Manual

PROFINET

UNITRONIC ACCESS IO-Link Master

Multi-protocol:
MP08IOLA08DIO (8 × IO-Link Class A)

Single-protocol:
PN08IOLA08DIO (8 × IO-Link Class A)



Contents

1 About this manual 9

1.1 General information 9
1.2 Explanation of symbols 10
1.2.1 Use of danger information 10
1.2.2 Use of general information 10
1.3 Version information 10

2 Safety instructions 11

2.1 Intended use 11
2.2 Qualified personnel 12

3 Designations and synonyms 13

4 System description 17

4.1 About UNITRONIC® ACCESS 17
4.2 Device variants 18
4.3 I/O port overview 20

5 Overview of product features 22

5.1 PROFINET product features 22
5.2 I/O port features 26
5.3 Integrated Web server 27
5.4 Security features 28
5.5 Other features 29

6 Assembly and wiring 30

6.1 General information 30
6.2 Outer dimensions 31
6.2.1 UNITRONIC® ACCESS multi-protocol variants 31
6.2.2 UNITRONIC® ACCESS single-protocol variants with PROFINET 32
6.2.3 Notifications 33
6.3 Port assignments 34
6.3.1 Ethernet ports, M12 socket, 4-pin, D-coded 34
6.3.2 Power supply with M12 power L-coded 35
6.3.2.1 IO-Link Master with Class A ports 36
6.3.3 I/O ports as M12 sockets 36
6.3.3.1 IO-Link ports (Class A) 37

7 Starting operation 38

7.1 GSDML file 38
7.2 MAC addresses 38
7.3 State on delivery 39
7.4 Setting the rotary encoding switches 40
7.4.1 PROFINET 43
7.4.2 Factory reset 43
7.5 SNMPv1 43

8 Configuration and operation with SIEMENS TIA Portal® 44

8.1 Assigning a device name and IP address 46
8.2 Configuring the IO-Link channels 49
8.2.1 Deleting the configuration of a specified IO-Link channel 50
8.2.2 Creating an IO-Link channel configuration 51

8.3 Parameterization of the Status/Control Module 54
8.3.1 General Device Settings 55
8.3.2 I/O mapping configuration of Status/Control data 57
8.3.3 General Diagnostic Settings 60
8.3.3.1 Diagnostic settings for modules with IO-Link Class A ports 60
8.4 Parameterization of the I/O ports X1 .. X8 61
8.4.1 Enhanced port parameters 63
8.4.2 Failsafe port parameters for Ch. A in IO-Link mode 70
8.4.3 Standardized port parameters 73
8.5 IO-Link Device Parameterization 82
8.5.1 SIEMENS IO-Link library 82
8.5.1.1 SIEMENS function block FB50001 – write example 84
8.5.1.2 SIEMENS function block FB50001 – read example 86
8.5.2 SIEMENS WRREC and RDREC 88
8.5.2.1 Write sequence 88
8.5.2.2 Read sequence 95
8.5.2.3 Error PDU for the Read/Write sequence 102
8.6 Media Redundancy Protocol (MRP) 104
8.7 Identification & maintenance (I&M) 106
8.7.1 Supported I&M features 106
8.7.1.1 I&M data of the PN-IO Device 106
8.7.1.2 I&M data of the IOL-Master proxy (Status/Control Module) 109
8.7.1.3 I&M data of the IOL-Device proxy 110
8.7.2 Reading and writing I&M data 113
8.7.2.1 I&M Read Record 114
8.7.2.2 I&M Write Record 117
8.8 Fast Start Up (FSU) / Prioritized Startup 118
8.9 Suspend / Resume of IO-Link port operation 120
8.9.1 Automatic tool changer application use case 120
8.9.2 Concept 120
8.9.3 Use cases 122
8.9.4 Suspend and Resume cycle 123
8.9.4.1 Write Record Suspend – port command 123
8.9.4.2 Read Record Suspend – port status 124

8.9.4.3 Write Record Resume – port command 126
8.9.4.4 Read Record Resume – port status 127

9 Process data assignment 130

9.1 Process data Status/Control Module, I/O system 1.1 130
9.1.1 Status/Control data with Bit Mapping 132
9.1.1.1 Mode 5 132
9.1.1.2 PROFINET channel diagnostics mapping 132
9.2 Process data IO-Link ports, Slot 1.2 – 1.9 132

10 Diagnostics 136

10.1 Detailed diagnostics description 136
10.1.1 Detected error of the system/sensor power supply U_S 136
10.1.2 Error of the actuator power supply U_L 137
10.1.3 Overload/short-circuit of the I/O port sensor supply outputs 138
10.1.4 Overload/short circuit of the I/O port Ch. A as actuator outputs 139
10.1.5 Overload/short circuit of the I/O port Ch. B as actuator outputs 141
10.1.6 IO-Link C/Q error 143
10.1.7 Generic parameter error 144
10.1.8 I/O mapping parameter error 144
10.1.9 Process data mismatch error 145
10.1.10 Force mode diagnostic 145
10.1.11 Internal module error detected 145
10.2 Table of IO-Link Master diagnostic codes 147
10.3 IO-Link Device diagnostics in PROFINET 149
10.4 Table of IO-Link Device diagnostic codes 151

11 IIoT functionality 154

11.1 MQTT 155

11.1.1 MQTT configuration 155
11.1.2 MQTT topics 158
11.1.2.1 Base topic 158
11.1.2.2 Publish topic 161
11.1.2.3 Command topic (MQTT Subscribe) 166
11.1.3 MQTT configuration - Quick start guide 170
11.1.3.1 MQTT configuration via JSON 170
11.2 OPC UA 172
11.2.1 OPC UA configuration 173
11.2.2 OPC UA address space 175
11.2.3 OPC UA configuration - Quick start guide 176
11.2.3.1 OPC UA configuration via JSON 176
11.3 REST API 178
11.3.1 Standard device information 180
11.3.2 Structure 181
11.3.3 Configuration and forcing 187
11.3.4 Reading and writing ISDU parameters 190
11.3.4.1 Reading ISDU 190
11.3.4.2 Writing ISDU 192
11.3.5 Upload and process an IODD file 194
11.3.6 Example: Reading ISDU 198
11.3.7 Example: Writing ISDU 198
11.4 CoAP server 199
11.4.1 CoAP configuration 199
11.4.2 REST API access via CoAP 200
11.4.3 CoAP configuration - Quick start guide 203
11.4.3.1 CoAP configuration via JSON 203
11.5 Syslog 205
11.5.1 Syslog configuration 205
11.5.2 Syslog configuration - Quick start guide 208
11.5.2.1 Syslog configuration via JSON 208
11.6 Network Time Protocol (NTP) 210
11.6.1 NTP configuration 210
11.6.2 NTP configuration - Quick start guide 212
11.6.2.1 NTP configuration via JSON 212

12 The integrated Web server 214

12.1 UNITRONIC® ACCESS MP08... variants 215
12.1.1 The Status page 215
12.1.2 The Ports page 216
12.1.2.1 IODD upload 216
12.1.3 The System page 218
12.1.3.1 License 219
12.1.3.2 Config upload/download 219
12.1.3.3 IODD 222
12.1.3.4 Restart device 223
12.1.3.5 Reset configuration to factory defaults 223
12.1.3.6 Firmware update 223
12.1.3.7 System diagnosis 224
12.1.4 The User page 225
12.2 UNITRONIC® ACCESS single-protocol PN0.../EIP0.../EC0.../MTCP0.../CCL0... variants 226
12.2.1 The System page 226

13 IODD 228

13.1 IO-Link Device parameters and ISDU requests 228
13.2 Web GUI functionality 229
13.2.1 Port Details page 229
13.2.2 Parameters page 231
13.2.3 IODD Management page 232

14 Technical data 234

14.1 General 235
14.2 PROFINET protocol 236
14.3 Power supply of the module electronics/sensors 237
14.4 Power supply of the actuators 238

14.4.1 IO-Link Class A devices (U_L) 238
14.5 I/O ports Channel A (Pin 4) 238
14.5.1 Configured as digital input, Ch. A (Pin 4) 239
14.5.2 Configured as digital output, Ch. A (Pin 4) 240
14.5.3 Configured as IO-Link port in COM mode, Ch. A 242
14.6 I/O ports Channel B (Pin 2) 243
14.6.1 Configured as a digital input, Ch. B (Pin 2) 243
14.6.2 Configured as a digital output, Ch. B (Pin 2) 244
14.7 LEDs 246
14.8 Data transfer times 249

15 Accessories 251

1 About this manual

1.1 General information

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

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

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

U.I. Lapp GmbH

Schulze-Delitzsch-Straße 25
D-70565 Stuttgart
Germany

mailto:info@lapp.com
https://www.lapp.com

U.I. Lapp GmbH reserves the right to make technical changes or changes to this document 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.

1.3 Version information

Version	Created	Changes
1.0	10/2024	

Table 1: Overview of manual revisions

2 Safety instructions

2.1 Intended use

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

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

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

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

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

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

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

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

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

Information on which cables and accessories are permitted for the installation can be obtained from U.I. Lapp GmbH or is contained in this manual.

2.2 Qualified personnel

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

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

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

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

Only U.I. Lapp GmbH is permitted to make changes to the hardware or software of the products that go beyond the scope of this manual.



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



Attention: U.I. Lapp GmbH 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 and IO-Link)
BUI	Back-Up Inconsistency (EIP diagnostics)
CC	CC-Link IE Field
C/Q	I/O port pin 4 mode, IO-Link communication/switching signal
Ch. A	Channel A (Pin 4) of I/O port
Ch. B	Channel B (Pin 2) of I/O port
CIP	Common Industrial Protocol (media independent protocol)
CIP Safety™	Common Industrial Protocol for Safety applications, CIP Safety™ is a registered trademark of ODVA
Class A	IO-Link port specification (Class A)
Class B	IO-Link port specification (Class B)
CoAP	Constrained Application Protocol
CSP+	Control & Communication System Profile Plus
DAT	Device Acknowledgement Time
DCP	Discovery and Configuration Protocol
DevCom	Device Comunicating (EIP diagnostics)
DevErr	Device Error (EIP diagnostics)
DI	Digital Input
DIA	Diagnostic LED
DO	Digital Output
DIO	Digital Input/Output
DTO	Device Temperature Overrun (EIP diagnostics)
DTU	Device Temperature Underrun (EIP diagnostics)
DUT	Device under test

EIP	EtherNet/IP™ is a registered trademark of ODVA
ERP	Enterprise Resource Planning system
ETH	ETHERNET
FE	Functional Earth
FME	Force Mode Enabled (EIP diagnostics)
FS	Functional Safety
FSU	Fast Start-Up
GSDML	General Station Description Markup Language
High-B	High-Byte
HTTPS	Hyper Text Transfer Protocol Secure
ICE	IO-Link port COM Error (EIP diagnostics)
ICT	Invalid Cycle Time (EIP diagnostics)
IDE	IO-Link port Device Error (EIP diagnostics)
IDN	IO-Link port Device Notification (EIP diagnostics)
IDW	IO-Link port Device Warning (EIP diagnostics)
IIoT	Industrial Internet of Things
ILE	Input process data Length Error (EIP diagnostics)
IME	Internal Module Error (EIP diagnostics)
I/O	Input / Output
I/O port	X1 .. X8
I/O port pin 2	Channel B of I/O ports
I/O port pin 4 (C/Q)	Channel A of I/O ports
IODD	I/O Device Description
IOL or IO-L	IO-Link
I/Q	I/O port pin 2 mode, Digital Input/switching signal
ISDU	Indexed Service Data Unit
IVE	IO-Link port Validation Error (EIP diagnostics)
I&M	Identification & Maintenance
JSON	JavaScript Object Notation (platform independent data format)
L+	I/O port pin 1, sensor power supply

UNITRONIC® ACCESS 60	UNITRONIC® ACCESS variants with a width of 60mm
Little Endian	Data format with Low-B on first place (EtherNet/IP)
LLDP	Link Layer Discovery Protocol
Low-B	Low-Byte
LSB	Least Significant Bit
LVA	Low Voltage Actuator Supply (EIP diagnostics)
LVS	Low Voltage System/Sensor Supply (EIP diagnostics)
MIB	Management Information Base
MP	Multi-protocol: PROFINET + EtherNet/IP + EtherCAT® + Modbus TCP (+ CC-Link IE Field Basic)
MQTT	Message Queuing Telemetry Transport (open networking protocol)
MSB	Most Significant Bit
M12	Metric thread according to DIN 13-1 with 12 mm diameter
NTP	Network Time Protocol
OFDT	One Fault Delay Time
OLE	Output process data Length Error (EIP diagnostics)
OPC UA	Open Platform Communications Unified Architecture (platform independent, service-oriented architecture)
PFH	Probability of dangerous Failure per Hour [h -1]
PD	Process Data
PDCT	Port and Device Configuration Tool
PLC	Programmable Logic Controller
PN	PROFINET
PWR	Power
Qualifier	Validity on a process value. Valid = "1"
REST	REpresentational State Transfer
RFC	Request for Comments
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)
SFRT	Safety Function Response Time
SIO mode	Standard Input Output mode
SLMP	Seamless Message Protocol
SNMP	Simple Network Management Protocol
SP	Single Protocol (PROFINET, EtherNet/IP, EtherCAT®, Modbus TCP or CC-Link IE Field Basic)
SPE	Startup Parameterization Error (EIP diagnostics)
T-B	Test Channel B
T-A	Test Channel A
U_{AUX}	$U_{Auxiliary}$, supply voltage for the load circuit (Actuator supply on Class B ports of Class A/B IO-Link Master)
UDP	User Datagram Protocol
UDT	User-Defined Data Types
UINT8	Byte in PLC (IB, QB)
UINT16	Unsigned integer with 16 bits or word in PLC (IW, QW)
U_L	U_{Load}, supply voltage for the load circuit (Actuator supply on Class A IO-Link-Master)
UL	Underwriters Laboratories Inc. (certification company)
UTC	Coordinated Universal Time (Temps Universel Coordonné)
WCDT	Worst Case Delay Time

Table 2: Designations and synonyms

4 System description

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

4.1 About UNITRONIC® ACCESS

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

4.2 Device variants

The following variants are available in the UNITRONIC® ACCESS Multi-protocol and the UNITRONIC® ACCESS single-protocol family:

Article number	Product designation	Description	I/O port functionality
381166717	MP08IOLA08DIO	UNITRONIC® ACCESS M12-60 mm, IO-Link Master Multi-protocol (PN, EIP, EC, MB, CC) Security	8 x IO-Link Class A
381166712	PN08IOLA08DIO	UNITRONIC® ACCESS single-protocol M12-60 mm, IO-Link Master PROFINET	8 x IO-Link Class A
381166713	EIP08IOLA08DIO	UNITRONIC® ACCESS single-protocol M12-60 mm, IO-Link Master EtherNet/IP	8 x IO-Link Class A
381166714	EC08IOLA08DIO	UNITRONIC® ACCESS single-protocol M12-60 mm, IO-Link Master EtherCAT®	8 x IO-Link Class A
381166715	MTCP08IOLA08DIO	UNITRONIC® ACCESS single-protocol M12-60 mm, IO-Link Master Modbus TCP	8 x IO-Link Class A
381166716	CCL08IOLA08DIO	UNITRONIC® ACCESS single-protocol M12-60 mm, IO-Link Master CC-Link IE Field Basic	8 x IO-Link Class A

Table 3: Overview of UNITRONIC® ACCESS Multi-protocol and UNITRONIC® ACCESS single-protocol variants

4.3 I/O port overview

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

UNITRONIC® ACCESS Class A IO-Link ports

Device variant	Port	Pin 1 U_S	Pin 4 / Ch. A (C/Q)				Pin 2 / Ch. B (I/Q)	
MP08...	**Info:**	–	Class A	Type 1	Supply by U_S [1]	Supply by U_L [2]	Type 1	Supply by U_L [2]
	X8:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)
	X7:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)
	X6:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)
	X5:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)
	X4:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)
	X3:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)
	X2:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)
	X1:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)

Table 4: Port configuration of MP08... variants

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

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

UNITRONIC® ACCESS single-protocol Class A IO-Link ports

Device variant	Port	Pin 1 U_S	Pin 4 / Ch. A (C/Q)			Pin 2 / Ch. B (I/Q)
PN08IOLA.../ EIP08IOLA.../ EC08IOLA.../ MTCP08IOLA.../ CCL08IOLA...	**Info:**	–	Class A	Type 1	Supply by U_S [1)]	Type 1
	X8:	Out (2 A)	IOL	DI	DO (0.5 A*)	DI
	X7:	Out (2 A)	IOL	DI	DO (0.5 A*)	DI
	X6:	Out (2 A)	IOL	DI	DO (0.5 A*)	DI
	X5:	Out (2 A)	IOL	DI	DO (0.5 A*)	DI
	X4:	Out (2 A)	IOL	DI	DO (0.5 A*)	DI
	X3:	Out (2 A)	IOL	DI	DO (0.5 A*)	DI
	X2:	Out (2 A)	IOL	DI	DO (0.5 A*)	DI
	X1:	Out (2 A)	IOL	DI	DO (0.5 A*)	DI

Table 5: Port configuration of PN08IOLA.../EIP08IOLA.../EC08IOLA.../ MTCP08IOLA.../CCL08IOLA... variants

[1)] With DO Switch Mode configured as "Push-Pull" (see description in the configuration chapters).

* For **UL applications**: Max. 0.25 A DO.

5 Overview of product features

5.1 PROFINET product features

Data connection

The connection option provided by UNITRONIC® ACCESS is the widely-used M12 connector with D-coding for the PROFINET IO network.

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

Data transmission rates

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

PROFINET RT IO Device

The UNITRONIC® ACCESS variants support PROFINET RT (real-time). This allows the transmission of time sensitive process data between network components in real-time communication.

PROFINET specification V2.35, Conformance Class C

The UNITRONIC® ACCESS variants comply with the PROFINET specification V2.35 and meet the requirements of Conformance Class C for the integrated switch. This means the device can be used in PROFINET IRT networks.

Integrated switch

The integrated Ethernet switch with Conformance Class C has two PROFINET ports and thus supports the establishment of a line or ring topology for the PROFINET IO network.

Media Redundancy Protocol

The additionally implemented Media Redundancy Protocol (MRP) enables the design of a highly available network infrastructure.

Fast Start-Up (FSU)

Fast Start-Up is an accelerated start-up process that enables UNITRONIC® ACCESS devices to start communicating on a PROFINET network after a very short time. This makes a faster tool change possible, for example. Thanks to the FSU feature, the network is ready to communicate in less than 2200 ms.[1]

Shared Device

With the shared device functionality, two controllers can access the same I/O device via a PROFINET interface. This option is done by copying the configuration of the I/O device into the first and second controller and assigning it to the second controller as shared device. Every sub slot with I/O data can be assigned to **one** of the two PLCs which share the I/O data of the I/O device.

DCP

The devices use the DCP protocol to automatically assign IP addresses.

Net Load Class III

The devices offer advanced robustness against net load according to Net Load Class III.

LLDP

The LLDP protocol is used to detect devices in the vicinity (neighborhood detection).

SNMPv1

The SNMPv1 protocol (according PROFINET standard V2.44) handles network component monitoring and communication between Master and Device (cannot be operated stand-alone).

Alarm and diagnostic messages

The modules support extended PROFINET alarm and diagnostic messages.

---

1 Measured according to the specification: Internal switch is able to forward telegrams.

I&M functions

Identification and maintenance data (I&M) means information stored on the module. The identification data consist of manufacturer details for the module and can only be read. The maintenance data consist of system specific details created during the course of configuration. The modules can be uniquely identified online via the I&M data.

The device supports I&M data related to the PNO 2.832 standard (IO-Link integration for PROFINET, Edition 2):

- I&M0 … I&M3 for the interface module (access slot, sub-slot 0x8000)
- I&M0 for the IO-Link Master proxy
- I&M0 and I&M5 for the IO-Link Device proxies

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

The GSDML offers the option of configuring and parameterizing the I/O ports on the master devices within an engineering tool of a PLC.

5.2 I/O port features

IO-Link specification.

UNITRONIC® ACCESS fulfills IO-Link specification v1.1.3.

8 x IO-Link Master ports

For detailed information see chapter I/O port overview on page 20.

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.

IO-Link port connections

The IO-Link port connection option provided by UNITRONIC® ACCESS devices is the 5-pin M12 connector. Pin 5 is not assigned for IO-Link Class A ports.

Validation & Backup

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

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

IO-Link Device parameterization

IO-Link Device parameterization in a PROFINET network is possible with the Siemens IO_LINK_DEVICE function block (FB50001) for Siemens TIA Portal®.

LED

The status of the ports is visible by the color of the matching LEDs and their flash pattern. For details on the meanings of the LED colors, please see section LEDs on page 246.

5.3 Integrated Web server

Network parameter display

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

Displaying diagnostics

View diagnostics via the integrated Web server.

User management

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

IO-Link Device parameters

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

5.4 Security features

Firmware signature

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

Syslog

The UNITRONIC® ACCESS multi-protocol variants support the traceability of messages centrally managed and logged via Syslog.

User manager

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

Default user settings:

User: admin

Password: private

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

5.5 Other features

Interface protection

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

For more details, see section Port assignments on page 34.

Failsafe

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

Industrial Internet of Things

UNITRONIC® ACCESS 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 UNITRONIC® ACCESS 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 UNITRONIC® ACCESS multi-protocol variants

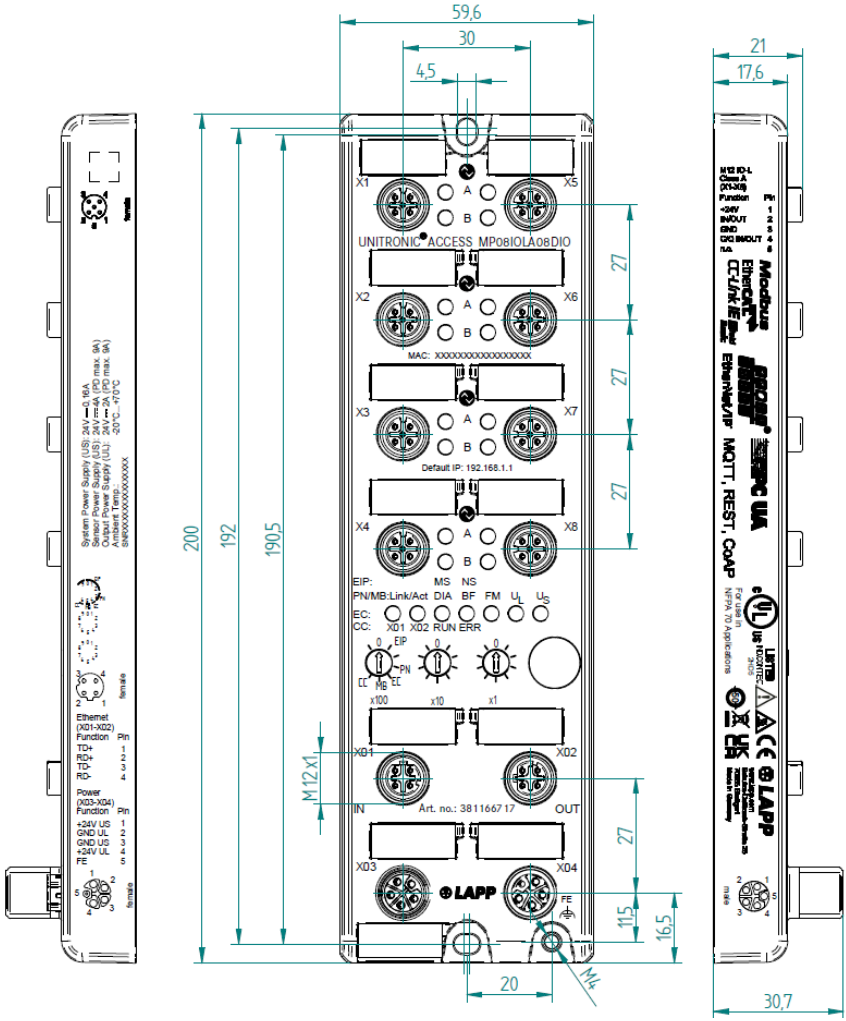


Figure 1: MP08IOLA08DIO

6.2.2 UNITRONIC® ACCESS single-protocol variants with PROFINET

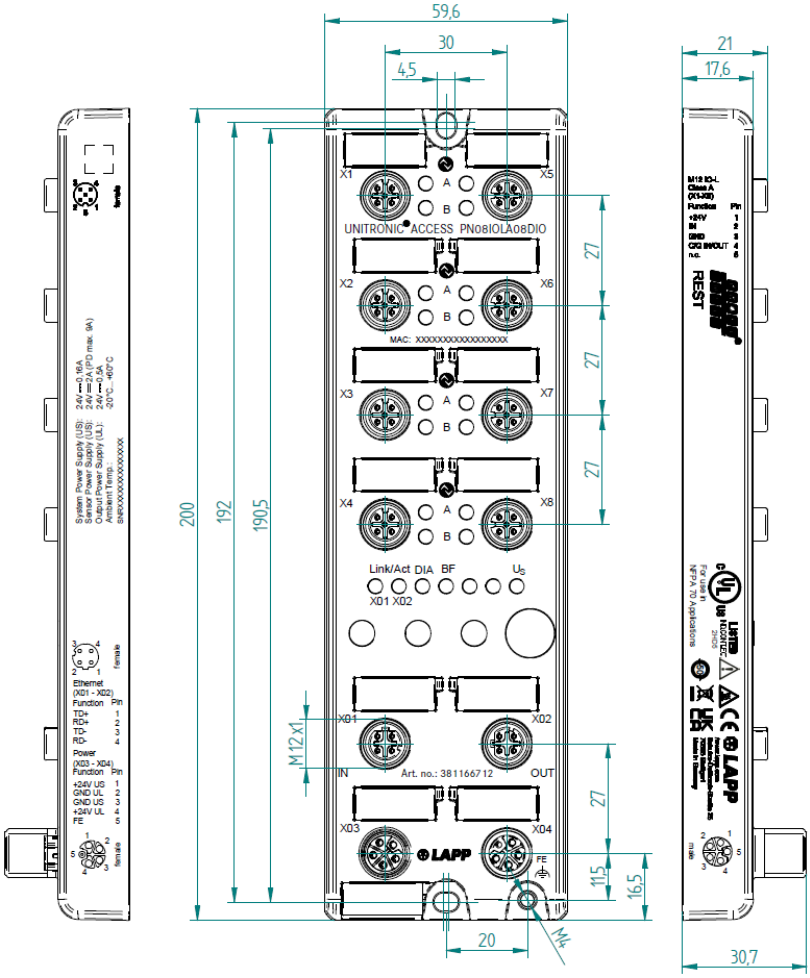


Figure 2: PN08IOLA08DIO

6.2.3 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 device variants.

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

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

6.3 Port assignments

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

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

Color coding: green

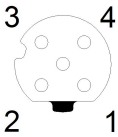


Figure 3: Schematic drawing, ports X01, X02

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 6: 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

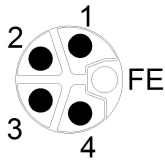


Figure 4: Schematic diagram of the M12 L-coding (connector X03 for Power In)

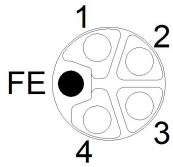


Figure 5: Schematic diagram of the M12 L-coding (socket X04 for Power Out)

6.3.2.1 IO-Link Master with Class A ports

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

Table 7: Power supply with M12-Power Class A

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

6.3.3 I/O ports as M12 sockets

Color coding: black

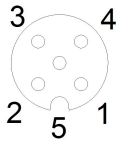


Figure 6: Schematic drawing I/O port as M12 socket IO-Link

[2] U_L and U_S ground connected in device

6.3.3.1 IO-Link ports (Class A)

MP08...	Pin	Signal	Function
IO-Link Class A, ports X1 .. X8	1	+24 V	power supply +24 V
	2	IN/OUT	Ch. B: Digital input or digital output
	3	GND	Ground/reference potential
	4	C/Q	Ch. A: IO-Link data communication, digital input or digital output
	5	n.c.	not connected

PN08IOLA.../ EIP08IOLA.../ EC08IOLA.../ MTCP08IOLA.../ CCL08IOLA...	Pin	Signal	Function
IO-Link Class A, ports X1 .. X8	1	+24 V	power supply +24 V
	2	IN	Ch. B: Digital input
	3	GND	Ground/reference potential
	4	C/Q	Ch. A: IO-Link data communication, digital input or digital output
	5	n.c.	not connected

Table 8: I/O ports as IO-Link Class A

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

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

7 Starting operation

7.1 GSDML file

A GSD file in XML format is required to configure the UNITRONIC® ACCESS variants. All device variants are grouped in a single GSDML file. The file can be downloaded from the product pages on our online catalog: https://lapp.com

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

The GSDML file and the associated bitmap files are grouped together in an archive file named `GSDML-V2.44-U.I. Lapp-UNITRONIC ACCESS-yyyymmdd.xml`.

`yyyymmdd` stands for the date on which the file was issued.

Download this file and unpack it.

In Siemens TIA Portal® you create a new project and open the hardware manager under **Configure a device**. Under the menu command **Options** > **Manage general station description files (GSD)** the GSD file is installed by defining the file path.

The UNITRONIC® ACCESS variants are then 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.

7.3 State on delivery

PROFINET parameters in state on delivery or after a factory reset:

PROFINET name:	Name not assigned
IP address:	0.0.0.0
Subnet mask:	0.0.0.0
Device designations:	MP08IOLA08DIO PN08IOLA08DIO
Vendor ID:	0x18F
Device ID:	0x0608

7.4 Setting the rotary encoding switches

The following UNITRONIC® ACCESS IO-Link Master 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):

- MP08IOLA08DIO



Caution: Risk of device damage due to corrupt device memory

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

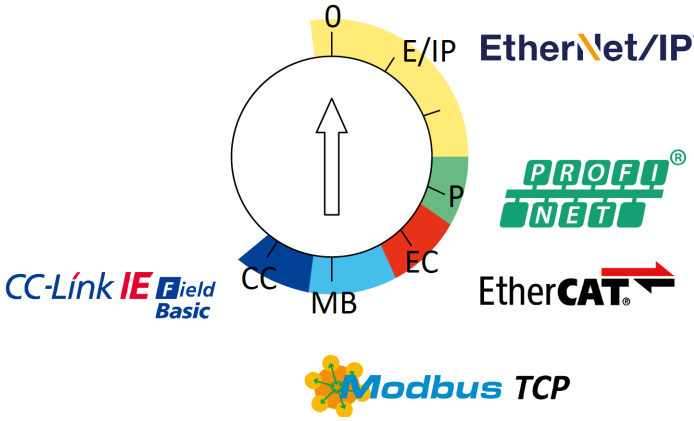
After selecting a protocol followed by a restart of the device, the new protocol is initialized. This can take up to 15 seconds. During this time the device is not usable and the LED indicators are out of function. When the protocol change is complete, the LED indicators return to normal operation and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

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

The UNITRONIC® ACCESS multiprotocol variants allow you to select different protocols for communication within an industrial Ethernet system. In this way the IO-Link Masters 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 IOL-Master in different environments.

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

The 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 9: 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 43.

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

7.4.1 PROFINET

If you decide to use PROFINET, set the first rotary encoding switch to the value of "P".

7.4.2 Factory reset

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

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

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

	x100	x10	x1
Factory Reset	9	7	9

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

For performing a factory reset via software configuration, see chapter OPC UA configuration on page 173 and the configuration section.

7.5 SNMPv1

The PROFINET IO-Link-Master supports SNMP objects required by the PROFINET specification as per protocol standard SNMPv1. These include objects from RFC 1213 MIB-II (System Group and Interfaces Group) and the LLDP MIB.

Passwords:

- Read Community: public
- Write Community: private

8 Configuration and operation with SIEMENS TIA Portal®

Attention: The displayed examples of SIEMENS TIA Portal® have been made with TIA V15.1.

After installing the GSDML files for the UNITRONIC® ACCESS PROFINET variants, they are available in the hardware catalog under **Other field devices** > **PROFINET IO** > **IO** > **U.I. Lapp GmbH** > **UNITRONIC® ACCESS**.

1. First, configure the TIA Portal® project and the control system in the usual way. Assign an IP address and subnet mask for the PROFINET port of the control unit.
2. Then choose the desired device from the Hardware catalog:

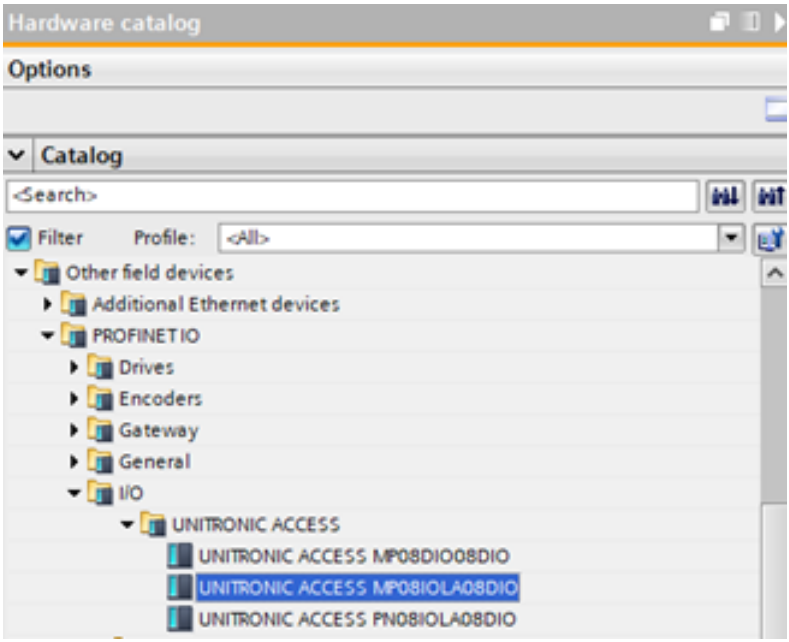


Figure 7: TIA Portal® Hardware catalog

3. Click on the article designations of the modules in the hardware catalog and drag and drop the desired device into the network view:

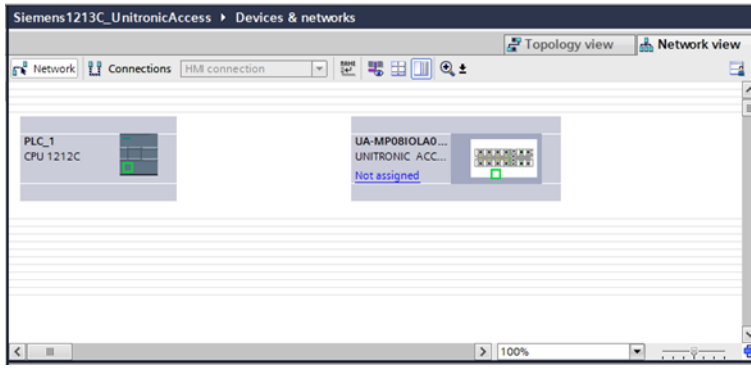


Figure 8: Network view

4. Assign the device to the PROFINET network:

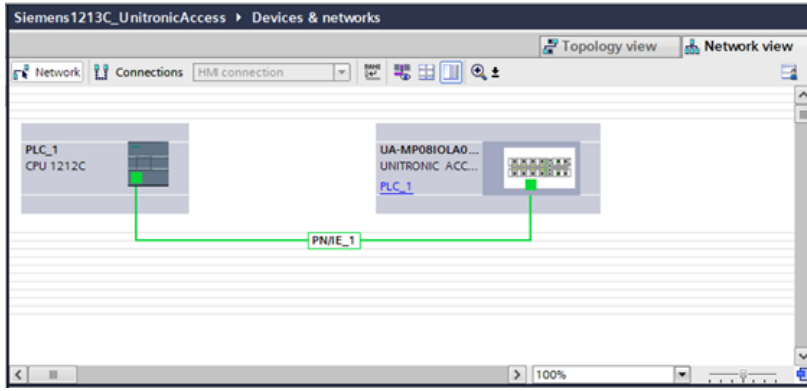


Figure 9: Assign device

5. Switch to the device configuration view and select the device to display configuration options:

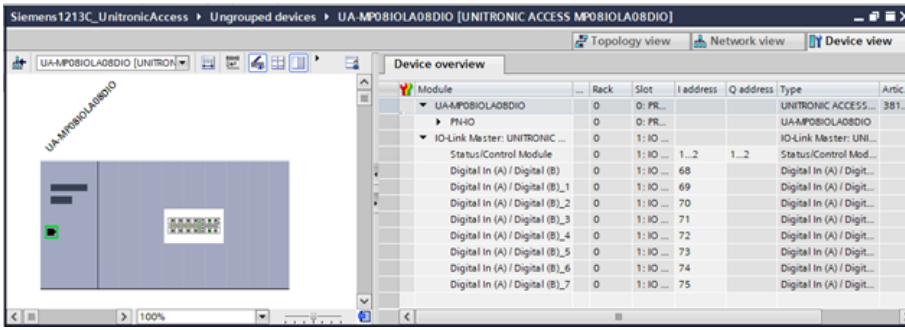


Figure 10: Device configuration

8.1 Assigning a device name and IP address

PROFINET IO devices are addressed on PROFINET via a unique device name. This can be freely assigned by the user but may only be used once on the network.

1. A click on the device icon or on the first line of the **Device overview** opens the settings for **PROFINET interface** > **Ethernet addresses**:

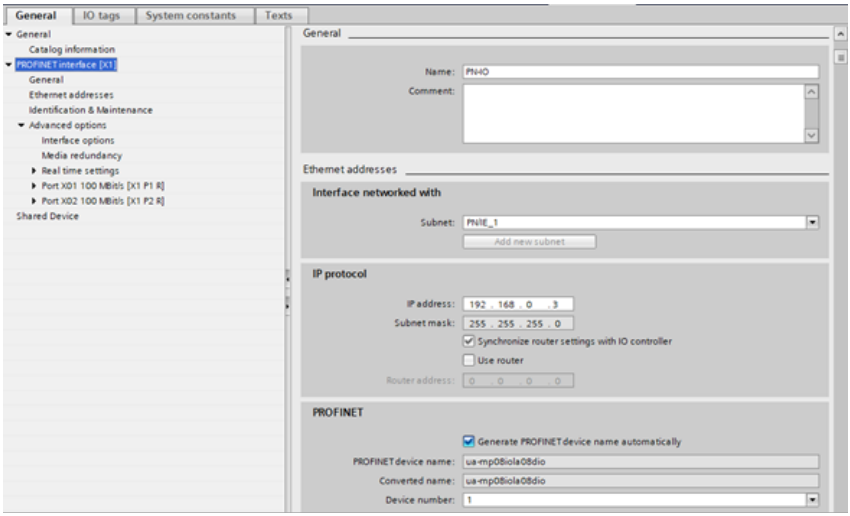


Figure 11: ETHERNET addresses

2. Check that the control unit and the I/O device are on the same Ethernet Subnet.
3. Accept the default settings for IP address and device name or change them if desired.
4. For a correctly working setup, the chosen device name must be programmed online in the I/O device. When the HW is already installed, you can easily change to online mode. The new I/O device should already be accessible via PROFINET:

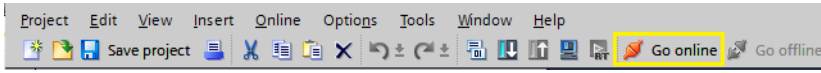


Figure 12: Go online

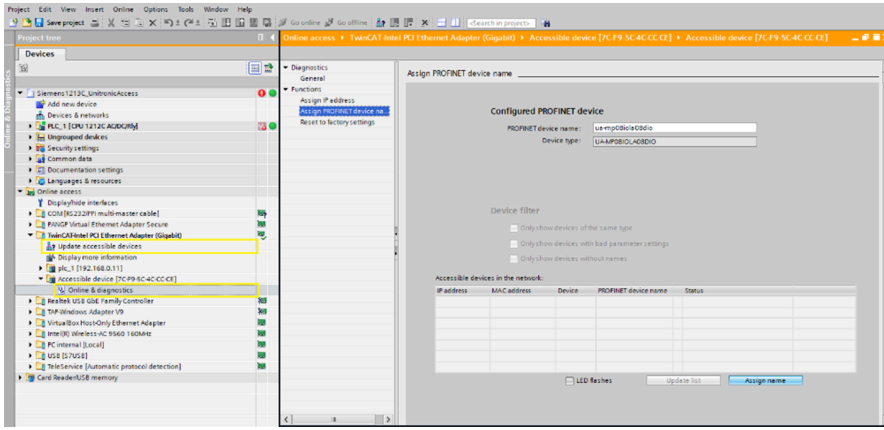


Figure 13: Online mode

5. Enter the same device name as configured in the offline project:

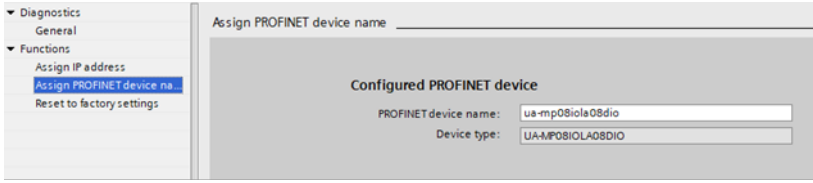
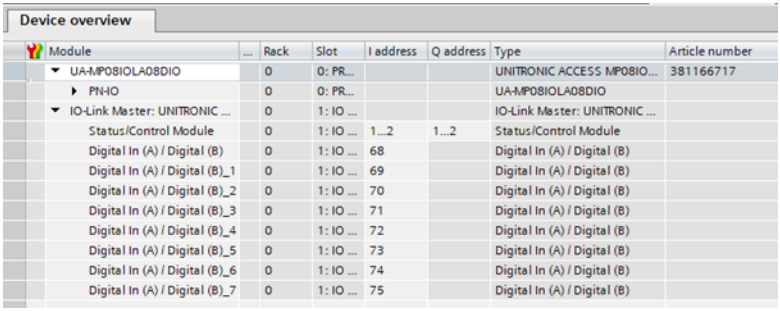


Figure 14: Assign device name

8.2 Configuring the IO-Link channels

By default, all channels are pre-configured as digital input.



Device overview

Module	...	Rack	Slot	I address	Q address	Type	Article number
UA-MP08IOLA08DIO		0	0: PR...			UNITRONIC ACCESS MP08IO...	381166717
PN-IO		0	0: PR...			UA-MP08IOLA08DIO	
IO-Link Master: UNITRONIC ...		0	1: IO ...			IO-Link Master: UNITRONIC ...	
Status/Control Module		0	1: IO ...	1...2	1...2	Status/Control Module	
Digital In (A) / Digital (B)		0	1: IO ...	68		Digital In (A) / Digital (B)	
Digital In (A) / Digital (B)_1		0	1: IO ...	69		Digital In (A) / Digital (B)	
Digital In (A) / Digital (B)_2		0	1: IO ...	70		Digital In (A) / Digital (B)	
Digital In (A) / Digital (B)_3		0	1: IO ...	71		Digital In (A) / Digital (B)	
Digital In (A) / Digital (B)_4		0	1: IO ...	72		Digital In (A) / Digital (B)	
Digital In (A) / Digital (B)_5		0	1: IO ...	73		Digital In (A) / Digital (B)	
Digital In (A) / Digital (B)_6		0	1: IO ...	74		Digital In (A) / Digital (B)	
Digital In (A) / Digital (B)_7		0	1: IO ...	75		Digital In (A) / Digital (B)	

Figure 15: Channels pre-configuration

The configurations of the IO-Link channels (C/Q or Ch. A/Pin 4 of the I/O port) in the sub-slots 2 - 9 (port X1 of the device is equivalent to sub-slot 2, ..., port X8 of the device is equivalent to sub-slot 9) are flexibly definable.

The input and output addresses defined in the device overview can be changed.

8.2.1 Deleting the configuration of a specified IO-Link channel

1. To delete an IO-Link channel, select the desired IO-Link channel(s) in *Device overview*:

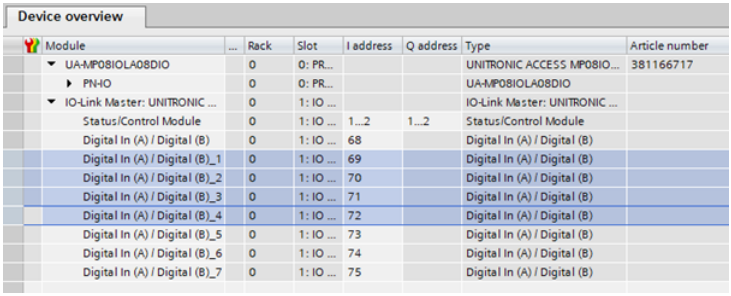


Figure 16: Device overview

2. Right click and then select the *Delete* option in the menu that appears:

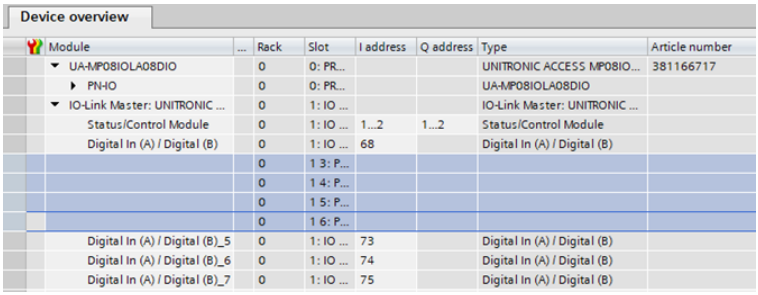


Figure 17: Free IO-Link channel(s)

8.2.2 Creating an IO-Link channel configuration

The *Submodules* folder of the I/O device inside the *Hardware catalog* shows all configurable options that can be selected:

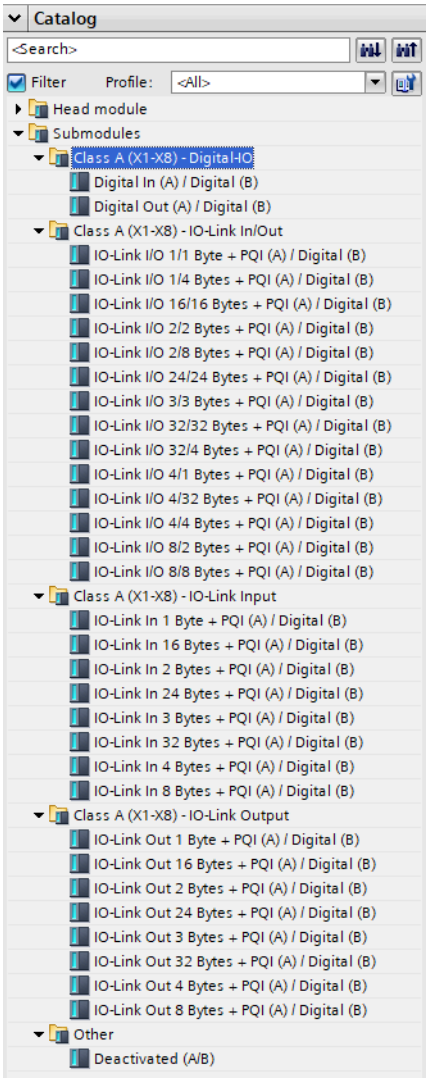
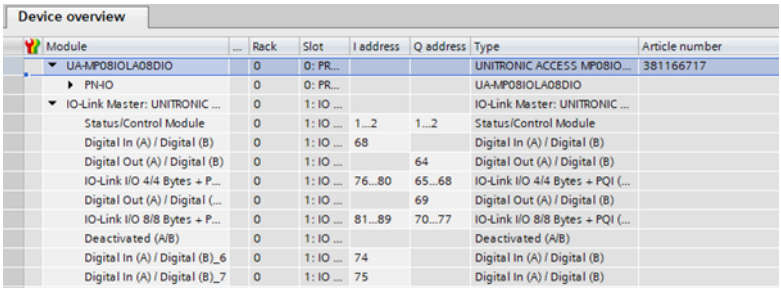


Figure 18: IO-Link channel configuration

Select the desired option, click and hold down the left mouse button to drag the configuration to a free IO-Link sub-slot:



The following options are available for the IO-Link C/Q channel (Ch. A/Pin 4):

Digital In (DI)

In this mode the channel operates as a digital input.

Digital Out (DO)

In this mode, the channel operates as a digital output.

Deactivated

This mode should be selected if the A-channels and the B-channels of the I/O ports (ports X1-X8) are not used. The L+ power supply (pin 1) of the port is disabled in this case.

IO-Link …

In this mode (IO-Link communication mode), the process data from or to the device are exchanged over a communication link. Depending on the port configuration, the IO-Link Master automatically starts communicating with the connected IO-Link Device, taking into account the baud rate. Additionally, this mode offers the option of parameterizing the IO-Link Device. Configuration modules with data lengths of 1-33 bytes for physical inputs and 1-32 bytes for physical outputs are available. If no suitable configuration module is available for the device, the next larger data length must be selected. After the first configuration of the device, this port configuration will be stored non-volatile on the IO-Link Master. This means that with the next power-up the I/O port will be pre-configured with these settings before the controller will send a

new port configuration. The sensor supply (I/O port Pin 1) and the auxiliary supply (I/O port Pin 2) will be powered up directly depending on the last active configuration. A configuration telegram of the PN-Controller is not being required. The I/O data remain invalid until a new configuration is received after the power-up of the IO-Link Master.

8.3 Parameterization of the Status/Control Module

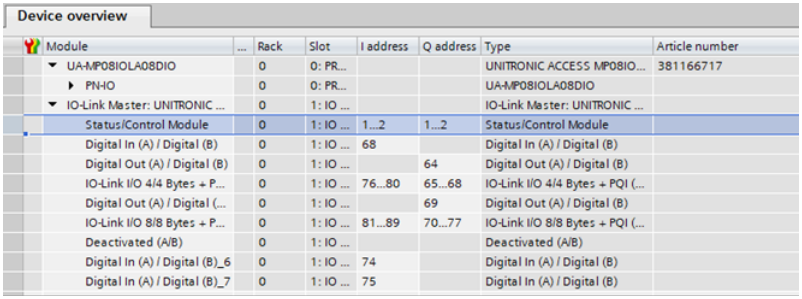


Figure 19: Status/Control Module

Parameters within the Status/Control Module:

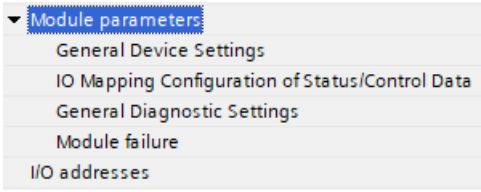


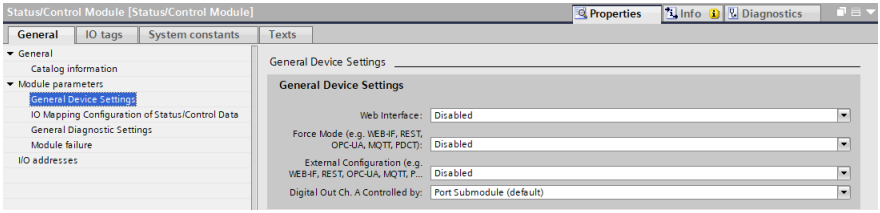
Figure 20: Parameters Status/Control Module

The Status/Control Module in slot 1/sub-slot 1 is pre-configured for each UNITRONIC® ACCESS IOL-Master. It consists of 2 bytes of input and 2 bytes of output data for the digital I/O data. The bit assignments are described in Section Process data assignment on page 130.

The Status/Control Module can also be used to perform some general parameterizations that do not affect any functionality of channels in IO-Link mode.

The following parameterizations are possible after clicking on the subitems under the *Module parameters*.

8.3.1 General Device Settings



Web Interface

The Web interface access can be set to "Enabled" or "Disabled" with this parameter. In case of the "Disabled" setting, the Web pages are not reachable.

Default: Enabled

Force Mode

The input and output I/O data can be forced (= changed) for implementation reasons. This can be done by different interfaces (e.g. Web-Interface, REST, OPC UA, MQTT). The support of interfaces for Forcing depends on the chosen software variant. With this function the possibility of forcing I/O data can be enabled or disabled.

Default: Disabled

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

External Configuration

Configuration and parameter data can be set over different external interfaces outside the GSDML configuration (e.g. Web interface, REST, OPC UA, MQTT). With this option, the “External Configuration” can be enabled or disabled. An external configuration can only be done, if no cyclic PLC connection is active. Every new PLC connection overwrites the external configuration settings.

Default: Disabled

Digital Out Ch. A controlled by...

- Port Sub-module:

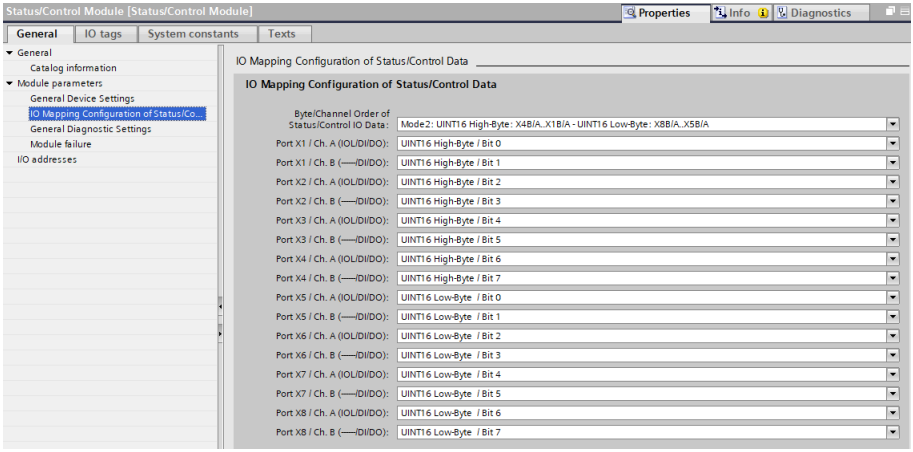
  For controlling the digital A-channels, the output **Byte 1/Bit 0** of the appropriate sub-slot module must be used.

- Status/Control Module:

  In this case, the digital A-channel outputs can be controlled by the Status/ Control Module output bits. The digital output can be controlled only by one data source.

Default: Port Sub-module

8.3.2 I/O mapping configuration of Status/Control data



Byte/Channel order of Status/Control I/O data

With this parameter, 4 (Mode 1 – 4) pre-defined bit mappings for the digital I/O bits can be selected. The I/O data will be mapped to the Status/Control Module's input and output bytes.

Mode 5 can be used for a free, user defined mapping. The parameter settings "Port X1 / Channel A" – "Port X8 / Channel B" must be used for this. These parameters enable all I/O channels to be freely assigned to a Bit in the Status/ Control I/O data. It should be noticed that duplicate assignments are not possible here. If faulty parameterization is detected in the UNITRONIC® ACCESS module, a fault will be registered.

When chosen Mode 1 – Mode 4, the "Port X1 / Channel A" – "Port X8 Channel B" settings will be ignored in the UNITRONIC® ACCESS module.

The chosen mapping will be used in the same way for input and output data direction.

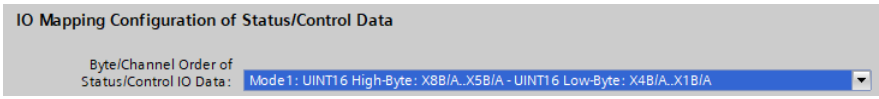
Key

UINT16 High-Byte = 1st / low address byte in a Siemens PLC

UINT16 Low-Byte = 2nd / high address byte in a Siemens PLC

(applicable for a Siemens PLC using Big-Endian format)

Mode 1:



Mode 2:

Default from GSDML-V2.44-U.I. Lapp-UNITRONIC ACCESS-20240719 and higher



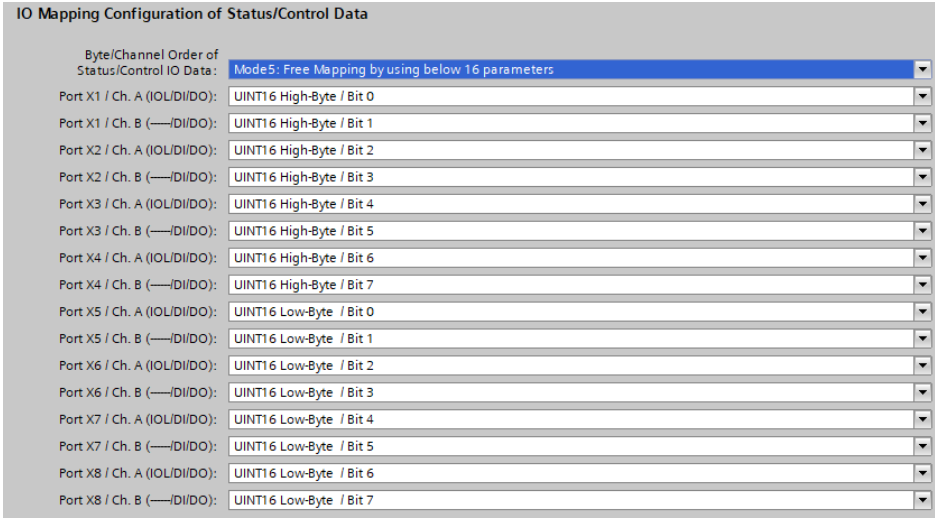
Mode 3:



Mode 4:



Mode 5:



For detailed I/O mapping refer to chapter Process data Status/Control Module, I/O system 1.1 on page 130.

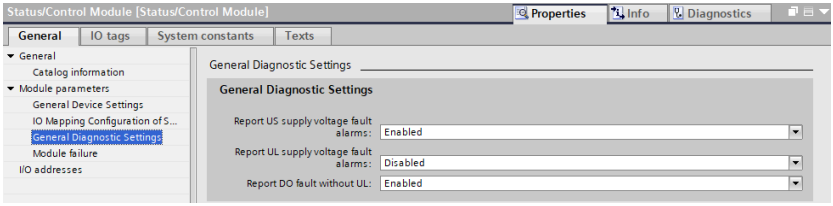
8.3.3 General Diagnostic Settings

Voltage information U_L

▶ The label U_L (Load voltage) is used for devices which exclusively have Class A IO-Link ports.

Warning: To preserve the galvanic isolation concept for Class B IO-Link ports, do not mix up Class A IO-Link port devices with Class A/ B IO-Link port devices in one power line.

8.3.3.1 Diagnostic settings for modules with IO-Link Class A ports



Report U_S supply voltage fault alarms

The U_S supply voltage fault alarm can be set to "Disabled" or "Enabled" with this parameter.

Default: Enabled

Report U_L supply voltage fault alarms

The U_L supply voltage fault alarm can be set to "Disabled", "Enabled" or "Auto Mode" with this parameter.

In "Auto Mode", the U_L diagnosis will be activated with the first rising slope detection after power-up.

Default: Disabled

Attention: "Report U_L supply voltage fault" is disabled in the default setting to avoid diagnostic messages due to switching the supply voltage on or off later on.

Report DO fault without U_L

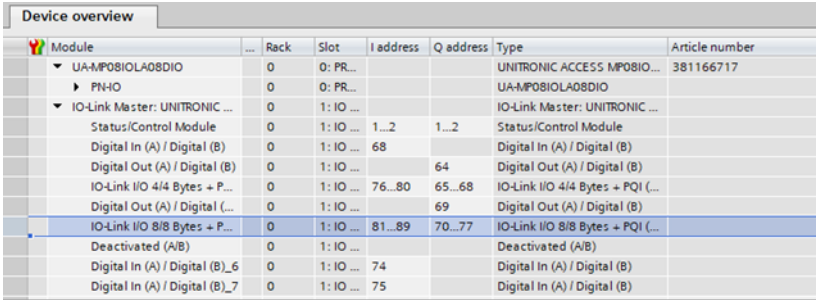
The diagnosis of digital outputs can be configured in dependency of the U_L status.

When the output will be active without active U_L while this parameter is set to "Enabled", a diagnosis message will be generated for the output channel.

Default: Enabled

8.4 Parameterization of the I/O ports X1 .. X8

In the device configuration mode, click on the corresponding IO-Link sub-slot in the Device overview and select the option *module parameters* for setting the following parameters:



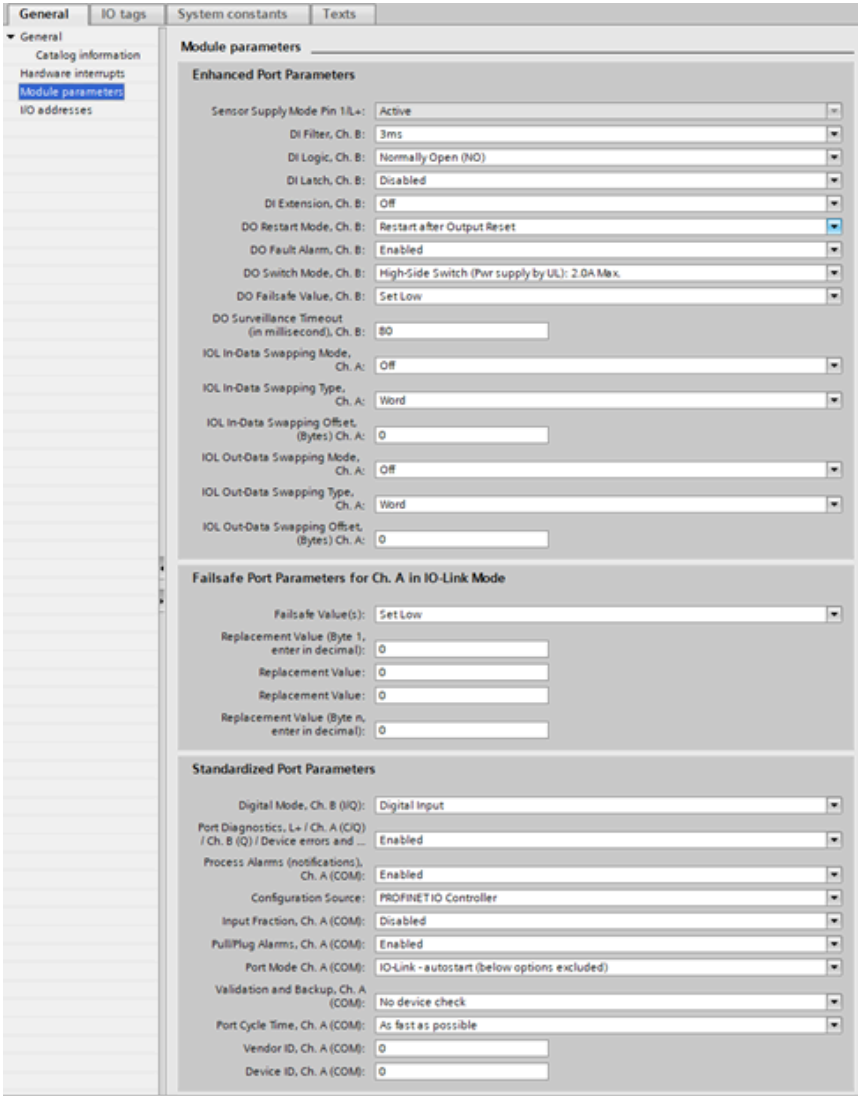


Figure 21: Parameter of the IO-Link channels

8.4.1 Enhanced port parameters

In dependency of the configured sub-module, some of the following described parameters can differ.

(Available only for special channel, otherwise not available.)

Sensor Supply Mode Pin 1 / L+

The sensor supply on pin 1 is always active and cannot be disabled.

DI Filter

With this parameter, the filter time of the digital input can be defined. The following options are available:

Off; 1 ms; 2 ms; 3 ms; 6 ms; 10 ms; 15 ms

Default: 3 ms

DI Latch

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

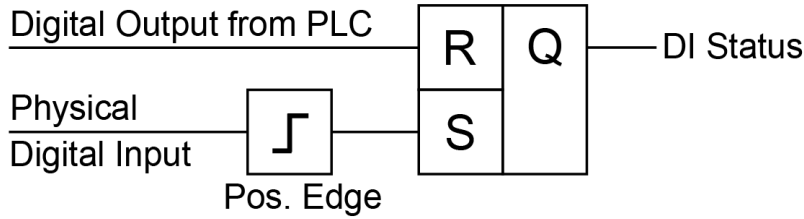


Figure 22: Input latch

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

Table 10: Truth table for input latch

When the input latch is enabled for a specific channel:

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

The behavior in detail:

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

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

For DI Latch reset, the appropriate DO channel in the Status/Control Module (slot 1 / sub-slot 1) must be used (PROFINET).

Default: Disabled

DI Extension

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

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

Example:

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

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

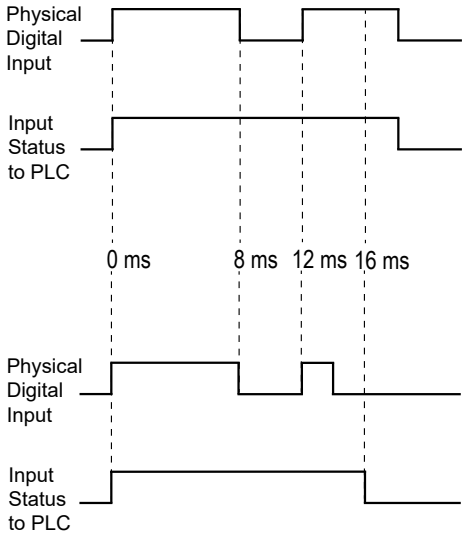


Figure 23: DI Extension

Available values: Off; 8 ms; 16 ms; 64 ms

Default: Off

DI Logic

This parameter can be used to configure the logic of the channels used as digital inputs.

- NO (Normally Open):

  A non-damped sensor has an open switching output (low signal) in this case. The device input detects a low signal and returns a "0" to the control unit.

  The LED of the channel shows the physical input state.

- NC (Normally Closed):

  A non-damped sensor has a closed switching output (high signal) in this case. The device input detects a high signal, inverts the signal, and returns a "0" to the control unit.

  The channel LED displays, independent of the setting, the physical input state.

Default: NO (Normally Open) for all channels

DO Restart Mode

With this parameter, the digital output restart behavior can be set.

- Automatic Restart after Failure:

  In case of detecting an output short circuit or overload, the output will be switched off by the IO-Link Master. However, after a time delay, the output will automatically be turned on again for checking if the overload or short circuit condition is active.

- Restart after Output Reset:

  In case of detecting an output short circuit or overload, the output will be switched off by the IO-Link Master.

  The output will not be set automatically. Before the output can be turned on again, it must be logically reset by the PLC.

Default: Automatic Restart after Failure

DO Switch Mode (MP08... variants only)

With this option, the mode of the digital output switch can be selected.

- Push Pull Switch (0.5 A):

  In this mode, the output will be switched active to high and low. In low state, the output can be a current sink. In this mode, the digital output will be supplied by U_S.

- High-Side Switch (0.5 A; 1.0 A; 1.5 A; 2,0 A; 2.0 A Max.):

  In this mode, the output will be switched active to high, but not to low. Output low means high impedance at digital output. Additionally, the current limitation can be selected for each digital output in High-Side Switch mode. This means that the level for actuator overload diagnostic can be managed by this selection. *2.0 A Max.* means, that current limitation is **not** active and the maximum output current for this output is available. In these modes, the digital output will be supplied by U_L or U_{Aux}, depending on the device variant.

Refer to chapter I/O port overview on page 20 for the voltage supply of the digital outputs.

Default: High-Side Switch (2.0 A Max.)

DO Failsafe Value

The device supports a failsafe function for the channels used as digital outputs. During configuration of the devices, the status of the PROFINET IO device outputs can be defined after an interruption, or loss of communication on the PROFINET IO network.

The following options can be selected:

- Set Low - the output channel is disabled and/or the output bit set to "0".
- Set High - the output channel is enabled and/or the output bit set to "1".
- Hold last – the last output state is kept.

Default: Set Low

DO Surveillance Timeout

For channels configured as digital output, the firmware of the modules allows you, for this special use case, to set a delay time before output status monitoring is enabled.

The delay time is referred to as the "Surveillance-Timeout" and can be configured for each output channel. The delay time begins with a rising edge of the output control bit. After this time has elapsed, the output is monitored, and error states are reported by diagnostics.

The *Surveillance-Timeout* parameter can be set from 0 to 255 ms. When an output channel is in static state, i.e., when the channel is permanently switched on or off, the typical filter value (not changeable) before a diagnostic message is 5 ms.

Default: 80 ms

IO-Link Input/Output Data Swapping

With the following parameters, the IO-Link byte data order can be setup separately for Input and Output data direction.

- Swapping Mode:

  The byte order swapping will be made for the selected count of data types or for the complete length of I/O data with the selected Data Type (Word = 2 Bytes or DWord = 4 Bytes).

  Default: Off

- Swapping Data Type:

  The swapping can be setup to Word (2 Bytes) or DWord (4 Bytes):

  - Word Swapping: Byte 1 - Byte 2 => Byte 2 - Byte 1
  - DWord Swapping: Byte 1 - Byte 4 => Byte 4 - Byte 1

  The Data Type value has no effect when the Swapping Mode is setup to “Off”.

  Default: Word

- Swapping Offset:

  A swapping offset in bytes can be setup in dependency of the configured I/O data length.

  When setup to "2", the swapping will be processed from the 3rd Byte.

  Default: 0

8.4.2 Failsafe port parameters for Ch. A in IO-Link mode

The following values are selectable (for output data only):

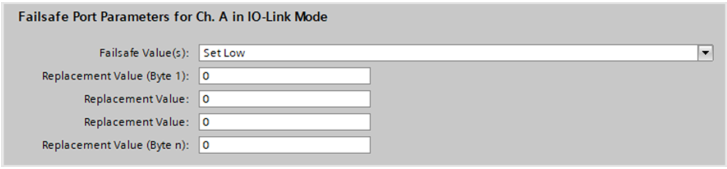


Figure 24: Failsafe Configuration

For a proper function of the IO-Link failsafe values, the IO-Link Device parameters should be set in the same way, if possible. In the case of a lost network connection, the IO-Link Master sends, according to its failsafe configuration, output data to the IO-Link Device. If the IO-Link Device connection is lost, the IO-Link Device uses the failsafe options parameterized inside the device, if supported.

When the device supports a failsafe mechanism, choose the option *IO-Link Master Command*.

Set Low

All bits of the output data with a value of "0" are transmitted to the IO-Link Device. (Default setting)

Set High

All bits of the output data with a value of "1" are transmitted to the IO-Link Device.

Hold Last

The last valid output value received by the control unit is continuously and cyclically transmitted to the IO-Link Device.

For proper *Hold Last* behavior, the appropriate IOL-Device parameters must also be set to *Hold Last*.

Replacement Value

If this option is selected, the value entered in the **Replacement Value** input field described below is continuously and cyclically transmitted to the IO-Link Device.

IO-Link Master Command

The *IO-Link Master Command* option allows the use of IO-Link-specific mechanisms for valid/invalid output process data. Thus, the device itself determines the behavior.

Replacement Value

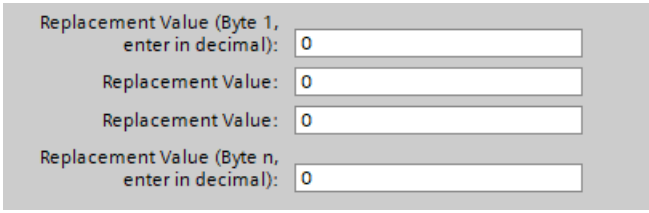


Figure 25: Byte data

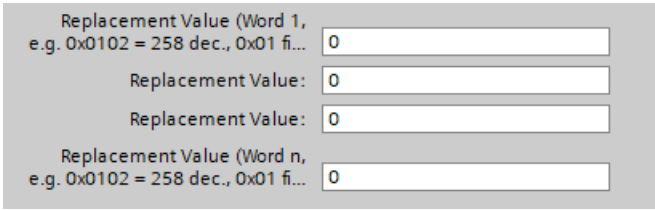


Figure 26: Word data

If the "Fail Safe Value(s)" option is set to "Replacement Value", the substitute value(s) entered in this/these input field(s) is/are used.

The value must be entered as a decimal value. Depending on the configured data length, the values must be entered as bytes (0-255) or word decimal values (0-65535) in the order shown.

- Byte 1 = high order byte (UINT8), in decimal
- Byte n = low order byte (UINT8), in decimal

- Word 1 = high order word (UINT16), in decimal
- Word n = low order word (UINT16), in decimal

"Word" examples: 0x0102 = 258 dec., 0x01 = first byte for IO-Link Device, 0x02 = second byte for IO-Link Device.

8.4.3 Standardized port parameters

Digital Mode, Ch. B

With this parameter the mode of channel B can be defined. The following modes are available:

- Disabled
- Digital Input
- Digital Output
- Power Supply Output (fed from U_L voltage)

The activated power supply output will be signaled by the white port LED.

Default: Digital Input

Port Diagnostic, Ch. A

The IO-Link Master port diagnostics and the IO-Link Device alarms of type "error" or "warning" can be enabled or disabled over this option.

Default: Enabled

Process Alarm, Ch. A (Device Notifications)

The IO-Link Device alarm notifications can be enabled or disabled with this option. Disabled means, that all notification alarm types from every IO-Link Device will be suppressed by the IO-Link Master. They will be not forwarded by PROFINET alarms to the controller.

Default: Enabled

Configuration Source, Ch. A

- PROFINET IO Controller:

  The IO-Link Master port configuration will be assigned by the PROFINET IO Controller.

- Port and Device Configuration Tool (not supported yet):

  The IO-Link Master port configuration will be assigned by an external IO-Link Port and Device Configuration Tool.

Default: PROFINET IO Controller

Input Fraction, Ch. A

If the user configures a sub-slot module with less than the real input data of the device, the IO-Link Master sends as much as possible IO-Link Device input bytes to the PLC inclusive the PQI byte of the sub-slot module. As a consequence, only “0” up to (Device Input Length - 1) octets of the input data of the device can be mapped to the PROFINET process input data of the IO-Link Master. If this option is disabled, a data length mismatch alarm is active in the case of a mismatching input data length. In case of an output data mismatch, a process data mismatch diagnosis will be generated independently of the selected "Input Fraction" setting.

Default: Disabled

Pull/Plug, Ch. A

Enables or Disables Pull/Plug alarms of an IOL-Device (Add/Remove Submodule). A failure or the return of an IO-Link Device can be mapped via PROFINET Plug or Pull alarms. This mapping is independent from any phase such as power-up or start-up.

- Plug Alarms:
  - Ready to Operate (IOL-Device is ready)
  - COM Fault (incorrect device or other problems) – IOL-Device started but not ready due to fault.
- Pull Alarms:
  - COM Fault (No IOL-Device)

In the “Disabled” option, channel diagnostics will be generated in case of losing an IO-Link Device.

Default: Enabled

Port Mode, Ch. A

- Deactivated:

  With the "Deactivated" option, an IO-Link port can be configured for later use. No diagnostics are generated if the IO-Link Device is not connected.

- IO-Link - Autostart:

  With the “Plug&Play” option no explicit port configuration is needed. Basic assignments such as *Validation and Backup* (Inspection Level), *Port Cycle Time*, *Vendor ID* and *Device ID* are not required.

- IO-Link - Manual:

  Explicit port configuration possible for *Validation and Backup* (Inspection Level), *Port Cycle Time*, *Vendor ID* and *Device ID*. These Parameters are GSD based and can be set over a PROFINET engineering system.

Default: IO-Link Autostart

Overview and dependencies of the *Port Mode* configuration type:

Feature	IO-Link - autostart	IO-Link - manual (GSD)
Access on Process Data (PD)	Yes	Yes
Diagnostics of port & device	Yes	Yes
I&M data (IM0) access	Yes	Yes
Device check (consolidated/real)	No	Yes
Backup & Restore	No	Yes
Device parameterization (PDCT)	No	No
Commissioning (online)	No	No

Table 11: Overview, port mode config types

Validation and Backup, Ch. A

For using the *Validation and Backup* functionality of the IOL-Master, the port mode must be set to *IO-Link - manual*.

In dependency of the *Validation and Backup* setting, the entry of the parameters *Vendor ID* and *Device ID* might be obligatory.

- No IOL-Device check (default):

  No check of connected *Vendor ID* or *Device ID* and no *Backup and Restore* support of the IOL-Master backup memory.

- Type compatible (V1.0) IOL-Device:

  Type compatible according to IO-Link specification V1.0 including validation of *Vendor ID* and *Device ID*. IO-Link specification V1.0 does not support IO-Link Master parameter backup memory and restore function.

- Type compatible (V1.1) IOL-Device:

  Type compatible according to IO-Link specification V1.1, check of *Vendor ID* and *Device ID* by IOL-Master.

- Type compatible (V1.1) IOL-Device with Backup & Restore:

  Type compatible according to IO-Link specification V1.1, check of *Vendor ID* and *Device ID* by IOL-Master with *Backup* (IOL-Device to IOL-Master) and *Restore* (IOL-Master to IOL-Device) of the IOL-Device parameters.

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

  Backup (Upload / IOL-Device to IOL-Master):

  During the first connection to an IO-Link Device after enabling this mode, the IOL-Master uploads the IOL-Device parameter into the backup memory. (In this case the backup memory had been empty. Watch below for further information about resetting the IO-Link Master parameter backup memory.)

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

- Parameters written to IOL-Device in *Block Parameter* mode: An IO-Link Device sets the DS_UPLOAD_FLAG self-dependent, if the parameters were written in block parameter mode to the IO-Link Device with the last system command ParamDownloadStore (e.g. by a third party USB IO-Link Master for commissioning).
- Parameters written to IOL-Device in *Single Parameter* mode: If parameters are written to the device in single parameter mode (e.g. one Sub-Index of a parameter Index), the device parameter backup memory on the IOL-Master can be updated by using the ParamDownloadStore system command (Index 0x0002, Sub-Index 0x00, Value 0x05). This command sets the DS_UPLOAD_FLAG (backup request) on the IOL-Device in direction of the IOL-Master, and thus the IOL-Master performs a transfer from IOL-Device to IOL-Master backup memory.

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

For each new connection to an IO-Link Device, the IOL-Master compares the stored parameters with the IOL-Device parameters and downloads the stored backup parameters to the IOL-Device in case of differences.

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

- Type compatible (V1.1) IOL-Device with Restore:

  Type compatible according to IO-Link specification V1.1, check of *Vendor ID* and *Device ID* by IOL-Master with *Restore* (IOL-Master to IOL-Device) of the IOL-Device parameters.

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

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

  During the first connection to an IO-Link Device after enabling this mode, the IOL-Master uploads the IOL-Device parameters once into the backup memory.

  With each further connection to an IO-Link Device, the IOL-Master compares the stored parameters with the IOL-Device parameters and downloads the stored backup parameters to the IOL-Device in case of differences.

  In the *Restore* mode no change of the IOL-Device parameters will be stored in the IOL-Master backup memory. When the IOL-Device sets the DS_UPLOAD_FLAG in this mode, the IOL-Device parameters will be restored from the IOL-Master.

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

▶ Reset conditions of IO-Link Master parameter backup memory:

The IO-Link Master backup memory will be deleted by the following events:

- IO-Link Master factory reset
- A port configuration change, e.g. from “Digital-Input” to “IO-Link Mode”
- A change in the *Validation and Backup* settings, e.g. from “No IOL-Device Check” to “Type compatible IOL-Device (V1.1) with Backup & Restore”

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

Default: No IOL-Device check

Attention: An IO-Link Device sets the “upload flag” self-dependent, if parameter were written in block mode to the IO-Link Device.

Port Cycle Time, Ch. A

(Port mode *IO-Link - manual* required)

▶ As fast as possible:

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

▶ 1.6, 3.2, 4.8, 8, 20.8, 40, 80, 120 ms:

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

Default: As fast as possible

Vendor ID, Ch. A

(Port mode *IO-Link - manual* required)

The Vendor ID of the connected IOL-Device used can be entered as a decimal value [0 … 65535] and will be in used in dependency of the Validation and Backup settings for the validation of the type compatibility.

Default: 0

Device ID, Ch. A

(Port mode *IO-Link - manual* required)

The *Device ID* of the connected IOL-Device can be entered as a decimal value [0 … 65535] and will be in used in dependency of the *Validation and Backup* settings for the validation of the type compatibility.

Default: 0

8.5 IO-Link Device Parameterization

8.5.1 SIEMENS IO-Link library

The SIEMENS *"IO_LINK_DEVICE"* function block (FB50001) can perform acyclic writing or reading data of an IOL-Device connected to the IO-Link-Master.

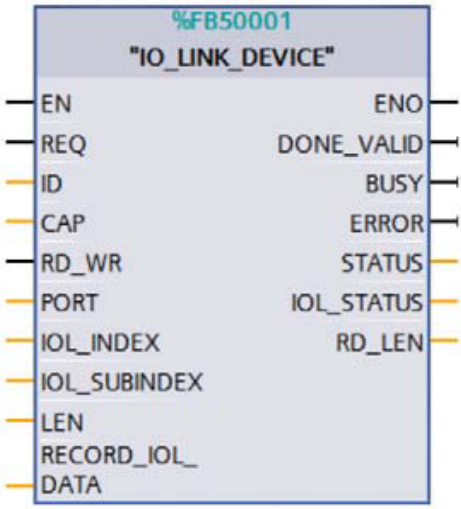


Figure 27: "IO_LINK_DEVICE" FB in STEP 7 V15.1

IOL-Device data are uniquely addressed via the index and the sub-index and can be accessed or written via the hardware identifier of the status/control module (ID) inputs, the client access point (CAP = 0xB400) and the corresponding IO-Link port (PORT: 1 - 8 for IO-Link ports).

The following TIA project shows the used hardware identifier of the sub-module for port X1 (282) with Write/Read examples. Alternatively, the hardware identifier of the Status/Control module can also be used (281 in this example).

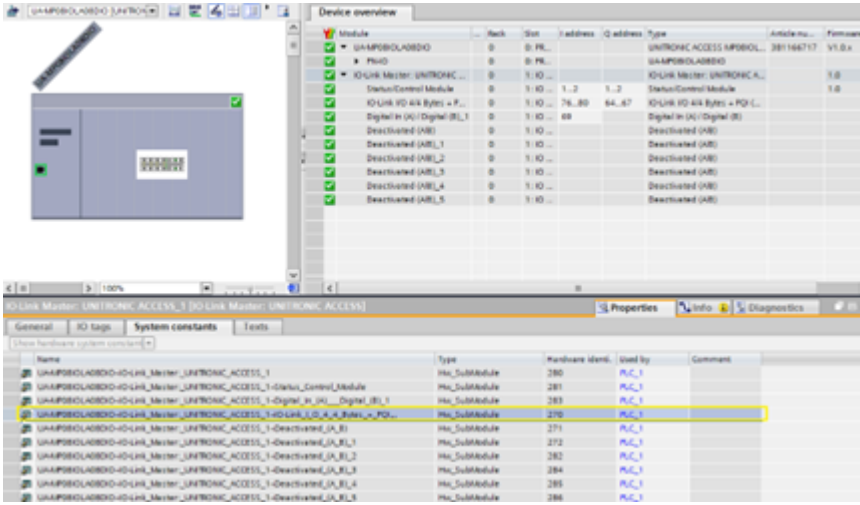


Figure 28: TIA project: Write/Read example with FB50001

8.5.1.1 SIEMENS function block FB50001 – write example

The following is a write example for an IOL-Device on port X1 at the application tag parameter (**IOL_INDEX=24**). The input data are in decimal. The write data are in hexadecimal. The written value is "test" (= 74 / 65 / 73 / 74 in *HEX*).

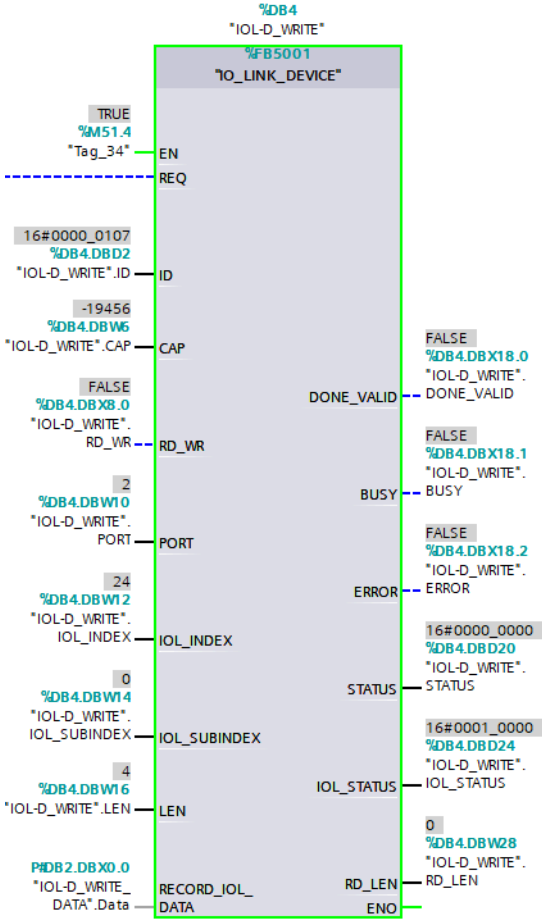


Figure 29: write example for FB50001

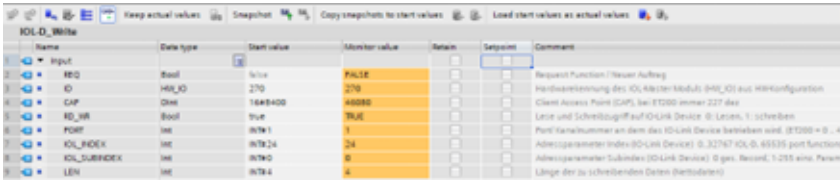


Figure 30: Input control data for write request with FB50001

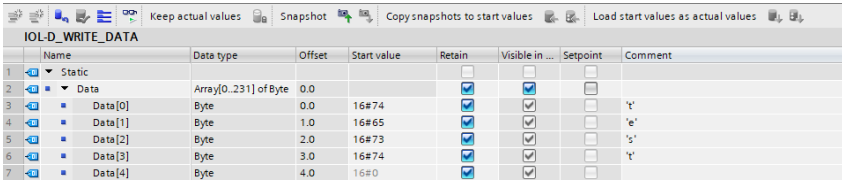


Figure 31: Data to be written with FB50001

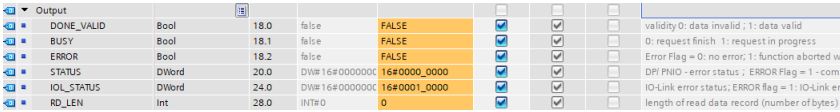


Figure 32: Output status for write request **with** FB50001

8.5.1.2 SIEMENS function block FB50001 – read example

The following is a read example for an IOL-Device on port X1 at the application tag parameter (**IOL_INDEX=24**). The input data are in decimal. The read data are in hexadecimal. The previously written value "test" (= 74 / 65 / 73 / 74 in *HEX*) will be read here.

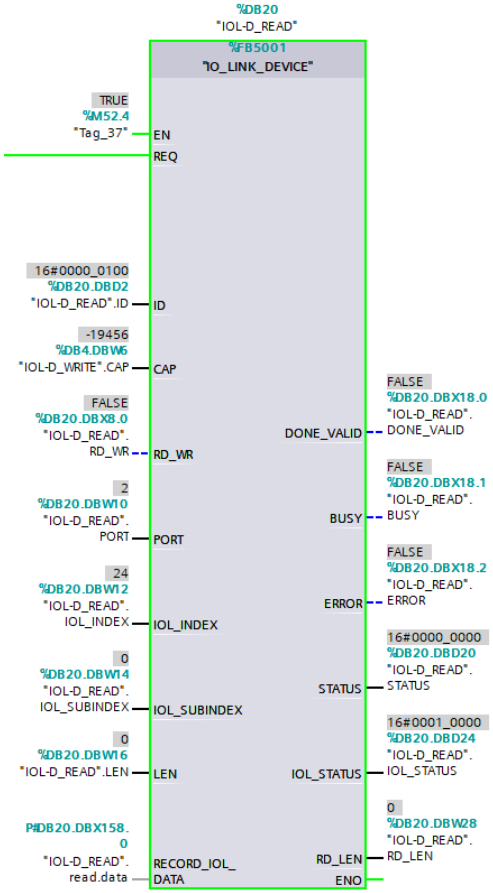


Figure 33: Read example for FB50001

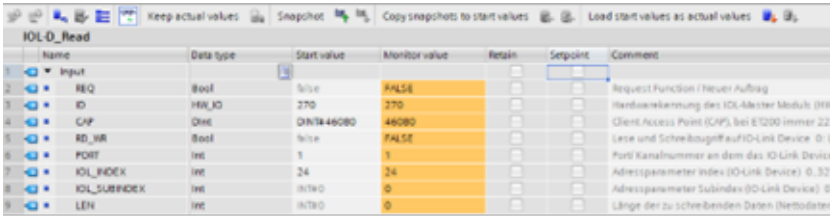


Figure 34: Control data for read request via FB50001

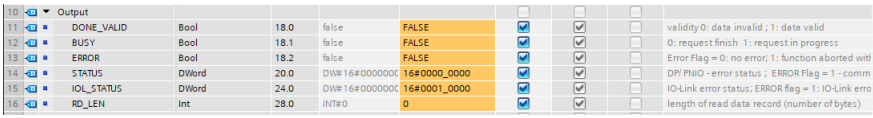


Figure 35: Status data for read request via FB50001

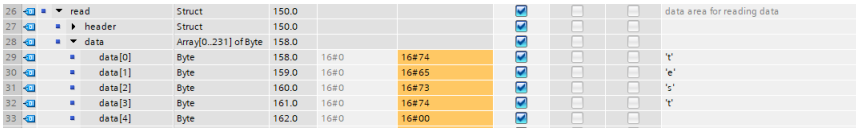


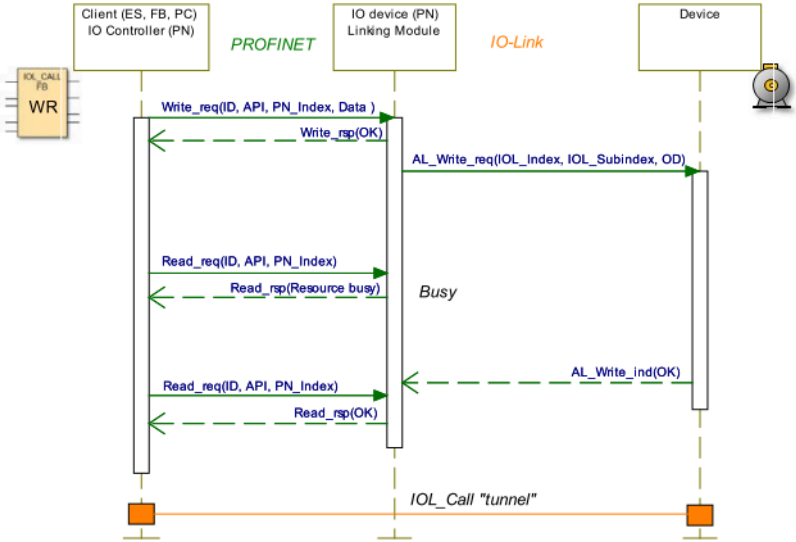
Figure 36: Read data from application tag of the IO-Link Device via FB50001

8.5.2 SIEMENS WRREC and RDREC

The Reading and Writing Parameters from the PLC over the IOL-Master to the connected IOL-Device modules can also be handled by using the SIEMENS Function Blocks *SFB52/RDREC* and *SFB53/WREC*.

