital Output restart for Port X1 Channel B
		0x2386:3	1	BOOL	Enable the Digital Output restart for Port X2 Channel A
		0x2386:4	1	BOOL	Enable the Digital Output restart for Port X2 Channel B
		0x2386:5	1	BOOL	Enable the Digital Output restart for Port X3 Channel A
		0x2386:6	1	BOOL	Enable the Digital Output restart for Port X3 Channel B
		0x2386:7	1	BOOL	Enable the Digital Output restart for Port X4 Channel A
		0x2386:8	1	BOOL	Enable the Digital Output restart for Port X4 Channel B
		0x2386:9	1	BOOL	Enable the Digital Output restart for Port X5 Channel A
		0x2386:10	1	BOOL	Enable the Digital Output restart for Port X5 Channel B
		0x2386:11	1	BOOL	Enable the Digital Output restart for Port X6 Channel A

SDO		SDO content			
		0x2386:12	1	BOOL	Enable the Digital Output restart for Port X6 Channel B
		0x2386:13	1	BOOL	Enable the Digital Output restart for Port X7 Channel A
		0x2386:14	1	BOOL	Enable the Digital Output restart for Port X7 Channel B
		0x2386:15	1	BOOL	Enable the Digital Output restart for Port X8 Channel A
		0x2386:16	1	BOOL	Enable the Digital Output restart for Port X8 Channel B

8.2.7 I/O mapping configuration

The device supports the configuration of the I/O mapping of the port.

SDO		SDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x2387	16	0x2387:1	1	UINT8	I/O Mapping Configuration Port X1 A: 0 .. 15 = "Process Data Channel 0 .. 15", 255 = "Inactive"
		0x2387:2	1	UINT8	I/O Mapping Configuration Port X1 B: 0 .. 15 = "Process Data Channel 0 .. 15", 255 = "Inactive"
		0x2387:3	1	UINT8	I/O Mapping Configuration Port X2 A: 0 .. 15 = "Process Data Channel 0 .. 15", 255 = "Inactive"
		0x2387:4	1	UINT8	I/O Mapping Configuration Port X2 B: 0 .. 15 = "Process Data Channel 0 .. 15", 255 = "Inactive"
		0x2387:5	1	UINT8	I/O Mapping Configuration Port X3 A: 0 .. 15 = "Process Data Channel 0 .. 15", 255 = "Inactive"
		0x2387:6	1	UINT8	I/O Mapping Configuration Port X3 B: 0 .. 15 = "Process Data Channel 0 .. 15", 255 = "Inactive"
		0x2387:7	1	UINT8	I/O Mapping Configuration Port X4 A: 0 .. 15 = "Process Data Channel 0 .. 15", 255 = "Inactive"
		0x2387:8	1	UINT8	I/O Mapping Configuration Port X4 B: 0 .. 15 = "Process Data Channel 0 .. 15", 255 = "Inactive"
		0x2387:9	1	UINT8	I/O Mapping Configuration Port X5 A: 0 .. 15 = "Process Data Channel 0 .. 15", 255 = "Inactive"
		0x2387:10	1	UINT8	I/O Mapping Configuration Port X5 B: 0 .. 15 = "Process Data Channel 0 .. 15", 255 = "Inactive"
		0x2387:11	1	UINT8	I/O Mapping Configuration Port X6 A: 0 .. 15 = "Process Data Channel 0 .. 15", 255 = "Inactive"

SDO		SDO content			
		0x2387:12	1	UINT8	I/O Mapping Configuration Port X6 B: 0 .. 15 = "Process Data Channel 0 .. 15", 255 = "Inactive"
		0x2387:13	1	UINT8	I/O Mapping Configuration Port X7 A: 0 .. 15 = "Process Data Channel 0 .. 15", 255 = "Inactive"
		0x2387:14	1	UINT8	I/O Mapping Configuration Port X7 B: 0 .. 15 = "Process Data Channel 0 .. 15", 255 = "Inactive"
		0x2387:15	1	UINT8	I/O Mapping Configuration Port X8 A: 0 .. 15 = "Process Data Channel 0 .. 15", 255 = "Inactive"
		0x2387:16	1	UINT8	I/O Mapping Configuration Port X8 B: 0 .. 15 = "Process Data Channel 0 .. 15", 255 = "Inactive"

8.2.8 Output current limit

The device supports the configuration of a current limit for the output for Channel A (Pin 4) and Channel B (Pin 2) of the port.

SDO		SDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x2388	16	0x2388:1	1	UINT16	Current limit (in mA) to turn off X1 A: Default: 2000
		0x2388:2	1	UINT16	Current limit (in mA) to turn off X1 B: Default: 2000
		0x2388:3	1	UINT16	Current limit (in mA) to turn off X2 A: Default: 2000
		0x2388:4	1	UINT16	Current limit (in mA) to turn off X2 B: Default: 2000
		0x2388:5	1	UINT16	Current limit (in mA) to turn off X3 A: Default: 2000
		0x2388:6	1	UINT16	Current limit (in mA) to turn off X3 B: Default: 2000
		0x2388:7	1	UINT16	Current limit (in mA) to turn off X4 A: Default: 2000
		0x2388:8	1	UINT16	Current limit (in mA) to turn off X4 B: Default: 2000
		0x2388:9	1	UINT16	Current limit (in mA) to turn off X5 A: Default: 2000
		0x2388:10	1	UINT16	Current limit (in mA) to turn off X5 B: Default: 2000
		0x2388:11	1	UINT16	Current limit (in mA) to turn off X6 A: Default: 2000

SDO		SDO content			
		0x2388:12	1	UINT16	Current limit (in mA) to turn off X6 B: Default: 2000
		0x2388:13	1	UINT16	Current limit (in mA) to turn off X7 A: Default: 2000
		0x2388:14	1	UINT16	Current limit (in mA) to turn off X7 B: Default: 2000
		0x2388:15	1	UINT16	Current limit (in mA) to turn off X8 A: Default: 2000
		0x2388:16	1	UINT16	Current limit (in mA) to turn off X8 B: Default: 2000

8.3 Configuration example with TwinCAT® 3

The configuration and start-up of the devices described below refer to the TwinCAT® 3 software by Beckhoff Automation GmbH & Co. KG. If you use the control system of another provider,please consider the related documentation.

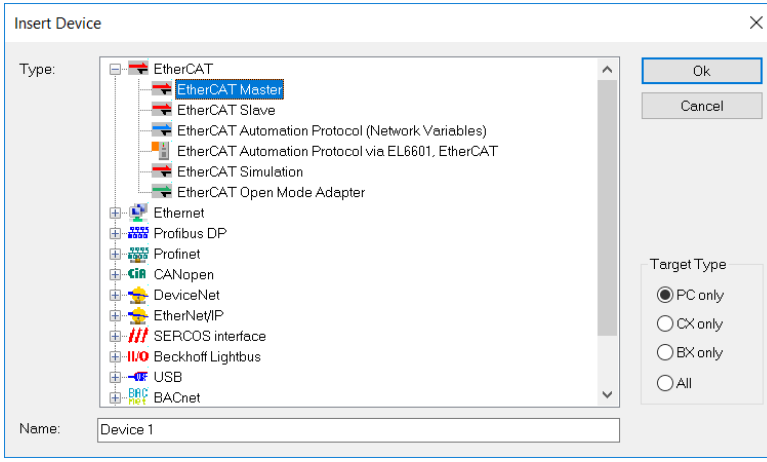
1. Install the ESI file of the device family in TwinCAT®. For TwinCAT®, the ESI file must be copied into the installation folder, e.g.: `C:\TwinCAT\3.1\Config\Io\EtherCAT`.

2. After installation, TwinCAT® needs a system restart. Alternatively, use the menu bars in TwinCAT® to reload the application: **TWINCAT** > **EtherCAT Devices** > **Reload Device Descriptions**.

The devices are now available in the hardware catalog.

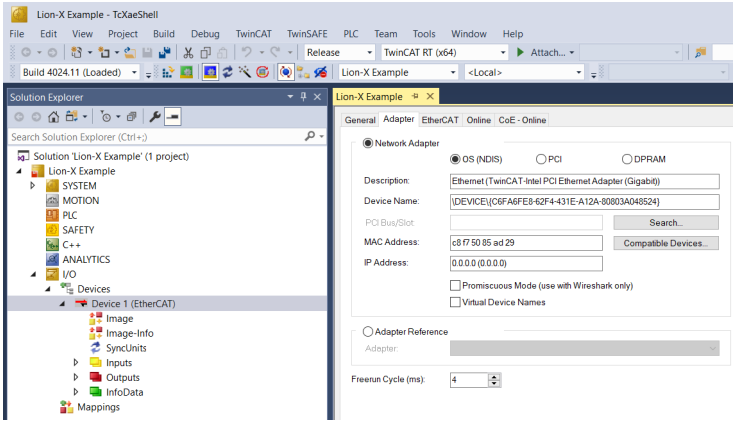
3. Start TwinCAT® and open a new project.

4. Browse to **Solution Explorer** > **I/O** > **Devices** in the left workspace window. Right-click on **Devices** and choose the option **Add New Item …** > **EtherCAT Master**.



5. If not already done, choose the network adapter and install the driver for EtherCAT® real time communication.

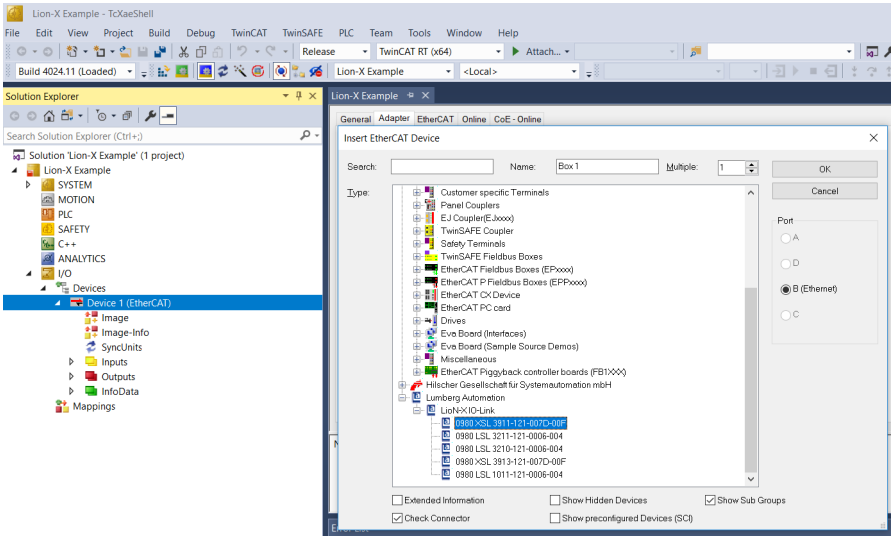
Browse to **Adapter** in the right workspace window and click on **Compatible Devices...** to choose the driver and start the installation.



8.3.1 Configuration of 0980 XSL 3900-121-... devices

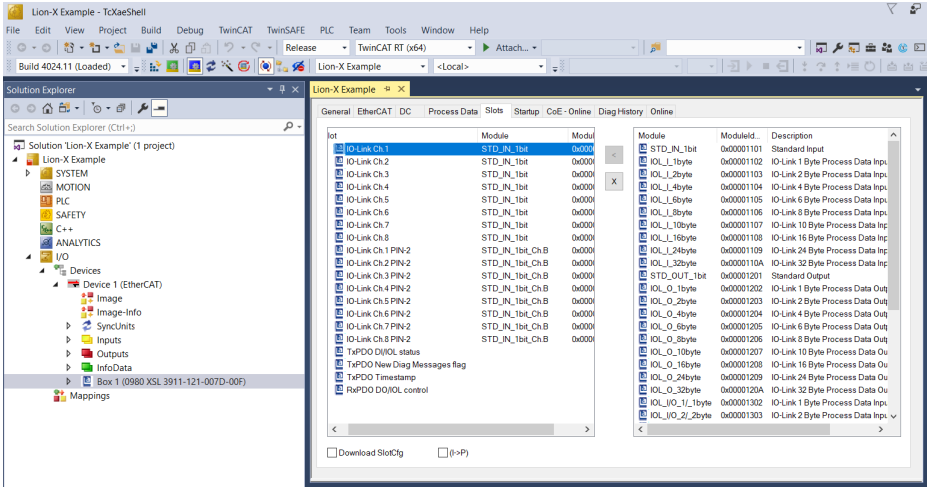
1. Select the I/O device from the hardware catalog:

Browse to **Solution Explorer** > **I/O** > **Devices** in the left workspace window. Right-click on **Device 1 (EtherCAT)** and choose the option **Add New Item ...**. Select the device and click on **OK**.



2. Configure the “Slots”:

Browse to **Slots** in the right workspace window and configure the DIO module, e.g. for byte-wise or bit-wise channel mode DI/DO. Additional PDOs like "TxPDO Error register", "TxPDO Diagnostic register", "TxPDO U_S/U_L measurements" and "Current measurements" can be set as well.



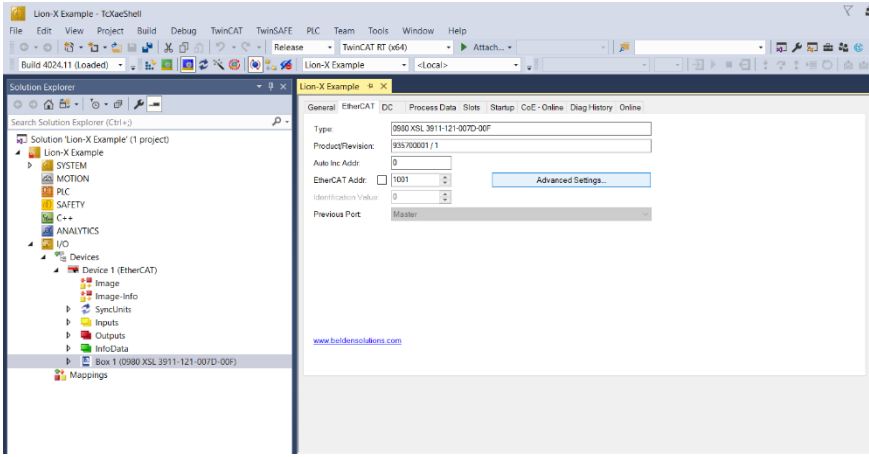
3. Configure the Process Data:

Browse to **Process Data** in the right workspace window and choose the PDOs for Inputs and Outputs.

8.3.2 EoE IP address

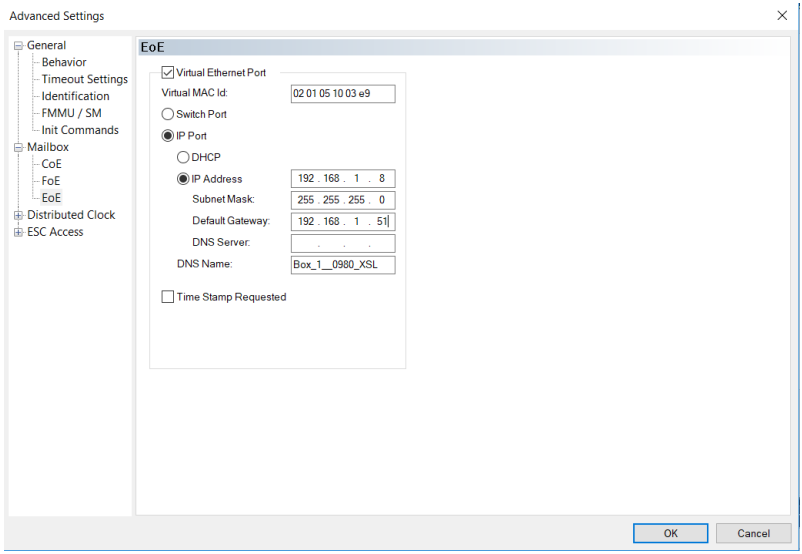
1. Set the IP address for the EoE (Ethernet over EtherCAT®) protocol:

For using the Web interface of the device, the IP address must be set. Click on **EtherCAT** > **Advanced Settings...** in the right workspace window and navigate to **Mailbox** > **EoE**.



2. Disable the option **Virtual Ethernet Port** when using no Web services.

3. Activate **IP Port** and **IP Address** when using Web services. Enter your IP settings depending from your local network adapter settings.



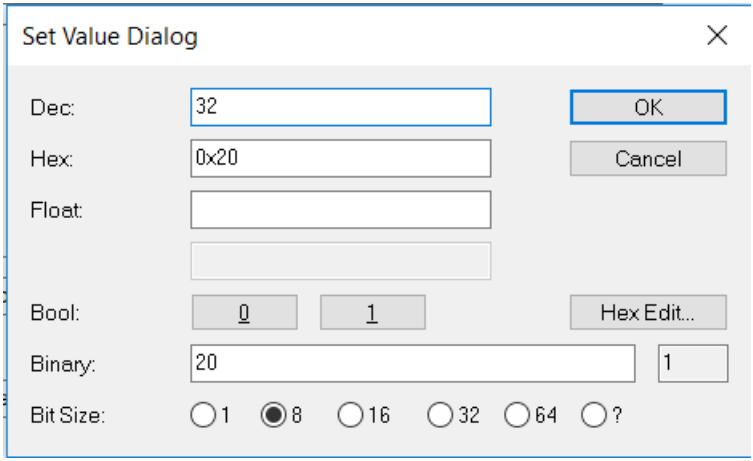
8.3.3 Activate configuration

Warning: Risk of personal injury or damage of the equipment. Keep away from moveable machine parts during setting up the inputs or outputs of the device.

1. When the device is connected to the EtherCAT® network, click on **TwinCAT** in the top ribbon and choose **Activate Configuration** in the upcoming window.