8.5.2.1 Write sequence

The following figure shows the sequence of WRREC and RDREC calls for writing data:



The following table shows the sequence with example data compared to the *FB50001*. The *FB50001* uses the WRREC and RDREC blocks also internally:

FB50001 Call	**WRREC**				**RDREC**	**RDREC Response**		
ID (address proxy)	ID (address proxy)				ID (address proxy)			
CAP	PN_Index = 0xB400				PN_Index = 0xB400			
WR	Data Header	Function (fixed)	0x08	Unsigned8		Data Header	Function (fixed)	0x08
Port		Port	1-8	Unsigned8			Port	1-8
		FI_Index (Fixed)	0xFE4A	Unsigned16			FI_Index (Fixed)	0xFE4A
		Control/ Status (→Write)	0x02	Unsigned8			Control/**Status**	0x00
IOL-Index		IOL-Index (0-32767; 65535)	0x...	Unsigned16			IOL-Index (0-32767; 65535)	0x...
IOLSubIndex		IOL-Sub-Index (0-255)	0x00	Unsigned8			IOL-Sub-Index (0-255)	0x00
IOL-Data		WR-Data					Data (opt. Error PDU)	

Table 12: WRREC ID

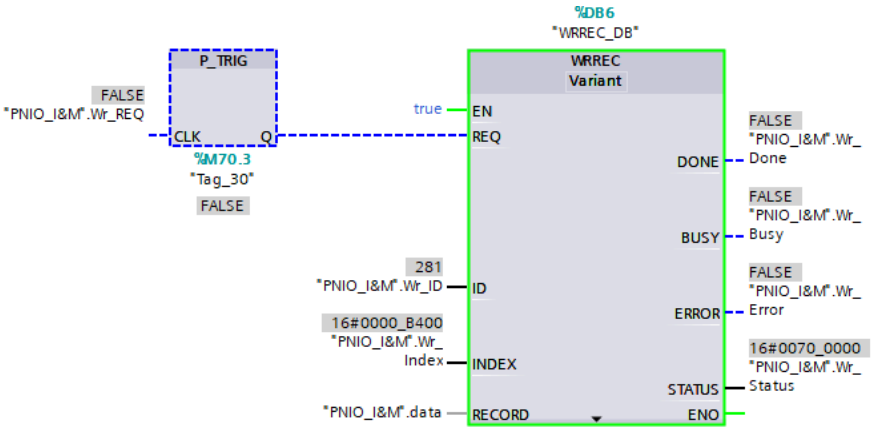
Attention: Unsigned16 values must be entered in Big Endian format for PROFINET.

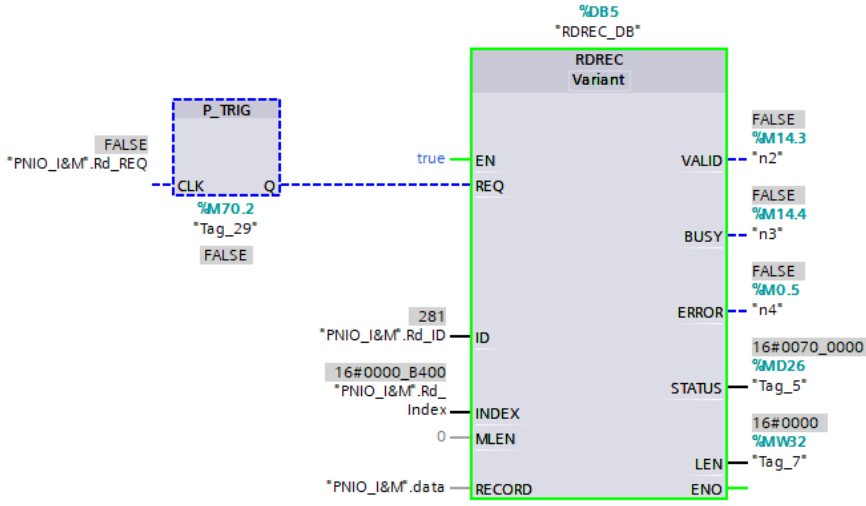
Bit 7	Bit 6	Bit 5	Bit 4	Bit3	Bit2	Bit 1	Bit 0	Definition of Control octets
0	0	0	0	0	0	0	0	Cancel / Release IOL_CALL
0	0	0	0	0	0	0	1	IDLE Sequence
0	0	0	0	0	0	1	0	Write On-request Data or Port function
0	0	0	0	0	0	1	1	Read On-request Data
Other codings								Reserved

Table 13: Control Parameter

Bit 7	Bit 6	Bit 5	Bit 4	Bit3	Bit2	Bit 1	Bit 0	Definition of Status octets
0	0	0	0	0	0	0	0	Done / Transfer terminated
0	0	0	0	0	0	0	1	IDLE Sequence
1	0	0	0	0	0	0	0	IOL_Error PDU
Other codings								Reserved

Table 14: Status Parameter




%DB5
"RDREC_DB"
RDREC
Variant
P_TRIG
FALSE
"PNIO_I&M".Rd_REQ
CLK
Q
%M70.2
"Tag_29"
FALSE
true
EN
REQ
281
"PNIO_I&M".Rd_ID
ID
16#0000_B400
"PNIO_I&M".Rd_
Index
INDEX
0
MLEN
"PNIO_I&M".data
RECORD
VALID
FALSE
%M14.3
"n2"
BUSY
FALSE
%M14.4
"n3"
ERROR
FALSE
%M0.5
"n4"
STATUS
16#0070_0000
%MD26
"Tag_5"
LEN
16#0000
%MW32
"Tag_7"
ENO

Wr_REQ	Bool	false	FALSE
Wr_Index	DWord	16#0	16#0000_B400
Wr_ID	HW_IO	0	281
Wr_Done	Bool	false	FALSE
Wr_Busy	Bool	false	FALSE
Wr_Error	Bool	false	FALSE
Wr_Status	DWord	16#0	16#0000_0000
Wr_Len	UInt	0	0
data	Array[0..39] of Byte		
data[0]	Byte	16#0	16#08
data[1]	Byte	16#0	16#05
data[2]	Byte	16#0	16#FE
data[3]	Byte	16#0	16#4A
data[4]	Byte	16#0	16#02
data[5]	Byte	16#0	16#00
data[6]	Byte	16#0	16#18
data[7]	Byte	16#0	16#00
data[8]	Byte	16#0	16#54
data[9]	Byte	16#0	16#45
data[10]	Byte	16#0	16#53
data[11]	Byte	16#0	16#54
data[12]	Byte	16#0	16#00
data[13]	Byte	16#0	16#00
data[14]	Byte	16#0	16#00
data[15]	Byte	16#0	16#00
data[16]	Byte	16#0	16#00
data[17]	Byte	16#0	16#00

Figure 37: Example of data before writing

	Wr_REQ	Bool	false	TRUE
	Wr_Index	DWord	16#0	16#0000_B400
	Wr_ID	HW_IO	0	281
	Wr_Done	Bool	false	FALSE
	Wr_Busy	Bool	false	FALSE
	Wr_Error	Bool	false	FALSE
	Wr_Status	DWord	16#0	16#0000_0000
	Wr_Len	UInt	0	0
	data	Array[0..39] of Byte		
	data[0]	Byte	16#0	16#08
	data[1]	Byte	16#0	16#05
	data[2]	Byte	16#0	16#FE
	data[3]	Byte	16#0	16#4A
	data[4]	Byte	16#0	16#02
	data[5]	Byte	16#0	16#00
	data[6]	Byte	16#0	16#18
	data[7]	Byte	16#0	16#00
	data[8]	Byte	16#0	16#54
	data[9]	Byte	16#0	16#45
	data[10]	Byte	16#0	16#53
	data[11]	Byte	16#0	16#54
	data[12]	Byte	16#0	16#00
	data[13]	Byte	16#0	16#00
	data[14]	Byte	16#0	16#00
	data[15]	Byte	16#0	16#00
	data[16]	Byte	16#0	16#00
	data[17]	Byte	16#0	16#00

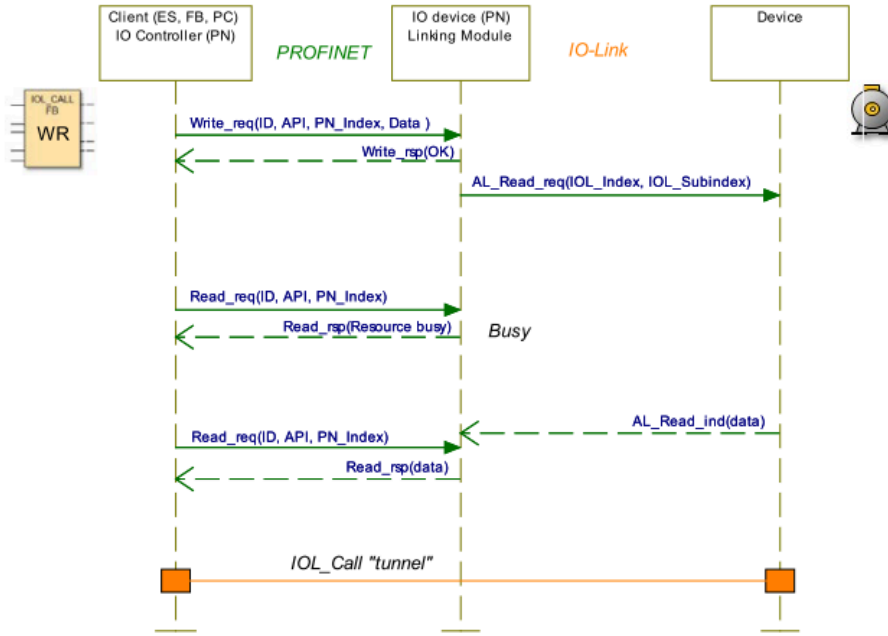
Figure 38: Example of data after writing

	Name	Data type	Start value	Monitor value
	Static			
	Rd_REQ	Bool	false	TRUE
	Rd_Index	DWord	16#0	16#0000_B400
	Rd_ID	HW_IO	0	281
	Rd_Valid	Bool	false	FALSE
	Rd_Busy	Bool	false	FALSE
	Rd_Error	Bool	false	FALSE
	Rd_Status	DWord	16#0	16#0000_0000
	Rd_Len	UInt	0	0
	data	Array[0..39] of Byte		
	data[0]	Byte	16#0	16#08
	data[1]	Byte	16#0	16#05
	data[2]	Byte	16#0	16#FE
	data[3]	Byte	16#0	16#4A
	data[4]	Byte	16#0	16#00
	data[5]	Byte	16#0	16#00
	data[6]	Byte	16#0	16#18
	data[7]	Byte	16#0	16#00
	data[8]	Byte	16#0	16#54
	data[9]	Byte	16#0	16#45
	data[10]	Byte	16#0	16#53
	data[11]	Byte	16#0	16#54
	data[12]	Byte	16#0	16#00
	data[13]	Byte	16#0	16#00
	data[14]	Byte	16#0	16#00
	data[15]	Byte	16#0	16#00
	data[16]	Byte	16#0	16#00
	data[17]	Byte	16#0	16#00

Figure 39: Read data after writing

8.5.2.2 Read sequence

The following figure shows the sequence of WRREC and RDREC calls for reading data:



The following table shows the sequence with example data compared to the *FB50001*. The *FB50001* uses the WRREC and RDREC blocks also internally:

FB50001 Call	**WRREC**				**RDREC**	**RDREC Response**		
ID (address proxy)	ID (address proxy)				ID (address proxy)			
CAP	PN_Index = 0xB400				PN_Index = 0xB400			
WR	Data Header	Function (fixed)	0x08	Unsigned8		Data Header	Function (fixed)	0x08
Port		Port	1-8	Unsigned8			Port	1-8
		FI_Index (Fixed)	0xFE4A	Unsigned16			FI_Index (Fixed)	0xFE4A
		Control/ Status (→Read)	0x03	Unsigned8			Control/**Status**	0x00
IOL-Index		IOL-Index (0-32767; 65535)	0x...	Unsigned16			IOL-Index (0-32767; 65535)	0x...
IOLSubIndex		IOL-Sub-Index (0-255)	0x00	Unsigned8			IOL-Sub-Index (0-255)	0x00
IOL-Data		–					Data (opt. Error PDU)	

Table 15: RDREC ID

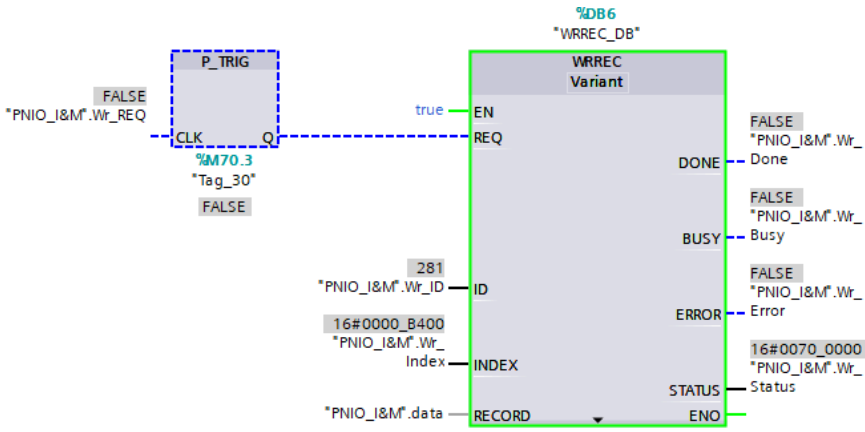
Attention: Unsigned16 values must be entered in Big Endian format for PROFINET.

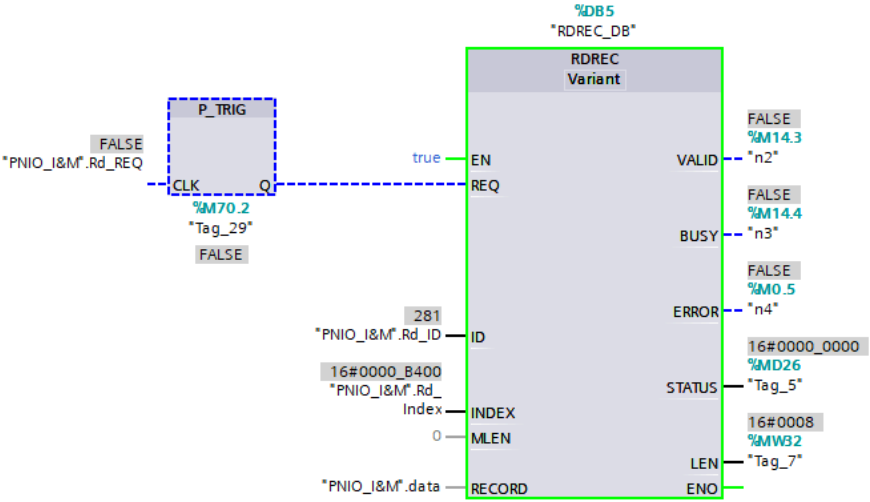
Bit 7	Bit 6	Bit 5	Bit 4	Bit3	Bit2	Bit 1	Bit 0	Definition of Control octets
0	0	0	0	0	0	0	0	Cancel / Release IOL_CALL
0	0	0	0	0	0	0	1	IDLE Sequence
0	0	0	0	0	0	1	0	Write On-request Data or Port function
0	0	0	0	0	0	1	1	Read On-request Data
Other codings								Reserved

Table 16: Control Parameter

Bit 7	Bit 6	Bit 5	Bit 4	Bit3	Bit2	Bit 1	Bit 0	Definition of Status octets
0	0	0	0	0	0	0	0	Done / Transfer terminated
0	0	0	0	0	0	0	1	IDLE Sequence
1	0	0	0	0	0	0	0	IOL_Error PDU
Other codings								Reserved

Table 17: Status Parameter




%DB5
"RDREC_DB"
RDREC
Variant
P_TRIG
FALSE
"PNIO_I&M".Rd_REQ
CLK
Q
%M70.2
"Tag_29"
FALSE
true
EN
REQ
281
"PNIO_I&M".Rd_ID
ID
16#0000_B400
"PNIO_I&M".Rd_
Index
INDEX
0
MLEN
"PNIO_I&M".data
RECORD
VALID
FALSE
%M14.3
"n2"
BUSY
FALSE
%M14.4
"n3"
ERROR
FALSE
%M0.5
"n4"
STATUS
16#0000_0000
%MD26
"Tag_5"
LEN
16#0008
%MW32
"Tag_7"
ENO

	Static			
	Rd_REQ	Bool	false	FALSE
	Rd_Index	DWord	16#0	16#0000_B400
	Rd_ID	HW_IO	0	281
	Rd_Valid	Bool	false	FALSE
	Rd_Busy	Bool	false	FALSE
	Rd_Error	Bool	false	FALSE
	Rd_Status	DWord	16#0	16#0000_0000
	Rd_Len	UInt	0	0
	Wr_REQ	Bool	false	FALSE
	Wr_Index	DWord	16#0	16#0000_B400
	Wr_ID	HW_IO	0	281
	Wr_Done	Bool	false	FALSE
	Wr_Busy	Bool	false	FALSE
	Wr_Error	Bool	false	FALSE
	Wr_Status	DWord	16#0	16#0000_0000
	Wr_Len	UInt	0	0
	data	Array[0..39] of Byte		
	data[0]	Byte	16#0	16#08
	data[1]	Byte	16#0	16#05
	data[2]	Byte	16#0	16#FE
	data[3]	Byte	16#0	16#4A
	data[4]	Byte	16#0	16#03
	data[5]	Byte	16#0	16#00
	data[6]	Byte	16#0	16#18
	data[7]	Byte	16#0	16#00
	data[8]	Byte	16#0	16#00
	data[9]	Byte	16#0	16#00
	data[10]	Byte	16#0	16#00
	data[11]	Byte	16#0	16#00
	data[12]	Byte	16#0	16#00
	data[13]	Byte	16#0	16#00
	data[14]	Byte	16#0	16#00
	data[15]	Byte	16#0	16#00
	data[16]	Byte	16#0	16#00
	data[17]	Byte	16#0	16#00

Figure 40: Example of data before reading

Wr_REQ	Bool	false	TRUE
Wr_Index	DWord	16#0	16#0000_B400
Wr_ID	HW_IO	0	281
Wr_Done	Bool	false	FALSE
Wr_Busy	Bool	false	FALSE
Wr_Error	Bool	false	FALSE
Wr_Status	DWord	16#0	16#0000_0000
Wr_Len	UInt	0	0
data	Array[0..39] of Byte		
data[0]	Byte	16#0	16#08
data[1]	Byte	16#0	16#05
data[2]	Byte	16#0	16#FE
data[3]	Byte	16#0	16#4A
data[4]	Byte	16#0	16#03
data[5]	Byte	16#0	16#00
data[6]	Byte	16#0	16#18
data[7]	Byte	16#0	16#00
data[8]	Byte	16#0	16#00
data[9]	Byte	16#0	16#00
data[10]	Byte	16#0	16#00
data[11]	Byte	16#0	16#00
data[12]	Byte	16#0	16#00
data[13]	Byte	16#0	16#00
data[14]	Byte	16#0	16#00
data[15]	Byte	16#0	16#00
data[16]	Byte	16#0	16#00
data[17]	Byte	16#0	16#00

Figure 41: Example of data after reading

Name	Data type	Start value	Monitor value
Static			
Rd_REQ	Bool	false	TRUE
Rd_Index	DWord	16#0	16#0000_B400
Rd_ID	HW_IO	0	281
Rd_Valid	Bool	false	FALSE
Rd_Busy	Bool	false	FALSE
Rd_Error	Bool	false	FALSE
Rd_Status	DWord	16#0	16#0000_0000
Rd_Len	UInt	0	0
data	Array[0..39] of Byte		
data[0]	Byte	16#0	16#08
data[1]	Byte	16#0	16#05
data[2]	Byte	16#0	16#FE
data[3]	Byte	16#0	16#4A
data[4]	Byte	16#0	16#00
data[5]	Byte	16#0	16#00
data[6]	Byte	16#0	16#18
data[7]	Byte	16#0	16#00
data[8]	Byte	16#0	16#54
data[9]	Byte	16#0	16#45
data[10]	Byte	16#0	16#53
data[11]	Byte	16#0	16#54
data[12]	Byte	16#0	16#00
data[13]	Byte	16#0	16#00
data[14]	Byte	16#0	16#00
data[15]	Byte	16#0	16#00
data[16]	Byte	16#0	16#00
data[17]	Byte	16#0	16#00

Figure 42: Read data after reading

8.5.2.3 Error PDU for the Read/Write sequence

Offset	Parameter	Content	Data type
0	Port Error	Error Codes detected by the Linking Module or Client	Unsigned16
2	Error Code	IO-Link Error codes according AL_Read/ AL_Write services	Unsigned8
3	Additional Code	IO-Link Error codes according AL_Read/ AL_Write services	Unsigned8

Table 18: Error PDU

Port Error Code	Definition	Coding	Originator
No error	No error detected	0x0000	Server
Reserved	–	0x0001 to 0x06FFF	–
IOL_CALL conflict	Inconsistent Header information	0x7000	Server and/or Client
Incorrect IOL_CALL	Inconsistent Header information (send-/response)	0x7001	Server and/or Client
Port blocked	Port temporary not available	0x7002	Server
Reserved	–	0x7003 to 0x7FFF	–
Timeout	No correct termination of IOL_CALL (Resource Busy detection)	0x8000	Client
Invalid port number	Invalid port Number or port not supported	0x8001	Client and/or Server
Invalid IOL_Index	Invalid Index	0x8002	Client
Invalid IOL_Subindex	Invalid Subindex	0x8003	Client
No Device	No device	0x8004	Client
Reserved	–	0x8005 to 0x8051	–
RDREC Fault	Fault during Read record invocation	0x8052	Client
WRREC Fault	Fault during Write record invocation	0x8053	Client
Unexpected Error	Unspecific Error detected	0x8054	Client
Port Function error	Port function failed	0x8055	Server

Port Error Code	Definition	Coding	Originator
Port Function not available	Port function is not available (in this state)	0x8056	Server
Port Function not supported	Port function (for this port) not supported	0x8057	Server
Manu	Manufacturer specific	0x8058 to 0xFFFF	Server

Table 19: Port Error of Error PDU

8.6 Media Redundancy Protocol (MRP)

Redundant PROFINET communication can be implemented with the UNITRONIC® ACCESS devices via a ring topology without the use of additional switches. An MRP redundancy manager terminates the ring, detects individual failures, and transmits the data packets on the redundant path in case of error.

The following conditions must be met to use MRP:

- All devices must support MRP.
- MRP must be enabled on all devices.
- Connections to the devices are only possible via the ring ports. A mesh topology is not permissible.
- A max. of 50 devices are permissible in the ring.
- All devices share the same redundancy domain.
- One device must be configured as the redundancy manager.
- All other devices must be configured as redundancy clients.
- Prioritized boot (FSU) is permissible.
- The response monitoring time of all devices must be greater than the reconfiguration time (typically 200 ms, min. 90 ms for UNITRONIC® ACCESS devices).
- It is recommended to use automatic network settings on all devices.

The following figures show a possible MRP ring configuration. The PLC is used as the redundancy manager while all other devices are clients. To detect an individual failure, it is advisable to use the diagnostics alerts.

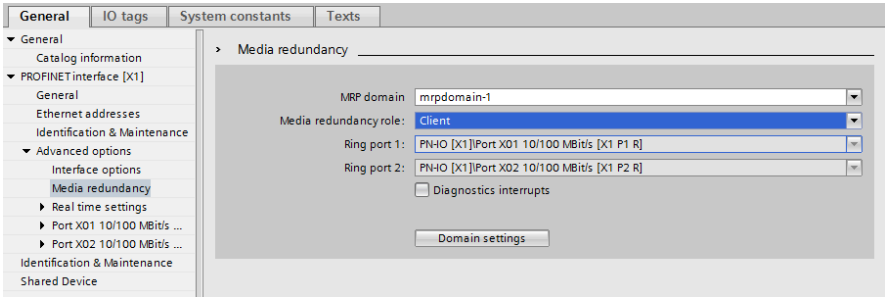


Figure 43: Example of setting up an MRP redundancy client in TIA Portal®

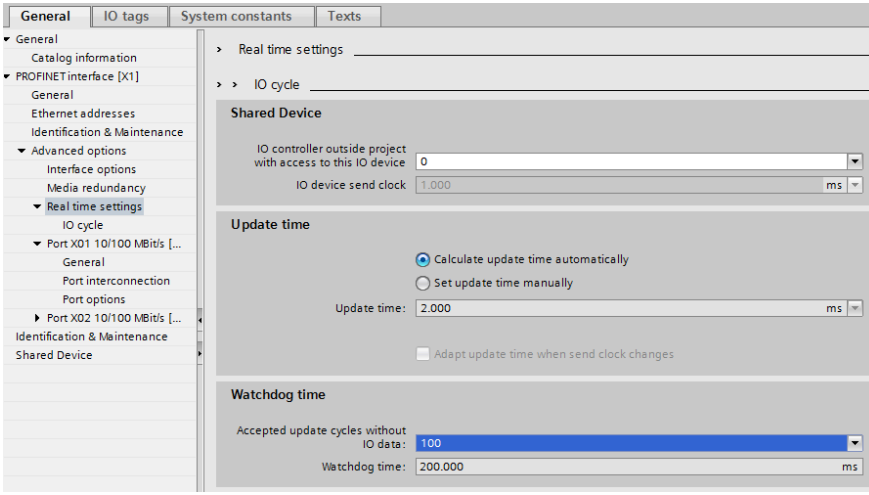


Figure 44: Example of setting up the Watchdog time monitoring in TIA Portal® for using MRP

8.7 Identification & maintenance (I&M)

The PROFINET IO-Link Master has the ability to uniquely identify the devices installed in the system via an electronic nameplate. This device-specific data can be read acyclic by the user at any time. Furthermore, the installation date, location code and further descriptions can be stored in the device during installing the system. The I&M functions provide the following functionality.

8.7.1 Supported I&M features

8.7.1.1 I&M data of the PN-IO Device

For reading (I&M 0 - 3) and writing (I&M 1 - 3) I&M data, the appropriate Hardware identifier for Slot **0: PROFINET Interface X1** must be chosen:

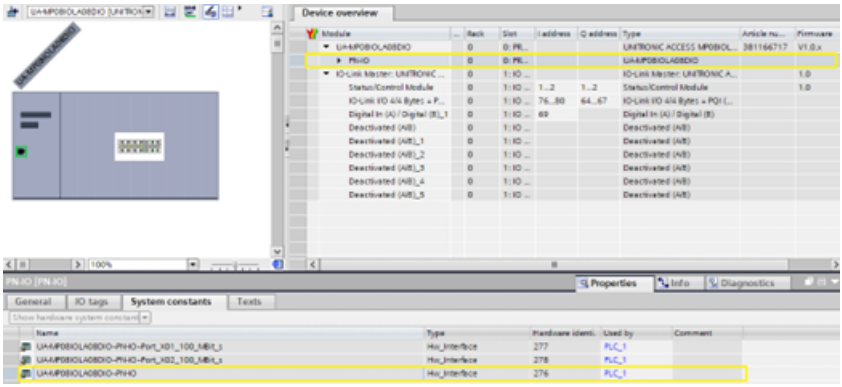


Figure 45: TIA Portal® hardware identifier of PROFINET interface for I&M 0-3 RDREC/WRREC

The device-specific I&M features can be read (0-3) or written (1-3) via slot 0. The specified index is used for mapping the data sets.

Data object	Length [byte]	Access	Default value / Description
MANUFACTURER_ID	2	Read	0x18F (U.I. Lapp GmbH)
ORDER_ID	20	Read	Order number of module in ASCII
SERIAL_NUMBER	16	Read	Defined in production process in ASCII
HARDWARE_REVISION	2	Read	Hardware revision of device
SOFTWARE_REVISION	4	Read	Software revision of device
REVISION_COUNTER	2	Read	Incremented for every statically stored parameter change on IO-Link Master (e.g., device name or IP address)
PROFILE_ID	2	Read	0xF600 (Generic device)
PROFILE_SPECIFIC_TYPE	2	Read	0x0003 (IO modules)
IM_VERSION	2	Read	0x0101 (I&M Version 1.1)
IM_SUPPORTED	2	Read	0x000E (I&M 1 … 3 & 5 is supported)

Table 20: I&M 0 (Slot 0: PROFINET Interface X1, Index 0xAFF0)

Data object	Length [byte]	Access	Default value / Description
TAG_FUNCTION	32	Read/ Write	0x20 ff. (empty)
TAG_LOCATION	22	Read/ Write	0x20 ff. (empty)

Table 21: I&M 1 (Slot 0: PROFINET Interface X1, Index 0xAFF1)

Data object	Length [byte]	Access	Default value / Description
INSTALLATION_DATE	16	Read/ Write	0x20 ff. (empty); Supported data format is a visible string with a fix length of 16 byte; “YYYY-MM-DD hh:mm” or “YYYY-MM-DD” filled with blank spaces

Table 22: I&M 2 (Slot 0: PROFINET Interface X1, Index 0xAFF2)

Data object	Length [byte]	Access	Default value / Description
DESCRIPTOR	54	Read/ Write	0x20 ff. (empty)

Table 23: I&M 3 (Slot 0: PROFINET Interface X1, Index 0xAFF3)

8.7.1.2 I&M data of the IOL-Master proxy (Status/Control Module)

For reading *I&M 0* data, the appropriate hardware identifier for Slot **1: IO System 1.1** must be chosen:

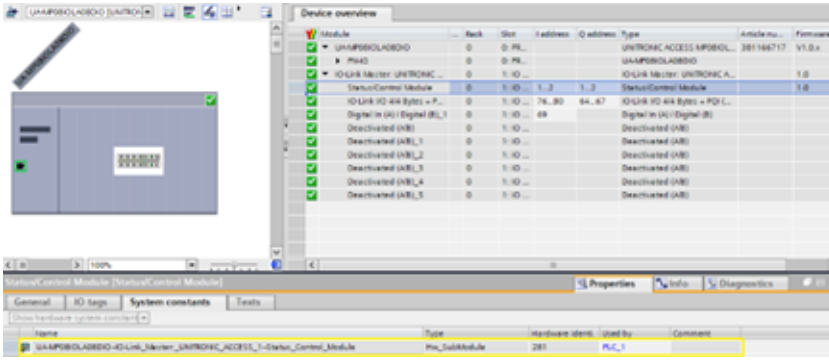


Figure 46: Hardware identifier of the Status/Control Module for RDREC "IO System 1"

Data object	Length [byte]	Access	Default value / Description
MANUFACTURER_ID	2	Read	0x18F (U.I. Lapp GmbH)
ORDER_ID	20	Read	Order number of module in ASCII
SERIAL_NUMBER	16	Read	Defined in production process in ASCII
HARDWARE_REVISION	2	Read	Hardware revision of device
SOFTWARE_REVISION	4	Read	Software revision of device
REVISION_COUNTER	2	Read	Incremented for every statically stored parameter change on IO-Link Master (e.g., device name or IP address)
PROFILE_ID	2	Read	0x4E01 (IOL-Master proxy)
PROFILE_SPECIFIC_TYPE	2	Read	0x0000 (unspecified)
IM_VERSION	2	Read	0x0101 (I&M Version 1.1)
IM_SUPPORTED	2	Read	0x0000

Table 24: I&M 0 (Slot 1: IO System 1.1, Index 0xAFF0)

8.7.1.3 I&M data of the IOL-Device proxy

The IO-Link Device-specific *I&M 0* and *I&M 5* data can be read via slot 1 and the appropriate sub-slot (**1.2/Port X1 .. 1.9/Port X8**). The specified index is used for mapping the data sets. Only data not equal to zero are received if a connection to an IO-Link Device can be established.

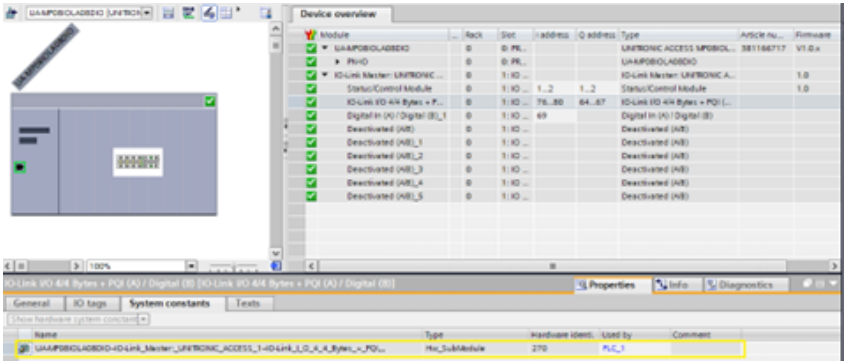


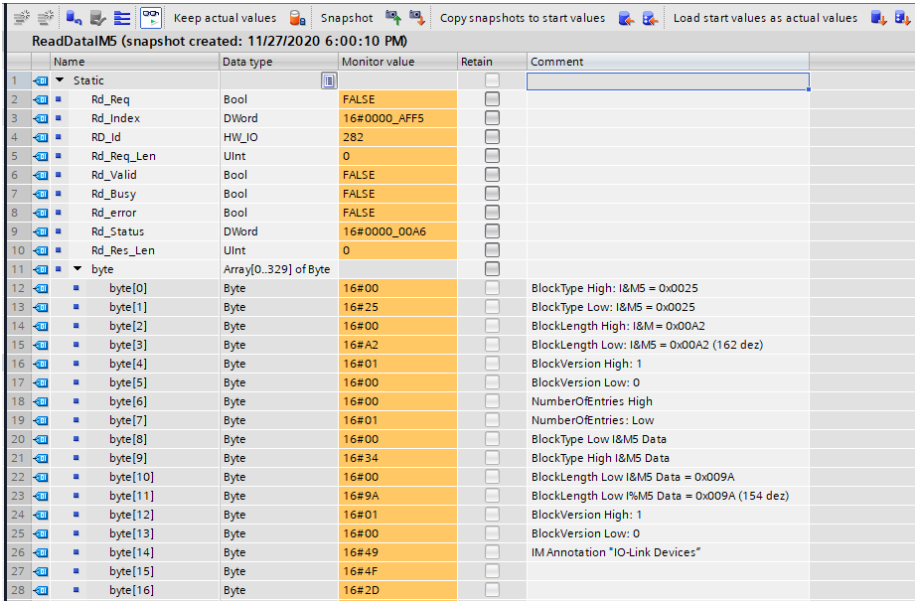
Figure 47: Hardware identifier of the Status/Control module for RDREC "IO System 1.2"

I&M0 data	Octets	Data type	Mapping rules
VendorID	2	Unsigned16	IO-Link Direct parameter page 1: VendorID. Direct mapping, for example "0x136". Exceptions: 1 → 93; 26 → 257; 87 → 467.
OrderID	20	Visible String	"Product Name" or "DeviceID".
IM_Serial_Number	16	Visible String	Insert SerialNumber of Device (IO-Link Index 21). If it is not available set to "Not accessible".
IM_Hardware_Revision	2	Unsigned8	Set to 0x0000 (Default value)
IM_Software_Revision	4	Char,3 x Unsigned8	Set to V0.0.0 (official release but not detectable)
IM_RevisionCounter	2	Unsigned16	Set to "0" (0x0000)
IM_Profile_ID	2	Unsigned16	IO-Link (API = 0x4E01)
IM_Profile_Specific_Type	2	Unsigned16	Set to "0" (0x0000)
IM_Version	2	2 x Unsigned8	Octet 1 (MSB): set to 0x01 Octet 2 (LSB): set to 0x00
IM_Supported	2	Unsigned16 (Bit Array)	Profile specific I&M: 0x0000 (Bit 0 for I&M0 is always 0)

Table 25: I&M 0 (Slot 1: IO System 1.2 - 1.9, Index 0xAFF0)

I&M5 data	Octets	Data type	Mapping rules
IM_Annotation	64	String (UTF8)	"IO-Link Devices"
IM_OrderID	64	Visible String	"Product Name" or "DeviceID".
IM_VendorID	2	Unsigned16	"VendorID"
IM_Serial_Number	16	Visible String	Insert SerialNumber of device (IO-Link Index 21). If it is not available, set to "Not accessible".
IM_Hardware_Revision	2	Unsigned8	Set to 0x0000 (default value)
IM_Software_Revision	4	Char,3 x Unsigned8	Set to V0.0.0 (official release but not detectable)

Table 26: I&M 5 (Slot 1: IO System 1.2 - 1.9, Index 0xAFF5)



	Name	Data type	Monitor value	Retain	Comment
1	Static				
2	Rd_Req	Bool	FALSE		
3	Rd_Index	DWord	16#0000_AFF5		
4	RD_Id	HW_IO	282		
5	Rd_Req_Len	UInt	0		
6	Rd_Valid	Bool	FALSE		
7	Rd_Busy	Bool	FALSE		
8	Rd_error	Bool	FALSE		
9	Rd_Status	DWord	16#0000_00A6		
10	Rd_Res_Len	UInt	0		
11	byte	Array[0..329] of Byte			
12	byte[0]	Byte	16#00		BlockType High: I&M5 = 0x0025
13	byte[1]	Byte	16#25		BlockType Low: I&M5 = 0x0025
14	byte[2]	Byte	16#00		BlockLength High: I&M = 0x00A2
15	byte[3]	Byte	16#A2		BlockLength Low: I&M5 = 0x00A2 (162 dez)
16	byte[4]	Byte	16#01		BlockVersion High: 1
17	byte[5]	Byte	16#00		BlockVersion Low: 0
18	byte[6]	Byte	16#00		NumberOfEntries High
19	byte[7]	Byte	16#01		NumberOfEntries: Low
20	byte[8]	Byte	16#00		BlockType Low I&M5 Data
21	byte[9]	Byte	16#34		BlockType High I&M5 Data
22	byte[10]	Byte	16#00		BlockLength Low I&M5 Data = 0x009A
23	byte[11]	Byte	16#9A		BlockLength Low I%M5 Data = 0x009A (154 dez)
24	byte[12]	Byte	16#01		BlockVersion High: 1
25	byte[13]	Byte	16#00		BlockVersion Low: 0
26	byte[14]	Byte	16#49		IM Annotation "IO-Link Devices"
27	byte[15]	Byte	16#4F		
28	byte[16]	Byte	16#2D		

Figure 48: Read example I&M5 on port X1 with connected IOL-Device

8.7.2 Reading and writing I&M data

In its standard library, SIEMENS offers TIA Portal® system function modules that allow I&M data to be read and written. A data set contains a 6-byte *BlockHeader* and the I&M record.

The data requested on reading, or the data to be written thus only start after the existing header. For writing, the header content must additionally be taken into account. Table 27: Data set with BlockHeader and I&M Record on page 113 shows the structure of a data set.

- For reading I&M 0..3, the RDREC block must be configured with `LEN = 6 Byte Block Header + I&M data length`.
- For reading I&M 5, the RDREC block must be configured with `LEN = 6 Byte Block Header + 8 Byte I&M + I&M data length`.

Data object	Length [byte]	Data type	Coding	Description
BlockType	2	Word	I&M 0: 0x0020 I&M 1: 0x0021 I&M 2: 0x0022 I&M 3: 0x0023 I&M 5: 0x0025	BlockHeader
BlockLength	2	Word	I&M 0: 0x0038 I&M 1: 0x0038 I&M 2: 0x0012 I&M 3: 0x0038 I&M 5: 0x0098	
BlockVersionHigh	1	Byte	0x01	
BlockVersionLow	1	Byte	0x00	
I&M Data	I&M 0: 54 I&M 1: 54 I&M 2: 16 I&M 3: 54 I&M 5: 152	Byte		I&M Record

Table 27: Data set with BlockHeader and I&M Record

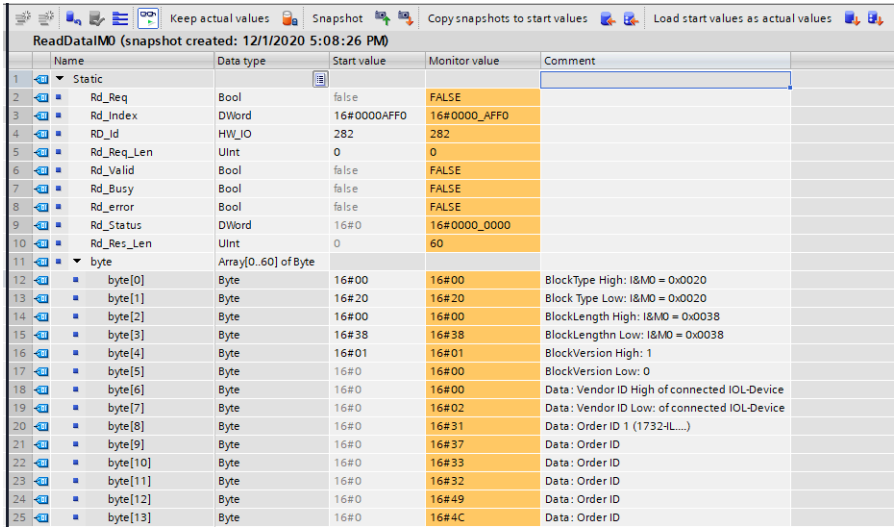
8.7.2.1 I&M Read Record

I&M data can be read via the standard RDREC (SFB52) function block in the **Siemens PLC**. The logical address of the slot/sub-slot (ID) and the I&M index (INDEX) must be used as handover parameters.

ReadDataIM0

	Name	Data type	Start value	Monitor value	Retain	Setpoint	Comment
1	Static				☐	☐	
2	Rd_Req	Bool	false	FALSE	☐	☑	
3	Rd_Index	DWord	16#0000AFF0	16#0000_AFF0	☐	☐	
4	RD_Id	HW_IO	276	276	☐	☐	
5	Rd_Req_Len	UInt	60	60	☐	☐	
6	Rd_Valid	Bool	false	FALSE	☐	☑	
7	Rd_Busy	Bool	false	FALSE	☐	☐	
8	Rd_error	Bool	false	FALSE	☐	☐	
9	Rd_Status	DWord	16#0	16#0000_0000	☐	☐	
10	Rd_Res_Len	UInt	0	60	☐	☐	
11	byte	Array[0..60] of Byte			☐	☐	
12	byte[0]	Byte	16#00	16#00	☐	☐	BlockType High: I&M0 = 0x0020
13	byte[1]	Byte	16#20	16#20	☐	☐	
14	byte[2]	Byte	16#00	16#00	☐	☐	BlockLength High: I&M = 0x0038
15	byte[3]	Byte	16#38	16#38	☐	☐	
16	byte[4]	Byte	16#1	16#01	☐	☐	BlockVersion High: 1
17	byte[5]	Byte	16#0	16#00	☐	☐	BlockVersion Low: 0
18	byte[6]	Byte	16#0	16#01	☐	☐	Data: Vendor ID High
19	byte[7]	Byte	16#0	16#8F	☐	☐	Data: Vendor ID Low
20	byte[8]	Byte	16#0	16#33	☐	☐	Data: Order ID 1 (381166717)
21	byte[9]	Byte	16#0	16#38	☐	☐	Data: Order ID
22	byte[10]	Byte	16#0	16#31	☐	☐	Data: Order ID
23	byte[11]	Byte	16#0	16#31	☐	☐	Data: Order ID
24	byte[12]	Byte	16#0	16#36	☐	☐	Data: Order ID
25	byte[13]	Byte	16#0	16#36	☐	☐	Data: Order ID
26	byte[14]	Byte	16#0	16#37	☐	☐	Data: Order ID
27	byte[15]	Byte	16#0	16#31	☐	☐	Data: Order ID
28	byte[16]	Byte	16#0	16#37	☐	☐	Data: Order ID
29	byte[17]	Byte	16#0	16#20	☐	☐	Data: Order ID
30	byte[18]	Byte	16#0	16#20	☐	☐	Data: Order ID
31	byte[19]	Byte	16#0	16#20	☐	☐	Data: Order ID
32	byte[20]	Byte	16#0	16#20	☐	☐	...

Figure 49: Read example I&M0 of PROFINET IO device

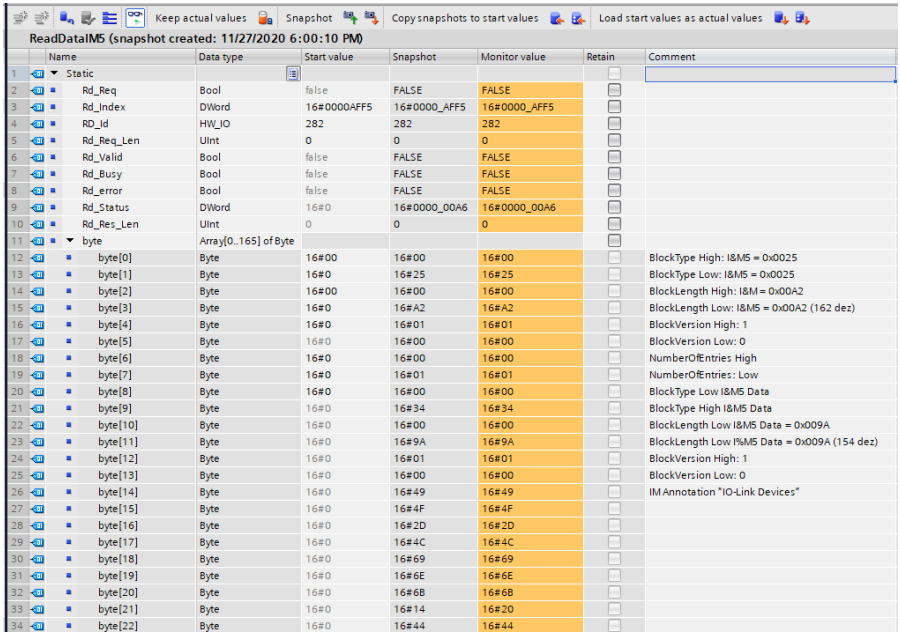


Keep actual values | Snapshot | Copy snapshots to start values | Load start values as actual values

ReadDataIM0 (snapshot created: 12/1/2020 5:08:26 PM)

	Name	Data type	Start value	Monitor value	Comment
1	Static				
2	Rd_Req	Bool	false	FALSE	
3	Rd_Index	DWord	16#0000AFF0	16#0000_AFF0	
4	RD_Id	HW_IO	282	282	
5	Rd_Req_Len	UInt	0	0	
6	Rd_Valid	Bool	false	FALSE	
7	Rd_Busy	Bool	false	FALSE	
8	Rd_error	Bool	false	FALSE	
9	Rd_Status	DWord	16#0	16#0000_0000	
10	Rd_Res_Len	UInt	0	60	
11	byte	Array[0..60] of Byte			
12	byte[0]	Byte	16#00	16#00	BlockType High: I&M0 = 0x0020
13	byte[1]	Byte	16#20	16#20	Block Type Low: I&M0 = 0x0020
14	byte[2]	Byte	16#00	16#00	BlockLength High: I&M0 = 0x0038
15	byte[3]	Byte	16#38	16#38	BlockLengthn Low: I&M0 = 0x0038
16	byte[4]	Byte	16#01	16#01	BlockVersion High: 1
17	byte[5]	Byte	16#0	16#00	BlockVersion Low: 0
18	byte[6]	Byte	16#0	16#00	Data: Vendor ID High of connected IOL-Device
19	byte[7]	Byte	16#0	16#02	Data: Vendor ID Low: of connected IOL-Device
20	byte[8]	Byte	16#0	16#31	Data: Order ID 1 (1732-IL....)
21	byte[9]	Byte	16#0	16#37	Data: Order ID
22	byte[10]	Byte	16#0	16#33	Data: Order ID
23	byte[11]	Byte	16#0	16#32	Data: Order ID
24	byte[12]	Byte	16#0	16#49	Data: Order ID
25	byte[13]	Byte	16#0	16#4C	Data: Order ID

Figure 50: Read example I&M0 on I/O port with connected IOL-Device



Keep actual values | Snapshot | Copy snapshots to start values | Load start values as actual values

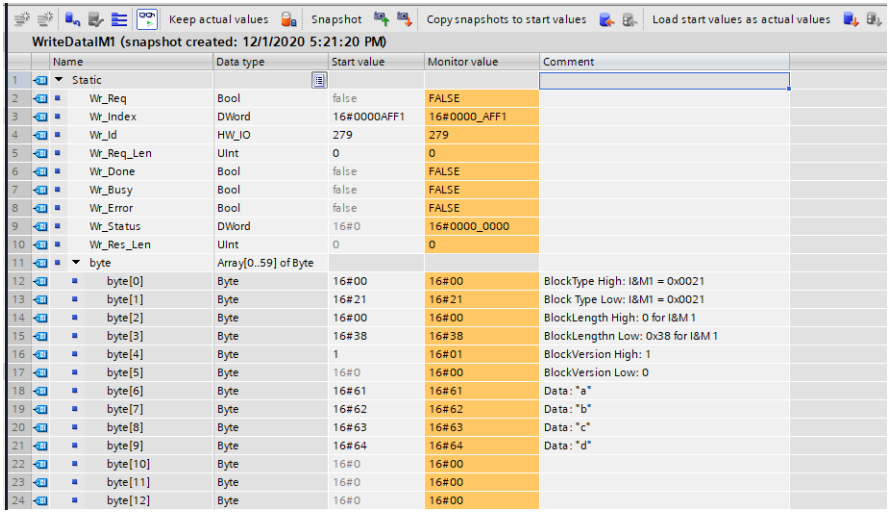
ReadDataIM5 (snapshot created: 11/27/2020 6:00:10 PM)

	Name	Data type	Start value	Snapshot	Monitor value	Retain	Comment
1	Static						
2	Rd_Req	Bool	false	FALSE	FALSE		
3	Rd_Index	DWord	16#0000AFF5	16#0000_AFF5	16#0000_AFF5		
4	RD_Id	HW_IO	282	282	282		
5	Rd_Req_Len	UInt	0	0	0		
6	Rd_Valid	Bool	false	FALSE	FALSE		
7	Rd_Busy	Bool	false	FALSE	FALSE		
8	Rd_error	Bool	false	FALSE	FALSE		
9	Rd_Status	DWord	16#0	16#0000_00A6	16#0000_00A6		
10	Rd_Res_Len	UInt	0	0	0		
11	byte	Array[0..165] of Byte					
12	byte[0]	Byte	16#00	16#00	16#00		BlockType High: I&M5 = 0x0025
13	byte[1]	Byte	16#0	16#25	16#25		BlockType Low: I&M5 = 0x0025
14	byte[2]	Byte	16#00	16#00	16#00		BlockLength High: I&M = 0x00A2
15	byte[3]	Byte	16#0	16#A2	16#A2		BlockLength Low: I&M5 = 0x00A2 (162 dez)
16	byte[4]	Byte	16#0	16#01	16#01		BlockVersion High: 1
17	byte[5]	Byte	16#0	16#00	16#00		BlockVersion Low: 0
18	byte[6]	Byte	16#0	16#00	16#00		NumberOfEntries High
19	byte[7]	Byte	16#0	16#01	16#01		NumberOfEntries: Low
20	byte[8]	Byte	16#0	16#00	16#00		BlockType Low I&M5 Data
21	byte[9]	Byte	16#0	16#34	16#34		BlockType High I&M5 Data
22	byte[10]	Byte	16#0	16#00	16#00		BlockLength Low I&M5 Data = 0x009A
23	byte[11]	Byte	16#0	16#9A	16#9A		BlockLength Low I%M5 Data = 0x009A (154 dez)
24	byte[12]	Byte	16#0	16#01	16#01		BlockVersion High: 1
25	byte[13]	Byte	16#0	16#00	16#00		BlockVersion Low: 0
26	byte[14]	Byte	16#0	16#49	16#49		IM Annotation "IO-Link Devices"
27	byte[15]	Byte	16#0	16#4F	16#4F		
28	byte[16]	Byte	16#0	16#2D	16#2D		
29	byte[17]	Byte	16#0	16#4C	16#4C		
30	byte[18]	Byte	16#0	16#69	16#69		
31	byte[19]	Byte	16#0	16#6E	16#6E		
32	byte[20]	Byte	16#0	16#6B	16#6B		
33	byte[21]	Byte	16#0	16#14	16#20		
34	byte[22]	Byte	16#0	16#44	16#44		

Figure 51: Read example I&M5 on I/O port with connected IOL-Device

8.7.2.2 I&M Write Record

I&M data can be written via the standard WRREC (SFB53) function block in the **Siemens PLC**. The logical address of the slot/sub-slot (ID), the I&M index (INDEX) and the data length (LEN) must be used as handover parameters. The return parameters contain a status or error message.



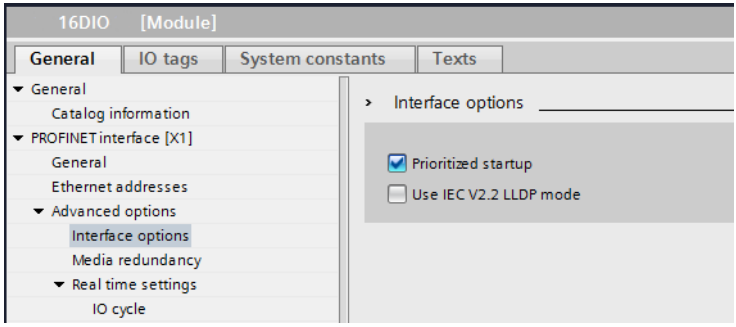
	Name	Data type	Start value	Monitor value	Comment
1	Static				
2	Wr_Req	Bool	false	FALSE	
3	Wr_Index	DWord	16#0000AFF1	16#0000_AFF1	
4	Wr_Id	HW_IO	279	279	
5	Wr_Req_Len	UInt	0	0	
6	Wr_Done	Bool	false	FALSE	
7	Wr_Busy	Bool	false	FALSE	
8	Wr_Error	Bool	false	FALSE	
9	Wr_Status	DWord	16#0	16#0000_0000	
10	Wr_Res_Len	UInt	0	0	
11	byte	Array[0..59] of Byte			
12	byte[0]	Byte	16#00	16#00	BlockType High: I&M1 = 0x0021
13	byte[1]	Byte	16#21	16#21	Block Type Low: I&M1 = 0x0021
14	byte[2]	Byte	16#00	16#00	BlockLength High: 0 for I&M 1
15	byte[3]	Byte	16#38	16#38	BlockLengthn Low: 0x38 for I&M 1
16	byte[4]	Byte	1	16#01	BlockVersion High: 1
17	byte[5]	Byte	16#0	16#00	BlockVersion Low: 0
18	byte[6]	Byte	16#61	16#61	Data: "a"
19	byte[7]	Byte	16#62	16#62	Data: "b"
20	byte[8]	Byte	16#63	16#63	Data: "c"
21	byte[9]	Byte	16#64	16#64	Data: "d"
22	byte[10]	Byte	16#0	16#00	
23	byte[11]	Byte	16#0	16#00	
24	byte[12]	Byte	16#0	16#00	

Figure 52: Example of a completed I&M1 write action of a PROFINET IO device

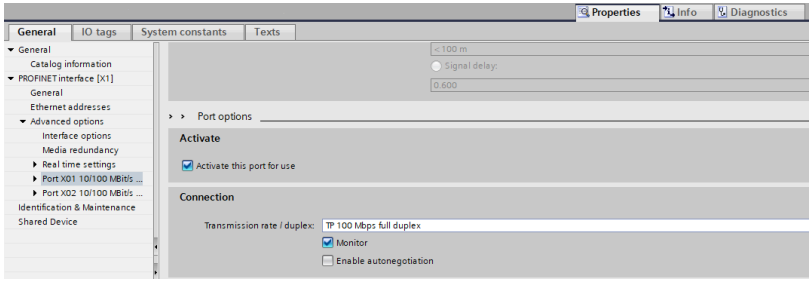
8.8 Fast Start Up (FSU) / Prioritized Startup

Devices with Fast Start-Up (FSU) support an optimized system start-up. This guarantees a faster restart after the power supply is restored.

Fast Start-Up can be activated via **PROFINET interface [X1]** > **Advanced options** > **Interface options** with the option *Prioritized start-up*.



For better FSU performance, the transmission settings of ports X01 and X02 should be set to:



Attention: The settings for the local and the partner port must be identical.

Measured boot times

PROFINET FSU time:[1)]

< 2200 ms

Start time **with** FSU activated:[2)]

MP08... variants: ~2400 ms

PN0.../EIP0.../EC0.../MTCP0.../CCL0... variants: ~12000 ms

Start time **without** FSU activated:[2)]

MP08... variants: ~5400 ms

PN0.../EIP0.../EC0.../MTCP0.../CCL0... variants: ~16000 ms

1) Measured according to specification: Internal switch is able to forward telegrams.

2) PLC reads one digital input and sets one digital output on IO-Link Master after power-up of the DUT (IO-Link Master). PLC is connected directly to DUT port X01 without any additional switch between PLC and DUT.

8.9 Suspend / Resume of IO-Link port operation

8.9.1 Automatic tool changer application use case

Depending on the state of a production process, a tool change inside of a machine is required by undocking a particular tool, for example a gripper, in a magazine and docking another one. This docking and undocking comprise mechanical joint and electrical connections for power supply as well as for communication.

With the following IO-Link calls (e.g. via a Siemens FB50001)

- Suspend port operation
- Resume port operation

the IO-Link port operation can be changed dynamically during the cyclic data exchange.

8.9.2 Concept

The basic concept of the user function "Suspend Port operation" is to suppress the entire PROFINET fault indications to the system/user, since it concerns an intended action. In essence, after the suspension all pending diagnosis messages of the related Port and device are deleted.

The current Port status is always visible to the user via the flag bit "PortActive" in the "Port Qualifier Information – PQI". Three activities characterize the port operations:

- Automatic Port operation
- Suspend Port operation
- Resume Port operation

Automatic Port operation

The following activities will turn a port automatically into state "Port operation resumed", indicated by flag bit "PortActive" = 1:

- Power-on of the IO-Link Device or Master
- Configuration change of the IOL-Master port
- Port configuration mode set to Digital Input or Digital Output

Suspend/Resume Port operation

Figure 53: Suspend/Resume Port operation on page 121 provides an overview of the mechanisms and serves as visualization of the following actions:

- Successfully suspended Port operation leads to flag bit indication "PortActive" = 0 and “DevErr” = 0.
- Undocking of the Tool/Device leads to flag bits "PQ" = 0 and "DevCom" = 0
- Docking of a "new" Tool/Device leads to flag bits "PQ" = 1 and "DevCom" = 1
- Successfully resumed Port operation leads to flag bit indication "PortActive" = 1

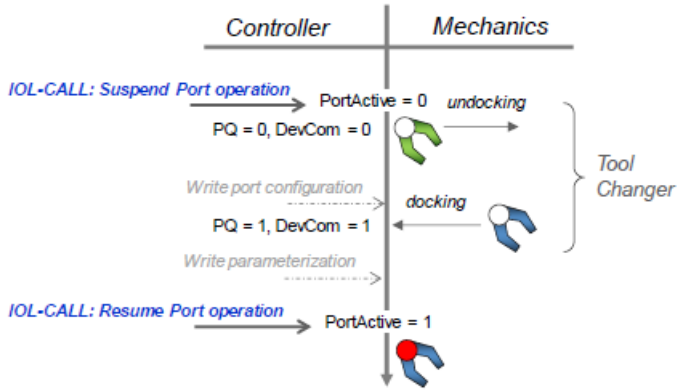


Figure 53: Suspend/Resume Port operation

8.9.3 Use cases

Use case	Inspection level (Backup & Restore)	Description
No. 1: A Device will be replaced by a Device of the same type with identical parameters.	0: no Device check 1: type compatible Device (V1.0) 2: type compatible Device (V1.1) 3: type compatible Device (V1.1) with Backup + Restore 4: type compatible Device (V1.1) with Restore	All Inspection Levels are permitted in use case No. 1. Recommended: type compatible Device V1.1 with Backup + Restore
No. 2: A Device will be replaced by a Device of the same type with different parameters.	0: no device check 1: type compatible Device (V1.0) 2: type compatible Device (V1.1)	Backup and Restore not reasonable in use case No. 2. Recommended: type compatible Device (V1.0 or V1.1)
No. 3: A Device will be replaced by a Device of a different type.	0: no Device check 1: type compatible Device (V1.0) 2: type compatible Device (V1.1)	Backup and Restore not reasonable in use case No. 3. Recommended: type compatible Device (V1.0 or V1.1)

- Port configuration can be adapted while in state "Port operation suspended" (use case no. 3).
- Additionally, the parameterization of the Device can be adapted after active Communication (DevCom =1) through control program (use case no. 2).
- Especially in use case no. 2 and no. 3 it is recommended to deactivate the Backup & Restore function for a better transparency and start-up performance.

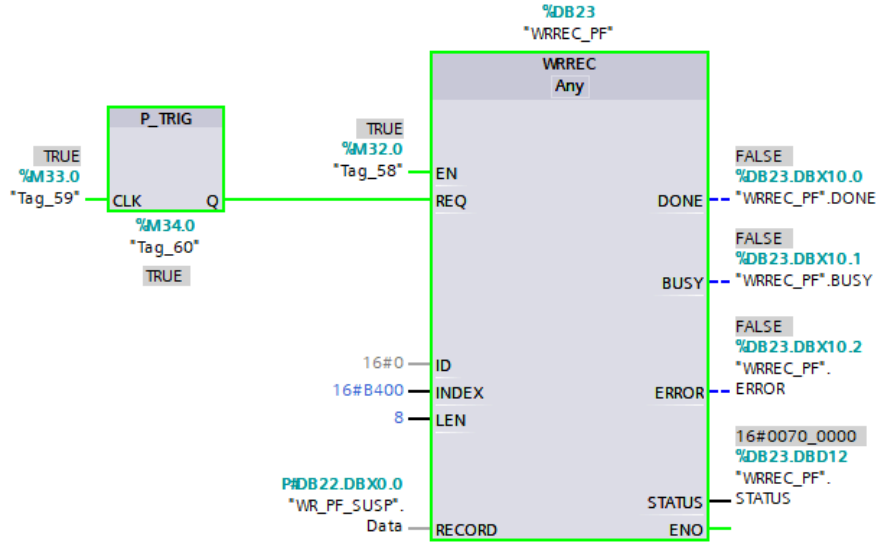
8.9.4 Suspend and Resume cycle

For a complete Suspend and Resume cycle the following read and write requests must be performed step by step.

After writing the commands Suspend or Resume, the successful execution of the command must be verified by the appropriate read request.

8.9.4.1 Write Record Suspend – port command

The following example illustrates how to suspend an IO-Link port operation with the TIA WRREC function block:



ID = 0 for addressing the IO-Link Master proxy

INDEX = 0xB400

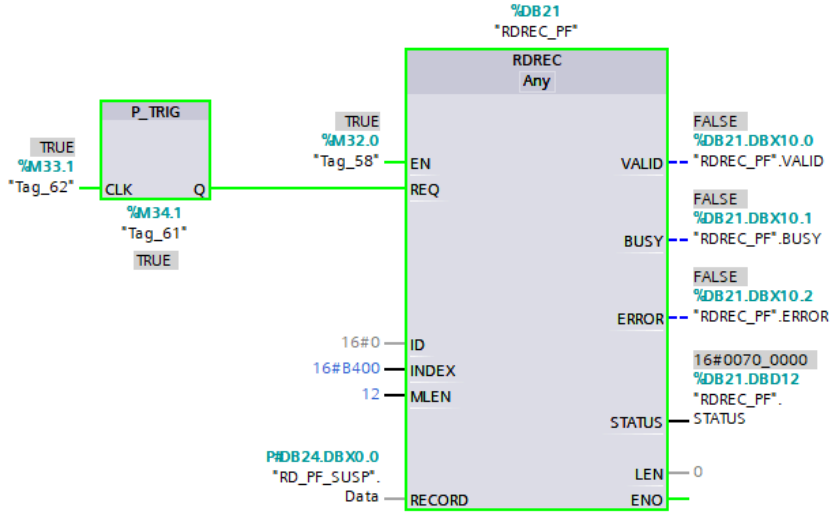
LEN = 8 Bytes for command

	Name	Data type	Offset	Start value	Comment
1	Static				
2	Data	Array[0..59] of Byte	0.0		
3	Data[0]	Byte	0.0	16#8	Call Header
4	Data[1]	Byte	1.0	16#1	Port Number (1... 8)
5	Data[2]	Byte	2.0	16#FE	Call Fixed
6	Data[3]	Byte	3.0	16#4A	Call Fixed
7	Data[4]	Byte	4.0	16#02	Call Write
8	Data[5]	Byte	5.0	16#FF	Index Port Command
9	Data[6]	Byte	6.0	16#FF	Index Port Command
10	Data[7]	Byte	7.0	16#3	Command Suspend

Figure 54: WRREC data

8.9.4.2 Read Record Suspend – port status

Use this read request to verify that the previous writing of the suspend port command has been performed successfully.



ID = 0 for addressing the IO-Link Master proxy

INDEX = 0xB400

LEN = 12 bytes, 8 bytes for command + 4 bytes for error PDU

If the suspend port command has been performed successfully, the read data look the following:

	Name	Data type	Offset	Start value	Monitor value	Comment
1	Static					
2	Data	Array[0..11] of Byte	0.0			
3	Data[0]	Byte	0.0	16#0	16#08	Call Header
4	Data[1]	Byte	1.0	16#0	16#01	Port Number (1..8)
5	Data[2]	Byte	2.0	16#0	16#FE	Call Fixed
6	Data[3]	Byte	3.0	16#0	16#4A	Call Fixed
7	Data[4]	Byte	4.0	16#0	16#00	Status 0x00 = OK, 0x80 = Error PDU
8	Data[5]	Byte	5.0	16#0	16#FF	Index Port Command
9	Data[6]	Byte	6.0	16#0	16#FF	Index Port Command
10	Data[7]	Byte	7.0	16#0	16#03	Command Suspend
11	Data[8]	Byte	8.0	16#0	16#00	Error PDU
12	Data[9]	Byte	9.0	16#0	16#00	Error PDU
13	Data[10]	Byte	10.0	16#0	16#00	SMI Job Error
14	Data[11]	Byte	11.0	16#0	16#00	SMI Job Error

The IO-Link Device can be disconnected now.

If the suspend process has not been finished on the IO-Link Master before the read record is received, a negative PROFINET response will be sent with the code "Resource busy – 0x80C2".

Possible error PDU codes:

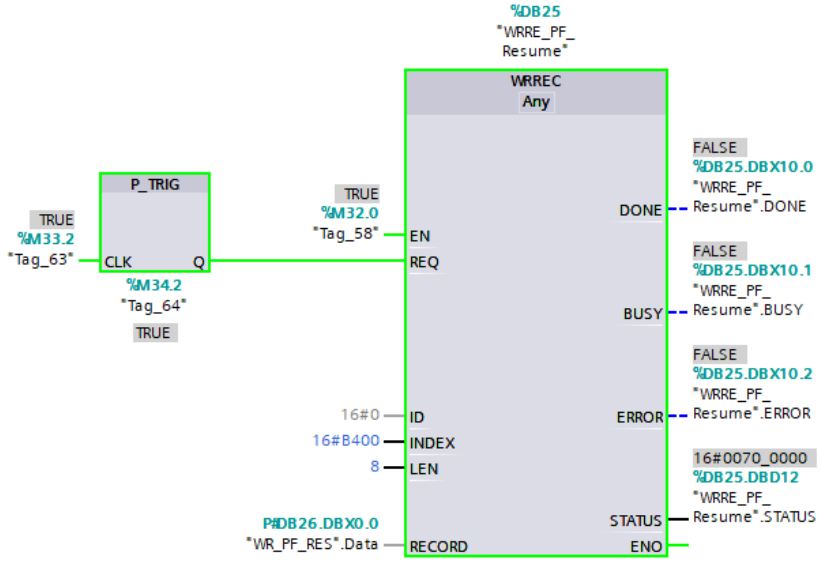
NO_ERROR	0x0000
IOL_CALL_CONFLICT	0x7000
INCORRECT_IOL_CALL	0x7001
PORT_BLOCKED	0x7002
TIMEOUT	0x8000
INVALID_PORT_NUMBER	0x8001
INVALID_IOL_INDEX	0x8002
INVALID_IOL_SUBINDEX	0x8003
NO_DEVICE	0x8004
DECODE_ERROR	0x8051
RDREC_FAULT	0x8052
WREC_FAULT	0x8053
UNEXPECTED_ERROR_SEQ	0x8054

Possible error PDU codes:

FUNCTION_ERROR	0x8055
FUNCTION_NOT_AVAILABLE	0x8056
FUNCTION_NOT_SUPPORTED	0x8057

8.9.4.3 Write Record Resume – port command

The following example illustrates how to resume an IO-Link port operation with the TIA WRREC function block (after the IO-Link Device has been connected successfully):

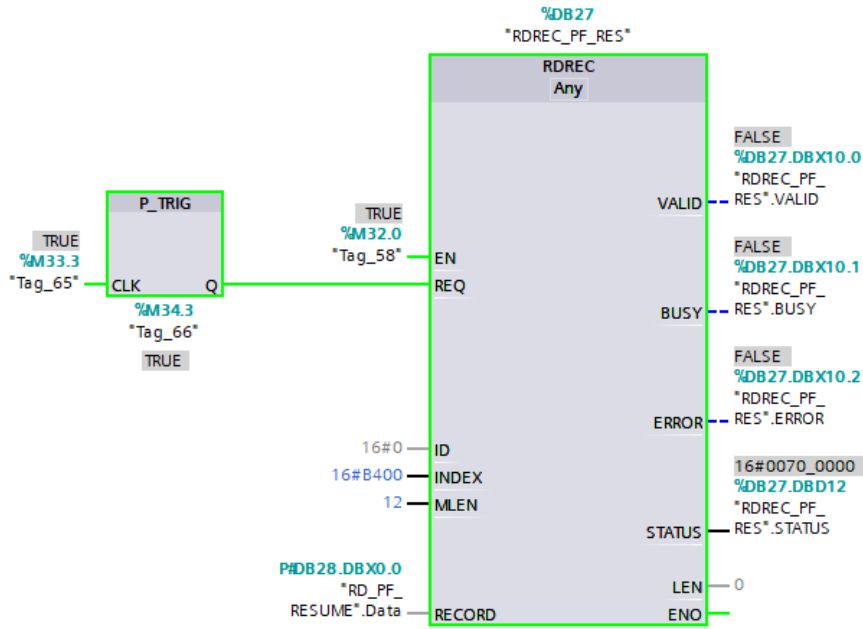


	Name	Data type	Offset	Start value	Comment
1	Static				
2	Data	Array[0..31] of Byte	0.0		
3	Data[0]	Byte	0.0	16#8	Call Header
4	Data[1]	Byte	1.0	16#1	Port Number (1...8)
5	Data[2]	Byte	2.0	16#FE	Call Fixed
6	Data[3]	Byte	3.0	16#4A	Call Fixed
7	Data[4]	Byte	4.0	16#02	Call Write
8	Data[5]	Byte	5.0	16#FF	Index Port Command
9	Data[6]	Byte	6.0	16#FF	Index Port Command
10	Data[7]	Byte	7.0	16#4	Command Resume = 0x04

Figure 55: WRREC data

8.9.4.4 Read Record Resume – port status

Use this read request to verify that the previous writing of the resume port command has been performed successfully.



ID = 0 for addressing the IO-Link Master proxy

INDEX = 0xB400

LEN = 12 bytes, 8 bytes for command + 4 bytes for error PDU

If the resume port command has been performed successfully, the read data look the following:

	Name	Data type	Offset	Start value	Monitor value	Comment
1	Static					
2	Data	Array[0..231] of Byte	0.0			
3	Data[0]	Byte	0.0	16#0	16#08	Call Header
4	Data[1]	Byte	1.0	16#0	16#01	Port Number (1..8)
5	Data[2]	Byte	2.0	16#0	16#FE	Call Fixed
6	Data[3]	Byte	3.0	16#0	16#4A	Call Fixed
7	Data[4]	Byte	4.0	16#0	16#00	Status 0x00=OK, 0x80 = Error PDU
8	Data[5]	Byte	5.0	16#0	16#FF	Index Port Command
9	Data[6]	Byte	6.0	16#0	16#FF	Index Port Command
10	Data[7]	Byte	7.0	16#0	16#04	Command Resume
11	Data[8]	Byte	8.0	16#0	16#00	Error PDU
12	Data[9]	Byte	9.0	16#0	16#00	Error PDU
13	Data[10]	Byte	10.0	16#0	16#00	SMI Job Error
14	Data[11]	Byte	11.0	16#0	16#00	SMI Job Error

If the resume process has not been finished on the IO-Link Master before the read record is received, a negative PROFINET response will be sent with the code "Resource busy – 0x80C2".

Possible error PDU codes:

NO_ERROR	0x0000
IOL_CALL_CONFLICT	0x7000
INCORRECT_IOL_CALL	0x7001
PORT_BLOCKED	0x7002
TIMEOUT	0x8000
INVALID_PORT_NUMBER	0x8001
INVALID_IOL_INDEX	0x8002
INVALID_IOL_SUBINDEX	0x8003
NO_DEVICE	0x8004
DECODE_ERROR	0x8051

Possible error PDU codes:

RDREC_FAULT	0x8052
WREC_FAULT	0x8053
UNEXPECTED_ERROR_SEQ	0x8054
FUNCTION_ERROR	0x8055
FUNCTION_NOT_AVAILABLE	0x8056
FUNCTION_NOT_SUPPORTED	0x8057

9 Process data assignment

The UNITRONIC® ACCESS IO-Link-Master uses a modular device model. Slot 1/sub-slot 1 contains the Status/Control module of the IO-Link Master. The module has 2 bytes of input data and 2 bytes of output data. It is always statically configured when selecting a UNITRONIC® ACCESS IO-Link-Master from the GSD file.

The IO-Link ports use the following sub-slots 2 through 9 of slot 1; they can have a different operating mode and data length depending on the configuration.

9.1 Process data Status/Control Module, I/O system 1.1

The Status/Control Module has one Unsigned16 (UINT16/Word) for digital input data and one Unsigned16 (UINT16/Word) for digital output data.

Status Data (input)

The two input bytes (Unsigned16) contain the status of the digital inputs. For the digital A-channel inputs, the data are also available in the input byte of the appropriate sub-slot module.

Control Data (output)

The two output bytes (Unsigned16) contain the control bits for the digital outputs of the B-channels.

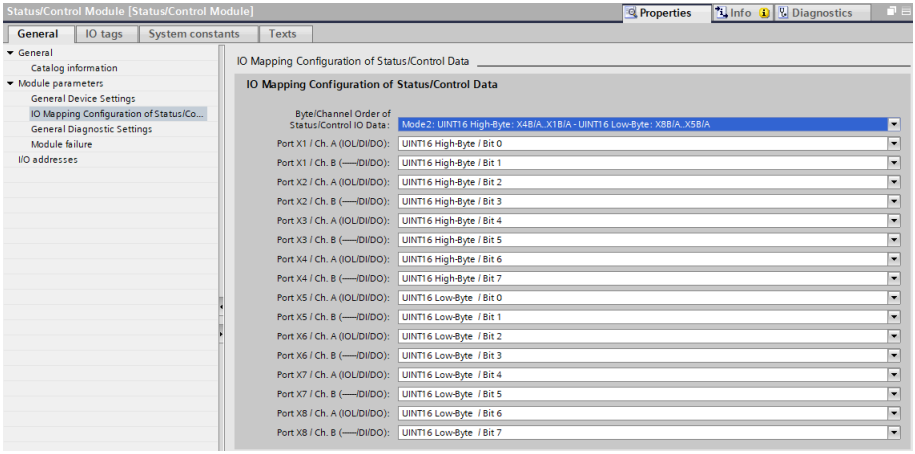
For controlling the digital A-Channels, the output *Byte 1/Bit 0* of the appropriate sub-slot module must be used.

With the *General Device Settings* parameter *Digital Out Ch. A Controlled By: Status/Control Module* the control can be switched to the *Control Bits*. In this case, the outputs cannot be controlled by the sub-slot output *Byte 1/Bit 0*.

The digital output can be controlled only by one data source.

Parameter Dependencies for Digital-I/O data mapping

Please refer to chapter I/O mapping configuration of Status/Control data on page 57 for the Bit Mapping



9.1.1 Status/Control data with Bit Mapping

The described bit mapping status/control examples are exclusively valid for the following device variants:

- UNITRONIC® ACCESS MP08IOLA08DIO

For details on bit mapping configuration, see chapters I/O mapping configuration of Status/Control data on page 57 and I/O port overview on page 20.

Key

X1A = Port 1, Channel A

UINT16 High-Byte = 1st / low address byte in a Siemens PLC

UINT16 Low-Byte = 2nd / high address byte in a Siemens PLC

(applicable for a Siemens PLC using Big-Endian format)

9.1.1.1 Mode 5

The mapping for this mode depends on the user settings.

9.1.1.2 PROFINET channel diagnostics mapping

Port	X8	X7	X6	X5	X4	X3	X2	X1
I/O Pin	2/4	2/4	2/4	2/4	2/4	2/4	2/4	2/4
I/O Channel	B/A	B/A	B/A	B/A	B/A	B/A	B/A	B/A
PN Diagn. Channel	8	7	6	5	4	3	2	1

Table 28: PROFINET channel diagnostics mapping

9.2 Process data IO-Link ports, Slot 1.2 – 1.9

The process data lengths of the IO-Link ports in COM mode depend on the IO-Link port configurations X1 – X8. Data lengths between 1 – 33 bytes of input data and/or 1 – 32 bytes of output data are configurable.

The data content can be taken from the descriptions of the IO-Link Devices. If a precise data length is not available for the IO-Link Device configuration, always select the next larger data length.

The last byte of the port input data contains the PQI byte (Port Qualifier Information). This byte is added to the IOL-Device input data by the IOL-Master.

Ch. A configuration as Digital Input

If the Port is configured as digital input, the port data length is one byte and the digital input status will be set on bit 0. The digital input status will also be mapped to the status bytes of the Status/Control Module.

The mapping mode selected for the Status/Control Module has no influence on the process data of the IO-Link ports.

INPUT	Input	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Slot 1.2	X1 Byte 1 – 33	► If the IO-Link Port is in “Digital-In” Mode, the “DI-C/Q” state (Channel A, Pin 4) will be set in Bit 0 of Byte 1. In this case no PQI Byte is available. ► The last Byte contains the PQI (Port Qualifier Information).							
Slot 1.3	X2 Byte 1 – 33								
Slot 1.4	X3 Byte 1 – 33								
Slot 1.5	X4 Byte 1 – 33								
Slot 1.6	X5 Byte 1 – 33								
Slot 1.7	X6 Byte 1 – 33								
Slot 1.8	X7 Byte 1 – 33								
Slot 1.9	X8 Byte 1 – 33								

Table 29: Input data: Sub-slots 1.2 – 1.9

Bit	Acronym	Short Description	Value	Description
0	–	Reserved	0	Reserved
			–	–
1	–	Reserved	0	Reserved
			–	–
2	NewParam	New parameter	0	*Not supported yet, don't evaluate this bit!*
			1	*Not supported yet, don't evaluate this bit!*
3	SubstDev	Substitute Device detection	0	*Not supported yet, don't evaluate this bit!*
			1	*Not supported yet, don't evaluate this bit!*
4	PortActive	Port operation	0	port deactivated via port function
			1	port activated (default)
5	DevCom	Device communication	0	no IOL-Device available
			1	IOL-Device detected and is in PREOPERATE or OPERATE state
6	DevErr	Port/Device error indication	0	no error/warning occured
			1	error/warning assigned to IOL-Device or IOL-Master port occured
7	PQ	Device Process Data validity	0	invalid I/O process data from IOL-Device
			1	valid I/O process data from device

Table 30: PQI description

OUTPUT	Output	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Slot 1.2	X1 Byte 1 – 32	▶ optional / If IO-Link Port is in “Digital-Out” Mode, the “DO-C/Q” state (Channel A, Pin 4) will be set in Bit 0 of Byte 1.							
Slot 1.3	X2 Byte 1 – 32								
Slot 1.4	X3 Byte 1 – 32								
Slot 1.5	X4 Byte 1 – 32								
Slot 1.6	X5 Byte 1 – 32								
Slot 1.7	X6 Byte 1 – 32								
Slot 1.8	X7 Byte 1 – 32								
Slot 1.9	X8 Byte 1 – 32								

Table 31: Output data: Sub-slots 1.2 – 1.9

Ch. A configuration as Digital Output

If the Port is configured as digital output, the port data length is one byte (one byte with digital output control in bit 0).

If the *General device* parameter *Digital Out Ch. A Controlled by* is set to "Status/Control Module", the output cannot be controlled by bit 0 in the port output byte.

10 Diagnostics

10.1 Detailed diagnostics description

10.1.1 Detected error of the system/sensor power supply U_S

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

Caution: Ensure that the supply voltage measured at the most remote participant is not below 21 V DC from the perspective of the system power supply.

The following IO-Link Master diagnostic is generated:

Channel number of diagnostic	0x8000 (diagnostic not channel-specific)
Channel related diagnostic code	`0x0002`
Channel related diagnostic code message	`Undervoltage`

- For **disabled** U_S supply voltage fault alarms, the U_S indicator LED is "off" in case of voltage drops below approx. 18 V. No PROFINET diagnosis will be generated.
- For **enabled** U_S supply voltage fault alarms, the U_S indicator LED is "red" in case of voltage drops below approx. 18 V.

10.1.2 Error of the actuator power supply U_L

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

- MP08IOLA08DIO

The voltage value for the incoming U_L power supply is monitored globally for the IO-Link Master. 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 IO-Link Master diagnostic is generated:

Channel number of diagnostic	0x8000 (diagnostic not channel-specific)
Channel related diagnostic code	`0x0118`
Channel related diagnostic code message	`Low voltage or over voltage of actuator power supply (UL)`
Extended description	`Check wire connection and UL power supply inclusive tolerance`

- 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.

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

In case of an overload or a short circuit between pin 1 (L+) and pin 3 (GND) on the ports (X1 .. X8), the following channel-specific diagnostic messages are generated:

Channel number of diagnostics	0x01 .. 0x08
Channel related diagnostic code	`0x1806`
Channel related diagnostic code message	`Short circuit at L+`
Extended description	`Short circuit on sensor power supply at pin 1 (L+) of I/O port. Check wire connection.`

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

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

The digital outputs on the Channel A (C/Q / pin 4) 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 device sends the following PROFINET diagnostic message in the case of a fault:

Channel number of diagnostics	0x01 .. 0x08
Channel related diagnostic code	`0x1811`
Channel related diagnostic code message	`Short circuit at C/Q`
Extended description	`Short circuit or overload on digital output at pin 4 / Ch.A of IOL port in DIO mode. Check wire connection and also power supply`

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

Attention: For variant MP08IOLA08DIO, the digital output is **supplied by the U_L power**.

Attention: For variant PN08IOLA08DIO, the digital output is **supplied by the U_S power**.

10.1.5 Overload/short circuit of the I/O port Ch. B as actuator outputs

Digital outputs on Channel B (I/Q / pin 2) are only applicable for the following device variants:

- MP08IOLA08DIO

The digital outputs on the Channel B (I/Q / pin 2) 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 "inactive" via PLC, before the output can be set again to "active".

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 device sends the following PROFINET diagnostic message in the case of a fault:

Channel number of diagnostics	0x01 .. 0x08
Channel related diagnostic code	`0x1810`
Channel related diagnostic code message	`Short circuit at I/Q`
Extended description	`Short circuit on digital output at pin 2 / Ch.B of I/O port in DO mode. Check wire connection and also power supply`

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

Attention: For variant MP08IOLA08DIO the digital output of Channel B is **supplied by the U_L power**.

10.1.6 IO-Link C/Q error

If an IO-Link Device in COM mode is unplugged, an incorrect IO-Link Device is plugged in, or an electrical fault occurs (for example due to a short circuit), an error message is generated.

If parameter Pull Plug Alarms is enabled (default):

A pull sub-module alarm will be send to the PROFINET controller. A message text like the following will be visible in the controller diagnostic buffer: “Hardware component removed or missing”.

- The dedicated green port IO-Link indicator is blinking when a device is missing.
- The dedicated red port DIA indicator is off when a device is missing.

If parameter Pull Plug Alarms is disabled & parameter Port Diagnostics is enabled:

The following diagnostic alarm will be sent to the PROFINET controller:

Channel number of diagnostics	0x01 .. 0x08
Channel related diagnostic code	0x1800
Channel related diagnostic code message	No Device/communication lost

- The dedicated green port IO-Link indicator is blinking when a device is missing.
- The dedicated red port DIA indicator is off when a device is missing.

10.1.7 Generic parameter error

When an IO-Link Master parameter will be written to an invalid address (e.g. Sub-Slot / Index) or the parameter data content is detected as invalid for the IO-Link Master, the following IO-Link Master specific diagnostic messages will be generated:

Channel number of diagnostics	0x8000 (diagnostics not channel-specific)
Channel related diagnostic code	`0x0010`
Channel related diagnostic code message	`Parameter error`

10.1.8 I/O mapping parameter error

The individual I/O data mapping parameter of the Status/Control data will be checked by the IO-Link Master. When an error is detected inside this parameter block (e.g. a bit is mapped twice), the following message will be generated:

Channel number of diagnostics	0x8000 (diagnostics not channel-specific)
Channel related diagnostic code	`0x011A`
Channel related diagnostic code message	`I/O mapping configuration faulty`

10.1.9 Process data mismatch error

The IO-Link Master checks the configured IO-Link Sub-module data length with the detected IO-Link Device data length. In dependency of the parameter *Input Fraction*, the IO-Link Master generates the following diagnostic message in case of an error:

Channel number of diagnostics	0x8000 (diagnostics not channel-specific)
Channel related diagnostic code	`0x17FF`
Channel related diagnostic code message	`Process Data mismatch`

10.1.10 Force mode diagnostic

Forcing of the I/O data is possible via the Web interface for the following device variant:

- MP08IOLA08DIO

In case of activated forcing, the following diagnostic message will be generated:

Channel number of diagnostics	0x8000 (diagnostics not channel-specific)
Channel related diagnostic code	`0x000A`
Channel related diagnostic code message	`Simulation active`

10.1.11 Internal module error detected

Internal module error states (e.g. internal abnormal states) will be reported by the following diagnostic message. For detailed information also use the Web interface of the device.

Channel number of diagnostics	0x8000 (diagnostics not channel-specific)
Channel related diagnostic code	`0x0009`
Channel related diagnostic code message	`Error`

10.2 Table of IO-Link Master diagnostic codes

The following table gives an overview of the defined diagnostic codes in PROFINET (0x0000 – 0x17FF) and IO-Link (0x1800 – 0xFFFF) specification. Not all listed codes are used.

Diagnostic code	Definition	Type
0x0000	Reserved	–
0x0002	Undervoltage	Error detection
0x0009	Error detected	Error detection
0x000A	Simulation active	Error detection
0x0010	Parameter error detected	Error detection
0x0118	Low voltage of actuator power supply (U_L). Check power supply	Error detection
0x011A	I/O mapping configuration faulty	Error detection
0x0120	Internal test pulse error detected on Channel A / Pin 1	Error detection
0x0121	Internal test pulse error detected on Channel B / Pin 5	Error detection
0x0122	Short circuit of sensor supply on Channel A / Pin 1	Maintenance
0x0123	Short circuit of sensor supply on Channel B / Pin 5	Maintenance
0x0124	Test pulse overload on Channel A / Pin 1	Error detection
0x0125	Test pulse overload on Channel B / Pin 5	Error detection
0x0126	Discrepancy error detected on digital input port	Error detection
0x0127	Short circuit error detected by signal read back on Channel A / Pin 4	Error detection
0x0128	Short circuit error detected by signal read back on Channel B / Pin 2	Error detection
0x0129	Short circuit of test pulse on Channel A / Pin 1	Error detection
0x012A	Short circuit of test pulse on Channel B / Pin 5	Error detection

Diagnostic code	Definition	Type
0x012B	Overload on Channel A / Pin 4	Error detection
0x012C	Overload on Channel B / Pin 2	Error detection
0x012D	Overtemperature of I/O device	Error detection
0x012E	Undervoltage of Sensor / System supply	Error detection
0x012F	Functional safety controller error 1	Error detection
0x0130	Functional safety controller error 2	Error detection
0x0131	Functional safety controller error 3	Error detection
0x0133	Internal Module Error detected	Error detection
0x0134	Safety program: F-I/O passivated	Error detection
0x17FF	Process Data mismatch – check submodule configuration	Error detection
0x1800	No Device	Error detection
0x1801	Startup parametrization error detected - check parameter	Error detection
0x1802	Incorrect VendorID - Inspection Level mismatch	Error detection
0x1803	Incorrect DeviceID – Inspection Level mismatch	Error detection
0x1804	Short circuit at C/Q – check wire connection	Error detection
0x1805	PHY over temperature – Check master temperature and load	Error detection
0x1806	Short circuit at L+ - check wire connection	Error detection
0x1807	Overcurrent at L+ - check power supply (e.g. L1+)	Error detection
0x1808	Device Event overflow	Error detection
0x1809	Backup inconsistency - memory out of range	Error detection
0x180A	Backup inconsistency – identity fault	Error detection
0x180B	Backup inconsistency – parameter storage unspecific error detected	Error detection
0x180C	Backup inconsistency – upload fault	Error detection
0x180D	Parameter inconsistency – download fault	Error detection
0x1810	Short circuit at I/Q – check wiring	Error detection
0x1811	Short circuit at C/Q (if digital output) – check wiring	Error detection
0x1812	Overcurrent at I/Q – check load	Error detection
0x1813	Overcurrent at C/Q (if digital output) – check load	Error detection

Diagnostic code	Definition	Type
0x1814 to 0x1EFF	Reserved	
0x1F00 to 0x1FFF	Vendor specific	
0x2000 to 0x2FFF	Safety extensions	
0x3000 to 0x3FFF	Wireless extensions	
0x4000 to 0x5FFF	Reserved	
0x6000	Invalid cycle time	Error detection
0x6001	Revision fault	Error detection
0x6002	ISDU batch failed	Error detection
0x6003 to 0xFF20	Reserved	Error detection
0xFF21	Reserved	Notification
0xFF22	Reserved	Notification
0xFF23	Reserved	Notification
0xFF23	Reserved	Notification
0xFF24	Reserved	Notification
0xFF25	Reserved	Notification
0xFF26[3]	Port status changed	Notification
0xFF27[2]	Data Storage upload completed and new data object available	Notification
0xFF28 to 0xFF30	Reserved	
0xFF31	Reserved	Notification
0xFF32 to 0xFFFF	Reserved	Notification

10.3 IO-Link Device diagnostics in PROFINET

Diagnostics (Events) of the IO-Link Device which are sent to the IO-Link-Master, are reported to the PROFINET controller via a standard channel diagnostic or an extended channel diagnostic.

3 For IO-Link Master internal use

Standard channel diagnostic message:

Channel number of diagnostics	0x01 - 0x08
Channel related diagnostic code	Depends on IO-Link Device diagnostics
Channel related diagnostic code message	Depends on IO-Link Device diagnostics

Extended channel diagnostic message:

Channel number of diagnostics	0x01 - 0x08
Ext. channel related diagnostic code	IO-Link Device event code

For IO-Link event codes in the range of 0x8000 - 0x7FFF, the MSB bit will be set to "0" in the PROFINET extended channel diagnostic code.