2. Click again on **TwinCAT** in the top ribbon and choose **Restart TwinCAT (Config Mode)**. Accept the dialog boxes by clicking on **Yes**. The device will now be changed to “OP” state and will be transferring I/O data.

3. Click on **Write...** to set up any output of the device.



9 Diagnostics processing

The devices provide advanced diagnosis behavior, in particular for the output channels. The firmware of the devices distinguishes between 5 different types of error.

9.1 Channel error

A channel error is determined by comparing the target value set by a controller with the actual value of an output channel.

Target value	Actual value	Comment
Active	Active	OK, no diagnostics
Off	Off	OK, no diagnostics
Active	Off	Short-circuit Channel indicator is red. Channel error bit in the diagnostics is set. Channel is locked after the error is rectified. (Automatic output restart is parametrized as default value for the 16DIO “Universal” devices.)
Off	Active	Voltage is fed back in Red and yellow/white channel indicators are activated. Channel error bit in the diagnosis is set. Channel is not locked after the error is rectified.

Table 12: Interpretation of channel errors

If both output channels of an M12 slot are activated and a channel error occurs, the controller locks both channels, even if only one channel is affected by the error. If only one channel is activated, the controller exclusively locks this one. Locked channels are deactivated and remain in the "Off" state if you do not reset them via the controller.

When an output channel is activated (rising edge of the channel state) or deactivated (falling edge), the channel errors are filtered for the period that you set using the "Surveillance-Timeout" parameter during the configuration

of the module. The value of this parameter can range from 0 to 255 ms; the factory setting is 80 ms.

The filter is used to avoid premature error messages when a capacitive load is activated or an inductive load is deactivated, and during other voltage peaks when a status changes.

In static state of the output channel, that is, while the channel is permanently switched on or off, the controller uses a fixed specific filter time of 100 ms between error detection and the diagnostic message.

9.2 Voltage error at the M12 slots (sensor short-circuit)

At every M12 input socket of the modules, pin 1 supplies a monitored sensor voltage U_S.

In the case of a sensor short-circuit, a voltage error is reported. Both channel indicators of the M12 input socket light up in red, and the respective error bit for the sensor short-circuit is set in the diagnosis bytes.

9.2.1 Error of the actuator power supply U_L

For the following device variants, the digital outputs are supplied by the U_L power:

- 0980 XSL 3900-121-007D-01F
- 0980 XSL 3903-121-007D-01F

The voltage value for the incoming U_L power supply is monitored globally in the I/O Module. If U_L supply voltage alarms are enabled, an error message is generated in case the voltage drops below approx. 18 V or exceeds approx. 30 V.

If output channels are active, additional error messages caused by the voltage failure are generated on the I/O ports. U_L supply voltage alarms are disabled by default and can be enabled via parameterization.

The following LED behavior is visible:

- For disabled U_L supply voltage fault alarms, the U_L indicator LED is "off" in case of voltage drops below approx. 18 V.
- For enabled U_L supply voltage fault alarms, the U_L indicator LED is "red" in case of voltage drops below approx. 18 V.

9.2.2 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), the following LED behavior is visible:

- The dedicated red port DIA indicator is active when an error is detected.

9.2.3 Overload/short circuit of the I/O port Ch. A as actuator outputs

The digital outputs on Channel A and B are protected against short circuits and overloads. In case of a fault, the output is automatically switched to "inactive" and then cyclically switched back to "active" when the default setting is used (*DO Restart Mode* Parameter = “Automatic Restart after Failure”).

In *DO Restart Mode* Parameter = “Restart after Output Reset”, the output must be set to "low" via PLC, before the output can be set again to "high".

When an output channel is activated (rising edge of the channel state), the channel errors are filtered for the period that you set using the *Surveillance-Timeout* parameter during the configuration of the device. The value of this parameter can range from 0 to 255 ms; the factory setting is 80 ms.

The filter is used to avoid premature error messages when a capacitive load is activated.

- The dedicated red port DIA indicator is active when an error is detected.

Attention: For variant 0980 XSL 3903-121-007D-01F, the digital outputs are supplied as follows:

- "X1 .. X8 / Channel A" are supplied by the U_S power
- "X1 .. X4 / Channel B" are supplied by the U_S power
- "X5 .. X8 / Channel B" are supplied by the U_{AUX} power

9.2.4 Internal module error

Internal module error states (e.g. internal abnormal states) will be reported by the following diagnostic message.

9.3 Emergency messages

When parametrized, the Device sends emergency messages to the Master in case of a detected diagnosis on the Device. The coding of the first part of the emergency messages alludes to the CiA 301 and CiA 401 specifications. The second part of the emergency messages is the known error register, which can be also added to the cyclic input data via PDO.

The emergency message has a format of 8 Bytes and is coded as follows:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Emergency error code		Error register CoE 0x1001	Diagnostic register				

Table 13: Byte content of the emergency message

Emergency error code	B7	B6	B5	B4	B3	B2	B1	B0	Error description
0x0000	0	0	0	0	0	0	0	0	No error
0x2300	–	0	0	0	0	0	1	1	Sensor short circuit
0x3100	–	0	0	0	0	1	–	1	U_S Voltage error
0x3300	–	0	0	0	0	1	–	0	U_L Voltage error
0xF000	1	0	0	0	0	0	–	1	Additional function forcing
0xFF00	1	0	0	0	0	0	–	1	Additional function parameter error

Table 14: Content of the error register (CoE register 0x1001)

10 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.

10.1 MQTT

The MQTT (Message Queueing 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.

10.1.1 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 request. For more information see chapter MQTT configuration - Quick start guide on page 93.

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
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 15: 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 78.

10.1.2 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 a string and they are allowed to contain slashes (`/`). In topic filters, there also wildcard symbols like e.g. (#) allowed.

10.1.2.1 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 16: Base topic variables on page 78.

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
ip0 ip1 ip2 ip3	IP address octets

Table 16: 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, 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 data.

Table 17: 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, state	8	Interval
[base-topic]/process/gateway	All Digital IN/OUT	1	Interval
[base-topic]/process/port/n	Digital IN/OUT per port, pdValid	8	Interval

Table 18: 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 19: Use case examples

10.1.2.2 Publish topic

Overview of all publish JSON data for the defined topics:

Identity/gateway	
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
family	json_string
location	json_string
country	json_string
fax	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 20: Identity/gateway

Config/gateway				
Key	**Data type**	**Range**	**Default value**	**Remarks**
fieldbus_protocol	json_string	PROFINET EtherNet/IP EtherCAT® Modbus TCP CC-Link IE Field Basic		
network_configuration	json_string	PROFINET: ▶ DCP ▶ Manual EtherNet/IP: ▶ Manual ▶ Rotary ▶ DHCP EtherCAT®: ▶ Manual Modbus TCP: ▶ Manual ▶ DHCP ▶ Rotary CC-Link IE Field Basic: ▶ Manual ▶ Rotary		
rotary_switches	json_integer	0 .. 999		
ip_address	json_string		192.168.1.1	
subnet_mask	json_string		255.255.255.0	
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	

Config/gateway				
Key	**Data type**	**Range**	**Default value**	**Remarks**
fast_startup	json_boolean	true / false	false	PROFINET and EIP only

Table 21: Config/gateway

Status/gateway				
Key	**Data type**	**Range**	**Default value**	**Remarks**
protocol	json_string	PROFINET: ▶ UNKNOWN ▶ OFFLINE ▶ STOP ▶ IDLE ▶ OPERATE EtherNet/IP: ▶ CONNECTED ▶ DISCONNECTED EtherCAT®: ▶ PREOP ▶ SAFEOP ▶ OP ▶ INIT ▶ UNKNOWN Modbus TCP: ▶ No Connections ▶ Connected CC-Link IE Feld Basic: ▶ ON ▶ STOP ▶ DISCONNECTED ▶ ERROR		
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 22: Status/gateway

Process/gateway				
Key	**Data type**	**Range**	**Default value**	**Remarks**
Input_data	json_integer[]			
output_data	json_integer[]			

Table 23: Process/gateway

Identity/port/1 .. 8				
Key	**Data type**	**Range**	**Default value**	**Remarks**
port	json_integer	1 .. 8		
type	json_string	Digital Input DIO Digital Output DIO Pin 4 Only DI Pin 4 Only DO Pin 4 Only Not available Unknown		
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	Digital Input Digital Output DIO Digital Input/Output Auxiliary Power Auxiliary with DO Not available Unknown		
channel_chb	json_string	Digital Input Digital Output DIO Digital Input/Output Auxiliary Power Auxiliary with DO Not available Unknown		

Table 24: Identity/port/1 .. 8

Config/port/1 .. 8				
Key	**Data type**	**Range**	**Default value**	**Remarks**
port	json_integer	1 .. 8		
direction_cha	json_string	Output Input Inactive Auxiliary Power DIO Unknown		
direction_chb	json_string	Output Input Inactive Auxiliary Power DIO Unknown		
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		
failsafe_cha	json_string	set_low set_high hold_last	set_low	
failsafe_chb	json_string	set_low set_high hold_last	set_low	
surveillance_timeout_cha	json_integer	0 .. 255	80	

Config/port/1 .. 8				
Key	**Data type**	**Range**	**Default value**	**Remarks**
surveillance_timeout_chb	json_integer	0 .. 255	80	
io_mapping_cha	json_integer	0 .. 15	channel number	16DIO only
io_mapping_chb	json_integer	0 .. 15	channel number	16DIO only

Table 25: Config/port/1 .. 8

Status/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 26: Status/port/1 .. 8

10.1.2.3 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 28: Force object: Digital on page 90)		

Table 27: 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 28: Force object: Digital

[…]/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 30: Config object: Portmode on page 91)		
ip_address	string	"192.168.1.5"	
subnet_mask	string	"255.255.255.0"	
gateway	string	"192.168.1.100"	

Table 29: Config object properties

For the *Conig 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		

Table 30: 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 31: 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).

10.1.3 MQTT configuration - Quick start guide

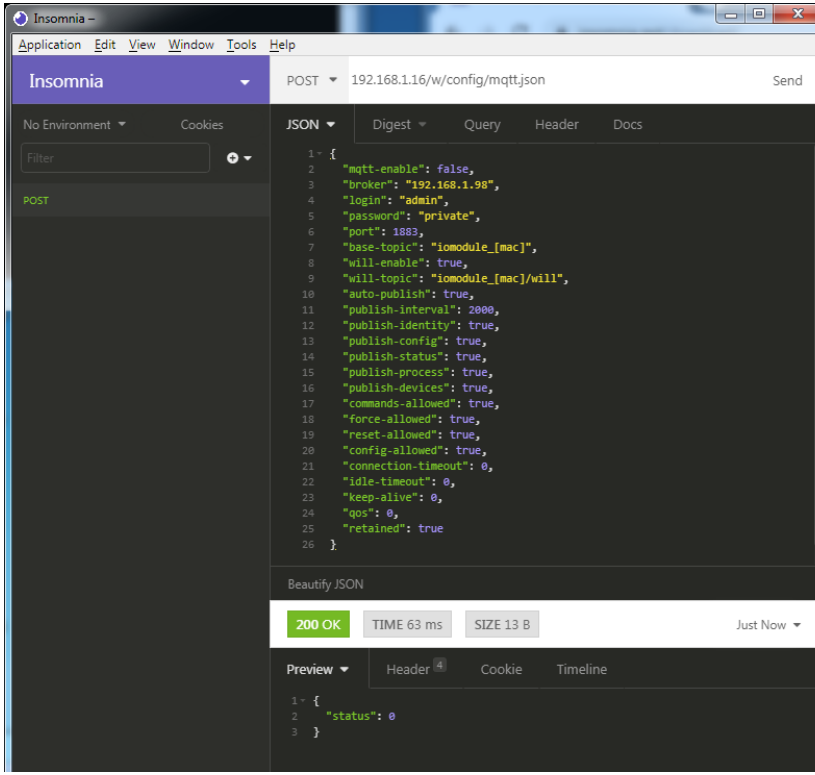
Attention: Lumberg Automation™ is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

10.1.3.1 MQTT configuration via JSON

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

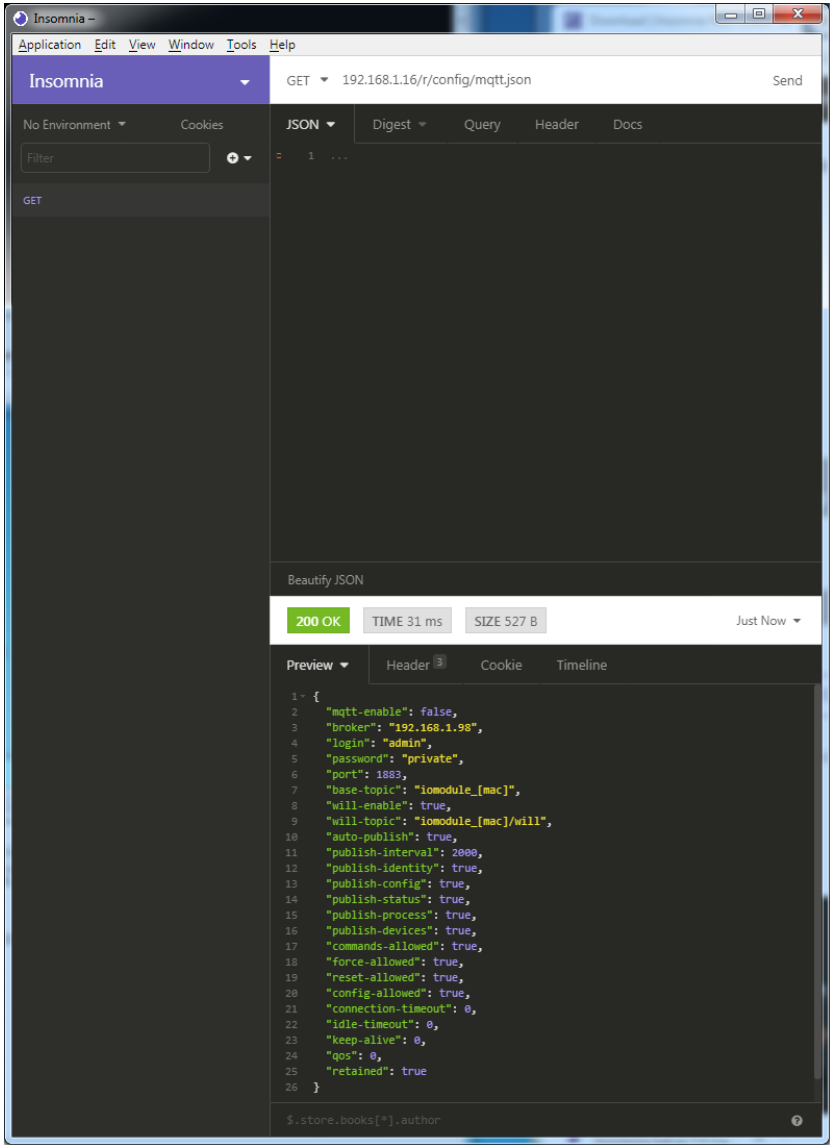
2. Configure MQTT:

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



3. Read MQTT:

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



10.2 OPC UA

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.

10.2.1 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 request.

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.

Tree overview of OPC UA objects:

- Gateway
  - Identity
    - Name
    - MAC
    - Ordering Number
    - Production Date
    - Capabilites
    - Firmware Versions
  - Status (r)
    - US present
    - UL present
    - US diag
    - UL diag
    - US Voltage
    - UL Voltage
    - IME
    - Forcemode Diag
    - Rotary positions
  - Forcing (r)
    - Forcing active
    - Forcing client
    - OwnForcing flag
  - Config (rw)
    - IP Config
    - suppressActuatorDiagWithoutUL
    - suppressUSDiag
    - suppressULDiag
    - quickConnect
  - Process (r)
    - Digital Inputs
    - Digital Outputs
    - Producing Data (to PLC)
    - Consuming Data (from PLC)
    - Valid masks
  - Commands (w)
    - Restart
    - Factory Reset
    - Forcemode enable

- Ports
  - Port *n ("X1"-"X8")*
    - Identity
      - Port Name
      - Port Type
    - Channel *m* ("Pin 4" / "Pin 2")
      - Identity (r)
        - Channel Name
        - Channel Type
        - MaxOutputCurrent
      - Status (r)
        - Actuator Diag
        - Actuator Voltage
        - Actuator Current
        - Channel Failsafe flag
      - Config (rw)
        - Surveillance Timeout
        - Failsafe Config
        - Channel Direction
        - Channel Current Limit
        - Auto Restart
        - InputFilterTime
        - InputLogic
      - Process (r)
        - Output Bit
        - Input Bit
        - Consuming Bit
        - Producing Bit
      - Forcing (rw)
        - Force channel on/off
        - Force value on/off
        - Simulate channel
        - SImulate value
    - Status (r)
      - Pin 1 Short Circuit Dia
      - Pin 1 Voltage
      - Pin 1 Current
    - Config (rw)
      - Pin 1 Current limit