Event Code

Diagnostic code reported by the IO-Link Device. Use the IO-Link Device documentation to interpret the error message.

Channel Number

1 - 8 of the IO-Link Master port whose connected device is reporting an error.

10.4 Table of IO-Link Device diagnostic codes

The following table shows the defined diagnostic codes (events) of the IO-Link specification. Use the IO-Link Device documentation for vendor specific codes.

Diagnostic code	Definition	Type
0x0000	No malfunction	Notifcation
0x1000	General malfunction – unknown error detected	Error detection
0x1001 to 0x17FF	Reserved	
0x1800 to 0x18FF	Vendor specific	
0x1900 to 0x3FF	Reserved	
0x4000	Temperature fault detected – Overload	Error detection
0x4001 to 0x420F	Reserved	
0x4210	Device temperature overrun – Clear source of heat	Warning
0x4211 to 0x421F	Reserved	
0x4220	Device temperature underrun – Insulate Device	Warning
0x4221 to 0x4FFF	Reserved	
0x5000	Device hardware fault detected – Device exchange	Error detection
0x5001 to 0x500F	Reserved	
0x5010	Component malfunction – Repair or exchange	Error detection
0x5011	Non volatile memory loss detected – Check batteries	Error detection
0x5012	Batteries low – Exchange batteries	Warning
0x5013 to 0x50FF	Reserved	
0x5100	General power supply fault detected – Check availability	Error detection
0x5101	Fuse blown/open – Exchange fuse	Error detection
0x5102 to 0x510F	Reserved	
0x5013 to 0x50FF	Reserved	

Diagnostic code	Definition	Type
0x5100	General power supply fault detected – Check availability	Error detection
0x5101	Fuse blown/open – Exchange fuse	detection
0x5102 to 0x510F	Reserved	
0x5110	Primary supply voltage overrun – Check tolerance	Warning
0x5111	Primary supply voltage underrun – Check tolerance	Warning
0x5113 to 0x5FFF	Reserved	
0x6000		
0x6001 to 0x631F	Reserved	
0x6320	Parameter error detected – Check data sheet and values	Error detection
0x6321	Parameter missing – Check data sheet	Error detection
0x6322 to 0x634F	Reserved	
0x6350	Reserved	
0x6351 to 0x76FF	Reserved	
0x7700	Wire break of a subordinate device – Check installation	Error detection
0x7701 to 0x770F	Wire break of subordinate device 1 …device 15 – Check installation	Error detection
0x7710	Short circuit – Check installation	Error detection
0x7711	Ground fault detected – Check installation	Error detection
0x7712 to 0x8BFF	Reserved	
0x8C00	Technology specific application fault detected – Reset Device	Error detection
0x8C01	Simulation active – Check operational mode	Warning
0x8C02 to 0x8C0F	Reserved	
0x8C10	Process variable range overrun – Process Data uncertain	Warning
0x8C11 to 0x8C1F	Reserved	
0x8C20	Measurement range exceeded – Check application	Error detection
0x8C21 to 0x8C2F	Reserved	
0x8C30	Process variable range underrun – Process Data uncertain	
0x8C31 to 0x8C3F	Reserved	
0x8C40	Maintenance required – Cleaning	Warning

Diagnostic code	Definition	Type
0x8C41	Maintenance required – Refill	Warning
0x8C42	Maintenance required – Exchange wear and tear parts	Warning
0x8C43 to 0x8C9F	Reserved	
0x8CA0 to 0x8DFF	Vendor specific	
0x8E00 to 0xAFFF	Reserved	
0xB000 to 0xB0FF	Reserved for Safety extensions	
0xB100 to 0xBFFF	Reserved for Profiles	
0xC000 to 0xFF90	Reserved	
0xFF91	Internal	Notification
0xFF92 to 0xFFAF	Reserved	
0xFFB0 to 0xFFB7	Reserved for Wireless extensions	
0xFFB8 to 0xFFFF	Reserved	

11 IIoT functionality

The UNITRONIC® ACCESS 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 UNITRONIC® ACCESS 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 UNITRONIC® ACCESS variants provide user administration, which is also applicable for accessing and configuring the IIoT protocols. This allows you to manage all modification options for the device settings via personalized user authorizations.

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

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

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

11.1 MQTT

MQTT functions are **only** applicable for the following UNITRONIC® ACCESS variants:

- MP08IOLA08DIO

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

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

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

11.1.1 MQTT configuration

In **delivery state**, MQTT functions are **disabled**. The MQTT client can be configured either using the Web interface or directly via a JSON object sent in an HTTP/HTTPS request. For more information see chapter MQTT configuration - Quick start guide on page 170.

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: "**unitronic**")
will-enable	boolean	If true, the device provides a last will message to the broker	true / **false**
will-topic	string	The topic for the last will message.	(Default: **null**)
auto-publish	boolean	If true, all enabled domains will be published automatically in the specified interval.	**true** / false
publish-interval	number	The publish interval in ms if auto-publish is enabled. Minimum is 250 ms.	**2000**
publish-identity	boolean	If true, all identity domain data will be published	**true** / false
publish-config	boolean	If true, all config domain data will be published	**true** / false
publish-status	boolean	If true, all status domain data will be published	**true** / false
publish-process	boolean	If true, all process domain data will be published	**true** / false
publish-devices	boolean	If true, all IO-Link Device domain data will be published	true / **false**
commands-allowed	boolean	Master switch for MQTT commands. If false, the device will not subscribe to any command topic, even if specific command topics are activated below.	true / **false**
force-allowed	boolean	If true, the device accepts force commands via MQTT.	true / **false**
reset-allowed	boolean	If true, the device accepts restart and factory reset commands via MQTT.	true / **false**
config-allowed	boolean	If true, the device accepts configuration changes via MQTT.	true / **false**

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

Table 32: 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 158.

11.1.2 MQTT topics

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

11.1.2.1 Base topic

For all UNITRONIC® ACCESS 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 33: Base topic variables on page 158.

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

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

Table 33: Base topic variables

Example:

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

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

`Base-Topic/domain/….`

There are the following domains:

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

Table 34: Data domains

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

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

Table 35: 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 36: Use case examples

11.1.2.2 Publish topic

Overview of all publish JSON data for the defined topics:

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

Table 37: Identity/gateway

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

Table 38: Config/gateway

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

Table 39: Status/gateway

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

Table 40: Process/gateway

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

Table 41: Identity/port/1 .. 8

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

Table 42: Config/port/1 .. 8

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

Table 43: Status/port/1 .. 8

11.1.2.3 Command topic (MQTT Subscribe)

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

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

`[base-topic]/command`

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

[…]/forcing

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

Property	Data type	Example values	Remarks
forcemode	boolean	true / false	Forcing Authority: on/off
digital	array (Table 45: Force object: Digital on page 167)		
iol	array (Table 46: Force object: IOL (IO-Link devices only) on page 167)		

Table 44: 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 45: Force object: Digital

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

Table 46: Force object: IOL (IO-Link devices only)

[…]/config

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

Property	Data type	Example values	Remarks
portmode	array (Table 48: Config object: Portmode on page 168)		
ip_address	string	"192.168.1.5"	
subnet_mask	string	"255.255.255.0"	
gateway	string	"192.168.1.100"	

Table 47: Config object properties

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

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

Table 48: 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 49: Reset object properties

[…]/publish

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

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

11.1.3 MQTT configuration - Quick start guide

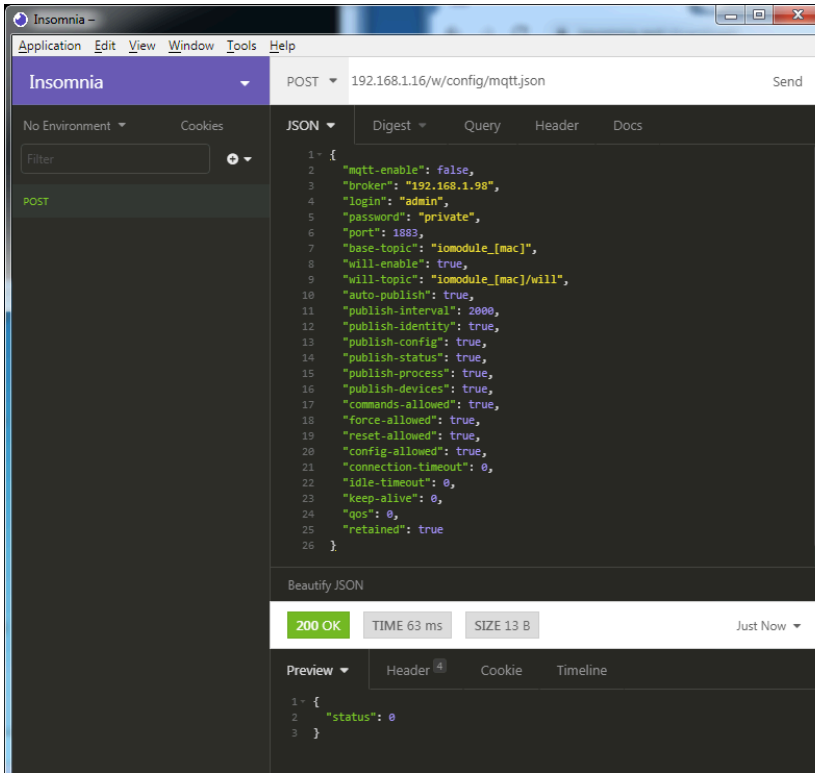
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

11.1.3.1 MQTT configuration via JSON

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

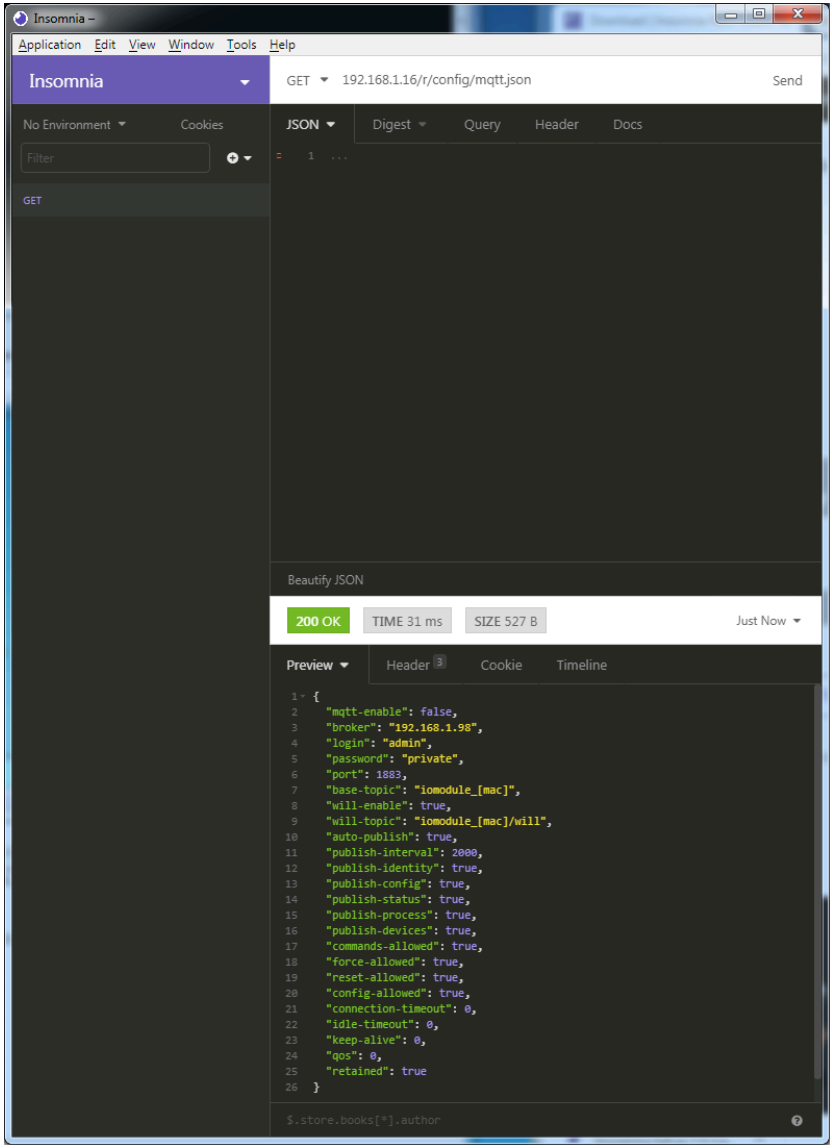
2. Configure MQTT:

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



3. Read MQTT:

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



11.2 OPC UA

OPC UA functions are **only** applicable for the following UNITRONIC® ACCESS variants:

- MP08IOLA08DIO

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. UNITRONIC® ACCESS provides an OPC UA server on field device level, with which an OPC UA client can connect for information exchange secure in transmission.

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

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

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

Table 50: Non-supported OPC UA features according to the IO-Link Companion Specification

11.2.1 OPC UA configuration

In **delivery state**, OPC UA functions are **disabled**. The OPC UA Server can be configured either using the Web interface or directly via a JSON Object sent in an HTTP/HTTPS request. For more information see OPC UA configuration - Quick start guide on page 176.

The configuration URL is:

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

The configuration can also read back as a JSON file:

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

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

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

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

Table 51: OPC UA Configuration

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

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

Response:

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

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

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

Examples:

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

{"status": 0}

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

11.2.2 OPC UA address space

OPC UA provides different services on the UNITRONIC® ACCESS 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 UNITRONIC® ACCESS devices. The objects and information displayed depend on the device variant used.

11.2.3 OPC UA configuration - Quick start guide

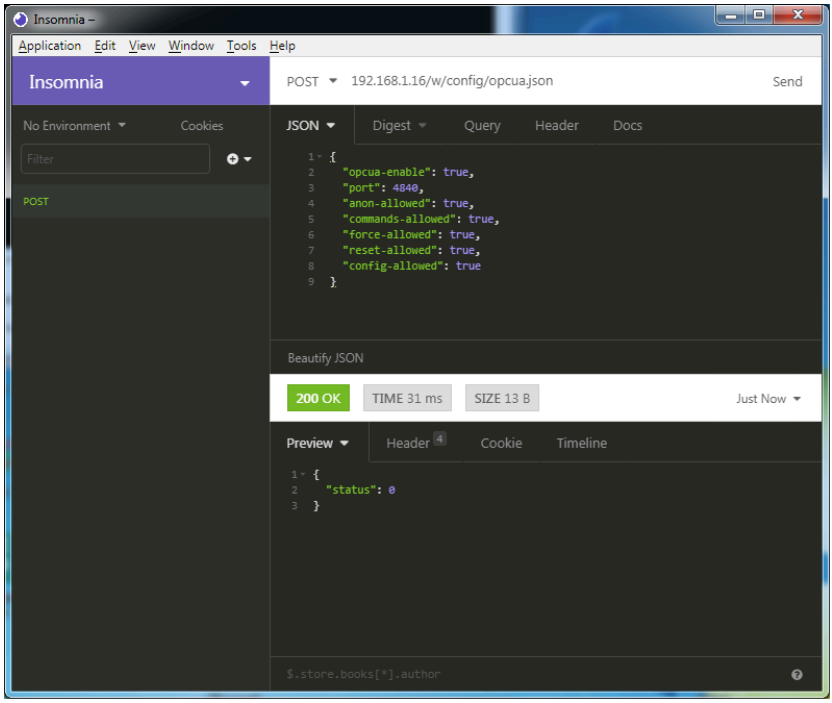
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

11.2.3.1 OPC UA configuration via JSON

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

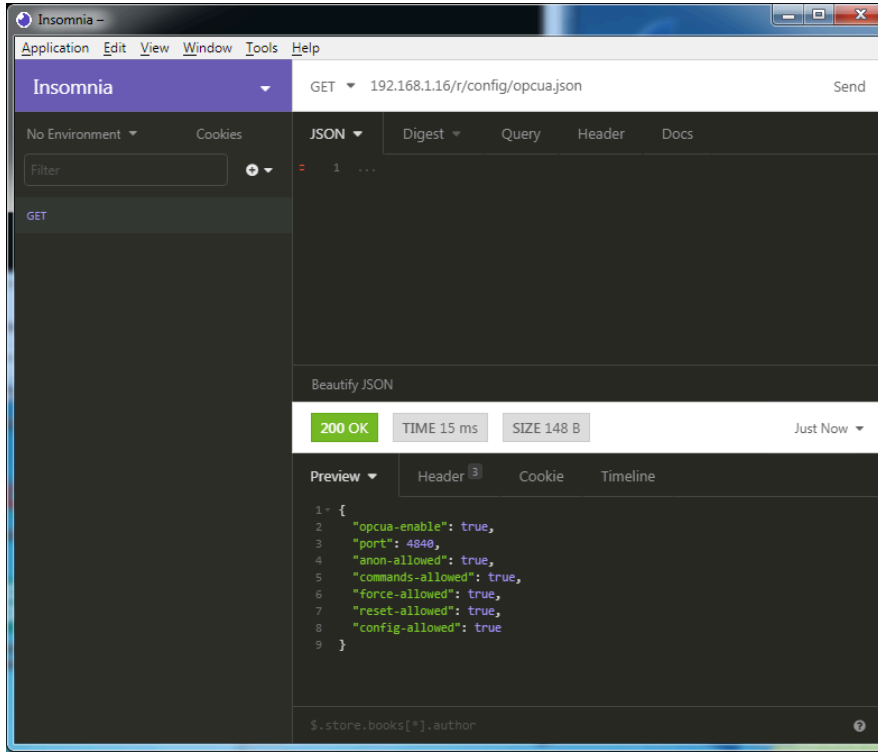
2. Configure OPC UA:

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



3. Read OPC UA:

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



11.3 REST API

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

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

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

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

   `JSON_Integration_10222_V100_Mar20.pdf`

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

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

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

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

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

Table 52: Support of REST API features according to the IO-Link specification

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

11.3.1 Standard device information

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

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

11.3.2 Structure

Name	Data type	Description	Example
name	string	Device name	"MP08IOLA08DIO"
order-id	string	Ordering number	"381166717"
fw-version	string	Firmware version	"V.11.2.0.0 - 08.08.2024"
hw-version	string	Hardware version	"V.1.00"
mac	string	MAC address of the device	"7C F9 5C 4C CC CE"
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
iol	IOL Object	Contains all IO-Link related information such as events, port states, device parameters.	
iol/diagGateway	array of DIAG	Array of currently active device/ gateway related events	
iol/diagMaster	array of DIAG	Array of currently active IOL-Master related events	
iol/ports	array of PORT (8)	Contains one element for each IO-Link port	
CHANNEL Object			
name	string	Name of channel	
type	number	Hardware channel type as number: 0 = DIO 1 = Input 2 = Output 3 = Input/Output 4 = IO-Link 5 = IOL AUX 6 = IOL AUX with DO 7 = IOL AUX with DO. Can be deactivated. 8 = Channel not available	
type_text	string	Textual representation of the channel type	
config	number	Current configuration of the channel: 0 = DIO 1 = Input 2 = Output 3 = IO-Link 4 = Deactivated 5 = IOL AUX	
config_text	string	Textual representation of the current config	
inputState	boolean	Input data (producing data) bit to the PLC	
outputState	boolean	Output data bit to the physical output pin	

Name	Data type	Description	Example
forced	boolean	True, if the output pin of this channel is forced	
simulated	boolean	True, if the input value to the PLC of this channel is simulated	
actuatorDiag	boolean	True, if the output is in short circuit / overload condition	
sensorDiag	boolean	True, if the sensor supply (Pin 1) is in short circuit / overload condition	
maxOutputCurrent_mA	number	Maximum output current of the output in mA	
current_mA	number	Measured current of the output in mA (if current measurement is available)	
voltage_mV	number	Measured voltage of this output in mV (if voltage measurement is available)	
PORT Object			
port_type	string	Textual representation of the IO-Link port type	
iolink_mode	number	**Current port mode:** 0 = Inactive 1 = Digital output 2= Digital input 3 = SIO 4 = IO-Link	
iolink_text	string	Textual representation of the current port mode	"Digital Input"
aux_mode	number	Indicates the configured mode for the Pin 2: 0 = No AUX 1 = AUX output (always on) 2 = Digital output (can be controlled by cyclic data) 3 = Digital input	
aux_text	string	Textual representation of the current aux mode	"AUX Output"
cq_mode	number	Port mode according to IOL specification	
iq_mode	number	Pin2 mode according to IOL specification	

Name	Data type	Description	Example
port_status	number	Port status according to IOL specification	
ds_fault	number	Data storage error number	
ds_fault_text	string	Textual data storage error.	
device	DEVICE Object	IO-Link device parameters. → Null if no IO-Link communication active	
diag	array of DIAG (n)	Array of port related events	
DIAG Object			
error	number	Error code	
source	string	Source of the current error.	"device" "master"
eventcode	number	Event code according to IO-Link specification	
eventqualifier	number	Event qualifier according to IO-Link specification	
message	string	Error message	"Supply Voltage fault"
DEVICE Object		Standard parameters of the IOL-Device	
device_id	number		
vendor_id	number		
serial	string		
baudrate	string	Baudrate (COM1,2,3)	
cycle_time	number	Cycle time in microseconds	
input_len	array of numbers (n)	IOL input length in bytes	
output_len	array of numbers (n)	IOL output length in bytes	
input_data	array of numbers (n)	IOL input data	
output_data	array of numbers (n)	IOL output data	
pd_valid	number	"1", if IOL input data is valid	
pdout_valid	number	"1", if IOL output data is valid	
FORCING Object		Forcing information of the device	
forcingActive	boolean	Force mode is currently active	

Name	Data type	Description	Example
forcingPossible	boolean	True, if forcing is possible and force mode can be activated	
ownForcing	boolean	True, if forcing is performed by REST API at the moment	
forcingClient	string	Current forcing client identifier	
digitalOutForced	array of numbers (2)	The force values of all 16 digital output channels.	
digitalOutMask	array of numbers (2)	The forcing mask of all 16 digital output channels.	
digitalInForced	array of numbers (2)	The force values of all 16 digital input channels.	
digitalInMask	array of numbers (2)	The forcing mask of all 16 digital input channels.	

11.3.3 Configuration and forcing

Method: POST

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

Parameters: None

Post-Body: JSON Object

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

Table 53: Root object

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

Table 54: 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 55: Digital object

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

Table 56: IOL object

11.3.4 Reading and writing ISDU parameters

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

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

11.3.4.1 Reading ISDU

Method: POST

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

Parameters: port (0-7)

Example:

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

Post-Body: JSON array of read ISDU object

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

Table 57: Read ISDU object

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

Table 58: Read ISDU response object

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

Table 59: Read ISDU data object

11.3.4.2 Writing ISDU

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

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

Table 60: Write ISDU object

Response: Write ISDU response object

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

Table 61: Write ISDU response object

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

Table 62: Write ISDU data object

Attention: For UNITRONIC® ACCESS device variants with HTTPS feature, https:// must be used in front of <ip> for every REST API.

11.3.5 Upload and process an IODD file

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

Perform the following work steps:

1. Check file upload status

   Send request: GET file_upload

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

   Expected response:

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

   Check the status ID. If status is '0', you can start a new iodd upload process. For reference, see tables Table 63: Status ID and meaning on page 196 and Table 64: Error ID and meaning on page 197. Proceed with the next step.
2. Initiate file upload

   Send request: POST file_upload

   Content-Type: application/json

   Purpose: Send details about the file to be uploaded.

   Expected response:

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

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

The action will allways be iodd.

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

The correct content type has to be set.

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

3. Upload file content

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

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

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

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

4. Monitor upload status

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

   Purpose: Get the current file upload status.

   Expected response:

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

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

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

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

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

Table 63: Status ID and meaning

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

Table 64: Error ID and meaning

11.3.6 Example: Reading ISDU

ISDU read request

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

Response

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

11.3.7 Example: Writing ISDU

ISDU write request

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

Response

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

11.4 CoAP server

The CoAP server functions are **only** applicable for the following UNITRONIC® ACCESS variants:

- MP08IOLA08DIO

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

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

11.4.1 CoAP configuration

In delivery state, CoAP functions are *disabled*. The CoAP server can be configured either using the Web interface or directly via a JSON object sent in an HTTP/HTTPS request. For more information see chapter CoAP configuration - Quick start guide on page 203.

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 65: CoAP configuration

CoAP response:

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

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

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

Examples:

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

{"status": 0}

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

11.4.2 REST API access via CoAP

A connection to the CoAP server running on the UNITRONIC® ACCESS multi-protocol variants can be established via the following URL:

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

For UNITRONIC® ACCESS, the following REST API Requests (JSON format) can be accessed via a CoAP endpoint:

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

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

Table 66: REST API access via CoAP

11.4.3 CoAP configuration - Quick start guide

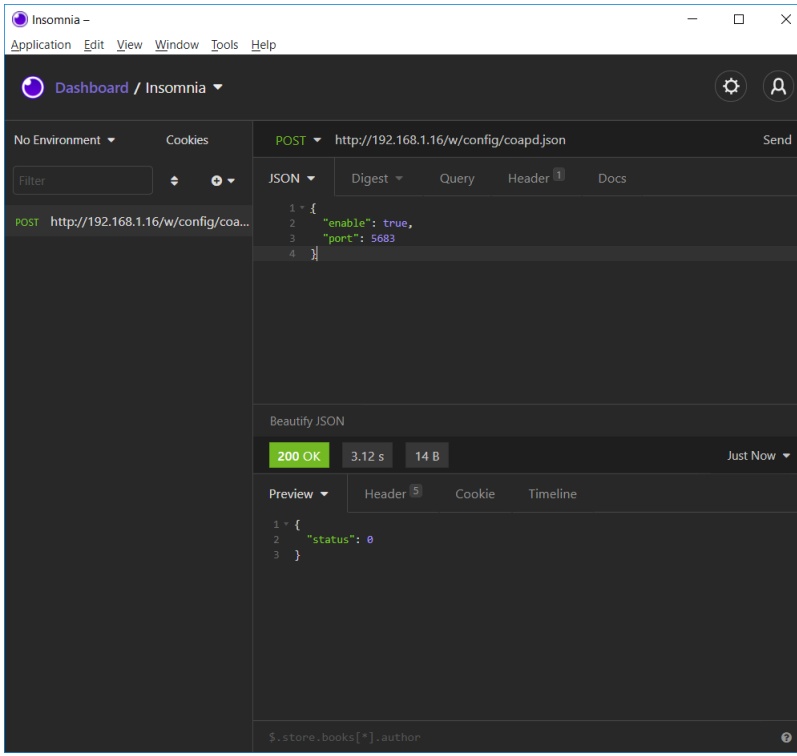
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

11.4.3.1 CoAP configuration via JSON

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

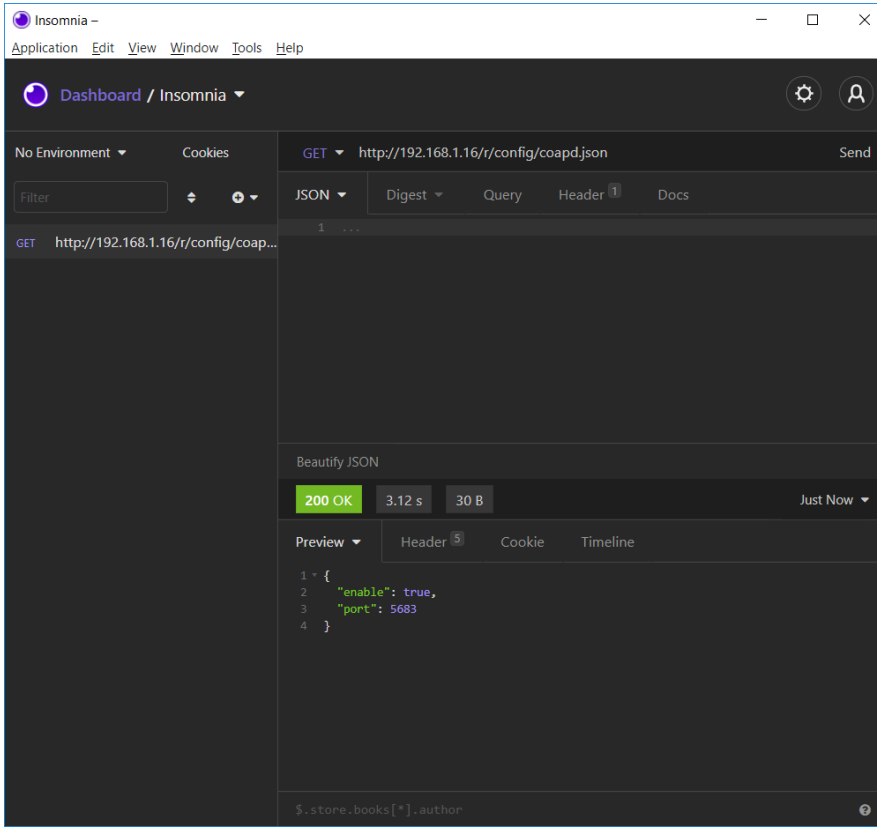
2. Configure CoAP:

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



3. Read CoAP configuration:

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



11.5 Syslog

Syslog functions are **only** applicable for the following UNITRONIC® ACCESS variants:

- MP08IOLA08DIO

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

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

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

11.5.1 Syslog configuration

In **delivery state**, Syslog functions are **disabled**. The Syslog client can be configured either using the Web interface or directly via a JSON object sent in an HTTP/HTTPS request. For more information see chapter Syslog configuration - Quick start guide on page 208.

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 67: Syslog configuration

Syslog response:

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

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

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

Examples:

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

{"status": 0}

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

11.5.2 Syslog configuration - Quick start guide

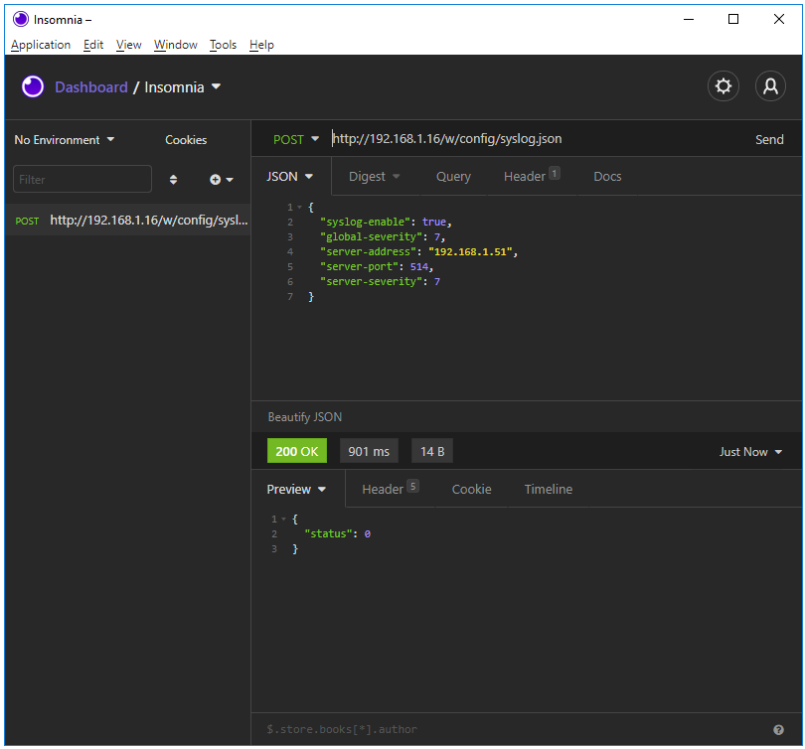
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

11.5.2.1 Syslog configuration via JSON

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

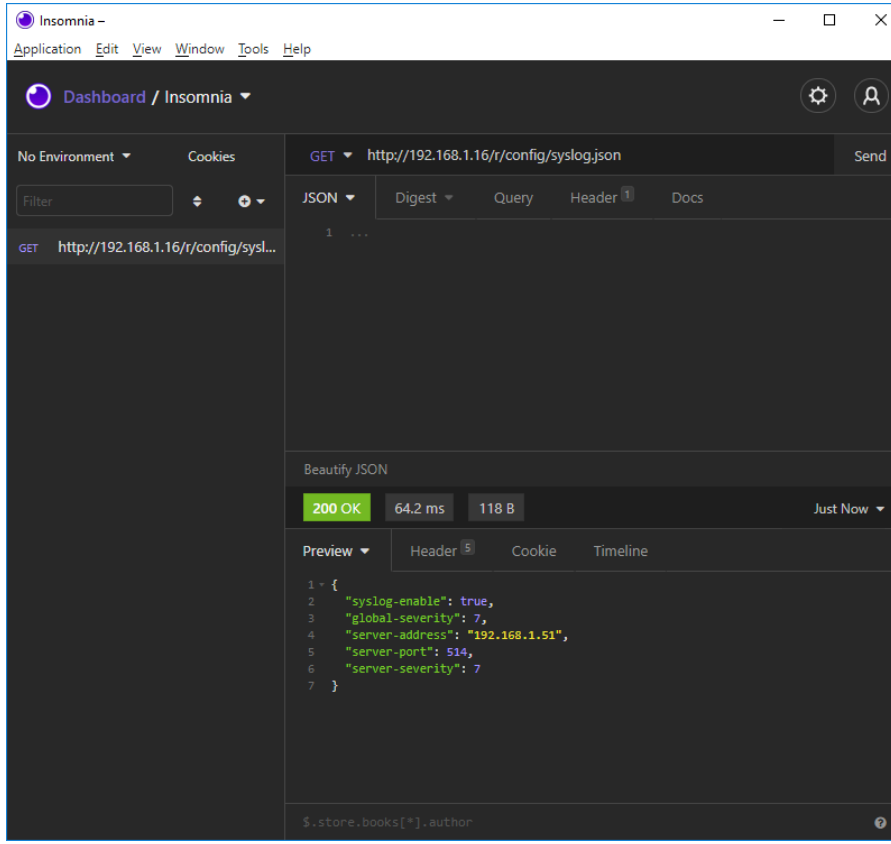
2. Configure Syslog:

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



3. Read Syslog configuration:

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



11.6 Network Time Protocol (NTP)

The NTP function is **only** applicable for the following UNITRONIC® ACCESS variant:

- MP08IOLA08DIO

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

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

11.6.1 NTP configuration

In **delivery state**, the NTP client is **disabled**. The NTP client can be configured either using the Web interface or directly via a JSON object sent in an HTTP/HTTPS request. For more information see chapter NTP configuration - Quick start guide on page 212.

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 68: NTP configuration

NTP response:

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

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

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

Examples:

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

{"status": 0}

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

11.6.2 NTP configuration - Quick start guide

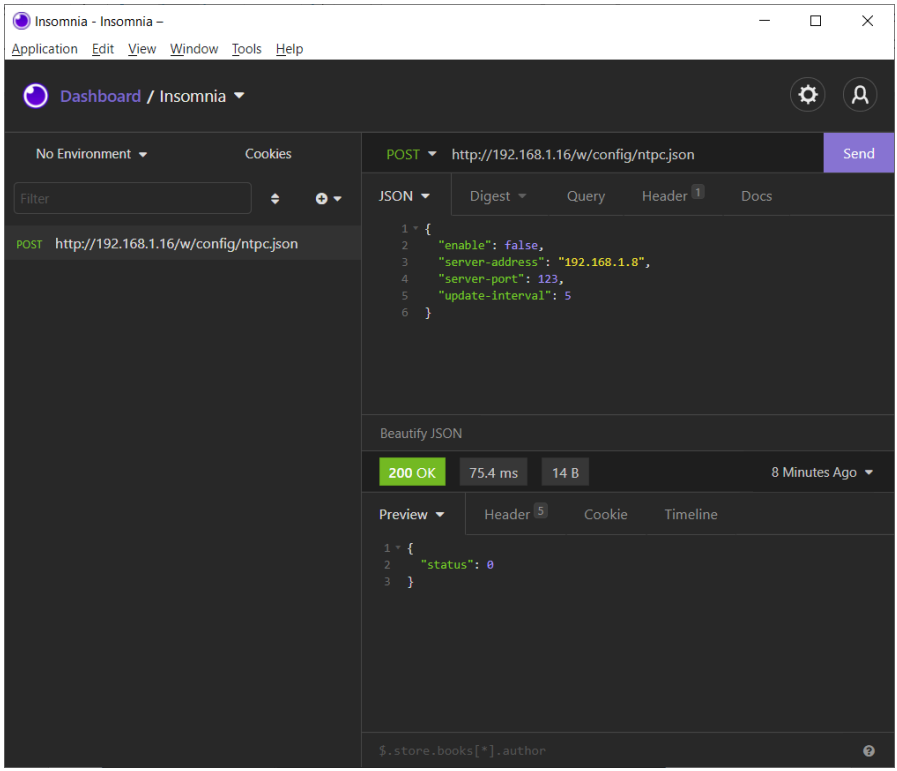
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

11.6.2.1 NTP configuration via JSON

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

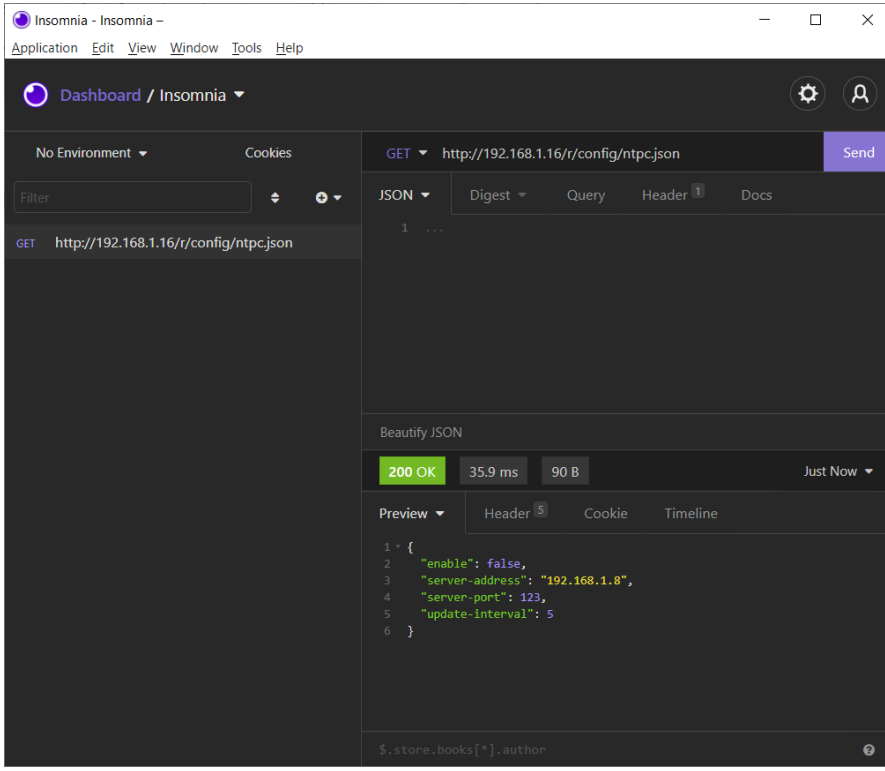
2. Configure NTP:

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



3. Read NTP configuration:

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



12 The integrated Web server

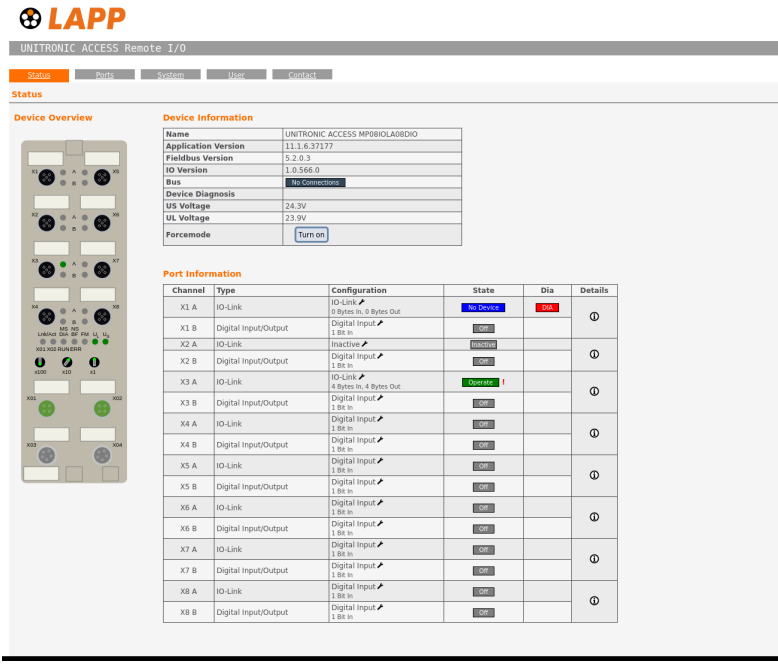
All device variants are equipped with an integrated Web server which makes functions for the device configuration and the display of status and diagnostic information available via a Web interface.

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

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

12.1 UNITRONIC® ACCESS MP08... variants

12.1.1 The Status page



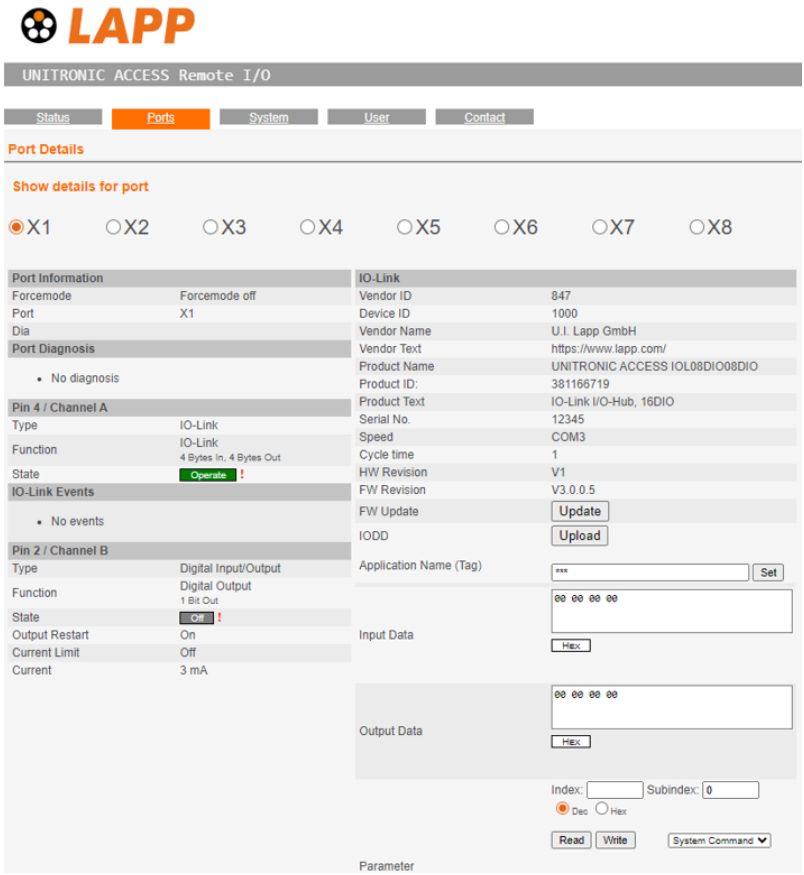
The status page provides a quick overview of the current state of the device.

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

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

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

12.1.2 The Ports page



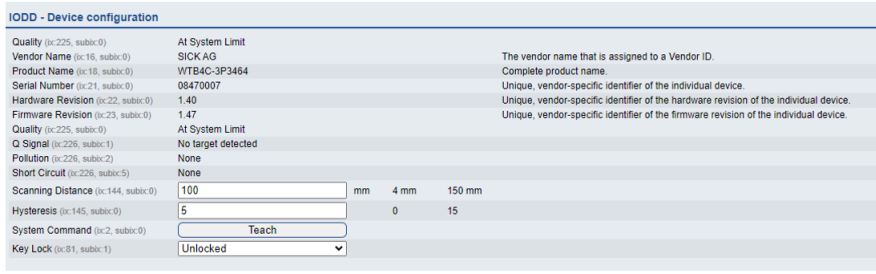
The page shows detailed port information. In the field **Port Diagnosis**, incoming and outgoing diagnostics are displayed as clear text. **Pin 2** and **Pin 4** contain information about the configuration and state of the port. For IO-Link ports, additional information relating to the connected sensor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

12.1.2.1 IODD upload

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

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

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



12.1.3 The System page



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

IP Settings

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

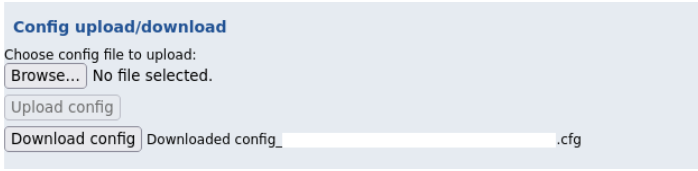
For PROFINET, this is only useful during commissioning. Normally, the PLC sets the IP address at start-up by detecting the PROFINET module via its device name.

12.1.3.1 License

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

12.1.3.2 Config upload/download

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



The following settings will be stored inside this file:

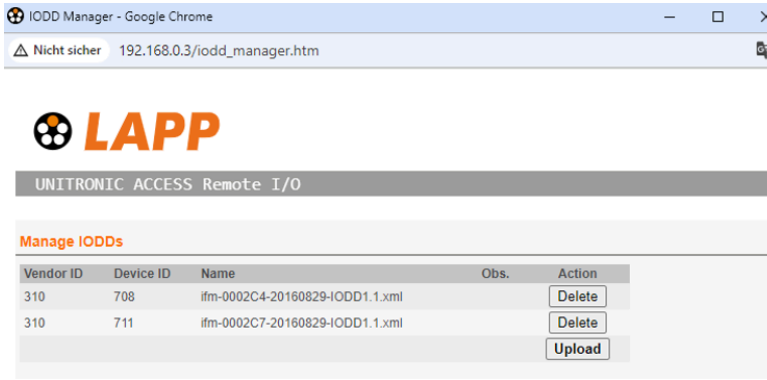
Scope	Type	Setting	Options	Details
Gateway		deviceID		To check device identity.
	iol	applicationSpecificTag		
	iol	functionTag		
	iol	locationTag		
		forcing		Enable/disable forcing
		channel_count		
		network configuration	ip	
			snMask	
			gw	
			source	1 - manual 2 - dhcp 3 - rotary 4 - dcp
Channel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

Scope	Type	Setting	Options	Details
	digital	force value		
	digital	simulate		
	digital	sim value		
	digital	inputPolarity		
	digital	autorestart mode		
	digital	inputFilter100us		
	digital	currentLimit		
	digital	outputRestartMode		
	digital	failsafeMode		
	digital	surveillanceTimeouMs		
OPC UA		opcua	opcua-enable	
			port	
			anon-allowed	
			commands-allowed	
			force-allowed	
			reset-allowed	
			config-allowed	
	digital		dcu-allowed	
MQTT		mqtt	mqtt-enable	
			broker	
			login	
			password	
			port	
			base-topic	
			will-enable	
			will-topic	
			auto-publish	
			publish-interval	
			publish-identity	

Scope	Type	Setting	Options	Details
			publish-config	
			publish-status	
			publish-process	
	iol		publish-devices	
			commands-allowed	
			force-allowed	
			reset-allowed	
			config-allowed	
			qos	
SYSLOG		syslog	syslog-enable	
			global-severity	
			server-address	
			server-port	
			server-severity	
COAP		coap	enable	
			port	
NTP		ntpc	enable	
			server-address	
			server-port	
			update-interval	

12.1.3.3 IODD

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



12.1.3.4 Restart device

The module initializes a software reset.

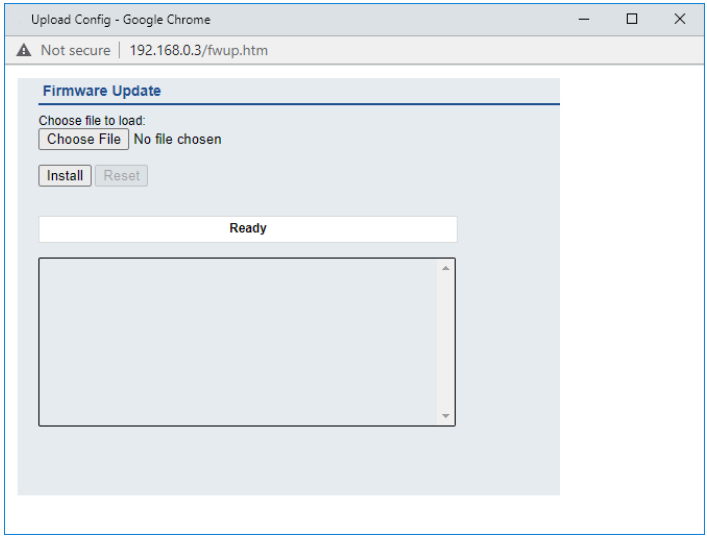
12.1.3.5 Reset configuration to factory defaults

The module restores to the default factory settings.

12.1.3.6 Firmware update

The module initializes a Firmware update.

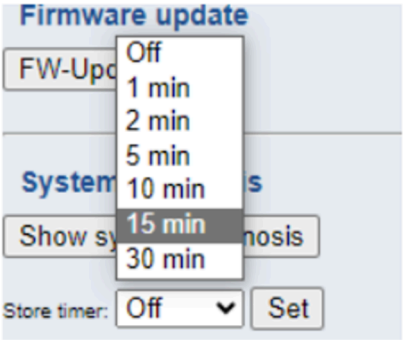
For a firmware update choose the **.ZIP* container available on our website or ask our support team. Afterwards follow the instructions shown on your screen.



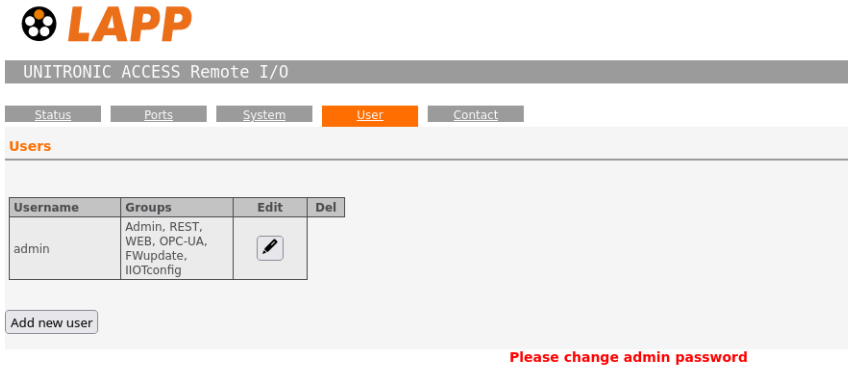
12.1.3.7 System diagnosis

All Syslog messages will be displayed in a ring buffer with 512 entries. By activation of the 'Store timer', the buffer content will be stored nonvolatile in the selected interval of 1, 2, 5, 10, 15 or 30 minutes.

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



12.1.4 The User page



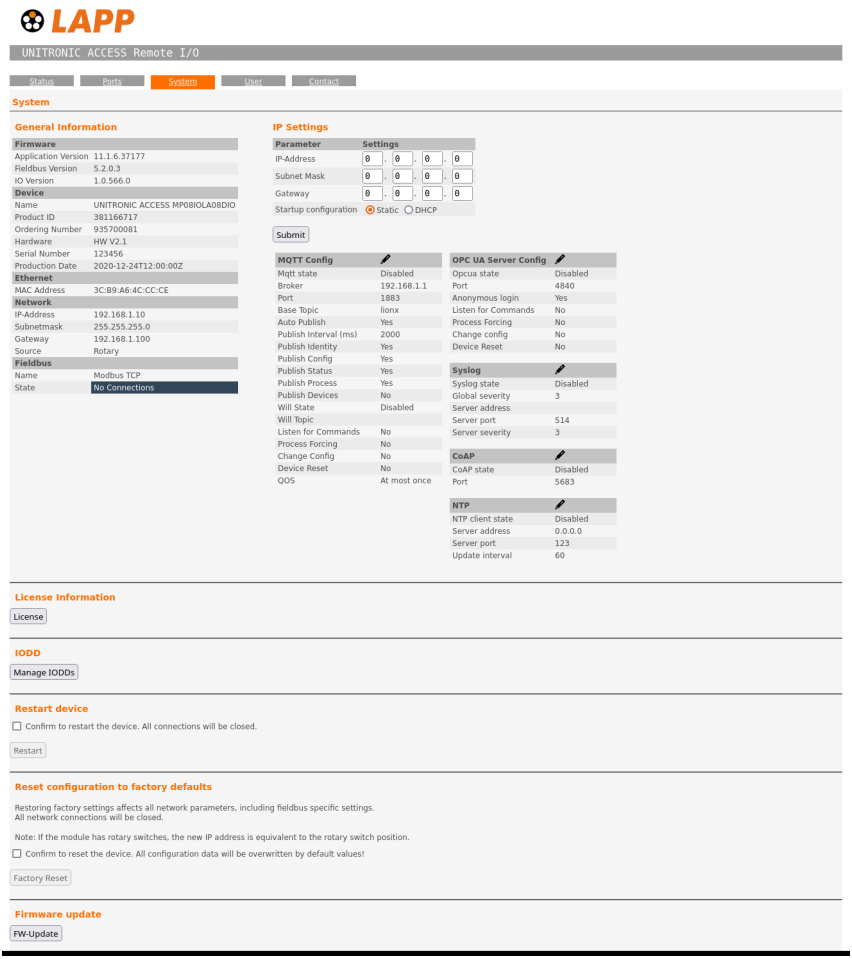
The User page provides the user management of the Web interface. New users with access rights "Admin" or "Write" can be added here. For security reasons please change the default admin password immediately after configuring the device.

Default user login data:

- User: admin
- Password: private

12.2 UNITRONIC® ACCESS single-protocol PN0.../EIP0.../EC0.../MTCP0.../CCL0... variants

12.2.1 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

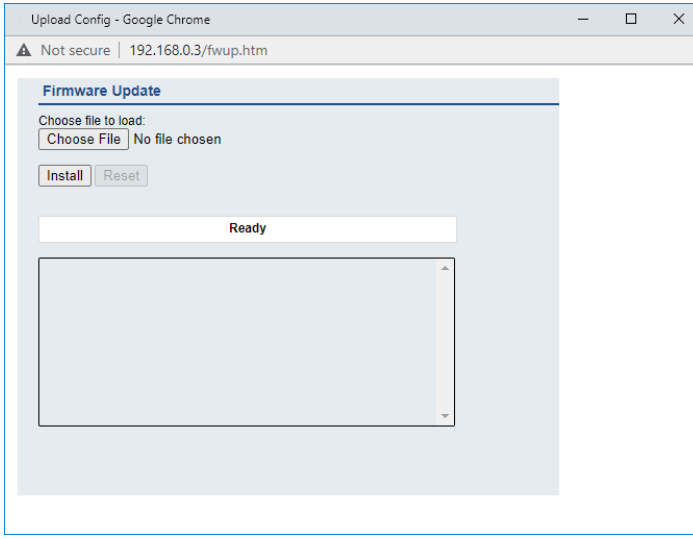
Use this parameter to change the current IP address of the module.

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.



13 IODD

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

- MP08IOLA08DIO

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

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

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

13.1 IO-Link Device parameters and ISDU requests

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

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

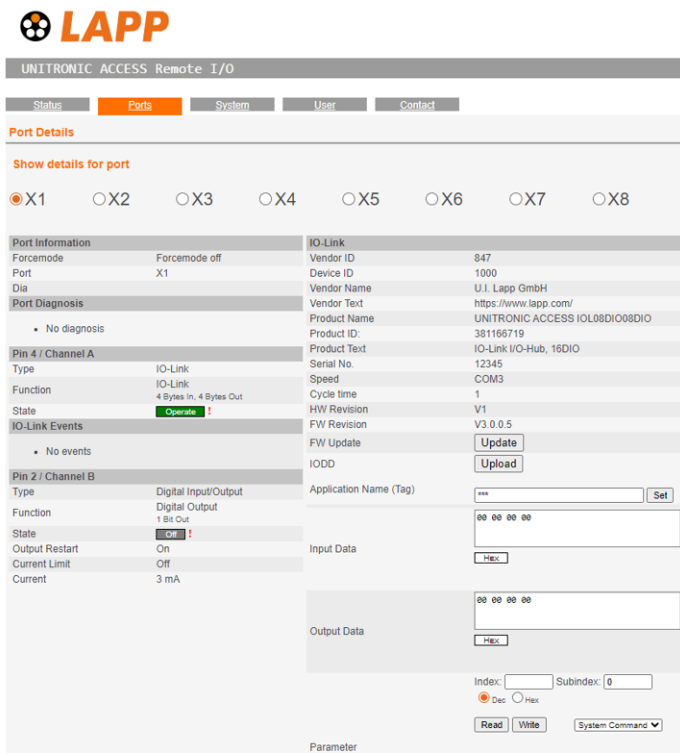
A vendor can use additional parameters and therefore more indices for their devices in order to provide additional configuration options. These vendor specific parameters can be described in an IODD. The "IODD on Module" feature of the UNITRONIC® ACCESS IO-Link Masters can read and parse this information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

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

13.2 Web GUI functionality

All of the "IODD on Module" features are accessible via the UNITRONIC® ACCESS Web interface.

13.2.1 Port Details page



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

IODD buttons

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

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

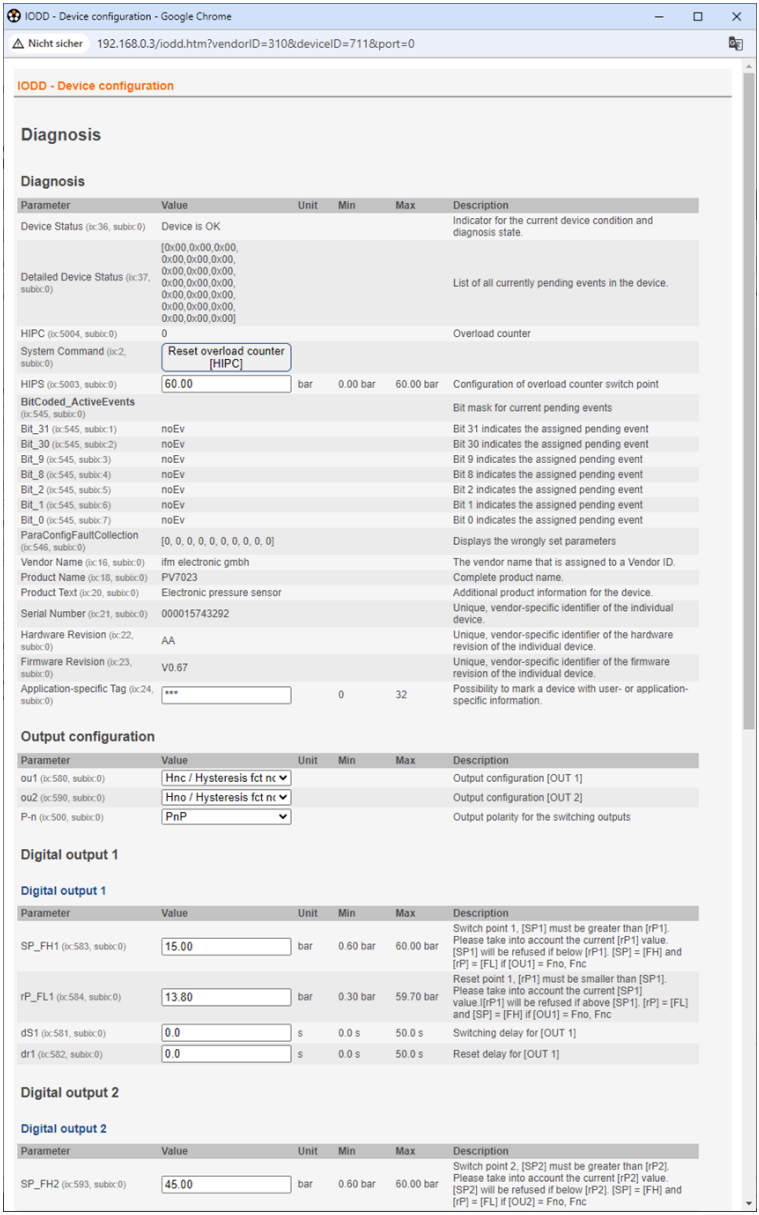
If there is a matching IODD for the currently connected device already stored in the system, the button *CONFIGURE* is shown in the interface. By clicking this button, the Parameter Page will open to configure the device.

Process data

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

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

13.2.2 Parameters page



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

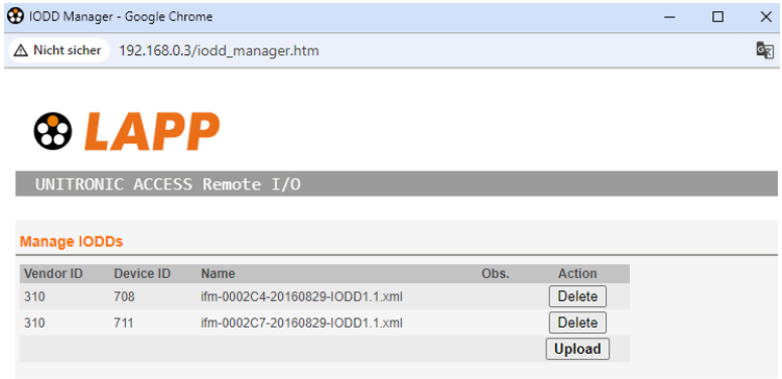
The stored IODD reads the parameter meta data, such as names, units, min/max values, descriptions etc. The values will be obtained directly from the connected device. For that reason it may take several seconds until the page is updated.

If not already saved into the browser, you will be asked for your credentials to continue. A valid user access with Web Interface group membership is needed in order to edit the device parameters. After the registration, enabled values can be changed. Disabled values cannot be changed and may be marked as "read-only" in the IODD. All values are directly written back to the device after any change.

Limitations

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

13.2.3 IODD Management page



The IODD Management Page can be accessed via the System page displaying all IODDs that are currently stored in the system. All IODDs

matching connected devices are marked. On the IODD Management page, you can manually delete any IODD in the system.

14 Technical data

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

14.1 General

Protection class (Only applies if the connectors are screwed together or if protective caps are used.)[4]	IP65 IP67 IP69K	
Ambient temperature (during operation and storage)	MP08...	-40 °C .. +70 °C (-40 °F .. +158 °F)
	PN08IOLA.../EIP08IOLA.../EC08IOLA.../ MTCP08IOLA.../CCL08IOLA...	-20 °C .. +60 °C (-4 °F .. +140 °F)
Weight	UNITRONIC® ACCESS 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 69: General information

4 Not under UL investigation.

14.2 PROFINET protocol

Protocol	PROFINET IO device V2.44
Conformance Class	C
Netload Class	III
Update cycle	1 ms
GSDML file	GSDML-V2.xx-U.I. Lapp-UNITRONIC ACCESS-yyyymmdd.xml
Transmission rate	100 Mbit/s, full duplex
Transmission procedure Autonegotiation	100BASE-TX is supported
Vendor ID	18F_H
Device ID	0x0608 (same for all UNITRONIC® ACCESS variants)
Supported Ethernet protocols	Ping ARP LLDP SNMPv1 (network diagnostics) ▶ Read Community: public ▶ Write Community: private DCP HTTP / HTTPS TCP/IP MRP Client
PROFINET feature	Fast Start UP (Prioritized startup) Shared Device
Switch functionality	Integrated IRT is supported
PROFINET interface Connections Autocrossing	2 M12 sockets, 4-pin, D-coded (see pin assignments) 2 M12 Hybrid male/female, 8-pin is supported
Electrically isolated Ethernet ports -> FE	2000 V DC

Table 70: PROFINET protocol

14.3 Power supply of the module electronics/ sensors

Port X03, X04	M12-L-coded Power, connector/socket, 5-pole Pin 1 / Pin 3		
Nominal voltage U_S	24 V DC (SELV/PELV)		
Current U_S	Max. 16 A		
Voltage range	21 .. 30 V DC		
Power consumption of module electronics	Typically 160 mA (+/-20 % at U_S nominal voltage)		
Power supply interruption	Max. 10 ms		
Voltage ripple U_S	Max. 5 %		
Current consumption sensor system (L+ / Pin 1)	MP08...	Port X1 .. X8 (Pin 1)	max. 4 A per port (at $T_{ambient}$ = 30° C)
	PN08IOLA.../ EIP08IOLA.../ EC08IOLA.../ MTCP08IOLA.../ CCL08IOLA...	Port X1 .. X8 (Pin 1)	max. 2 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 71: Information on the power supply of the module electronics/ sensors

14.4 Power supply of the actuators

14.4.1 IO-Link Class A devices (U_L)

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

Table 72: Information on the power supply of the actuators

14.5 I/O ports Channel A (Pin 4)

MP08...	Port X1 .. X8	Class A	IOL, DI, DO	M12 socket, 5-pin, Pin 4
PN08IOLA.../EIP08IOLA.../ EC08IOLA.../ MTCP08IOLA.../ CCL08IOLA...	Port X1 .. X8	Class A	IOL, DI, DO	

Table 73: IO-Link Master ports: Functional overview for Ch. A (Pin 4)

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

Input connection	MP08... PN08IOLA.../EIP08IOLA.../EC08IOLA.../ MTCP08IOLA.../CCL08IOLA...		Type 1 as per IEC 61131-2
Nominal input voltage	24 V DC		
Input current	Typically 3 mA		
Channel type	Normally open, p-switching		
Number of digital inputs	MP08... PN08IOLA.../EIP08IOLA.../ EC08IOLA.../MTCP08IOLA.../ CCL08IOLA...	X1 .. X8	8
Status indicator	yellow LED		
Diagnostic indicator	red LED per channel		

Table 74: I/O ports Ch. A (Pin 4) configured as digital inputs

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

Attention: For variant MP08IOLA08DIO, the digital outputs of Channel A is **supplied by the U_L power** when parameterized to "High-Side Switch" mode.

Attention: For variant PN08IOLA08DIO, the digital outputs of Channel A is **supplied by the U_S power**.

Output type	normally open, p-switching (parameterized to "High-Side Switch" mode)	
Nominal output voltage per channel Signal status “1” Signal status “0”	 min. (U_S -1 V) **or** min. (U_L -1 V) depending on the device variant max. 2 V	
Max. output current per device	MP08...	9 A (power supplied via U_L)
Max. output current per device	PN08IOLA.../EIP08IOLA.../ EC08IOLA.../MTCP08IOLA.../ CCL08IOLA...	4 A (power supplied via U_S)
Max. output current per channel[5]	MP08... (X1 .. X8)	2 A (power supplied via U_S)
	PN08IOLA.../EIP08IOLA.../ EC08IOLA.../MTCP08IOLA.../ CCL08IOLA... (X1 .. X8)	0.5 A (power supplied via U_S) 0.25 A for **UL applications**
Short-circuit/overload protected	yes/yes	
Behavior in case of short circuit or overload	deactivation with automatic power-on (parameterized)	
Number of digital outputs	MP08... (X1 .. X8) PN08IOLA.../EIP08IOLA.../ EC08IOLA.../MTCP08IOLA.../ CCL08IOLA... (X1 .. X8)	8
Status indicator	yellow LED per output	
Diagnostic indicator	red LED per channel	

Table 75: I/O ports Ch. A (Pin 4) configured as digital outputs

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

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

IO-Link Master specification	v1.1.3, IEC 61131-9	
Communication rates	4.8 kbaud (COM 1) 38.4 kbaud (COM 2) 230.4 kbaud (COM 3)	
Line lengths in the IO-Link Device	max. 20 m	
Number of IO-Link ports	MP08... (X1 .. X8) PN08IOLA.../EIP08IOLA.../EC08IOLA.../ MTCP08IOLA.../CCL08IOLA... (X1 .. X8)	8
Min. IO-Link cycle time	400 µs	

Table 76: Configured as IO-Link port in COM mode

14.6 I/O ports Channel B (Pin 2)

MP08...	Port X1 .. X8	Class A	DI, DO	M12 socket, 5-pin, Pin 2
PN08IOLA.../EIP08IOLA.../ EC08IOLA.../ MTCP08IOLA.../ CCL08IOLA...	Port X1 .. X8	Class A	DI	

Table 77: IO-Link Master ports: Functional overview for Ch. B (Pin 2)

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

Input connection	MP08... PN08IOLA.../EIP08IOLA.../EC08IOLA.../ MTCP08IOLA.../CCL08IOLA...		Type 1 as per IEC 61131-2
Nominal input voltage	24 V DC		
Input current	Typically 3 mA		
Channel type	Normally open, p-switching		
Number of digital inputs	MP08...	X1 .. X8	8
Number of digital inputs	PN08IOLA.../EIP08IOLA.../ EC08IOLA.../MTCP08IOLA.../ CCL08IOLA...	X1 .. X8	8
Status indicator	white LED		
Diagnostic indicator	red LED per channel		

Table 78: I/O ports Ch. B (Pin 2) configured as digital inputs

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

Attention: For variant MP08IOLA08DIO, the digital outputs of Channel B is **supplied by the U_L power**.

Attention: For variant PN08IOLA08DIO, the digital outputs of Channel B is **supplied by the U_S power**.

Output type	normally open, p-switching	
Nominal output voltage per channel Signal status “1” Signal status “0”	min. (U_S -1 V) **or** min. (U_L -1 V) **or** min. (U_{AUX} -1 V) depending on the device variant max. 2 V	
Max. output current per device	MP08...	9 A (power supplied via U_L)
Max. output current per device	PN08IOLA.../ EIP08IOLA.../ EC08IOLA.../ MTCP08IOLA.../ CCL08IOLA...	4 A (power supplied via U_S)
Max. output current per channel [6]	MP08...	2 A (power supplied via U_S)
Max. output current per channel [6]	PN08IOLA.../ EIP08IOLA.../ EC08IOLA.../ MTCP08IOLA.../ CCL08IOLA...	0 A (no outputs on Ch. B)
Short-circuit/overload protected	yes/yes	
Behavior in case of short circuit or overload	deactivation with automatic power-on (parameterized)	
Number of digital outputs	MP08...	8
Number of digital outputs	PN08IOLA.../ EIP08IOLA.../ EC08IOLA.../ MTCP08IOLA.../ CCL08IOLA...	–
Status indicator	white LED per output	
Diagnostic indicator	red LED per channel	

Table 79: I/O ports Ch. B (Pin 2) configured as digital outputs

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

14.7 LEDs

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

LED	Color	Description
BF	Red	Bus fault. No configuration, no or slow physical connection.
	Red flashing at 2 Hz	Link exists but no communication link to the PROFINET controller.
	OFF	PROFINET controller has established an active connection to the device.
DIA	Red	PROFINET module diagnostic alarm active.
	Red flashing at 1 Hz	Watchdog time-out; fail safe mode is active.
	Red flashing at 2 Hz, 3 sec	DCP signal service is initiated via the bus.
	Red double flash	Firmware update
	OFF	None of the above conditions.
MS	Green	Device is ready for operation.
	Green flashing	Device is ready but not configured yet.
	Red	Serious error that cannot be resolved.
	Red flashing	Minor error that can be resolved Example: An incorrect or contradictory configuration is classified as a minor error.
	Flashing alternately: Red / Green	The device is performing a self-test.
	OFF	The device is switched off.

LED	Color	Description
NS	Green	Connected: The device has at least one connection.
	Green flashing	No connection: The device has no connection. IP address exists.
	Red	Duplicate IP address: The device has detected that the assigned IP address is already being used by another device.
	Red flashing	Connection has exceeded time limit or connection interrupted.
	Flashing alternately: Red / Green	The device is performing a self-test.
	OFF	The device is switched off or has not been assigned an IP address.

Table 80: Information on the LED colors

14.8 Data transfer times

The following tables give an overview of the internal data transfer times of the UNITRONIC® ACCESS IO-Link Master with a connected IO-Link Device as digital I/O extension (LAPP article IOL08DIO08DIO 16DIO Hub with a minimum cycle time of 1 ms).

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

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

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

The configurable digital input filter value on IOL08DIO08DIO was set to "off" (0 ms).

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

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

Use case 1:

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

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

Use case 2:

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

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

15 Accessories

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

https://www.lapp.com

Manual

EtherNet/IP

UNITRONIC ACCESS IO-Link Master

Multi-protocol:
MP08IOLA08DIO (8 × IO-Link Class A)

Single-protocol:
EIP08IOLA08DIO (8 × IO-Link Class A)



Contents

1 About this manual 10

1.1 General information 10
1.2 Explanation of symbols 11
1.2.1 Use of danger information 11
1.2.2 Use of general information 11
1.3 Version information 11

2 Safety instructions 12

2.1 Intended use 12
2.2 Qualified personnel 13

3 Designations and synonyms 14

4 System description 18

4.1 About UNITRONIC® ACCESS 18
4.2 Device variants 19
4.3 I/O port overview 21

5 Overview of product features 23

5.1 EtherNet/IP product features 23
5.2 I/O port features 25
5.3 Integrated Web server 26
5.4 Security features 27
5.5 Other features 28

6 Assembly and wiring 29

6.1 General information 29
6.2 Outer dimensions 30
6.2.1 UNITRONIC® ACCESS multi-protocol variants 30
6.2.2 UNITRONIC® ACCESS single-protocol variants with EtherNet/IP 31
6.2.3 Notifications 32
6.3 Port assignments 33
6.3.1 Ethernet ports, M12 socket, 4-pin, D-coded 33
6.3.2 Power supply with M12 power L-coded 34
6.3.2.1 IO-Link Master with Class A ports 35
6.3.2.2 IO-Link Master with Class A/B ports 36
6.3.3 I/O ports as M12 sockets 36
6.3.3.1 IO-Link ports (Class A) 37

7 Starting operation 38

7.1 EDS file 38
7.2 MAC addresses 38
7.3 State on delivery 39
7.4 Setting network parameters 40
7.4.1 IP address for UNITRONIC® ACCESS variants 40
7.4.2 IP address for UNITRONIC® ACCESS single-protocol variants 40
7.5 Setting the rotary encoding switches 41
7.5.1 EtherNet/IP selection and IP configuration via rotary encoding switches 44
7.5.2 Factory reset 45

8 Configuration EtherNet/IP 46

8.1 Assembly types 46

8.2 Connections 47
8.2.1 IO-Link parameters (Exclusive Owner) 48
8.2.2 IO-Link parameters (Listen Only) 48
8.2.3 IO-Link parameters Omron (Exclusive Owner) 49
8.2.4 IO-Link parameters Min (Exclusive Owner) 50

9 Configuration parameters 51

9.1 General settings 52
9.1.1 Force mode lock 53
9.1.2 Web interface lock 53
9.1.3 Report U_L/U_{AUX} supply voltage fault 53
9.1.4 Report DO Fault without U_L/U_{Aux} 53
9.1.5 CIP object configuration lock 53
9.1.6 External configuration lock 54
9.1.7 IO Mapping Mode 54
9.1.8 General Settings (parameter) 55
9.2 Channel settings 57
9.2.1 IO Mapping (Ch1 .. 16) 59
9.2.2 DO Surveillance Timeout (Ch1 .. 16) 59
9.2.3 DO Failsafe (Ch1 .. 16) 59
9.2.4 DO Restart Mode (Ch1 .. 16) 60
9.2.5 DO Switch Mode (Ch1 .. 16) 60
9.2.6 DI Logic (Ch1 .. 16) 61
9.2.7 DI Filter (Ch1 .. 16) 61
9.2.8 DI Latch 61
9.2.9 DI Extension 62
9.2.10 Channel Mode (Ch1 .. 16) 64
9.2.11 Port mode for Channel A (Pin 4) 66
9.2.12 Port mode for Channel B (Pin 2) 66
9.3 IO-Link diagnosis settings 67
9.3.1 IO-Link Master Diagnosis 67
9.3.2 IO-Link Device Error 67
9.3.3 IO-Link Device Warning 68
9.3.4 IO-Link Device Notification 68

9.3.5 IO-Link Device Diagnosis Port 1 .. 8 68
9.4 IO-Link Port 1 .. 8 settings 69
9.4.1 Output Data Size 72
9.4.2 Input Data Size 73
9.4.3 Input Data Extension 73
9.4.4 Output Data Swapping Mode 73
9.4.5 Output Data Swapping Offset 74
9.4.6 Input Data Swapping Mode 74
9.4.7 Input Data Swapping Offset 74
9.4.8 IOL Failsafe 74
9.4.9 Port Mode 75
9.4.10 IO-Link Mode 76
9.4.11 Validation and Backup 76
9.4.12 Vendor ID 80
9.4.13 Device ID 80
9.4.14 Cycle Time 80

10 Process data assignment 81

10.1 Consuming data image (output) 82
10.1.1 Digital output channel control 82
10.1.2 IO-Link output data 82
10.2 Producing data image (input) 83
10.2.1 Digital input channel status 83
10.2.2 General diagnostics 83
10.2.3 Sensor diagnostics 84
10.2.4 Actuator/U_L/U_{Aux} diagnostics 85
10.2.5 IO-Link diagnostics 85
10.2.6 IO-Link input data 86
10.3 Sample applications 89
10.3.1 Process data images – default configuration 89
10.3.2 Process data images with modified data sizes 91

11 Configuration and operation with Rockwell Automation Studio 5000® 95

11.1 Basic commissioning 95
11.2 Add-On Instruction (AOI) 100

12 CIP object classes 110

12.1 EtherNet/IP object classes 110
12.1.1 Identity Object (0x01) 111
12.1.2 Assembly Object (0x04) 114
12.1.3 Discrete Input Point Object (0x08) 115
12.1.4 DLR Object (0x47) 116
12.1.5 QoS Object (0x48) 118
12.1.6 TCP/IP Object (0xF5) 120
12.1.7 Ethernet Link Object (0xF6) 122
12.1.8 LLDP Management Object (0x109) 125
12.2 Vendor specific object classes 127
12.2.1 General Settings Object (0xA0) 127
12.2.2 Channel Settings Object (0xA1) 129
12.2.3 IO-Link Diagnosis Settings Object (0xA2) 132
12.2.4 IO-Link Port Settings Object (0xA3) 133
12.2.5 IO-Link Failsafe Parameter Object (0xA4) 136
12.2.6 IO-Link Device Parameter Object (0xA5) 137
12.3 Message configuration in Rockwell Automation Studio 5000® 141
12.3.1 Get/Set attribute single 142
12.3.2 Get/Set ISDU data 143

13 Diagnostics processing 145

13.1 Error of the system/sensor power supply 145
13.2 Error of the auxiliary/actuator power supply 146

13.3 Overload/short-circuit of the I/O port sensor supply outputs 147
13.4 Overload/short circuit of the digital outputs 148
13.5 IO-Link COM error 150
13.6 IO-Link validation error 151
13.7 IO-Link device diagnostics 152

14 IIoT functionality 153

14.1 MQTT 154
14.1.1 MQTT configuration 154
14.1.2 MQTT topics 157
14.1.2.1 Base topic 157
14.1.2.2 Publish topic 160
14.1.2.3 Command topic (MQTT Subscribe) 165
14.1.3 MQTT configuration - Quick start guide 169
14.1.3.1 MQTT configuration via JSON 169
14.2 OPC UA 171
14.2.1 OPC UA configuration 172
14.2.2 OPC UA address space 174
14.2.3 OPC UA configuration - Quick start guide 175
14.2.3.1 OPC UA configuration via JSON 175
14.3 REST API 177
14.3.1 Standard device information 179
14.3.2 Structure 180
14.3.3 Configuration and forcing 186
14.3.4 Reading and writing ISDU parameters 189
14.3.4.1 Reading ISDU 189
14.3.4.2 Writing ISDU 191
14.3.5 Upload and process an IODD file 193
14.3.6 Example: Reading ISDU 197
14.3.7 Example: Writing ISDU 197
14.4 CoAP server 198
14.4.1 CoAP configuration 198
14.4.2 REST API access via CoAP 199
14.4.3 CoAP configuration - Quick start guide 202

14.4.3.1 CoAP configuration via JSON 202
14.5 Syslog 204
14.5.1 Syslog configuration 204
14.5.2 Syslog configuration - Quick start guide 207
14.5.2.1 Syslog configuration via JSON 207
14.6 Network Time Protocol (NTP) 209
14.6.1 NTP configuration 209
14.6.2 NTP configuration - Quick start guide 211
14.6.2.1 NTP configuration via JSON 211

15 The integrated Web server 213

15.1 UNITRONIC® ACCESS MP08... variants 214
15.1.1 The Status page 214
15.1.2 The Ports page 215
15.1.2.1 IODD upload 215
15.1.3 The System page 217
15.1.3.1 License 218
15.1.3.2 Config upload/download 218
15.1.3.3 IODD 221
15.1.3.4 Restart device 222
15.1.3.5 Reset configuration to factory defaults 222
15.1.3.6 Firmware update 222
15.1.3.7 System diagnosis 223
15.1.4 The User page 224
15.2 UNITRONIC® ACCESS single-protocol PN0.../EIP0.../EC0.../MTCP0.../CCL0... variants 225
15.2.1 The System page 225

16 IODD 227

16.1 IO-Link Device parameters and ISDU requests 227
16.2 Web GUI functionality 228
16.2.1 Port Details page 228

16.2.2 Parameters page 230
16.2.3 IODD Management page 231

17 Technical data 233

17.1 General 234
17.2 EtherNet/IP protocol 235
17.3 Power supply of the module electronics/sensors 236
17.4 Power supply of the actuators 237
17.4.1 IO-Link Class A devices (U_L) 237
17.4.2 IO-Link Class A/B devices (U_{AUX}) 237
17.5 I/O ports Channel A (Pin 4) 238
17.5.1 Configured as digital input, Ch. A (Pin 4) 238
17.5.2 Configured as digital output, Ch. A (Pin 4) 239
17.5.3 Configured as IO-Link port in COM mode, Ch. A 241
17.6 I/O ports Channel B (Pin 2) 242
17.6.1 Configured as a digital input, Ch. B (Pin 2) 242
17.6.2 Configured as a digital output, Ch. B (Pin 2) 243
17.7 LEDs 245
17.8 Data transfer times 248

18 Accessories 250

1 About this manual

1.1 General information

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

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

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

U.I. Lapp GmbH

Schulze-Delitzsch-Straße 25
D-70565 Stuttgart
Germany

mailto:info@lapp.com
https://www.lapp.com

U.I. Lapp GmbH reserves the right to make technical changes or changes to this document 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.

1.3 Version information

Version	Created	Changes
1.0	10/2024	

Table 1: Overview of manual revisions

2 Safety instructions

2.1 Intended use

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

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

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

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



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

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

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

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

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

Information on which cables and accessories are permitted for the installation can be obtained from U.I. Lapp GmbH or is contained in this manual.

2.2 Qualified personnel

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

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

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

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

Only U.I. Lapp GmbH is permitted to make changes to the hardware or software of the products that go beyond the scope of this manual.



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



Attention: U.I. Lapp GmbH 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 and IO-Link)
BUI	Back-Up Inconsistency (EIP diagnostics)
CC	CC-Link IE Field
C/Q	I/O port pin 4 mode, IO-Link communication/switching signal
Ch. A	Channel A (Pin 4) of I/O port
Ch. B	Channel B (Pin 2) of I/O port
CIP	Common Industrial Protocol (media independent protocol)
CIP Safety™	Common Industrial Protocol for Safety applications, CIP Safety™ is a registered trademark of ODVA
Class A	IO-Link port specification (Class A)
Class B	IO-Link port specification (Class B)
CoAP	Constrained Application Protocol
CSP+	Control & Communication System Profile Plus
DAT	Device Acknowledgement Time
DCP	Discovery and Configuration Protocol
DevCom	Device Comunicating (EIP diagnostics)
DevErr	Device Error (EIP diagnostics)
DI	Digital Input
DIA	Diagnostic LED
DO	Digital Output
DIO	Digital Input/Output
DTO	Device Temperature Overrun (EIP diagnostics)
DTU	Device Temperature Underrun (EIP diagnostics)
DUT	Device under test

EIP	EtherNet/IP™ is a registered trademark of ODVA
ERP	Enterprise Resource Planning system
ETH	ETHERNET
FE	Functional Earth
FME	Force Mode Enabled (EIP diagnostics)
FS	Functional Safety
FSU	Fast Start-Up
GSDML	General Station Description Markup Language
High-B	High-Byte
HTTPS	Hyper Text Transfer Protocol Secure
ICE	IO-Link port COM Error (EIP diagnostics)
ICT	Invalid Cycle Time (EIP diagnostics)
IDE	IO-Link port Device Error (EIP diagnostics)
IDN	IO-Link port Device Notification (EIP diagnostics)
IDW	IO-Link port Device Warning (EIP diagnostics)
IIoT	Industrial Internet of Things
ILE	Input process data Length Error (EIP diagnostics)
IME	Internal Module Error (EIP diagnostics)
I/O	Input / Output
I/O port	X1 .. X8
I/O port pin 2	Channel B of I/O ports
I/O port pin 4 (C/Q)	Channel A of I/O ports
IODD	I/O Device Description
IOL or IO-L	IO-Link
I/Q	I/O port pin 2 mode, Digital Input/switching signal
ISDU	Indexed Service Data Unit
IVE	IO-Link port Validation Error (EIP diagnostics)
I&M	Identification & Maintenance
JSON	JavaScript Object Notation (platform independent data format)
L+	I/O port pin 1, sensor power supply

UNITRONIC® ACCESS 60	UNITRONIC® ACCESS variants with a width of 60mm
Little Endian	Data format with Low-B on first place (EtherNet/IP)
LLDP	Link Layer Discovery Protocol
Low-B	Low-Byte
LSB	Least Significant Bit
LVA	Low Voltage Actuator Supply (EIP diagnostics)
LVS	Low Voltage System/Sensor Supply (EIP diagnostics)
MIB	Management Information Base
MP	Multi-protocol: PROFINET + EtherNet/IP + EtherCAT® + Modbus TCP (+ CC-Link IE Field Basic)
MQTT	Message Queuing Telemetry Transport (open networking protocol)
MSB	Most Significant Bit
M12	Metric thread according to DIN 13-1 with 12 mm diameter
NTP	Network Time Protocol
OFDT	One Fault Delay Time
OLE	Output process data Length Error (EIP diagnostics)
OPC UA	Open Platform Communications Unified Architecture (platform independent, service-oriented architecture)
PFH	Probability of dangerous Failure per Hour [h -1]
PD	Process Data
PDCT	Port and Device Configuration Tool
PLC	Programmable Logic Controller
PN	PROFINET
PWR	Power
Qualifier	Validity on a process value. Valid = "1"
REST	REpresentational State Transfer
RFC	Request for Comments
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)
SFRT	Safety Function Response Time
SIO mode	Standard Input Output mode
SLMP	Seamless Message Protocol
SNMP	Simple Network Management Protocol
SP	Single Protocol (PROFINET, EtherNet/IP, EtherCAT®, Modbus TCP or CC-Link IE Field Basic)
SPE	Startup Parameterization Error (EIP diagnostics)
T-B	Test Channel B
T-A	Test Channel A
U_{AUX}	$U_{Auxiliary}$, supply voltage for the load circuit (Actuator supply on Class B ports of Class A/B IO-Link Master)
UDP	User Datagram Protocol
UDT	User-Defined Data Types
UINT8	Byte in PLC (IB, QB)
UINT16	Unsigned integer with 16 bits or word in PLC (IW, QW)
U_L	U_{Load}, supply voltage for the load circuit (Actuator supply on Class A IO-Link-Master)
UL	Underwriters Laboratories Inc. (certification company)
UTC	Coordinated Universal Time (Temps Universel Coordonné)
WCDT	Worst Case Delay Time

Table 2: Designations and synonyms

4 System description

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

4.1 About UNITRONIC® ACCESS

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

4.2 Device variants

The following variants are available in the UNITRONIC® ACCESS Multi-protocol and the UNITRONIC® ACCESS single-protocol family:

Article number	Product designation	Description	I/O port functionality
381166717	MP08IOLA08DIO	UNITRONIC® ACCESS M12-60 mm, IO-Link Master Multi-protocol (PN, EIP, EC, MB, CC) Security	8 x IO-Link Class A
381166712	PN08IOLA08DIO	UNITRONIC® ACCESS single-protocol M12-60 mm, IO-Link Master PROFINET	8 x IO-Link Class A
381166713	EIP08IOLA08DIO	UNITRONIC® ACCESS single-protocol M12-60 mm, IO-Link Master EtherNet/IP	8 x IO-Link Class A
381166714	EC08IOLA08DIO	UNITRONIC® ACCESS single-protocol M12-60 mm, IO-Link Master EtherCAT®	8 x IO-Link Class A
381166715	MTCP08IOLA08DIO	UNITRONIC® ACCESS single-protocol M12-60 mm, IO-Link Master Modbus TCP	8 x IO-Link Class A
381166716	CCL08IOLA08DIO	UNITRONIC® ACCESS single-protocol M12-60 mm, IO-Link Master CC-Link IE Field Basic	8 x IO-Link Class A

Table 3: Overview of UNITRONIC® ACCESS Multi-protocol and UNITRONIC® ACCESS single-protocol variants

4.3 I/O port overview

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

UNITRONIC® ACCESS Class A IO-Link ports

Device variant	Port	Pin 1 U_S	Pin 4 / Ch. A (C/Q)				Pin 2 / Ch. B (I/Q)	
MP08...	**Info:**	–	Class A	Type 1	Supply by U_S [1)]	Supply by U_L [2)]	Type 1	Supply by U_L [2)]
	X8:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)
	X7:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)
	X6:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)
	X5:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)
	X4:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)
	X3:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)
	X2:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)
	X1:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)

Table 4: Port configuration of MP08... variants

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

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

UNITRONIC® ACCESS single-protocol Class A IO-Link ports

Device variant	Port	Pin 1 U_S	Pin 4 / Ch. A (C/Q)			Pin 2 / Ch. B (I/Q)
PN08IOLA.../ EIP08IOLA.../ EC08IOLA.../ MTCP08IOLA.../ CCL08IOLA...	**Info:**	–	Class A	Type 1	Supply by U_S [1)]	Type 1
	X8:	Out (2 A)	IOL	DI	DO (0.5 A*)	DI
	X7:	Out (2 A)	IOL	DI	DO (0.5 A*)	DI
	X6:	Out (2 A)	IOL	DI	DO (0.5 A*)	DI
	X5:	Out (2 A)	IOL	DI	DO (0.5 A*)	DI
	X4:	Out (2 A)	IOL	DI	DO (0.5 A*)	DI
	X3:	Out (2 A)	IOL	DI	DO (0.5 A*)	DI
	X2:	Out (2 A)	IOL	DI	DO (0.5 A*)	DI
	X1:	Out (2 A)	IOL	DI	DO (0.5 A*)	DI

Table 5: Port configuration of PN08IOLA.../EIP08IOLA.../EC08IOLA.../ MTCP08IOLA.../CCL08IOLA... variants

[1)] With DO Switch Mode configured as "Push-Pull" (see description in the configuration chapters).

* For **UL applications**: Max. 0.25 A DO.

5 Overview of product features

5.1 EtherNet/IP product features

Data connection

The connection option provided by UNITRONIC® ACCESS is the widely-used M12 connector with D-coding for the EtherNet/IP network.

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

Data transmission rates

Featuring a transmission rate of up to 10/100 MBit/s, the EtherNet/IP devices can handle both fast transmission of I/O data and transmission of larger volumes of data.

EtherNet/IP Adapter Device

The UNITRONIC® ACCESS and UNITRONIC® ACCESS single-protocol IO-Link Master variants support the EtherNet/IP protocol. This allows the transmission of time sensitive process data between network components in real-time communication.

ODVA CIP specification V3.27

The UNITRONIC® ACCESS and UNITRONIC® ACCESS single-protocol IO-Link Master variants comply with ODVA CIP specification V3.27.

Integrated switch

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

DHCP/BOOTP

The supported Dynamic Host Configuration Protocol (DHCP) and the Bootstrap Protocol (BOOTP) provide mechanisms for automatic obtaining of an IP address from a server managing the devices.

Device Level Ring

The additionally implemented Device Level Ring (DLR) enables the design of a highly available network infrastructure of up to 50 DLR ring nodes. If a connection is interrupted, the UNITRONIC® ACCESS devices immediately switch to an alternative ring segment and thus ensure interruption-free operation. These DLR ring nodes are “beacon-based” according to the EtherNet/IP specification.

SNMP

The SNMPv1 protocol handles network component monitoring and communication between the master and device.

Diagnostic data

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

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

The EDS offers the option of configuring and parameterizing the I/O ports on the master devices.

5.2 I/O port features

IO-Link specification.

UNITRONIC® ACCESS is ready for IO-Link specification v1.1.3.

8 x IO-Link Master ports

Depending on the device variant, the device has 4 IO-Link Class A ports, 4 IO-Link Class A ports and 4 IO-Link Class B ports, or 8 IO-Link Class A ports with an additional digital input and optional output on pin 2 of the I/O port. For detailed information see chapter I/O port overview on page 21.



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.

IO-Link port connections

The IO-Link port connection option provided by the device series is the 5-pin M12 connector (Pin 5 not used at IO-Link Class A ports).

Validation & Backup

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

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

IO-Link Device parameterization

IO-Link Device parameterization in EtherNet/IP via vendor specific IO-Link Device parameter object class and Read/Write ISDU services.

LED

You can see the status of a port by the color of the matching LEDs and their flash pattern. For details on the meanings of the LED colors, please see section LEDs on page 245.

5.3 Integrated Web server

Network parameter display

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

Displaying diagnostics

View diagnostics via the integrated Web server.

User management

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

IO-Link Device parameters

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

5.4 Security features

Firmware signature

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

Syslog

The UNITRONIC® ACCESS multi-protocol variants support the traceability of messages centrally managed and logged via Syslog.

User manager

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

Default user settings:

User: admin

Password: private

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

5.5 Other features

Interface protection

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

For more details, see section Port assignments on page 33.

Failsafe

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

Industrial Internet of Things

UNITRONIC® ACCESS 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 UNITRONIC® ACCESS 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 UNITRONIC® ACCESS multi-protocol variants

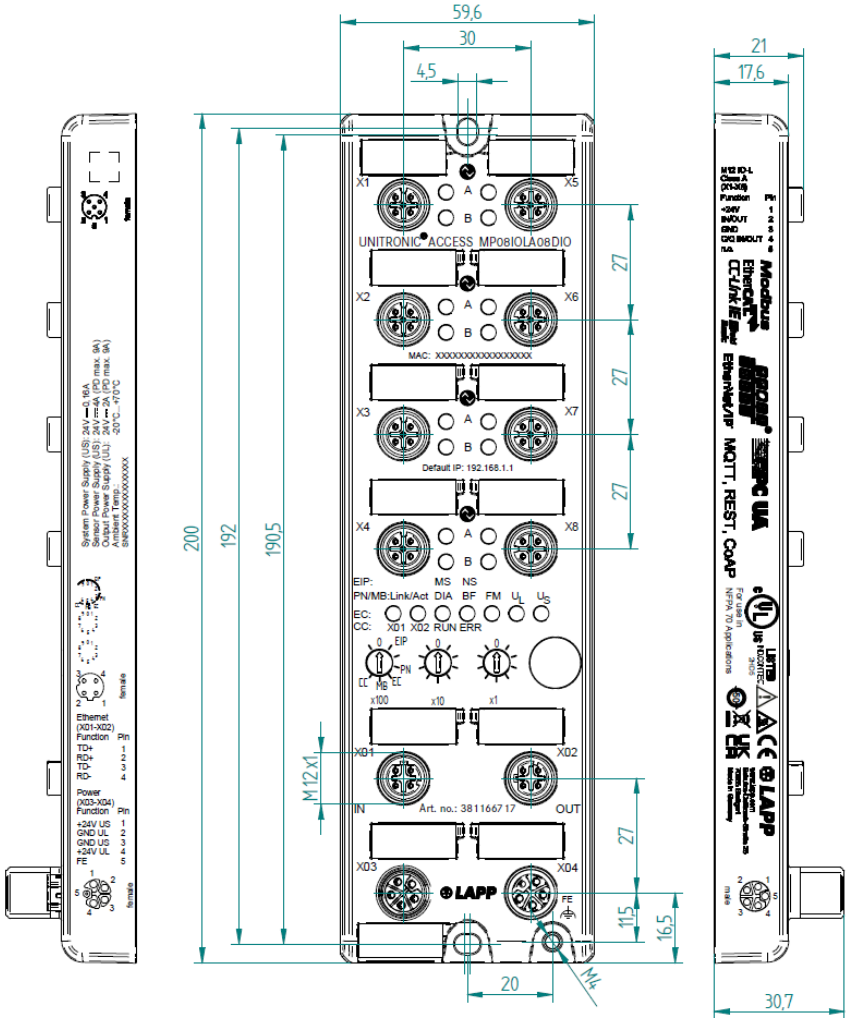


Figure 1: MP08IOLA08DIO

6.2.2 UNITRONIC® ACCESS single-protocol variants with EtherNet/IP

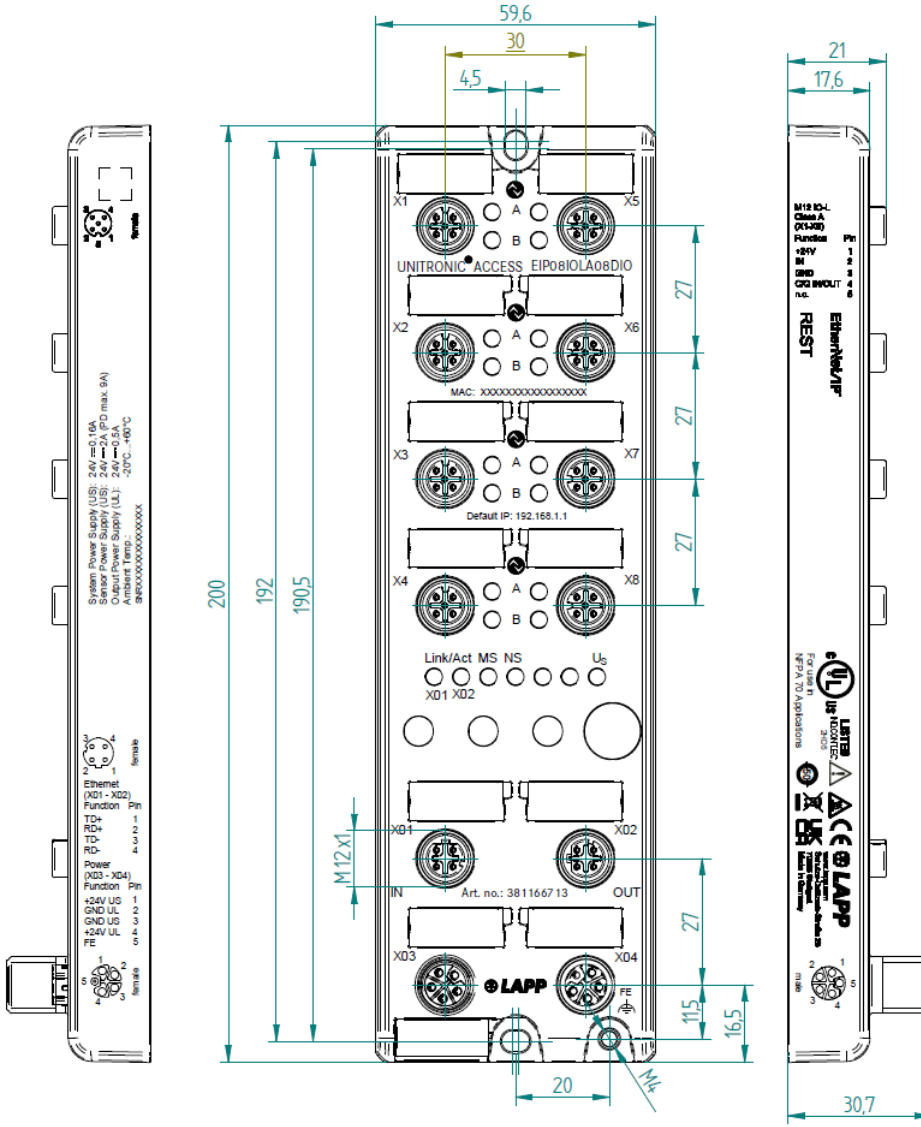


Figure 2: EIP08IOLA08DIO

6.2.3 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 device variants.

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

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

6.3 Port assignments

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

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

Color coding: green

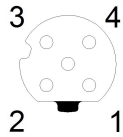


Figure 3: Schematic drawing, ports X01, X02

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 6: 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

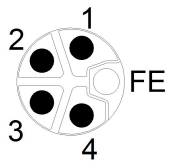


Figure 4: Schematic diagram of the M12 L-coding (connector X03 for Power In)

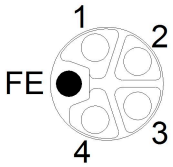


Figure 5: Schematic diagram of the M12 L-coding (socket X04 for Power Out)

6.3.2.1 IO-Link Master with Class A ports

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

Table 7: Power supply with M12-Power Class A

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

[1] U_L and U_S ground connected in device

6.3.2.2 IO-Link Master with Class A/B ports

Power supply	Pin	Signal	Function
Mixed IO-Link (Class A/B) I/O ports	1	U_S (+24 V)	Sensor/system power supply
	2	GND_U_{AUX}	Ground/reference potential U_{AUX} (electrically **isolated** to GND_U_S internally in device)
	3	GND_U_S	Ground/reference potential U_S
	4	U_{AUX} (+24 V)	Auxiliary supply (electrically **isolated** to U_S internally in device)
	5	FE	Functional ground

Table 8: Power supply with M12-Power Class A/B

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

6.3.3 I/O ports as M12 sockets

Color coding: black

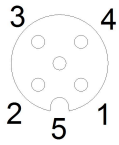


Figure 6: Schematic drawing I/O port as M12 socket IO-Link

6.3.3.1 IO-Link ports (Class A)

MP08...	Pin	Signal	Function
IO-Link Class A, ports X1 .. X8	1	+24 V	power supply +24 V
	2	IN/OUT	Ch. B: Digital input or digital output
	3	GND	Ground/reference potential
	4	C/Q	Ch. A: IO-Link data communication, digital input or digital output
	5	n.c.	not connected

PN08IOLA.../ EIP08IOLA.../ EC08IOLA.../ MTCP08IOLA.../ CCL08IOLA...	Pin	Signal	Function
IO-Link Class A, ports X1 .. X8	1	+24 V	power supply +24 V
	2	IN	Ch. B: Digital input
	3	GND	Ground/reference potential
	4	C/Q	Ch. A: IO-Link data communication, digital input or digital output
	5	n.c.	not connected

Table 9: I/O ports as IO-Link Class A

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

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

7 Starting operation

7.1 EDS file

An EDS file describes the EtherNet/IP device and can be installed in the engineering tool for the configuration of the UNITRONIC® ACCESS device. Each of the UNITRONIC® ACCESS variants requires its own EDS file. The file can be downloaded from the product pages on our online catalog: https://lapp.com

On request, the EDS file is also sent to you by the support team.

The EDS files are grouped together in an archive file named **EDS-V3.xx.x-LAPP-UNITRONIC-ACCESS-...08DIO-yyyymmdd.eds**.

yyyymmdd stands for the date on which the file was issued.

Download this file and unpack it.

Install the EDS file for the respective device variant by using the hardware or network configuration tool of your controller manufacturer.

In Rockwell Automation Studio 5000®, install the files with the *EDS Hardware Installation Tool*.

The UNITRONIC® ACCESS Multi-protocol and UNITRONIC® ACCESS single-protocol variants are then available in the hardware catalog as *Communications Adapter*.

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.

7.3 State on delivery

EtherNet/IP parameters in state on delivery or after a factory reset:

Network mode:	DHCP
Static IP address:	192.168.1.XXX (XXX = rotary switch position or last stored data)
Subnet mask:	255.255.255.0
Gateway address	0.0.0.0
Device designations:	MP08IOLA08DIO EIP08IOLA08DIO
Vendor code:	21
Product type:	12 (Communications Adapter)

7.4 Setting network parameters

There are multiple ways to configure the network parameters. By default, DHCP is enabled and the network parameters are requested by DHCP requests to a server. If you want to request the network parameters with BOOTP requests, you must activate the BOOTP function through the Web interface or the TCP/IP interface object (CIP Class ID 0xF5, attribute 3 (0x03)). It is also possible to set static network parameters via this CIP object.

7.4.1 IP address for UNITRONIC® ACCESS variants

The UNITRONIC® ACCESS multi-protocol variants support IP address configuration via the three rotary encoding switches on the front of the device (see chapter Setting the rotary encoding switches). The network parameters are also settable via the Web interface or the IIoT protocols.

7.4.2 IP address for UNITRONIC® ACCESS single-protocol variants

The UNITRONIC® ACCESS single-protocol variants cannot be configured via rotary encoding switches. If your network does not provide a DHCP server, you can set a static IP address via the NetIdent protocol.

7.5 Setting the rotary encoding switches



Caution: Risk of device damage due to corrupt device memory

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

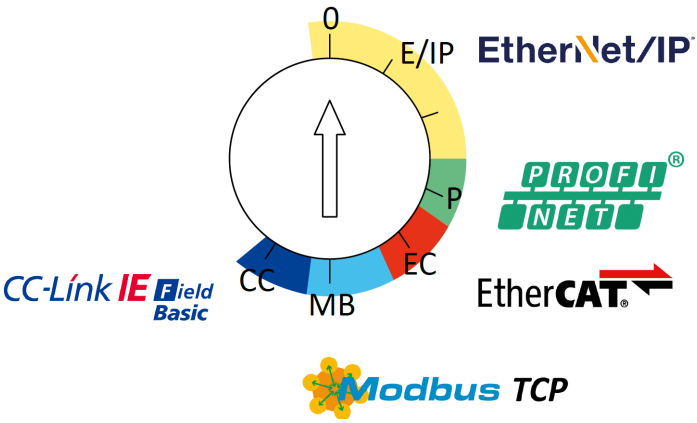
After selecting a protocol followed by a restart of the device, the new protocol is initialized. This can take up to 15 seconds. During this time the device is not usable and the LED indicators are out of function. When the protocol change is complete, the LED indicators return to normal operation and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

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

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

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

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



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

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

Table 10: 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 45.

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

7.5.1 EtherNet/IP selection and IP configuration via rotary encoding switches

The EtherNet/IP protocol can be selected by the first rotary encoding switch (x100) with a value between 0 – 2.

Use all three rotary encoding switches on the front of the device to set the last octet of the static IP address. The first three octets of the IP address are set by default to 192.168.1.

Each rotary encoding switch in the EtherNet/IP setting is assigned to one decimal digit, so that you can configure a number between 0 – 299. During start-up, the position of the rotary encoding switches is typically read within one time cycle.

For example, the rotary encoding switch setting 2 (x100), 1 (x10) and 0 (x1) is interpreted by default as the IP address 192.168.1.210.

Rotary encoding switch setting	Function
000 (state on delivery, default setting)	On delivery, the DHCP function is enabled. The network parameters are requested by DHCP requests to a server. If you want to request the network parameters with BOOTP requests, you must activate the BOOTP function through the Web server or the TCP/IP interface object (CIP Class ID 0xF5, attribute 3 (0x03)). The network parameters are not saved automatically, but the integrated Web server can be used to save them.
000 (network parameters already saved)	The network parameters last saved are used (IP address, subnet mask, gateway address, DHCP on/off, BOOTP on/off).
001 ... 254	The last 3 digits of the saved or preset IP address are overwritten by the setting of the rotary encoding switches. DHCP or BOOTP are disabled if necessary, and the device will start up with a static IP address.
255 ... 298	The network parameters are requested through DHCP or BOOTP but are not saved.
299	The factory default setting of the IP address (192.168.001.001) is used.
979	The device performs a reset to the factory settings. The network parameters are also reset to the default values. Communication is not possible in this operation mode.

Table 11: Setting options of the rotary encoding switches for EtherNet/IP

7.5.2 Factory reset

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

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

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

	x100	x10	x1
Factory Reset	9	7	9

Follow the steps from section Setting the rotary encoding switches on page 41 again to select a new protocol.

For performing a factory reset via software configuration, see chapter OPC UA configuration on page 172 and the configuration section.

8 Configuration EtherNet/IP

The devices support *Implicit Messaging* and *Explicit Messaging* for the EthetNet/IP communication. I/O process data is transferred cyclically via the assembly object connection using *Implicit Messaging*.

Non-critical low priority data, configuration settings and diagnostic data can be exchanged via acyclic messages using *Explicit Messaging*. The exchange is done via EtherNet/IP and vendor specific object classes. For more details on object classes, see chapter CIP object classes on page 110.

8.1 Assembly types

The UNITRONIC® ACCESS devices support three different assembly types which are defined as follows:

Assembly ID	Assembly Name	Size	Payload
130	Output Connection Point Assembly	0 .. 260 Byte	Consuming Data Image
131	Input Connection Point Assembly	0 .. 446 Byte	Producing Data Image
145	Configuration Assembly	0 or 400 Byte	Module Configuration Data
146	Configuration Assembly	0 or 300 Byte	Module Configuration Data (Omron)
147	Configuration Assembly	0 or 210 Byte	Module Configuration Data (Min)

The *Consuming Data Image* and the *Producing Data Image* have dynamic sizes which depend on the complete input and output data size of all connected IO-Link devices and on additional input status information. The general input and output process data sizes of each connection can be configured in the engineering tool. Each IO-Link Device process data size can be configured by the *Module Configuration Data*.

The contents of the *Consuming Data Image* and the *Producing Data Image* are specified in chapter Process data assignment on page 81.

The *Module Configuration Data* is defined in chapter Configuration parameters on page 51.

8.2 Connections

The UNITRONIC® ACCESS devices support four different connection types which are defined as follows:

Connection name	Connection type	Output connection point assembly	Output data size	Input connection point assembly	Input data size	Configuration assembly	Configuration data size
IO-Link (Exclusive Owner)	Exclusive Owner	130	0..260 Byte	131	0..446 Byte	145	0 or 400 Byte
IO-Link (Listen Only)	Listen Only	192	0	131	0..446 Byte	n/a	0 Byte
IO-Link Omron (Exclusive Owner)	Exclusive Owner	130	0..260 Byte	131	0..446 Byte	146	0 or 300 Byte
IO-Link Min (Exclusive Owner)	Exclusive Owner	130	0..260 Byte	131	0..446 Byte	147	0 or 210 Byte

The dynamic data sizes depend on the complete input and output data size of all connected IO-Link devices and additional input status information. The general input and output process data sizes of each connection can be configured in the engineering tool. Each IO-Link device process data size can be configured by the *Module Configuration Data*.

Some engineering tools require the immediate configuration of the connection parameters. For the configuration use the parameters listed in the following chapters.

8.2.1 IO-Link parameters (Exclusive Owner)

Connection properties	
Connection name	IO-Link (Exclusive Owner)
Application type	Exclusive Owner
Trigger mode	Cyclic
RPI	min. 1 ms

Connection parameters (O->T)	
Real time transfer format	32 Bit Run/Idle Header
Connection type	POINT2POINT
Assembly ID	130
Data size	0..260 Byte
Data type	INT (2 Byte)

Connection parameters (T->O)	
Real time transfer format	Pure data and modeless
Connection type	MULTICAST, POINT2POINT
Assembly ID	131
Data size	0..446 Byte
Data type	INT (2 Byte)

8.2.2 IO-Link parameters (Listen Only)

Connection properties	
Connection name	IO-Link (Listen Only)
Application type	Listen Only
Trigger mode	Cyclic
RPI	min. 1 ms

Connection parameters (O->T)	
Real time transfer format	Heartbeat
Connection type	POINT2POINT
Assembly ID	192
Data size	0 Byte
Data type	INT (2 Byte)

Connection parameters (T->O)	
Real time transfer format	Pure data and modeless
Connection type	MULTICAST
Assembly ID	131
Data size	0..446 Byte
Data type	INT (2 Byte)

8.2.3 IO-Link parameters Omron (Exclusive Owner)

Connection properties	
Connection name	IO-Link Omron (Exclusive Owner)
Application type	Exclusive Owner
Trigger mode	Cyclic
RPI	min. 1 ms

Connection parameters (O->T)	
Real time transfer format	32 Bit Run/Idle Header
Connection type	POINT2POINT
Assembly ID	130
Data size	0..260 Byte
Data type	INT (2 Byte)

Connection parameters (T->O)	
Real time transfer format	Pure data and modeless
Connection type	MULTICAST, POINT2POINT
Assembly ID	131
Data size	0..446 Byte
Data type	INT (2 Byte)

8.2.4 IO-Link parameters Min (Exclusive Owner)

Connection properties	
Connection name	IO-Link Min (Exclusive Owner)
Application type	Exclusive Owner
Trigger mode	Cyclic
RPI	min. 1 ms

Connection parameters (O->T)	
Real time transfer format	32 Bit Run/Idle Header
Connection type	POINT2POINT
Assembly ID	130
Data size	0..260 Byte
Data type	INT (2 Byte)

Connection parameters (T->O)	
Real time transfer format	Pure data and modeless
Connection type	MULTICAST, POINT2POINT
Assembly ID	131
Data size	0..446 Byte
Data type	INT (2 Byte)

9 Configuration parameters

Parameters of the UNITRONIC® ACCESS device can be configured via the configuration assembly, CIP object classes, Web server or IIoT protocols. A configuration assembly is sent when an *Exclusive Owner* connection is established. They are optional in this assembly. However, when sending, all existing parameters will be overwritten by this data. Therefore, the content of the configuration assembly has the highest valence.

To avoid parameter overwriting by CIP object classes, Web server or IIoT protocols during operation, some lock parameters can be enabled in the PLC configuration respectively configuration assembly.

The following chapters represent different setting groups with its configuration parameters. They are ingredients of the configuration assembly and can be set via *Explicit Messaging* by the specified CIP object classes. The **default values** are highlighted.

9.1 General settings

Configuration parameter	Byte offset config. assembly			Data type	Valid values	CIP object class 0xA0, Instance 1
	145	146	147			
Force Mode Lock	1	0	–	SINT	**0: Disable** 1: Enable	Attribute 2
Web Interface Lock	2	1	–	SINT	**0: Disable** 1: Enable	Attribute 3
Report U_L/ U_{Aux} Supply Voltage Fault	4	3	–	SINT	0: Disable **1: Enable**	Attribute 5
Report DO Fault without U_L/U_{Aux}	5	4	–	SINT	0: Disable **1: Enable**	Attribute 6
CIP object configuration lock	24	5	–	SINT	**0: Disable** 1: Enable	Attribute 25
External configuration lock	25	6	–	SINT	0: Disable **1: Enable**	Attribute 26
IO Mapping Mode	31	7	–	SINT	**0: Default Assignment** 1: Byte Swap 2: LSB Ch.A - MSB Ch.B 3: LSB Ch.B - MSB Ch.A 4: Free I/O Mapping	Attribute 32
General Settings	–	–	4	SINT	-128 .. 127 **(0)**	–

9.1.1 Force mode lock

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



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

9.1.2 Web interface lock

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

9.1.3 Report U_L/U_{AUX} supply voltage fault

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

9.1.4 Report DO Fault without U_L/U_{Aux}

With this parameter you suppress the actuator diagnosis message that is sent if no U_L/U_{Aux} supply is connected while the output data of a digital channel is controlled.

9.1.5 CIP object configuration lock

When there is no *Exclusive Owner* connection established, all configuration parameters can be set by vendor specific CIP object classes. To exclude parameter changes the setting function of these objects can be blocked.

When the *CIP object* configuration lock is enabled, the vendor specific configuration parameters cannot be set via the CIP services. This relates also to the *CIP object* configuration lock itself. A reset of this parameter can be done by a configuration assembly when an *Exclusive Owner* connection is established.

9.1.6 External configuration lock

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

9.1.7 IO Mapping Mode

The UNITRONIC® ACCESS devices support 5 different I/O mapping modes for the *Digital Output Channel Control* and the *Input Channel Status*. Modes 0 to 3 are pre-defined bit mappings. Mode 4 is a free user defined mapping which can be used in conjunction with the I/O mapping of channel 1 .. 16 in the channel settings.

Default Assignment (Mode 0):

DO Ch. Control / DI Ch. Status	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0 (LSB)	X4B	X4A	X3B	X3A	X2B	X2A	X1B	X1A
Byte 1 (MSB)	X8B	X8A	X7B	X7A	X6B	X6A	X5B	X5A

Byte Swap (Mode 1):

DO Ch. Control / DI Ch. Status	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0 (LSB)	X8B	X8A	X7B	X7A	X6B	X6A	X5B	X5A
Byte 1 (MSB)	X4B	X4A	X3B	X3A	X2B	X2A	X1B	X1A

LSB Ch.A - MSB Ch.B (Mode 2):

DO Ch. Control / DI Ch. Status	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0 (LSB)	X8A	X7A	X6A	X5A	X4A	X3A	X2A	X1A
Byte 1 (MSB)	X8B	X7B	X6B	X5B	X4B	X3B	X2B	X1B

LSB Ch.B - MSB Ch.A (Mode 3):

DO Ch. Control / DI Ch. Status	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0 (LSB)	X8B	X7B	X6B	X5B	X4B	X3B	X2B	X1B
Byte 1 (MSB)	X8A	X7A	X6A	X5A	X4A	X3A	X2A	X1A

Free IO Mapping (Mode 4):

IO Mapping Channel 1 .. 16 is used (see chapter Channel settings on page 57).

9.1.8 General Settings (parameter)

This parameter is used in the *Min Configuration Assembly* to configure all general settings with a single parameter. It is an 8bit signed integer used as bit-field for the different settings.

Position	Setting	Value
Bit 0 (LSB)	Force Mode Lock	0: Disabled 1: Enabled
Bit 1	Web Interface Lock	0: Disabled 1: Enabled
Bit 2	Output Auto Restart	0: Enabled 1: Disabled
Bit 3	Report U_L/U_{Aux} Supply Voltage Fault	0: Enabled 1: Disabled
Bit 4	Report DO Fault without U_L/U_{Aux}	0: Enabled 1: Disabled
Bit 5	CIP object configuration lock	0: Disabled 1: Enabled
Bit 6	External configuration lock	0: Disabled 1: Enabled
Bit 7 (MSB)	Reserved	0

Table 12: Parameter General Settings - configuration options

See sub chapters under General settings on page 52 for a detailed description of those different settings.

9.2 Channel settings

Configuration parameter	Byte offset config. assembly			Data type	Valid values	CIP object class 0xA1, Instance 1 .. 16
	145	146	147			
IO Mapping (Ch1 .. 16)	32	8	–	SINT[16]	**0 .. 15**: Bit number of 16 channel process data 16: Inactive	Attribute 1
DO Surveillance Timeout (Ch1 .. 16)	48	–	–	INT[16]	0 .. 255 **(80)**	Attribute 2
DO Surveillance Timeout Omron (Ch1 .. 16)	–	24	–	USINT[16]	0 .. 255 **(80)**	–
DO Failsafe (Ch1 .. 16)	80	40	–	SINT[16]	**0: Set Low** 1: Set High 2: Hold Last	Attribute 3
DO Restart Mode (Ch1 .. 16)	96	56	–	SINT[16]	0: Disable **1: Enable**	Attribute 4
DO Switch Mode (Not available for UNITRONIC® ACCESS single-protocol IO-Link Master variants)	112	72	–	SINT[16]	0: Push-Pull (U_S, 0.5 A) 1: High-Side (U_L, 0.5 A) 2: High-Side (U_L, 1.0 A) 3: High-Side (U_L, 1.5 A) 4: High-Side (U_L, 2.0 A) **5: High-Side (U_L, 2.0 A max)**	Attribute 5
DI Logic (Ch1 .. 16)	128	88	–	SINT[16]	**0: Normally Open** 1: Normally Close	Attribute 6
DI Filter (Ch1 .. 16)	144	104	–	SINT[16]	0: Disabled 1: 1 ms 2: 2 ms **3: 3 ms** 4: 6 ms 5: 10 ms 6: 15 ms	Attribute 7
DI Latch	160	–	–	INT	-32768 .. 32767 **(0)**	–

Configuration parameter	Byte offset config. assembly			Data type	Valid values	CIP object class 0xA1, Instance 1 .. 16
	145	146	147			
DI Extension	176	–	–	DINT	-2147483648 .. 2147483647 **(0)**	–
Channel Mode (Ch1 .. 16)	192	120	–	SINT[16]	0: Inactive 1: Digital Output 2: Digital Input 3: IO-Link 4: Auxiliary Power **The supported Channel Mode and the default value depend on the device variant.**	Attribute 10
Port mode for Channel A (Pin 4)	–	–	0	INT	-32768 .. 32767 **(0)**	–
Port mode for Channel B (Pin 2)	–	–	2	INT	-32768 .. 32767 **(0)**	–

Assignment of channels:

Channel 1	Port X1.ChA	CIP object instance 1
Channel 2	Port X1.ChB	CIP object instance 2
[…]	[…]	[…]
Channel 15	Port X8.ChA	CIP object instance 15
Channel 16	Port X8.ChB	CIP object instance 16

9.2.1 IO Mapping (Ch1 .. 16)

These configuration parameters can be used to set a user defined IO mapping. It is valid for the input and output data direction. Duplicated assignment are not allowed. In case of an inconsistent mapping, the complete assembly configuration is rejected with an error code.

To use these parameters, it is required to configure the IO mapping mode of the *General settings* to *Free IO Mapping (Mode 4)*. The default value for each parameter is its own channel number.

9.2.2 DO Surveillance Timeout (Ch1 .. 16)

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

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

9.2.3 DO Failsafe (Ch1 .. 16)

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

Set Low:

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

Set High:

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

Hold Last:

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

9.2.4 DO Restart Mode (Ch1 .. 16)

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

If the *DO Restart Mode* for this channel is enabled, the output will automatically be turned on again after a fix time delay for checking if the overload or short circuit condition is still active. When it is active, the channel is switched off again.

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

9.2.5 DO Switch Mode (Ch1 .. 16)

Only applicable for the following device variants:

- MP08IOLA08DIO

With this parameter you can configure the current limitations for the digital outputs by selecting a DO Switch Mode. You can choose between two different output switch modes:

- Push-Pull (U_S,0.5 A):

  If a channel is set to "Push-Pull", the output will be switched active to high or low. In low state, the output can be a current sink. The digital output is supplied by U_S with a maximum current of *0.5 A*. This option is not available for the channel B of any port.

- High-Side (U_L, 0.5 A..2.0 A max):

  If a channel is set to "High-Side", the output will be switched active to high but not to low. In low state, the output has a high impedance. The digital output is supplied by U_L or U_{Aux}, depending on the device variant, and has a selectable current limit. This means that the actuator channel error diagnosis is reported when this limit is exceeded. If the you set the level to *2.0 A Max.*, the current limitation is not active and the maximum output current is available.

Refer to chapter I/O port overview on page 21 to get the available voltage supply for the digital outputs of every UNITRONIC® ACCESS variant.

9.2.6 DI Logic (Ch1 .. 16)

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

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

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

9.2.7 DI Filter (Ch1 .. 16)

A filter time for every digital input channel can be configured by these parameters. When there is no need for a filter it can be disabled.

9.2.8 DI Latch

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

This parameter is used to configure the DI Latch for all 16 channels with a single parameter.

It is a 16bit signed integer used as a bit-field, with each channel occupying 1 bit. Starting with Port 1 Channel A at bit 0 (LSB), Port 1 Channel B at bit 1, Port 2 Channel A at bit 2, …, Port 8 Channel B at bit 15 (MSB).

For each channel, the DI Latch is encoded as follows:

- 0: Disabled
- 1: Enabled

For example, to enable the DI Latch for Port 1 Channel B and Port 6 Channel A while keep it disabled for all other channels, the corresponding bit-field would be '0000010000000010', so the parameter would have to be configured to '1026'.

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

In detail:

• If the DI channel is in a low state and a high input of any duration is detected, the channel will report a high input indefinitely, regardless of the actual physical input. In other words, the latch will be triggered.

• If the channel is in a high state, a transition is required at first to 'low' and then to 'high' in order for the latch to be triggered.

The latch is triggered only when a rising edge occurs on the channel.

The latch can be reset by setting the appropriate output channel to 'true'. The input status will not be changed, it will only deactivate the latch.

This setting only works for channels that have been set to 'digital input mode'.

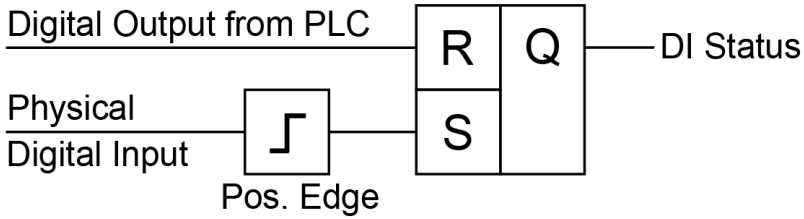


Figure 7: Input latch

Default: 0 (Disabled)

9.2.9 DI Extension

This parameter is used to configure the DI Extension for all 16 Channels with a single parameter.

It is a 32bit signed integer used as a bit-field, with each channel occupying 2 bits. Starting with Port 1 Channel A at bit 0 (LSB), Port 1 Channel B at bit 2, Port 2 Channel A at bit 4, …, Port 8 Channel B at bit 30 (MSB).

For each channel, the DI Extension is encoded as follows:

- (00)0: Disabled
- (01)1: 8 ms
- (10)2: 16 ms
- (11)3: 64 ms

For example, to configure the DI Extension for Port 1 Channel B to '6 ms' and for Port 6 Channel A to '255 ms' while keeping it disabled for all other channels, the corresponding bit-field would be '00000000000000000000111100000000000000000000000000000010000000', so the parameter would have to be configured to '16492674416768'.

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

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

Example:

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

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

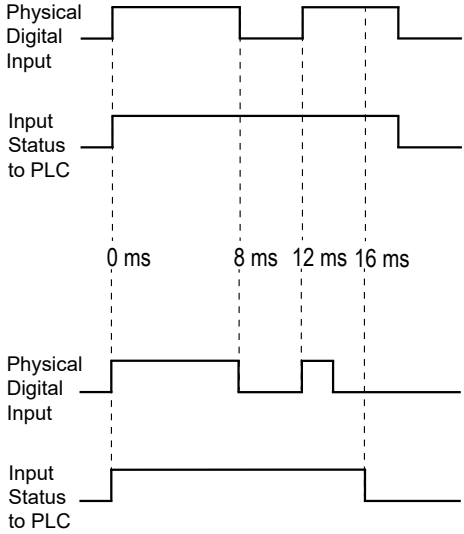


Figure 8: DI Extension

Available values: Off; 8 ms; 16 ms; 64 ms

Default: Off

9.2.10 Channel Mode (Ch1 .. 16)

The operation mode of every channel can be configured by these parameters. The usability of this setting depends on the hardware variant and can be figured out in the description (e.g. for a 8 IO-Link Class A Master, an IO-Link mode can only be configured for channel A and not for channel B).

Inactive:

This mode should be selected when the channel is not in use.



Attention: If channel A of a port is set to inactive, the corresponding channel B is also set to inactive regardless of its configuration. In this case, the entire port is therefore deactivated.

Digital Output:

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

Digital Input:

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

IO-Link:

In this mode, the channel tries to establish a communication with an IO-Link Device. IO-Link process data can be exchanged via a communication link between the IO-Link Master and the IO-Link Device. The size of the IO-Link input and output data as well as the port mode depend on the IO-Link port settings.



Attention: Not all channels support this configuration.

Auxiliary Power:

IO-Link Master variants with Class B ports provide an auxiliary voltage output on channel B. When *Auxiliary Power* is configured, the voltage output cannot be controlled for this channel and is fed from the U_{AUX} supply input. IO-Link Class A ports do not support this configuration.

Attention: Not all channels support this configuration.

9.2.11 Port mode for Channel A (Pin 4)

This parameter is used in the *Min Configuration Assembly* to configure the port modes for all 8 A Channels (Pin 4) with a single parameter.

It is a 16bit signed integer used as bit-field, with each channel occupying 2 bits. Starting with port 1 at bit 0 (LSB), port 2 at bit 2, …, port 8 at bit 14 (MSB).

For each channel, the *Port Mode* is encoded as follows:

(00)0: IO-Link (mapped to Digital Input for channels not supporting IO-Link)

(01)1: Digital Output

(10)2: Digital Input

(11)3: Inactive

See chapter Channel Mode (Ch1 .. 16) on page 64 for a detailed description of those different channel modes.

For example, to configure ports 1 .. 4 Channel A to "IO-Link" and ports 5 .. 8 Channel A to "Digital Output"', the corresponding bit-field would be '0101010100000000', so the parameter would have to be configured to '21760'.

9.2.12 Port mode for Channel B (Pin 2)

This parameter is used in the *Min Configuration Assembly* to configure the port modes for all 8 B Channels (Pin 2) with a single parameter.

It is a 16bit signed integer used as bit-field, with each channel occupying 2 bits. Starting with port 1 at bit 0 (LSB), port 2 at bit 2, …, port 8 at bit 14 (MSB).

For each channel, the *Port Mode* is encoded as follows:

(00)0: Digital Input (mapped to Auxiliary Power for IO-Link Type B channels not supporting Digital Input)

(01)1: Digital Output

(10)2: Auxiliary Power (only IO-Link Type B)

(11)3: Inactive

See chapter Channel Mode (Ch1 .. 16) on page 64 for a detailed description of those different channel modes.

For example, to configure ports 1 .. 4 Channel B to "Digital Output" and ports 5 .. 8 Channel B to "Inactive"', the corresponding bit-field would be '1111111101010101', so the parameter would have to be configured to '-171'.

9.3 IO-Link diagnosis settings

Configuration parameter	Byte offset config. assembly	Data type	Valid values	CIP object class 0xA2, Instance 1
IO-Link Master Diagnosis	208	SINT	0: Disable **1: Enable**	Attribute 1
IO-Link Device Error	209	SINT	0: Disable **1: Enable**	Attribute 2
IO-Link Device Warning	210	SINT	0: Disable **1: Enable**	Attribute 3
IO-Link Device Notification	211	SINT	0: Disable **1: Enable**	Attribute 4
IO-Link Device Diagnosis Port 1 .. 8	212 .. 219	SINT[8]	0: Disable **1: Enable**	Attribute 5 .. 12

9.3.1 IO-Link Master Diagnosis

If this parameter is enabled, the *IO-Link Master Diagnosis* is transferred to the IO-Link diagnoses of the input process data. If configured, additional diagnoses and information are transferred in the *IO-Link Extended Status* and in the *IO-Link events*.

If this parameter is disabled, no *IO-Link Master Diagnosis* is reported.

9.3.2 IO-Link Device Error

If this parameter is enabled, the *IO-Link Device Errors* are transferred in the IO-Link diagnoses of the input process data. If configured, additional diagnoses and information are transferred in the *IO-Link Extended Status* and the *IO-Link events*.

If this parameter is disabled, no *IO-Link Device Error* is reported.

9.3.3 IO-Link Device Warning

If this parameter is enabled, the *IO-Link Device Warnings* are transferred in the IO-Link diagnoses of the input process data. If configured, additional diagnoses and information are transferred in the *IO-Link Extended Status* and the *IO-Link events*.

If this parameter is disabled, no *IO-Link Device Warning* is reported.

9.3.4 IO-Link Device Notification

If this parameter is enabled, the *IO-Link Device Notifications* are transferred in the IO-Link diagnoses of the input process data. If configured, additional diagnoses and information are transferred in the *IO-Link Extended Status* and the *IO-Link events*.

If this parameter is disabled, no *IO-Link Device Notification* is reported.

9.3.5 IO-Link Device Diagnosis Port 1 .. 8

If this parameter is enabled for an IO-Link port, the respective diagnoses are transferred in the IO-Link diagnoses of the input process data. If configured, additional diagnoses and information are transferred in the *IO-Link Extended Status* and the *IO-Link events*.

If this parameter is disabled for an IO-Link port, no respective diagnosis is reported.

9.4 IO-Link Port 1 .. 8 settings

Configuration parameter	Byte offset config. assembly			Data type	Valid values	CIP object class 0xA3, Instance 1 .. 8
	145	146	147			
Output Data Size	224, 246, 268, 290, 312, 334, 356, 378	148, 167. 186, 205, 224, 243, 262, 281	–	SINT	0: No data 1: 2 Byte 2: 4 Byte 3: 8 Byte 4: 16 Byte **5: 32 Byte**	Attribute 1
Input Data Size	225, 247, 269, 291, 313, 335, 357, 379	149, 168, 187, 206, 225, 244, 263, 282	–	SINT	0: No data 1: 2 Byte 2: 4 Byte 3: 8 Byte 4: 16 Byte **5: 32 Byte**	Attribute 2
Input Data Extension	226, 248, 270, 292, 314, 336, 358, 380	150, 169, 188, 207, 226, 245, 264, 283	–	SINT	0: No Data 1: Extended Status 2: Events **3: Extended Status + Events**	Attribute 3
Output Data Swapping Mode	227, 249, 271, 293, 315, 337, 359, 381	151, 170, 189, 208, 227, 246, 265, 284	–	SINT	**0: Raw IO-Link Data** 1 .. 16: 1 .. 16 WORD 17 .. 24: 1 .. 8 DWORD	Attribute 4
Output Data Swapping Offset	228, 250, 272, 294, 316, 338, 360, 382	152, 171, 190, 209, 228, 247, 266, 285	–	SINT	0 .. 30 Byte **("0")**	Attribute 5

Configuration parameter	Byte offset config. assembly			Data type	Valid values	CIP object class 0xA3, Instance 1 .. 8
	145	146	147			
Input Data Swapping Mode	229, 251, 273, 295, 317, 339, 361, 383	153, 172, 191, 210, 229, 248, 267, 286	–	SINT	**0: Raw IO-Link Data** 1 .. 16: 1 .. 16 WORD 17 .. 24: 1 .. 8 DWORD	Attribute 6
Input Data Swapping Offset	230, 252, 274, 296, 318, 340, 362, 384	154, 173, 192, 211, 230, 249, 268, 287	–	SINT	0 .. 30 Byte **("0")**	Attribute 7
IOL Failsafe	231, 253, 275, 297, 319, 341, 363, 385	155, 174, 193, 212, 231, 250, 269, 288	–	SINT	**0: Set Low** 1: Set High 2: Hold Last 3: Replacement Value (transferred via IO-Link Failsafe Parameter Object) 4: IO-Link Master Command	Attribute 8
Port Mode	232, 254, 276, 298, 320, 342, 364, 386	156, 175, 194, 213, 232, 251, 270, 289	–	SINT	0: Deactivated 1: Manual (with validation and backup config) **2: Autostart (no validation and backup config)**	Attribute 9
IO-Link Mode	–	–	5	SINT	-128 .. 127 **(0)**	–

Configuration parameter	Byte offset config. assembly			Data type	Valid values	CIP object class 0xA3, Instance 1 .. 8
	145	146	147			
Validation and Backup	233, 255, 277, 299, 321, 343, 365, 387	157, 176, 195, 214, 233, 252, 271, 290	7, 32, 57, 82, 107, 132, 157, 182	SINT	**0: No device check and clear (no data storage)** 1: Type compatible V1.0 device (no data storage) 2: Type compatible V1.1 device (no data storage) 3: Type compatible V1.1 device with Backup + Restore (download + upload) 4 Type compatible V1.1 device with Restore (download master to device)	Attribute 10
Vendor ID	234, 256, 278, 300, 322, 344, 366, 388	158, 177, 196, 215, 234, 253, 272, 291	8, 33, 58, 83, 108, 133, 158, 183	DINT	0 .. 65535 **("0")**	Attribute 11
Device ID	238, 260, 282, 304, 326, 348, 370, 392	162, 181, 200, 219, 238, 257, 276, 295	12, 37, 62, 87, 112, 137, 162, 187	DINT	0 .. 16777215 **("0")**	Attribute 12

Configuration parameter	Byte offset config. assembly			Data type	Valid values	CIP object class 0xA3, Instance 1 .. 8
	145	146	147			
Cycle Time	242, 264, 286, 308, 330, 352, 374, 396	–	6, 31, 56, 81, 106, 131, 156, 181	SINT	**0: As fast as possible** 1: 1.6 ms 2: 3.2 ms 3: 4.8 ms 4: 8.0 ms 5: 20.8 ms 6: 40.0 ms 7: 80.0 ms 8: 120.0 ms	Attribute 13

Assignment of the IO-Link ports:

IO-Link port 1	Port X1.ChA	CIP object instance 1
[…]	[…]	[…]
IO-Link port 8	Port X8.ChA	CIP object instance 8

The number of IO-Link ports depends on the IO-Link Master variant. IO-Link Masters with less than 8 IO-Link ports only provide configuration parameters for their own count. Unused configuration data bytes are sent as "zero bytes" inside the configuration assembly.

Configuration parameters of an IO-Link port are only taken into account by the application when the corresponding Channel Mode of the Channel Settings is set to *IO-Link*.

9.4.1 Output Data Size

The *Output Data Size* of the respective IO-Link device can be configured by this parameter. There can be up to 32 Bytes of IO-Link output data per port.

The *Output Data Size* of every IO-Link device has influence on the total *Output Data Size* of the connection. It has to be taken into account that all IO-Link output data fits into the total size.

This parameter is only settable when no connection is active.

9.4.2 Input Data Size

The *Input Data Size* of the respective IO-Link device can be configured by this parameter. There can be up to 32 Bytes of IO-Link input data.

The *Input Data Size* of every IO-Link device has influence on the total *Input Data Size* of the connection. It has to be taken into account that all IO-Link input data fits into the total size.

This parameter is only settable when no connection is active.

9.4.3 Input Data Extension

The *Input Data Extension* can be selected to extend each IO-Link input data with extended status information and/or IO-Link events.

The *Input Data Extension* of every IO-Link device has influence on the total input data size of the connection. It has to be taken into account that all IO-Link output data including the extension fits into the total size.

This parameter is only settable when no connection is active.

9.4.4 Output Data Swapping Mode

The byte order of IO-Link is big endian which is not compatible to EtherNet/IP's little endian format. For setting output data in the correct format, the parameters *Output Data Swapping Mode* and *Output Data Swapping Offset* support the user. There can be selected up to 16 words or up to 8 double words for converting the output data.

Raw IO-Link Data:

No byte swap

Data type WORD:

Data byte order: Byte 0, Byte 1

Order after Swap: Byte 1, Byte 0

Data type DWORD:

Data byte order: Byte 0, Byte 1, Byte 2, Byte 3

Order after Swap: Byte 3, Byte 2, Byte 1, Byte 0

9.4.5 Output Data Swapping Offset

The *Output Data Swapping Offset* describes the start point in the process data for using the configured *Output Data Swapping Mode*. Both parameters are dependent on the configured output data size.

9.4.6 Input Data Swapping Mode

The byte order of IO-Link is big endian which is not compatible to EtherNet/IP's little endian format. For receiving input data in the correct format, the parameters *Input Data Swapping Mode* and *Input Data Swapping Offset* support the user. There can be selected up to 16 words or up to 8 double words for converting the input data.

Raw IO-Link Data:

No byte swap

Data type WORD:

Data byte order: Byte 0, Byte 1

Order after Swap: Byte 1, Byte 0

Data type DWORD:

Data byte order: Byte 0, Byte 1, Byte 2, Byte 3

Order after Swap: Byte 3, Byte 2, Byte 1, Byte 0

9.4.7 Input Data Swapping Offset

The *Input Data Swapping Offset* describes the start point in the process data for using the configured *Input Data Swapping Mode*. Both parameters are dependent on the configured input data size and the optional input data extension.

9.4.8 IOL Failsafe

The UNITRONIC® ACCESS devices support a failsafe function for the output data of the IO-Link channels. In case of an internal device error, the PLC is in STOP state and cannot provide valid process data, the connection

is interrupted or the communication is lost: The output data of the IO-Link channels is controlled by the configured failsafe values.

Set Low:

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

Set High:

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

Hold Last:

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

Replacement Value:

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

IO-Link Master Command:

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

9.4.9 Port Mode

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

Deactivated:

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

IO-Link Autostart:

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

IO-Link Manual:

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

9.4.10 IO-Link Mode

This parameter is used in the *Min Configuration Assembly* to configure the IO-Link Mode for all 8 ports with a single parameter.

It is an 8bit-signed integer used as bit-field. Starting with port 1 at bit 0 (LSB), port 2 at bit 1, …, port 8 at bit 7 (MSB).

For each channel, the *IO-Link Mode* is encoded as follows:

0: Auto

1: Manual

See chapter Port Mode on page 75 for a detailed description of those different IO-Link Modes.

For example, to configure ports 1 .. 4 to "IO-Link Auto" and ports 5 .. 8 to "IO-Link Manual"', the corresponding bit-field would be '11110000', so the parameter would have to be configured to '-16'.

9.4.11 Validation and Backup

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

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

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

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

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

No device check (no data storage):

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

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

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

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

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

Type compatible V1.1 device with Backup + Restore

(upload + download):

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

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

- Backup (Device to Master):

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

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

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

- Restore (Master to Device):

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

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

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

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

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

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

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

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

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

9.4.12 Vendor ID

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

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

9.4.13 Device ID

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

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

9.4.14 Cycle Time

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

The precondition for using *Cycle Time* is that you configure *Port Mode* to "IO-Link Manual".

As fast as possible:

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

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

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

10 Process data assignment

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

The following chapters describe the data images for the consuming and producing data direction which are assigned to the output and input assemblies.

10.1 Consuming data image (output)

Output data frame	Digital output channel control	Reserved (e.g. feature control)	IO-Link output data
Consuming data size	2 Byte, INT	2 Byte, INT	0..256 Byte, INT

The complete *Output data frame* has a variable size of up to 260 Bytes. In general, a 4 Byte Run/Idle Header precedes, resulting in up to 264 Bytes in total.

The following chapters describe the bit assignment.

10.1.1 Digital output channel control

Digital output channel control	Bit	7	6	5	4	3	2	1	0
Channel number (default mapping)	Byte 0	8	7	6	5	4	3	2	1
	Byte 1	16	15	14	13	12	11	10	9

The control values are effective if the respective channels are configured as outputs and the *Digital Output Control* is set to *DO Channel Control*.

10.1.2 IO-Link output data

IO-Link output data	IO-Link port 1 control	IO-Link port 2 control	IO-Link port 3 control	IO-Link port 4 control	IO-Link port 5 control	IO-Link port 6 control	IO-Link port 7 control	IO-Link port 8 control
IO-Link port output size	0 Byte 2 Byte 4 Byte 8 Byte 16 Byte 32 Byte	0 Byte 2 Byte 4 Byte 8 Byte 16 Byte 32 Byte	0 Byte 2 Byte 4 Byte 8 Byte 16 Byte 32 Byte	0 Byte 2 Byte 4 Byte 8 Byte 16 Byte 32 Byte	0 Byte 2 Byte 4 Byte 8 Byte 16 Byte 32 Byte	0 Byte 2 Byte 4 Byte 8 Byte 16 Byte 32 Byte	0 Byte 2 Byte 4 Byte 8 Byte 16 Byte 32 Byte	0 Byte 2 Byte 4 Byte 8 Byte 16 Byte 32 Byte

The IO-Link port output size is independent of the configured Channel Mode. It is always considered in the IO-Link output data and therefore the offsets do **not** need to be calculated again by the user in case of a channel mode reconfiguration. Every IO-Link port can be set to its required size. The control

data is transferred to the device. However, the content depends on the IO-Link *Output Data Swapping Mode* and *Output Data Swapping Offset*.

If there is no IO-Link port configured, the *Consuming data image* has no IO-Link output data.

10.2 Producing data image (input)

Input data frame	Digital input channel status	General diagnostics	Sensor diagnostics	Actuator/ U_{Aux} diagnostics	IO-Link diagnostics	IO-Link input data
Producing data size	2 Byte, INT	2 Byte, INT	2 Byte, INT	2 Byte, INT	0 Byte 6 Byte, INT	0..432 Byte, INT

The complete *Input data frame* has a variable size of up to 446 Bytes.

The following chapters describe the bit assignment.

10.2.1 Digital input channel status

Digital input channel status	Bit	7	6	5	4	3	2	1	0
Channel number (default mapping)	Byte 0	8	7	6	5	4	3	2	1
	Byte 1	16	15	14	13	12	11	10	9

Each status value is effective if the channel is configured as Input.

10.2.2 General diagnostics

General diagnostics	Bit	7	6	5	4	3	2	1	0
General Bit	Byte 0	IME	FME	DTO	DTU	SCA	SCS	LVA	LVS
	Byte 1	0	0	0	0	IDN	IDW	IDE	IVE

LVS Low Voltage System/Sensor Supply

LVA	Low Voltage Actuator Supply
SCS	Short Circuit Sensor
SCA	Short Circuit Actuator/U_L/U_{Aux}
DTU	Device Temperature Underrun
DTO	Device Temperature Overrun
FME	Force Mode Enabled
IME	Internal Module Error
IVE	IO-Link Validation Error (collective error)
IDE	IO-Link Device Error (collective error)
IDW	IO-Link Device Warning (collective error)
IDN	IO-Link Device Notification (collective error)
0	Reserved

10.2.3 Sensor diagnostics

Sensor diagnostics	Bit	7	6	5	4	3	2	1	0
Port number	Bvte 0	X8	X7	X6	X5	X4	X3	X2	X1
	Byte 1	0	0	0	0	0	0	0	0

X1 .. 8	Sensor Short Circuit on Port X1 .. X8
0	Reserved

10.2.4 Actuator/U_L/U_{Aux} diagnostics

Actuator/U_{Aux} diagnostics	Bit	7	6	5	4	3	2	1	0
Channel number (fix)	Byte 0	8	7	6	5	4	3	2	1
	Byte 1	16	15	14	13	12	11	10	9

1 .. 16 Actuator/U_L/U_{Aux} channel error on channel 1 .. 16

10.2.5 IO-Link diagnostics

IO-Link diagnostics	Bit	7	6	5	4	3	2	1	0
General Bit	Byte 0	ICE8	ICE7	ICE6	ICE5	ICE4	ICE3	ICE2	ICE1
	Byte 1	0	0	0	0	0	0	0	0
	Byte 2	IVE8	IVE7	IVE6	IVE5	IVE4	IVE3	IVE2	IVE1
	Byte 3	IDE8	IDE7	IDE6	IDE5	IDE4	IDE3	IDE2	IDE1
	Byte 4	IDW8	IDW7	IDW6	IDW5	IDW4	IDW3	IDW2	IDW1
	Byte 5	IDN8	IDN7	IDN6	IDN5	IDN4	IDN3	IDN2	IDN1

ICE1..8 IO-Link Port COM Error (device missing, broken wire, short circuit)

IVE1..8 IO-Link Port Validation Error

IDE1..8 IO-Link Port Device Error

IDW1..8 IO-Link Port Device Warning

IDN1..8 IO-Link Port Device Notification

0 Reserved

If there is no IO-Link port configured, the input data image doesn't show IO-Link diagnostics.

10.2.6 IO-Link input data

IO-Link input data	IO-Link port 1				[…]	IO-Link port 8			
	Status	PQI	Extended status	Events	[…]	Status	PQI	Extended status	Events
IO-Link port input size	0 Byte 2 Byte 4 Byte 8 Byte 16 Byte 32 Byte	2 Byte	0 Byte 8 Byte	0 Byte 12 Byte	[…]	0 Byte 2 Byte 4 Byte 8 Byte 16 Byte 32 Byte	2 Byte	0 Byte 8 Byte	0 Byte 12 Byte

The IO-Link port input size is independent of the configured Channel Mode. It is always considered in the IO-Link input data and therefore the offsets do **not** need to be calculated again by the user in case of a channel mode reconfiguration. Every IO-Link port can be set to its required size. The device input data is mapped to the **Status** field and the content depends on the IO-Link *Input Data Swapping Mode* and *Input Data Swapping Offset*.

An IO-Link port can be configured via the Channel Mode. The PQI provides some IO-Link information, is always available and is independent of the Status size. The Extended Status and Events can be enabled by the IO-Link port configuration.

Port Qualifier Information (PQI):

PQI (Port Qualifier Information)	Bit	7	6	5	4	3	2	1	0
General Bit	Byte 0	PQ	DevErr	DevCom	PortActive	SubstDev	NewPar	0	0
	Byte 1	0	0	0	0	0	0	0	0

NewPar Update of Device parameter detected

SubstDev Substitute device detected (different SerialNumber)

PortActive Port activated

DevCom Device detected and is in PREOPERATE or OPERATE state

DevErr Error/warning assigned to Device or Port occurred

PQ Valid I/O Process Data from Device

0 Reserved

Extended status:

IO-Link Extended status	Bit	7	6	5	4	3	2	1	0
Extended diagnostics	Byte 0	0	0	0	ICT	BUI	SPE	ILE	OLE
	Byte 1	0							
Vendor ID	Byte 2	Vendor ID (LSB)							
	Byte 3	Vendor ID (MSB)							
Device ID	Byte 4	Device ID (LSB)							
	Byte 5	Device ID							
	Byte 6	Device ID (MSB)							
	Byte 7	0							

OLE Output process data length error (device mismatch)

ILE Input process data length error (device mismatch)

SPE	Startup parameterization error (direct parameter error)
BUI	Backup inconsistency (parameter storage error)
ICT	Invalid Cycle Time
0	Reserved

Events:

IO-Link events	Bit	7	6	5	4	3	2	1	0
Event Qualifier 1	Byte 0	Mode		Type		0	0	Instance	
	Byte 1	0	0	0	0	0	0	0	0
Event Code 1	Byte 2	Event Code							
	Byte 3								
Event Qualifier 2	Byte 4	Mode		Type		0	0	Instance	
	Byte 5	0	0	0	0	0	0	0	0
Event Code 2	Byte 6	Event Code							
	Byte 7								
Event Qualifier 3	Byte 8	Mode		Type		0	0	Instance	
	Byte 9	0	0	0	0	0	0	0	0
Event Code 3	Byte 10	Event Code							
	Byte 11								

Instance	Unknown ("0"), Reserved (Physical Layer PL ("1"), Data Link Layer DL ("2"), Application Layer AL ("3")), Application ("4")
Type	Notification ("1"), Warning ("2"), Error ("3")

Mode	Event single shot ("1"), Event disappears ("2"), Event appears ("3")
Event Code	Diagnostic code reported by the IO-Link device
0	Reserved

10.3 Sample applications

The connection and configuration parameters of the device with its variable data sizes provides you an individual approach for realizing your application. The size of each IO-Link port can be determined which also has influence on the process data offsets.

The following application samples describe the process data assignments for the input and output data including the byte offsets. When there is no need to configure the data sizes, use the first sample to get the default byte offsets for your application. When you decide to reduce the data sizes to set them for example to the required IO-Link data lengths or you when do not need the extended status, have a look at the second sample to understand how the data mapping works.

For Rockwell Automation/Allen Bradley PLC customer, it is recommended to use an Add-On Instruction in Studio 5000® as an interface to the process data as described in chapter Add-On Instruction (AOI) on page 100.

10.3.1 Process data images – default configuration

The default configuration of the IO-Link port input and output data sizes is set to the maximum sizes in the EDS files. This means the user gets all data of each IO-Link port. The following tables provide you an overview of the data structures and the byte offsets for input and output data:

Connection parameters

Output data size	260
Input data size	446

Byte offset	Output data
0	Digital output channel control (2 bytes)
2	Reserved (2 bytes)
4	IO-Link port1 data (control, 32 bytes)
36	IO-Link port2 data (control, 32 bytes)
68	IO-Link port3 data (control, 32 bytes)
100	IO-Link port4 data (control, 32 bytes)
132	IO-Link port5 data (control, 32 bytes)
164	IO-Link port6 data (control, 32 bytes)
196	IO-Link port7 data (control, 32 bytes)
228	IO-Link port8 data (control, 32 bytes)

Table 13: Default output process data

Byte offset	Input data
0	Digital input channel status (2 bytes)
2	General diagnostics (2 bytes)
4	Sensor diagnostics (2 bytes)
6	Actuator diagnostics (2 bytes)
8	IO-Link diagnostics (6 bytes)
14	IO-Link port1 data (status, 32 bytes)
46	IO-Link port1 PQI (2 bytes)
48	IO-Link port1 extended status (8 bytes)
56	IO-Link port1 events (12 bytes)
68	IO-Link port2 data (status, 32 bytes)
100	IO-Link port2 PQI (2 bytes)
102	IO-Link port2 extended status (8 bytes)
110	IO-Link port2 events (12 bytes)
122	IO-Link port3 data (status, 32 bytes)
154	IO-Link port3 PQI (2 bytes)
156	IO-Link port3 extended status (8 bytes)

Byte offset	Input data
164	IO-Link port3 events (12 bytes)
176	IO-Link port4 data (status, 32 bytes)
208	IO-Link port4 PQI (2 bytes)
210	IO-Link port4 extended status (8 bytes)
218	IO-Link port4 events (12 bytes)
230	IO-Link port5 data (status, 32 bytes)
262	IO-Link port5 PQI (2 bytes)
264	IO-Link port5 extended status (8 bytes)
272	IO-Link port5 events (12 bytes)
284	IO-Link port6 data (status, 32 bytes)
316	IO-Link port6 PQI (2 bytes)
318	IO-Link port6 extended status (8 bytes)
326	IO-Link port6 events (12 bytes)
338	IO-Link port7 data (status, 32 bytes)
370	IO-Link port7 PQI (2 bytes)
372	IO-Link port7 extended status (8 bytes)
380	IO-Link port7 events (12 bytes)
392	IO-Link port8 data (status, 32 bytes)
424	IO-Link port8 PQI (2 bytes)
426	IO-Link port8 extended status (8 bytes)
434	IO-Link port8 events (12 bytes)

Table 14: Default input process data

10.3.2 Process data images with modified data sizes

The IO-Link port input and output data sizes and the existence of the extended status can be modified by the configuration assembly. This means the user can decide about which data is mapped to the process data. The following configuration tables provide you a sample and an overview of possible data structures and the byte offsets for input and output data:

Connection parameters

Output data size	62
Input data size	66

IO-Link Port1

Output data size	2 Byte
Input data size	2 Byte
Input data extension	No Data

IO-Link Port2

Output data size	32 Byte
Input data size	0 Byte
Input data extension	Extended Status

IO-Link Port3

Output data size	16 Byte
Input data size	4 Byte
Input data extension	Extended Status + Events

IO-Link Port4

Output data size	8 Byte
Input data size	2 Byte
Input data extension	No Data

IO-Link Port5 .. 8

Output data size	0 Byte
Input data size	0 Byte
Input data extension	No Data

Byte offset	Output data	Input data
0	Digital output channel control (2 bytes)	Digital input channel status (2 bytes)
2	Reserved (2 bytes)	General diagnostics (2 bytes)
4	IO-Link port1 data (control, 2 bytes)	Sensor diagnostics (2 bytes)
6	IO-Link port2 data (control, 32 bytes)	Actuator diagnostics (2 bytes)
8		IO-Link diagnostics (6 bytes)
10		
12		
14		IO-Link port1 data (status, 2 bytes)
16		IO-Link port1 PQI (2 bytes)
18		IO-Link port2 PQI (2 bytes)
20		IO-Link port2 extended status (8 bytes)
22		
24		
26		
28		IO-Link port3 data (status, 4 bytes)
30		
32		IO-Link port3 PQI (2 bytes)
34		IO-Link port3 extended status (8 bytes)
36		
38	IO-Link port3 data (control, 16 bytes)	
40		
42		IO-Link port3 events (12 bytes)
44		
46		
48		
50		
52		
54	IO-Link port4 data (control, 8 bytes)	IO-Link port4 data (status, 2 bytes)
56		IO-Link port4 PQI (2 bytes)

Byte offset	Output data	Input data
58		IO-Link port5 PQI (2 bytes)
60		IO-Link port6 PQI (2 bytes)
62		IO-Link port7 PQI (2 bytes)
64		IO-Link port8 PQI (2 bytes)
66		

Table 15: Modified process data

11 Configuration and operation with Rockwell Automation Studio 5000®

The configuration and start-up of the devices described on the following pages refers to Rockwell Automation Studio 5000®, V30. If you are using an engineering tool from another provider, please consider the related documentation.

11.1 Basic commissioning

Perform the following working steps:

1. Create a new project in Studio 5000®.

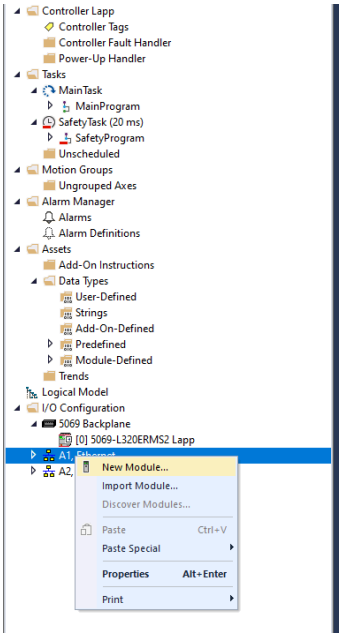
2. Select the correct controller.

3. When no integrated EtherNet/IP interface is available, add the proper communication interface to your backplane under **Controller Organizer** > **I/O-Configuration**.

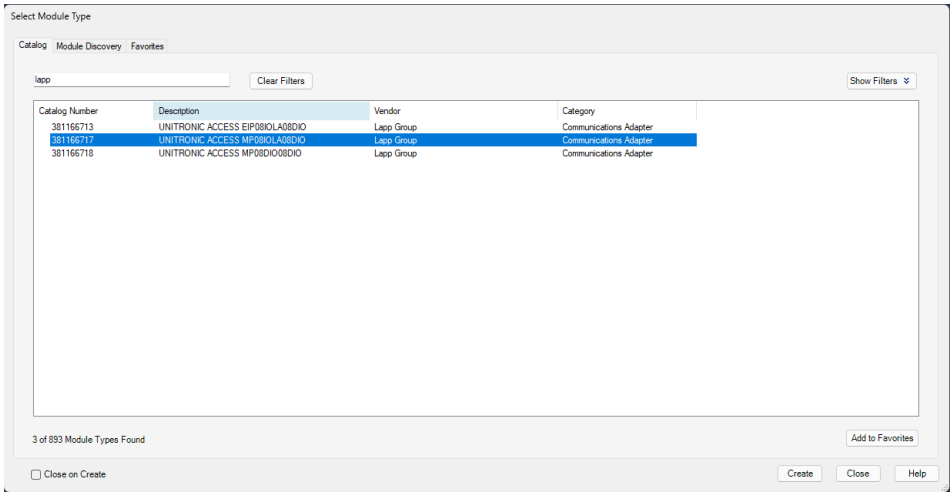
4. Set a communication path to enable the project download.

5. Install the EDS files of the UNITRONIC® ACCESS devices in Studio 5000® with the EDS hardware installation tool.

6. Go to **Controller Organizer** > **I/O-Configuration** and right-click the Ethernet interface.

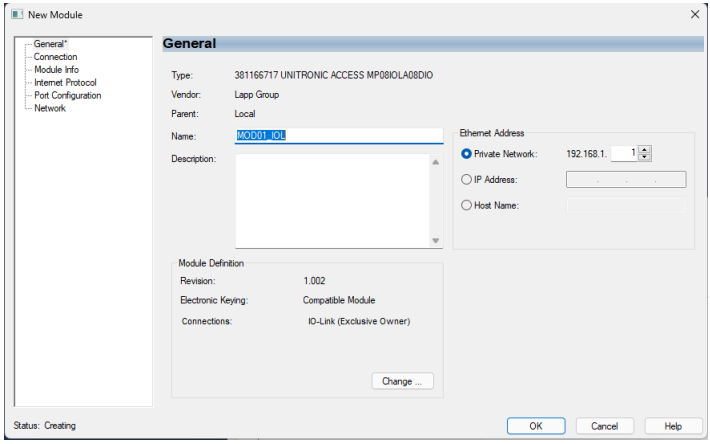


7. Select **New Module** in the menu. The following selection window opens:



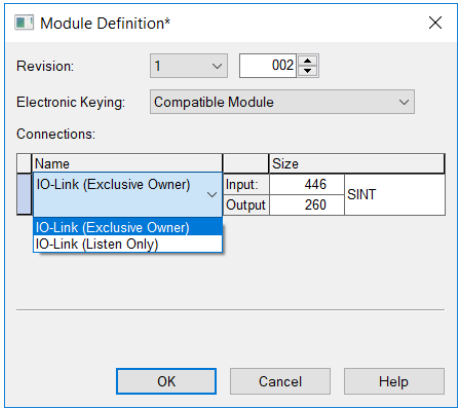
8. Use the **Module Type Vendor Filter** on the right side to display all installed devices of U.I. Lapp GmbH.

9. Select the device you wish to add and click on **Create**.



10. Enter a name for the device and set the chosen IP address. In this example, the name is **MOD01_IOL** and the IP address is **192.168.1.1**.

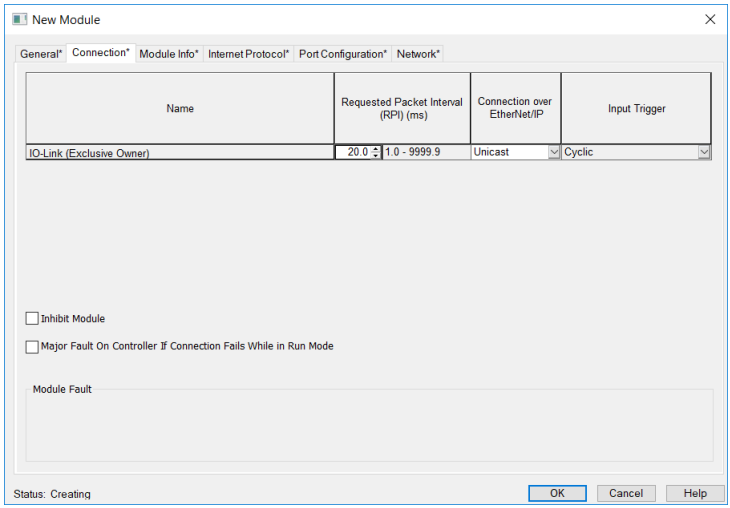
11. Click on **Change** in order to change the settings for the device revision, electronic keying and connection type.



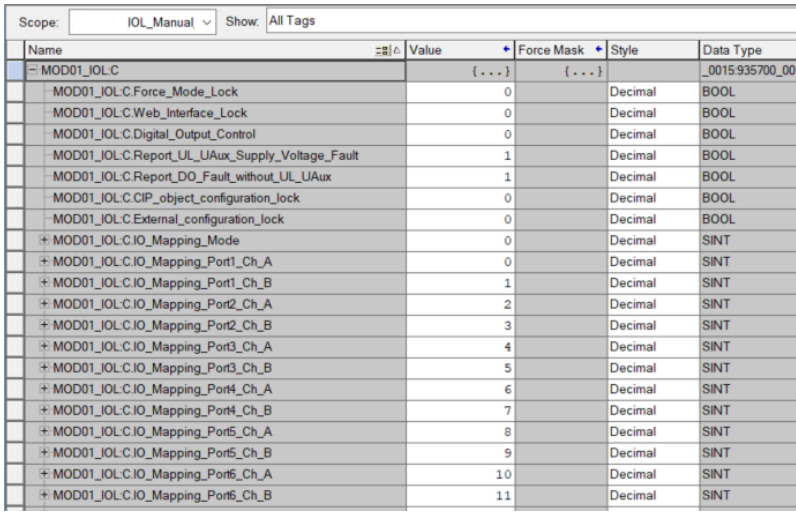
12. Select the connection type and configure the total sizes of the input and output process data. The sizes depend on the number of connected IO-Link

devices and their data lengths of both directions. Each device input and output data size must also be set later in the IO-Link port configuration. The selection of the data type refers to the type in which Studio 5000® maps the input and output data. The default data type is SINT. The INT type is selectable when each size is a multiple of 2. The DINT type is selectable when each size is a multiple of 4. Click on **OK**.

13. In the **Connection** folder of the **Module Properties**, you see the selected connection. This folder also lets you define the **Requested Packet Interval (RPI)** and the EtherNet/IP connection type. A value of 1 ms is the minimum for parameter RPI and the connection types *Unicast* or *Multicast* can be chosen. Apply the settings.



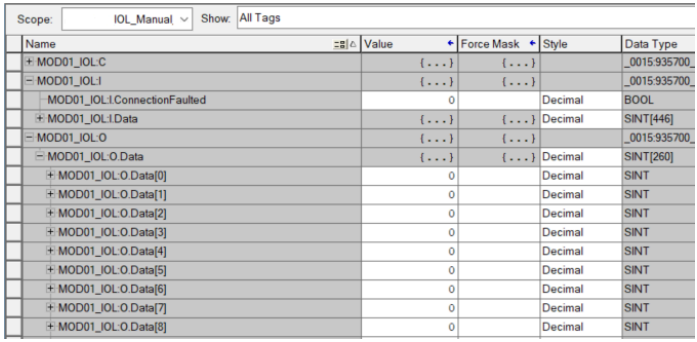
14. Move to **Controller-Tags** in **Controller Organizer**. The controller tags for the configuration parameters contain the name of the device, followed by a "**:C**". The configuration parameters can be set under **Value** and are described in chapter Configuration parameters on page 51.



Scope: IOL_Manual Show: All Tags

Name	Value	Force Mask	Style	Data Type
MOD01_IOL:C	{...}	{...}		_0015:935700_001_
MOD01_IOL:C.Force_Mode_Lock	0		Decimal	BOOL
MOD01_IOL:C.Web_Interface_Lock	0		Decimal	BOOL
MOD01_IOL:C.Digital_Output_Control	0		Decimal	BOOL
MOD01_IOL:C.Report_UL_UAux_Supply_Voltage_Fault	1		Decimal	BOOL
MOD01_IOL:C.Report_DO_Fault_without_UL_UAux	1		Decimal	BOOL
MOD01_IOL:C.CIP_object_configuration_lock	0		Decimal	BOOL
MOD01_IOL:C.External_configuration_lock	0		Decimal	BOOL
MOD01_IOL:C.IO_Mapping_Mode	0		Decimal	SINT
MOD01_IOL:C.IO_Mapping_Port1_Ch_A	0		Decimal	SINT
MOD01_IOL:C.IO_Mapping_Port1_Ch_B	1		Decimal	SINT
MOD01_IOL:C.IO_Mapping_Port2_Ch_A	2		Decimal	SINT
MOD01_IOL:C.IO_Mapping_Port2_Ch_B	3		Decimal	SINT
MOD01_IOL:C.IO_Mapping_Port3_Ch_A	4		Decimal	SINT
MOD01_IOL:C.IO_Mapping_Port3_Ch_B	5		Decimal	SINT
MOD01_IOL:C.IO_Mapping_Port4_Ch_A	6		Decimal	SINT
MOD01_IOL:C.IO_Mapping_Port4_Ch_B	7		Decimal	SINT
MOD01_IOL:C.IO_Mapping_Port5_Ch_A	8		Decimal	SINT
MOD01_IOL:C.IO_Mapping_Port5_Ch_B	9		Decimal	SINT
MOD01_IOL:C.IO_Mapping_Port6_Ch_A	10		Decimal	SINT
MOD01_IOL:C.IO_Mapping_Port6_Ch_B	11		Decimal	SINT

15. The tag of the input process data contain the name of the device, followed by a "**:I.Data**". The output process data has the same name followed by a "**:O.Data**". Both arrays show its configured data sizes. The content of them is described in chapter Process data assignment on page 81.



Scope: IOL_Manual Show: All Tags

Name	Value	Force Mask	Style	Data Type
MOD01_IOL:C	{...}	{...}		_0015:935700_
MOD01_IOL:I	{...}	{...}		_0015:935700_
MOD01_IOL:I.ConnectionFaulted	0		Decimal	BOOL
MOD01_IOL:I.Data	{...}	{...}	Decimal	SINT[446]
MOD01_IOL:O	{...}	{...}		_0015:935700_
MOD01_IOL:O.Data	{...}	{...}	Decimal	SINT[260]
MOD01_IOL:O.Data[0]	0		Decimal	SINT
MOD01_IOL:O.Data[1]	0		Decimal	SINT
MOD01_IOL:O.Data[2]	0		Decimal	SINT
MOD01_IOL:O.Data[3]	0		Decimal	SINT
MOD01_IOL:O.Data[4]	0		Decimal	SINT
MOD01_IOL:O.Data[5]	0		Decimal	SINT
MOD01_IOL:O.Data[6]	0		Decimal	SINT
MOD01_IOL:O.Data[7]	0		Decimal	SINT
MOD01_IOL:O.Data[8]	0		Decimal	SINT

16. When the configuration is completed, the parameters can be downloaded to the EtherNet/IP controller.

11.2 Add-On Instruction (AOI)

Rockwell Automation Studio 5000® provides the user a mechanism for the optimization and encapsulation of data and logic via an Add-On Instruction. This AOI can be added to a rung as any other pre-defined instruction in the controller and is useful for the preprocessing of the input and output data of a device.

With the help of User-Defined Data Types (UDT), the user gets a comprehensible interface with a clear naming and description for each field of the process data. The advantage is that it is no longer required to calculate byte offsets of the input and output data. Every field of the process data can be directly addressed via a unique name.

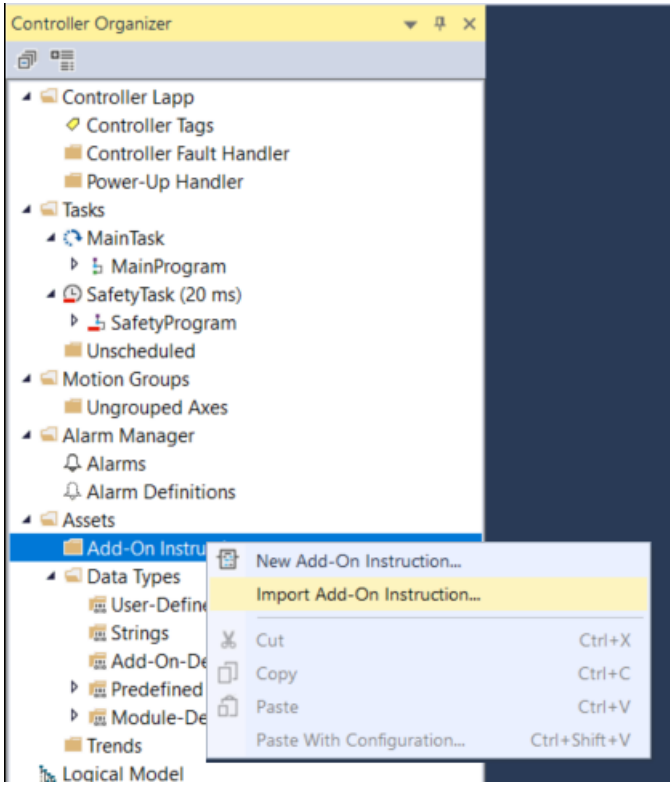
LAPP provides AOIs for customers which can be downloaded from the product pages on our online catalog: https://lapp.com.

On request, an AOI is also sent to you by the LAPP support team.

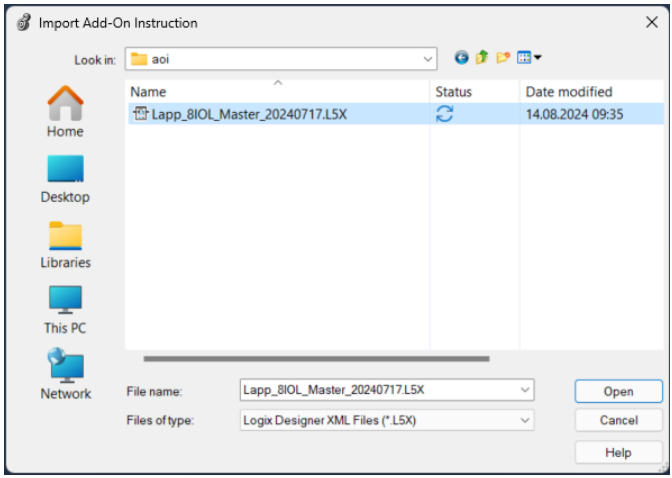
Perform the following working steps for using an AOI:

Attention: For version V36, the edits (steps 5, 6, 7 and 8) should be done directly in the LSX file before starting the import (step 1).

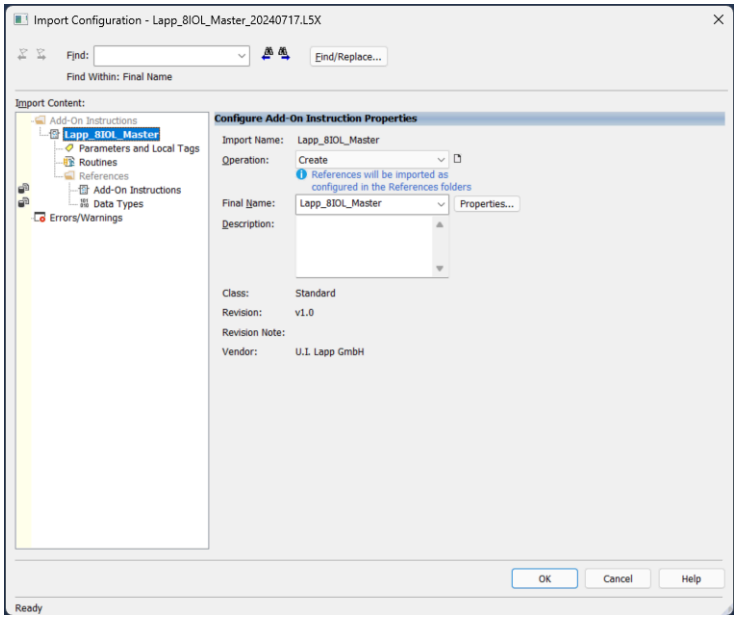
1. In your Studio 5000® project, navigate to **Controller Organizer**, right-click on **Add-On Instructions** and click **Import Add-On Instruction...**:


Controller Organizer
Controller Lapp
Controller Tags
Controller Fault Handler
Power-Up Handler
Tasks
MainTask
MainProgram
SafetyTask (20 ms)
SafetyProgram
Unscheduled
Motion Groups
Ungrouped Axes
Alarm Manager
Alarms
Alarm Definitions
Assets
Add-On Instru
Data Types
User-Defin
Strings
Add-On-De
Predefined
Module-De
Trends
Logical Model
New Add-On Instruction...
Import Add-On Instruction...
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Paste With Configuration... Ctrl+Shift+V

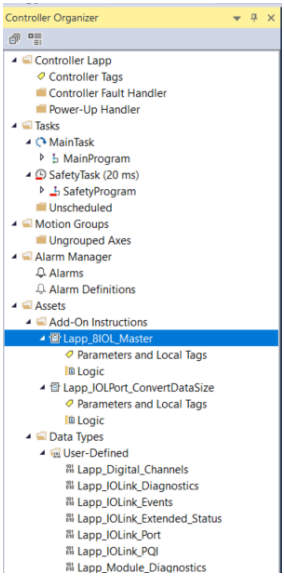
2. Open the *.L5X file:



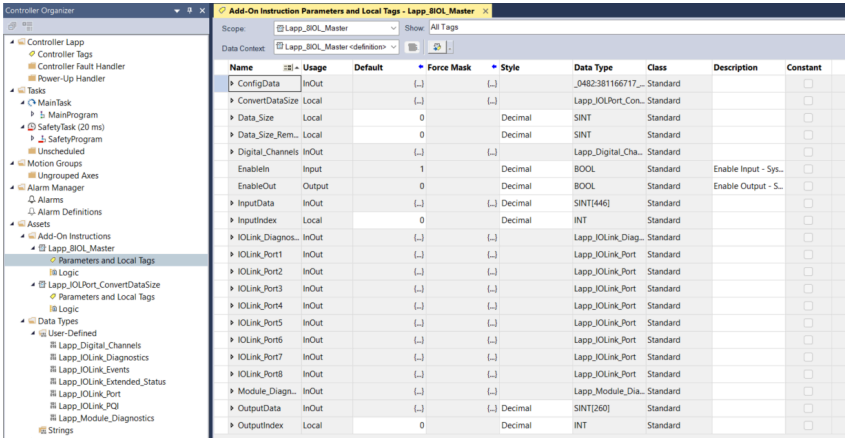
3. Click OK to create the AOI with all necessary User-Defined Data Types:



4. The imported components are now shown in the **Controller Organizer**:

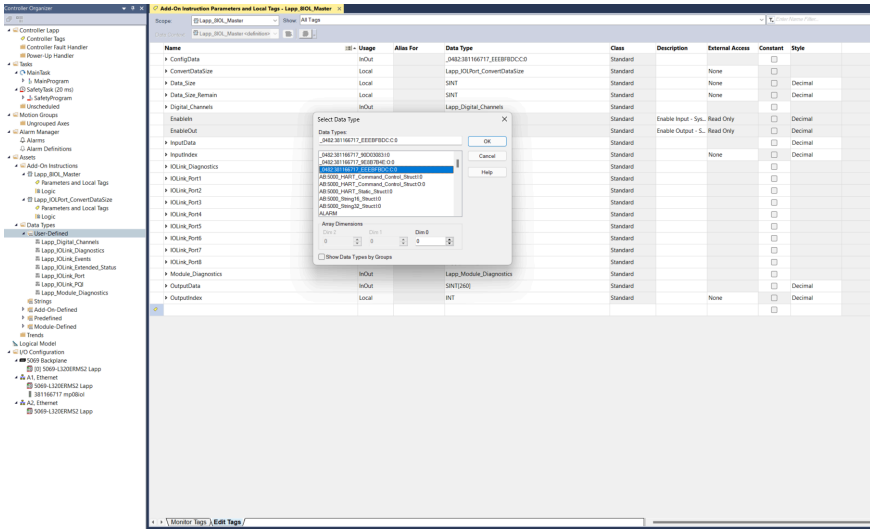


5. Check if an error is shown in the AOI tags (red circle with a white cross). This can happen for the configuration data when you have imported an AOI on your system for the first time:



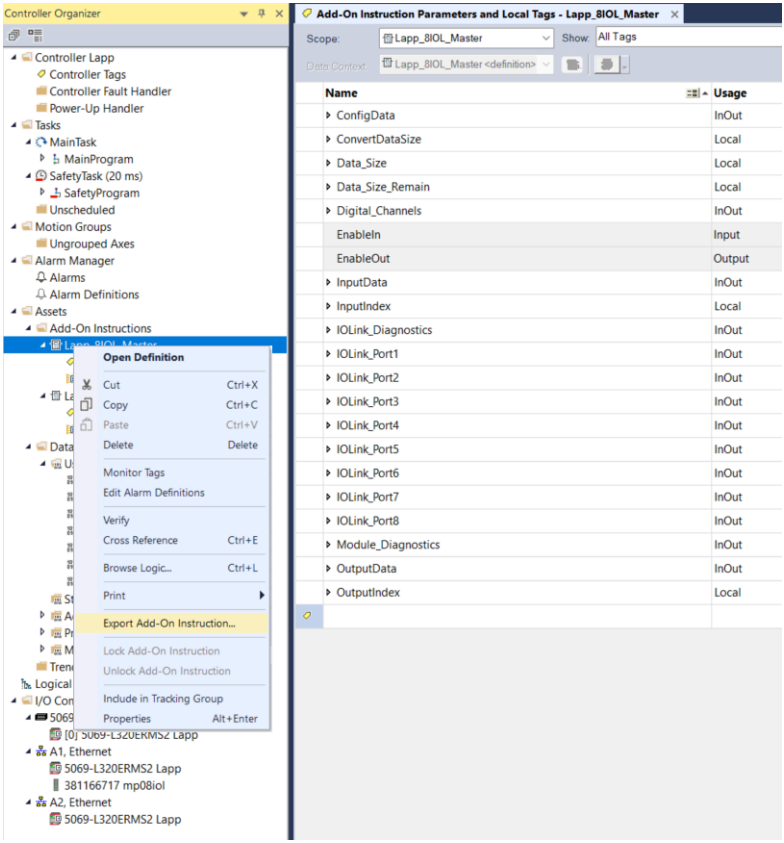
If no error occurred, continue with step **9.**

6. Go to **Edit Tags** and change the data type to the Module-Defined type on your system:

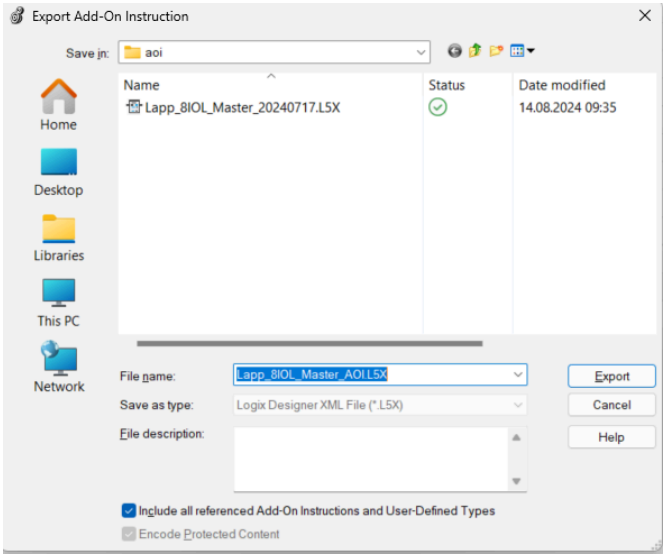


The data type must match the string until the third underscore (_). The CRC32 before **:C:0** is system dependent and does not match the one from the imported AOI. The error is resolved when the red symbol in front of the line is cleared.