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"}]}
```

10.2.1.1 Gateway objects

Identity

Name	Data type	Example
Device Name	UA_STRING	
Device ID	UA_STRING	
MAC address	UA_STRING	
Ordering Number	UA_STRING	
Serial Number	UA_STRING	
Production Date	UA_STRING	
Hardware Version	UA_STRING	
App Firmware Version	UA_STRING	
Fieldbus Firmware Version	UA_STRING	
IO Firmware Version	UA_STRING	
Running Fieldbus	UA_STRING	
Forcemode supported	UA_BOOLEAN	Forcing supported by module variant

Status (read)

Name	Data type	Unit	Example
US present	UA_BOOLEAN		
UL present	UA_BOOLEAN		
US diagnosis	UA_BOOLEAN		
UL diagnosis	UA_BOOLEAN		
Internal Module Error diag	UA_BOOLEAN		

Name	Data type	Unit	Example
Forcemode diag	UA_BOOLEAN		
US voltage	UA_DOUBLE	V	23.2
UL voltage	UA_DOUBLE	V	22.9
Rotary position	UA_UINT16		343

Forcing (read)

Name	Data type	Example
Forcing active	UA_BOOLEAN	
Forcing client	UA_STRING	if forcemode is not active, string is empty
Own Forcing	UA_BOOLEAN	Indicates if OPC UA is currently forcing
Forcing possible	UA_BOOLEAN	true if forcing by OPC UA is possible
Forcemode lock	UA_BOOLEAN	Forcing locked by PLC

Config (read + write)

Name	Data type	Example
IP address	UA_STRING	
Subnet Mask	UA_STRING	
Default Gateway IP	UA_STRING	
Suppress US diag	UA_BOOLEAN	
Suppress UL diag	UA_BOOLEAN	
Supppres Actuator Diag w/o UL	UA_BOOLEAN	
QuickConnect	UA_BOOLEAN	

Process (read)

Name	Data type	Example
Input Data	UA_UINT16	ioInput for all channels
Output Data	UA_UINT16	ioOutput for all channels
Consuming Data	UA_UINT16	Data from the PLC to the device
Producing Data	UA_UINT16	Data from the device to the PLC

Commands (write)

Name	Arguments	Return	Example
Restart	void	UA_INT32	
Factory reset	void	UA_INT32	
Forcemode enable	void	UA_INT32	
Forcemode disable	void	UA_INT32	

10.2.1.2 Ports objects

Identity

Name	Data type	Example
Name	UA_STRING	"X1"
Type	UA_STRING	"DIO"

Channel *m* ("Pin 4" / "Pin 2")

See details in Channel objects on page 102.

Status (read)

Name	Data type	Unit	Example
Sensor Diag	UA_BOOLEAN		
Pin 1 Voltage	UA_DOUBLE	V	22.5
Pin 1 Current	UA_INT16	mA	1900

Config (read + write)

Name	Data type	Unit	Example
Pin 1 Current Limit	UA_INT16	mA	1000

10.2.1.3 Channel objects

Identity (read)

Name	Data type	Unit	Example
Name	UA_STRING		"X1A"
Type	UA_STRING		"DIO"
MaxOutputCurrent	UA_INT16	mA	1300

Status (read)

Name	Data type	Unit	Example
Actuator Diag	UA_BOOL		
Actuator Voltage	UA_DOUBLE	V	23.5
Actuator Current	UA_INT16	mA	800
Channel Failsafe	UA_BOOL		

Config (read + write)

Name	Data type	Unit	Example / Remarks
Surveillance Timeout	UA_UINT8	ms	80 ms
Failsafe Config	UA_ENUMERATION		Low Hi Hold Last
Channel DIrection	UA_ENUMERATION		DIO Input Output Inactive
Channel Current Limit	UA_UINT16	mA	2000 mA
Auto Restart	UA_BOOL		

Name	Data type	Unit	Example / Remarks
InputFilterTime	UA_UINT8	ms	3ms
InputLogic	UA_ENUMERATION		NO NC

Process (read)

Name	Data type	Example / Remarks
Output	UA_BOOLEAN	Output type channels only.
Input	UA_BOOLEAN	Input type channels only.
Consuming	UA_BOOLEAN	
Producing	UA_BOOLEAN	

Forcing (read + write)

Name	Data type	Example / Remarks
Force channel	UA_BOOLEAN	Enable forcing with the current force value or disable forcing for this channel. Output type channels only.
Force value	UA_BOOLEAN	When changed by the user it will start forcing with the new value if forcing is enabled for opcua. Output type channels only.
Simulate channel	UA_BOOLEAN	Enable simulation with the current force value or disable simulation for this channel. Input type channels only.

Name	Data type	Example / Remarks
Simulate value	UA_BOOLEAN	When changed by the user it will start simulation with the new value if forcing is enabled for opcua. Input type channels only.

10.2.2 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.

The following figures illustrate the OPC UA address space of the LioN-X devices. The objects and information displayed depend on the device variant used.

10.2.3 OPC UA configuration - Quick start guide

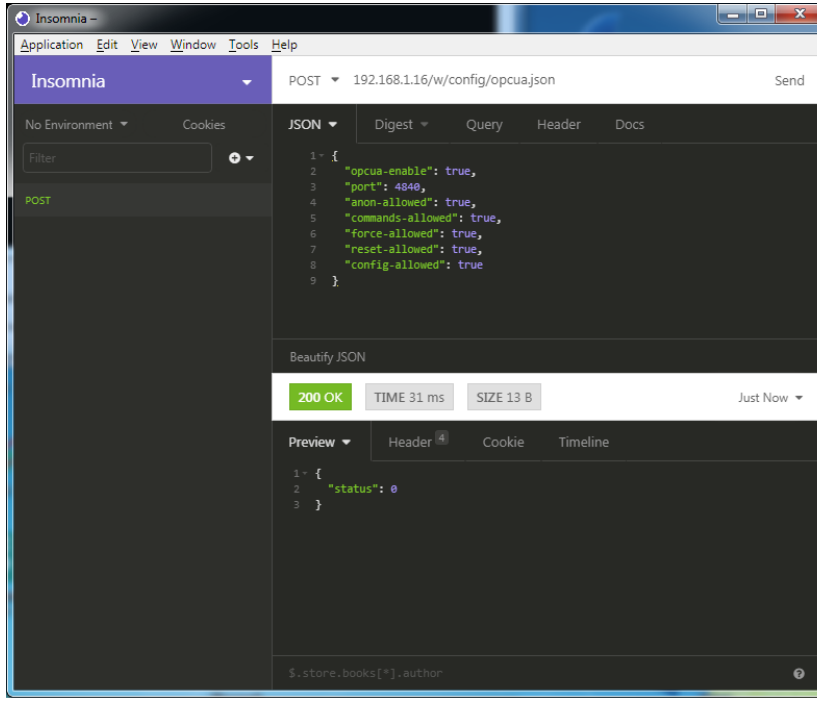
Attention: Lumberg Automation™ is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

10.2.3.1 OPC UA configuration via JSON

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

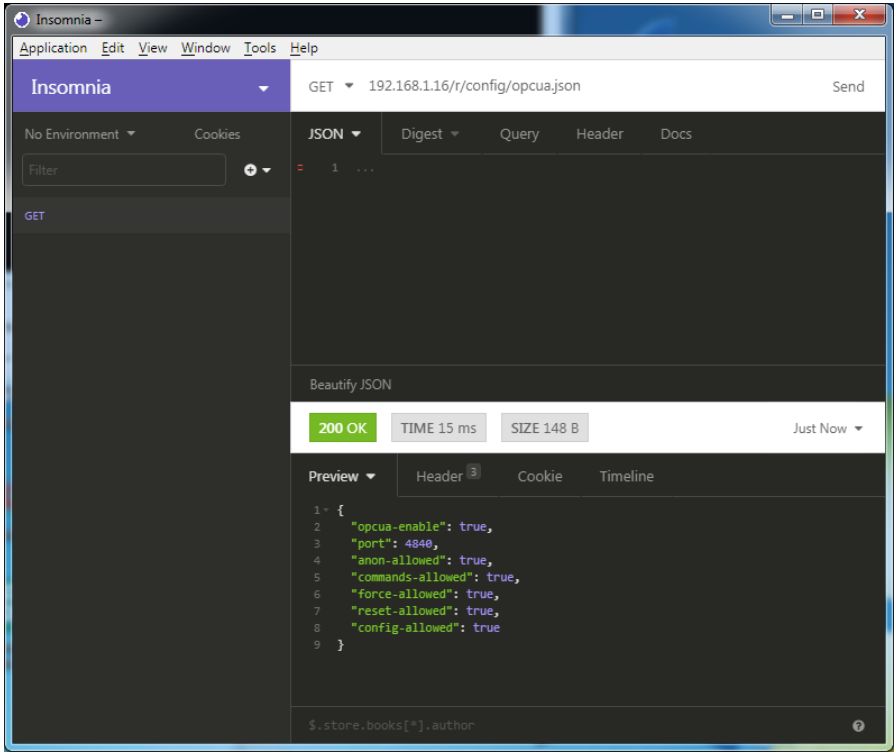
2. Configure OPC UA:

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



3. Read OPC UA:

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



10.3 REST API

The Representational State Transfer – Application Programming Interface (REST API) is a programmable interface which uses HTTP 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 multiprotocol variants, the REST API can also be used to write configuration and forcing data.

The customized Belden REST API is described in the following chapters.

10.3.1 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.

10.3.2 Structure

Name	Data type	Description	Example
name	string	Device name	"0980 XSL 3912-121-007D-00F"
order-id	string	Ordering number	"935 700 001"
fw-version	string	Firmware version	"V.1.1.0.0 - 01.01.2021"
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		
channels	Array of CHANNEL (16)	Basic information about all input/output channels		

Name	Data type	Description	Example
CHANNEL Object			
name	string	Name of channel	
type	number	Hardware channel type as number: 0 = DIO 1 = Input 2 = Output 3 = Input/Output 4 = Channel not available 5 = Channel not available 6 = Channel not available 7 = Channel not available 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 = Channel not available 4 = Deactivated 5 = Channel not available	
config_text	string	Textual representation of the current config	
inputState	boolean	Input data (producing data) bit to the PLC	
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	

Name	Data type	Description	Example
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 port type	
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"
ds_fault	number	Data storage error number	
ds_fault_text	string	Textual data storage error.	
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"
message	string	Error message	"Supply Voltage fault"
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	
AuthPossible	boolean	True, if the JSON Interface can obtain forcing autorization	
ownForcing	boolean	True, if forcing is performed by REST API at the moment	
currentClient	string	Current forcing client identifier	

Name	Data type	Description	Example
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.	

10.3.3 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)		

Table 32: Root object

Property	Data type	Example values	Remarks
port	integer	0..7	
channel	integer	"a","b"	optional default is "a"
direction	string	"dio","di","do", "off", "aux"	
inlogica	string	"no","nc"	
inlogicb	string	"no","nc"	

Table 33: 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 34: Digital object

10.4 CoAP server

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 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 multiprotocol variants provide CoAP server functionalities via a REST API interface over UDP.

10.4.1 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 request. For more information see chapter CoAP configuration - Quick start guide on page 117.

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 35: 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"}]}
```

10.4.2 REST API access via CoAP

A connection to the CoAP server running on the LioN-X multiprotocol 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	

Table 36: REST API access via CoAP

10.4.3 CoAP configuration - Quick start guide

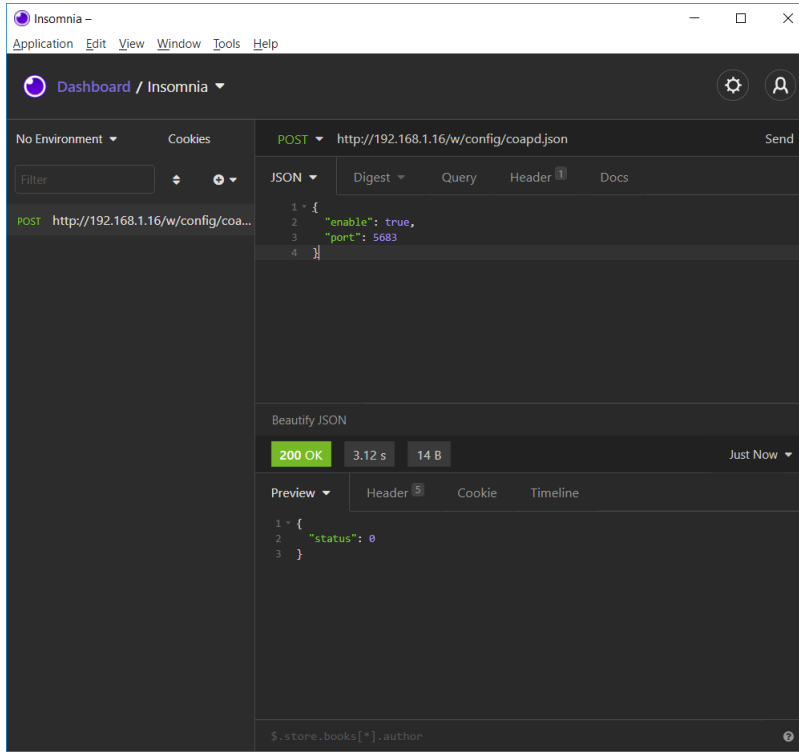
Attention: Lumberg Automation™ is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

10.4.3.1 CoAP configuration via JSON

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

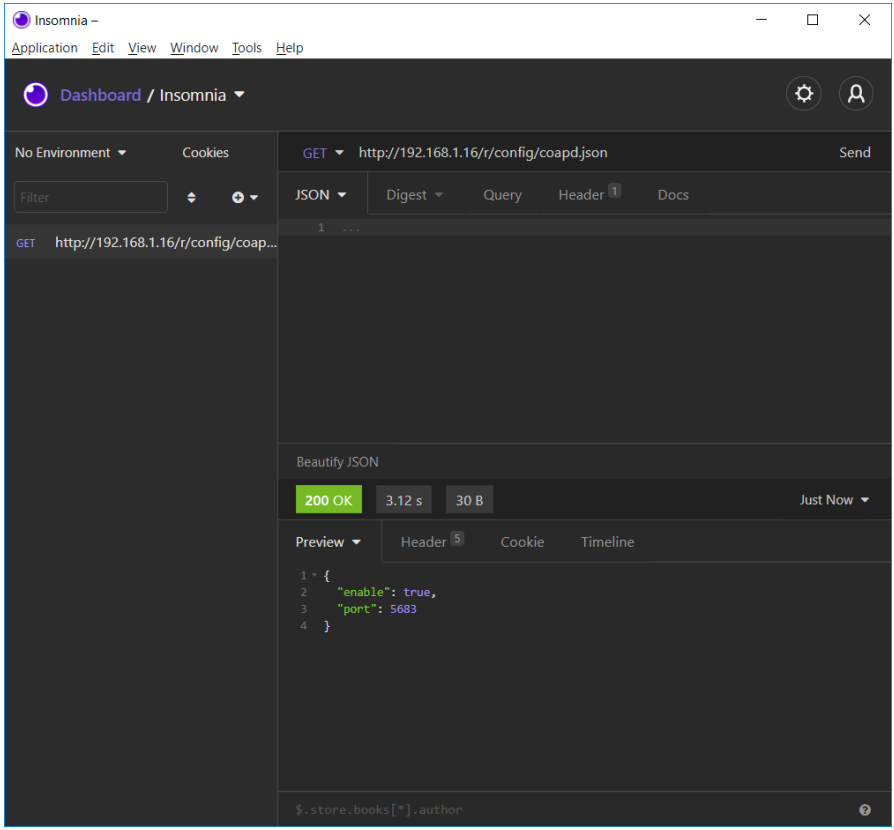
2. Configure CoAP:

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



3. Read CoAP configuration:

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



10.5 Syslog

The LioN-X multiprotocol 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 will not store any message permanently.

10.5.1 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 request. For more information see chapter Syslog configuration - Quick start guide on page 122.