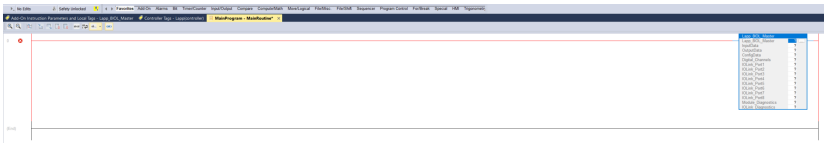
7. When you have changed a data type in the AOI, you should do an export to store this version for future use in other projects on your system. Right-click the AOI and click **Export Add-On Instruction…**:



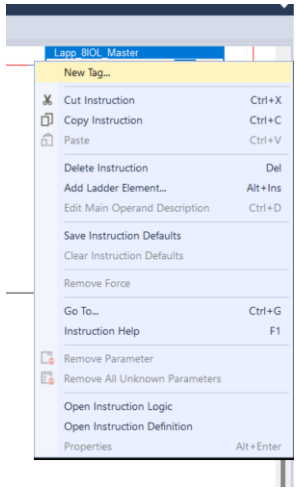
8. Edit the file name and save the AOI:



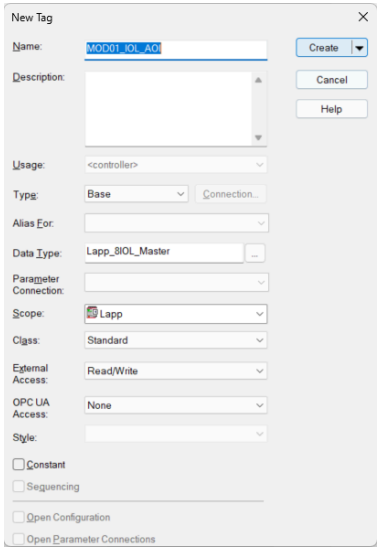
9. For using the AOI go to a logic, e.g. the *MainRoutine*, and add the IO-Link Master AOI via drag and drop to the rung:



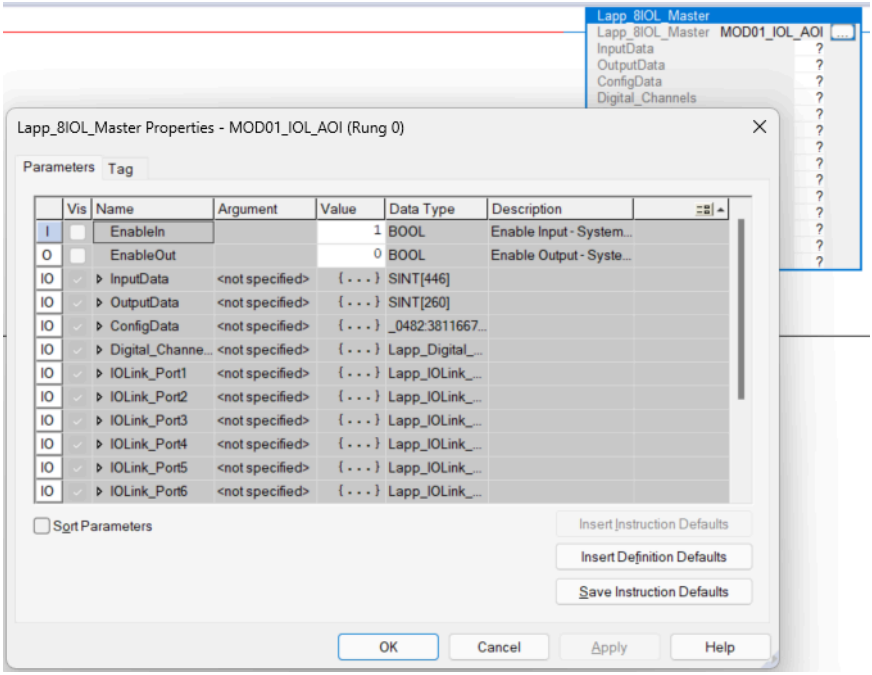
10. Right-Click on the first element of the AOI and click **New Tag...**:



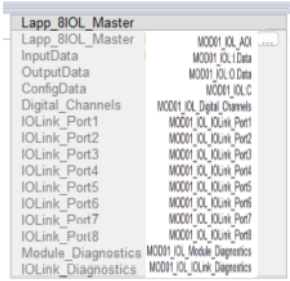
11. Enter a name and create the tag for the AOI:



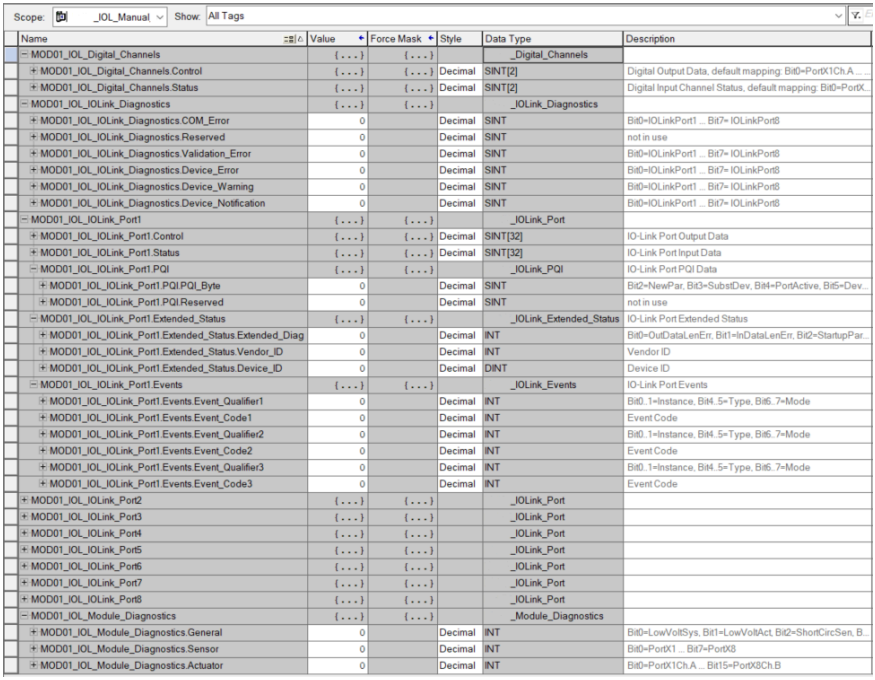
12. Assign the input, output and configuration data of the module:



13. Create the tags for the remaining elements as shown in steps **10.** and **11.**:



14. From now on, your logic does not need to synchronously copy the input and output data anymore. It just uses the new data tags as the interface for exchanging process data with the module:



Scope: _IOL_Manual Show: All Tags

Name	Value	Force Mask	Style	Data Type	Description
MOD01_IOL_Digital_Channels	{...}	{...}		_Digital_Channels	
MOD01_IOL_Digital_Channels.Control	{...}	{...}	Decimal	SINT[2]	Digital Output Data, default mapping: Bit0=PortX1Ch.A ...
MOD01_IOL_Digital_Channels.Status	{...}	{...}	Decimal	SINT[2]	Digital Input Channel Status, default mapping: Bit0=PortX...
MOD01_IOL_IOLink_Diagnostics	{...}	{...}		_IOLink_Diagnostics	
MOD01_IOL_IOLink_Diagnostics.COM_Error	0		Decimal	SINT	Bit0=IOLinkPort1 ... Bit7= IOLinkPort8
MOD01_IOL_IOLink_Diagnostics.Reserved	0		Decimal	SINT	not in use
MOD01_IOL_IOLink_Diagnostics.Validation_Error	0		Decimal	SINT	Bit0=IOLinkPort1 ... Bit7= IOLinkPort8
MOD01_IOL_IOLink_Diagnostics.Device_Error	0		Decimal	SINT	Bit0=IOLinkPort1 ... Bit7= IOLinkPort8
MOD01_IOL_IOLink_Diagnostics.Device_Warning	0		Decimal	SINT	Bit0=IOLinkPort1 ... Bit7= IOLinkPort8
MOD01_IOL_IOLink_Diagnostics.Device_Notification	0		Decimal	SINT	Bit0=IOLinkPort1 ... Bit7= IOLinkPort8
MOD01_IOL_IOLink_Port1	{...}	{...}		_IOLink_Port	
MOD01_IOL_IOLink_Port1.Control	{...}	{...}	Decimal	SINT[32]	IO-Link Port Output Data
MOD01_IOL_IOLink_Port1.Status	{...}	{...}	Decimal	SINT[32]	IO-Link Port Input Data
MOD01_IOL_IOLink_Port1.PQI	{...}	{...}		_IOLink_PQI	IO-Link Port PQI Data
MOD01_IOL_IOLink_Port1.PQI.PQI_Byte	0		Decimal	SINT	Bit2=NewPar, Bit3=SubstDev, Bit4=PortActive, Bit5=Dev...
MOD01_IOL_IOLink_Port1.PQI.Reserved	0		Decimal	SINT	not in use
MOD01_IOL_IOLink_Port1.Extended_Status	{...}	{...}		_IOLink_Extended_Status	IO-Link Port Extended Status
MOD01_IOL_IOLink_Port1.Extended_Status.Extended_Diag	0		Decimal	INT	Bit0=OutDataLenErr, Bit1=InDataLenErr, Bit2=StartupPar...
MOD01_IOL_IOLink_Port1.Extended_Status.Vendor_ID	0		Decimal	INT	Vendor ID
MOD01_IOL_IOLink_Port1.Extended_Status.Device_ID	0		Decimal	DINT	Device ID
MOD01_IOL_IOLink_Port1.Events	{...}	{...}		_IOLink_Events	IO-Link Port Events
MOD01_IOL_IOLink_Port1.Events.Event_Qualifier1	0		Decimal	INT	Bit0..1=Instance, Bit4..5=Type, Bit6..7=Mode
MOD01_IOL_IOLink_Port1.Events.Event_Code1	0		Decimal	INT	Event Code
MOD01_IOL_IOLink_Port1.Events.Event_Qualifier2	0		Decimal	INT	Bit0..1=Instance, Bit4..5=Type, Bit6..7=Mode
MOD01_IOL_IOLink_Port1.Events.Event_Code2	0		Decimal	INT	Event Code
MOD01_IOL_IOLink_Port1.Events.Event_Qualifier3	0		Decimal	INT	Bit0..1=Instance, Bit4..5=Type, Bit6..7=Mode
MOD01_IOL_IOLink_Port1.Events.Event_Code3	0		Decimal	INT	Event Code
MOD01_IOL_IOLink_Port2	{...}	{...}		_IOLink_Port	
MOD01_IOL_IOLink_Port3	{...}	{...}		_IOLink_Port	
MOD01_IOL_IOLink_Port4	{...}	{...}		_IOLink_Port	
MOD01_IOL_IOLink_Port5	{...}	{...}		_IOLink_Port	
MOD01_IOL_IOLink_Port6	{...}	{...}		_IOLink_Port	
MOD01_IOL_IOLink_Port7	{...}	{...}		_IOLink_Port	
MOD01_IOL_IOLink_Port8	{...}	{...}		_IOLink_Port	
MOD01_IOL_Module_Diagnostics	{...}	{...}		_Module_Diagnostics	
MOD01_IOL_Module_Diagnostics.General	0		Decimal	INT	Bit0=LowVoltSys, Bit1=LowVoltAct, Bit2=ShortCircSen, B...
MOD01_IOL_Module_Diagnostics.Sensor	0		Decimal	INT	Bit0=PortX1 ... Bit7=PortX8
MOD01_IOL_Module_Diagnostics.Actuator	0		Decimal	INT	Bit0=PortX1Ch.A ... Bit15=PortX8Ch.B

Note:

If you decide to reduce the input and output data sizes of the connection, you also must set these new sizes to the SINT arrays of *InputData* and *OutputData* within the AOI. In steps **6.** to **8.** it is described how to modify data types of an AOI and how to store the changes.

12 CIP object classes

12.1 EtherNet/IP object classes

According to the CIP specification, the UNITRONIC® ACCESS variants support the following standard EtherNet/IP object classes:

Object Class	Object ID	Instances
Identity Object	0x01	0, 1
Message Router Object	0x02	0 (only on class level)
Assembly Object	0x04	0, 130, 131, 145
Connection Manager Object	0x06	0 (only on class level)
Discrete Input Point Object	0x08	0, 1 .. 16
DLR Object	0x47	0, 1
QoS Object	0x48	0, 1
TCP/IP Interface Object	0xF5	0, 1
Ethernet Link Object	0xF6	0, 1 .. 2
LLDP Management Object	0x109	0, 1

All objects with instance attributes are described in the following chapters.

12.1.1 Identity Object (0x01)

Supported services:

Get Attributes All (0x01)

Reset (0x05): 0 = Reset Module (Warmstart), 1 = Reset to Factory Default

Get Attribute Single (0x0E)

Class attribute (Instance 0)

Attribute	Name	Access	Data type	Description
1	Revision	Get	UINT	Revision of this object
2	Max. Instance	Get	UINT	Maximum instance number of an object currently created in this class level of the device
6	Maximum ID Number Class Attributes	Get	UINT	The attribute ID number of the last class attribute of the class definition implemented in the device
7	Maximum ID Number Instance Attributes	Get	UINT	The attribute ID number of the last instance attribute of the class definition implemented in the device.

Instance attribute (Instance 1)

Attribute	Name	Access	Data Type	Description
1	Vendor ID	Get	UINT	Vendor Identification
2	Device Type	Get	UINT	Indication of general type of product
3	Product Code	Get	UINT	Identification of a particular product of an individual vendor
4	Revision	Get	USINT, USINT	Structure with major and minor revision
5	Status	Get	WORD	Summary status of device: b0: Owned b1: Reserved ("0") b2: Configured b3: Reserved ("0") b4 .. 7: Extended Device Status 0 = Self-Testing or Unknown 1 = Firmware Update in Progress 2 = At least one faulted I/O connection 3 = No I/O connections established 4 = Non-Volatile Configuration bad 5 = Major Fault 6 = At least one I/O connection in RUN mode 7 = At least one I/O connection established, all in IDLE mode 8 = Unused (valid only for instances grater than "1") 9 = Reserved 10 .. 15 = Vendor specific b8: Minor Recoverable Fault b9: Minor Unrecoverable Fault b10: Major Recoverable Fault b11: Major Unrecoverable Fault b12 .. 15: Reserved ("0")
6	Serial Number	Get	UDINT	Serial number of device
7	Product Name	Get	STRING	Human readable identification

Attribute	Name	Access	Data Type	Description
8	State	Get	USINT	Present state of the device: 0 = Nonexistent 1 = Device Self Testing 2 = Standby 3 = Operational 4 = Major Recoverable Fault 5 = Major Unrecoverable Fault 6 .. 254 = Reserved 255 = Default Value
9	Configuration Consistency Value	Get	UINT	Can be a CRC, incrementing count or any other mechanism (vendor specific behavior) to reflect a non-volatile configuration change
19	Protection Mode	Get	WORD	Current protection mode of the device: b0: Implicit Protection enabled b1 .. 2: Reserved b3: Explicit Protection enabled b4 .. 15: Reserved

12.1.2 Assembly Object (0x04)

Supported services:

Get Attribute Single (0x0E)

Set Attribute Single (0x10)

Class attribute (Instance 0)

Attribute	Name	Access	Data type	Description
1	Revision	Get	UINT	Revision of this object
2	Max. Instance	Get	UINT	Maximum instance number of an object currently created in this class level of the device
3	Number of Instances	Get	UINT	Number of Instances currently created in this class level of the device
6	Maximum ID Number Class Attributes	Get	UINT	The attribute ID number of the last class attribute of the class definition implemented in the device
7	Maximum ID Number Instance Attributes	Get	UINT	The attribute ID number of the last instance attribute of the class definition implemented in the device.

Instance attribute (Instance <AssemblyID>)

Attribute	Name	Access	Data Type	Description
3	Data	Get, Set	ARRAY	Assembly Data (Set service only available for consuming assemblies that are not part of an active implicit connection)
4	Size	Get	UINT	Number of bytes in Attribute 3

12.1.3 Discrete Input Point Object (0x08)

Supported services:

Get Attribute Single (0x0E)

Class attribute (Instance 0)

Attribute	Name	Access	Data type	Description
1	Revision	Get	UINT	Revision of this object

Instance attribute (Instance 1 .. 16)

Attribute	Name	Access	Data type	Description
3	Value	Get	BOOL	Input Point Value (0 = OFF, 1 = ON)
4	Status	Get	BOOL	Input Point Status (0 = OK, 1 = Alarm)

12.1.4 DLR Object (0x47)

Supported services:

Get Attributes All (0x01)

Get Attribute Single (0x0E)

Class attribute (Instance 0)

Attribute	Name	Access	Data type	Description
1	Revision	Get	UINT	Revision of this object
2	Max. Instance	Get	UINT	Maximum instance number of an object currently created in this class level of the device
6	Maximum ID Number Class Attributes	Get	UINT	The attribute ID number of the last class attribute of the class definition implemented in the device
7	Maximum ID Number Instance Attributes	Get	UINT	The attribute ID number of the last instance attribute of the class definition implemented in the device.

Instance attribute (Instance 1)

Attribute	Name	Access	Data type	Description
1	Network Topology	Get	BOOL	0 = Linear 1 = Ring
2	Network Status	Get	BOOL	0 = Normal operation 1 = Ring Fault 2 = Unexpected Loop Detected 3 = Partial Network Fault 4 = Rapid Fault/Restore Cycle
10	Active Supervisor Address	Get	ARRAY	Supervisor IP Address, Supervisor MAC Address (0 = not configured)
12	Capability Flags	Get	DWORD	Flag description: b0: Announce-based Ring Node ("0") b1: Beacon-based Ring Node ("1") b2 .. 4: Reserved ("0") b5: Supervisor Capable ("0") b6: Redundant Gateway Capable ("0") b7: Flush_Table frame Capable ("1") b8 .. 15: Reserved ("0")

12.1.5 QoS Object (0x48)

Supported services:

Get Attribute Single (0x0E)

Set Attribute Single (0x10)

Class attribute (Instance 0)

Attribute	Name	Access	Data type	Description
1	Revision	Get	UINT	Revision of this object
2	Max. Instance	Get	UINT	Maximum instance number of an object currently created in this class level of the device
6	Maximum ID Number Class Attributes	Get	UINT	The attribute ID number of the last class attribute of the class definition implemented in the device
7	Maximum ID Number Instance Attributes	Get	UINT	The attribute ID number of the last instance attribute of the class definition implemented in the device.

Instance attribute (Instance 1)

Attribute	Name	Access	Data type	Description
1	802.1Q Tag Enable	Get, Set	USINT	Enables ("1") or disables ("0") sending 802.1Q frames on CIP and IEEE 1588 messages (default value "0")
2	DSCP PTP Event	Get, Set	USINT	DSCP value for PTP Event frames (default value "59")
3	DSCP PTP General	Get, Set	USINT	DSCP value for PTP General frames (default value "47")
4	DSCP Urgent	Get, Set	USINT	CIP transport class 0/1 messages with Urgent priority (default value "55")
5	DSCP Scheduled	Get, Set	USINT	CIP transport class 0/1 messages with Scheduled priority (default value "47")
6	DSCP High	Get, Set	USINT	CIP transport class 0/1 messages with High priority (default value "43")
7	DSCP Low	Get, Set	USINT	CIP transport class 0/1 messages with Low priority (default value "31")
8	DSCP Explicit	Get, Set	USINT	CIP UCMM, CIP transport class 2/3, All other EtherNet/IP encapsulation messages (default value "27")

12.1.6 TCP/IP Object (0xF5)

Supported services:

Get Attributes All (0x01)

Get Attribute Single (0x0E)

Set Attribute Single (0x10)

Class attribute (Instance 0)

Attribute	Name	Access	Data type	Description
1	Revision	Get	UINT	Revision of this object
2	Max. Instance	Get	UINT	Maximum instance number of an object currently created in this class level of the device.

Instance attribute (Instance 1)

Attribute	Name	Access	Data type	Description
1	Status	Get	DWORD	Interface Status description: b0 .. 3: Interface Configuration Status 0 = Not configured 1 = Configuration obtained by BOOTP, DHCP or stored value 2 = Configuration obtained by hardware settings (e.g. rotary switches) 3 .. 15 = Reserved b4: Mcast Pending b5: Interface Configuration Pending b6: Acd Status b7: Acd Fault b8 .. 31: Reserved ("0")

Attribute	Name	Access	Data type	Description
2	Configuration Capability	Get	DWORD	Interface Capability Flags: b0: BOOTP Client ("1") b1: DNS Client ("0") b2: DHCP Client ("1") b3: DHCP-DNS Update ("0") b4: Configuration Settable ("1") b5: Hardware Configurable (0 = no rotary switches; 1 = rotary switches available) b6: Interface Configuration Change Requires Reset ("0") b7: Acd Capable ("1") b8 .. 31: Reserved ("0")
3	Configuration Control	Get, Set	DWORD	Interface Control Flags: b0 .. 3: Configuration Method: 0 = Stored Value 1 = BOOTP 2 = DHCP 3 .. 15 = Reserved b4: DNS Enable ("0") b5 .. 31: Reserved ("0")
4	Physical Link Object	Get	STRUCT	Path to physical link object
5	Interface Configuration	Get, Set	STRUCT	TCP/IP network interface configuration
6	Host Name	Get, Set	STRING	Host name of the device (length of 0 = not configured)
10	Select Acd	Get, Set	BOOL	Enables ("1") or disables ("0") the use of ACD (default value "1")
11	Last Conflict Detected	Get, Set	STRUCT	Structure containing information related to the last conflict detected
13	Encapsulation Inactivity Timeout	Get, Set	UINT	Number of seconds of inactivity before TCP connection is closed: 0 = disable 1 .. 3600 = timeout in seconds 120 = default value

12.1.7 Ethernet Link Object (0xF6)

Supported services:

Get Attribute Single (0x0E)

Set Attribute Single (0x10)

Get and Clear (0x4C)

Class attribute (Instance 0)

Attribute	Name	Access	Data type	Description
1	Revision	Get	UINT	Revision of this object
2	Max. Instance	Get	UINT	Maximum instance number of an object currently created in this class level of the device.
3	Number of Instances	Get	UINT	Number of object instances currently created at this class level of the device (in this case number of ethernet ports)

Instance attribute (Instance 1 .. 2)

Attribute	Name	Access	Data type	Description
1	Interface Speed	Get	UDINT	Current Interface speed in Mbps
2	Interface Flags	Get	DWORD	Interface Flags: b0: Link Status b1: Half ("0") or Full ("1") Duplex b2 .. 4: Negotiation Status: 0 = Auto-negotiation in progress 1 = Auto-negotiation and speed detection failed (using default 10Mbps and half duplex) 2 = Auto negotiation failed but detected speed (using default half duplex) 3 = Successfully negotiated speed and duplex 4 = Auto-negotiation not attempted (forced speed and duplex) b5: Manual Setting Requires Reset b6: Local Hardware Fault b7 .. 31: Reserved ("0")
3	Physical Address	Get	ARRAY	MAC address
4	Interface Counters	Get	STRUCT	Interface Counters
5	Media Counters	Get	STRUCT	Media-specific counters
6	Interface Control	Get, Set	STRUCT	Configuration for physical interface Control Bits (WORD): b0: Auto-negotiate b1: Forced Duplex Mode (0 = Half Duplex; 1 = Full Duplex, only valid when Auto-negotiate = 0) b2 .. 15: Reserved ("0") *Forced Interface Speed in Mbps (UINT)*

Attribute	Name	Access	Data type	Description
7	Interface Type	Get	USINT	Type of interface: 0 = Unknown interface type 1 = Internal interface 2 = Twisted-pair 3 = Optical fiber 4 .. 255 = Reserved
8	Interface State	Get	USINT	State of interface: 0 = Unknown 1 = Enabled and ready to send and receive data 2 = Disabled 3 = Testing 4 .. 255 = Reserved
9	Admin State	Get, Set	USINT	Administrative state: 0 = Reserved 1 = Enable interface 2 = Disable interface 3 .. 255 = Reserved
10	Interface Label	Get	STRING	Human readable identification (size max. 64)
11	Interface Capability	Get	STRUCT	Interface Capability Flags (DWORD): b0: Manual Setting Requires Reset ("0") b1: Auto-negotiate ("1") b2: Auto-MDIX ("1") b3: Manual Speed/Duplex ("1") b4 .. 31: Reserved ("0") Speed/Duplex Array Count of following struct (USINT, 4) Interface Speed in Mbps (UINT, 10/100) Interface Duplex Mode (USINT, 0/1): 0 = Half Duplex 1 = Full Duplex 2 .. 255 = Reserved

12.1.8 LLDP Management Object (0x109)

Supported services:

Get Attributes All (0x01)

Get Attribute Single (0x0E)

Set Attribute Single (0x10)

Class attribute (Instance 0)

Attribute	Name	Access	Data type	Description
1	Revision	Get	UINT	Revision of this object
2	Max. Instance	Get	UINT	Maximum instance number of an object currently created in this class level of the device.
3	Number of Instances	Get	UINT	Number of object instances currently created at this class level of the device (in this case number of ethernet ports)
6	Maximum ID Number Class Attributes	Get	UINT	Attribute ID number of the last class attribute
7	Maximum ID Number Instance Attributes	Get	UINT	Attribute ID number of the last class attribute

Instance attribute (Instance 1)

Attribute	Name	Access	Data type	Description
1	LLDP Enable	Get/Set	STRUCT	LLDP Enable Array Length (UINT): 1 + Class attribute 2 from the Ethernet Link Object (0xF6) = 3 LLDP Enable Array (BYTE): b0: Global Enable, LLDP Tx & Rx Enabled (1) b1: LLDP Tx Enabled (Intance 1 of Ethernet Link Object) (1) b2: LLDP Tx Enabled (Intance 2 of Ethernet Link Object) (1)
2	msgTxInterval	Get/Set	UINT	From 802.1AB-2016: Interval in seconds for transmitting LLDP frames from this device 0 .. 4 = Reserved 5 .. 32768 = Message Transmission Interval for LLDP frames (30) 32769 .. 65535 = Reserved
3	msgTxHold	Get/Set	USINT	From 802.1AB-2016: Multiplier of msgTxInterval to determine the value of the TTL TLV sent to neighboring devices 0 = Reserved 1 .. 100 = Message Transmission Multiplier for LLDP Frames (4) 101 .. 255 = Reserved
4	LLDP Datastore	Get	WORD	Indication of the retrieval methods for the LLDP database: b0: LLDP Data Table Object (0) b1: SNMP (1) b2: NETCONF YANG (0) b3: RESTCONF YANG (0) b4 .. b15: Reserved (0)
5	Last Change	Get	UDINT	Counter in seconds from the last time any entry in the local LLDP database changed or power up

12.2 Vendor specific object classes

The UNITRONIC® ACCESS and UNITRONIC® ACCESS single-protocol EtherNet/IP variants support the following vendor specific object classes:

Object Class	Instances
General Settings Object (0xA0)	0, 1
Channel Settings Object (0xA1)	0, 1 .. 16
IO-Link Diagnosis Settings Object (0xA2)	0, 1
IO-Link Port Settings Object (0xA3)	0, 1 .. n*
IO-Link Failsafe Parameter Object (0xA4)	0, 1 .. n*
IO-Link Device Parameter Object (0xA5)	0, 1 .. n*

*) The available instances depend on the number of IO-Link ports of the device variant. Up to 8 IO-Link ports and instances are supported.

12.2.1 General Settings Object (0xA0)

Supported services:

Get Attribute Single (0x0E)

Set Attribute Single (0x10)

Class attribute (Instance 0)

Attribute	Name	Access	Data type	Description
1	Revision	Get	UINT	Revision of this object
2	Max. Instance	Get	UINT	Maximum instance number of an object currently created in this class level of the device.

Instance attribute (Instance 1)

Attribute	Name	Access	Data type	Description
2	Force Mode Lock	Get, Set	BOOL	0: Disable 1: Enable
3	Web Interface Lock	Get, Set	BOOL	0: Disable 1: Enable
5	Report U_L/U_{Aux} Supply Voltage Fault	Get, Set	BOOL	0: Disable 1: Enable
6	Report DO Fault without U_S	Get, Set	BOOL	0: Disable 1: Enable
7..24	Reserved			
25	CIP object configuration lock	Get, Set	BOOL	0: Disable 1: Enable
26	External configuration lock	Get, Set	BOOL	0: Disable 1: Enable
27..31	Reserved			
32	IO Mapping Mode	Get, Set	SINT	0: Default Assignment 1: Byte Swap 2: LSB Ch.A - MSB Ch.B 3: LSB Ch.B - MSB Ch.A 4: Free IO Mapping

12.2.2 Channel Settings Object (0xA1)

Supported services:

Get Attribute Single (0x0E)

Set Attribute Single (0x10)

Class attribute (Instance 0)

Attribute	Name	Access	Data type	Description
1	Revision	Get	UINT	Revision of this object
2	Max. Instance	Get	UINT	Maximum instance number of an object currently created in this class level of the device.

Instance attribute (Instance 1 .. 16)

Attribute	Name	Access	Data type	Description
1	IO Mapping	Get, Set	SINT	0 .. 15: Bit number of 16 channel process data 16: Inactive
2	DO Surveillance Timeout	Get, Set	INT	0 .. 255
3	DO Failsafe	Get, Set	SINT	0: Set Low 1: Set High 2: Hold Last
4	DO Restart Mode	Get, Set	SINT	0: Disable 1: Enable
5*	DO Switch Mode	Get, Set	SINT	0: Push-Pull (U_S, 0.5 A) 1: High-Side (U_L, 0.5 A) 2: High-Side (U_L, 1.0 A) 3: High-Side (U_L, 1.5 A) 4: High-Side (U_L, 2.0 A) 5: High-Side (U_L, 2.0 A max)
6	DI Logic	Get, Set	SINT	0: Normally Open 1: Normally Close
7	DI Filter	Get, Set	SINT	0: Disabled 1: 1 ms 2: 2 ms 3: 3 ms 4: 6 ms 5: 10 ms 6: 15 ms
8	DI Latch	Get, Set	SINT	0: Disable 1: Enable
9	DI Extension	Get, Set	SINT	0 .. 127
10	Channel Mode	Get, Set	SINT	0: Inactive 1: Digital Output 2: Digital Input 3: IO-Link 4: Auxiliary Power **The supported Channel Mode depends on the device variant.**

* Not available for UNITRONIC® ACCESS single-protocol IO-Link Master variants

12.2.3 IO-Link Diagnosis Settings Object (0xA2)

Only applicable for mixmodule variants.

Supported services:

Get Attribute Single (0x0E)

Set Attribute Single (0x10)

Class attribute (Instance 0)

Attribute	Name	Access	Data type	Description
1	Revision	Get	UINT	Revision of this object
2	Max. Instance	Get	UINT	Maximum instance number of an object currently created in this class level of the device.

Instance attribute (Instance 1)

Attribute	Name	Access	Data type	Description
1	IO-Link Master Diagnosis	Get, Set	BOOL	0: Disable 1: Enable
2	IO-Link Device Error	Get, Set	BOOL	0: Disable 1: Enable
3	IO-Link Device Warning	Get, Set	BOOL	0: Disable 1: Enable
4	IO-Link Device Notification	Get, Set	BOOL	0: Disable 1: Enable
11 .. 12	IO-Link Device Diagnosis Port 7 .. 8	Get, Set	BOOL	0: Disable 1: Enable

12.2.4 IO-Link Port Settings Object (0xA3)

Only applicable for mixmodule variants.

Supported services:

Get Attribute Single (0x0E)

Set Attribute Single (0x10)

Class attribute (Instance 0)

Attribute	Name	Access	Data type	Description
1	Revision	Get	UINT	Revision of this object
2	Max. Instance	Get	UINT	Maximum instance number of an object currently created in this class level of the device.

Instance attribute (Instance 7 .. 8)

Attribute	Name	Access	Data type	Description
1	Output Data Size	Get, Set	SINT	0: No data 1: 2 Byte 2: 4 Byte 3: 8 Byte 4: 16 Byte 5: 32 Byte Only settable when no connection is established.
2	Input Data Size	Get, Set	SINT	0: No data 1: 2 Byte 2: 4 Byte 3: 8 Byte 4: 16 Byte 5: 32 Byte Only settable when no connection is established.

Attribute	Name	Access	Data type	Description
3	Input Data Extension	Get, Set	SINT	0: No Data 1: Extended Status 2: Events 3: Extended Status + Events Only settable when no connection is established.
4	Output Data Swapping Mode	Get, Set	SINT	0: Raw IO-Link Data 1 .. 16: 1 .. 16 WORD 17 .. 24: 1 .. 8 DWORD Only settable when no connection is established.
5	Output Data Swapping Offset	Get, Set	SINT	0 .. 30 Byte Only settable when no connection is established.
6	Input Data Swapping Mode	Get, Set	SINT	0: Raw IO-Link Data 1 .. 16: 1 .. 16 WORD 17 .. 24: 1 .. 8 DWORD Only settable when no connection is established.
7	Input Data Swapping Offset	Get, Set	SINT	0 .. 30 Byte Only settable when no connection is established.
8	IOL Failsafe	Get, Set	SINT	0: Set Low 1: Set High 2: Hold Last 3: Replacement Value (transferred via IO-Link Failsafe Parameter Object) 4: IO-Link Master Command
9	Port Mode	Get, Set	SINT	0: Deactivated 1: Manual (with validation and backup config) 2: Autostart (no validation and backup config)

Attribute	Name	Access	Data type	Description
10	Validation and Backup	Get, Set	SINT	0: No device check and clear (no data storage) 1: Type compatible V1.0 device (no data storage) 2: Type compatible V1.1 device (no data storage) 3: Type compatible V1.1 device with Backup + Restore (Download + Upload) 4 Type compatible V1.1 device with Restore (Download Master to Device)
11	Vendor ID	Get, Set	DINT	0 .. 65535
12	Device ID	Get, Set	DINT	0 .. 16777215
13	Cycle Time	Get, Set	SINT	0: As fast as possible 1: 1.6 ms 2: 3.2 ms 3: 4.8 ms 4: 8.0 ms 5: 20.8 ms 6: 40.0 ms 7: 80.0 ms 8: 120.0 ms

12.2.5 IO-Link Failsafe Parameter Object (0xA4)

Only applicable for mixmodule variants.

Supported services:

Get Attribute Single (0x0E)

Set Attribute Single (0x10)

Class attribute (Instance 0)

Attribute	Name	Access	Data type	Description
1	Revision	Get	UINT	Revision of this object
2	Max. Instance	Get	UINT	Maximum instance number of an object currently created in this class level of the device.

Instance attribute (Instance 7 .. 8)

Attribute	Name	Access	Data type	Description
1	Failsafe value of IO-Link port	Get, Set	Array of Bytes	Depends on configured process data lengths, content must consider possible swapping configuration (failsafe value format must match output data format)

12.2.6 IO-Link Device Parameter Object (0xA5)

Only applicable for mixmodule variants.

Supported services:

Instance 0

Get Attribute Single (0x0E)

Instance 7 .. 8

Get ISDU data (0x4B)

Set ISDU data (0x4C)

Class attribute (Instance 0)

Attribute	Name	Access	Data type	Description
1	Revision	Get	UINT	Revision of this object
2	Max. Instance	Get	UINT	Maximum instance number of an object currently created in this class level of the device.

Instance attribute (Instance 7 .. 8)

Attribute	Name	Access	Data type	Description
1	ISDU data of IO-Link port	Get, Set	Array of Bytes	ISDU data of an IO-Link Device can be read using the Get ISDU data service and written using the Set ISDU data service (see description of these services below).

Get ISDU data

The Index and Subindex shall be set in the source data. The different protocol data formats between EtherNet/IP (little-endian) and IO-Link (big-endian) must be considered. The data length of the response depends on the data type of the IO-Link Device.

Protocol	EtherNet/IP		
Byte	0	1	2
Data type	UINT		USINT
Endianness	LSB	MSB	–
Content	Index		Subindex

Table 16: Source

Protocol	IO-Link		
Byte	0	…	n
Data type	Depends on device data type		
Endianness	MSB	…	LSB
Content	Data or Error occurred (max. 232 Bytes)		

Table 17: Destination

Set ISDU data

The Index, Subindex and IO-Link data shall be set in the source data. The data length of the request depends on the data type of the IO-Link Device. The different protocol data formats between EtherNet/IP (little-endian) and IO-Link (big-endian) must be considered. There is only data in the response of the IO-Link Device available if an Error occurred.

Protocol	EtherNet/IP			IO-Link		
Byte	0	1	2	3	…	n
Data type	UINT		USINT	Depends on device data type		
Endianness	LSB	MSB	–	MSB	…	LSB
Content	Index		Subindex	Data (max. 232 Bytes)		

Table 18: Source

Protocol	IO-Link		
Byte	0	…	n
Data type	Depends on device data type		
Endianness	MSB	…	LSB
Content	Error if occurred (max. 232 Bytes)		

Table 19: Destination

If the read or write request is not successful (CIP response status is unequal "0"), the following response format of 4 bytes is available:

Name	Data type	Error code description	Error code
IO-Link Master Error	UINT	Service not available	1
		Port blocked	2
		Timeout	3
		Invalid index	4
		Invalid sub-index	5
		Wrong port	6
		Wrong port function	7
		Invalid length	8
		ISDU not supported	9
IO-Link Device Error	USINT	Refer to IO-Link specification	–
IO-Link Device Additional Error	USINT	Refer to IO-Link specification	–

In Get/Set ISDU data on page 143, you find an example for Rockwell Automation Studio 5000®.

12.3 Message configuration in Rockwell Automation Studio 5000®

Attributes of CIP object classes can be handled in Rockwell Automation Studio 5000® by the *Message instruction*. This requires the selection of the proper message and service type with its respective service code.

The channels as in the *Channel Settings Object* are each assigned in ascending order to an instance ID.

Assignment of the channels:

Channel 0	Port X1.ChA	CIP object instance 1
Channel 1	Port X1.ChB	CIP object instance 2
[…]	[…]	[…]
Channel 14	Port X8.ChA	CIP object instance 15
Channel 15	Port X8.ChB	CIP object instance 16

The IO-Link ports as in the *IO-Link Port Settings Object*, *IO-Link Failsafe Parameter Object* and *IO-Link Device Parameter Object* are each assigned in ascending order to an instance ID.

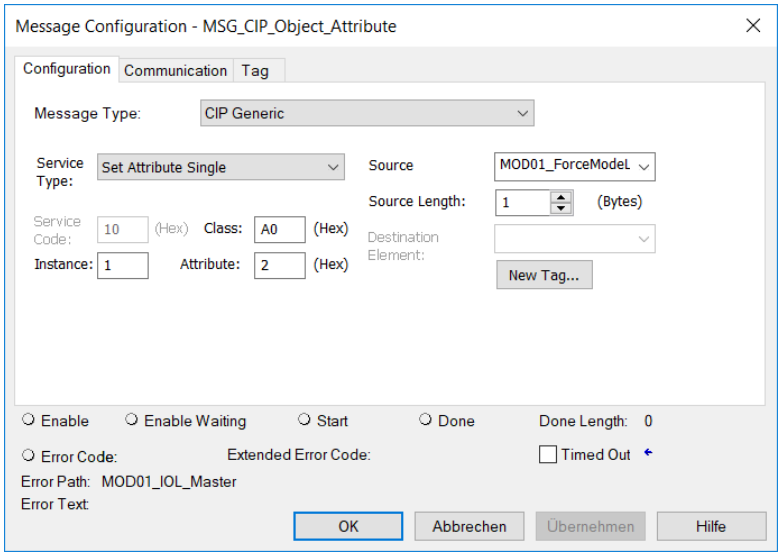
Assignment of the IO-Link ports:

IO-Link port 1	Port X1.ChA	CIP object instance 1
[…]	[…]	[…]
IO-Link port 8	Port X8.ChA	CIP object instance 8

12.3.1 Get/Set attribute single

Every attribute with exception of the IO-Link Device Parameter Object Instance 1..n can be accessed using the *Get/Set attribute single* service by the CIP object class ID, the instance ID and attribute ID. The respective data is described in the previous chapters.

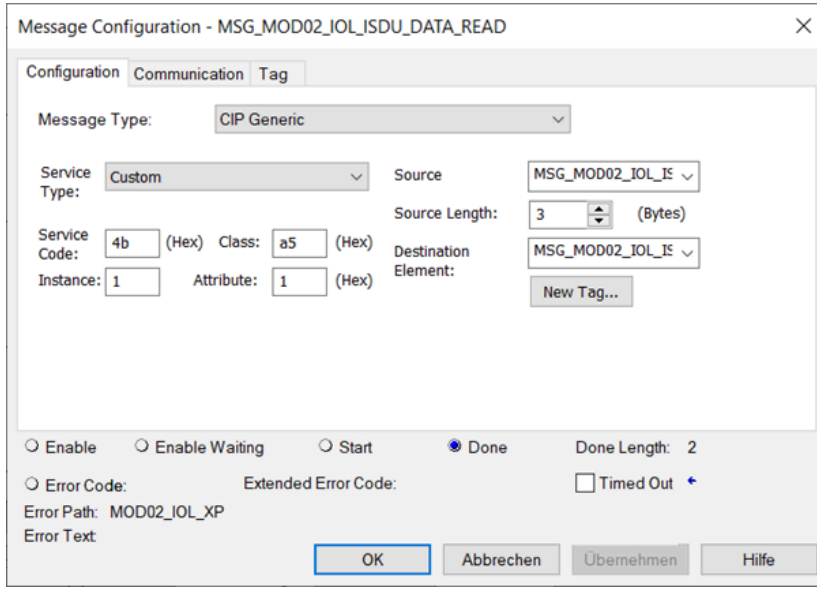
The following image shows an example of how to set *Force Mode Lock* (Attribute 2) of the *General Settings Object (0xA0)* with the *Message instruction*:



12.3.2 Get/Set ISDU data

The IO-Link Device Parameter Object Instance 1..n can be accessed using the vendor specific *Get/Set ISDU data* service by the CIP object class ID, the instance ID and attribute ID. The Index and Subindex shall be set in the source data. For the *Set ISDU data* service, the IO-Link data must be appended. The different protocol data formats between EtherNet/IP (little-endian) and IO-Link (big-endian) must be considered. The respective data is described in the previous chapters.

The following figure shows an example of how to get an IO-Link Device parameter using the *Get ISDU data (0x4B)* service of the *IO-Link Device Parameter Object (0xA5)* with the *Message instruction:*



The Index (0x003C) and Subindex (0x01) of the IO-Link device parameter are set in the little-endian format of EtherNet/IP in the source data:

− MSG_MOD02_IOL_ISDU_SRC_READ	{...}	{...}	Hex	SINT[8]
+ MSG_MOD02_IOL_ISDU_SRC_READ[0]	16#3c		Hex	SINT
+ MSG_MOD02_IOL_ISDU_SRC_READ[1]	16#00		Hex	SINT
+ MSG_MOD02_IOL_ISDU_SRC_READ[2]	16#01		Hex	SINT

The response data of the IO-Link device can be found in the destination element. In the following example, the received value is of the type UINT in the big-endian format of IO-Link (0x0546 = 1350):

− MSG_MOD02_IOL_ISDU_DST_READ	{...}	{...}	Hex	SINT[8]
+ MSG_MOD02_IOL_ISDU_DST_READ[0]	16#05		Hex	SINT
+ MSG_MOD02_IOL_ISDU_DST_READ[1]	16#46		Hex	SINT

13 Diagnostics processing

13.1 Error of the system/sensor power supply

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

The green U_S indicator is off.

The error diagnosis has no effect on the outputs.



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

The following diagnostics are generated in the producing data image:

General diagnostics	Bit	7	6	5	4	3	2	1	0
General Bit	Byte 0	IME	FME	DTO	DTU	SCA	SCS	LVA	**LVS**
	Byte 1	0	0	0	0	IDN	IDW	IDE	IVE

LVS Low Voltage System/Sensor Supply

13.2 Error of the auxiliary/actuator power supply

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

The following diagnostics are generated in the *producing data image*:

General diagnostics	**Bit**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
General Bit	Byte 0	IME	FME	DTO	DTU	SCA	SCS	**LVA**	LVS
	Byte 1	0	0	0	IDN	IDW	IDE	IVE	0

LVA Low Voltage Actuator Supply

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

The following diagnostics are generated in the *producing data image*:

Actuator/U_{AUX} diagnostics	**Bit**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Channel number (fix)	Byte 0	8	7	6	5	4	3	2	1
	Byte 1	16	15	14	13	12	11	10	9

1 .. 16 Actuator/U_L/U_{Aux} channel error on channel 1 .. 16

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

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

In case of an overload or a short circuit between pin 1 and pin 3 on the ports (X1 .. X8), the following channel-specific diagnostics in the producing data image are generated:

General diagnostics	Bit	7	6	5	4	3	2	1	0
General Bit	Byte 0	IME	FME	DTO	DTU	SCA	**SCS**	LVA	LVS
	Byte 1	0	0	0	0	IDN	IDW	IDE	IVE

SCS Short Circuit Sensor

Sensor diagnostics	Bit	7	6	5	4	3	2	1	0
Port number	Byte 0	**X8**	**X7**	**X6**	**X5**	**X4**	**X3**	**X2**	**X1**
	Byte 1	0	0	0	0	0	0	0	0

X1 .. 8 Sensor Short Circuit on Port X1 .. X8

13.4 Overload/short circuit of the digital outputs

In case of an overload or a short circuit of an output channel, the following channel-specific diagnostics are generated in the *producing data image*:

General diagnostics	Bit	7	6	5	4	3	2	1	0
General Bit	Byte 0	IME	FME	DTO	DTU	**SCA**	SCS	LVA	LVS
	Byte 1	0	0	0	0	IDN	IDW	IDE	IVE

SCA Short Circuit Actuator/U_L/U_{AUX}

Actuator/U_{AUX} diagnostics	Bit	7	6	5	4	3	2	1	0
Channel number (fix)	Byte 0	8	7	6	5	4	3	2	1
	Byte 1	16	15	14	13	12	11	10	9

1 .. 16 Actuator/U_L/U_{AUX} channel error on channel 1 .. 16

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

When an output channel is activated (rising edge of the channel state), the channel errors are filtered for the period that is set by the "Surveillance-Timeout" parameter via 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 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, the filter time between error detection and the diagnosis is typically 5 ms.

13.5 IO-Link COM error

If an IO-Link Device in COM mode is unplugged, an incorrect IO-Link Device is plugged in, or an electrical fault occurs on the C/Q (Pin 4) line, for example, due to a short circuit, the following diagnostics are generated in the *producing data image*:

IO-Link diagnostics	Bit	7	6	5	4	3	2	1	0
General Bit	Byte 0	**ICE8**	**ICE7**	**ICE6**	**ICE5**	**ICE4**	**ICE3**	**ICE2**	**ICE1**
	Byte 1	0	0	0	0	0	0	0	0
	Byte 2	IVE8	IVE7	IVE6	IVE5	IVE4	IVE3	IVE2	IVE1
	Byte 3	IDE8	IDE7	IDE6	IDE5	IDE4	IDE3	IDE2	IDE1
	Byte 4	IDW8	IDW7	IDW6	IDW5	IDW4	IDW3	IDW2	IDW1
	Byte 5	IDN8	IDN7	IDN6	IDN5	IDN4	IDN3	IDN2	IDN1

ICE1 .. 8 IO-Link Port COM Error (device missing, broken wire, short circuit)

13.6 IO-Link validation error

If an IO-Link Device is exchanged by a new device, the validation is configured. The vendor ID and/or device ID do not match the data of the device and the following diagnostics are generated in the *producing data image*:

IO-Link diagnostics	Bit	7	6	5	4	3	2	1	0
General Bit	Byte 0	ICE8	ICE7	ICE6	ICE5	ICE4	ICE3	ICE2	ICE1
	Byte 1	0	0	0	0	0	0	0	0
	Byte 2	**IVE8**	**IVE7**	**IVE6**	**IVE5**	**IVE4**	**IVE3**	**IVE2**	**IVE1**
	Byte 3	IDE8	IDE7	IDE6	IDE5	IDE4	IDE3	IDE2	IDE1
	Byte 4	IDW8	IDW7	IDW6	IDW5	IDW4	IDW3	IDW2	IDW1
	Byte 5	IDN8	IDN7	IDN6	IDN5	IDN4	IDN3	IDN2	IDN1

IVE1 .. 8 IO-Link Port Validation Error

If extended status data is enabled by the configuration of an IO-Link port, the vendor ID and device ID are additionally transferred in the *producing data image*.

13.7 IO-Link device diagnostics

The diagnostics of an IO-Link Device come in three different levels: Error, Warning or Notification. The following diagnostics are generated in the *producing data image*:

IO-Link diagnostics	Bit	7	6	5	4	3	2	1	0
General Bit	Byte 0	ICE8	ICE7	ICE6	ICE5	ICE4	ICE3	ICE2	ICE1
	Byte 1	0	0	0	0	0	0	0	0
	Byte 2	IVE8	IVE7	IVE6	IVE5	IVE4	IVE3	IVE2	IVE1
	Byte 3	**IDE8**	**IDE7**	**IDE6**	**IDE5**	**IDE4**	**IDE3**	**IDE2**	**IDE1**
	Byte 4	**IDW8**	**IDW7**	**IDW6**	**IDW5**	**IDW4**	**IDW3**	**IDW2**	**IDW1**
	Byte 5	**IDN8**	**IDN7**	**IDN6**	**IDN5**	**IDN4**	**IDN3**	**IDN2**	**IDN1**

IDE1 .. 8 IO-Link Port Device Error

IDW1 .. 8 IO-Link Port Device Warning

IDN1 .. 8 IO-Link Port Device Notification

If IO-Link event data is enabled by the configuration of an IO-Link port the device additionally reports event codes in the *producing data image*. Use the IO-Link Device documentation to interpret the error message.

14 IIoT functionality

The UNITRONIC® ACCESS 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 UNITRONIC® ACCESS 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 UNITRONIC® ACCESS variants provide user administration, which is also applicable for accessing and configuring the IIoT protocols. This allows you to manage all modification options for the device settings via personalized user authorizations.

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

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

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

14.1 MQTT

MQTT functions are **only** applicable for the following UNITRONIC® ACCESS variants:

- MP08IOLA08DIO

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

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

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

14.1.1 MQTT configuration

In **delivery state**, MQTT functions are **disabled**. The MQTT client can be configured either using the Web interface or directly via a JSON object sent in an HTTP/HTTPS request. For more information see chapter MQTT configuration - Quick start guide on page 169.

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: "**unitronic**")
will-enable	boolean	If true, the device provides a last will message to the broker	true / **false**
will-topic	string	The topic for the last will message.	(Default: **null**)
auto-publish	boolean	If true, all enabled domains will be published automatically in the specified interval.	**true** / false
publish-interval	number	The publish interval in ms if auto-publish is enabled. Minimum is 250 ms.	**2000**
publish-identity	boolean	If true, all identity domain data will be published	**true** / false
publish-config	boolean	If true, all config domain data will be published	**true** / false
publish-status	boolean	If true, all status domain data will be published	**true** / false
publish-process	boolean	If true, all process domain data will be published	**true** / false
publish-devices	boolean	If true, all IO-Link Device domain data will be published	true / **false**
commands-allowed	boolean	Master switch for MQTT commands. If false, the device will not subscribe to any command topic, even if specific command topics are activated below.	true / **false**
force-allowed	boolean	If true, the device accepts force commands via MQTT.	true / **false**
reset-allowed	boolean	If true, the device accepts restart and factory reset commands via MQTT.	true / **false**
config-allowed	boolean	If true, the device accepts configuration changes via MQTT.	true / **false**

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

Table 20: 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 157.

14.1.2 MQTT topics

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

14.1.2.1 Base topic

For all UNITRONIC® ACCESS 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 21: Base topic variables on page 157.

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

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

Table 21: Base topic variables

Example:

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

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

`Base-Topic/domain/….`

There are the following domains:

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

Table 22: Data domains

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

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

Table 23: 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 24: Use case examples

14.1.2.2 Publish topic

Overview of all publish JSON data for the defined topics:

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

Table 25: Identity/gateway

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

Table 26: Config/gateway

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

Table 27: Status/gateway

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

Table 28: Process/gateway

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

Table 29: Identity/port/1 .. 8

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

Table 30: Config/port/1 .. 8

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

Table 31: Status/port/1 .. 8

14.1.2.3 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 33: Force object: Digital on page 166)		
iol	array (Table 34: Force object: IOL (IO-Link devices only) on page 166)		

Table 32: 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 33: Force object: Digital

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

Table 34: Force object: IOL (IO-Link devices only)

[…]/config

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

Property	Data type	Example values	Remarks
portmode	array (Table 36: Config object: Portmode on page 167)		
ip_address	string	"192.168.1.5"	
subnet_mask	string	"255.255.255.0"	
gateway	string	"192.168.1.100"	

Table 35: Config object properties

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

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

Table 36: 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 37: 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).

14.1.3 MQTT configuration - Quick start guide

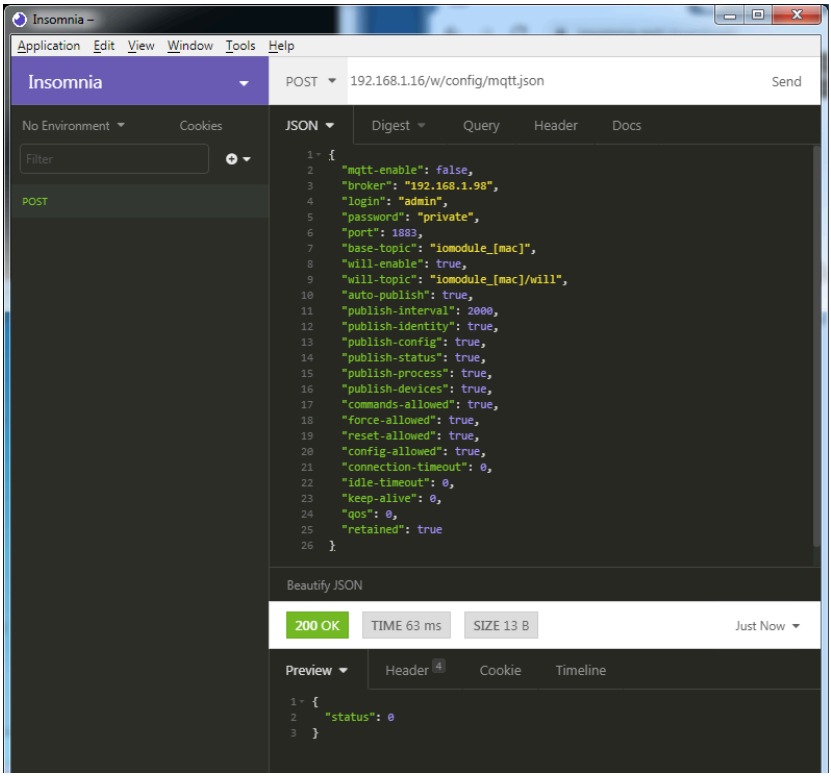
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

14.1.3.1 MQTT configuration via JSON

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

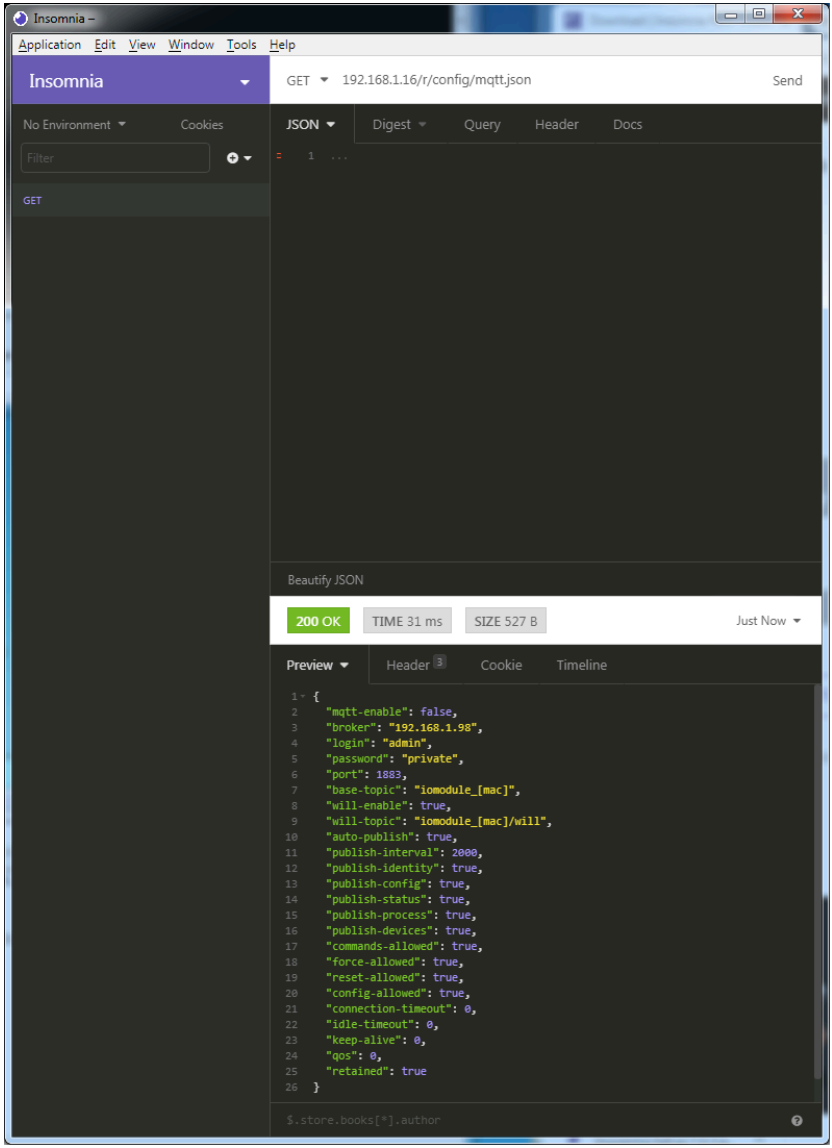
2. Configure MQTT:

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



3. Read MQTT:

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



14.2 OPC UA

OPC UA functions are **only** applicable for the following UNITRONIC® ACCESS variants:

- MP08IOLA08DIO

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. UNITRONIC® ACCESS provides an OPC UA server on field device level, with which an OPC UA client can connect for information exchange secure in transmission.

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

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

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

Table 38: Non-supported OPC UA features according to the IO-Link Companion Specification

14.2.1 OPC UA configuration

In **delivery state**, OPC UA functions are **disabled**. The OPC UA Server can be configured either using the Web interface or directly via a JSON Object sent in an HTTP/HTTPS request. For more information see OPC UA configuration - Quick start guide on page 175.

The configuration URL is:

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

The configuration can also read back as a JSON file:

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

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

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

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

Table 39: OPC UA Configuration

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

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

Response:

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

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

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

Examples:

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

{"status": 0}

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

14.2.2 OPC UA address space

OPC UA provides different services on the UNITRONIC® ACCESS 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 UNITRONIC® ACCESS devices. The objects and information displayed depend on the device variant used.

14.2.3 OPC UA configuration - Quick start guide

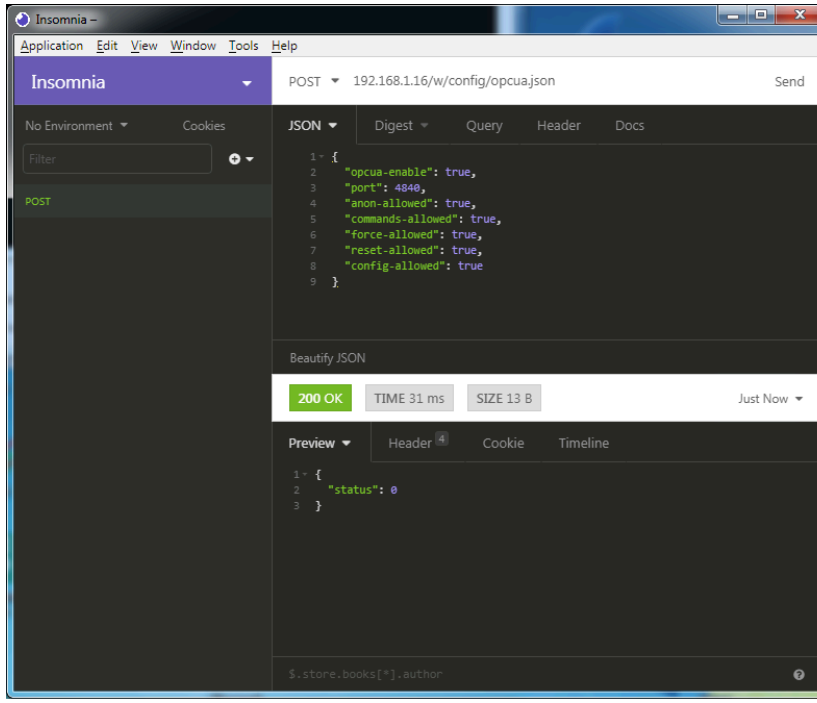
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

14.2.3.1 OPC UA configuration via JSON

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

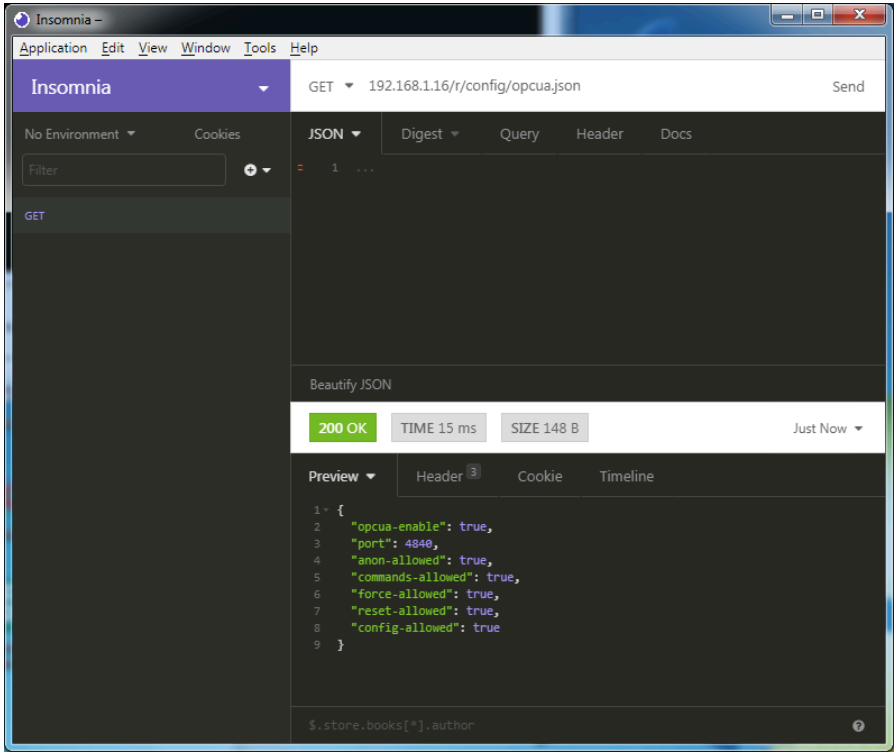
2. Configure OPC UA:

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



3. Read OPC UA:

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



14.3 REST API

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

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

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

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

   `JSON_Integration_10222_V100_Mar20.pdf`

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

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

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

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

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

Table 40: Support of REST API features according to the IO-Link specification

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

14.3.1 Standard device information

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

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

14.3.2 Structure

Name	Data type	Description	Example
name	string	Device name	"MP08IOLA08DIO"
order-id	string	Ordering number	"381166717"
fw-version	string	Firmware version	"V.11.2.0.0 - 08.08.2024"
hw-version	string	Hardware version	"V.1.00"
mac	string	MAC address of the device	"7C F9 5C 4C CC CE"
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
iol	IOL Object	Contains all IO-Link related information such as events, port states, device parameters.	
iol/diagGateway	array of DIAG	Array of currently active device/ gateway related events	
iol/diagMaster	array of DIAG	Array of currently active IOL-Master related events	
iol/ports	array of PORT (8)	Contains one element for each IO-Link port	
CHANNEL Object			
name	string	Name of channel	
type	number	Hardware channel type as number: 0 = DIO 1 = Input 2 = Output 3 = Input/Output 4 = IO-Link 5 = IOL AUX 6 = IOL AUX with DO 7 = IOL AUX with DO. Can be deactivated. 8 = Channel not available	
type_text	string	Textual representation of the channel type	
config	number	Current configuration of the channel: 0 = DIO 1 = Input 2 = Output 3 = IO-Link 4 = Deactivated 5 = IOL AUX	
config_text	string	Textual representation of the current config	
inputState	boolean	Input data (producing data) bit to the PLC	
outputState	boolean	Output data bit to the physical output pin	

Name	Data type	Description	Example
forced	boolean	True, if the output pin of this channel is forced	
simulated	boolean	True, if the input value to the PLC of this channel is simulated	
actuatorDiag	boolean	True, if the output is in short circuit / overload condition	
sensorDiag	boolean	True, if the sensor supply (Pin 1) is in short circuit / overload condition	
maxOutputCurrent_mA	number	Maximum output current of the output in mA	
current_mA	number	Measured current of the output in mA (if current measurement is available)	
voltage_mV	number	Measured voltage of this output in mV (if voltage measurement is available)	
PORT Object			
port_type	string	Textual representation of the IO-Link port type	
iolink_mode	number	**Current port mode:** 0 = Inactive 1 = Digital output 2= Digital input 3 = SIO 4 = IO-Link	
iolink_text	string	Textual representation of the current port mode	"Digital Input"
aux_mode	number	Indicates the configured mode for the Pin 2: 0 = No AUX 1 = AUX output (always on) 2 = Digital output (can be controlled by cyclic data) 3 = Digital input	
aux_text	string	Textual representation of the current aux mode	"AUX Output"
cq_mode	number	Port mode according to IOL specification	
iq_mode	number	Pin2 mode according to IOL specification	

Name	Data type	Description	Example
port_status	number	Port status according to IOL specification	
ds_fault	number	Data storage error number	
ds_fault_text	string	Textual data storage error.	
device	DEVICE Object	IO-Link device parameters. → Null if no IO-Link communication active	
diag	array of DIAG (n)	Array of port related events	
DIAG Object			
error	number	Error code	
source	string	Source of the current error.	"device" "master"
eventcode	number	Event code according to IO-Link specification	
eventqualifier	number	Event qualifier according to IO-Link specification	
message	string	Error message	"Supply Voltage fault"
DEVICE Object		Standard parameters of the IOL-Device	
device_id	number		
vendor_id	number		
serial	string		
baudrate	string	Baudrate (COM1,2,3)	
cycle_time	number	Cycle time in microseconds	
input_len	array of numbers (n)	IOL input length in bytes	
output_len	array of numbers (n)	IOL output length in bytes	
input_data	array of numbers (n)	IOL input data	
output_data	array of numbers (n)	IOL output data	
pd_valid	number	"1", if IOL input data is valid	
pdout_valid	number	"1", if IOL output data is valid	
FORCING Object		Forcing information of the device	
forcingActive	boolean	Force mode is currently active	

Name	Data type	Description	Example
forcingPossible	boolean	True, if forcing is possible and force mode can be activated	
ownForcing	boolean	True, if forcing is performed by REST API at the moment	
forcingClient	string	Current forcing client identifier	
digitalOutForced	array of numbers (2)	The force values of all 16 digital output channels.	
digitalOutMask	array of numbers (2)	The forcing mask of all 16 digital output channels.	
digitalInForced	array of numbers (2)	The force values of all 16 digital input channels.	
digitalInMask	array of numbers (2)	The forcing mask of all 16 digital input channels.	

14.3.3 Configuration and forcing

Method: POST

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

Parameters: None

Post-Body: JSON Object

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

Table 41: Root object

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

Table 42: 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 43: Digital object

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

Table 44: IOL object

14.3.4 Reading and writing ISDU parameters

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

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

14.3.4.1 Reading ISDU

Method:	POST
URL:	`<ip>/r/isdu.json`
Parameters:	port (0-7)
Example:	`192.168.1.20/r/isdu.json?port=5`
Post-Body:	JSON array of read ISDU object

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

Table 45: Read ISDU object

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

Table 46: Read ISDU response object

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

Table 47: Read ISDU data object

14.3.4.2 Writing ISDU

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

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

Table 48: Write ISDU object

Response: Write ISDU response object

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

Table 49: Write ISDU response object

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

Table 50: Write ISDU data object

Attention: For UNITRONIC® ACCESS device variants with HTTPS feature, https:// must be used in front of <ip> for every REST API.

14.3.5 Upload and process an IODD file

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

Perform the following work steps:

1. Check file upload status

   Send request: GET file_upload

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

   Expected response:

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

   Check the status ID. If status is '0', you can start a new iodd upload process. For reference, see tables Table 51: Status ID and meaning on page 195 and Table 52: Error ID and meaning on page 196. Proceed with the next step.

2. Initiate file upload

   Send request: POST file_upload

   Content-Type: application/json

   Purpose: Send details about the file to be uploaded.

   Expected response:

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

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

The action will allways be iodd.

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

The correct content type has to be set.

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

3. Upload file content

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

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

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

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

4. Monitor upload status

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

   Purpose: Get the current file upload status.

   Expected response:

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

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

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

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

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

Table 51: Status ID and meaning

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

Table 52: Error ID and meaning

14.3.6 Example: Reading ISDU

ISDU read request

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

Response

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

14.3.7 Example: Writing ISDU

ISDU write request

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

Response

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

14.4 CoAP server

The CoAP server functions are **only** applicable for the following UNITRONIC® ACCESS variants:

- MP08IOLA08DIO

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

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

14.4.1 CoAP configuration

In delivery state, CoAP functions are *disabled*. The CoAP server can be configured either using the Web interface or directly via a JSON object sent in an HTTP/HTTPS request. For more information see chapter CoAP configuration - Quick start guide on page 202.

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 53: 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"}]}
```

14.4.2 REST API access via CoAP

A connection to the CoAP server running on the UNITRONIC® ACCESS multi-protocol variants can be established via the following URL:

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

For UNITRONIC® ACCESS, the following REST API Requests (JSON format) can be accessed via a CoAP endpoint:

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

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

Table 54: REST API access via CoAP

14.4.3 CoAP configuration - Quick start guide

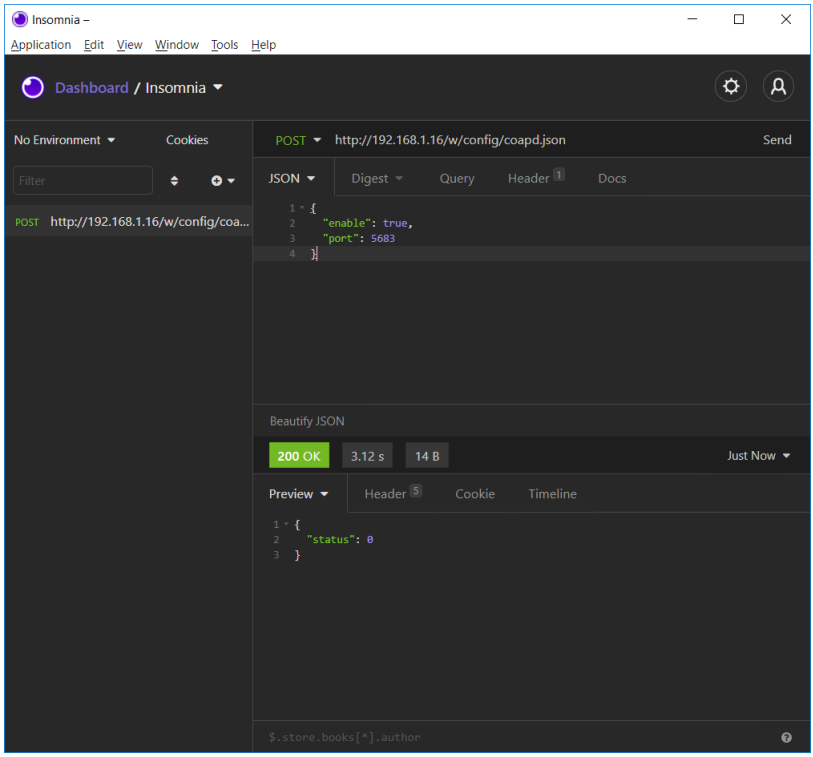
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

14.4.3.1 CoAP configuration via JSON

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

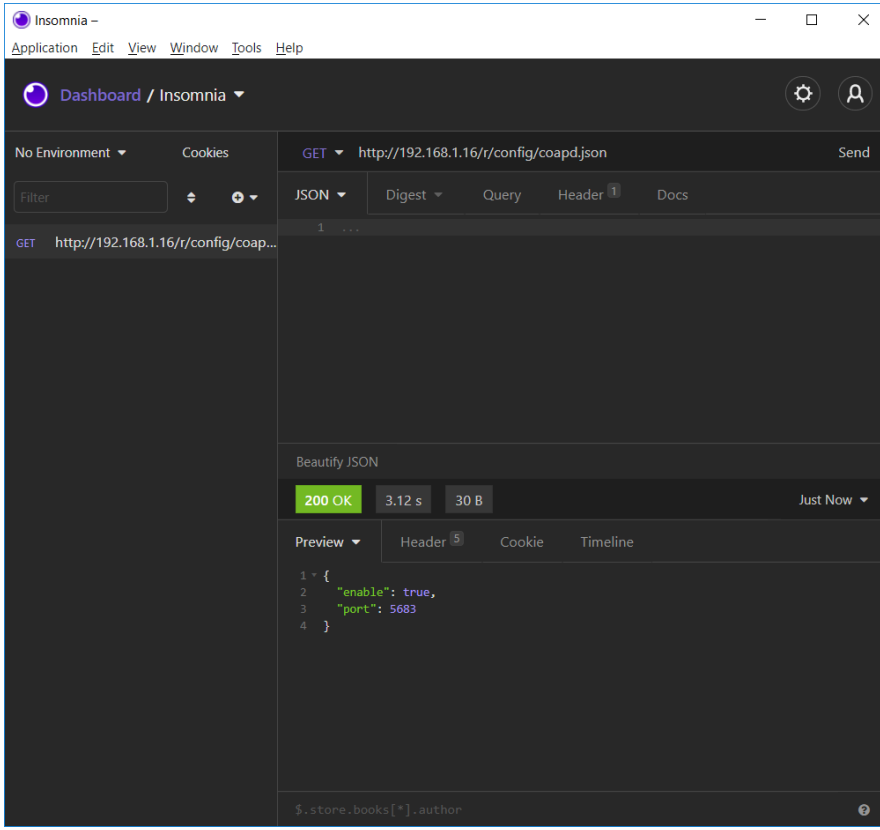
2. Configure CoAP:

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



3. Read CoAP configuration:

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



14.5 Syslog

Syslog functions are **only** applicable for the following UNITRONIC® ACCESS variants:

- MP08IOLA08DIO

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

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

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

14.5.1 Syslog configuration

In **delivery state**, Syslog functions are **disabled**. The Syslog client can be configured either using the Web interface or directly via a JSON object sent in an HTTP/HTTPS request. For more information see chapter Syslog configuration - Quick start guide on page 207.

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 55: 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"}]}
```

14.5.2 Syslog configuration - Quick start guide

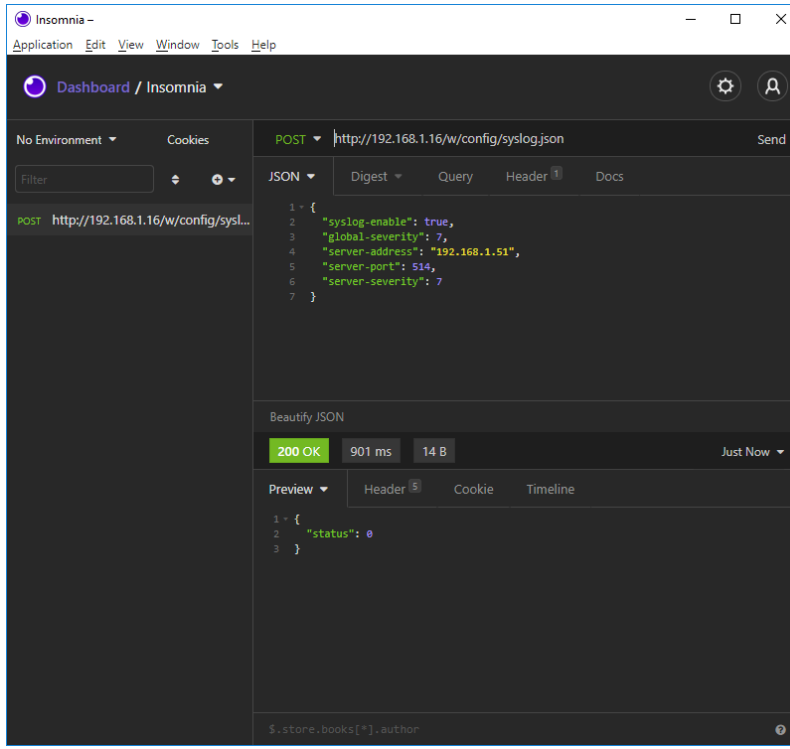
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

14.5.2.1 Syslog configuration via JSON

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

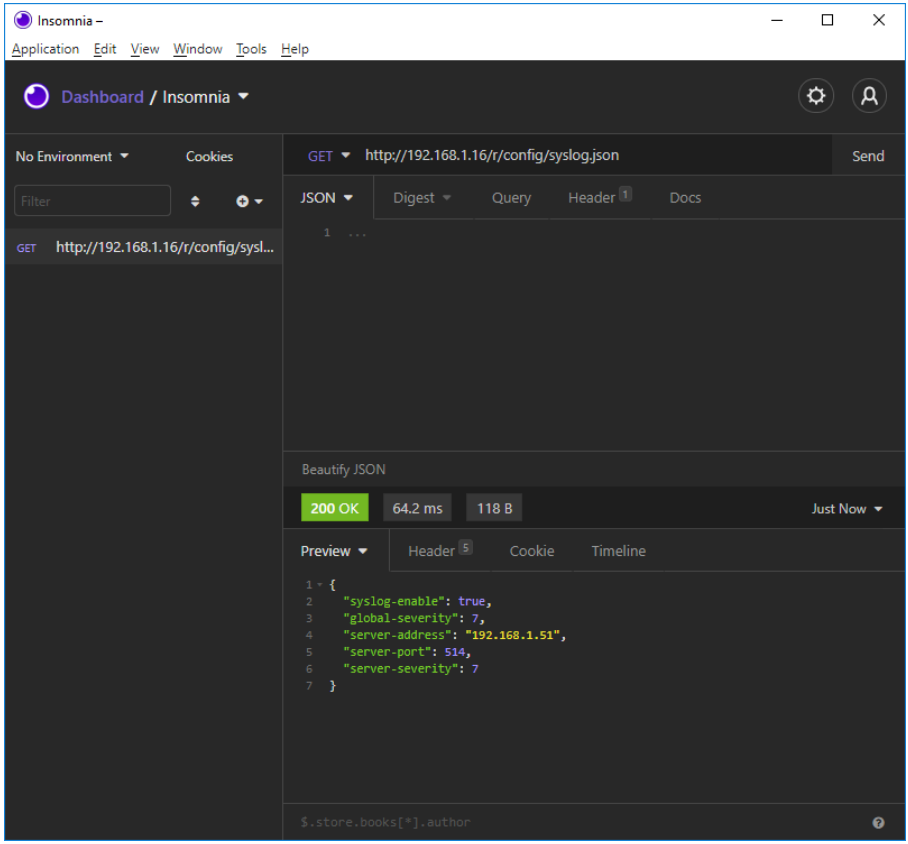
2. Configure Syslog:

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



3. Read Syslog configuration:

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



14.6 Network Time Protocol (NTP)

The NTP function is **only** applicable for the following UNITRONIC® ACCESS variant:

- MP08IOLA08DIO

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

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

14.6.1 NTP configuration

In **delivery state**, the NTP client is **disabled**. The NTP client can be configured either using the Web interface or directly via a JSON object sent in an HTTP/HTTPS request. For more information see chapter NTP configuration - Quick start guide on page 211.

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 56: 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"}]}
```

14.6.2 NTP configuration - Quick start guide

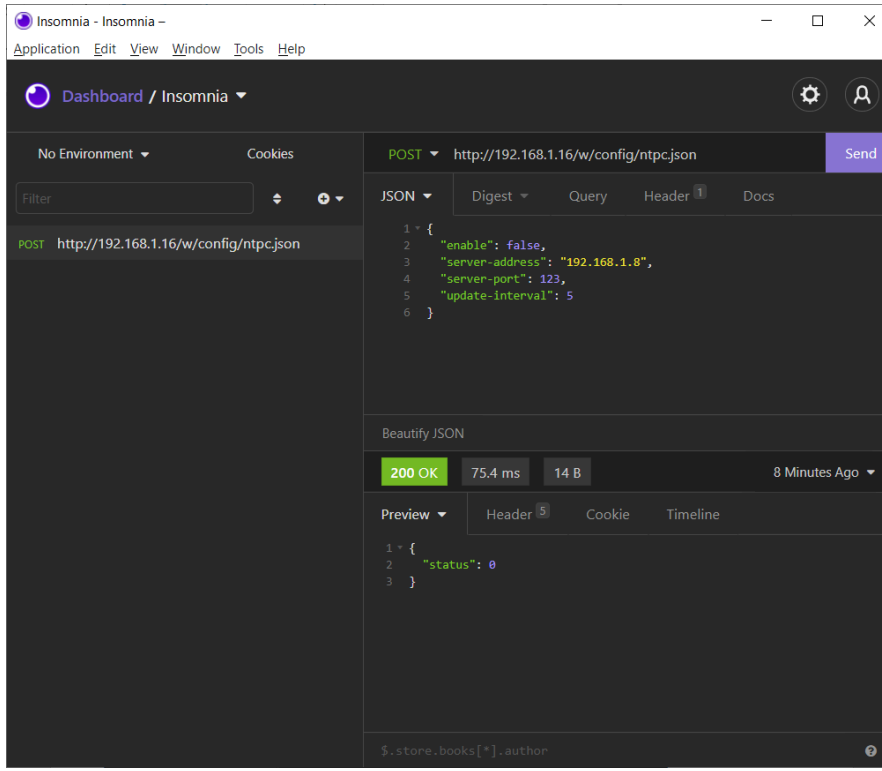
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

14.6.2.1 NTP configuration via JSON

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

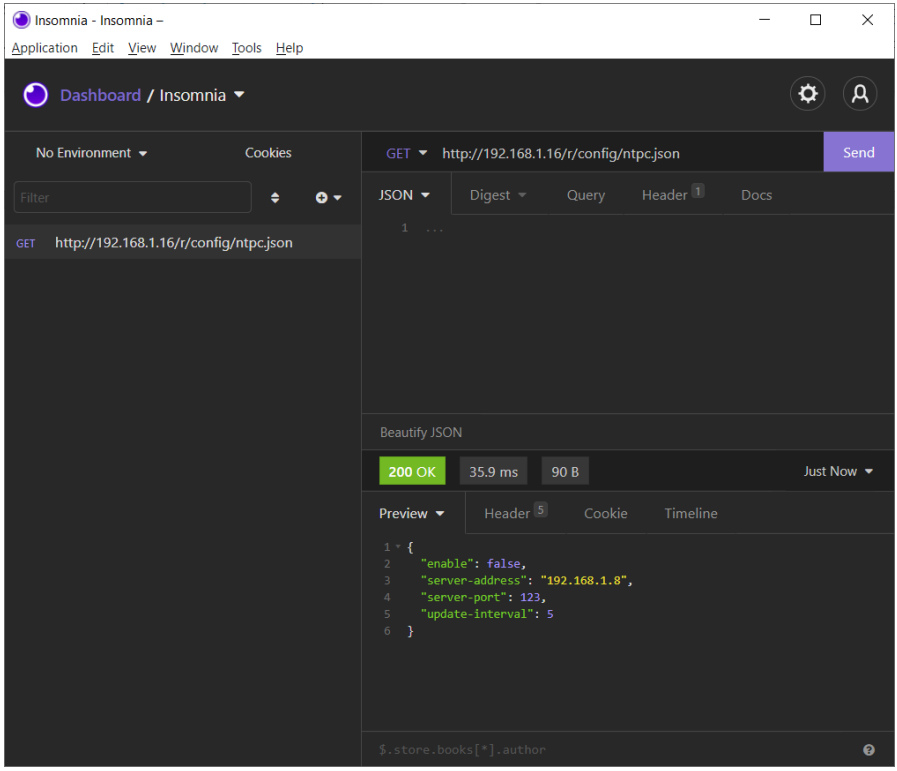
2. Configure NTP:

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



3. Read NTP configuration:

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



15 The integrated Web server

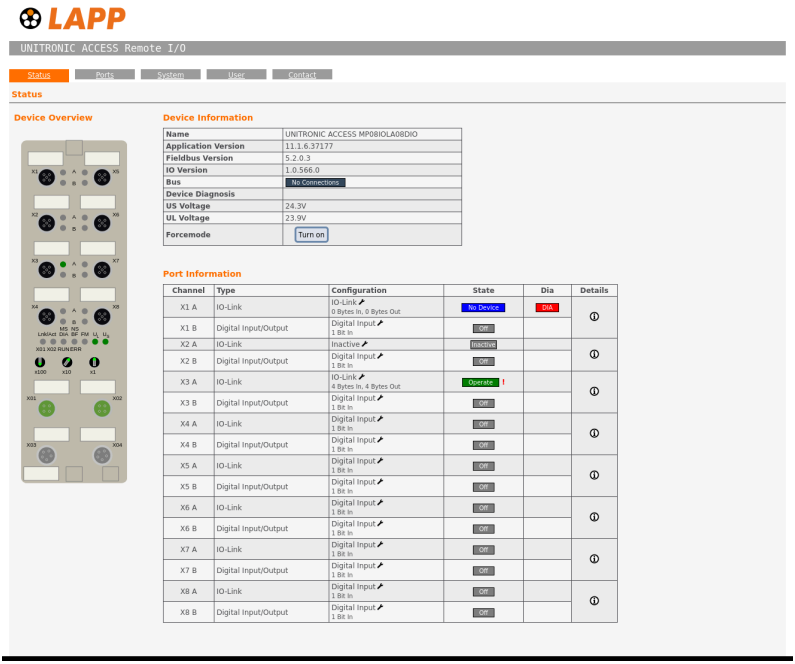
All device variants are equipped with an integrated Web server which makes functions for the device configuration and the display of status and diagnostic information available via a Web interface.

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

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

15.1 UNITRONIC® ACCESS MP08... variants

15.1.1 The Status page



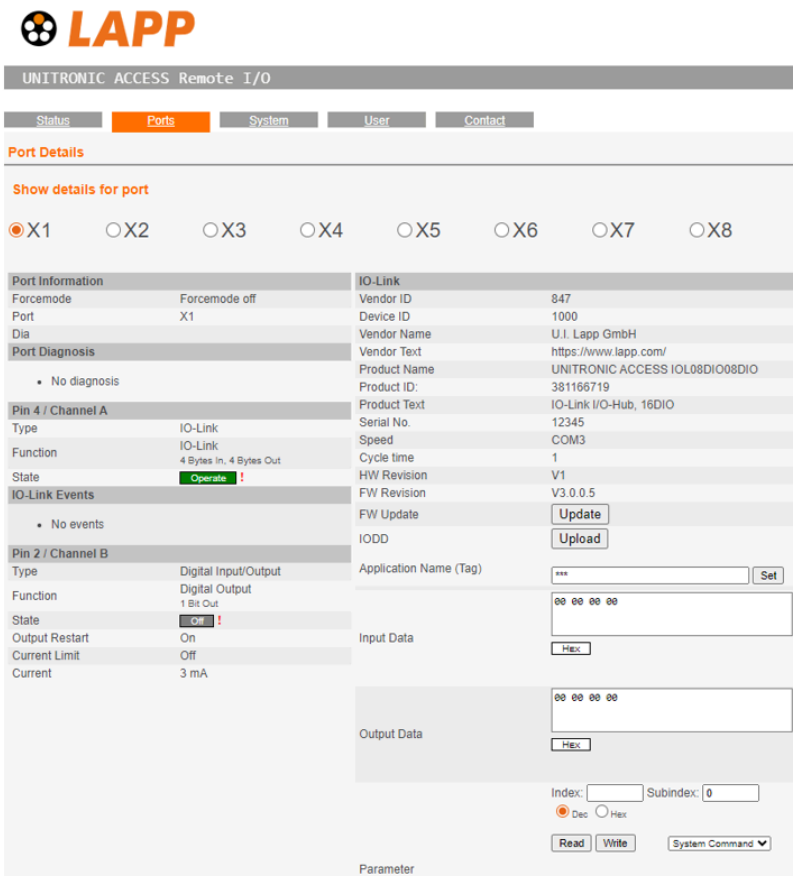
The status page provides a quick overview of the current state of the device.

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.

15.1.2 The Ports page



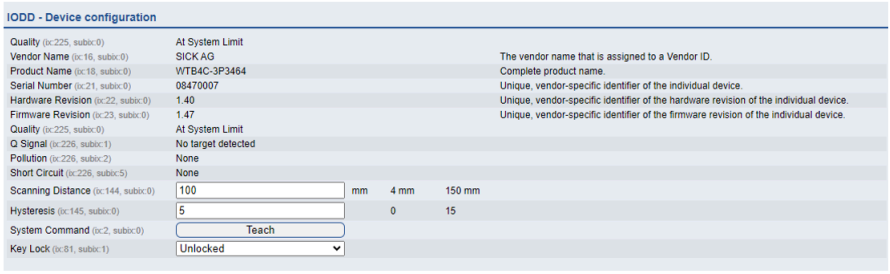
The page shows detailed port information. In the field **Port Diagnosis**, incoming and outgoing diagnostics are displayed as clear text. **Pin 2** and **Pin 4** contain information about the configuration and state of the port. For IO-Link ports, additional information relating to the connected sensor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

15.1.2.1 IODD upload

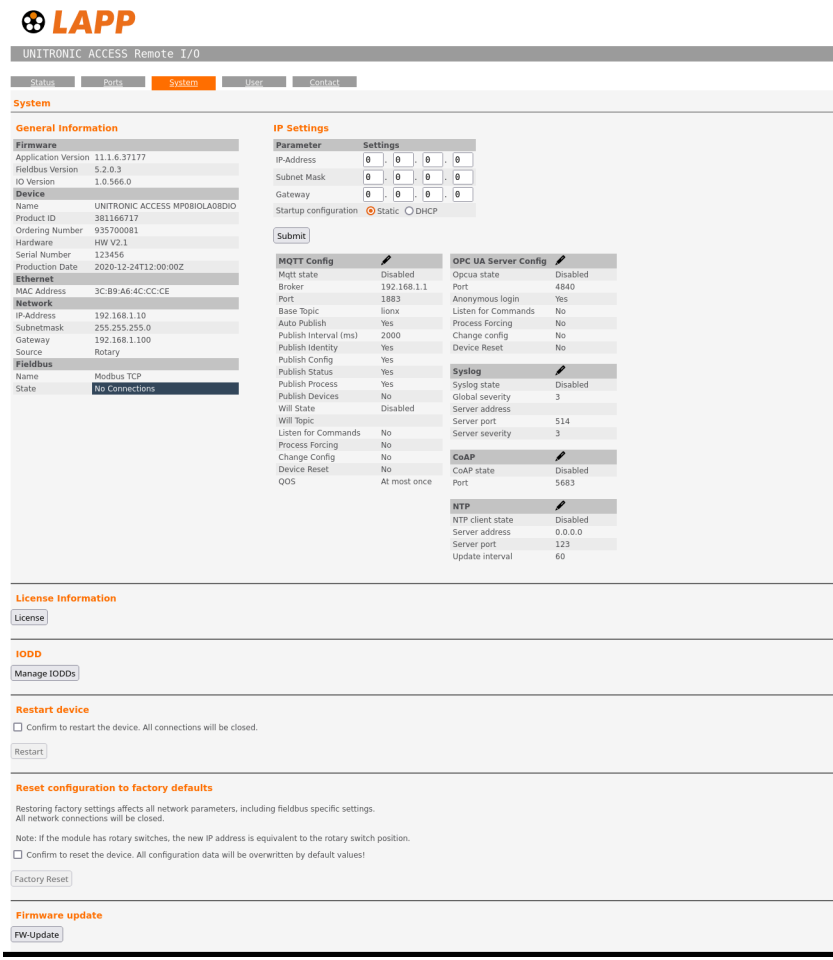
The button **UPLOAD** allows uploading an IODD file into the module, regardless of the device for which the IODD is designed.

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

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



15.1.3 The System page



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

IP Settings

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

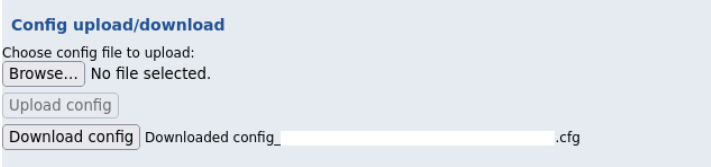
For PROFINET, this is only useful during commissioning. Normally, the PLC sets the IP address at start-up by detecting the PROFINET module via its device name.

15.1.3.1 License

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

15.1.3.2 Config upload/download

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



The following settings will be stored inside this file:

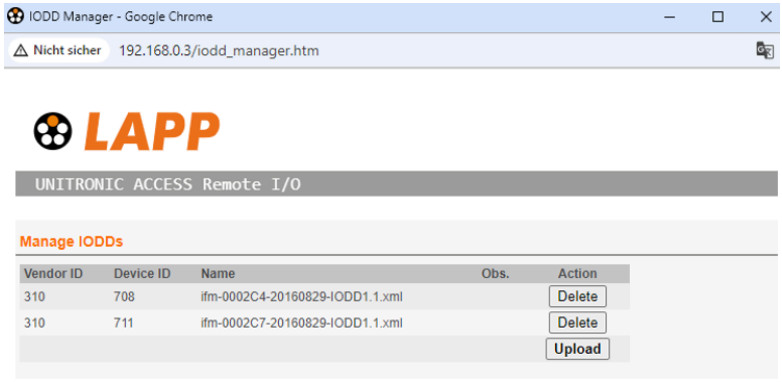
Scope	Type	Setting	Options	Details
Gateway		deviceID		To check device identity.
	iol	applicationSpecificTag		
	iol	functionTag		
	iol	locationTag		
		forcing		Enable/disable forcing
		channel_count		
		network configuration	ip	
			snMask	
			gw	
			source	1 - manual 2 - dhcp 3 - rotary 4 - dcp
Channel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

Scope	Type	Setting	Options	Details
	digital	force value		
	digital	simulate		
	digital	sim value		
	digital	inputPolarity		
	digital	autorestart mode		
	digital	inputFilter100us		
	digital	currentLimit		
	digital	outputRestartMode		
	digital	failsafeMode		
	digital	surveillanceTimeouMs		
OPC UA		opcua	opcua-enable	
			port	
			anon-allowed	
			commands-allowed	
			force-allowed	
			reset-allowed	
			config-allowed	
	digital		dcu-allowed	
MQTT		mqtt	mqtt-enable	
			broker	
			login	
			password	
			port	
			base-topic	
			will-enable	
			will-topic	
			auto-publish	
			publish-interval	
			publish-identity	

Scope	Type	Setting	Options	Details
			publish-config	
			publish-status	
			publish-process	
	iol		publish-devices	
			commands-allowed	
			force-allowed	
			reset-allowed	
			config-allowed	
			qos	
SYSLOG		syslog	syslog-enable	
			global-severity	
			server-address	
			server-port	
			server-severity	
COAP		coap	enable	
			port	
NTP		ntpc	enable	
			server-address	
			server-port	
			update-interval	

15.1.3.3 IODD

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



15.1.3.4 Restart device

The module initializes a software reset.

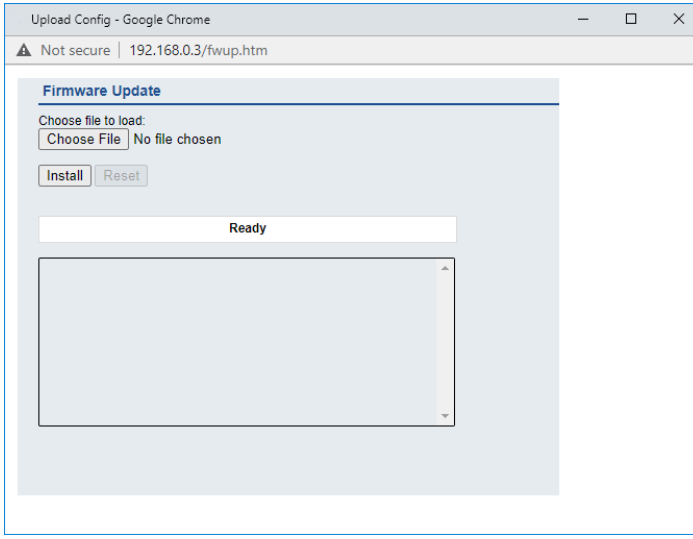
15.1.3.5 Reset configuration to factory defaults

The module restores to the default factory settings.

15.1.3.6 Firmware update

The module initializes a Firmware update.

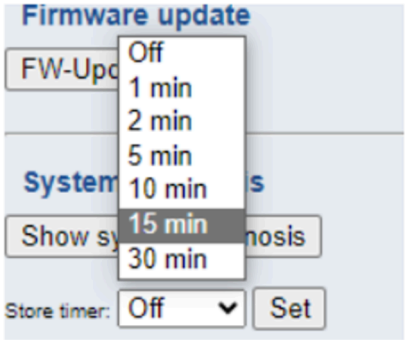
For a firmware update choose the **.ZIP* container available on our website or ask our support team. Afterwards follow the instructions shown on your screen.



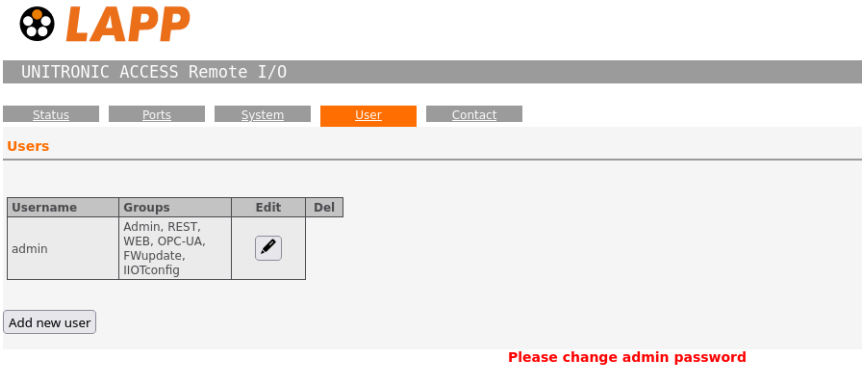
15.1.3.7 System diagnosis

All Syslog messages will be displayed in a ring buffer with 512 entries. By activation of the 'Store timer', the buffer content will be stored nonvolatile in the selected interval of 1, 2, 5, 10, 15 or 30 minutes.

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



15.1.4 The User page



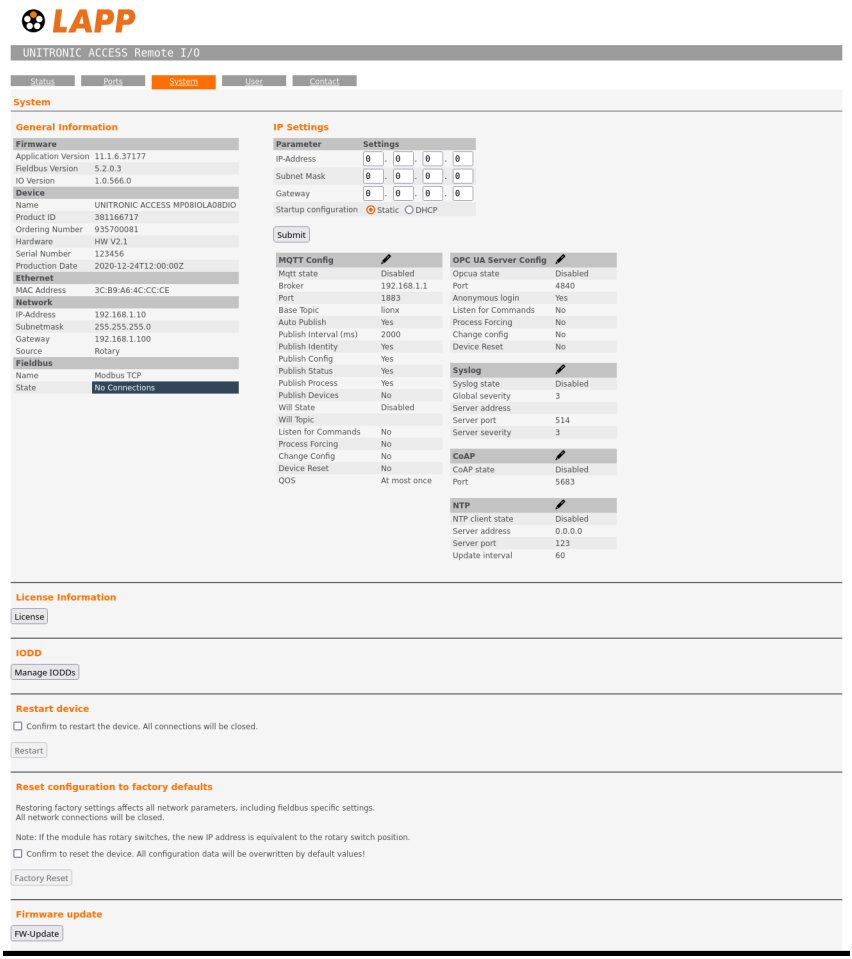
The User page provides the user management of the Web interface. New users with access rights "Admin" or "Write" can be added here. For security reasons please change the default admin password immediately after configuring the device.

Default user login data:

- User: admin
- Password: private

15.2 UNITRONIC® ACCESS single-protocol PN0.../EIP0.../EC0.../MTCP0.../CCL0... variants

15.2.1 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

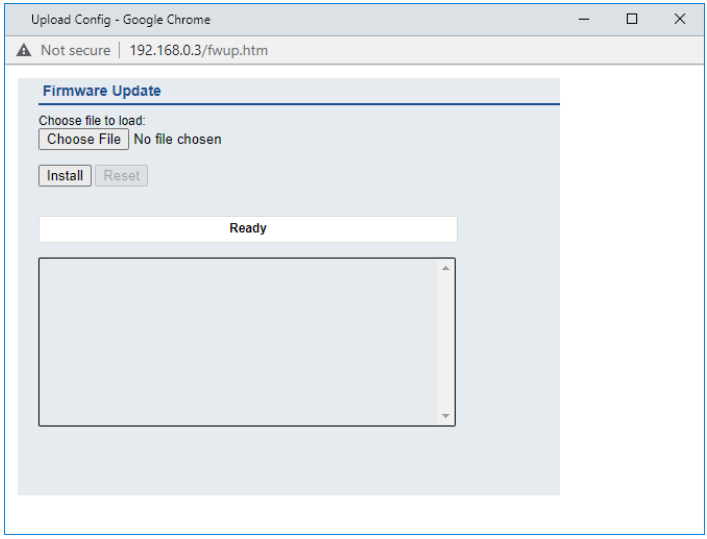
Use this parameter to change the current IP address of the module.

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.



16 IODD

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

- MP08IOLA08DIO

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

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

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

16.1 IO-Link Device parameters and ISDU requests

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

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

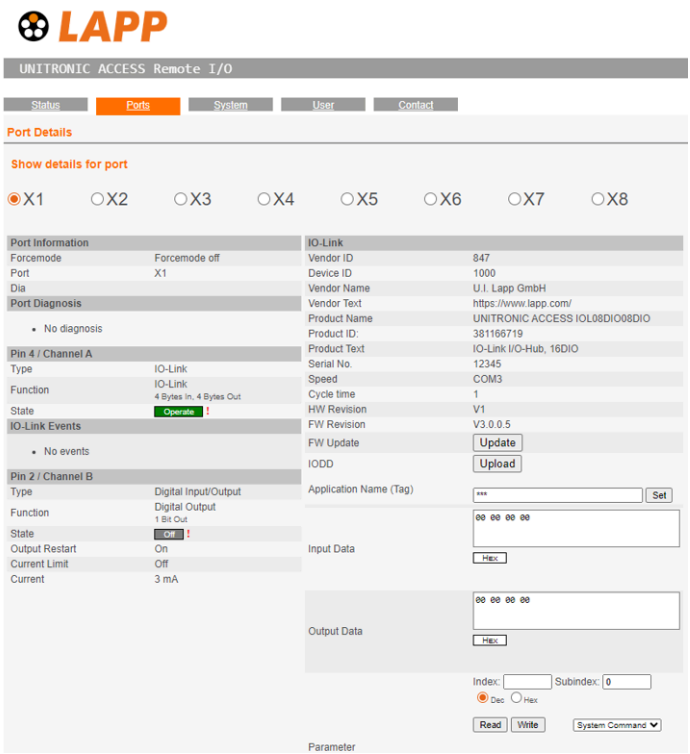
A vendor can use additional parameters and therefore more indices for their devices in order to provide additional configuration options. These vendor specific parameters can be described in an IODD. The "IODD on Module" feature of the UNITRONIC® ACCESS IO-Link Masters can read and parse this information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

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

16.2 Web GUI functionality

All of the "IODD on Module" features are accessible via the UNITRONIC® ACCESS Web interface.

16.2.1 Port Details page



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

IODD buttons

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

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

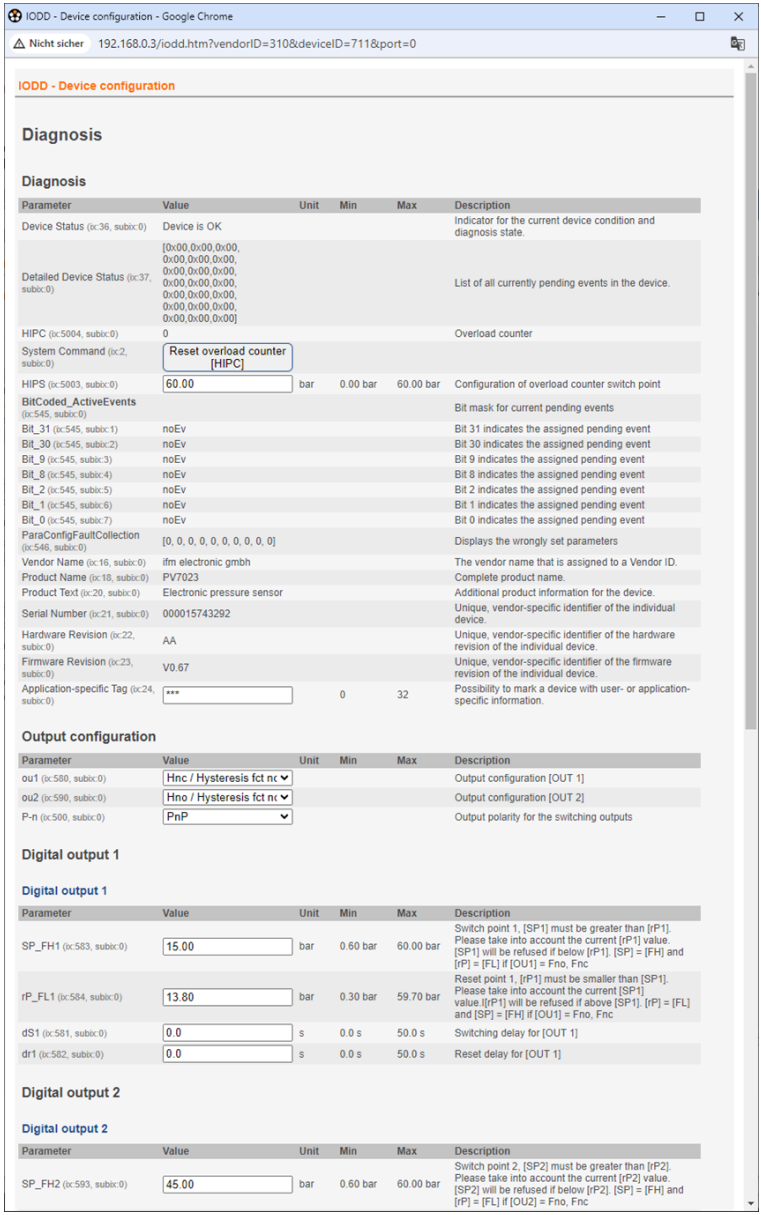
If there is a matching IODD for the currently connected device already stored in the system, the button *CONFIGURE* is shown in the interface. By clicking this button, the Parameter Page will open to configure the device.

Process data

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

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

16.2.2 Parameters page



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

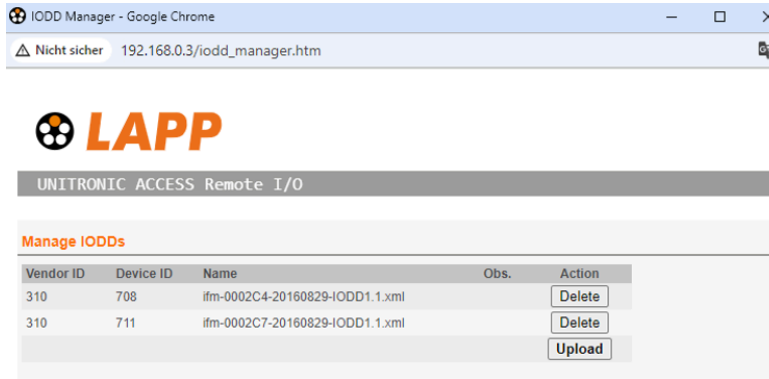
The stored IODD reads the parameter meta data, such as names, units, min/max values, descriptions etc. The values will be obtained directly from the connected device. For that reason it may take several seconds until the page is updated.

If not already saved into the browser, you will be asked for your credentials to continue. A valid user access with Web Interface group membership is needed in order to edit the device parameters. After the registration, enabled values can be changed. Disabled values cannot be changed and may be marked as "read-only" in the IODD. All values are directly written back to the device after any change.

Limitations

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

16.2.3 IODD Management page



The IODD Management Page can be accessed via the System page displaying all IODDs that are currently stored in the system. All IODDs

matching connected devices are marked. On the IODD Management page, you can manually delete any IODD in the system.

17 Technical data

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

17.1 General

Protection class (Only applies if the connectors are screwed together or if protective caps are used.)[2]	IP65 IP67 IP69K	
Ambient temperature (during operation and storage)	MP08...	-40 °C .. +70 °C (-40 °F .. +158 °F)
	PN08IOLA.../EIP08IOLA.../EC08IOLA.../ MTCP08IOLA.../CCL08IOLA...	-20 °C .. +60 °C (-4 °F .. +140 °F)
Weight	UNITRONIC® ACCESS 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 57: General information

2 Not under UL investigation.

17.2 EtherNet/IP protocol

Protocol	EtherNet/IP, CIP V3.27
Update cycle	1 ms
EDS file	EDS-V3.xx.x-LAPP-UNITRONIC-ACCESS-...08DIO-yyyymmdd.eds
Transmission rate	10/100 Mbit/s, half/full duplex
Transmission procedure Autonegotiation	10BASE-T/100BASE-TX supported
RPI min.	1 ms
Vendor ID	21
Product type	12 (Communications Adapter)
Product code	41001 (EIP08IOLA08DIO, 381166713) 41005 (MP08IOLA08DIO, 381166717)
Supported Ethernet protocols	Ping ARP- HTTP / HTTPS TCP/IP DHCP/BOOTP
Switch functionality	Integrated
EtherNet/IP interface Connections Autocrossing	2 M12 sockets, 4-pin, D-coded (see pin assignments) 2 M12 Hybrid male/female, 8-pin supported
Electrically isolated Ethernet ports -> FE	2000 V DC

Table 58: EtherNet/IP protocol

17.3 Power supply of the module electronics/ sensors

Port X03, X04	M12-L-coded Power, connector/socket, 5-pole Pin 1 / Pin 3		
Nominal voltage U_S	24 V DC (SELV/PELV)		
Current U_S	Max. 16 A		
Voltage range	21 .. 30 V DC		
Power consumption of module electronics	Typically 160 mA (+/-20 % at U_S nominal voltage)		
Power supply interruption	Max. 10 ms		
Voltage ripple U_S	Max. 5 %		
Current consumption sensor system (L+ / Pin 1)	MP08...	Port X1 .. X8 (Pin 1)	max. 4 A per port (at $T_{ambient}$ = 30° C)
	PN08IOLA.../ EIP08IOLA.../ EC08IOLA.../ MTCP08IOLA.../ CCL08IOLA...	Port X1 .. X8 (Pin 1)	max. 2 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 59: Information on the power supply of the module electronics/ sensors

17.4 Power supply of the actuators

17.4.1 IO-Link Class A devices (U_L)

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

Table 60: Information on the power supply of the actuators

17.4.2 IO-Link Class A/B devices (U_{AUX})

Nominal voltage U_{AUX}	24 V DC (SELV/PELV)
Voltage range	18 .. 30 V DC
Current U_{AUX}	Max. 16 A
Voltage ripple U_{AUX}	Max. 5 %
Reverse polarity protection	Yes
Electric isolation $U_S \leftrightarrow U_{AUX}$	500 V
Operational indicator (U_{AUX})	LED green: 18 V (+/- 1 V) < U_{AUX} LED red: U_{AUX} < 18 V (+/- 1 V) or U_{AUX} > 30 V (+/- 1 V) * if "Report U_{AUX} supply voltage fault" is enabled.
Port X03, X04	M12_L-coded Power, connector/socket, 5-pole Pin 2 / Pin 4

Table 61: Information on the power supply of the actuators

17.5 I/O ports Channel A (Pin 4)

MP08...	Port X1 .. X8	Class A	IOL, DI, DO	M12 socket, 5-pin, Pin 4
PN08IOLA.../EIP08IOLA.../ EC08IOLA.../ MTCP08IOLA.../ CCL08IOLA...	Port X1 .. X8	Class A	IOL, DI, DO	

Table 62: IO-Link Master ports: Functional overview for Ch. A (Pin 4)

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

Input connection	MP08... PN08IOLA.../EIP08IOLA.../EC08IOLA.../ MTCP08IOLA.../CCL08IOLA...		Type 1 as per IEC 61131-2
Nominal input voltage	24 V DC		
Input current	Typically 3 mA		
Channel type	Normally open, p-switching		
Number of digital inputs	MP08... PN08IOLA.../EIP08IOLA.../ EC08IOLA.../MTCP08IOLA.../ CCL08IOLA...	X1 .. X8	8
Status indicator	yellow LED		
Diagnostic indicator	red LED per channel		

Table 63: I/O ports Ch. A (Pin 4) configured as digital inputs

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

Attention: For variant MP08IOLA08DIO, the digital outputs of Channel A is **supplied by the U_L power** when parameterized to "High-Side Switch" mode.

Attention: For variant PN08IOLA08DIO, the digital outputs of Channel A is **supplied by the U_S power**.

Output type	normally open, p-switching (parameterized to "High-Side Switch" mode)	
Nominal output voltage per channel Signal status “1” Signal status “0”	 min. (U_S -1 V) **or** min. (U_L -1 V) depending on the device variant max. 2 V	
Max. output current per device	MP08...	9 A (power supplied via U_L)
Max. output current per device	PN08IOLA.../EIP08IOLA.../ EC08IOLA.../MTCP08IOLA.../ CCL08IOLA...	4 A (power supplied via U_S)
Max. output current per channel[3]	MP08... (X1 .. X8)	2 A (power supplied via U_S)
	PN08IOLA.../EIP08IOLA.../ EC08IOLA.../MTCP08IOLA.../ CCL08IOLA... (X1 .. X8)	0.5 A (power supplied via U_S) 0.25 A for **UL applications**
Short-circuit/overload protected	yes/yes	
Behavior in case of short circuit or overload	deactivation with automatic power-on (parameterized)	
Number of digital outputs	MP08... (X1 .. X8) PN08IOLA.../EIP08IOLA.../ EC08IOLA.../MTCP08IOLA.../ CCL08IOLA... (X1 .. X8)	8
Status indicator	yellow LED per output	
Diagnostic indicator	red LED per channel	

Table 64: I/O ports Ch. A (Pin 4) configured as digital outputs

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

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

IO-Link Master specification	v1.1.3, IEC 61131-9	
Communication rates	4.8 kbaud (COM 1) 38.4 kbaud (COM 2) 230.4 kbaud (COM 3)	
Line lengths in the IO-Link Device	max. 20 m	
Number of IO-Link ports	MP08... (X1 .. X8) PN08IOLA.../EIP08IOLA.../EC08IOLA.../ MTCP08IOLA.../CCL08IOLA... (X1 .. X8)	8
Min. IO-Link cycle time	400 µs	

Table 65: Configured as IO-Link port in COM mode

17.6 I/O ports Channel B (Pin 2)

MP08...	Port X1 .. X8	Class A	DI, DO	M12 socket, 5-pin, Pin 2
PN08IOLA.../EIP08IOLA.../ EC08IOLA.../ MTCP08IOLA.../ CCL08IOLA...	Port X1 .. X8	Class A	DI	

Table 66: IO-Link Master ports: Functional overview for Ch. B (Pin 2)

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

Input connection	MP08... PN08IOLA.../EIP08IOLA.../EC08IOLA.../ MTCP08IOLA.../CCL08IOLA...		Type 1 as per IEC 61131-2
Nominal input voltage	24 V DC		
Input current	Typically 3 mA		
Channel type	Normally open, p-switching		
Number of digital inputs	MP08...	X1 .. X8	8
Number of digital inputs	PN08IOLA.../EIP08IOLA.../ EC08IOLA.../MTCP08IOLA.../ CCL08IOLA...	X1 .. X8	8
Status indicator	white LED		
Diagnostic indicator	red LED per channel		

Table 67: I/O ports Ch. B (Pin 2) configured as digital inputs

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

Attention: For variant MP08IOLA08DIO, the digital outputs of Channel B is **supplied by the U_L power**.

Attention: For variant PN08IOLA08DIO, the digital outputs of Channel B is **supplied by the U_S power**.

Output type	normally open, p-switching	
Nominal output voltage per channel Signal status “1” Signal status “0”	min. (U_S -1 V) **or** min. (U_L -1 V) **or** min. (U_{AUX} -1 V) depending on the device variant max. 2 V	
Max. output current per device	MP08...	9 A (power supplied via U_L)
Max. output current per device	PN08IOLA.../ EIP08IOLA.../ EC08IOLA.../ MTCP08IOLA.../ CCL08IOLA...	4 A (power supplied via U_S)
Max. output current per channel [4]	MP08...	2 A (power supplied via U_S)
Max. output current per channel [4]	PN08IOLA.../ EIP08IOLA.../ EC08IOLA.../ MTCP08IOLA.../ CCL08IOLA...	0 A (no outputs on Ch. B)
Short-circuit/overload protected	yes/yes	
Behavior in case of short circuit or overload	deactivation with automatic power-on (parameterized)	
Number of digital outputs	MP08...	8
Number of digital outputs	PN08IOLA.../ EIP08IOLA.../ EC08IOLA.../ MTCP08IOLA.../ CCL08IOLA...	–
Status indicator	white LED per output	
Diagnostic indicator	red LED per channel	

Table 68: I/O ports Ch. B (Pin 2) configured as digital outputs

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

17.7 LEDs

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

LED	Color	Description
BF	Red	Bus fault. No configuration, no or slow physical connection.
	Red flashing at 2 Hz	Link exists but no communication link to the EtherNet/IP controller.
	OFF	EtherNet/IP controller has established an active connection to the device.
DIA	Red	EtherNet/IP module diagnostic alarm active.
	Red flashing at 1 Hz	Watchdog time-out; fail safe mode is active.
	Red flashing at 2 Hz, 3 sec	DCP signal service is initiated via the bus.
	Red double flash	Firmware update
	OFF	None of the above conditions.
MS	Green	Device is ready for operation.
	Green flashing	Device is ready but not configured yet.
	Red	Serious error that cannot be resolved.
	Red flashing	Minor error that can be resolved Example: An incorrect or contradictory configuration is classified as a minor error.
	Flashing alternately: Red / Green	The device is performing a self-test.
	OFF	The device is switched off.

LED	Color	Description
NS	Green	Connected: The device has at least one connection.
	Green flashing	No connection: The device has no connection. IP address exists.
	Red	Duplicate IP address: The device has detected that the assigned IP address is already being used by another device.
	Red flashing	Connection has exceeded time limit or connection interrupted.
	Flashing alternately: Red / Green	The device is performing a self-test.
	OFF	The device is switched off or has not been assigned an IP address.

Table 69: Information on the LED colors

17.8 Data transfer times

The following tables give an overview of the internal data transfer times of the UNITRONIC® ACCESS IO-Link Master with a connected IO-Link Device as digital I/O extension (LAPP article IOL08DIO08DIO 16DIO Hub with a minimum cycle time of 1 ms).

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

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

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

The configurable digital input filter value on IOL08DIO08DIO was set to "off" (0 ms).

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

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

Use case 1:

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

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

Use case 2:

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

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

18 Accessories

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

https://www.lapp.com

Manual

EtherCAT®

UNITRONIC ACCESS IO-Link Master

Multi-protocol:
MP08IOLA08DIO (8 × IO-Link Class A)

Single-protocol:
EC08IOLA08DIO (8 × IO-Link Class A)



Contents

1 About this manual 8

1.1 General information 8
1.2 Explanation of symbols 9
1.2.1 Use of danger information 9
1.2.2 Use of general information 9
1.2.3 EtherCAT® trademark information 9
1.3 Version information 10

2 Safety instructions 11

2.1 Intended use 11
2.2 Qualified personnel 12

3 Designations and synonyms 13

4 System description 17

4.1 Designations and synonyms 18
4.2 About UNITRONIC® ACCESS 21
4.3 Device variants 23
4.4 I/O port overview 25

5 Overview of product features 27

5.1 EtherCAT® product features 27
5.2 I/O port features 28
5.3 Integrated Web server 29

5.4 Security features 30
5.5 Other features 31

6 Assembly and wiring 32

6.1 General information 32
6.2 Outer dimensions 33
6.2.1 UNITRONIC® ACCESS multi-protocol variants 33
6.2.2 UNITRONIC® ACCESS single-protocol variants with EtherCAT® 34
6.2.3 Notifications 35
6.3 Port assignments 36
6.3.1 Ethernet ports, M12 socket, 4-pin, D-coded 36
6.3.2 Power supply with M12 power L-coded 37
6.3.2.1 IO-Link Master with Class A ports 38
6.3.2.2 IO-Link Master with Class A/B ports 39
6.3.3 I/O ports as M12 sockets 39
6.3.3.1 IO-Link ports (Class A) 40

7 Starting operation 41

7.1 ESI file 41
7.2 MAC addresses 42
7.3 Setting the rotary encoding switches 43
7.3.1 Factory reset 46

8 Configuration and operation in TwinCAT® 3 47

8.1 PDO assignments 47
8.1.1 Input data 47
8.1.2 Output data 53
8.1.3 Modular slots 57
8.2 Device parameters 60
8.2.1 Extended parameters 60

8.2.2 Failsafe replacement values 61
8.2.3 Failsafe mode for the digital output 62
8.2.4 General device settings 65
8.2.5 Surveillance timeout 66
8.2.6 Digital I/O mode, Channel B 68
8.2.7 Digital Input logic 70
8.2.8 Digital Input filter 72
8.2.9 Digital Output restart 74
8.2.10 Digital Input latch 76
8.2.11 Digital Input extension 80
8.2.12 Additional IO-Link port settings 82
8.2.13 IO-Link Parameterization 84
8.2.14 IO-Link configuration data 85
8.2.15 IO-Link serial number 86
8.2.16 IO-Link information data 87
8.2.17 IO-Link serial number of connected devices 88
8.3 Configuration example with TwinCAT® 3 89
8.3.1 Configuration of MP08... devices 91
8.3.2 EoE IP address 93
8.3.3 Activate configuration 95

9 Diagnostics processing 96

9.1 Diagnostics (0x2001) 96
9.2 Error of the system/sensor power supply 96
9.3 Error of the auxiliary/actuator power supply 97
9.4 Overload/short-circuit of the I/O port sensor supply outputs 97
9.5 Overload/short circuit of the digital outputs 97
9.6 IO-Link C/Q error 98
9.7 Diagnostic history object (0x10F3) 98
9.7.1 Maximum Messages 98
9.7.2 Newest Message 98
9.7.3 Newest Acknowledge Message 98
9.7.4 New Messages Available 100
9.7.5 Flags 100

9.7.6 Diagnosis Message Buffer 101
9.8 Diagnostic data object (0xA0n0) 102
9.9 Device status object (0xF100) 103
9.10 Emergency messages 105

10 IIoT functionality 106

10.1 MQTT 107
10.1.1 MQTT configuration 107
10.1.2 MQTT topics 110
10.1.2.1 Base topic 110
10.1.2.2 Publish topic 113
10.1.2.3 Command topic (MQTT Subscribe) 118
10.1.3 MQTT configuration - Quick start guide 122
10.1.3.1 MQTT configuration via JSON 122
10.2 OPC UA 124
10.2.1 OPC UA configuration 125
10.2.2 OPC UA address space 127
10.2.3 OPC UA configuration - Quick start guide 128
10.2.3.1 OPC UA configuration via JSON 128
10.3 REST API 130
10.3.1 Standard device information 132
10.3.2 Structure 133
10.3.3 Configuration and forcing 139
10.3.4 Reading and writing ISDU parameters 142
10.3.4.1 Reading ISDU 142
10.3.4.2 Writing ISDU 144
10.3.5 Upload and process an IODD file 146
10.3.6 Example: Reading ISDU 150
10.3.7 Example: Writing ISDU 150
10.4 CoAP server 151
10.4.1 CoAP configuration 151
10.4.2 REST API access via CoAP 152
10.4.3 CoAP configuration - Quick start guide 155
10.4.3.1 CoAP configuration via JSON 155

10.5 Syslog 157
10.5.1 Syslog configuration 157
10.5.2 Syslog configuration - Quick start guide 160
10.5.2.1 Syslog configuration via JSON 160
10.6 Network Time Protocol (NTP) 162
10.6.1 NTP configuration 162
10.6.2 NTP configuration - Quick start guide 164
10.6.2.1 NTP configuration via JSON 164

11 The integrated Web server 166

11.1 UNITRONIC® ACCESS MP08... variants 167
11.1.1 The Status page 167
11.1.2 The Ports page 168
11.1.2.1 IODD upload 168
11.1.3 The System page 170
11.1.3.1 License 171
11.1.3.2 Config upload/download 171
11.1.3.3 IODD 174
11.1.3.4 Restart device 175
11.1.3.5 Reset configuration to factory defaults 175
11.1.3.6 Firmware update 175
11.1.3.7 System diagnosis 176
11.1.4 The User page 177
11.2 UNITRONIC® ACCESS single-protocol PN0.../EIP0.../EC0.../MTCP0.../CCL0... variants 178
11.2.1 The System page 178

12 IODD 180

12.1 IO-Link Device parameters and ISDU requests 180
12.2 Web GUI functionality 181
12.2.1 Port Details page 181
12.2.2 Parameters page 183

12.2.3 IODD Management page 184

13 Firmware update 186

13.1 Firmware update via FoE 186

14 Technical data 190

14.1 General 191
14.2 EtherCAT® protocol 192
14.3 Power supply of the module electronics/sensors 193
14.4 Power supply of the actuators 194
14.4.1 IO-Link Class A devices (U_L) 194
14.4.2 IO-Link Class A/B devices (U_{AUX}) 194
14.5 I/O ports Channel A (Pin 4) 195
14.5.1 Configured as digital input, Ch. A (Pin 4) 195
14.5.2 Configured as digital output, Ch. A (Pin 4) 196
14.5.3 Configured as IO-Link port in COM mode, Ch. A 198
14.6 I/O ports Channel B (Pin 2) 199
14.6.1 Configured as a digital input, Ch. B (Pin 2) 199
14.6.2 Configured as a digital output, Ch. B (Pin 2) 200
14.7 LEDs 202
14.8 Data transfer times 205

15 Accessories 207

16 References 208

1 About this manual

1.1 General information

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

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

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

U.I. Lapp GmbH

Schulze-Delitzsch-Straße 25
D-70565 Stuttgart
Germany

mailto:info@lapp.com
https://www.lapp.com

U.I. Lapp GmbH reserves the right to make technical changes or changes to this document 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	10/2024	

Table 1: Overview of manual revisions

2 Safety instructions

2.1 Intended use

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

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

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

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

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

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

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

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

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

Information on which cables and accessories are permitted for the installation can be obtained from U.I. Lapp GmbH or is contained in this manual.

2.2 Qualified personnel

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

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

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

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

Only U.I. Lapp GmbH is permitted to make changes to the hardware or software of the products that go beyond the scope of this manual.



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



Attention: U.I. Lapp GmbH 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 and IO-Link)
BUI	Back-Up Inconsistency (EIP diagnostics)
CC	CC-Link IE Field
C/Q	I/O port pin 4 mode, IO-Link communication/switching signal
Ch. A	Channel A (Pin 4) of I/O port
Ch. B	Channel B (Pin 2) of I/O port
CIP	Common Industrial Protocol (media independent protocol)
CIP Safety™	Common Industrial Protocol for Safety applications, CIP Safety™ is a registered trademark of ODVA
Class A	IO-Link port specification (Class A)
Class B	IO-Link port specification (Class B)
CoAP	Constrained Application Protocol
CSP+	Control & Communication System Profile Plus
DAT	Device Acknowledgement Time
DCP	Discovery and Configuration Protocol
DevCom	Device Comunicating (EIP diagnostics)
DevErr	Device Error (EIP diagnostics)
DI	Digital Input
DIA	Diagnostic LED
DO	Digital Output
DIO	Digital Input/Output
DTO	Device Temperature Overrun (EIP diagnostics)
DTU	Device Temperature Underrun (EIP diagnostics)
DUT	Device under test

EIP	EtherNet/IP™ is a registered trademark of ODVA
ERP	Enterprise Resource Planning system
ETH	ETHERNET
FE	Functional Earth
FME	Force Mode Enabled (EIP diagnostics)
FS	Functional Safety
FSU	Fast Start-Up
GSDML	General Station Description Markup Language
High-B	High-Byte
HTTPS	Hyper Text Transfer Protocol Secure
ICE	IO-Link port COM Error (EIP diagnostics)
ICT	Invalid Cycle Time (EIP diagnostics)
IDE	IO-Link port Device Error (EIP diagnostics)
IDN	IO-Link port Device Notification (EIP diagnostics)
IDW	IO-Link port Device Warning (EIP diagnostics)
IIoT	Industrial Internet of Things
ILE	Input process data Length Error (EIP diagnostics)
IME	Internal Module Error (EIP diagnostics)
I/O	Input / Output
I/O port	X1 .. X8
I/O port pin 2	Channel B of I/O ports
I/O port pin 4 (C/Q)	Channel A of I/O ports
IODD	I/O Device Description
IOL or IO-L	IO-Link
I/Q	I/O port pin 2 mode, Digital Input/switching signal
ISDU	Indexed Service Data Unit
IVE	IO-Link port Validation Error (EIP diagnostics)
I&M	Identification & Maintenance
JSON	JavaScript Object Notation (platform independent data format)
L+	I/O port pin 1, sensor power supply

UNITRONIC® ACCESS 60	UNITRONIC® ACCESS variants with a width of 60mm
Little Endian	Data format with Low-B on first place (EtherNet/IP)
LLDP	Link Layer Discovery Protocol
Low-B	Low-Byte
LSB	Least Significant Bit
LVA	Low Voltage Actuator Supply (EIP diagnostics)
LVS	Low Voltage System/Sensor Supply (EIP diagnostics)
MIB	Management Information Base
MP	Multi-protocol: PROFINET + EtherNet/IP + EtherCAT® + Modbus TCP (+ CC-Link IE Field Basic)
MQTT	Message Queuing Telemetry Transport (open networking protocol)
MSB	Most Significant Bit
M12	Metric thread according to DIN 13-1 with 12 mm diameter
NTP	Network Time Protocol
OFDT	One Fault Delay Time
OLE	Output process data Length Error (EIP diagnostics)
OPC UA	Open Platform Communications Unified Architecture (platform independent, service-oriented architecture)
PFH	Probability of dangerous Failure per Hour [h -1]
PD	Process Data
PDCT	Port and Device Configuration Tool
PLC	Programmable Logic Controller
PN	PROFINET
PWR	Power
Qualifier	Validity on a process value. Valid = "1"
REST	REpresentational State Transfer
RFC	Request for Comments
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)
SFRT	Safety Function Response Time
SIO mode	Standard Input Output mode
SLMP	Seamless Message Protocol
SNMP	Simple Network Management Protocol
SP	Single Protocol (PROFINET, EtherNet/IP, EtherCAT®, Modbus TCP or CC-Link IE Field Basic)
SPE	Startup Parameterization Error (EIP diagnostics)
T-B	Test Channel B
T-A	Test Channel A
U_{AUX}	$U_{Auxiliary}$, supply voltage for the load circuit (Actuator supply on Class B ports of Class A/B IO-Link Master)
UDP	User Datagram Protocol
UDT	User-Defined Data Types
UINT8	Byte in PLC (IB, QB)
UINT16	Unsigned integer with 16 bits or word in PLC (IW, QW)
U_L	U_{Load}, supply voltage for the load circuit (Actuator supply on Class A IO-Link-Master)
UL	Underwriters Laboratories Inc. (certification company)
UTC	Coordinated Universal Time (Temps Universel Coordonné)
WCDT	Worst Case Delay Time

Table 2: Designations and synonyms

4 System description

The Lapp U.I. UNITRONIC® ACCESS modules function as the interface in an industrial Ethernet system: A central controller on the management level is able to communicate with the decentralized sensors and actuators on the field level. The line or ring topologies for which UNITRONIC® 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 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 and IO-Link)
BUI	Back-Up Inconsistency (EIP diagnostics)
CC	CC-Link IE Field
C/Q	I/O port pin 4 mode, IO-Link communication/switching signal
Ch. A	Channel A (Pin 4) of I/O port
Ch. B	Channel B (Pin 2) of I/O port
CIP	Common Industrial Protocol (media independent protocol)
CIP Safety™	Common Industrial Protocol for Safety applications, CIP Safety™ is a registered trademark of ODVA
Class A	IO-Link port specification (Class A)
Class B	IO-Link port specification (Class B)
CoAP	Constrained Application Protocol
CSP+	Control & Communication System Profile Plus
DAT	Device Acknowledgement Time
DCP	Discovery and Configuration Protocol
DevCom	Device Comunicating (EIP diagnostics)
DevErr	Device Error (EIP diagnostics)
DI	Digital Input
DIA	Diagnostic LED
DO	Digital Output
DIO	Digital Input/Output
DTO	Device Temperature Overrun (EIP diagnostics)
DTU	Device Temperature Underrun (EIP diagnostics)
DUT	Device under test

EIP	EtherNet/IP™ is a registered trademark of ODVA
ERP	Enterprise Resource Planning system
ETH	ETHERNET
FE	Functional Earth
FME	Force Mode Enabled (EIP diagnostics)
FS	Functional Safety
FSU	Fast Start-Up
GSDML	General Station Description Markup Language
High-B	High-Byte
HTTPS	Hyper Text Transfer Protocol Secure
ICE	IO-Link port COM Error (EIP diagnostics)
ICT	Invalid Cycle Time (EIP diagnostics)
IDE	IO-Link port Device Error (EIP diagnostics)
IDN	IO-Link port Device Notification (EIP diagnostics)
IDW	IO-Link port Device Warning (EIP diagnostics)
IIoT	Industrial Internet of Things
ILE	Input process data Length Error (EIP diagnostics)
IME	Internal Module Error (EIP diagnostics)
I/O	Input / Output
I/O port	X1 .. X8
I/O port pin 2	Channel B of I/O ports
I/O port pin 4 (C/Q)	Channel A of I/O ports
IODD	I/O Device Description
IOL or IO-L	IO-Link
I/Q	I/O port pin 2 mode, Digital Input/switching signal
ISDU	Indexed Service Data Unit
IVE	IO-Link port Validation Error (EIP diagnostics)
I&M	Identification & Maintenance
JSON	JavaScript Object Notation (platform independent data format)
L+	I/O port pin 1, sensor power supply

UNITRONIC® ACCESS 60	UNITRONIC® ACCESS variants with a width of 60mm
Little Endian	Data format with Low-B on first place (EtherNet/IP)
LLDP	Link Layer Discovery Protocol
Low-B	Low-Byte
LSB	Least Significant Bit
LVA	Low Voltage Actuator Supply (EIP diagnostics)
LVS	Low Voltage System/Sensor Supply (EIP diagnostics)
MIB	Management Information Base
MP	Multi-protocol: PROFINET + EtherNet/IP + EtherCAT® + Modbus TCP (+ CC-Link IE Field Basic)
MQTT	Message Queuing Telemetry Transport (open networking protocol)
MSB	Most Significant Bit
M12	Metric thread according to DIN 13-1 with 12 mm diameter
NTP	Network Time Protocol
OFDT	One Fault Delay Time
OLE	Output process data Length Error (EIP diagnostics)
OPC UA	Open Platform Communications Unified Architecture (platform independent, service-oriented architecture)
PFH	Probability of dangerous Failure per Hour [h -1]
PD	Process Data
PDCT	Port and Device Configuration Tool
PLC	Programmable Logic Controller
PN	PROFINET
PWR	Power
Qualifier	Validity on a process value. Valid = "1"
REST	REpresentational State Transfer
RFC	Request for Comments
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)
SFRT	Safety Function Response Time
SIO mode	Standard Input Output mode
SLMP	Seamless Message Protocol
SNMP	Simple Network Management Protocol
SP	Single Protocol (PROFINET, EtherNet/IP, EtherCAT®, Modbus TCP or CC-Link IE Field Basic)
SPE	Startup Parameterization Error (EIP diagnostics)
T-B	Test Channel B
T-A	Test Channel A
U_{AUX}	$U_{Auxiliary}$, supply voltage for the load circuit (Actuator supply on Class B ports of Class A/B IO-Link Master)
UDP	User Datagram Protocol
UDT	User-Defined Data Types
UINT8	Byte in PLC (IB, QB)
UINT16	Unsigned integer with 16 bits or word in PLC (IW, QW)
U_L	U_{Load}, supply voltage for the load circuit (Actuator supply on Class A IO-Link-Master)
UL	Underwriters Laboratories Inc. (certification company)
UTC	Coordinated Universal Time (Temps Universel Coordonné)
WCDT	Worst Case Delay Time

Table 3: Designations and synonyms

4.2 About UNITRONIC® ACCESS

The UNITRONIC® ACCESS device variants convert standard input, standard output or IO-Link signals from sensors & actuators into an industrial Ethernet protocol (PROFINET, EtherNet/IP, EtherCAT®, Modbus TCP, CC-Link IE Field Basic) and/or into a cloud protocol (REST API, CoAP, OPC UA, MQTT). For the first time, there is now Syslog on board. The robust 8 port housing design

allows the use even in harsh environments where e.g. weld field immunity, high temperature ranges or protection class IP67 & IP69K are needed.

4.3 Device variants

The following variants are available in the UNITRONIC® ACCESS Multi-protocol and the UNITRONIC® ACCESS single-protocol family:

Article number	Product designation	Description	I/O port functionality
381166717	MP08IOLA08DIO	UNITRONIC® ACCESS M12-60 mm, IO-Link Master Multi-protocol (PN, EIP, EC, MB, CC) Security	8 x IO-Link Class A
381166712	PN08IOLA08DIO	UNITRONIC® ACCESS single-protocol M12-60 mm, IO-Link Master PROFINET	8 x IO-Link Class A
381166713	EIP08IOLA08DIO	UNITRONIC® ACCESS single-protocol M12-60 mm, IO-Link Master EtherNet/IP	8 x IO-Link Class A
381166714	EC08IOLA08DIO	UNITRONIC® ACCESS single-protocol M12-60 mm, IO-Link Master EtherCAT®	8 x IO-Link Class A
381166715	MTCP08IOLA08DIO	UNITRONIC® ACCESS single-protocol M12-60 mm, IO-Link Master Modbus TCP	8 x IO-Link Class A
381166716	CCL08IOLA08DIO	UNITRONIC® ACCESS single-protocol M12-60 mm, IO-Link Master CC-Link IE Field Basic	8 x IO-Link Class A

Table 4: Overview of UNITRONIC® ACCESS Multi-protocol and UNITRONIC® ACCESS single-protocol variants

4.4 I/O port overview

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

UNITRONIC® ACCESS Class A IO-Link ports

Device variant	Port	Pin 1 U_S	Pin 4 / Ch. A (C/Q)				Pin 2 / Ch. B (I/Q)	
MP08...	**Info:**	–	Class A	Type 1	Supply by U_S [1)]	Supply by U_L [2)]	Type 1	Supply by U_L [2)]
	X8:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)
	X7:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)
	X6:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)
	X5:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)
	X4:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)
	X3:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)
	X2:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)
	X1:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)

Table 5: Port configuration of MP08... variants

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

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

UNITRONIC® ACCESS single-protocol Class A IO-Link ports

Device variant	Port	Pin 1 U_S	Pin 4 / Ch. A (C/Q)			Pin 2 / Ch. B (I/Q)
PN08IOLA.../ EIP08IOLA.../ EC08IOLA.../ MTCP08IOLA.../ CCL08IOLA...	**Info:**	–	Class A	Type 1	Supply by U_S [1)]	Type 1
	X8:	Out (2 A)	IOL	DI	DO (0.5 A*)	DI
	X7:	Out (2 A)	IOL	DI	DO (0.5 A*)	DI
	X6:	Out (2 A)	IOL	DI	DO (0.5 A*)	DI
	X5:	Out (2 A)	IOL	DI	DO (0.5 A*)	DI
	X4:	Out (2 A)	IOL	DI	DO (0.5 A*)	DI
	X3:	Out (2 A)	IOL	DI	DO (0.5 A*)	DI
	X2:	Out (2 A)	IOL	DI	DO (0.5 A*)	DI
	X1:	Out (2 A)	IOL	DI	DO (0.5 A*)	DI

Table 6: Port configuration of PN08IOLA.../EIP08IOLA.../EC08IOLA.../ MTCP08IOLA.../CCL08IOLA... variants

[1)] With DO Switch Mode configured as "Push-Pull" (see description in the configuration chapters).

* For **UL applications**: Max. 0.25 A DO.

5 Overview of product features

5.1 EtherCAT® product features

Data connection

The connection option provided by UNITRONIC® ACCESS 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 with Conformance Class C (CC-C) 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 messages and alarms via a ring buffer based *Diagnosis History Object*.

5.2 I/O port features

IO-Link specification.

UNITRONIC® ACCESS is ready for IO-Link specification v1.1.3.

8 x IO-Link Master ports

Depending on the variant, the Master device has 4 IO-Link Class A ports, 4 IO-Link Class A ports and 4 IO-Link Class B ports, or 8 IO-Link Class A ports with an additional digital input and optional output on pin 2 of the I/O port. For detailed information see chapter I/O port overview on page 25.



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.

IO-Link port connections

The IO-Link port connection option provided by UNITRONIC® ACCESS devices is the 5-pin M12 connector. Pin 5 is not assigned for IO-Link Class A ports.

Validation & Backup

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

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

IO-Link Device parameterization

IO-Link Device parameterization in an EtherCAT® network is possible via TwinCAT®.

LED

You can see the status of a port by the color of the matching LEDs and their flash pattern. For details on the meanings of the LED colors, please see section LEDs on page 202.

5.3 Integrated Web server

Network parameter display

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

Displaying diagnostics

View diagnostics via the integrated Web server.

User management

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

IO-Link Device parameters

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

5.4 Security features

Firmware signature

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

Syslog

The UNITRONIC® ACCESS multi-protocol variants support the traceability of messages centrally managed and logged via Syslog.

User manager

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

Default user settings:

User: admin

Password: private

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

5.5 Other features

Interface protection

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

For more details, see section Port assignments on page 36.

Failsafe

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

Industrial Internet of Things

UNITRONIC® ACCESS 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 UNITRONIC® ACCESS 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 UNITRONIC® ACCESS multi-protocol variants

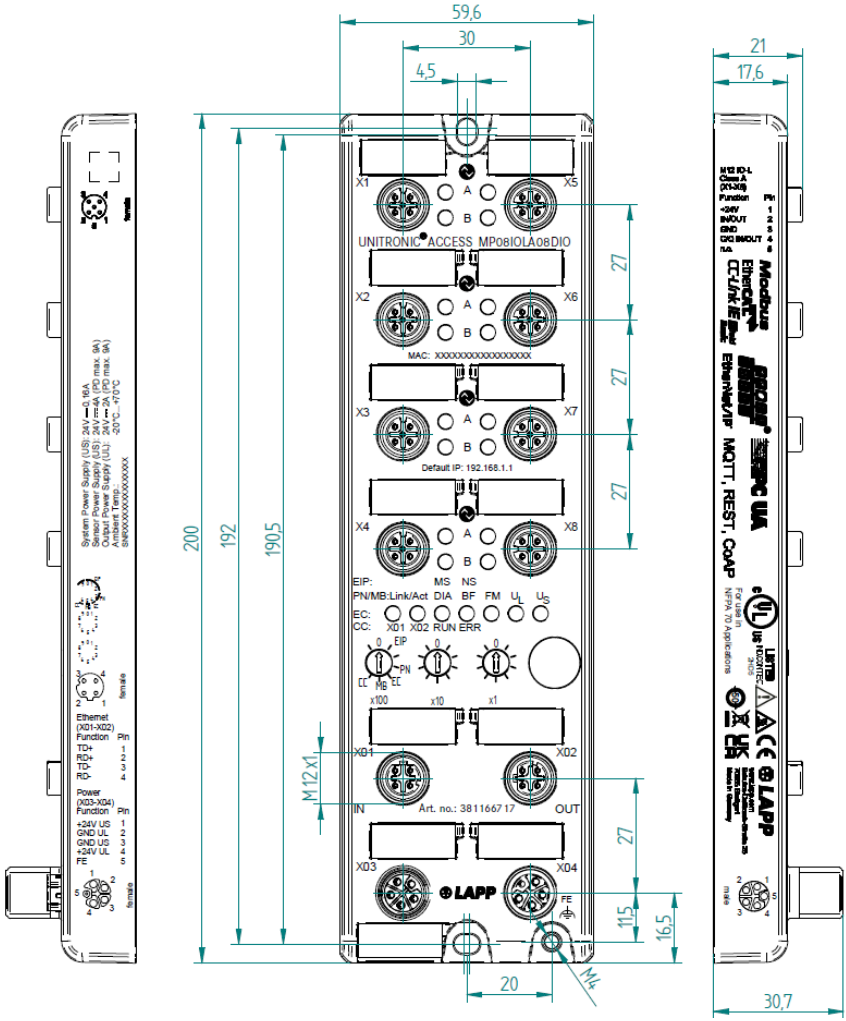


Figure 1: MP08IOLA08DIO

6.2.2 UNITRONIC® ACCESS single-protocol variants with EtherCAT®

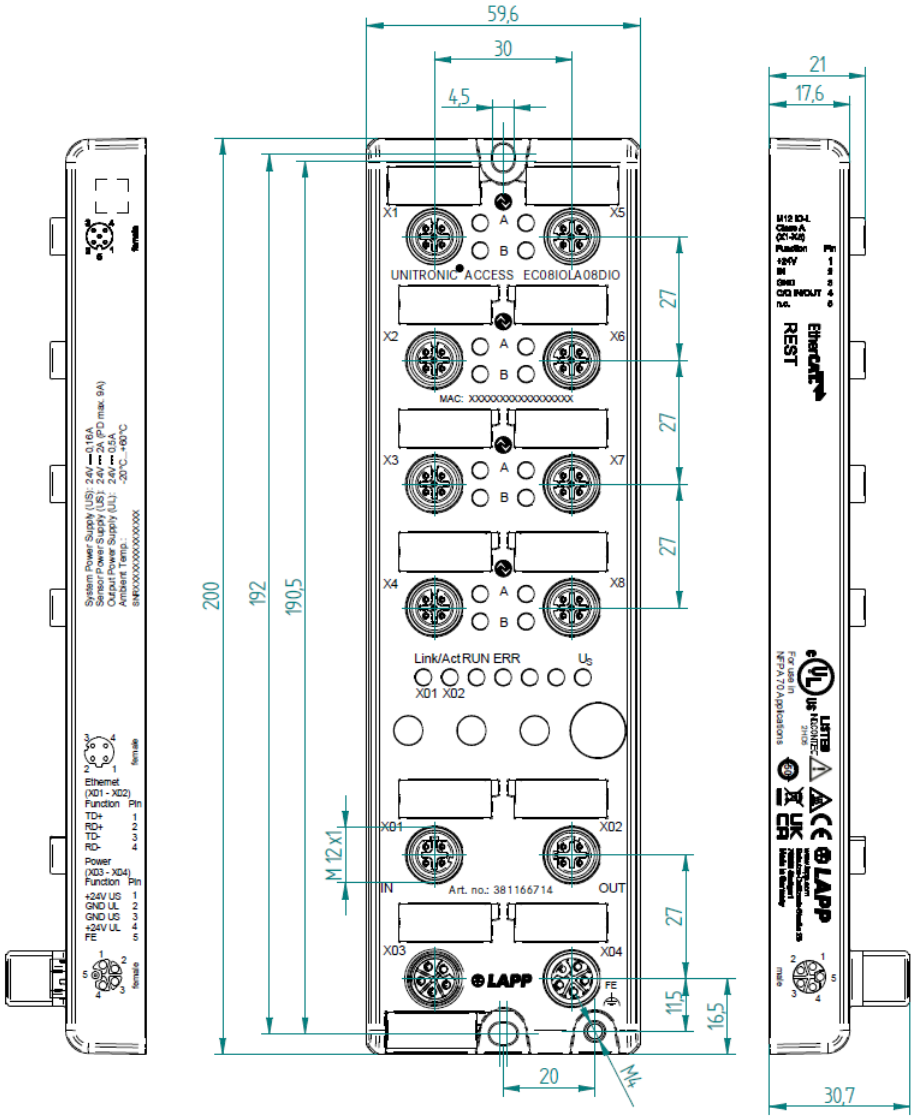


Figure 2: EC08IOLA08DIO

6.2.3 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 device variants.

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

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

6.3 Port assignments

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

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

Color coding: green

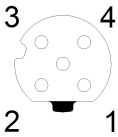


Figure 3: Schematic drawing, ports X01, X02

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 7: 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

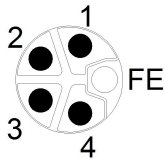


Figure 4: Schematic diagram of the M12 L-coding (connector X03 for Power In)

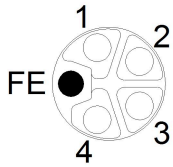


Figure 5: Schematic diagram of the M12 L-coding (socket X04 for Power Out)

6.3.2.1 IO-Link Master with Class A ports

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

Table 8: Power supply with M12-Power Class A

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

[1] U_L and U_S ground connected in device

6.3.2.2 IO-Link Master with Class A/B ports

Power supply	Pin	Signal	Function
Mixed IO-Link (Class A/B) I/O ports	1	U_S (+24 V)	Sensor/system power supply
	2	GND_U_{AUX}	Ground/reference potential U_{AUX} (electrically **isolated** to GND_U_S internally in device)
	3	GND_U_S	Ground/reference potential U_S
	4	U_{AUX} (+24 V)	Auxiliary supply (electrically **isolated** to U_S internally in device)
	5	FE	Functional ground

Table 9: Power supply with M12-Power Class A/B

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

6.3.3 I/O ports as M12 sockets

Color coding: black

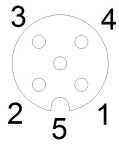


Figure 6: Schematic drawing I/O port as M12 socket IO-Link

6.3.3.1 IO-Link ports (Class A)

MP08...	Pin	Signal	Function
IO-Link Class A, ports X1 .. X8	1	+24 V	power supply +24 V
	2	IN/OUT	Ch. B: Digital input or digital output
	3	GND	Ground/reference potential
	4	C/Q	Ch. A: IO-Link data communication, digital input or digital output
	5	n.c.	not connected
PN08IOLA.../ EIP08IOLA.../ EC08IOLA.../ MTCP08IOLA.../ CCL08IOLA...	**Pin**	**Signal**	**Function**
IO-Link Class A, ports X1 .. X8	1	+24 V	power supply +24 V
	2	IN	Ch. B: Digital input
	3	GND	Ground/reference potential
	4	C/Q	Ch. A: IO-Link data communication, digital input or digital output
	5	n.c.	not connected

Table 10: I/O ports as IO-Link Class A

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

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

7 Starting operation

7.1 ESI file

An ESI file in XML format is required to configure the UNITRONIC® ACCESS multi-protocol and UNITRONIC® ACCESS single-protocol devices. All device variants are grouped in a single ESI file. The file can be downloaded from the product pages on our online catalog: https://lapp.com

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

The file for the UNITRONIC® ACCESS EtherCAT® devices is named:

U.I.Lapp-UnitronicAccess-IO-Link-Master.xml

Download this file and unpack it.

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

For TwinCAT®, the ESI file 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**.

The UNITRONIC® ACCESS multi-protocol and UNITRONIC® ACCESS single-protocol devices are then available in the hardware catalog.

7.2 MAC addresses

Every device has three unique MAC address assigned by the manufacturer 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 device.

7.3 Setting the rotary encoding switches

The following UNITRONIC® ACCESS IO-Link Master variants additionally provide the protocol CC-Link IE Field Basic (CC):

- MP08IOLA08DIO

Caution: Risk of device damage due to corrupt device memory

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

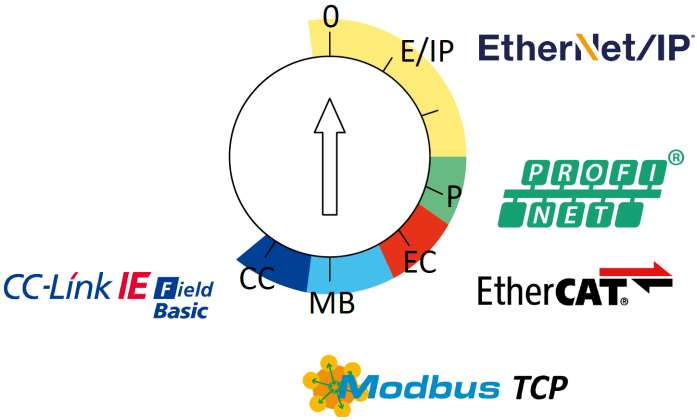
After selecting a protocol followed by a restart of the device, the new protocol is initialized. This can take up to 15 seconds. During this time the device is not usable and the LED indicators are out of function. When the protocol change is complete, the LED indicators return to normal operation and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

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

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

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

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



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

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

Table 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 46.

If you position the rotary encoding switch in a manner that is invalid, the device signals this to you with a blink code (the RUN and ERR LEDs are off).

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

	x100	x10	x1
Factory Reset	9	7	9

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

For performing a factory reset via software configuration, see chapter OPC UA configuration on page 125 and the configuration section.

8 Configuration and operation in TwinCAT® 3

8.1 PDO assignments

The UNITRONIC® ACCESS Multi-protocol and UNITRONIC® ACCESS single-protocol devices support 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 devices feature a dynamic, slot-based PDO assignment. The following PDO assignments are provided:

8.1.1 Input data

PDO 0x1A00

PDO		PDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x1A00	32	0x6000:1	1	UINT32	1st byte of IO-Link input data
		…	…	…	…
		0x6000:32	1	UINT32	32nd byte of IO-Link input data

PDO 0x1A01

PDO		PDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x1A01	32	0x6010:1	1	UINT32	1st byte of IO-Link input data
		…	…	…	…
		0x6010:32	1	UINT32	32nd byte of IO-Link input data

PDO 0x1A02

PDO		PDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x1A02	32	0x6020:1	1	UINT32	1st byte of IO-Link input data
		…	…	…	…
		0x6020:32	1	UINT32	32nd byte of IO-Link input data

PDO 0x1A03

PDO		PDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x1A03	32	0x6030:1	1	UINT32	1st byte of IO-Link input data
		…	…	…	…
		0x6030:32	1	UINT32	32nd byte of IO-Link input data

PDO 0x1A04

PDO		PDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x1A04	32	0x6040:1	1	UINT32	1st byte of IO-Link input data
		…	…	…	…
		0x6040:32	1	UINT32	32nd byte of IO-Link input data

PDO 0x1A05

PDO		PDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x1A05	32	0x6050:1	1	UINT32	1st byte of IO-Link input data
		…	…	…	…
		0x6050:32	1	UINT32	32nd byte of IO-Link input data

PDO 0x1A06

PDO		PDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x1A06	32	0x6060:1	1	UINT32	1st byte of IO-Link input data
		…	…	…	…
		0x6060:32	1	UINT32	32nd byte of IO-Link input data

PDO 0x1A07

PDO		PDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x1A07	32	0x6070:1	1	UINT32	1st byte of IO-Link input data
		…	…	…	…
		0x6070:32	1	UINT32	32nd byte of IO-Link input data

PDO 0x1A08

PDO		PDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x1A08	4	0x2080:1	1	UINT32	Digital input status of ports X1 .. X4 Bitwise
		0x2080:2	1	UINT32	Digital input status of ports X5 .. X8 Bitwise
		0x2080:3	1	UINT32	State of IO-Link Communication Bitwise
		0x2080:4	1	UINT32	Status of IO-Link Process Data valid Bitwise

PDO 0x1A09

PDO		PDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x1A09	1	0x10F3: 4	1	UINT32	Flag “New Messages Available” from Diagnostic Object 0x10F3

PDO 0x1A0A

PDO		PDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x1A0A	1	0x10F8: 0	1	UINT32	Timestamp Object 0x10F8

PDO 0x1A80

PDO		PDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x1A80	8	0xF100:1	1	UINT32	Status IO-Link port 1
		0xF100:2	1	UINT32	Status IO-Link port 2
		0xF100:3	1	UINT32	Status IO-Link port 3
		0xF100:4	1	UINT32	Status IO-Link port 4
		0xF100:5	1	UINT32	Status IO-Link port 5
		0xF100:6	1	UINT32	Status IO-Link port 6
		0xF100:7	1	UINT32	Status IO-Link port 7
		0xF100:8	1	UINT32	Status IO-Link port 8

8.1.2 Output data

PDO 0x1600

PDO		PDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x1600	32	0x7000:1	1	UINT32	1st byte of IO-Link output data
		…	…	…	…
		0x7000:32	1	UINT32	32nd byte of IO-Link output data

PDO 0x1601

PDO		PDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x1601	32	0x7010:1	1	UINT32	1st byte of IO-Link output data
		…	…	…	…
		0x7010:32	1	UINT32	32nd byte of IO-Link output data

PDO 0x1602

PDO		PDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x1602	32	0x7020:1	1	UINT32	1st byte of IO-Link output data
		…	…	…	…
		0x7020:32	1	UINT32	32nd byte of IO-Link output data

PDO 0x1603

PDO		PDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x1603	32	0x7030:1	1	UINT32	1st byte of IO-Link output data
		…	…	…	…
		0x7030:32	1	UINT32	32nd byte of IO-Link output data

PDO 0x1604

PDO		PDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x1604	32	0x7040:1	1	UINT32	1st byte of IO-Link output data
		…	…	…	…
		0x7040:32	1	UINT32	32nd byte of IO-Link output data

PDO 0x1605

PDO		PDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x1605	32	0x7050:1	1	UINT32	1st byte of IO-Link output data
		…	…	…	…
		0x7050:32	1	UINT32	32nd byte of IO-Link output data

PDO 0x1606

PDO		PDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x1606	32	0x7060:1	1	UINT32	1st byte of IO-Link output data
		…	…	…	…
		0x7060:32	1	UINT32	32nd byte of IO-Link output data

PDO 0x1607

PDO		PDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x1607	32	0x7070:1	1	UINT32	1st byte of IO-Link output data
		…	…	…	…
		0x7070:32	1	UINT32	32nd byte of IO-Link output data

PDO 0x1608

PDO		PDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x1608	4	0x2280:1	1	UINT32	Digital output mapping for ports X1 .. X4 Bitwise
		0x2280:2	1	UINTtwincat32	Digital output mapping for ports X5 .. X8 Bitwise
		0x2280:3	1	UINT32	Control of IO-Link COM mode Bitwise
		0x2280:4	1	UINT32	Reserved

8.1.3 Modular slots

The ESI file features a modular slot-based PDO configuration for different IO-Link configurations. The following slots are available:

Slot name	Description
STD_IN_1_bit	IO-Link port as standard Digital Input
IOL_I_1byte	IO-Link, 1 byte of process data input
IOL_I_2byte	IO-Link, 2 bytes of process data input
IOL_I_4byte	IO-Link, 4 bytes of process data input
IOL_I_6byte	IO-Link, 6 bytes of process data input
IOL_I_8byte	IO-Link, 8 bytes of process data input
IOL_I_10byte	IO-Link, 10 bytes of process data input
IOL_I_16byte	IO-Link, 16 bytes of process data input
IOL_I_24byte	IO-Link, 24 bytes of process data input
IOL_I_32byte	IO-Link, 32 bytes of process data input
STD_OUT_1_bit	IO-Link port as standard Digital Output
IOL_O_1byte	IO-Link, 1 bytes of process data output
IOL_O_2byte	IO-Link, 2 bytes of process data output
IOL_O_4byte	IO-Link, 4 bytes of process data output