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 37: 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"}]}
```

10.5.2 Syslog configuration - Quick start guide

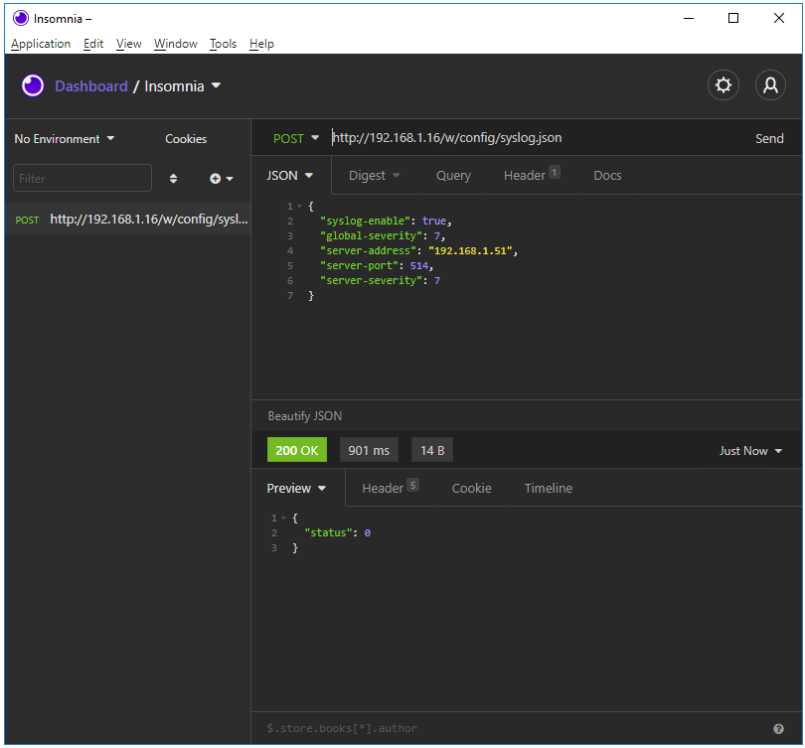
Attention: Lumberg Automation™ is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

10.5.2.1 Syslog configuration via JSON

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

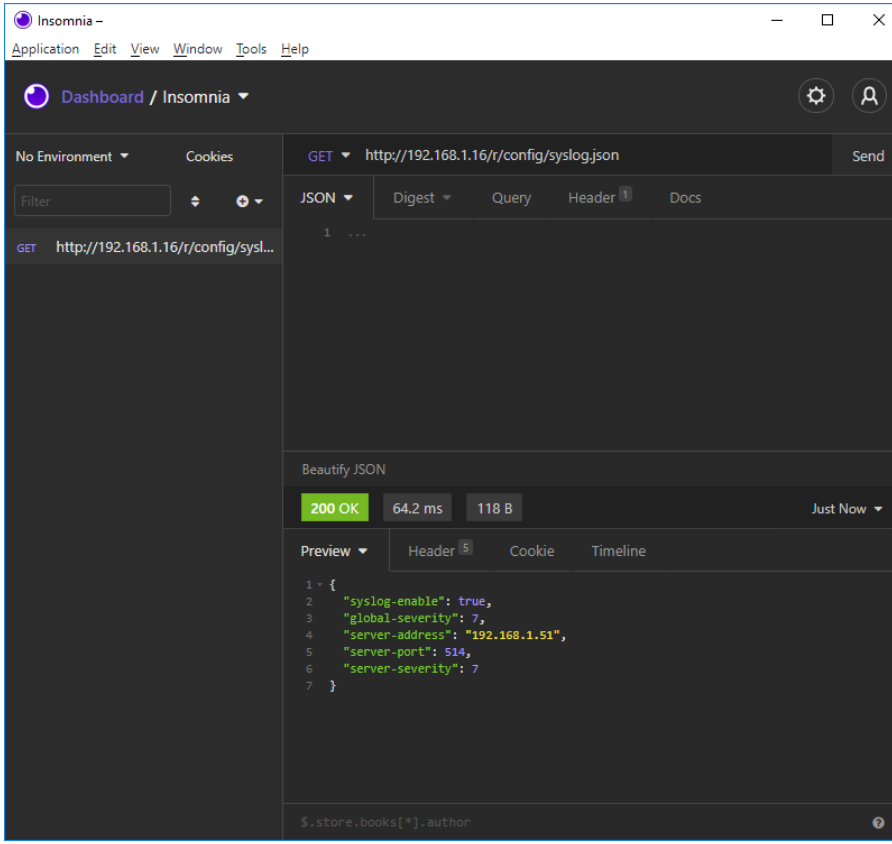
2. Configure Syslog:

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



3. Read Syslog configuration:

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



10.6 Network Time Protocol (NTP)

The LioN-X multiprotocol 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.)

10.6.1 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 request. For more information see chapter NTP configuration - Quick start guide on page 126.

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 38: 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"}]}
```

10.6.2 NTP configuration - Quick start guide

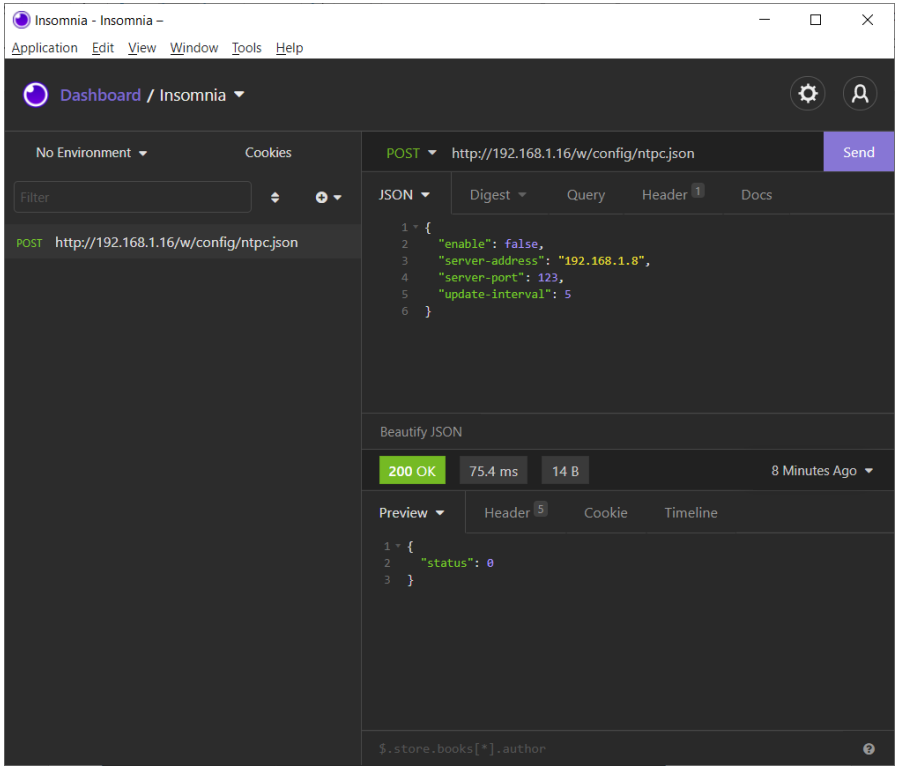
Attention: Lumberg Automation™ is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

10.6.2.1 NTP configuration via JSON

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

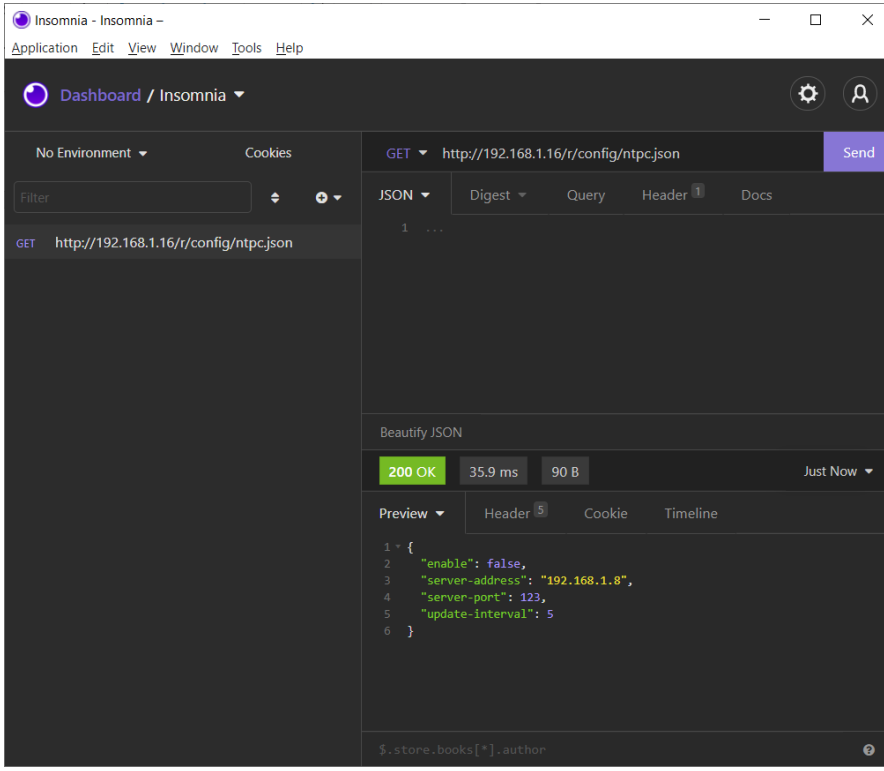
2. Configure NTP:

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



3. Read NTP configuration:

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



11 The integrated Web server

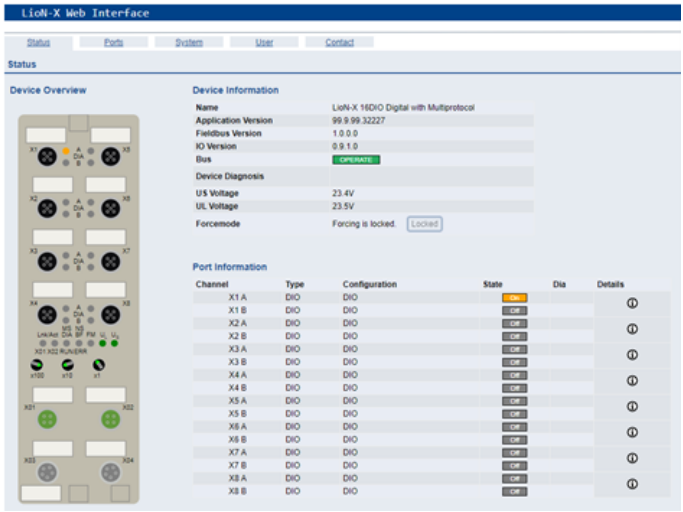
All device variants are equipped with an integrated Web server which makes functions for the device configuration and the display of status and diagnostic information available via a Web interface.

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://" 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.

11.1 LioN-X 0980 XSL... variants

11.1.1 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.

11.1.2 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.

11.1.3 The System page

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

Restart Device

The module initializes a software reset.

Reset to Factory Settings

The module restores to the default factory settings.

IP Settings

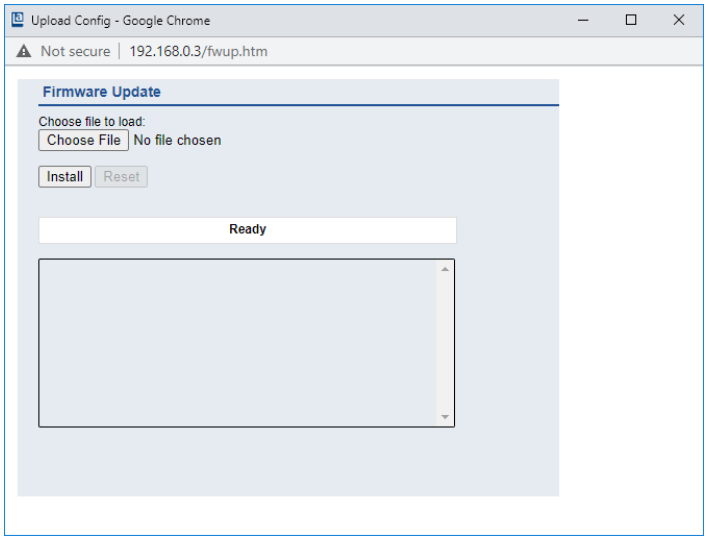
Use this parameter to change the current IP address of the module.

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.

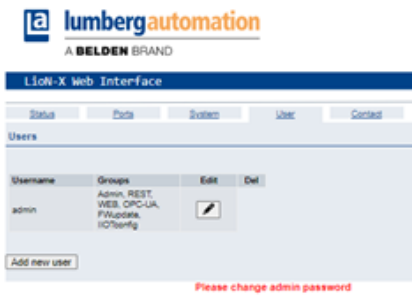
Firmware Update

The module initializes a Firmware 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.



11.1.4 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

12 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 catalog.belden.com.

12.1 General

Protection class (Only applies if the connectors are screwed together or if protective caps are used.)[1]	IP65 IP67 IP69K	
Ambient temperature (during operation and storage)	0980 XSL 3x00-121... 0980 XSL 3x01-121... 0980 XSL 3x03-121...	-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 39: General information

[1] Not under UL investigation.

12.2 EtherCAT® protocol

Protocol	EtherCAT® (ETG.1000 V1.2)
ESI file	LumbergAutomation-LioN-X-Digital.xml
Transmission rate	100 Mbit/s, full duplex
Type of addressing	Auto-increment addressing, Fixed addressing
Min. cycle time	1 ms
Vendor ID	$16A_H$
Device ID	0x0400 (same for all LioN-X devices)
Mailbox protocols	CanOpen over EtherCAT® (CoE) File access over EtherCAT® (FoE) Ethernet over EtherCAT® (EoE)
Supported Ethernet protocols	Ping ARP HTTP TCP/IP
Switch functionality	Integrated
EtherCAT® interface Port	2x M12 sockets 4-pin, D-coded (see pin assignments)

Table 40: EtherCAT® protocol

12.3 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 (Pin 1)	0980 XSL 3x00-121... 0980 XSL 3x01-121...	Port X1 .. X8 (Pin 1)	max. 4 A per port (at $T_{ambient}$ = 30° C)
	0980 XSL 3x03-121...	Port X1 .. X4 (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 41: Information on the power supply of the module electronics/ sensors

Attention: Do not exceed the following maximum currents for the sensor supply:

- Max. 4.0 A per port

- Max. 5.0 A for each port pair X1/X2, X3/X4, X5/X6, X7/X8
- Max. 9.0 A in total for the whole port group X1 .. X8

Pay attention to the derating!

12.4 Power supply of the actuators

Port X03, X04	M12_L-coded Power, connector/socket, 5-pole Pin 2 / Pin 4
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.

Table 42: Information on the power supply of the actuators

12.5 I/O ports

0980 XSL 3900-121...	Ports X1 .. X8	DI, DO	M12 socket, 5-pin
0980 XSL 3901-121...	Ports X1 .. X8	DI	
0980 XSL 39x3-121...	Ports X1 .. X4	DI	
	Ports X5 .. X8	DO	

Table 43: I/O ports: Overview of functions

12.5.1 Digital inputs

Input connection	0980 XSL 3900-121...		Type 3 as per IEC 61131-2
	0980 XSL 3901-121...		
	0980 XSL 39x3-121...		
Nominal input voltage	24 V DC		
Input current	Typically 3 mA		
Channel type	Normally open, p-switching		
Number of digital inputs	0980 XSL 3900-121...	X1 .. X8	16
	0980 XSL 3901-121...		
	0980 XSL 39x3-121...	X1 .. X4	8
Status indicator	yellow LED for Channel A (Pin 4) white LED for Channel B (Pin 2)		
Diagnostic indicator	red LED per port		

Table 44: I/O ports configured as digital input

12.5.2 Digital outputs

Attention: Do not exceed the following maximum currents for the sensor supply:

- Max. 2.0 A per port
- Max. 5.0 A for each port pair X1/X2, X3/X4, X5/X6, X7/X8
- Max. 9.0 A in total for the whole port group X1 .. X8 (X5 .. X8 at 8DI8DO devices)

  Pay attention to the derating!

Output type	normally open, p-switching	
Nominal output voltage per channel Signal status “1” Signal status “0”	 min. (U_L -1 V) max. 2 V	
Max. output current per device	0980 XSL 3900-121...	9 A
	0980 XSL 39x3-121...	9 A
Max. output current per channel	0980 XSL 3900-121... (X1 .. X8)	2 A
	0980 XSL 39x3-121... (X5 .. X8)	2 A
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 3900-121... (X1 .. X8)	16
	0980 XSL 39x3-121... (X5 .. X8)	8
Status indicator	yellow LED per output Channel A (Pin 4) white LED per output Channel B (Pin 2)	
Diagnostic indicator	red LED per channel	

Table 45: I/O ports configured as digital output

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

12.6 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	Yellow	Status of digital input or digital output on pin 4 line "on".
	Red	Short circuit on pin 4 line. / Overload or short circuit on L+ (pin 1) line / communication error
	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.
BF	Red	Bus fault. No configuration, no or slow physical connection.
	Red flashing at 2 Hz	Link exists but no communication link to the EtherCAT® controller.

LED	Color	Description
	OFF	EtherCAT® controller has established an active connection to the device.
DIA	Red	EtherCAT® module diagnostic alarm active.
	Red flashing at 1 Hz	Watchdog time-out; fail safe mode is active.
	Red double flash	Firmware update
	OFF	None of the above conditions.

Table 46: Information on the LED colors

12.7 Data transfer times

The following tables give an overview of the internal data transfer times of LioN-X.

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

- **PLC to DO:** Transfer of a changed PLC output data to the digital output channel.
- **DI to PLC:** Transfer of a changed digital input signal on digital input channel to PLC.
- **Round-trip time (RTT):** Transfer of a changed PLC output data to digital output. The digital output is connected to a digital input. Transfer of the changed digital input signal on the channel 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.

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.

Use case 1:

LioN-X Digital-I/O configuration with enabled Web interface and *disabled* IIoT protocols

16DIO variant (0980 XSL 3900-121-007D-01F):

Data direction	Data transfer time in ms		
	Minimum	Average	Maximum
PLC to DO	2.2	3.6	5.0
DI to PLC	3.1	3.0	4.7
RTT	6.0	7.6	9.0

8DI/8DO variant without galvanic isolation (0980 XSL 3913-121-007D-01F):

Data direction	Data transfer time in ms		
	Minimum	Average	Maximum
PLC to DO	1.9	3.2	4.7
DI to PLC	2.1	2.6	3.1
RTT	4.0	5.8	7.0

8DI/8DO variant with galvanic isolation (0980 XSL 3903-121-007D-01F):

Data direction	Data transfer time in ms		
	Minimum	Average	Maximum
PLC to DO	2.2	3.6	5.3
DI to PLC	3.3	4.0	4.6
RTT	6.0	7.6	9.0

Use case 2:

LioN-X Digital-I/O configuration with enabled Web interface and *enabled* IIoT protocols

16DIO variant (0980 XSL 3900-121-007D-01F):

Data direction	Data transfer time in ms		
	Minimum	Average	Maximum
PLC to DO	3.4	5.1	7.6
DI to PLC	5.8	6.4	7.6
RTT	10.0	11.5	14.0

8DI/8DO variant without galvanic isolation (0980 XSL 3913-121-007D-01F):

Data direction	Data transfer time in ms		
	Minimum	Average	Maximum
PLC to DO	3.2	4.8	7.1
DI to PLC	3.3	3.8	4.3
RTT	7.0	8.6	11.0

8DI/8DO variant with galvanic isolation (0980 XSL 3903-121-007D-01F):

Data direction	Data transfer time in ms		
	Minimum	Average	Maximum
PLC to DO	3.5	5.2	7.6
DI to PLC	5.7	6.4	7.1
RTT	10.0	11.6	14.0

13 Accessories

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

http://www.beldensolutions.com