IOL_O_6byte	IO-Link, 6 bytes of process data output
IOL_O_8byte	IO-Link, 8 bytes of process data output
IOL_O_10byte	IO-Link, 10 bytes of process data output
IOL_O_16byte	IO-Link, 16 bytes of process data output
IOL_O_24byte	IO-Link, 24 bytes of process data output
IOL_O_32byte	IO-Link, 32 bytes of process data output
IOL_I/O_1/1byte	IO-Link, 1 byte of process data input 1 byte of process data output

Slot name	Description
IOL_I/O_2/2byte	IO-Link, 2 bytes of process data input 2 bytes of process data output
IOL_I/O_2/4byte	IO-Link, 2 bytes of process data input 4 bytes of process data output
IOL_I/O_4/4byte	IO-Link, 4 bytes of process data input 4 bytes of process data output
IOL_I/O_4/2byte	IO-Link, 4 bytes of process data input 2 bytes of process data output
IOL_I/O_2/8byte	IO-Link, 2 bytes of process data input 8 bytes of process data output
IOL_I/O_4/8byte	IO-Link, 4 bytes of process data input 8 bytes of process data output
IOL_I/O_8/2byte	IO-Link, 8 bytes of process data input 2 bytes of process data output
IOL_I/O_8/4byte	IO-Link, 8 bytes of process data input 4 bytes of process data output
IOL_I/O_8/8byte	IO-Link, 8 bytes of process data input 8 bytes of process data output
IOL_I/O_4/32byte	IO-Link, 4 bytes of process data input 32 bytes of process data output
IOL_I/O_32/4byte	IO-Link, 32 bytes of process data input 4 bytes of process data output

Slot name	Description
IOL_I/O_16/16byte	IO-Link, 16 bytes of process data input 16 bytes of process data output
IOL_I/O_24/24byte	IO-Link, 24 bytes of process data input 24 bytes of process data output
IOL_I/O_32/32byte	IO-Link, 32 bytes of process data input 32 bytes of process data output

8.2 Device parameters

The UNITRONIC® ACCESS Multi-protocol and UNITRONIC® ACCESS single-protocol devices support different parameters. The parameters must be transferred from the controller to the device during startup. The following blocks of parameters can be adjusted.

8.2.1 Extended parameters

The following table represents extended parameters of the IO-Link ports.

SDO		SDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name (specification)**
0x23n0	2	0x23n0:1	1	UINT8	**Data Storage** 0 = Data Storage disabled (stored data set is preserved) 1 = Data Storage download only (IOLM → IOLD) 2 = Data Storage upload only (IOLD → IOLM) 3 = Data Storage download and upload 4 = Data Storage disabled and cleared (a previously stored data set is cleared) Others: reserved
		0x23n0:2	1	UINT8	**Fail Safe Mode** 0 = Set Low 1 = Set High 2 = Hold Last Value 3 = Replacement value (refer object 0x2301) 4 = IO-Link master command Others: reserved

Key:

n = number between 0 .. 7 (port number -1)

8.2.2 Failsafe replacement values

The firmware of the devices provides a fail-safe function for the port while in IO-Link mode. During device configuration, you have the option to define the replacement value of the IO-Link output data.

The following table represents failsafe replacement values of IO-Link output data while the port is in IO-Link mode.

SDO		SDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x23n1	2	0x23n1:1	1	UINT8	Failsafe replacement value byte 1
		…	…	…	…
		0x23n1:32	32	UINT8	Failsafe replacement value byte 32

Key:

n = number between 0 .. 7 (port number -1)

8.2.3 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.

The following options are available:

Set low (default value)	Deactivate the output channel (value = 0)
Set high	Activate the output channel (value = 1)
Hold last	Hold the last output status (value = 2)

SDO		SDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x2380	16	0x2380 : 1	1	UINT8	**Port 1 Channel A** 0 = Set Low 1 = Set High 2 = Hold Last Others: reserved
		0x2380 : 2	1	UINT8	**Port 1 Channel B** 0 = Set Low 1 = Set High 2 = Hold Last Others: reserved
		0x2380 : 3	1	UINT8	**Port 2 Channel A** 0 = Set Low 1 = Set High 2 = Hold Last Others: reserved

SDO		SDO content			
		0x2380 : 4	1	UINT8	**Port 2 Channel B** 0 = Set Low 1 = Set High 2 = Hold Last Others: reserved
		0x2380 : 5	1	UINT8	**Port 3 Channel A** 0 = Set Low 1 = Set High 2 = Hold Last Others: reserved
		0x2380 : 6	1	UINT8	**Port 3 Channel B** 0 = Set Low 1 = Set High 2 = Hold Last Others: reserved
		0x2380 : 7	1	UINT8	**Port 4 Channel A** 0 = Set Low 1 = Set High 2 = Hold Last Others: reserved
		0x2380 : 8	1	UINT8	**Port 4 Channel B** 0 = Set Low 1 = Set High 2 = Hold Last Others: reserved
		0x2380 : 9	1	UINT8	**Port 5 Channel A** 0 = Set Low 1 = Set High 2 = Hold Last Others: reserved
		0x2380 : 10	1	UINT8	**Port 5 Channel B** 0 = Set Low 1 = Set High 2 = Hold Last Others: reserved

SDO		SDO content			
		0x2380 : 11	1	UINT8	**Port 6 Channel A** 0 = Set Low 1 = Set High 2 = Hold Last Others: reserved
		0x2380 : 12	1	UINT8	**Port 6 Channel B** 0 = Set Low 1 = Set High 2 = Hold Last Others: reserved
		0x2380 : 13	1	UINT8	**Port 7 Channel A** 0 = Set Low 1 = Set High 2 = Hold Last Others: reserved
		0x2380 : 14	1	UINT8	**Port 7 Channel B** 0 = Set Low 1 = Set High 2 = Hold Last Others: reserved
		0x2380 : 15	1	UINT8	**Port 8 Channel A** 0 = Set Low 1 = Set High 2 = Hold Last Others: reserved
		0x2380 : 16	1	UINT8	**Port 8 Channel B** 0 = Set Low 1 = Set High 2 = Hold Last Others: reserved

8.2.4 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

8.2.5 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. While an output channel is in static state (permanently switched on or off), the respective value is 100 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

8.2.6 Digital I/O mode, Channel B

The device supports the configuration of channel B (Pin 2) of the IO-Link port via SDO 0x2383 in input or output mode.

SDO		SDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x2383	8	0x2383 : 1	1	UINT8	**Digital I/O Mode Port 1 Channel B** 1 = Input 2 = Output Others = reserved
		0x2383: 2	1	UINT8	**Digital I/O Mode, Port 2 Channel B** 1 = Input 2 = Output Others = reserved
		0x2383: 3	1	UINT8	**Digital I/O Mode, Port 3 Channel B** 1 = Input 2 = Output Others = reserved
		0x2383: 4	1	UINT8	**Digital I/O Mode, Port 4 Channel B** 1 = Input 2 = Output Others = reserved
		0x2383: 5	1	UINT8	**Digital I/O Mode, Port 5 Channel B** 1 = Input 2 = Output Others = reserved
		0x2383: 6	1	UINT8	**Digital I/O Mode, Port 6 Channel B** 1 = Input 2 = Output Others = reserved
		0x2383: 7	1	UINT8	**Digital I/O Mode, Port 7 Channel B** 1 = Input 2 = Output Others = reserved

SDO		SDO content			
		0x2383: 8	1	UINT8	**Digital I/O Mode, Port 8 Channel B** 1 = Input 2 = Output Others = reserved

8.2.7 Digital Input logic

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

The following values are only applicable for ports in *Digital Input* mode.

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

8.2.8 Digital Input filter

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

The following values are only applicable for ports in *Digital Input* mode.

(Entered value multiplied with 0.1 ms = input filter value in ms)

SDO		SDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x2385	16	0x2385 : 1	1	UINT8	Digital Input Filter Port 1 Channel A in 0.1 ms
		0x2385: 2	1	UINT8	Digital Input Filter Port 1 Channel B in 0.1 ms
		0x2385: 3	1	UINT8	Digital Input Filter Port 2 Channel A in 0.1 ms
		0x2385: 4	1	UINT8	Digital Input Filter Port 2 Channel B in 0.1 ms
		0x2385: 5	1	UINT8	Digital Input Filter Port 3 Channel A in 0.1 ms
		0x2385: 6	1	UINT8	Digital Input Filter Port 3 Channel B in 0.1 ms
		0x2385: 7	1	UINT8	Digital Input Filter Port 4 Channel A in 0.1 ms
		0x2385: 8	1	UINT8	Digital Input Filter Port 4 Channel B in 0.1 ms
		0x2385: 9	1	UINT8	Digital Input Filter Port 5 Channel A in 0.1 ms
		0x2385: 10	1	UINT8	Digital Input Filter Port 5 Channel B in 0.1 ms
		0x2385: 11	1	UINT8	Digital Input Filter Port 6 Channel A in 0.1 ms

SDO		SDO content			
		0x2385: 12	1	UINT8	Digital Input Filter Port 6 Channel B in 0.1 ms
		0x2385: 13	1	UINT8	Digital Input Filter Port 7 Channel A in 0.1 ms
		0x2385: 14	1	UINT8	Digital Input Filter Port 7 Channel B in 0.1 ms
		0x2385: 15	1	UINT8	Digital Input Filter Port 8 Channel A in 0.1 ms
		0x2385: 16	1	UINT8	Digital Input Filter Port 8 Channel B in 0.1 ms

8.2.9 Digital Output restart

The device supports the configuration of a digital output timeout before restart for Channel A (Pin 4) and Channel B (Pin 2) of the IO-Link port.

The following values are only applicable for ports in *Digital Output* mode.

SDO		SDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x2386	16	0x2386 : 1	1	BOOL	Digital output timeout before restart in ms Port 1 Channel A
		0x2386: 2	1	BOOL	Digital output timeout before restart in ms Port 1 Channel B
		0x2386: 3	1	BOOL	Digital output timeout before restart in ms Port 2 Channel A
		0x2386: 4	1	BOOL	Digital output timeout before restart in ms Port 2 Channel B
		0x2386: 5	1	BOOL	Digital output timeout before restart in ms Port 3 Channel A
		0x2386: 6	1	BOOL	Digital output timeout before restart in ms Port 3 Channel B
		0x2386: 7	1	BOOL	Digital output timeout before restart in ms Port 4 Channel A
		0x2386: 8	1	BOOL	Digital output timeout before restart in ms Port 4 Channel B
		0x2386: 9	1	BOOL	Digital output timeout before restart in ms Port 5 Channel A
		0x2386: 10	1	BOOL	Digital output timeout before restart in ms Port 5 Channel B
		0x2386: 11	1	BOOL	Digital output timeout before restart in ms Port 6 Channel A

SDO		SDO content			
		0x2386: 12	1	BOOL	Digital output timeout before restart in ms Port 6 Channel B
		0x2386: 13	1	BOOL	Digital output timeout before restart in ms Port 7 Channel A
		0x2386: 14	1	BOOL	Digital output timeout before restart in ms Port 7 Channel B
		0x2386: 15	1	BOOL	Digital output timeout before restart in ms Port 8 Channel A
		0x2386: 16	1	BOOL	Digital output timeout before restart in ms Port 8 Channel B

8.2.10 Digital Input latch

The device supports a digital input latch function for Channel A (Pin 4) and Channel B (Pin 2) of the IO-Link port.

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

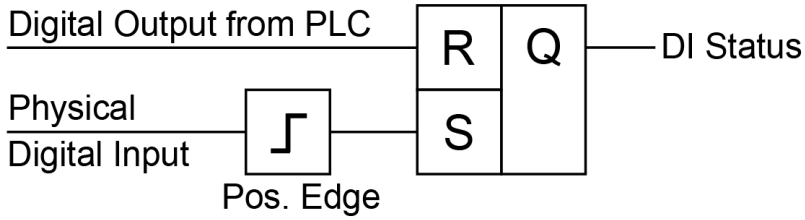


Figure 7: Input latch

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

Table 12: Truth table for input latch

When the input latch is enabled for a specific channel:

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

The behavior in detail:

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

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

The following values are only applicable for ports in *Digital Input* mode.

SDO		SDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x2389	16	0x2389 : 1	1	BOOL	Digital Input Latch Port 1 Channel A 0 = false, 1 = true
		0x2389: 2	1	BOOL	Digital Input Latch Port 1 Channel B 0 = false, 1 = true
		0x2389: 3	1	BOOL	Digital Input Latch Port 2 Channel A 0 = false, 1 = true
		0x2389: 4	1	BOOL	Digital Input Latch Port 2 Channel B 0 = false, 1 = true
		0x2389: 5	1	BOOL	Digital Input Latch Port 3 Channel A 0 = false, 1 = true
		0x2389: 6	1	BOOL	Digital Input Latch Port 3 Channel B 0 = false, 1 = true
		0x2389: 7	1	BOOL	Digital Input Latch Port 4 Channel A 0 = false, 1 = true
		0x2389: 8	1	BOOL	Digital Input Latch Port 4 Channel B 0 = false, 1 = true
		0x2389: 9	1	BOOL	Digital Input Latch Port 5 Channel A 0 = false, 1 = true
		0x2389: 10	1	BOOL	Digital Input Latch Port 5 Channel B 0 = false, 1 = true
		0x2389: 11	1	BOOL	Digital Input Latch Port 6 Channel A 0 = false, 1 = true

SDO		SDO content			
		0x2389: 12	1	BOOL	Digital Input Latch Port 6 Channel B 0 = false, 1 = true
		0x2389: 13	1	BOOL	Digital Input Latch Port 7 Channel A 0 = false, 1 = true
		0x2389: 14	1	BOOL	Digital Input Latch Port 7 Channel B 0 = false, 1 = true
		0x2389: 15	1	BOOL	Digital Input Latch Port 8 Channel A 0 = false, 1 = true
		0x2389: 16	1	BOOL	Digital Input Latch Port 8 Channel B 0 = false, 1 = true

8.2.11 Digital Input extension

The device supports the configuration of an extension for the digital input signal for Channel A (Pin 4) and Channel B (Pin 2) of the IO-Link port.

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

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

Example:

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

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

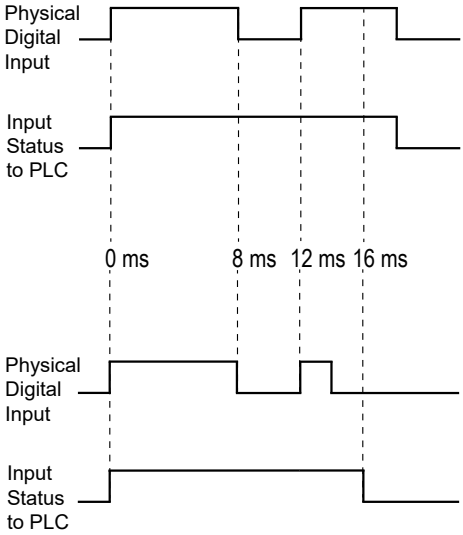


Figure 8: DI Extension

Available values: Off; 8 ms; 16 ms; 64 ms

Default: Off

The following values are only applicable for ports in *Digital Input* mode.

SDO		SDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x2390	16	0x2390 : 1	1	UINT8	Digital Input Extension in ms Port 1 Channel A
		0x2390: 2	1	UINT8	Digital Input Extension in ms Port 1 Channel B
		0x2390: 3	1	UINT8	Digital Input Extension in ms Port 2 Channel A
		0x2390: 4	1	UINT8	Digital Input Extension in ms Port 2 Channel B
		0x2390: 5	1	UINT8	Digital Input Extension in ms Port 3 Channel A
		0x2390: 6	1	UINT8	Digital Input Extension in ms Port 3 Channel B
		0x2390: 7	1	UINT8	Digital Input Extension in ms Port 4 Channel A
		0x2390: 8	1	UINT8	Digital Input Extension in ms Port 4 Channel B
		0x2390: 9	1	UINT8	Digital Input Extension in ms Port 5 Channel A
		0x2390: 10	1	UINT8	Digital Input Extension in ms Port 5 Channel B
		0x2390: 11	1	UINT8	Digital Input Extension in ms Port 6 Channel A
		0x2390: 12	1	UINT8	Digital Input Extension in ms Port 6 Channel B
		0x2390: 13	1	UINT8	Digital Input Extension in ms Port 7 Channel A
		0x2390: 14	1	UINT8	Digital Input Extension in ms Port 7 Channel B
		0x2390: 15	1	UINT8	Digital Input Extension in ms Port 8 Channel A
		0x2390: 16	1	UINT8	Digital Input Extension in ms Port 8 Channel B

8.2.12 Additional IO-Link port settings

The device firmware provides the following additional setting options for each individual IO-Link port via SDO.

SDO		SDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x30n0	6	0x30n0:1	1	UINT8	Swap Mode
		0x30n0:2	1	UINT8	Swap Length
		0x30n0:3	1	UINT8	Swap Offset
		0x30n0:4	1	BOOL	Sensor Supply Enabled 0 = false, 1 = true
		0x30n0:5	1	BOOL	Pin 2 LED Enabled 0 = false, 1 = true
		0x30n0:6	1	BOOL	Suppress All Diagnosis 0 = false, 1 = true
		0x30n0:7	1	BOOL	Pin 4 DO use of Push-Pull 0 = Use High-Side switches 1 = Use Push-Pull
		0x30n0:8	1	UINT16	Pin 4 current limit in mA (Maximum current limit till Pin 4 is turned off) Default: 2000
		0x30n0:9	1	UINT16	Pin 2 DO use of Push-Pull in mA (Maximum current limit till Pin 2 is turned off) Default: 2000

Key:

n = number between 0 .. 7 (port number -1)

Swap Mode

The byte order swapping will be made for the selected count of data types or for the complete length of I/O-Data with the selected Data Type (Word = 2 Bytes or DWord = 4 Bytes).

Swap Length

The swapping can be setup to Word (2 Bytes) or DWord (4 Bytes)

- Word Swapping: Byte 1 - Byte 2 => Byte 2 - Byte 1
- DWord Swapping: Byte 1 - Byte 4 => Byte 4 – Byte 1

Swapping Offset

A swapping offset in bytes can be setup in dependency of the configured I/O-Data length. When setup to "2", the swapping will be processed from the 3rd Byte.

8.2.13 IO-Link Parameterization

The device features read or write of ISDU (IO-Link Service Data) parameters via SDO 0x40n0.

SDO		SDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x40n0	7	0x40n0:1	1	UINT8	**Control** 0x00: no action 0x02: read 0x03: write
		0x40n0:2	1	UINT8	**Status** 0x00: No activity 0x01: Active/busy 0x02: Access finished 0x03: Error 0x04: ISDU blocked
		0x40n0:3	1	UINT16	Index
		0x40n0:4	2	UINT8	Subindex
		0x40n0:5	1	UINT8	Length
		0x40n0:6	232	UINT232	Data
		0x40n0:7	2	UINT16	**Error** 0x00: No error in ISDU request 0x01: Error during ISDU request

Key:

n = number between 0 .. 7 (port number -1)

8.2.14 IO-Link configuration data

This is the standard 0x80n0 object for the IO-Link configuration data as per the IO-Link device profile.

SDO		SDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x80n0	7	0x80n0:4	4	UINT32	Device ID
		0x80n0:5	4	UINT32	Vendor ID
		0x80n0:32	1	UINT8	**IO LINK revision** Version of the implemented IO-Link specification (of the connected IO-Link device) according to version 1.0 of the IO-Link specification. **Bit 0...3:** Minor Rev **Bit 4...7:** Major Rev
		0x80n0:33	1	UINT8	Frame Capability
		0x80n0:34	1	UINT8	Cycle Time
		0x80n0:35	1	UINT8	Offset Time
		0x80n0:36	1	UINT8	**Process data IN length** Number and structure of input data This value is transmitted in IO-Link format as "ProcessDataIn" according to version 1.0 of the IO-Link specification. **Bit 0...4:** Length **Bit 5:** reserved **Bit 6:** SIO Indicator (if device supports standard I/O mode) **Bit 7:** Byte Indicator (if value of length is interpreted as bit length or as byte length +1)
		0x80n0:37	1	UINT8	**Process data OUT length** Number and structure of input data This value is transmitted in IO-Link format as "ProcessDataOut" according to version 1.0 of the IO-Link specification. **Bit 0...4:** Length **Bit 5:** reserved **Bit 6:** SIO Indicator (if device supports standard I/O mode) **Bit 7:** Byte Indicator (if value of length is interpreted as bit length or as byte length +1)

SDO		SDO content			
		0x80n0:38	2	UINT16	Compatible ID
		0x80n0:39	2	UINT16	Reserved
		0x80n0:40	2	UINT16	**Master Control** Control of the IO-Link master port. Defines the different operating modes of the IO-Link master. **Bits 0...3:** 0: Inactive 1: Digital Input Port 2: Digital Output Port 3: Communication over IO-Link Protocol 4: Digital input with support for acyclic IO-Link

Key:

n = number between 0 .. 7 (port number -1)

8.2.15 IO-Link serial number

This is the standard 0x80n1 object for the IO-Link serial number as per the IO-Link device profile.

SDO		SDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x80n1	20	0x80n1:1	1	VISIBLE_STRING	Serial Number

Key:

n = number between 0 .. 7 (port number -1)

8.2.16 IO-Link information data

This is standard 0x90n0 object for the IO-Link information data as per the IO-Link device profile.

SDO		SDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x90n0	7	0x90n0:4	4	UINT32	Device ID
		0x90n0:5	4	UINT32	Vendor ID
		0x90n0:32	1	UINT8	**IO LINK revision** Version of the implemented IO-Link specification (of the connected IO-Link device) according to version 1.0 of the IO-Link specification. **Bit 0...3:** Minor Rev **Bit 4...7:** Major Rev
		0x90n0:33	1	UINT8	Frame Capability
		0x90n0:34	1	UINT8	Cycle Time
		0x90n0:35	1	UINT8	Offset Time
		0x90n0:36	1	UINT8	**Process data IN length** Number and structure of input data This value is transmitted in IO-Link format as "ProcessDataIn" according to version 1.0 of the IO-Link specification. **Bit 0...4:** Length **Bit 5:** reserved **Bit 6:** SIO Indicator (if device supports standard I/O mode) **Bit 7:** Byte Indicator (if value of length is interpreted as bit length or as byte length +1)

SDO		SDO content			
		0x90n0:37	1	UINT8	**Process data OUT length** Number and structure of input data This value is transmitted in IO-Link format as "ProcessDataOut" according to version 1.0 of the IO-Link specification. **Bit 0...4:** Length **Bit 5:** reserved **Bit 6:** SIO Indicator (if device supports standard I/O mode) **Bit 7:** Byte Indicator (if value of length is interpreted as bit length or as byte length +1)

Key:

n = number between 0 .. 7 (port number -1)

8.2.17 IO-Link serial number of connected devices

This is the standard 0x90n1 object for the IO-Link serial number of connected devices as per the IO-Link device profile.

SDO		SDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x9n01	20	0x90n1:1	1	VISIBLE_STRING	Serial Number

Key:

n = number between 0 .. 7 (port number -1)

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, refer to the corresponding documentation.

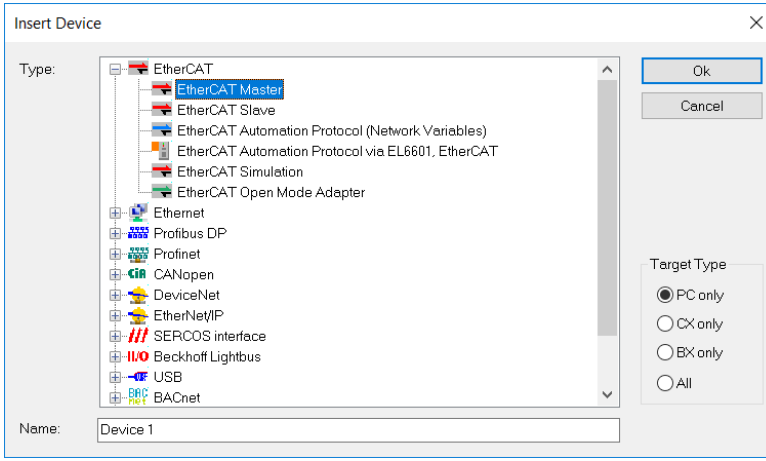
1. Install the ESI file of the device family in TwinCAT®. For TwinCAT®, the ESI file must be copied into the installation folder, e.g.: `C:\TwinCAT\3.1\Config\Io\EtherCAT`.

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.

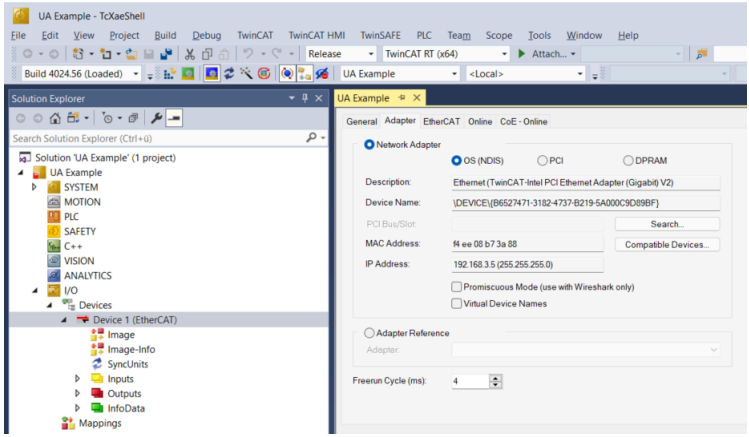
3. Start TwinCAT® and open a new project.

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.

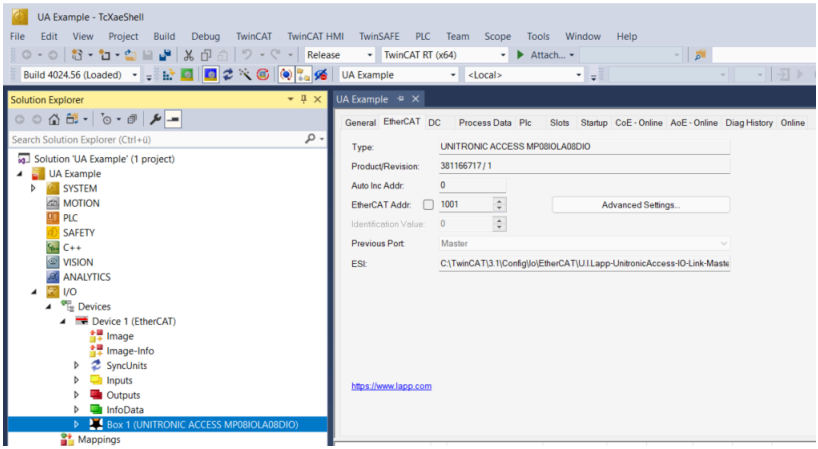
Browse to **Adapter** in the right workspace window and click on **Compatible Devices...** to choose the driver and start the installation.



8.3.1 Configuration of MP08... devices

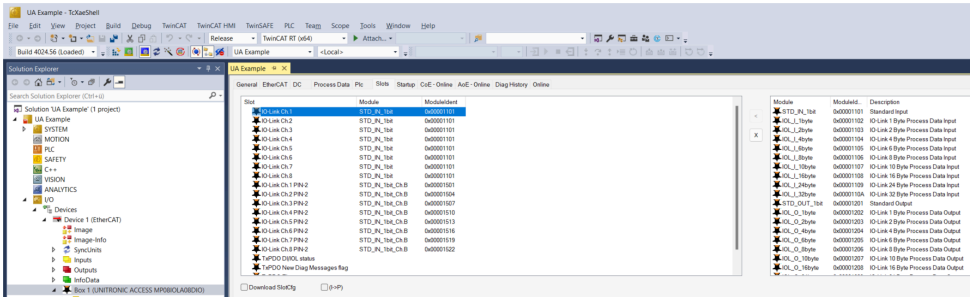
1. Selecting the I/O device from the hardware catalog:

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. Configuring the “Slots”:

Browse to **Slots** in the right workspace window and configure the IO-Link channels. You can change, for example, the Input/Output length, the channel mode, or the I/O mode of Channel B (Pin 2). Additional PDOs like "TXPDO" for the DI/IOL status, "TxPDO" for a new diagnostic messages flag, "TxPDO" for a timestamp and "RxPDO" for DO/IOL control can be set as well.



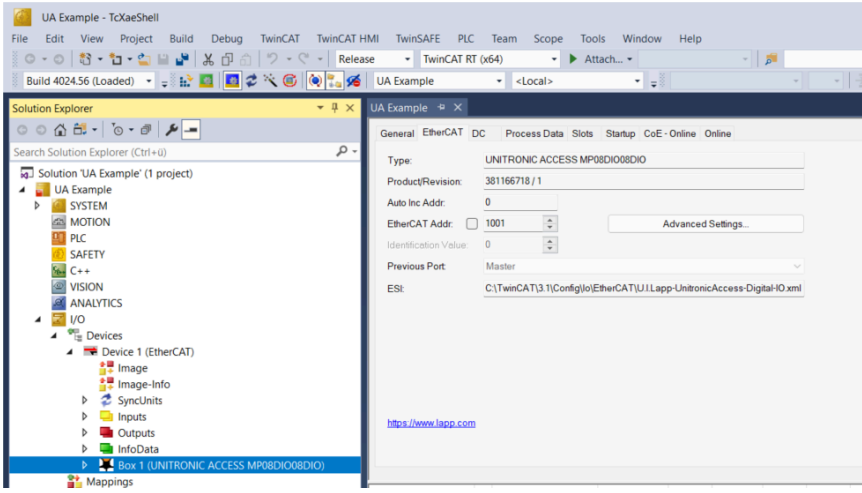
3. Configure the Process Data:

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

8.3.2 EoE IP address

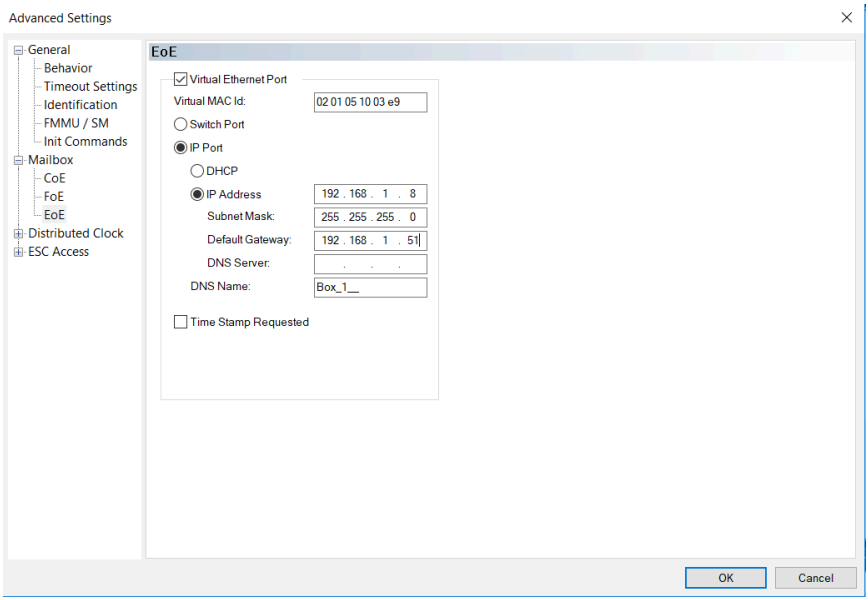
1. Set the IP address for the EoE (Ethernet over EtherCAT®) protocol:

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.



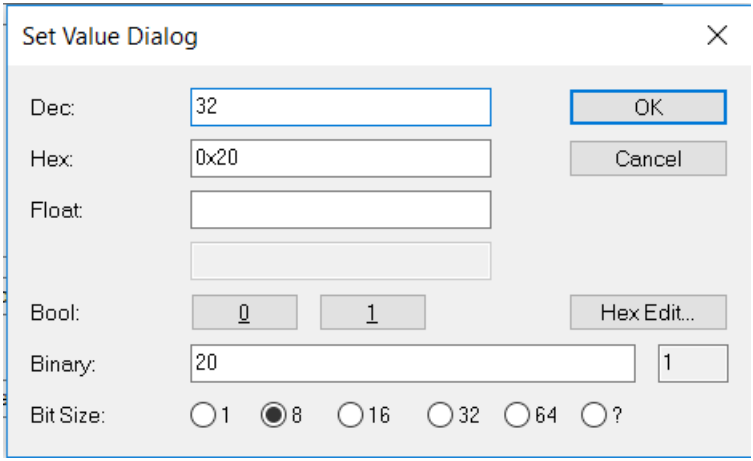
8.3.3 Activate configuration

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

9.1 Diagnostics (0x2001)

Index	Size	Index	Size	Type	Name
0x2001	09	0x2001:1	1	BOOL	U_S supply fault
		0x2001:2	1	BOOL	U_L supply present
		0x2001:3	1	BOOL	U_L supply fault
		0x2001:4	1	BOOL	Internal Module Error
		0x2001:5	1	BOOL	Force Mode Diagnostic
		0x2001:6	1	UINT8	Sensor Short Fault
		0x2001:7	1	UINT8	CQ Short Fault
		0x2001:8	1	UINT8	IQ Short Fault
		0x2001:9	1	UINT8	P24 Short Fault (Class B)

9.2 Error of the system/sensor power supply

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

The green U_S indicator is off.

The error message has no effect on the outputs.

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

9.3 Error of the auxiliary/actuator power supply

The voltage value for the incoming auxiliary/actuator power supply is also monitored globally. If U_{Aux} diagnostic messaging is enabled, an error message is generated if the voltage drops below approx. 18 V or exceeds approx. 30 V.

The U_{Aux} indicator shows red. If output channels are enabled, additional error messages caused by the voltage failure are generated on the I/O ports. U_{Aux} diagnostic messaging is disabled by default and must be enabled via parameterization.

9.4 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 channel-specific diagnostic messages are generated:

9.5 Overload/short circuit of the digital outputs

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

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 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, the filter time between error detection and the diagnostic message is typically 5 ms.

9.6 IO-Link C/Q error

If an IO-Link Device in COM mode is unplugged, an incorrect IO-Link Device is plugged in, or an electrical fault occurs on the C/Q (Pin 4) line, for example, due to a short circuit, an error message is generated.

9.7 Diagnostic history object (0x10F3)

SDO		SDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0x10F3	69	0x10F3: 1	1	UINT8	Maximum Messages
		0x10F3: 2	1	UINT8	Newest Message
		0x10F3: 3	1	UINT8	Newest Acknowledged Message
		0x10F3: 4	1	BOOL	New Messages Available
		0x10F3: 5	1	UINT16	Flags
		0x10F3: 6-69	1	OCTET_STRING	Diagnosis Message Buffer

9.7.1 Maximum Messages

Number of diagnosis messages which can be stored in the diagnosis history (subindex 6 onwards).

9.7.2 Newest Message

Sub-index of the newest diagnosis message (6 .. 69).

9.7.3 Newest Acknowledge Message

Overwrite mode (SI5, bit 4 = 0)

- Read = 0:

  When the message queue will be overwritten, the slave sets SI3 to "0".

- Writing = 0:

  The slave clears all messages, i.e. resets SI2, SI3, SI4 and SI5 bit 5 Messages are deleted even if they were not acknowledged or read before.

- Writing = 1 .. 5:

  The slave returns SDO abort with code 0x06090032 (value of parameter written too low).

- Writing = 6 .. 69:

  SI3 = written value (without checking)

  All messages up to the age of the message which is in the written sub-index are acknowledged. The slave does not check if those messages have been read before.

  The slave returns SDO abort with code 0x06090030 (value range of parameter exceeded) in the following case: SI3 is written with a value of a sub-index which does not hold a message.

- Writing = 69 .. 255:

  The slave returns SDO abort with code 0x06090031 (value of parameter written too high).

Acknowledge mode (SI5, bit 4 = 1)

- Read = 0:

  No messages have been acknowledged so far.

- Read = 0:

  Sub-index of latest acknowledged diagnosis message (6-69)

- Writing = 0:

  All acknowledged messages will be deleted.

- Writing = 1 .. 5:

  The slave returns SDO abort with code 0x06090032 (value of parameter written too low).

- Writing = 6 .. 69:

  Messages are acknowledged

  All messages up to the age of the message which is in the written sub-index are acknowledged. The slave does not check if those messages have been read before. The slave returns SDO abort with code 0x06090030 (value range of parameter exceeded) in the following case: SI3 is written with a value of a sub-index which does not hold a message.

- Writing = 69 .. 255:

  The slave returns SDO abort with code 0x06090031 (value of parameter written too high).

9.7.4 New Messages Available

Overwrite mode:

- 0: newest message was read
- 1: newest message was not read

Acknowledge mode:

- 0: no unacknowledged message
- 1: diagnosis messages are available which can be acknowledged (SI2 ! = SI3).

9.7.5 Flags

Flags to control sending and storing of diagnosis messages:

Bit 0: Enable Emergency sending

- 0: Default, if device does not support emergency sending.
- 1: New diagnosis messages shall be sent as emergency message.

Bit 1: Disable info messages

- 0: Info messages are stored in the diagnosis message queue (default).

- 1: Info messages will not be stored in the diagnosis message queue.

Bit 2: Disable warning messages

- 0: Warning messages are stored in the diagnosis message queue (default).
- 1: Warning messages will not be stored in the diagnosis message queue.

Bit 3: Disable error messages

- 0: Error messages are stored in the diagnosis message queue (default).
- 1: Error messages will not be stored in the diagnosis message queue.

Bit 4: Mode selection for diagnosis history handling

- 0: Overwrite mode. Old messages are overwritten by new ones when buffer is full.
- 1: Acknowledge mode. New messages do only overwrite messages which were acknowledged before.

Bit 5: Overwrite/Discard Information (read only)

In Overwrite mode:

- 1: Unacknowledged messages have been overwritten (= buffer overrun) (SI3 is set to "0", too).

In Acknowledge mode:

- 1: Message buffer is full with acknowledged messages and a new message is discarded.

Bit 6 .. 15: reserved

9.7.6 Diagnosis Message Buffer

Depending on SI1, the EtherCAT® slave can store up to 64 messages; the first message is stored in sub-index 6, the second in sub-index 7and so on. When the queue is full, the EtherCAT® slave shall overwrite sub-index 6 and the following, in a way that always the latest maximum messages (SI1) shall be accessible by the EtherCAT® master.

9.8 Diagnostic data object (0xA0n0)

SDO		SDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0xA0n0	2	0xA0n0: 1	1	UINT8	IO-Link State
		0xA0n0: 2	1	UINT8	Lost Frames

IO-Link State

State of the IO-Link Master (state machine of the IO-Link port)

0x0 = Port inactive

0x01 = Digital input

0x02 = Digital output

0x03 = Establish communication

0x04-0x07: reserved

0x08 = Communication OPERATE

0x09 = Communication STOP

Lost Frames

Counter of lost IO-Link telegrams. Reset counter during start-up.

9.9 Device status object (0xF100)

SDO		SDO content			
Index	**Size**	**Index**	**Size**	**Type**	**Name**
0xF100	8	0xF100: 1	1	UINT8	Status Port 1
		0xF100: 2	1	UINT8	Status Port 2
		0xF100: 3	1	UINT8	Status Port 3
		0xF100: 4	1	UINT8	Status Port 4
		0xF100: 5	1	UINT8	Status Port 5
		0xF100: 6	1	UINT8	Status Port 6
		0xF100: 7	1	UINT8	Status Port 7
		0xF100: 8	1	UINT8	Status Port 8

Status Ports 1 .. 8

Status of the IO-Link port (state machine of the IO-Link port)

0x_0 = Port Inactive

0x_1 = Port in SIO Digital Input

0x_2 = Port in SIO Digital Output

0x_3 = Port in Communication OP

0x_4 = Port in Communication STOP

0x0_ = No Error

0x1_ = Watchdog Error

0x2_ = Buffer Overflow

0x3_ = Invalid Device Id

0x4_ = Invalid Vendor Id

0x5_ = Invalid IO-Link Version

0x6_ = Invalid Frame Capability

0x7_ = Invalid Cycle Time

0x8_ = Invalid PD in length

0x9_ = Invalid PD out length

0xA_ = No Device detected

0xB_ = Error PreOP

9.10 Emergency messages

If set in the parameters, the Device sends emergency messages to the Master in case of a detected diagnosis on the Device.

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

10 IIoT functionality

The UNITRONIC® ACCESS 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 UNITRONIC® ACCESS 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 UNITRONIC® ACCESS variants provide user administration, which is also applicable for accessing and configuring the IIoT protocols. This allows you to manage all modification options for the device settings via personalized user authorizations.

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

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

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

10.1 MQTT

MQTT functions are **only** applicable for the following UNITRONIC® ACCESS variants:

- MP08IOLA08DIO

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

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

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

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/HTTPS request. For more information see chapter MQTT configuration - Quick start guide on page 122.

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: "**unitronic**")
will-enable	boolean	If true, the device provides a last will message to the broker	true / **false**
will-topic	string	The topic for the last will message.	(Default: **null**)
auto-publish	boolean	If true, all enabled domains will be published automatically in the specified interval.	**true** / false
publish-interval	number	The publish interval in ms if auto-publish is enabled. Minimum is 250 ms.	**2000**
publish-identity	boolean	If true, all identity domain data will be published	**true** / false
publish-config	boolean	If true, all config domain data will be published	**true** / false
publish-status	boolean	If true, all status domain data will be published	**true** / false
publish-process	boolean	If true, all process domain data will be published	**true** / false
publish-devices	boolean	If true, all IO-Link Device domain data will be published	true / **false**
commands-allowed	boolean	Master switch for MQTT commands. If false, the device will not subscribe to any command topic, even if specific command topics are activated below.	true / **false**
force-allowed	boolean	If true, the device accepts force commands via MQTT.	true / **false**
reset-allowed	boolean	If true, the device accepts restart and factory reset commands via MQTT.	true / **false**
config-allowed	boolean	If true, the device accepts configuration changes via MQTT.	true / **false**

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

Table 13: MQTT configuration

MQTT response:

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

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

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

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

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

Examples:

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

{"status": 0}

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

For more information see chapter MQTT topics on page 110.

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 any string and they are allowed to contain slashes (`/`) as well as wildcard symbols (`*, #`).

10.1.2.1 Base topic

For all UNITRONIC® ACCESS variants there is a configurable Base topic which is the prefix for all topics. The Base topic can be chosen freely by the user. The Base topic can also contain selected variables as shown in Table 14: Base topic variables on page 110.

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

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

Table 14: Base topic variables

Example:

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

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

`Base-Topic/domain/….`

There are the following domains:

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

Table 15: Data domains

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

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

Table 16: Data model

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

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

Table 17: Use case examples

10.1.2.2 Publish topic

Overview of all publish JSON data for the defined topics:

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

Table 18: Identity/gateway

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

Table 19: Config/gateway

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

Table 20: Status/gateway

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

Table 21: Process/gateway

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

Table 22: Identity/port/1 .. 8

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

Table 23: Config/port/1 .. 8

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

Table 24: Status/port/1 .. 8

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 26: Force object: Digital on page 119)		
iol	array (Table 27: Force object: IOL (IO-Link devices only) on page 119)		

Table 25: Force object properties

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

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

Table 26: Force object: Digital

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

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

[…]/config

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

Property	Data type	Example values	Remarks
portmode	array (Table 29: Config object: Portmode on page 120)		
ip_address	string	"192.168.1.5"	
subnet_mask	string	"255.255.255.0"	
gateway	string	"192.168.1.100"	

Table 28: Config object properties

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

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

Table 29: Config object: Portmode

*channelA = Pin 4, channelB = Pin 2

[…]/reset

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

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

Table 30: Reset object properties

[…]/publish

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

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

10.1.3 MQTT configuration - Quick start guide

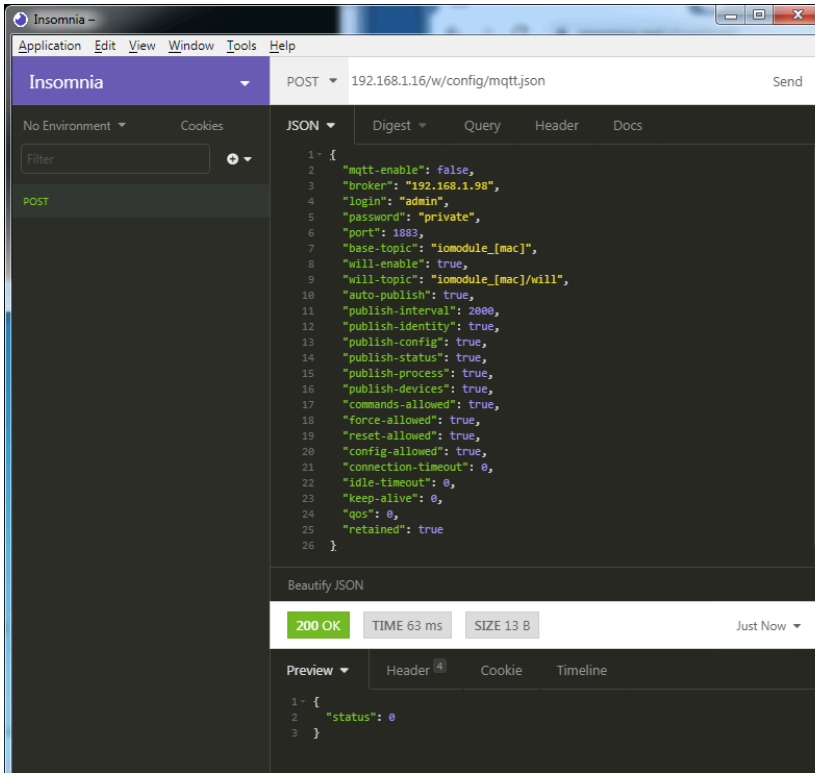
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

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/

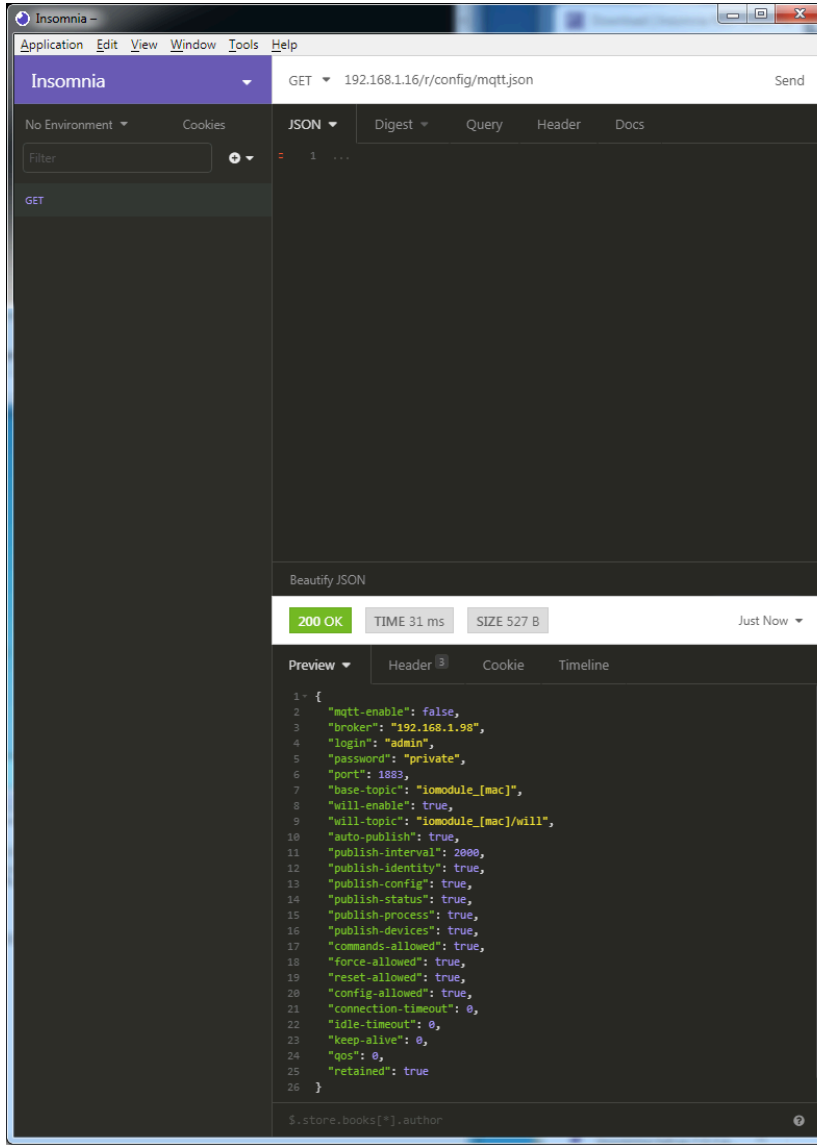
2. Configure MQTT:

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



3. Read MQTT:

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



10.2 OPC UA

OPC UA functions are **only** applicable for the following UNITRONIC® ACCESS variants:

- MP08IOLA08DIO

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. UNITRONIC® ACCESS provides an OPC UA server on field device level, with which an OPC UA client can connect for information exchange secure in transmission.

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

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

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

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

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/HTTPS request. For more information see OPC UA configuration - Quick start guide on page 128.

The configuration URL is:

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

The configuration can also read back as a JSON file:

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

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

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

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

Table 32: OPC UA Configuration

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

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

Response:

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

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

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

Examples:

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

{"status": 0}

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

10.2.2 OPC UA address space

OPC UA provides different services on the UNITRONIC® ACCESS 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 UNITRONIC® ACCESS devices. The objects and information displayed depend on the device variant used.

10.2.3 OPC UA configuration - Quick start guide

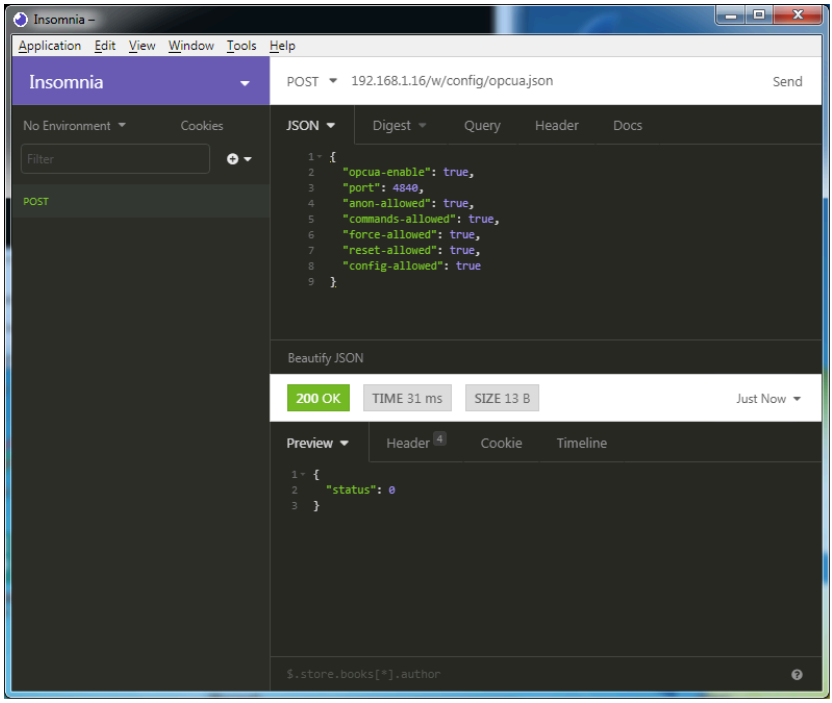
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

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/

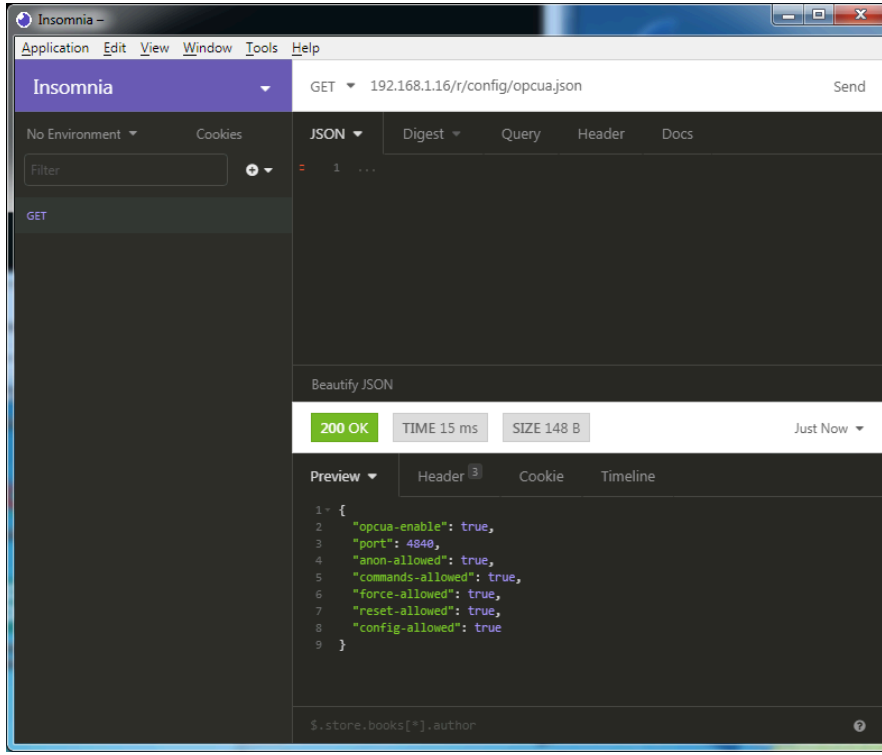
2. Configure OPC UA:

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/HTTPS requests to GET and POST data. This enables the access to detailed device information.

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

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

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

   `JSON_Integration_10222_V100_Mar20.pdf`

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

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

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

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

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

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

2. A customized LAPP REST API that 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. For IO-Link devices, all ports with connected IO-Link device information are included.

10.3.2 Structure

Name	Data type	Description	Example
name	string	Device name	"MP08IOLA08DIO"
order-id	string	Ordering number	"381166717"
fw-version	string	Firmware version	"V.11.2.0.0 - 08.08.2024"
hw-version	string	Hardware version	"V.1.00"
mac	string	MAC address of the device	"7C F9 5C 4C CC CE"
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
iol	IOL Object	Contains all IO-Link related information such as events, port states, device parameters.	
iol/diagGateway	array of DIAG	Array of currently active device/ gateway related events	
iol/diagMaster	array of DIAG	Array of currently active IOL-Master related events	
iol/ports	array of PORT (8)	Contains one element for each IO-Link port	
CHANNEL Object			
name	string	Name of channel	
type	number	Hardware channel type as number: 0 = DIO 1 = Input 2 = Output 3 = Input/Output 4 = IO-Link 5 = IOL AUX 6 = IOL AUX with DO 7 = IOL AUX with DO. Can be deactivated. 8 = Channel not available	
type_text	string	Textual representation of the channel type	
config	number	Current configuration of the channel: 0 = DIO 1 = Input 2 = Output 3 = IO-Link 4 = Deactivated 5 = IOL AUX	
config_text	string	Textual representation of the current config	
inputState	boolean	Input data (producing data) bit to the PLC	
outputState	boolean	Output data bit to the physical output pin	

Name	Data type	Description	Example
forced	boolean	True, if the output pin of this channel is forced	
simulated	boolean	True, if the input value to the PLC of this channel is simulated	
actuatorDiag	boolean	True, if the output is in short circuit / overload condition	
sensorDiag	boolean	True, if the sensor supply (Pin 1) is in short circuit / overload condition	
maxOutputCurrent_mA	number	Maximum output current of the output in mA	
current_mA	number	Measured current of the output in mA (if current measurement is available)	
voltage_mV	number	Measured voltage of this output in mV (if voltage measurement is available)	
PORT Object			
port_type	string	Textual representation of the IO-Link port type	
iolink_mode	number	**Current port mode:** 0 = Inactive 1 = Digital output 2= Digital input 3 = SIO 4 = IO-Link	
iolink_text	string	Textual representation of the current port mode	"Digital Input"
aux_mode	number	Indicates the configured mode for the Pin 2: 0 = No AUX 1 = AUX output (always on) 2 = Digital output (can be controlled by cyclic data) 3 = Digital input	
aux_text	string	Textual representation of the current aux mode	"AUX Output"
cq_mode	number	Port mode according to IOL specification	
iq_mode	number	Pin2 mode according to IOL specification	

Name	Data type	Description	Example
port_status	number	Port status according to IOL specification	
ds_fault	number	Data storage error number	
ds_fault_text	string	Textual data storage error.	
device	DEVICE Object	IO-Link device parameters. → Null if no IO-Link communication active	
diag	array of DIAG (n)	Array of port related events	
DIAG Object			
error	number	Error code	
source	string	Source of the current error.	"device" "master"
eventcode	number	Event code according to IO-Link specification	
eventqualifier	number	Event qualifier according to IO-Link specification	
message	string	Error message	"Supply Voltage fault"
DEVICE Object		Standard parameters of the IOL-Device	
device_id	number		
vendor_id	number		
serial	string		
baudrate	string	Baudrate (COM1,2,3)	
cycle_time	number	Cycle time in microseconds	
input_len	array of numbers (n)	IOL input length in bytes	
output_len	array of numbers (n)	IOL output length in bytes	
input_data	array of numbers (n)	IOL input data	
output_data	array of numbers (n)	IOL output data	
pd_valid	number	"1", if IOL input data is valid	
pdout_valid	number	"1", if IOL output data is valid	
FORCING Object		Forcing information of the device	
forcingActive	boolean	Force mode is currently active	

Name	Data type	Description	Example
forcingPossible	boolean	True, if forcing is possible and force mode can be activated	
ownForcing	boolean	True, if forcing is performed by REST API at the moment	
forcingClient	string	Current forcing client identifier	
digitalOutForced	array of numbers (2)	The force values of all 16 digital output channels.	
digitalOutMask	array of numbers (2)	The forcing mask of all 16 digital output channels.	
digitalInForced	array of numbers (2)	The force values of all 16 digital input channels.	
digitalInMask	array of numbers (2)	The forcing mask of all 16 digital input channels.	

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)		
iol	array (IOL object)		

Table 34: Root object

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

Table 35: Port mode object

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

Table 36: Digital object

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

Table 37: IOL object

10.3.4 Reading and writing ISDU parameters

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

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

10.3.4.1 Reading ISDU

Method: POST

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

Parameters: port (0-7)

Example:

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

Post-Body: JSON array of read ISDU object

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

Table 38: Read ISDU object

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

Table 39: Read ISDU response object

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

Table 40: Read ISDU data object

10.3.4.2 Writing ISDU

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

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

Table 41: Write ISDU object

Response: Write ISDU response object

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

Table 42: Write ISDU response object

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

Table 43: Write ISDU data object

Attention: For UNITRONIC® ACCESS device variants with HTTPS feature, https:// must be used in front of <ip> for every REST API.

10.3.5 Upload and process an IODD file

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

Perform the following work steps:

1. Check file upload status

   Send request: GET file_upload

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

   Expected response:

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

   Check the status ID. If status is '0', you can start a new iodd upload process. For reference, see tables Table 44: Status ID and meaning on page 148 and Table 45: Error ID and meaning on page 149. Proceed with the next step.

2. Initiate file upload

   Send request: POST file_upload

   Content-Type: application/json

   Purpose: Send details about the file to be uploaded.

   Expected response:

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

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

The action will allways be iodd.

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

The correct content type has to be set.

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

3. Upload file content

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

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

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

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

4. Monitor upload status

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

   Purpose: Get the current file upload status.

   Expected response:

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

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

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

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

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

Table 44: Status ID and meaning

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

Table 45: Error ID and meaning

10.3.6 Example: Reading ISDU

ISDU read request

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

Response

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

10.3.7 Example: Writing ISDU

ISDU write request

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

Response

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

10.4 CoAP server

The CoAP server functions are **only** applicable for the following UNITRONIC® ACCESS variants:

- MP08IOLA08DIO

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

CoAP is based on the Server-Client principle and a service layer protocol that lets nodes and machines communicate with each other. The UNITRONIC® ACCESS multi-protocol 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/HTTPS request. For more information see chapter CoAP configuration - Quick start guide on page 155.

The configuration URL is:

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

The configuration can also read back as a JSON file:

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

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

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

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

Table 46: CoAP configuration

CoAP response:

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

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

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

Examples:

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

{"status": 0}

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

10.4.2 REST API access via CoAP

A connection to the CoAP server running on the UNITRONIC® ACCESS multi-protocol variants can be established via the following URL:

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

For UNITRONIC® ACCESS, the following REST API Requests (JSON format) can be accessed via a CoAP endpoint:

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

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

Table 47: REST API access via CoAP

10.4.3 CoAP configuration - Quick start guide

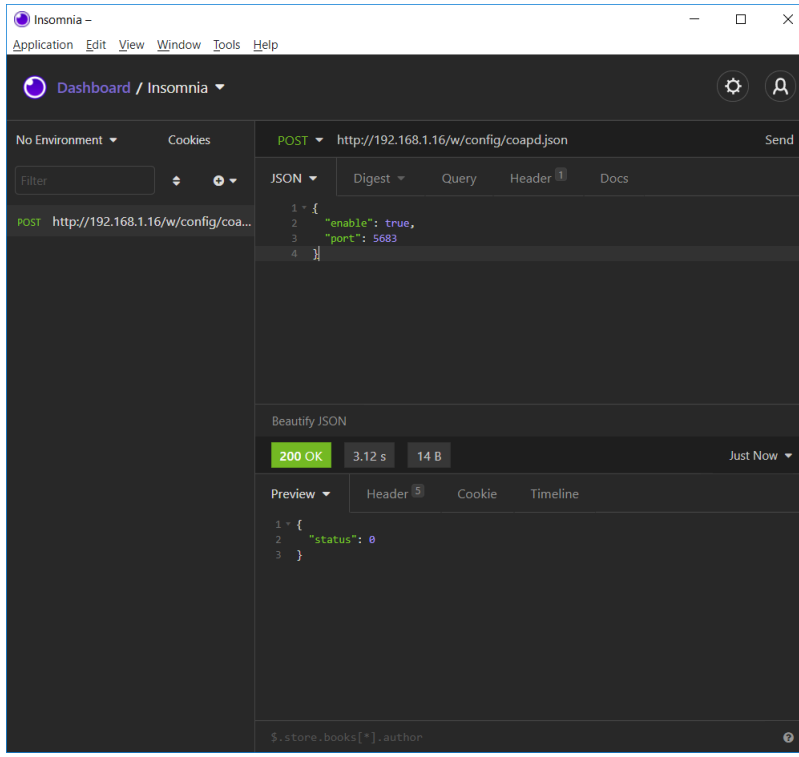
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

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/

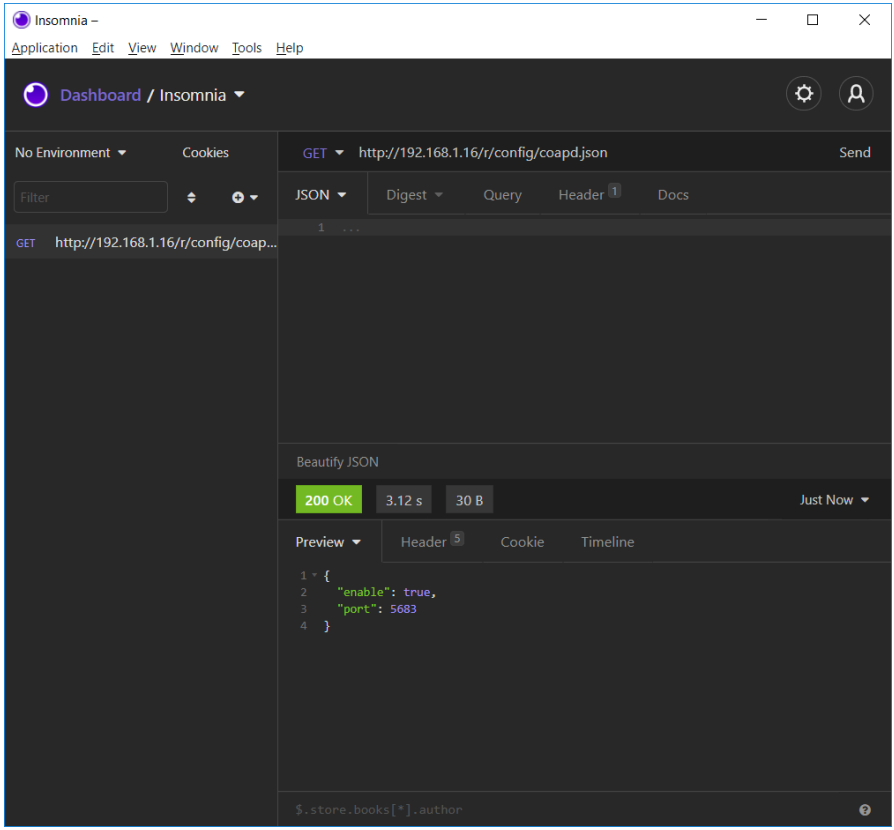
2. Configure CoAP:

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



3. Read CoAP configuration:

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



10.5 Syslog

Syslog functions are **only** applicable for the following UNITRONIC® ACCESS variants:

- MP08IOLA08DIO

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

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

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

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/HTTPS request. For more information see chapter Syslog configuration - Quick start guide on page 160.

The configuration URL is:

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

The configuration can also read back as a JSON file:

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

The configuration is a JSON object. Each JSON member is a configuration element. The object must not contain all elements. Only the provided

elements will be changed. The configuration changes apply only after a device restart.

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

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

Table 48: Syslog configuration

Syslog response:

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

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

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

Examples:

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

{"status": 0}

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

10.5.2 Syslog configuration - Quick start guide

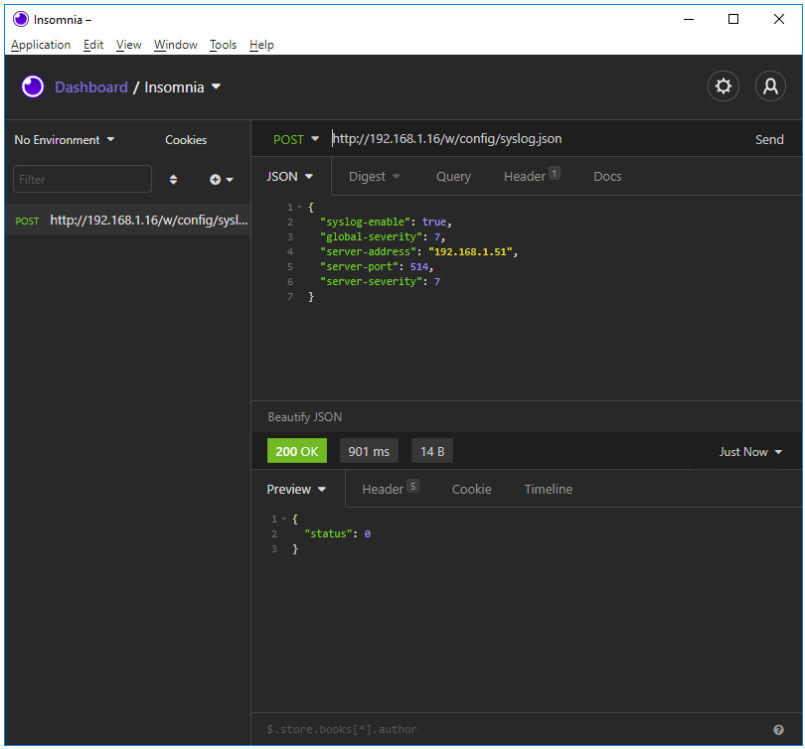
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

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/

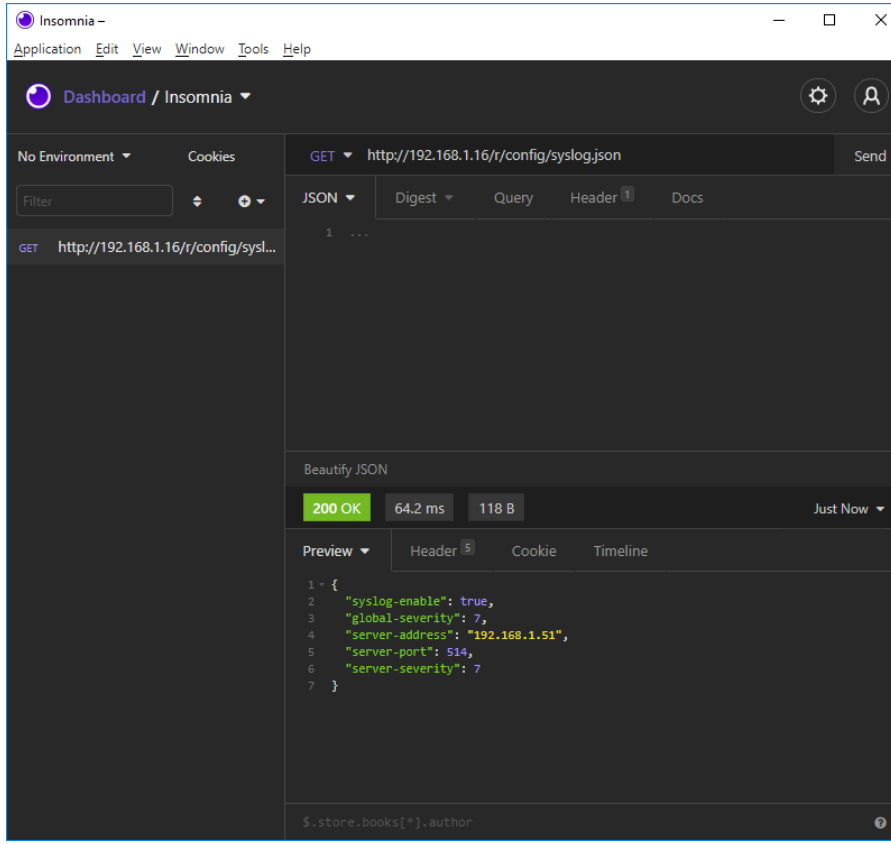
2. Configure Syslog:

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 NTP function is **only** applicable for the following UNITRONIC® ACCESS variant:

- MP08IOLA08DIO

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

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

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/HTTPS request. For more information see chapter NTP configuration - Quick start guide on page 164.

The configuration URL is:

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

The configuration can also read back as a JSON file:

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

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

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

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

Table 49: NTP configuration

NTP response:

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

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

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

Examples:

```
{"status": -1, "error": [{"Element": "ntpc-enable", "Message": "Boolean
 expected"}]}

{"status": 0}

{"status": -1, "error": [{"Element": "root", "Message": "Not a JSON
 object"}]}
```

10.6.2 NTP configuration - Quick start guide

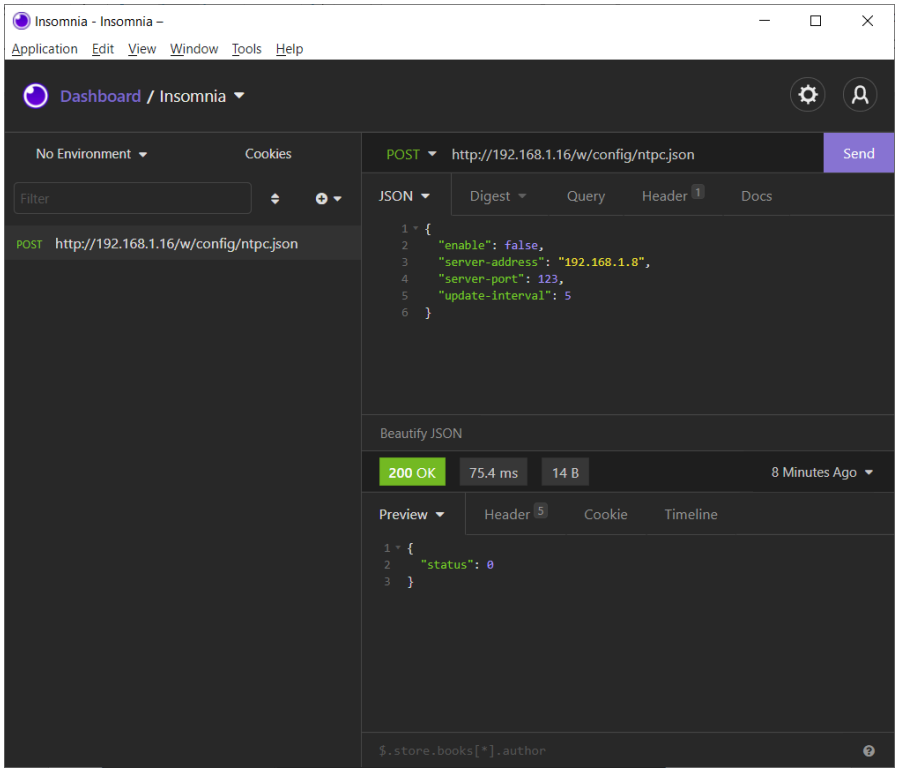
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

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/

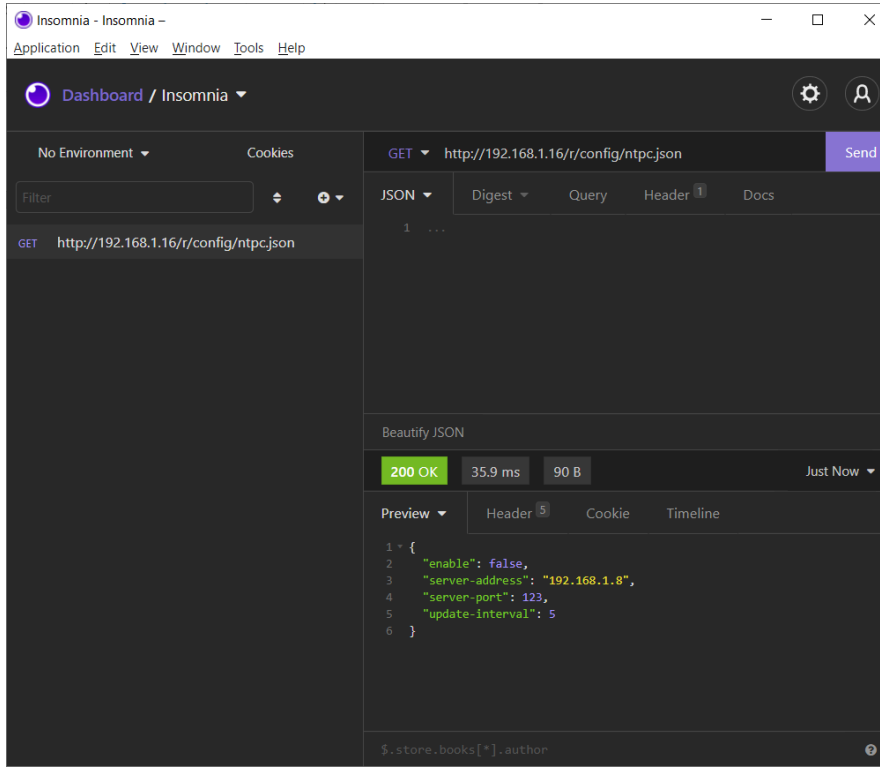
2. Configure NTP:

POST: `[IP-address]/w/config/ntpc.json`



3. Read NTP configuration:

GET: `[IP-address]/r/config/ntpc.json`



11 The integrated Web server

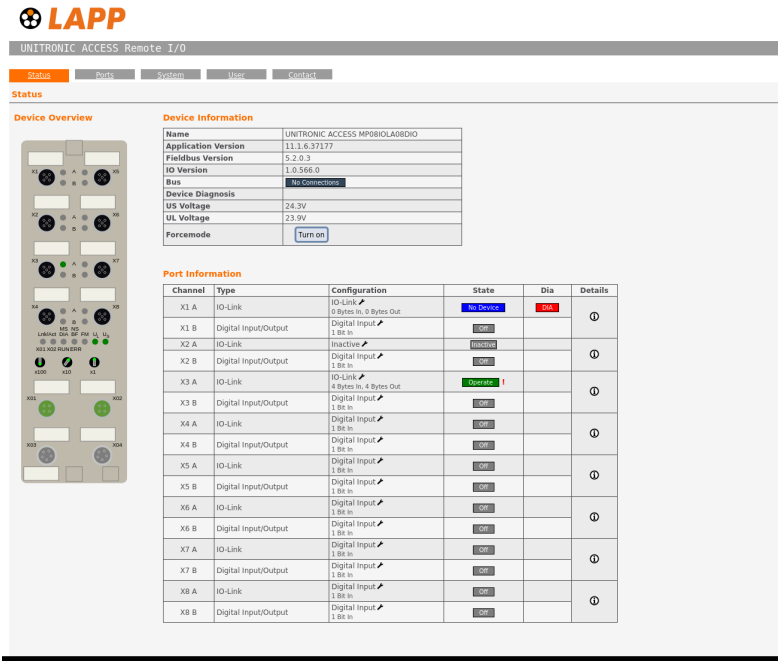
All device variants are equipped with an integrated Web server which makes functions for the device configuration and the display of status and diagnostic information available via a Web interface.

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://" or "https://" followed by the IP address, such as "http://192.168.1.5", in your Web browser’s address bar. If the status page of the device is not displayed, check your browser and firewall settings.

11.1 UNITRONIC® ACCESS MP08... variants

11.1.1 The Status page



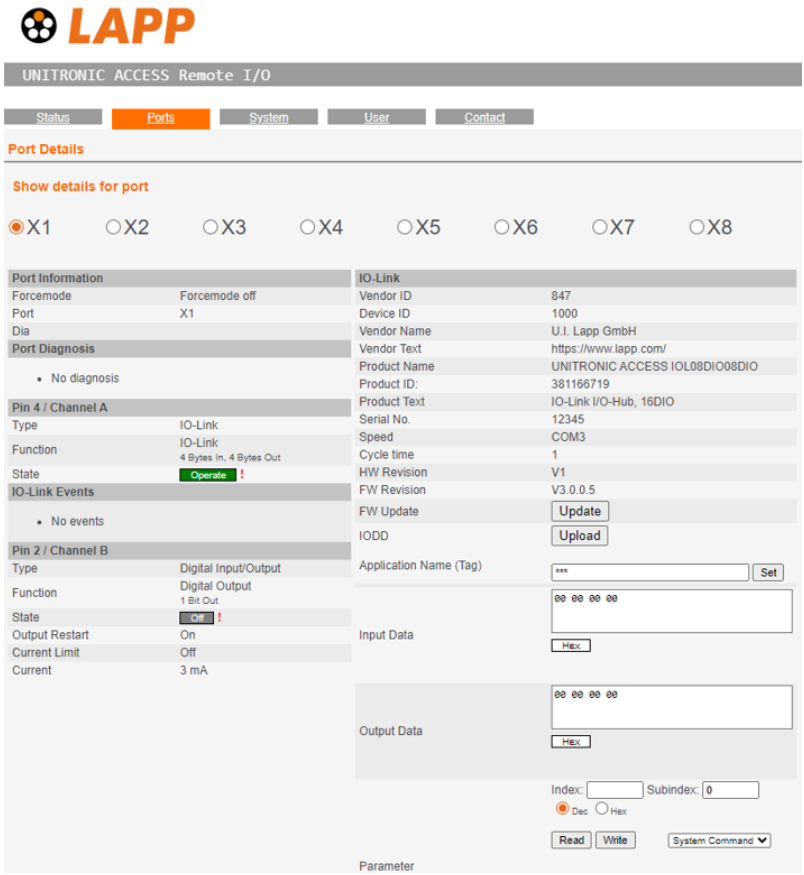
The status page provides a quick overview of the current state of the device.

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



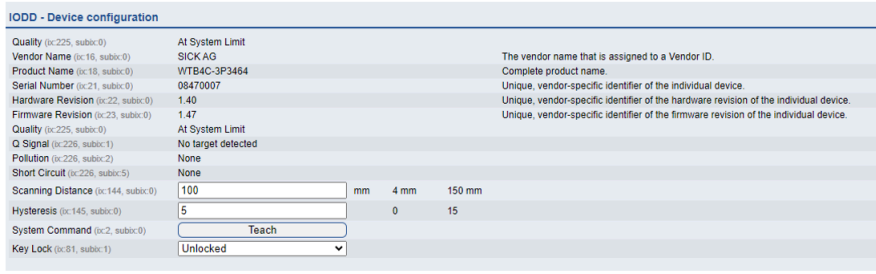
The page shows detailed port information. In the field **Port Diagnosis**, incoming and outgoing diagnostics are displayed as clear text. **Pin 2** and **Pin 4** contain information about the configuration and state of the port. For IO-Link ports, additional information relating to the connected sensor and the process data is displayed.

11.1.2.1 IODD upload

The button **UPLOAD** allows uploading an IODD file into the module, regardless of the device for which the IODD is designed.

The maximum number of IODDs is limited due to storage space. If there is no more space for a new IODD left, there will be a message of the detected error.

With the help of the IODD management page ("System" page), not used IODDs can be deleted. If there is already a matching IODD stored in the system for the connected IO-Link device, the button **CONFIGURE** is shown. By clicking this button, the page "IODD - Device configuration" will open, where the IO-Link device can be configured.



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.

IP Settings

Use this parameter to change the current IP address of the module.

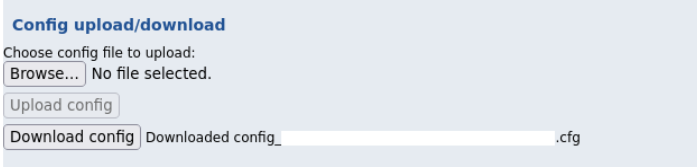
For PROFINET, this is only useful during commissioning. Normally, the PLC sets the IP address at start-up by detecting the PROFINET module via its device name.

11.1.3.1 License

This button opens a new window with Open Source Software information used in this product.

11.1.3.2 Config upload/download

With this feature, settings configured via the Web interface can be stored outside the I/O-Device (Download) for later Upload, e.g. after an I/O-Device change.



The following settings will be stored inside this file:

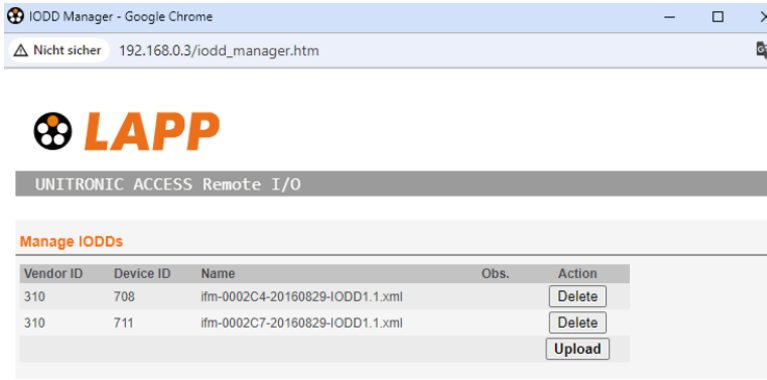
Scope	Type	Setting	Options	Details
Gateway		deviceID		To check device identity.
	iol	applicationSpecificTag		
	iol	functionTag		
	iol	locationTag		
		forcing		Enable/disable forcing
		channel_count		
		network configuration	ip	
			snMask	
			gw	
			source	1 - manual 2 - dhcp 3 - rotary 4 - dcp
Channel		index		channel index starting from 0
		channel configuration		0 - DIO 1 - IN 2 - OUT 3 - IOL 4 - AUX 5 - SAFIN 6 - SAFOUT
	iol	forced		
	iol	simulated		
	iol	force values		array
	iol	simulated		
	iol	sim values		array
	iol	validation	option	validation and backup
			vendorId	
			deviceId	
	digital	force		

Scope	Type	Setting	Options	Details
	digital	force value		
	digital	simulate		
	digital	sim value		
	digital	inputPolarity		
	digital	autorestart mode		
	digital	inputFilter100us		
	digital	currentLimit		
	digital	outputRestartMode		
	digital	failsafeMode		
	digital	surveillanceTimeouMs		
OPC UA		opcua	opcua-enable	
			port	
			anon-allowed	
			commands-allowed	
			force-allowed	
			reset-allowed	
			config-allowed	
	digital		dcu-allowed	
MQTT		mqtt	mqtt-enable	
			broker	
			login	
			password	
			port	
			base-topic	
			will-enable	
			will-topic	
			auto-publish	
			publish-interval	
			publish-identity	

Scope	Type	Setting	Options	Details
			publish-config	
			publish-status	
			publish-process	
	iol		publish-devices	
			commands-allowed	
			force-allowed	
			reset-allowed	
			config-allowed	
			qos	
SYSLOG		syslog	syslog-enable	
			global-severity	
			server-address	
			server-port	
			server-severity	
COAP		coap	enable	
			port	
NTP		ntpc	enable	
			server-address	
			server-port	
			update-interval	

11.1.3.3 IODD

The button **Manage IODDs** opens a new page for the IODD management on the I/O-Device. IODDs can be uploaded or deleted on this page, and all uploaded IODDs are listed here. For configuring connected IO-Link devices, open the related "Ports" page.



11.1.3.4 Restart device

The module initializes a software reset.

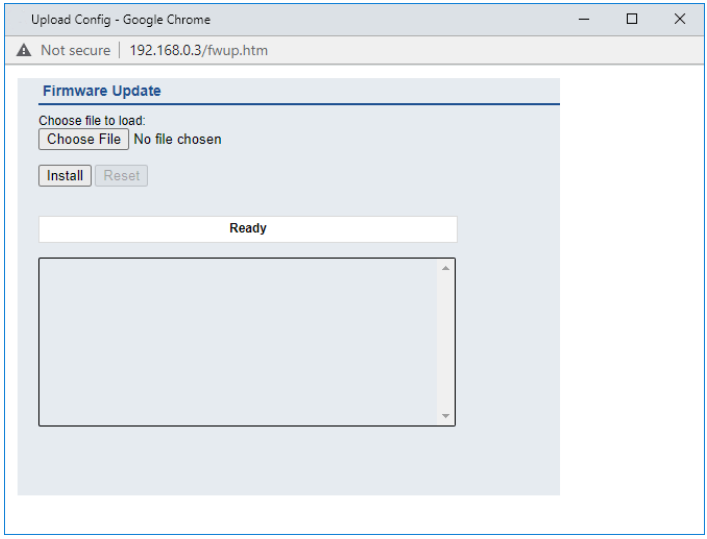
11.1.3.5 Reset configuration to factory defaults

The module restores to the default factory settings.

11.1.3.6 Firmware update

The module initializes a Firmware update.

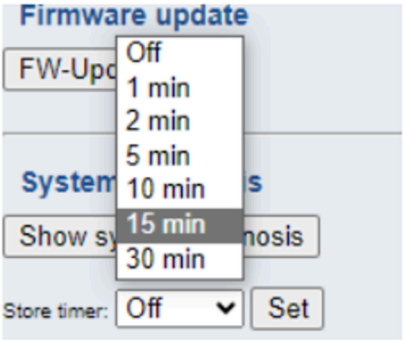
For a firmware update choose the **.ZIP* container available on our website or ask our support team. Afterwards follow the instructions shown on your screen.



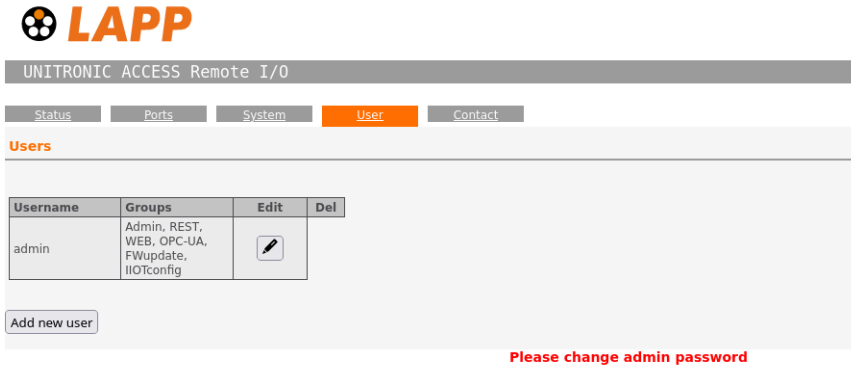
11.1.3.7 System diagnosis

All Syslog messages will be displayed in a ring buffer with 512 entries. By activation of the 'Store timer', the buffer content will be stored nonvolatile in the selected interval of 1, 2, 5, 10, 15 or 30 minutes.

The default value is 'Off' (no nonvolatile storage of system diagnosis ring buffer).



11.1.4 The User page



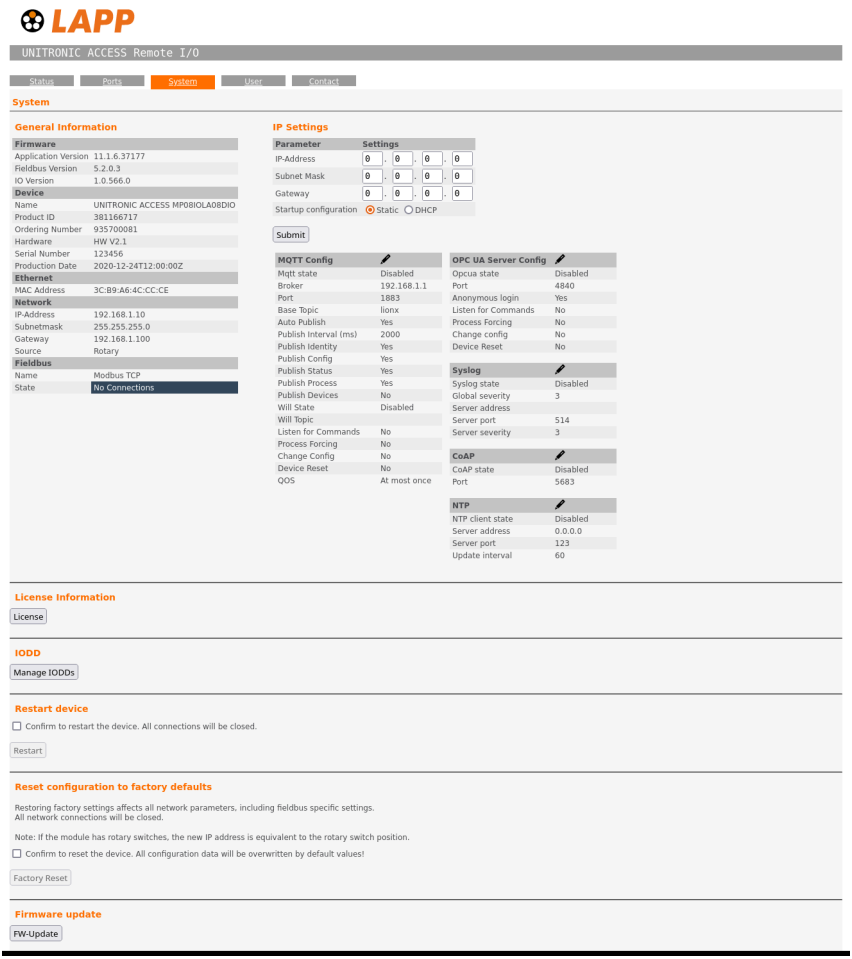
The User page provides the user management of the Web interface. New users with access rights "Admin" or "Write" can be added here. For security reasons please change the default admin password immediately after configuring the device.

Default user login data:

- User: admin
- Password: private

11.2 UNITRONIC® ACCESS single-protocol PN0.../EIP0.../EC0.../MTCP0.../CCL0... variants

11.2.1 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

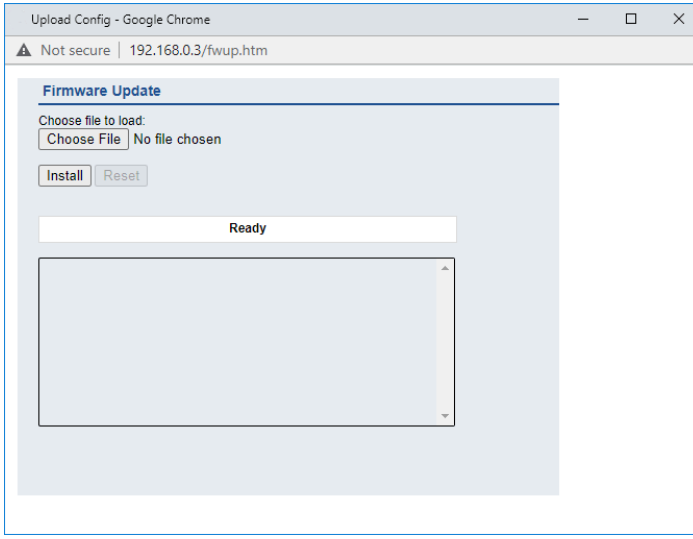
Use this parameter to change the current IP address of the module.

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.



12 IODD

IODD functions are **only** applicable for the following device variants:

- MP08IOLA08DIO

The **IO D**evice **D**escription (IODD) is a set of files formally describing an IO-Link Device. The IODD is created by the vendor and is mandatory for each IO-Link Device.

LAPP UNITRONIC® ACCESS IO-Link Masters with the "IODD on Module" functionality are ready to use IODDs in order to make the IO-Link Device configuration much more easier and the process data human readable in a better way. IODDs can be uploaded via the Web Interface and remanently stored on the IO-Link Master afterwards.

If a corresponding IO-Link Device is connected, the stored IODD is used to provide a user friendly configuration page, where all parameters of the device can be viewed and edited. Additionally, according to the IODD, the process data will also be formatted and displayed to the user.

12.1 IO-Link Device parameters and ISDU requests

Every IO-Link Device provides parameters that can be read and written via the special IO-Link service ISDU (**I**ndexed **S**ervice **D**ata **U**nit).

Every parameter is addressed by an index. Sub-indices are possible but optional. Some parameters (most of them read-only) are mandatory for IO-Link devices an can be found always on the same indices (See *Table B.8* in the *IO-Link Interface and System Specification*: https://io-link.com/share/Downloads/Package-2020/IOL-Interface-Spec_10002_V113_Jun19.pdf).

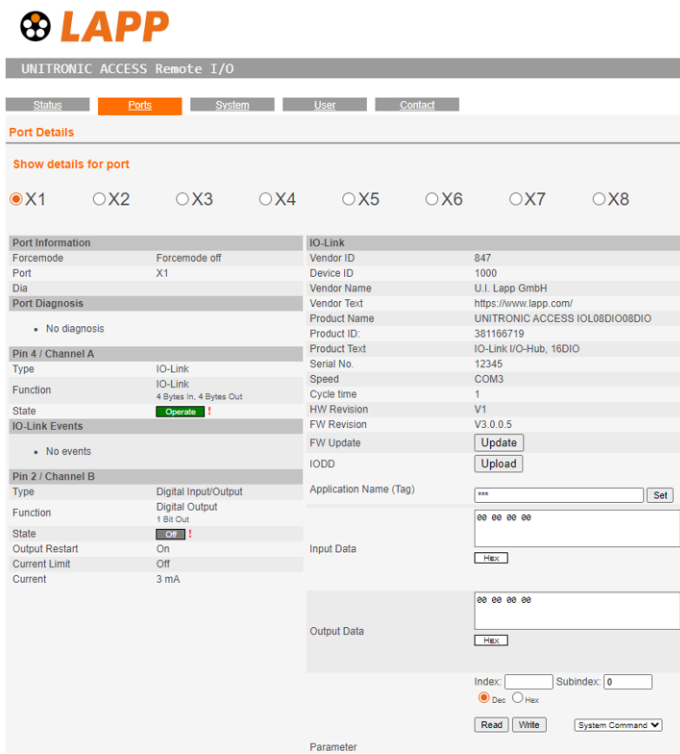
A vendor can use additional parameters and therefore more indices for their devices in order to provide additional configuration options. These vendor specific parameters can be described in an IODD. The "IODD on Module" feature of the UNITRONIC® ACCESS IO-Link Masters can read and parse this information out of an IODD and use it to provide the user viewing

and editing options for vendor specific parameters without any additional knowledge about the vendor specific device features.

12.2 Web GUI functionality

All of the "IODD on Module" features are accessible via the UNITRONIC® ACCESS Web interface.

12.2.1 Port Details page



The Port Details Page shows all information about the selected port. In the left column, all port and channel specific information is displayed. If the port is configured as IO-Link and there is an IO-Link Device connected, all IO-Link information for the connected device is displayed in the right column.

IODD buttons

The row called *IODD* provides access to the "IODD on Module" features. The button *UPLOAD* will let the user upload an IODD file into the module, regardless of the original device the IODD has been designed for.

The maximum number of IODDs is limited due to storage space. If there is no more space left for new IODDs, there will be an error message. In this case, navigate to the IODD Management page to delete IODDs which are no longer used.

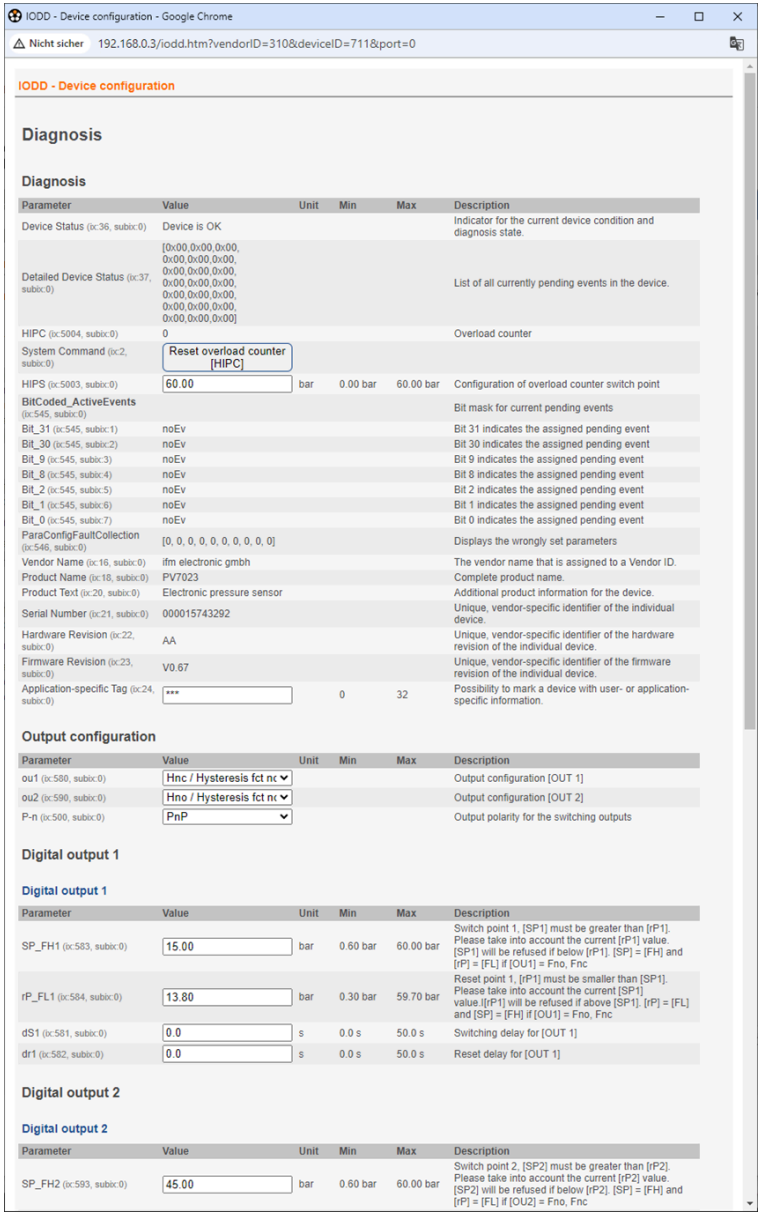
If there is a matching IODD for the currently connected device already stored in the system, the button *CONFIGURE* is shown in the interface. By clicking this button, the Parameter Page will open to configure the device.

Process data

For every connected IO-Link Device, raw process data for input and output direction (set of bytes) is on display.

If a matching IODD providing information about process data is already stored in the system, this data will also be displayed in a user-friendly format according to the IODD.

12.2.2 Parameters page



The parameters page "IODD – Device configuration" shows all parameters which are provided by the IODD of the device. That means the parameter set is variable and depends on the connected IO-Link Device.

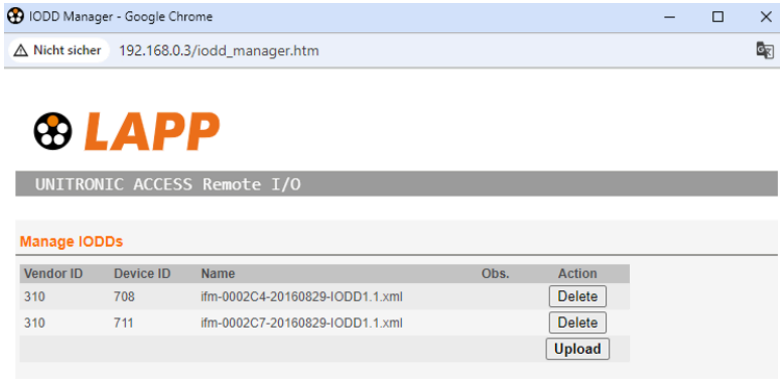
The stored IODD reads the parameter meta data, such as names, units, min/max values, descriptions etc. The values will be obtained directly from the connected device. For that reason it may take several seconds until the page is updated.

If not already saved into the browser, you will be asked for your credentials to continue. A valid user access with Web Interface group membership is needed in order to edit the device parameters. After the registration, enabled values can be changed. Disabled values cannot be changed and may be marked as "read-only" in the IODD. All values are directly written back to the device after any change.

Limitations

- Editing parameter values will directly change them inside the connected device. No parameter server action is triggered by that.
- There is a maximum size of the IODD in order to be uploaded into the system. This depends on several values, such as file size, parameter count, nesting levels etc.

12.2.3 IODD Management page



The IODD Management Page can be accessed via the System page displaying all IODDs that are currently stored in the system. All IODDs

matching connected devices are marked. On the IODD Management page, you can manually delete any IODD in the system.

13 Firmware update

A firmware update of the device is possible via the integrated Web server using the EoE protocol (Ethernet over EtherCAT®), or the FoE protocol (File over EtherCAT®).



Warning: Risk of data loss, damage to the device and injury due to uncontrolled machine movements.

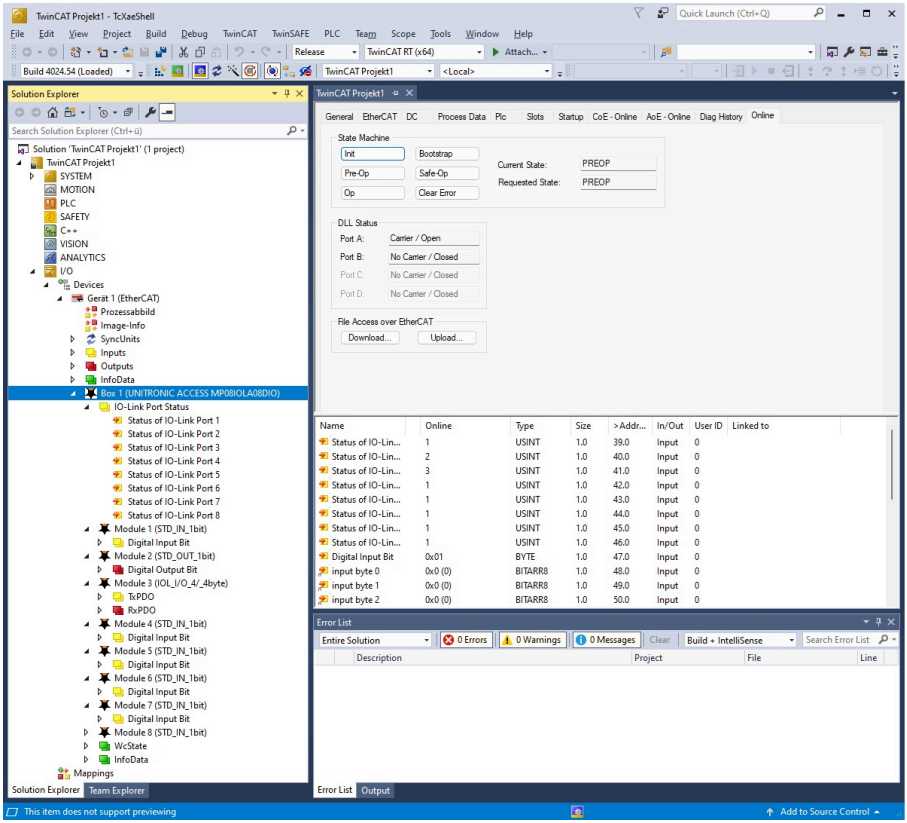
- Do **NOT** interrupt the power supply to the device during the update.

13.1 Firmware update via FoE

The FoE service must be supported by the IOL-Master and the IOL-Device. The FoE service on the UNITRONIC® ACCESS variants is supported by default. If the FoE service is configured and the device is in "Pre-Op" state, an update via FoE can be done by using TwinCAT® like in the following example:

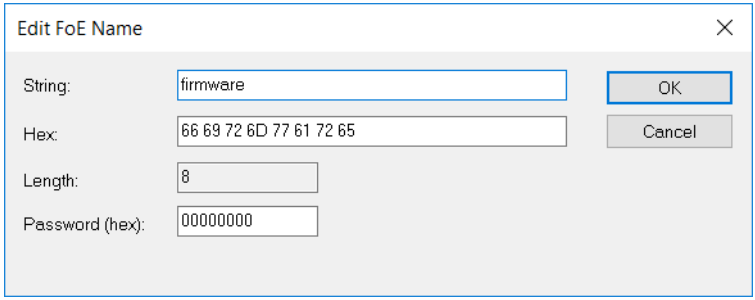
1. Rename the file name of the firmware update file provided by LAPP to “firmware”.

2. In TwinCAT®, select the device you want to update:

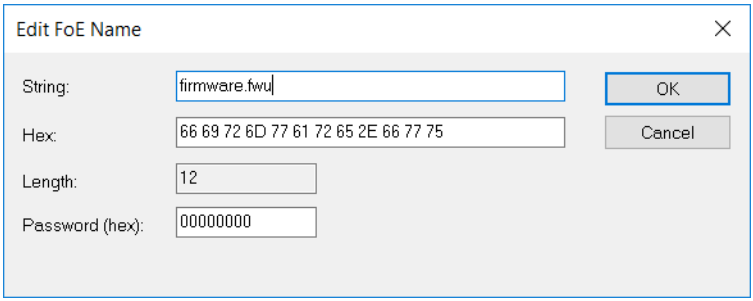


3. In the device window on the right, go to the box **File Access over EtherCAT** and press **Download**.

4. In the following window, select the update file provided by LAPP:


Edit FoE Name
String:
firmware
Hex:
66 69 72 6D 77 61 72 65
Length:
8
Password (hex):
00000000
OK
Cancel

4. Add the file extension ".fwu" to the file name in field **String:** if not visible:



5. Press **OK** and wait until the file has been transferred to the device.

Attention: After the file has been transferred, the device will reset automatically. During the restart, older firmware update files will be replaced by the files in the update package.

14 Technical data

The following sections give an overview of the most important functional data needed to operate the device. For further information and detailed technical data, see the respective **Data Sheet** of your required product in the product specific download area on https://lapp.com.

14.1 General

Protection class (Only applies if the connectors are screwed together or if protective caps are used.)[2]	IP65 IP67 IP69K	
Ambient temperature (during operation and storage)	MP08...	-40 °C .. +70 °C (-40 °F .. +158 °F)
	PN08IOLA.../EIP08IOLA.../EC08IOLA.../ MTCP08IOLA.../CCL08IOLA...	-20 °C .. +60 °C (-4 °F .. +140 °F)
Weight	UNITRONIC® ACCESS 60 mm	approx. 500 gr. (17.6 oz)
Ambient moisture	Max. 98% RH (For UL applications: Max. 80% RH)	
Housing material	Die-cast zinc	
Surface finish	Frosted nickel	
Flammability class	UL 94 (IEC 61010)	
Vibration resistance (oscillation) DIN EN 60068-2-6 (2008-11)	15 g / 5-500 Hz	
Shock resistance DIN EN 60068-2-27 (2010-02)	50 g / 11 ms +/- X,Y,Z	
Fastening torques	M4 fixing screws	1 Nm
	M4 ground connection	1 Nm
	M12 connector	0.5 Nm
Permitted cables	Ethernet cables according to IEEE 802.3, min. CAT 5 (shielded) Max. length of 100 m, not routed out of facility (= local network)	

Table 50: General information

2 Not under UL investigation.

14.2 EtherCAT® protocol

Protocol	EtherCAT® (ETG.1000 V1.2)
ESI file	U.I.Lapp-UnitronicAccess-IO-Link-Master.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 UNITRONIC® ACCESS devices)
Mailbox protocols	CanOpen over EtherCAT® (CoE) File access over EtherCAT® (FoE) Ethernet over EtherCAT® (EoE)
Supported Ethernet protocols	Ping ARP HTTP / HTTPS TCP/IP
Switch functionality	Integrated
EtherCAT® interface Port	2x M12 sockets 4-pin, D-coded (see pin assignments)

Table 51: EtherCAT® protocol

14.3 Power supply of the module electronics/ sensors

Port X03, X04	M12-L-coded Power, connector/socket, 5-pole Pin 1 / Pin 3		
Nominal voltage U_S	24 V DC (SELV/PELV)		
Current U_S	Max. 16 A		
Voltage range	21 .. 30 V DC		
Power consumption of module electronics	Typically 160 mA (+/-20 % at U_S nominal voltage)		
Power supply interruption	Max. 10 ms		
Voltage ripple U_S	Max. 5 %		
Current consumption sensor system (L+ / Pin 1)	MP08...	Port X1 .. X8 (Pin 1)	max. 4 A per port (at $T_{ambient}$ = 30° C)
	PN08IOLA.../ EIP08IOLA.../ EC08IOLA.../ MTCP08IOLA.../ CCL08IOLA...	Port X1 .. X8 (Pin 1)	max. 2 A per port (at $T_{ambient}$ = 30° C)
Voltage level of the sensor power supply	Min. (U_S – 1.5 V)		
Short circuit/overload protection of sensor supply	Yes, per port		
Reverse polarity protection	Yes		
Operational indicator (U_S)	LED green:	18 V (+/- 1 V) < U_S	
	LED red:	U_S < 18 V (+/- 1 V)	

Table 52: Information on the power supply of the module electronics/ sensors

14.4 Power supply of the actuators

14.4.1 IO-Link Class A devices (U_L)

Nominal voltage U_L	24 V DC (SELV/PELV)
Voltage range	18 .. 30 V DC
Current U_L	Max. 16 A
Voltage ripple U_L	Max. 5 %
Reverse polarity protection	Yes
Operational indicator (U_L)	LED green: 18 V (+/- 1 V) < U_L LED red: U_L < 18 V (+/- 1 V) or U_L > 30 V (+/- 1 V) * if "Report U_L supply voltage fault" is enabled.
Port X03, X04	M12_L-coded Power, connector/socket, 5-pole Pin 2 / Pin 4

Table 53: Information on the power supply of the actuators

14.4.2 IO-Link Class A/B devices (U_{AUX})

Nominal voltage U_{AUX}	24 V DC (SELV/PELV)
Voltage range	18 .. 30 V DC
Current U_{AUX}	Max. 16 A
Voltage ripple U_{AUX}	Max. 5 %
Reverse polarity protection	Yes
Electric isolation $U_S \leftrightarrow U_{AUX}$	500 V
Operational indicator (U_{AUX})	LED green: 18 V (+/- 1 V) < U_{AUX} LED red: U_{AUX} < 18 V (+/- 1 V) or U_{AUX} > 30 V (+/- 1 V) * if "Report U_{AUX} supply voltage fault" is enabled.
Port X03, X04	M12_L-coded Power, connector/socket, 5-pole Pin 2 / Pin 4

Table 54: Information on the power supply of the actuators

14.5 I/O ports Channel A (Pin 4)

MP08...	Port X1 .. X8	Class A	IOL, DI, DO	M12 socket, 5-pin, Pin 4
PN08IOLA.../EIP08IOLA.../ EC08IOLA.../ MTCP08IOLA.../ CCL08IOLA...	Port X1 .. X8	Class A	IOL, DI, DO	

Table 55: IO-Link Master ports: Functional overview for Ch. A (Pin 4)

14.5.1 Configured as digital input, Ch. A (Pin 4)

Input connection	MP08... PN08IOLA.../EIP08IOLA.../EC08IOLA.../ MTCP08IOLA.../CCL08IOLA...		Type 1 as per IEC 61131-2
Nominal input voltage	24 V DC		
Input current	Typically 3 mA		
Channel type	Normally open, p-switching		
Number of digital inputs	MP08... PN08IOLA.../EIP08IOLA.../ EC08IOLA.../MTCP08IOLA.../ CCL08IOLA...	X1 .. X8	8
Status indicator	yellow LED		
Diagnostic indicator	red LED per channel		

Table 56: I/O ports Ch. A (Pin 4) configured as digital inputs

14.5.2 Configured as digital output, Ch. A (Pin 4)

Attention: For variant MP08IOLA08DIO, the digital outputs of Channel A is **supplied by the U_L power** when parameterized to "High-Side Switch" mode.

Attention: For variant PN08IOLA08DIO, the digital outputs of Channel A is **supplied by the U_S power**.

Output type	normally open, p-switching (parameterized to "High-Side Switch" mode)	
Nominal output voltage per channel Signal status “1” Signal status “0”	 min. (U_S -1 V) **or** min. (U_L -1 V) depending on the device variant max. 2 V	
Max. output current per device	MP08...	9 A (power supplied via U_L)
Max. output current per device	PN08IOLA.../EIP08IOLA.../ EC08IOLA.../MTCP08IOLA.../ CCL08IOLA...	4 A (power supplied via U_S)
Max. output current per channel[3]	MP08... (X1 .. X8)	2 A (power supplied via U_S)
	PN08IOLA.../EIP08IOLA.../ EC08IOLA.../MTCP08IOLA.../ CCL08IOLA... (X1 .. X8)	0.5 A (power supplied via U_S) 0.25 A for **UL applications**
Short-circuit/overload protected	yes/yes	
Behavior in case of short circuit or overload	deactivation with automatic power-on (parameterized)	
Number of digital outputs	MP08... (X1 .. X8) PN08IOLA.../EIP08IOLA.../ EC08IOLA.../MTCP08IOLA.../ CCL08IOLA... (X1 .. X8)	8
Status indicator	yellow LED per output	
Diagnostic indicator	red LED per channel	

Table 57: I/O ports Ch. A (Pin 4) configured as digital outputs

3 Max. 2.0 A per channel; for every port pair X1/X2, X3/X4, X5/X6, X7/X8 max. 6.5 A (for **UL applications** max. 5.0 A); for the whole port group X1 .. X8 max. 9.0 A in total (with derating).

14.5.3 Configured as IO-Link port in COM mode, Ch. A

IO-Link Master specification	v1.1.3, IEC 61131-9	
Communication rates	4.8 kbaud (COM 1) 38.4 kbaud (COM 2) 230.4 kbaud (COM 3)	
Line lengths in the IO-Link Device	max. 20 m	
Number of IO-Link ports	MP08... (X1 .. X8) PN08IOLA.../EIP08IOLA.../EC08IOLA.../ MTCP08IOLA.../CCL08IOLA... (X1 .. X8)	8
Min. IO-Link cycle time	400 µs	

Table 58: Configured as IO-Link port in COM mode

14.6 I/O ports Channel B (Pin 2)

MP08...	Port X1 .. X8	Class A	DI, DO	M12 socket, 5-pin, Pin 2
PN08IOLA.../EIP08IOLA.../ EC08IOLA.../ MTCP08IOLA.../ CCL08IOLA...	Port X1 .. X8	Class A	DI	

Table 59: IO-Link Master ports: Functional overview for Ch. B (Pin 2)

14.6.1 Configured as a digital input, Ch. B (Pin 2)

Input connection	MP08... PN08IOLA.../EIP08IOLA.../EC08IOLA.../ MTCP08IOLA.../CCL08IOLA...		Type 1 as per IEC 61131-2
Nominal input voltage	24 V DC		
Input current	Typically 3 mA		
Channel type	Normally open, p-switching		
Number of digital inputs	MP08...	X1 .. X8	8
Number of digital inputs	PN08IOLA.../EIP08IOLA.../ EC08IOLA.../MTCP08IOLA.../ CCL08IOLA...	X1 .. X8	8
Status indicator	white LED		
Diagnostic indicator	red LED per channel		

Table 60: I/O ports Ch. B (Pin 2) configured as digital inputs

14.6.2 Configured as a digital output, Ch. B (Pin 2)

Attention: For variant MP08IOLA08DIO, the digital outputs of Channel B is **supplied by the U_L power**.

Attention: For variant PN08IOLA08DIO, the digital outputs of Channel B is **supplied by the U_S power**.

Output type	normally open, p-switching	
Nominal output voltage per channel Signal status “1” Signal status “0”	 min. (U_S -1 V) **or** min. (U_L -1 V) **or** min. (U_{AUX} -1 V) depending on the device variant max. 2 V	
Max. output current per device	MP08...	9 A (power supplied via U_L)
Max. output current per device	PN08IOLA.../ EIP08IOLA.../ EC08IOLA.../ MTCP08IOLA.../ CCL08IOLA...	4 A (power supplied via U_S)
Max. output current per channel [4]	MP08...	2 A (power supplied via U_S)
Max. output current per channel [4]	PN08IOLA.../ EIP08IOLA.../ EC08IOLA.../ MTCP08IOLA.../ CCL08IOLA...	0 A (no outputs on Ch. B)
Short-circuit/overload protected	yes/yes	
Behavior in case of short circuit or overload	deactivation with automatic power-on (parameterized)	
Number of digital outputs	MP08...	8
Number of digital outputs	PN08IOLA.../ EIP08IOLA.../ EC08IOLA.../ MTCP08IOLA.../ CCL08IOLA...	–
Status indicator	white LED per output	
Diagnostic indicator	red LED per channel	

Table 61: I/O ports Ch. B (Pin 2) configured as digital outputs

4 For Class A devices: Max. 2.0 A per channel; for every port pair X1/X2, X3/X4, X5/X6, X7/X8 max. 6.5 A (for **UL applications** max. 5.0 A); for the whole port group X1 .. X8 max. 9.0 A in total (with derating).

14.7 LEDs

LED	Color	Description
U_L/U_{AUX}	Green	Auxiliary sensor/actuator voltage OK 18 V (+/- 1 V) < U_L/U_{AUX} < 30 V (+/- 1 V)
	Red*	Auxiliary sensor/actuator voltage LOW U_L/U_{AUX} < 18 V (+/- 1 V) or U_L/U_{AUX} > 30 V (+/- 1 V) * if "Report U_L/U_{AUX} supply voltage fault" is enabled.
	OFF	None of the above conditions.
U_S	Green	System/sensor voltage OK 18 V (+/- 1 V) < U_S < 30 V (+/- 1 V)
	Red	System/sensor voltage LOW U_S < 18 V (+/- 1 V) or U_S > 30 V (+/- 1 V)
	Red flashing	Device performs a factory reset (position of rotary encoding switches: 9-7-9)
	OFF	None of the above conditions.
X1 .. X8 A	Green	IO-Link COM Mode: IO-Link communication exists.
	Green flashing	IO-Link COM Mode: No IO-Link communication.
	Yellow	Standard-I/O Mode: Status of digital input or output on C/Q (pin 4) line "on".
	OFF	None of the above conditions
X1 .. X8 B	White	Status of digital input or digital output on pin 2 line "on".
	Red	Short circuit on pin 4 and pin 2 line. / All modes: Overload or short circuit on L+ (pin 1) line / communication error
	OFF	None of the above conditions.
P1 Lnk/Act P2 Lnk/Act	Green	Ethernet connection to another subscriber exists. Link detected.
	Yellow flashing	Data exchange with another subscriber.
	OFF	No connection to another subscriber. No link, no data exchange.

LED	Color	Description
BF	Red	Bus fault. No configuration, no or slow physical connection.
	Red flashing at 2 Hz	Link exists but no communication link to the EtherCAT® controller.
	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 flashing at 2 Hz, 3 sec	DCP signal service is initiated via the bus.
	Red double flash	Firmware update
	OFF	None of the above conditions.
MS	Green	Device is ready for operation.
	Green flashing	Device is ready but not configured yet.
	Red	Serious error that cannot be resolved.
	Red flashing	Minor error that can be resolved Example: An incorrect or contradictory configuration is classified as a minor error.
	Flashing alternately: Red / Green	The device is performing a self-test.
	OFF	The device is switched off.

LED	Color	Description
NS	Green	Connected: The device has at least one connection.
	Green flashing	No connection: The device has no connection. IP address exists.
	Red	Duplicate IP address: The device has detected that the assigned IP address is already being used by another device.
	Red flashing	Connection has exceeded time limit or connection interrupted.
	Flashing alternately: Red / Green	The device is performing a self-test.
	OFF	The device is switched off or has not been assigned an IP address.

Table 62: Information on the LED colors

14.8 Data transfer times

The following tables give an overview of the internal data transfer times of the UNITRONIC® ACCESS IO-Link Master with a connected IO-Link Device as digital I/O extension (LAPP article IOL08DIO08DIO 16DIO Hub with a minimum cycle time of 1 ms).

There are three measured data direction values for each use case:

- **PLC to DO:** Transfer of a changed PLC output data to IO-Link Device digital output.
- **DI to PLC:** Transfer of a changed digital input signal on IO-Link Device to PLC.
- **Round-trip time (RTT):** Transfer of a changed PLC output data to IO-Link Device digital output. The digital output is connected to an IO-Link Device digital input. Transfer of the changed digital input signal on IO-Link Device to PLC. RTT = [PLC to DO] + [DI to PLC].

The measured values are taken from the ethernet data transmission line. The values are therefore without PLC processing times and PLC cycle time.

The configurable digital input filter value on IOL08DIO08DIO was set to "off" (0 ms).

For calculation of user specific data transfer and round-trip times of possible input filters, PLC processing and cycles times must be taken into calculation.

The measured values are valid for a maximum of 48 bytes of IO-Link data for the IO-Link Master in each direction (Input/Output).

Use case 1:

IO-Link Master configuration with enabled Web interface and *disabled* IIoT protocols

Data direction	Data transfer time in ms		
	Minimum	Average	Maximum
PLC to DO	3.7	6.0	7.7
DI to PLC	1.1	3.0	4.3
RTT	6.1	8.9	11.1

Use case 2:

IO-Link Master configuration with enabled Web interface and *enabled* IIoT protocols

Data direction	Data transfer time in ms		
	Minimum	Average	Maximum
PLC to DO	7.7	10.0	13.4
DI to PLC	3.3	4.4	5.6
RTT	12.1	14.3	17.0

15 Accessories

In order to get access to various types of accessories, please visit our Web page:

https://www.lapp.com

16 References

1. EtherCAT® Specification - Part 5: Application Layer service definition, ETG.1000.5 V1.0.4, ETG 2017-09.
2. EtherCAT® Specification - Part 6: Application Layer protocol specification, ETG.1000.6 V1.0.4, ETG 2017-09.
3. EtherCAT® Protocol Enhancements, ETG.1020 V1.2.0, ETG 2015-12.
4. EtherCAT® Indicator and Labeling Specification, ETG.1300 V1.1.1, ETG 2015-07.
5. EtherCAT® Slave Information Specification, ETG.2000 V1.0.10, ETG 2018-02.
6. EtherCAT® Modular Device Profile Part 1: General MDP Device Model, ETG.5001.1 V0.9.0, ETG 2016-07.
7. EtherCAT® Modular Device Profile Part 6220: IO-Link Master, ETG.5001.6220 V1.0.5, ETG 2017-04.
8. Protocol API EtherCAT® Slave V4.7.0, Hilscher Gesellschaft für Systemautomation mbH, DOC110909API10EN, Revision 10, 2017-10.
9. IO-Link Interface and System Specification Version 1.1.2, Order No: 10.002, IO-Link Community 2013-07.

Manual

Modbus TCP

UNITRONIC ACCESS IO-Link Master

Multi-protocol:
MP08IOLA08DIO (8 × IO-Link Class A)

Single-protocol:
MTCP08IOLA08DIO (8 × IO-Link Class A)



Contents

1 About this manual 8

1.1 General information 8
1.2 Explanation of symbols 9
1.2.1 Use of danger information 9
1.2.2 Use of general information 9
1.3 Version information 9

2 Safety instructions 10

2.1 Intended use 10
2.2 Qualified personnel 11

3 Designations and synonyms 12

4 System description 16

4.1 About UNITRONIC® ACCESS 16
4.2 Device variants 17
4.3 I/O port overview 19

5 Overview of product features 21

5.1 Modbus TCP product features 21
5.2 I/O port features 22
5.3 Integrated Web server 23
5.4 Security features 24
5.5 Other features 25

6 Assembly and wiring 26

6.1 General information 26
6.2 Outer dimensions 27
6.2.1 UNITRONIC® ACCESS multi-protocol variants 27
6.2.2 UNITRONIC® ACCESS single-protocol variants with Modbus TCP 28
6.2.3 Notifications 29
6.3 Port assignments 30
6.3.1 Ethernet ports, M12 socket, 4-pin, D-coded 30
6.3.2 Power supply with M12 power L-coded 31
6.3.2.1 IO-Link Master with Class A ports 32
6.3.2.2 IO-Link Master with Class A/B ports 33
6.3.3 I/O ports as M12 sockets 33
6.3.3.1 IO-Link ports (Class A) 34

7 Starting operation 35

7.1 Device identification 35
7.2 MAC addresses 35
7.3 Modbus function codes 36
7.4 State on delivery 36
7.5 Setting the rotary encoding switches 37
7.5.1 Modbus TCP 40
7.5.2 Factory reset 40
7.6 Setting network parameters 41

8 Configuration Modbus TCP 42

8.1 Configuration parameters 42
8.2 General settings 43
8.2.1 Force mode lock 44
8.2.2 Web interface lock 44

8.2.3 Report U_L/U_{AUX} supply voltage fault 44
8.2.4 Report actuator fault without U_L/U_{AUX} voltage 44
8.2.5 Report U_S voltage fault 44
8.2.6 External configuration lock 44
8.3 Channel settings 45
8.3.1 DO Surveillance Timeout 47
8.3.2 DO Failsafe 47
8.3.3 DO Restart Mode 47
8.3.4 DI Logic 48
8.3.5 DI Filter 48
8.3.6 DI Latch 48
8.3.7 DI Extension 50
8.4 IO-Link Port 1 .. 8 settings 51
8.4.1 Port Mode 54
8.4.2 Validation and Backup 55
8.4.3 IQ Mode 58
8.4.4 Cycle Time 58
8.4.5 Vendor ID 59
8.4.6 Device ID 59
8.4.7 Swapping Mode 59
8.4.8 Swapping Offset 60
8.4.9 IOL Failsafe 60
8.4.10 IOL Failsafe replacement values 60
8.4.11 Serial number 61

9 Process data assignment 62

9.1 Consuming data (output) 63
9.2 Producing data (input) 64
9.3 Channel B as digital output 65
9.4 Channel B as digital input 66

10 Diagnostics processing 67

10.1 Error of the system/sensor power supply 69
10.2 Error of the auxiliary/actuator power supply 69
10.3 Overload/short-circuit of the I/O port sensor supply outputs 70
10.4 Overload/short circuit of the digital outputs 70
10.5 Overload/short circuit of Actuator voltage supply P24 (Class B) 71
10.6 IO-Link channel status 71
10.7 IO-Link channel events 73

11 IIoT functionality 77

11.1 MQTT 78
11.1.1 MQTT configuration 78
11.1.2 MQTT topics 81
11.1.2.1 Base topic 81
11.1.2.2 Publish topic 84
11.1.2.3 Command topic (MQTT Subscribe) 89
11.1.3 MQTT configuration - Quick start guide 93
11.1.3.1 MQTT configuration via JSON 93
11.2 OPC UA 95
11.2.1 OPC UA configuration 96
11.2.2 OPC UA address space 98
11.2.3 OPC UA configuration - Quick start guide 99
11.2.3.1 OPC UA configuration via JSON 99
11.3 REST API 101
11.3.1 Standard device information 103
11.3.2 Structure 104
11.3.3 Configuration and forcing 110
11.3.4 Reading and writing ISDU parameters 113
11.3.4.1 Reading ISDU 113
11.3.4.2 Writing ISDU 115
11.3.5 Upload and process an IODD file 117
11.3.6 Example: Reading ISDU 121
11.3.7 Example: Writing ISDU 121
11.4 CoAP server 122
11.4.1 CoAP configuration 122

11.4.2 REST API access via CoAP 123
11.4.3 CoAP configuration - Quick start guide 126
11.4.3.1 CoAP configuration via JSON 126
11.5 Syslog 128
11.5.1 Syslog configuration 128
11.5.2 Syslog configuration - Quick start guide 131
11.5.2.1 Syslog configuration via JSON 131
11.6 Network Time Protocol (NTP) 133
11.6.1 NTP configuration 133
11.6.2 NTP configuration - Quick start guide 135
11.6.2.1 NTP configuration via JSON 135

12 The integrated Web server 137

12.1 UNITRONIC® ACCESS MP08... variants 138
12.1.1 The Status page 138
12.1.2 The Ports page 139
12.1.2.1 IODD upload 139
12.1.3 The System page 141
12.1.3.1 License 142
12.1.3.2 Config upload/download 142
12.1.3.3 IODD 145
12.1.3.4 Restart device 146
12.1.3.5 Reset configuration to factory defaults 146
12.1.3.6 Firmware update 146
12.1.3.7 System diagnosis 147
12.1.4 The User page 148
12.2 UNITRONIC® ACCESS single-protocol PN0.../EIP0.../EC0.../MTCP0.../CCL0... variants 149
12.2.1 The System page 149

13 IODD 151

13.1 IO-Link Device parameters and ISDU requests 151

13.2 Web GUI functionality 152
13.2.1 Port Details page 152
13.2.2 Parameters page 154
13.2.3 IODD Management page 155

14 Technical data 157

14.1 General 158
14.2 Modbus TCP protocol 159
14.3 Power supply of the module electronics/sensors 160
14.4 Power supply of the actuators 161
14.4.1 IO-Link Class A devices (U_L) 161
14.4.2 IO-Link Class A/B devices (U_{AUX}) 161
14.5 I/O ports Channel A (Pin 4) 162
14.5.1 Configured as digital input, Ch. A (Pin 4) 162
14.5.2 Configured as digital output, Ch. A (Pin 4) 163
14.5.3 Configured as IO-Link port in COM mode, Ch. A 165
14.6 I/O ports Channel B (Pin 2) 166
14.6.1 Configured as a digital input, Ch. B (Pin 2) 166
14.6.2 Configured as a digital output, Ch. B (Pin 2) 167
14.7 LEDs 169
14.8 Data transfer times 172

15 Accessories 174

1 About this manual

1.1 General information

Read the assembly and operating instructions on the following pages carefully before starting up the modules. Keep this information where it is accessible to all users.

The texts, figures, diagrams, and examples used in this document are exclusively used to explain how to operate and apply the modules.

Please contact us if you have any detailed questions on installing and starting up the devices.

U.I. Lapp GmbH

Schulze-Delitzsch-Straße 25
D-70565 Stuttgart
Germany

mailto:info@lapp.com
https://www.lapp.com

U.I. Lapp GmbH reserves the right to make technical changes or changes to this document 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.

1.3 Version information

Version	Created	Changes
1.0	10/2024	

Table 1: Overview of manual revisions

2 Safety instructions

2.1 Intended use

The products described in this manual are decentralized IO-Link Masters on an Industrial Ethernet Network.

We adhere to all safety standards when developing, producing, testing, and documenting our products. When you adhere to the handling specifications and safety instructions described for the configuration, assembly, and correct operation, there should not normally be any risks for people or equipment.

The modules fulfill the requirements of the EMC guidelines (2014/30/EU) and the low voltage guideline (2014/35/EU).

The IO-Link Masters are designed to be used in the industrial sector. The industrial environment is distinguished by the fact that the consumer is not connected directly to the public low voltage network. Additional measures are required for use in residential areas or in business and commercial sectors.

Attention: This equipment may cause radio interference in residential areas. In this case the operator may be requested to carry out appropriate measures.

The proper and safe operation of this product depends on proper transportation, storage, assembly, and installation, and careful operation.

A completely assembled device housing is required for the proper operation of the IO-Link Masters. Only connect devices that fulfill the requirements of EN 61558-2-4 and EN 61558-2-6 to the IO-Link Masters.

During the configuration, installation, start-up, maintenance, and testing of the devices, adhere to the safety and accident-prevention guidelines for the specific application.

Only install cables and accessories that fulfill the requirements and regulations for safety, electromagnetic compatibility, and, where applicable, telecommunication end devices, as well as the specification information.

Information on which cables and accessories are permitted for the installation can be obtained from U.I. Lapp GmbH or is contained in this manual.

2.2 Qualified personnel

The configuration, installation, start-up, maintenance, and testing of the devices may only be performed by a qualified electrician who is familiar with the safety standards of the automation technology.

The personnel requirements are based on the requirement profiles described by ZVEI, VDMA, or equivalent organizations.

Only electricians who are familiar with the content of all provided device documentation are authorized to install and maintain the devices described. These are persons who

- based on their technical training, knowledge, and experience, and their knowledge of the pertinent standards, can evaluate the work to be carried out and identify any potential risks or
- based on working for several years in a related sector, have the same level of knowledge as they would have from the relevant technical training.

Only U.I. Lapp GmbH is permitted to make changes to the hardware or software of the products that go beyond the scope of this manual.



Warning: Making unqualified changes to the hardware or software, or non-adherence to the warning information contained in this document, can result in serious personal injury or damage to equipment.



Attention: U.I. Lapp GmbH 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 and IO-Link)
BUI	Back-Up Inconsistency (EIP diagnostics)
CC	CC-Link IE Field
C/Q	I/O port pin 4 mode, IO-Link communication/switching signal
Ch. A	Channel A (Pin 4) of I/O port
Ch. B	Channel B (Pin 2) of I/O port
CIP	Common Industrial Protocol (media independent protocol)
CIP Safety™	Common Industrial Protocol for Safety applications, CIP Safety™ is a registered trademark of ODVA
Class A	IO-Link port specification (Class A)
Class B	IO-Link port specification (Class B)
CoAP	Constrained Application Protocol
CSP+	Control & Communication System Profile Plus
DAT	Device Acknowledgement Time
DCP	Discovery and Configuration Protocol
DevCom	Device Comunicating (EIP diagnostics)
DevErr	Device Error (EIP diagnostics)
DI	Digital Input
DIA	Diagnostic LED
DO	Digital Output
DIO	Digital Input/Output
DTO	Device Temperature Overrun (EIP diagnostics)
DTU	Device Temperature Underrun (EIP diagnostics)
DUT	Device under test

EIP	EtherNet/IP™ is a registered trademark of ODVA
ERP	Enterprise Resource Planning system
ETH	ETHERNET
FE	Functional Earth
FME	Force Mode Enabled (EIP diagnostics)
FS	Functional Safety
FSU	Fast Start-Up
GSDML	General Station Description Markup Language
High-B	High-Byte
HTTPS	Hyper Text Transfer Protocol Secure
ICE	IO-Link port COM Error (EIP diagnostics)
ICT	Invalid Cycle Time (EIP diagnostics)
IDE	IO-Link port Device Error (EIP diagnostics)
IDN	IO-Link port Device Notification (EIP diagnostics)
IDW	IO-Link port Device Warning (EIP diagnostics)
IIoT	Industrial Internet of Things
ILE	Input process data Length Error (EIP diagnostics)
IME	Internal Module Error (EIP diagnostics)
I/O	Input / Output
I/O port	X1 .. X8
I/O port pin 2	Channel B of I/O ports
I/O port pin 4 (C/Q)	Channel A of I/O ports
IODD	I/O Device Description
IOL or IO-L	IO-Link
I/Q	I/O port pin 2 mode, Digital Input/switching signal
ISDU	Indexed Service Data Unit
IVE	IO-Link port Validation Error (EIP diagnostics)
I&M	Identification & Maintenance
JSON	JavaScript Object Notation (platform independent data format)
L+	I/O port pin 1, sensor power supply

UNITRONIC® ACCESS 60	UNITRONIC® ACCESS variants with a width of 60mm
Little Endian	Data format with Low-B on first place (EtherNet/IP)
LLDP	Link Layer Discovery Protocol
Low-B	Low-Byte
LSB	Least Significant Bit
LVA	Low Voltage Actuator Supply (EIP diagnostics)
LVS	Low Voltage System/Sensor Supply (EIP diagnostics)
MIB	Management Information Base
MP	Multi-protocol: PROFINET + EtherNet/IP + EtherCAT® + Modbus TCP (+ CC-Link IE Field Basic)
MQTT	Message Queuing Telemetry Transport (open networking protocol)
MSB	Most Significant Bit
M12	Metric thread according to DIN 13-1 with 12 mm diameter
NTP	Network Time Protocol
OFDT	One Fault Delay Time
OLE	Output process data Length Error (EIP diagnostics)
OPC UA	Open Platform Communications Unified Architecture (platform independent, service-oriented architecture)
PFH	Probability of dangerous Failure per Hour [h -1]
PD	Process Data
PDCT	Port and Device Configuration Tool
PLC	Programmable Logic Controller
PN	PROFINET
PWR	Power
Qualifier	Validity on a process value. Valid = "1"
REST	REpresentational State Transfer
RFC	Request for Comments
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)
SFRT	Safety Function Response Time
SIO mode	Standard Input Output mode
SLMP	Seamless Message Protocol
SNMP	Simple Network Management Protocol
SP	Single Protocol (PROFINET, EtherNet/IP, EtherCAT®, Modbus TCP or CC-Link IE Field Basic)
SPE	Startup Parameterization Error (EIP diagnostics)
T-B	Test Channel B
T-A	Test Channel A
U_{AUX}	$U_{Auxiliary}$, supply voltage for the load circuit (Actuator supply on Class B ports of Class A/B IO-Link Master)
UDP	User Datagram Protocol
UDT	User-Defined Data Types
UINT8	Byte in PLC (IB, QB)
UINT16	Unsigned integer with 16 bits or word in PLC (IW, QW)
U_L	U_{Load}, supply voltage for the load circuit (Actuator supply on Class A IO-Link-Master)
UL	Underwriters Laboratories Inc. (certification company)
UTC	Coordinated Universal Time (Temps Universel Coordonné)
WCDT	Worst Case Delay Time

Table 2: Designations and synonyms

4 System description

The Lapp U.I. UNITRONIC® ACCESS modules function as the interface in an industrial Ethernet system: A central controller on the management level is able to communicate with the decentralized sensors and actuators on the field level. The line or ring topologies for which UNITRONIC® modules can be used ensure not only reliable data communication but also significantly reduce the number of cables required and thus also the costs for installation and maintenance. They additionally enable easy and quick extension.

4.1 About UNITRONIC® ACCESS

The UNITRONIC® ACCESS device variants convert standard input, standard output or IO-Link signals from sensors & actuators into an industrial Ethernet protocol (PROFINET, EtherNet/IP, EtherCAT®, Modbus TCP, CC-Link IE Field Basic) and/or into a cloud protocol (REST API, CoAP, OPC UA, MQTT). For the first time, there is now Syslog on board. The robust 8 port housing design allows the use even in harsh environments where e.g. weld field immunity, high temperature ranges or protection class IP67 & IP69K are needed.

4.2 Device variants

The following variants are available in the UNITRONIC® ACCESS Multi-protocol and the UNITRONIC® ACCESS single-protocol family:

Article number	Product designation	Description	I/O port functionality
381166717	MP08IOLA08DIO	UNITRONIC® ACCESS M12-60 mm, IO-Link Master Multi-protocol (PN, EIP, EC, MB, CC) Security	8 x IO-Link Class A
381166712	PN08IOLA08DIO	UNITRONIC® ACCESS single-protocol M12-60 mm, IO-Link Master PROFINET	8 x IO-Link Class A
381166713	EIP08IOLA08DIO	UNITRONIC® ACCESS single-protocol M12-60 mm, IO-Link Master EtherNet/IP	8 x IO-Link Class A
381166714	EC08IOLA08DIO	UNITRONIC® ACCESS single-protocol M12-60 mm, IO-Link Master EtherCAT®	8 x IO-Link Class A
381166715	MTCP08IOLA08DIO	UNITRONIC® ACCESS single-protocol M12-60 mm, IO-Link Master Modbus TCP	8 x IO-Link Class A
381166716	CCL08IOLA08DIO	UNITRONIC® ACCESS single-protocol M12-60 mm, IO-Link Master CC-Link IE Field Basic	8 x IO-Link Class A

Table 3: Overview of UNITRONIC® ACCESS Multi-protocol and UNITRONIC® ACCESS single-protocol variants

4.3 I/O port overview

The following tables show the main I/O port differences of the UNITRONIC® ACCESS IO-Link-Master family. Pin 4 and Pin 2 of the I/O ports can be configured partly to IO-Link, Digital Input or Digital Output.

UNITRONIC® ACCESS Class A IO-Link ports

Device variant	Port	Pin 1 U_S	Pin 4 / Ch. A (C/Q)				Pin 2 / Ch. B (I/Q)	
MP08...	**Info:**	–	Class A	Type 1	Supply by U_S [1)]	Supply by U_L [2)]	Type 1	Supply by U_L [2)]
	X8:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)
	X7:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)
	X6:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)
	X5:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)
	X4:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)
	X3:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)
	X2:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)
	X1:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)

Table 4: Port configuration of MP08... variants

[1)] DO switch mode configured as "Push-Pull" (description in the configuration chapters).

[2)] DO switch mode configured as "High-Side" (description in the configuration chapters).

UNITRONIC® ACCESS single-protocol Class A IO-Link ports

Device variant	Port	Pin 1 U_S	Pin 4 / Ch. A (C/Q)			Pin 2 / Ch. B (I/Q)
PN08IOLA.../ EIP08IOLA.../ EC08IOLA.../ MTCP08IOLA.../ CCL08IOLA...	**Info:**	–	Class A	Type 1	Supply by U_S [1)]	Type 1
	X8:	Out (2 A)	IOL	DI	DO (0.5 A*)	DI
	X7:	Out (2 A)	IOL	DI	DO (0.5 A*)	DI
	X6:	Out (2 A)	IOL	DI	DO (0.5 A*)	DI
	X5:	Out (2 A)	IOL	DI	DO (0.5 A*)	DI
	X4:	Out (2 A)	IOL	DI	DO (0.5 A*)	DI
	X3:	Out (2 A)	IOL	DI	DO (0.5 A*)	DI
	X2:	Out (2 A)	IOL	DI	DO (0.5 A*)	DI
	X1:	Out (2 A)	IOL	DI	DO (0.5 A*)	DI

Table 5: Port configuration of PN08IOLA.../EIP08IOLA.../EC08IOLA.../ MTCP08IOLA.../CCL08IOLA... variants

[1)] With DO Switch Mode configured as "Push-Pull" (see description in the configuration chapters).

* For **UL applications**: Max. 0.25 A DO.

5 Overview of product features

5.1 Modbus TCP product features

Modbus Mode

Features MODBUS server via a standard TCP network. The number of allowed operations for holding registers is dependent on the device configuration. The device supports 4 up to 8 TCP sockets for communication.

Data connection

The connection option provided by UNITRONIC® ACCESS is the widely-used M12 connector with D-coding for the Modbus TCP network.

The connectors are also color-coded to prevent the ports from being mixed up.

Data transmission rates

Featuring a transmission rate of up to 100 MBit/s, the Modbus TCP devices can handle both fast transmission of I/O data and transmission of larger volumes of data.

Diagnostic data

The devices support diagnosis flags and extended diagnostic data that can be appended to the I/O data.

5.2 I/O port features

IO-Link specification

UNITRONIC® ACCESS is ready for IO-Link specification v1.1.3.

8 x IO-Link Master ports

Depending on the device variant, the device has 4 IO-Link Class A ports, 4 IO-Link Class A ports and 4 IO-Link Class B ports, or 8 IO-Link Class A ports with an additional digital input and optional output on pin 2 of the I/O port. For detailed information see chapter I/O port overview on page 19.



Warning: If modules with electric isolation and modules without electric isolation are used within the same system, the electric isolation of all connected modules is annulled.

IO-Link port connections

The IO-Link port connection option provided by the module series is the 5-pin M12 connector.

Validation & Backup

The Validation & Backup function checks if the right device is connected and stores / monitors the parameters of the IO-Link Device. The function thus gives you an easy option for replacing the IO-Link Device.

This is possible as of IO-Link specification V1.1 and only if the IO-Link Device **and** the IO-Link Master support the function.

LED

You can see the status of a port by the color of the matching LEDs and their flash pattern. For details on the meanings of the LED colors, please see section LEDs on page 169.

5.3 Integrated Web server

Network parameter display

Get an overview of network parameters such as the IP address, subnet mask and gateway.

Displaying diagnostics

View diagnostics via the integrated Web server.

User management

Use the integrated Web server for convenient management of all users.

IO-Link Device parameters

Reading and writing of IO-Link Device parameters is supported. The system command Store parameters is needed after parameter writing, to take over the changed parameter into the IO-Link Master backup memory when enabled.

5.4 Security features

Firmware signature

The official firmware update packages contain a signature which helps prevent the system against manipulated firmware updates.

Syslog

The UNITRONIC® ACCESS multi-protocol variants support the traceability of messages centrally managed and logged via Syslog.

User manager

The Web server provides a user manager to help protect the Web interface against unauthorized access. You can manage the users by groups with different access levels “Admin” or “Write”.

Default user settings:

User: admin

Password: private

Attention: Change the default settings to help protect the device against unauthorized access.

5.5 Other features

Interface protection

The devices have reverse polarity, short-circuit and overload protection for each interface.

For more details, see section Port assignments on page 30.

Failsafe

The devices support a failsafe function. This allows you to define the behavior of every single channel configured as an output in the case of invalid PLC data (e.g. PLC in STOP) or of lost PLC communication.

Industrial Internet of Things

UNITRONIC® ACCESS 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 UNITRONIC® ACCESS 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 UNITRONIC® ACCESS multi-protocol variants

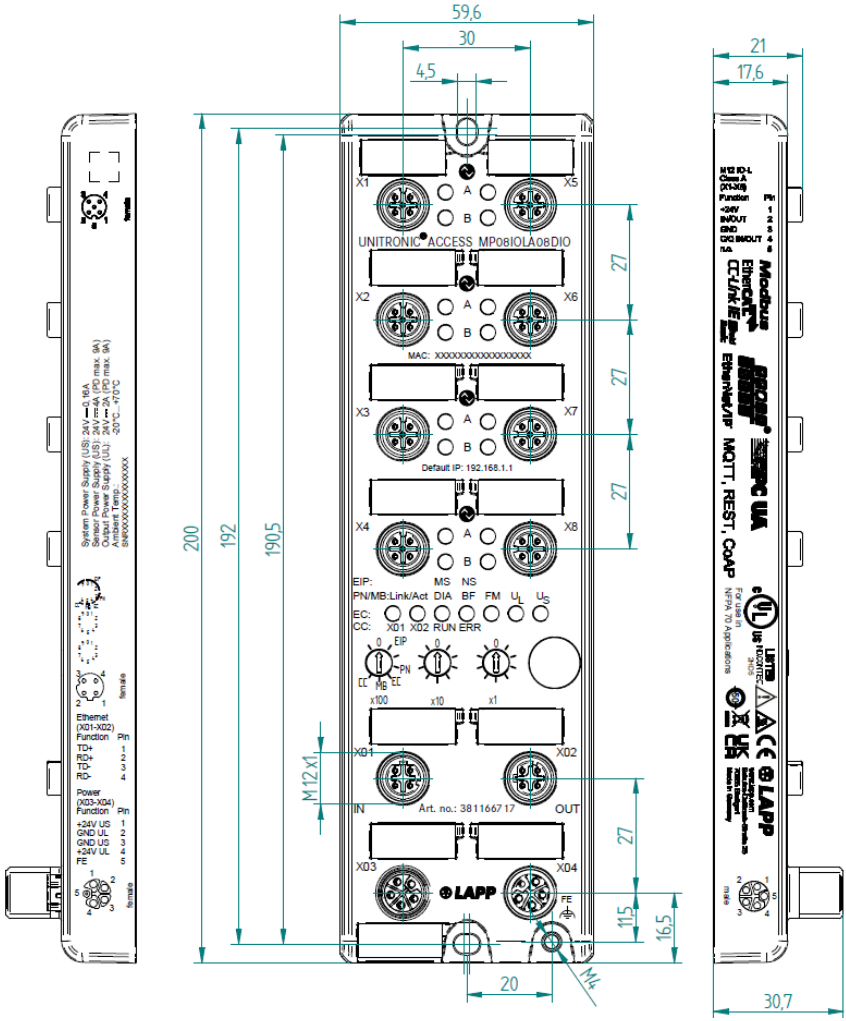


Figure 1: MP08IOLA08DIO

6.2.2 UNITRONIC® ACCESS single-protocol variants with Modbus TCP

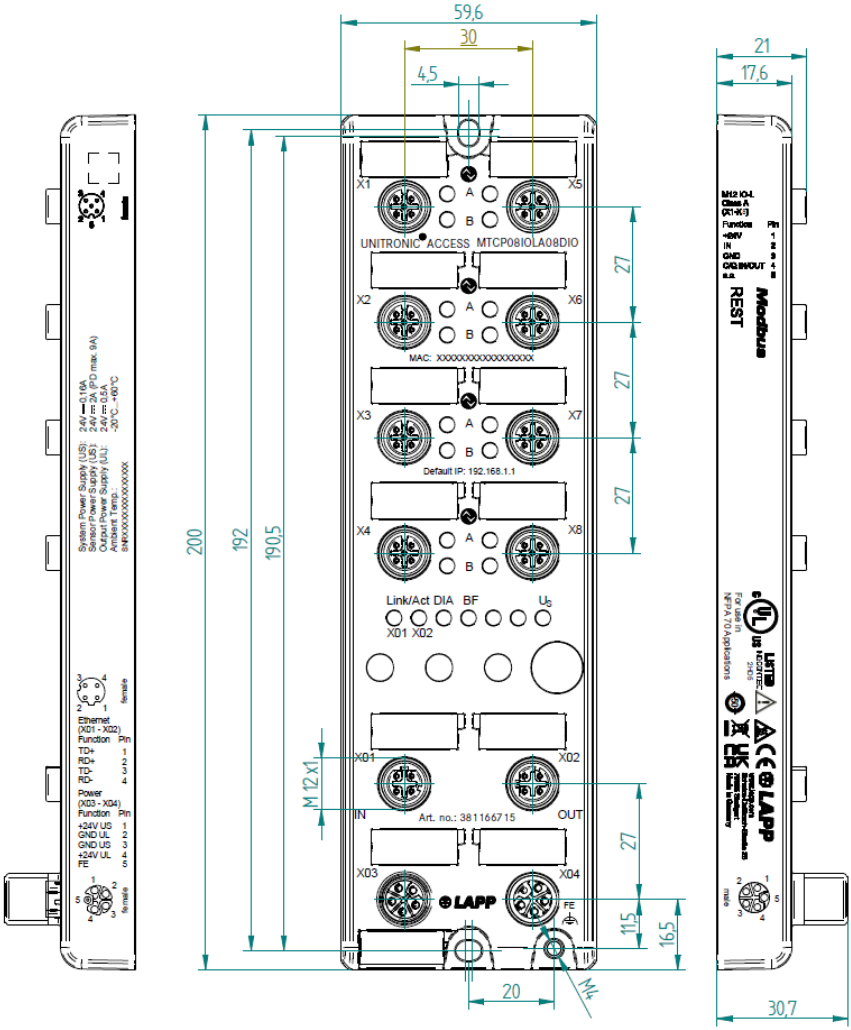


Figure 2: MTCP08IOLA08DIO

6.2.3 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 device variants.

Warning: Observe the following maximum output power for the sensor supply of Class A devices:

Max. 4.0 A per port; for **UL applications** max. 5 A for every port pair X1/X2, X3/X4, X5/X6, X7/X8; max. 9.0 A in total (with derating) for the whole port group X1 .. X8.

6.3 Port assignments

All the contact arrangements shown in this chapter show the frontal view of the connection area for the connectors.

6.3.1 Ethernet ports, M12 socket, 4-pin, D-coded

Color coding: green

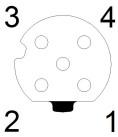


Figure 3: Schematic drawing, ports X01, X02

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 6: 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

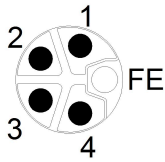


Figure 4: Schematic diagram of the M12 L-coding (connector X03 for Power In)

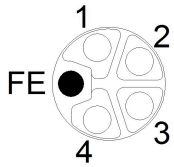


Figure 5: Schematic diagram of the M12 L-coding (socket X04 for Power Out)

6.3.2.1 IO-Link Master with Class A ports

Power supply	Pin	Signal	Function
	1	U_S (+24 V)	Sensor/system power supply
	2	GND_U_L	Ground/reference potential U_L
	3	GND_U_S	Ground/reference potential U_S[1]
	4	U_L (+24 V)	Load supply (**NOT** electrically isolated to U_S internally in device)
	5	FE	Functional ground

Table 7: Power supply with M12-Power Class A

Attention: Only use power supply units for the system/sensor and actuator supply that correspond to PELV (Protective Extra Low Voltage) or SELV (Safety Extra Low Voltage). Power supplies according to EN 61558-2-6 (transformers) or EN 60950-1 (switching power supply units) fulfill these requirements.

[1] U_L and U_S ground connected in device

6.3.2.2 IO-Link Master with Class A/B ports

Power supply	Pin	Signal	Function
Mixed IO-Link (Class A/B) I/O ports	1	U_S (+24 V)	Sensor/system power supply
	2	GND_U_{AUX}	Ground/reference potential U_{AUX} (electrically **isolated** to GND_U_S internally in device)
	3	GND_U_S	Ground/reference potential U_S
	4	U_{AUX} (+24 V)	Auxiliary supply (electrically **isolated** to U_S internally in device)
	5	FE	Functional ground

Table 8: Power supply with M12-Power Class A/B

Attention: Only use power supply units for the system/sensor and actuator supply that correspond to PELV (Protective Extra Low Voltage) or SELV (Safety Extra Low Voltage). Power supplies according to EN 61558-2-6 (transformers) or EN 60950-1 (switching power supply units) fulfill these requirements.

6.3.3 I/O ports as M12 sockets

Color coding: black

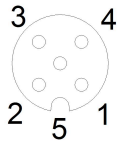


Figure 6: Schematic drawing I/O port as M12 socket IO-Link

6.3.3.1 IO-Link ports (Class A)

MP08...	Pin	Signal	Function
IO-Link Class A, ports X1 .. X8	1	+24 V	power supply +24 V
	2	IN/OUT	Ch. B: Digital input or digital output
	3	GND	Ground/reference potential
	4	C/Q	Ch. A: IO-Link data communication, digital input or digital output
	5	n.c.	not connected

PN08IOLA.../ EIP08IOLA.../ EC08IOLA.../ MTCP08IOLA.../ CCL08IOLA...	Pin	Signal	Function
IO-Link Class A, ports X1 .. X8	1	+24 V	power supply +24 V
	2	IN	Ch. B: Digital input
	3	GND	Ground/reference potential
	4	C/Q	Ch. A: IO-Link data communication, digital input or digital output
	5	n.c.	not connected

Table 9: I/O ports as IO-Link Class A

Used signal names compared to the IO-Link specification conventions:

Pin	UNITRONIC® ACCESS	IO-Link specification	Comment
1	+24 V	L+	Supplied by U_S
2	IN/OUT	I/Q	
3	GND	L-	
4	C/Q IN/OUT	C/Q	

7 Starting operation

7.1 Device identification

With any MODBUS client, the server running on UNITRONIC® ACCESS can be reached to get identification data including vendor name, product code and revision.

The following table represents the device identification data.

Register	Length	Description	Default value	Access
401025	1	Device firmware "Version major"	–	RO (Read Only)
401026	1	Device firmware "Version minor"	–	RO
401027	32	Name of IO-Link device	–	RO
401043	1	Source of IP address: 0: DHCP 1: Static	–	RO
401044	2	IP address of device	–	RO
401046	3	MAC address	–	RO
401047	1	Active TCP connections	–	RO

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.

7.3 Modbus function codes

UNITRONIC® ACCESS devices support the following Modbus function codes:

- Function code 03 (0x03)
- Function code 06 (0x06)
- Function code 16 (0x10)

Write access to holding registers is subject to the device feature set and the configuration of the holding register.

7.4 State on delivery

Modbus TCP parameters in state on delivery or after a factory reset:

Network mode:	DHCP
Static IP address:	192.168.1.XXX (XXX = rotary switch position or last stored data)
Subnet mask:	255.255.255.0
Gateway address	0.0.0.0
Device designations:	MTCP08IOLA08DIO
Product type:	Modbus TCP server

7.5 Setting the rotary encoding switches

The following UNITRONIC® ACCESS IO-Link Master variants additionally provide the protocol CC-Link IE Field Basic (CC):

- MP08IOLA08DIO

Caution: Risk of device damage due to corrupt device memory

Any interruption of the power supply to the device during and after protocol selection can lead to a corrupt device memory.

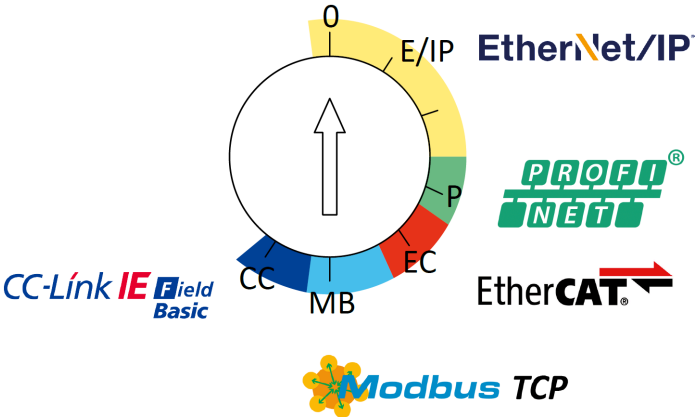
After selecting a protocol followed by a restart of the device, the new protocol is initialized. This can take up to 15 seconds. During this time the device is not usable and the LED indicators are out of function. When the protocol change is complete, the LED indicators return to normal operation and the device can be used again.

- Make sure that the power supply is maintained during the entire process.

The UNITRONIC® ACCESS multi-protocol variants allow you to select different protocols for communication within an industrial Ethernet system. In this way the IO-Link Masters with multi-protocol function can be integrated into different networks without it being necessary to purchase products specific for each protocol. This technology also gives you the option to use the same IOL-Master in different environments.

Using rotary encoding switches at the lower front of the devices, you can easily and conveniently set both the protocol and the address of the device, if the protocol to be used supports this. Once you have made a protocol selection and started the cyclical communication, the device stores this setting permanently and uses the selected protocol from this point on. To use another supported protocol with this device, perform a factory reset.

The multi-protocol devices have a total of three rotary encoding switches. With the first rotary encoding switch (x100) you set the protocol by using the corresponding switch position. Additionally, x100 is used to set the third last digit of the IP address for EIP.



With the other rotary encoding switches (x10 / x1), you set the last two digits of the IP address when you are using EtherNet/IP, Modbus TCP or CC-Link IE Field Basic.

Protocol	x100	x10	x1
EtherNet/IP	0-2	0-9	0-9
PROFINET	P	–	–
EtherCAT®	EC	–	–
Modbus TCP	MB	0-9	0-9
CC-Link IE Field	CC	0-9	0-9

Table 10: 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 40.

If you position the rotary encoding switch in a manner that is invalid, the device signals this to you with a solid red lighting BF LED.

7.5.1 Modbus TCP

If you decide to use Modbus TCP as a protocol, use the first rotary encoding switch to select the protocol. The second rotary encoding switch (x10) can be used to configure the 10 position of the last octet of the IP address, and the third rotary encoding switch (x1) allows you to configure the 1 position. Values between 0 and 9 can be selected for the second and third switches. The first three octets of the IP address are set by default to 192.168.1.

For example, the rotary encoding switch setting 5(x100), 1(x10) and 0(x1) gives you an IP address of 192.168.1.10 for Modbus TCP. It is only possible to assign IP addresses between 192.168.1.1 and 192.168.1.99 for Modbus TCP via the rotary switches.

Rotary switch setting	Function
500 (network parameters already saved)	The network parameters last saved are used (IP address, subnet mask, gateway address, DHCP on/off, BOOTP on/off).
501 ... 599	The last 2 digits of the saved or preset IP address are overwritten by the setting of the rotary switch.
979	The device performs a reset to the factory settings. The network parameters are also reset to the default values. Communication is not possible in this operation mode.

Table 11: Setting options of the rotary encoding switches for Modbus TCP

7.5.2 Factory reset

A factory reset restores the original factory settings and thus resets the changes and settings you have made up to that point. It also resets the protocol selection. To perform a factory reset, set the first rotary encoding switch (x100) to 9, the second (x10) to 7, and the third (x1) also to 9.

Afterwards perform a power cycle and wait 10 seconds due to internal memory write processes.

During the factory reset, the U_S LED lights up red. As soon as the factory settings have been restored, the U_S LED is flashing green.

	x100	x10	x1
Factory Reset	9	7	9

Follow the steps from section Setting the rotary encoding switches again to select a new protocol.

For performing a factory reset via software configuration, see chapter OPC UA configuration on page 96 and the configuration section.

7.6 Setting network parameters

Use the two right-hand rotary switches (x10 and x1) on the front of the device to set the last octet of the static IP address. Each rotary switch in the range of Modbus TCP is assigned to one decimal digit, so that you can configure a number between **0 – 99**. During startup, the position of the rotary switches is typically read within one time cycle.

The complete IP address, the subnet mask, the gateway address and the network mode (DHCP or BOOTP) can be configured and stored via the Web server or any other available configuration interfaces. New configuration interfaces can only be applied to after a restart of the device.

For additional information, see chapter Setting the rotary encoding switches.

8 Configuration Modbus TCP

The UNITRONIC® ACCESS devices support Modbus over a standard TCP network. It is possible to create 4 to 8 socket connections with devices.

8.1 Configuration parameters

Parameters of the UNITRONIC® ACCESS device can be configured via function code 06 and 16, depending on the features and the designated functionality of the holding register. To avoid malfunctioning of the device, there is a register which acts as latch switch.

Register	Length	Description	Default value	Access
400512	1	If **1** is written, then and only then the configuration registers described below can be written. If **0** is written, the updated configuration will be taken by the device Transition needs to be detected: **0** to **1**: Below described registers will be RW **1** to **0**: Configuration apply	0	RW ("Read/ Write")

8.2 General settings

Register	Length	Description	Default value	Access
400593	1	Report U_L/U_{AUX} supply voltage fault 0 = Report U_L/U_{AUX} supply voltage fault enabled 1 = Report U_L/U_{AUX} supply voltage fault disabled 2 = Auto	0	RW
400594	1	Report actuator fault without U_L/U_{AUX} voltage 0 = Report actuator fault without U_L/U_{AUX} voltage enabled 1 = Report actuator fault without U_L/U_{AUX} voltage disabled	0	RW
400595	1	Report U_S voltage fault 0 = Report U_S voltage fault enabled 1 = Report U_S voltage fault disabled	0	RW
400596	1	Reserved	0	
400597	1	Output auto restart 0 = Output auto restart disabled 1 = Output auto restart enabled	0	RW
400598	1	Web interface lock 0 = Web interface lock disabled 1 = Web interface lock enabled	0	RW
400599	1	Force mode lock 0 = Force lock disabled 1 = Force lock enabled	0	RW
400600	1	External configuration lock 0 = External configuration lock disabled 1 = External configuration lock enabled	1	RW

8.2.1 Force mode lock

The input and output process data can be forced via different interfaces (e.g. Web interface, REST, OPC UA, MQTT). The support of interfaces depends on the available software features. If the *Force mode lock* is enabled, it is no longer possible to force input and output process data through these interfaces.

Danger: Risk of physical injury or death! Unattended forcing can lead to unexpected signals and uncontrolled machine movements.

8.2.2 Web interface lock

The Web interface access can be configured. If *Web interface lock* is enabled, the Web pages are no longer reachable.

8.2.3 Report U_L/U_{AUX} supply voltage fault

During commissioning, it is possible that no power supply is connected to the U_L/U_{AUX} pins. Therefore it can be helpful to suppress and disable the *U_L/U_{AUX} supply voltage fault* diagnosis.

8.2.4 Report actuator fault without U_L/U_{AUX} voltage

During commissioning, it is possible that no power supply is connected to the U_L/U_{AUX} pins. Therefore it can be helpful to suppress and disable the *Report actuator fault without U_L/U_{AUX} voltage* diagnosis.

8.2.5 Report U_S voltage fault

During commissioning, it is possible that no power supply is connected to the U_S pins. Therefore it can be helpful to suppress and disable the *Report U_S voltage fault* diagnosis.

8.2.6 External configuration lock

Configuration parameters can be set via different alternative interfaces (e.g. Web interface, REST, OPC UA, MQTT). An external configuration can only be done, if no cyclic PLC connection is active. Every new PLC configuration overwrites the external configuration settings.

8.3 Channel settings

Register	Length	Description	Default value	Access
400513 .. 400528	1	DO Surveillance Timeout Port X1 Ch A .. Port X8 Ch B Valid values: 0 .. 255	0	RW ("Read/Write")
400529 .. 400544	1	DO Failsafe Port X1 Ch A .. Port X8 Ch B 0: Set Low 1: Set High 2: Hold Last	0	RW
400545 .. 400560	1	DI Filter Port X1 Ch A .. Port X8 Ch B 0: Disabled 1: 1 ms 2: 2 ms 3: 3 ms 4: 6 ms 5: 10 ms 6: 15 ms	0	RW
400561 .. 400576	1	DI Logic Port X1 Ch A .. Port X8 Ch B 0: Normally Open 1: Normally Close	0	RW
400577 .. 400592	1	DO Restart Port X1 Ch A .. Port X8 Ch B 0: Disable 1: Enable	1	RW
400601 .. 400616	1	DI Latch 0: DI Latch disabled 1: DI Latch enabled	0	RW

Register	Length	Description	Default value	Access
400617 .. 400632	1	DI Extension Valid values: 0 .. 255 (0 = 0 ms/Off, 16 = 16 ms)	0	RW

8.3.1 DO Surveillance Timeout

The digital output channels are monitored during runtime. The error states are detected and reported as a diagnosis. To avoid error states during the switching of output channels, the surveillance timeout can be configured as a delay with deactivated monitoring.

The delay time begins with a rising edge of the output control bit. After delay time has elapsed, the output is monitored and error states are reported by diagnosis. When the channel is permanently switched on or off, the typical filter value (not changeable) is 5 ms.

8.3.2 DO Failsafe

The UNITRONIC® ACCESS devices support a failsafe function for the channels used as digital outputs. In case of an internal device error, the PLC is in STOP state and cannot provide valid process data. The connection is interrupted or the communication is lost. The outputs are controlled according to the configured failsafe values.

Set Low:

If failsafe is active, the physical output pin of the channel is set to low ("0").

Set High:

If failsafe is active, the physical output pin of the channel is set to high ("1").

Hold Last:

If failsafe is active, the physical output pin of the channel holds the last valid process data state ("0" or "1").

8.3.3 DO Restart Mode

In case of a short circuit or overload at an output channel, a diagnosis is reported and the output is switched to "off".

If *DO Restart Mode* is disabled, the output channel is not automatically turned on again. It can be turned on after a logical reset of the process output data of the channel.

If *DO Restart Mode* for this channel is enabled, the output will automatically be turned on again after a fix time delay for checking if the overload or short

circuit condition is still active. When it is active, the channel is switched off again.

8.3.4 DI Logic

The logical state of an input channel can be configured via these parameters. If a channel is set to "Normally Open", a low signal ("0") is transferred to the process input data (e.g. if a non-damped sensor has an open switching output).

If a channel is set to "Normally Close", a high signal ("0") is transferred to the process input data (e.g. if a non-damped sensor has a closed switching output).

The channel LED shows, independent of these settings, the physical input state of the port pin.

8.3.5 DI Filter

A filter time for every digital input channel can be configured by these parameters. When there is no need for a filter it can be disabled.

8.3.6 DI Latch

With this parameter enabled, a rising edge at digital input is held high in the input status data (latched) as this is acknowledged by the PLC.

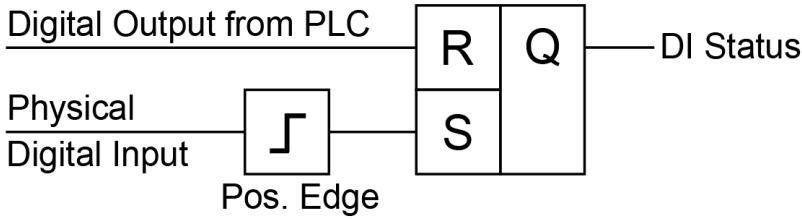


Figure 7: Input latch

R	S	Q
0	0	x (hold last)
0	1	1
1	0	0
1	1	0

Table 12: Truth table for input latch

When the input latch is enabled for a specific channel:

- The corresponding physical input of the channel is logically connected to the set input of a latch via an edge detector.
- The corresponding PLC output control bit (consuming data of I/O device) is logically connected to the reset input of the latch.
- The latch output is connected to the PLC input status data (producing data of the I/O device) of the corresponding channel.
- The latch will work according to the truth table above.
- It is not possible to directly read the physical input status data of that input channel, as the latch output is mapped to the PLC input status data.

The behavior in detail:

- A rising edge on the digital input will trigger the latch and set the latch output to '1'.
- The output will remain '1' until it is reset by the PLC program.
- A logical '1' on the corresponding PLC output control bit for that channel will reset the latch and set the latch output to '0', regardless of the set input or physical input state.
- If the input logic is inverted in the channel configuration, the inverted input logic is connected to the latch. Therefore, it will trigger on the falling edge regarding to a physical input signal.
- If the input is already high during enabling the latch, the latch will set to '1' (Q).

These DI Latch settings only work for channels that have been set to 'digital input mode'. It is recommended to always reset the latch before usage.

Default: Disabled

8.3.7 DI Extension

This parameter extends the duration of the digital input status after a state change at the physical input, when the input state change is faster than the extension time set.

The extension time will be applied on 'high' to 'low' and 'low' to 'high' input transitions. This setting only works for channels that have been set to 'digital input'.

Example:

The DI extension parameter is set to 16 ms, the physical input signal has low status => a high signal is detected for 8 ms.

In this case, the DI channel reports a high-status signal for 16 ms, regardless of other physical input signal transitions during this time.

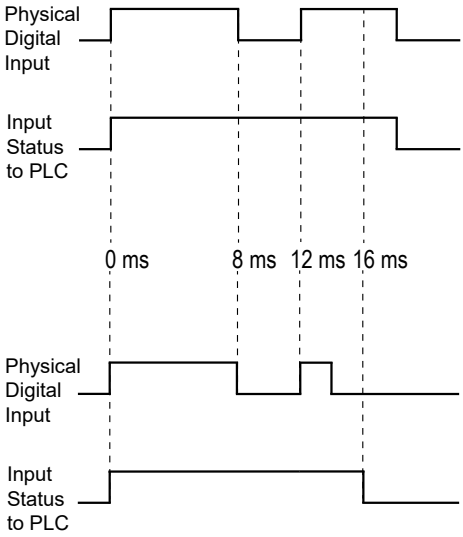


Figure 8: DI Extension

Available values: Off; 8 ms; 16 ms; 64 ms

Default: Off

8.4 IO-Link Port 1 .. 8 settings

Register	Length	Description	Default value	Access
401513	64	IO-Link Ch. 1 Settings	–	RW ("Read/Write")
401577	64	IO-Link Ch. 2 Settings	–	RW
401641	64	IO-Link Ch. 3 Settings	–	RW
401705	64	IO-Link Ch. 4 Settings	–	RW
401769	64	IO-Link Ch. 5 Settings	–	RW
401833	64	IO-Link Ch. 6 Settings	–	RW
401897	64	IO-Link Ch. 7 Settings	–	RW
401961	64	IO-Link Ch. 8 Settings	–	RW

Register	Length	Description	Default value	Access
402025	16	IO-Link Ch. 1 Serial Nr.	0	RO
402041	16	IO-Link Ch. 2 Serial Nr.	0	RO
402057	16	IO-Link Ch. 3 Serial Nr.	0	RO
402073	16	IO-Link Ch. 4 Serial Nr.	0	RO
402089	16	IO-Link Ch. 5 Serial Nr.	0	RO
402105	16	IO-Link Ch. 6 Serial Nr.	0	RO
402121	16	IO-Link Ch. 7 Serial Nr.	0	RO
402137	16	IO-Link Ch. 8 Serial Nr.	0	RO

Detailed channel settings

The following example for channel 1 represents all possible port settings for IO-Link. The values are identical for all 8 ports.

Register	Length	Description	Default value	Access
401513	1	Port Mode 0 = Deactivated 1 = IO-Link Manual 2 = IO-Link Auto 3 = Digital Input 4 = Digital Output	3	RW
401514	1	Validation and Backup 0: No device check (no data storage) 1: Type compatible V1.0 device (no data storage) 2: Type compatible V1.1 device (no data storage) 3: Type compatible V1.1 device with Backup + Restore (download + upload) 4 Type compatible V1.1 device with Restore (download master to device)	0	RW
401515	1	IQ Mode 0 = Deactivated 1 = Digital Input 2 = Digital Output 5 = AUX (Class A/B variants only)	1	RW
401516	1	Cycle Time 0: As fast as possible 1: 1.6 ms 2: 3.2 ms 3: 4.8 ms 4: 8.0 ms 5: 20.8 ms 6: 40.0 ms 7: 80.0 ms 8: 120.0 ms	0	RW
401517	1	Vendor ID 0 .. 65535 ("0")	0	RW
401518	2	Device ID 0 .. 16777215 ("0")	0	RW

Register	Length	Description	Default value	Access
401520	1	Swapping Mode (Producing): 0: Raw IO-Link 1 .. 16: 1 .. 16 WORD 17 .. 24: 1 .. 8 DWORD 0 .. 30 Bytes ("0")	0	RW
401521	1	Swapping Mode (Consuming): 0: Raw IO-Link 1 .. 16: 1 .. 16 WORD 17.. 24: 1 .. 8 DWORD 0 .. 30 Bytes ("0")	0	RW
401522	1	Swapping Offset (Producing) 0 .. 30 Bytes ("0")	0	RW
401523	1	Swapping Offset (Consuming) 0 .. 30 Bytes ("0")	1	RW
401524	1	Pin2 LED enabled 0: Disable LED on Channel B 1: Enable LED on Channel B	1	RW
401525	1	Suppress All Diagnosis 0: Generate diagnosis on this channel 1: Do not generate any diagnosis on this channel	0	RW
401526	1	Failsafe mode 0: Set Low 1: Set High 2: Hold Last 3: Replacement Value (transferred via offset 14-46)	2	RW
401527	32	IOL Failsafe replacement values	0	RW
401559	1	Use Push Pull for Pin 4 0: Use High-Side switches 1: Use Push Pull	0	RW
401560	1	Reserved	0	RW

Register	Length	Description	Default value	Access
401561	1	Pin 4 current limit 0: U_L, 0.5 A 1: U_L, 1.0 A 2: U_L, 1.5 A 3: U_L, 2.0 A 4: U_L, Max.	4	RW
401562	1	Pin 2 current limit 0: U_L, 0.5 A 1: U_L, 1.0 A 2: U_L, 1.5 A 3: U_L, 2.0 A 4: U_L, Max.	4	RW
401563 .. 401576	14	Reserved	0	RW

8.4.1 Port Mode

The *Port Mode* describes how the IO-Link Master handles the presence of an IO-Link device at the port.

Deactivated:

The IO-Link port is deactivated but can be configured for later use. No diagnostics are generated if the IO-Link device is not connected.

IO-Link Manual:

The IO-Link port is activated and explicit port configuration can be done for the parameters *Validation and Backup* (Inspection Level), *Vendor ID*, *Device ID* and *Cycle Time*.

IO-Link Auto:

The IO-Link port is activated and no explicit port configuration is needed. Configurations such as *Validation and Backup* (Inspection Level), *Vendor ID*, *Device ID* and *Cycle Time* are not required.

Digital Input:

In this mode, the channel operates as digital input. The channel state can be seen in the *Digital Input Channel* status of the cyclic process data.

Digital Output:

In this mode, the channel operates as digital output. The channel can be controlled by the *Digital Output Channel Control* (first two bytes of the output data) or by the *IO-Link Output Data* (first byte of each IO-Link device output data) of the cyclic process data. This depends on the *Digital Output Control* parameter in the general settings.

8.4.2 Validation and Backup

With this parameter, the user can set the behavior of the IO-Link ports regarding the type compatibility and data storage mechanism of the connected IO-Link Device.

The precondition for using *Validation and Backup* is that you configure the *Port Mode* to "IO-Link Manual".

The IO-Link Master has a backup memory which can be used for storing the device parameters and for restoring them on the device. This backup memory can be deleted by the following events:

- IO-Link Master factory reset
- *Channel Mode* reconfiguration, e.g. from “Digital-Input” to “IO-Link”
- *Validation and Backup* reconfiguration, e.g. from “No device check” to “Type compatible V1.1 device with Backup & Restore”

For further information refer to the ‘IO-Link Interface and System Specification’ version 1.1.3 which can be downloaded from https://io-link.com/

No device check (no data storage):

No check of connected Vendor ID or Device ID and no "Backup and Restore" support of the IO-Link Master parameter server.

Type compatible V1.0 device (no data storage):

Type compatible according IO-Link specification V1.0 which includes validation of Vendor ID and Device ID. The IO-Link specification V1.0 does not support IO-Link Master parameter server.

Type compatible V1.1 device (no data storage):

Type compatible according IO-Link specification V1.1 which includes validation of Vendor ID and Device ID. "Backup and Restore" is disabled.

Type compatible V1.1 device with Backup + Restore

(upload + download):

Type compatible according IO-Link specification V1.1 which includes validation of Vendor ID and Device ID. "Backup and Restore" is enabled.

Pay attention to the following explanations regarding *Backup and Restore* conditions:

- Backup (Device to Master):

  A Backup (upload from IOL-Device to IOL-Master) is performed when an IO-Link Device is connected and the Master does not have any valid parameter data. The read parameter data are permanently stored on the IO-Link Master.

  An upload will also be performed, when the IO-Link Device has set the DS_UPLOAD_FLAG (Data Storage Upload Flag). This IOL-Device flag can be set in two ways:

  - Parameters written to IOL-Device in *Block Parameter* mode: An IO-Link Device sets the DS_UPLOAD_FLAG self-dependent, if the parameters were written in block parameter mode to the IO-Link Device with the last system command ParamDownloadStore (e.g. by a third party USB IO-Link Master for commissioning).
  - Parameters written to IOL-Device in *Single Parameter* mode: If single parameter data is changed on the IOL-Device during runtime, the stored device parameters on the IOL-Master can be updated using the ParamDownloadStore (index 0x0002, subindex 0x00, value 0x05) command. This command sets the DS_UPLOAD_REQ flag on the IOL-Device and thus the IO-Link Master executes an upload procedure from the IO-Link Device.

- Restore (Master to Device):

  A Restore (download from IOL-Master to IOL-Device) is performed when an IO-Link Device is connected and the IO-Link Master has valid parameter data stored which are usable for the IOL-Device and not equal compared to the device parameters.

  The restore procedure can be blocked by the IO-Link Device via the *Device Access Locks* parameter when supported by the IO-Link Device (Index 0x000C, refer to vendor specific IO-Link Device documentation).

Type compatible V1.1 device with Restore (download Master to Device):

Type compatible according IO-Link specification V1.1 which includes validation of Vendor ID and Device ID. Only "Restore" is enabled.

Pay attention to the following explanations regarding *Restore* conditions:

- Restore (Download / IOL-Master to IOL-Device):

  A Restore (download from IOL-Master to IOL-Device) is performed when an IO-Link Device is connected and the IO-Link Master has valid parameter data stored which are usable for the IOL-Device and not equal compared to the device parameters.

  In the *Restore* mode no change of the IOL-Device parameters will be stored permanently on the IOL-Master. When the IOL-Device sets the DS_UPLOAD_FLAG in this mode, the device parameters will be restored by the IOL-Master.

  The restore procedure can be blocked by the IO-Link Device via the *Device Access Locks* parameter when supported by the IO-Link Device (Index 0x000C, refer to vendor specific IO-Link Device documentation).

8.4.3 IQ Mode

The operating mode of Pin 2 (Channel B) of the respective IO-Link channel can be configured via this parameter.

Digital Output:

In this mode, the channel operates as digital output. The channel can be controlled by the *Digital Output Channel Control* (first two bytes of the output data) or by the *IO-Link Output Data* (first byte of each IO-Link device output data) of the cyclic process data. This depends on the *Digital Output Control* parameter in the general settings.

Digital Input:

In this mode, the channel operates as digital input. The channel state can be seen in the *Digital Input Channel* status of the cyclic process data.

AUX:

This option is exclusively applicable for Class A/B Mixmodules.

In this mode, Pin 2 of the IO-Link port acts as an auxiliary voltage output. The auxiliary voltage is fed by the U_{AUX} supply input. The auxiliary voltage output cannot be controlled.

8.4.4 Cycle Time

The IO-Link cycle time can be configured by this parameter.

The precondition for using *Cycle Time* is that you configure *Port Mode* to "IO-Link Manual".

As fast as possible:

The IO-Link port uses the max. supported IO-Link Device and Master update cycle time for the cyclic I/O data update between IO-Link Master and IO-Link Device.

1.6 ms, 3.2 ms, 4.8 ms, 8.0 ms, 20.8 ms, 40.0 ms, 80.0 ms, 120.0 ms:

The cycle time can be set manually to the provided options. This option can be used e.g. for IO-Link devices which are connected over inductive couplers. Inductive couplers are normally the bottleneck in the update cycle

time between IO-Link Master and IO-Link Device. In this case, please refer to the data sheet of the inductive coupler.

8.4.5 Vendor ID

The *Vendor ID* is needed for the validation of the IO-Link device and can be configured with this parameter.

Precondition for using the *Vendor ID* is that you configure *Port Mode* to "IO-Link Manual". *Validation and Backup* must be set to a type compatible V1.X device.

8.4.6 Device ID

The *Device ID* is needed for the validation of the IO-Link device and can be configured with this parameter.

Precondition for using the *Device ID* is that you configure *Port Mode* to "IO-Link Manual". *Validation and Backup* must be set to a type compatible V1.X device.

8.4.7 Swapping Mode

The byte order of IO-Link is big endian. For setting output data in the correct format, the parameters *Swapping Mode* and *Swapping Offset* support the user. There can be selected up to 16 words or up to 8 double words for converting the output data.

Raw IO-Link Data:

No byte swap

Data type WORD:

Data byte order: Byte 1, Byte 2

Order after Swap: Byte 2, Byte 1

Data type DWORD:

Data byte order: Byte 1, Byte 2, Byte 3, Byte 4

Order after Swap: Byte 4, Byte 3, Byte 2, Byte 1

8.4.8 Swapping Offset

The *Swapping Offset* describes the start point in the process data for using the configured *Swapping Mode*.

8.4.9 IOL Failsafe

The UNITRONIC® ACCESS devices support a failsafe function for the output data of the IO-Link channels. In case of an internal device error, the PLC is in STOP state and cannot provide valid process data, the connection is interrupted or the communication is lost: The output data of the IO-Link channels is controlled by the configured failsafe values.

Set Low:

If failsafe is active, all bits of the IO-Link output data are set to low ("0").

Set High:

If failsafe is active, all bits of the IO-Link output data are set to high ("1").

Hold Last:

If failsafe is active, all bits of the IO-Link output data arel hold the last valid process data state ("0" or "1").

Replacement Value:

A replacement value can be set via the *IO-Link Failsafe* parameter object for every IO-Link device. If failsafe is active, these replacement values are transmitted to the IO-Link device. Take into account that in the case of an error the replacement values are sent instead of the output process data so that a configured *Swapping Mode* has influence on the byte order.

IO-Link Master Command:

If failsafe is active, an IO-Link-specific mechanism for valid/invalid output process data is used and the IO-Link device determines the behavior itself.

8.4.10 IOL Failsafe replacement values

32 holding registers represent byte-wise 32 replacement values. If failsafe is active, these values are transmitted to IO-Link device.

8.4.11 Serial number

The desired IO-Link device serial number, starting from 402025, can be written in the holding registers. Each serial number can contain a maximum of 32 bytes. 16 registers are assigned to each IO-Link channel.

9 Process data assignment

The UNITRONIC® ACCESS devices in general support process data communication in both directions. The consuming data in this context is defined as the process output data which controls physical outputs and IO-Link output data. The producing data in this context is defined as the process input data which contains the physical inputs, diagnostics and IO-Link input data with optional extended status and event data.

The following chapters describe the data images for the consuming and producing data direction which are assigned to the output and input assemblies.

9.1 Consuming data (output)

Register	Length	Description	Default value	Access
400001	16	Process Data Output Port 1	–	RW ("Read/Write")
400017	16	Process Data Output Port 2	–	RW
400033	16	Process Data Output Port 3	–	RW
400049	16	Process Data Output Port 4	–	RW
400065	16	Process Data Output Port 5	–	RW
400081	16	Process Data Output Port 6	–	RW
400097	16	Process Data Output Port 7	–	RW
400113	16	Process Data Output Port 8	–	RW

Each holding register holds two bytes of output data. Valid process data values depend on the port configuration.

Examples:

Port	Mode	Valid output data to be written
1	IO-Link 1 byte output	Least significant 1 byte of 400001
2	Digital output	Least significant bit (bit 0) of 400017
3	IO-Link 4 bytes output	All 2 bytes on registers 400033 to 400034
4	IO-Link 16 bytes output	All 2 bytes on registers 400049 to 400057
5	IO-Link 32 bytes output	All 2 bytes on registers 400065 to 400080

Table 13: Examples for consuming data

9.2 Producing data (input)

Register	Length	Description	Default value	Access
400257	16	Process Data Input Port 1	–	RO ("Read Only")
400273	16	Process Data Input Port 2	–	RO
400289	16	Process Data Input Port 3	–	RO
400305	16	Process Data Input Port 4	–	RO
400321	16	Process Data Input Port 5	–	RO
400337	16	Process Data Input Port 6	–	RO
400353	16	Process Data Input Port 7	–	RO
400369	16	Process Data Input Port 8	–	RO

Each holding register holds two bytes of input data. Valid process data values depend on the port configuration.

Examples:

Port	Mode	Valid input data to be read
1	IO-Link 1 byte input	Least significant 1 byte of 400257
2	Digital input	Least significant bit (bit 0) of 400273
3	IO-Link 4 bytes input	All 2 bytes on registers 400289 to 400290
4	IO-Link 16 bytes input	All 2 bytes on registers 400305 to 400312
5	IO-Link 32 bytes input	All 2 bytes on registers 400321 to 400336

Table 14: Examples for producing data

9.3 Channel B as digital output

Register	Length	Description	Default value	Access
400436	1	PIN 2 (Channel B) as DO (Bit wise)	–	RW ("Read/Write")

This holding register represents bitwise the values of Channel B (Pin 2) of IO-Link class-A port. Values written on this register will only have an effect if Channel B is configured as Digital Output.

400436	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Ch. B Digital Output	NA	NA	NA	NA	NA	NA	NA	NA	X8	X7	X6	X5	X4	X3	X2	X1

9.4 Channel B as digital input

Register	Length	Description	Default value	Access
400435	1	PIN 2 (Channel B) as DI (Bit wise)	–	RW ("Read/Write")

This holding register represents bitwise the values of Channel B (Pin 2) of IO-Link class-A port. Values written on this register will only have an effect if Channel B is configured as Digital Input.

400435	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Ch. B Digital Input	NA	NA	NA	NA	NA	NA	NA	NA	X8	X7	X6	X5	X4	X3	X2	X1

10 Diagnostics processing

Register	Length	Description	Default value	Access
400401	1	System/sensor power supply present 0: System/sensor power not present 1: System/sensor power supply present	–	RO ("Read Only")
400402	1	Error of the system/sensor power supply 0: System/sensor power supply fault not present 1: System/sensor power supply fault occurred	–	RO
400403	1	Auxiliary/Actuator power supply present 0: Auxiliary/Actuator power supply not present 1: Auxiliary/Actuator power supply present	–	RO
400404	1	Error of the auxiliary/actuator power supply 0: Auxiliary/actuator power supply fault not present 1: Auxiliary/actuator power supply fault occurred	–	RO
400405	1	Internal Module Error	–	RO
400406	1	Force mode diagnosis	–	RO
400407	1	Overload/short-circuit of the I/O port sensor supply outputs This register represent bitwise status of Overload/short-circuit.	–	RO
400408	1	Overload/short circuit of the digital outputs Channel A (CQ) This register represent bitwise status of overload/short-circuit on digital outputs Channel A (CQ).	–	RO

Register	Length	Description	Default value	Access
400409	1	IQ short fault Overload/short circuit of the digital outputs Channel B (IQ). This register represent bitwise status of overload/short-circuit on digital outputs Channel B (IQ).	–	RO
400410	1	Overload/short circuit between Actuator voltage supply (Class B) This register represent bitwise status of Actuator voltage supply (Class B).	–	RO

Register	Length	Description	Default value	Access
400426	1	Status of IO-Link Ch. 1	–	RO
400427	1	Status of IO-Link Ch. 2	–	RO
400428	1	Status of IO-Link Ch. 3	–	RO
400429	1	Status of IO-Link Ch. 4	–	RO
400430	1	Status of IO-Link Ch. 5	–	RO
400431	1	Status of IO-Link Ch. 6	–	RO
400432	1	Status of IO-Link Ch. 7	–	RO
400433	1	Status of IO-Link Ch. 8	–	RO

Register	Length	Description	Default value	Access
402501	200	IO-Link events Ch. 1	–	RO
402701	200	IO-Link events Ch. 2	–	RO
402901	200	IO-Link events Ch. 3	–	RO
403101	200	IO-Link events Ch. 4	–	RO
403301	200	IO-Link events Ch. 5	–	RO
403501	200	IO-Link events Ch. 6	–	RO
403701	200	IO-Link events Ch. 7	–	RO
403901	200	IO-Link events Ch. 8	–	RO

10.1 Error of the system/sensor power supply

The voltage value for the incoming system/sensor power supply is also monitored globally. If the voltage drops below approx. 18 V, or exceeds approx. 30 V, an error diagnosis is generated. The IO-Link specification requires at least 20 V at the L+ (pin1) output supply of the I/O ports. At least 21 V of U_S supply voltage for the IO-Link Master are required to minimize the risk of internal voltage drops in the IO-Link Master.

The green U_S indicator is off.

The error diagnosis has no effect on the outputs.

Caution: It must definitely be ensured that the supply voltage, measured at the most remote participant is not below 21 V DC from the perspective of the system power supply.

The diagnostics in the holding register 400401 will be updated.

10.2 Error of the auxiliary/actuator power supply

The voltage value for the incoming auxiliary/actuator power supply is also monitored globally. If *Report U_L/U_{AUX} Supply Voltage Fault* is enabled, an error message is generated when the voltage drops below approx. 18 V or exceeds approx. 30 V. The U_L/U_{AUX} indicator shows red.

Holding register 400402 can be read to get the present status of the auxiliary/actuator power supply. The diagnostic data on holding register 400403 will be updated.

If output channels are set to *High State* and *Report DO Fault without U_L/U_{AUX}*, additional error diagnostics, caused by the voltage failure, are generated on the channels.

If *Report U_L/U_{AUX} Supply Voltage Fault* is disabled, no U_L/U_{AUX} or channel diagnostics appear.

10.3 Overload/short-circuit of the I/O port sensor supply outputs

In case of an overload or a short circuit between pin 1 and pin 3 on the ports (X1 .. X8), holding register 400407 will be updated with the following data:

400407	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
sensor supply Short circuit	NA	NA	NA	NA	NA	NA	NA	NA	X8	X7	X6	X5	X4	X3	X2	X1

10.4 Overload/short circuit of the digital outputs

In case of an overload or a short circuit between pin 4 and pin 3 on the ports (X1 .. X8), holding register 400408 will be updated with the following data. This diagnosis is valid if the respective port's channel A is in *Digital Output Mode*.

400408	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Overload/ short circuit of Channel A (CQ)	NA	NA	NA	NA	NA	NA	NA	NA	X8	X7	X6	X5	X4	X3	X2	X1

In case of an overload or a short circuit between pin 2 (Channel B) and pin 3 on the ports (X1 .. X8), holding register 400409 will be updated with the following data. This diagnosis is valid if the respective port's channel B is in *Digital Output Mode*.

400409	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Overload/ short circuit of Channel B (IQ)	NA	NA	NA	NA	NA	NA	NA	NA	X8	X7	X6	X5	X4	X3	X2	X1

10.5 Overload/short circuit of Actuator voltage supply P24 (Class B)

In case of an overload or a short circuit of the Actuator voltage supply P24 (Class B) on the ports X5 .. X8, the holding register 400410 will be updated with the following data.

400410	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Overload/ short circuit of Actuator voltage supply P24 (Class B)	NA	NA	NA	NA	NA	NA	NA	NA	X8	X7	X6	X5	NA	NA	NA	NA

10.6 IO-Link channel status

IO-Link contains standard data for the channel status. The states of an individual IO-Link channel can be read on holding registers 400426 .. 400433.

States of IO-Link channel

0:	No device present
1:	Deactivated
2:	Diagnosis mode
3:	Pre-OP mode

4:	OP mode
5:	DI mode
6:	DO mode
254:	Power OFF
255:	No status available

10.7 IO-Link channel events

During operation, the IO-Link Master generates events for a particular channel. Holding registers 402501 .. 404100 will hold up to 32 events.

Register	Length	Description	Default value	Access
402501	1	IO-Link Ch. 1 No. of events present	–	RO ("Read Only")
402502	5	IO-Link Ch. 1 Event 1	–	RO
402507	5	IO-Link Ch. 1 Event 2	–	RO
…		…		…
402662	5	IO-Link Ch. 1 Event 32	–	RO
402667	33	Reserved	–	RO
402701	1	IO-Link Ch. 2 No. of events present	–	RO
402702	5	IO-Link Ch. 2 Event 1	–	RO
402707	5	IO-Link Ch. 2 Event 2	–	RO
…		…		…
402762	5	IO-Link Ch. 2 Event 32	–	RO
402767	33	Reserved	–	RO
…		…		…
…		…		…
403901	1	IO-Link Ch. 8 No. of events present	–	RO
403902	5	IO-Link Ch. 8 Event 1	–	RO
403907	5	IO-Link Ch. 8 Event 2	–	RO
…		…		…
403962	5	IO-Link Ch. 8 Event 32	–	RO
403967	33	Reserved	–	RO

The following example shows details of a single event which is applicable for all 32 events on 8 channels:

Register	Length	Description	Default value	Access
402502	2	Timestamp	–	RO
402504	1	Event Type (Classification)	–	RO
402505	1	Event Code	–	RO
402506	1	Reserved	–	RO

The 32 event registers act like a ring buffer. Once FIFO buffer is full, the earliest event will be removed from buffer and all other events will be pushed one step up. Latest event details will be available then on event 32.

Timestamp:

Local timestamp of a device. Takes 4 bytes.

Event Type:

Classification of events. There are 3 different types of events. The type is classified via the values 1 .. 3 on the respective holding register.

1: Notification

2: Warning

3: Error

Event Code:

The event code represents a unique code for individual events as per IO-Link specifications. The following event code(s) will be generated on the respective holding register.

0x1000:	General Malfunction
0x1800:	No Device
0x1801:	Startup parametrization error
0x1802:	Incorrect Vendor ID - Inspection Level mismatch
0x1803:	Incorrect Device ID - Inspection Level mismatch
0x1804:	Short circuit at C/Q
0x1805:	PHY over temperature
0x1806:	Short circuit at L+
0x1807:	Overcurrent at L+
0x1808:	Device Event overflow
0x1809:	Backup inconsistency - memory out of range
0x180A:	Backup inconsistency - identity fault
0x180B:	Backup inconsistency - Data Storage unspecific error
0x180C:	Backup inconsistency - upload fault
0x180D:	Parameter inconsistency - download fault
0x180E:	P24 (Class B) missing or under voltage
0x180F:	Short circuit at P24 (Class B)
0x1810:	Short circuit at I/Q
0x1811:	Short circuit at C/Q (Digital)

0x1812:	Overcurrent at I/Q
0x1813:	Overcurrent at C/Q (Digital)
0x6000:	Invalid cycle time
0x6001:	Revision fault
0x6002:	ISDU batch failed
0xFF21:	DL: Device plugged in
0xFF22:	Device communication lost
0xFF23:	Data Storage identification mismatch
0xFF24:	Data Storage buffer overflow
0xFF25:	Data Storage parameter access denied
0xFF26:	Port status changed
0xFF27:	Data Storage upload completed
0xFF31:	DL: Incorrect Event signaling

11 IIoT functionality

The UNITRONIC® ACCESS 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 UNITRONIC® ACCESS 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 UNITRONIC® ACCESS variants provide user administration, which is also applicable for accessing and configuring the IIoT protocols. This allows you to manage all modification options for the device settings via personalized user authorizations.

All IIoT protocols can be used and configured independently of the field bus. It is also possible to use the devices completely without the help of a PLC and control them via IIoT protocols.

Attention: When using the IIoT functionality, a protected local network environment without direct access to the Internet is recommended.

Attention: Only activate one of the IIoT protocols at a time. Exclusively use MQTT **or** OPC UA.

11.1 MQTT

MQTT functions are **only** applicable for the following UNITRONIC® ACCESS variants:

- MP08IOLA08DIO

The MQTT (Message Queuing Telemetry Transport) protocol is an open network protocol for machine-to-machine communication, which provides the transmission of telemetric data messages between devices. The integrated MQTT client allows the device to publish a specific set of information to an MQTT broker.

The publishing of messages can either occur periodically or be triggered manually.

Attention: When using MQTT, the OPC UA protocol must be disabled.

11.1.1 MQTT configuration

In **delivery state**, MQTT functions are **disabled**. The MQTT client can be configured either using the Web interface or directly via a JSON object sent in an HTTP/HTTPS request. For more information see chapter MQTT configuration - Quick start guide on page 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: "**unitronic**")
will-enable	boolean	If true, the device provides a last will message to the broker	true / **false**
will-topic	string	The topic for the last will message.	(Default: **null**)
auto-publish	boolean	If true, all enabled domains will be published automatically in the specified interval.	**true** / false
publish-interval	number	The publish interval in ms if auto-publish is enabled. Minimum is 250 ms.	**2000**
publish-identity	boolean	If true, all identity domain data will be published	**true** / false
publish-config	boolean	If true, all config domain data will be published	**true** / false
publish-status	boolean	If true, all status domain data will be published	**true** / false
publish-process	boolean	If true, all process domain data will be published	**true** / false
publish-devices	boolean	If true, all IO-Link Device domain data will be published	true / **false**
commands-allowed	boolean	Master switch for MQTT commands. If false, the device will not subscribe to any command topic, even if specific command topics are activated below.	true / **false**
force-allowed	boolean	If true, the device accepts force commands via MQTT.	true / **false**
reset-allowed	boolean	If true, the device accepts restart and factory reset commands via MQTT.	true / **false**
config-allowed	boolean	If true, the device accepts configuration changes via MQTT.	true / **false**

Element	Data type	Description	Example data
qos	number	Selects the "Quality of Service" status for all published messages.	**0 = At most once** 1 = At least once 2 = Exactly once

Table 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 81.

11.1.2 MQTT topics

MQTT mainly relates to topics. All messages are attached to a topic which adds context to the message itself. Topics may consist of any string and they are allowed to contain slashes (`/`) as well as wildcard symbols (`*, #`).

11.1.2.1 Base topic

For all UNITRONIC® ACCESS 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 81.

Variables in the Base topic have to be written in brackets ("[]"). The following variables are possible:

Variable	Description
mac	The MAC address of the device
name	The name of the device
order	The ordering number of the device
serial	The serial number of the device

Table 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, IO-Link Device status and data.
process	All process data which is produced and consumed by the device itself or by attached devices.	Digital inputs, digital outputs, cyclic IO-Link data.
iold	IO-Link Device parameters according to the IO-Link specification.	Vendor name, product name, serial number, hardware revision, software revision and more.

Table 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, IO-Link state, IO-Link Device events	8	Interval
[base-topic]/process/gateway	All Digital IN/OUT	1	Interval
[base-topic]/process/port/n	Digital IN/OUT per port, IOL-data, pdValid	8	Interval
[base-topic]/iold/port/n	IO-Link Device parameter	8	Interval

Table 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

11.1.2.2 Publish topic

Overview of all publish JSON data for the defined topics:

Key	Data type
product_name	json_string
ordering_number	json_string
device_type	json_string
serial_number	json_string
mac_address	json_string
production_date	json_string
fw_name	json_string
fw_date	json_string
fw_version	json_string
hw_version	json_string
vendor_name	json_string
vendor_address	json_string
vendor_phone	json_string
vendor_email	json_string
vendor_techn_support	json_string
vendor_url	json_string
vendor_id	json_integer
device_id	json_integer

Table 20: Identity/gateway

Key	Data type	Range	Default value	Remarks
fieldbus_protocol	json_string	PROFINET, EtherNet/IP, EtherCAT®		
ip_address	json_string		192.168.1.1	
subnet_mask	json_string		255.255.255.0	
report_alarms	json_boolean		0.0.0.0	
report_ul_alarm	json_boolean	true / false	true	
report_do_fault_without_ul	json_boolean	true / false	false	
force_mode_lock	json_boolean	true / false	false	
web_interface_lock	json_boolean	true / false	false	
do_auto_restart	json_boolean	true / false	true	
fast_startup	json_boolean	true / false	false	PROFINET and EIP only

Table 21: Config/gateway

Key	Data type	Range	Default value	Remarks
protocol	json_string	wait_for_io_system wait_for_io_Connection failsafe connected error		
ethernet_port1	json_string	100_mbit/s_full 100_mbit/s 10_mbit/s_full 100_mbit/s		
ethernet_port2	json_string	100_mbit/s_full 100_mbit/s 10_mbit/s_full 100_mbit/s		
module_restarts	json_integer	0 .. 4294967295		
channel_diagnosis	json_boolean	true / false		
failsafe_active	json_boolean	true / false		
system_voltage_fault	json_boolean	true / false		
actuator_voltage_fault	json_boolean	true / false		
internal_module_error	json_boolean	true / false		
simulation_active_diag	json_boolean	true / false		
us_voltage	json_integer	0 .. 32		in Volts
ul_voltage	json_integer	0 .. 32		in Volts
forcemode_enabled	json_boolean	true / false		

Table 22: Status/gateway

Key	Data type	Range	Default value	Remarks
Input_data	json_integer[]			
output_data	json_integer[]			

Table 23: Process/gateway

Key	Data type	Range	Default value	Remarks
port	json_integer	1 .. 8		
type	json_string	digital_universal digital_input digital_Output io_link		
max_output_power_cha	json_string	2.0_mA 0.5_mA		
max_output_power_chb	json_string	2.0_mA 0.5_mA		
channel_cha	json_string	input/output input output io_link aux		
channel_chb	json_string	input/output input output io_link aux		

Table 24: Identity/port/1 .. 8

Key	Data type	Range	Default value	Remarks
port	json_integer	1 .. 8		
direction_cha	json_string	input/output input output		
restart_mode_cha	json_string	Manual Auto		
restart_mode_chb	json_string	Manual Auto		
input_polarity_cha	json_string	NO NC		
input_polarity_chb	json_string	NO NC		
input_filter_cha	json_integer			ms
input_filter_chb	json_integer			ms
do_auto_restart_cha	json_boolean	true / false		
do_auto_restart_chb	json_boolean	true / false		

Table 25: Config/port/1 .. 8

Key	Data type	Range	Default value	Remarks
port	json_integer	1 .. 8		
physical_state_cha	json_integer	0 .. 1		
physical_state_chb	json_integer	0 .. 1		
actuator_short_circuit_cha	json_boolean	true / false		
actuator_short_circuit_chb	json_boolean	true / false		
sensor_short_circuit	json_boolean	true / false		
current_cha	json_integer			mA
current_chb	json_integer			mA
current_pin1	json_integer			mA

Table 26: Status/port/1 .. 8

11.1.2.3 Command topic (MQTT Subscribe)

The main purpose of MQTT is to publish data from the device to a broker. This data can then be received by any subscriber who is interested in this data. But also the other way round is possible. The device can subscribe to a topic on the broker and is then able to receive data. This data can contain configuration or forcing data. This allows the user to fully control a device via MQTT only, without using other ways of communication like Web or REST.

If the configuration allows commands in general, the device subscribes to special Command topics on which it can receive commands from other MQTT clients. The Command topic is based upon the Base topic. It always has the following form:

`[base-topic]/command`

After the Command topic, there are fixed topics for different writeable objects. The data format of the MQTT payload is always JSON. It is possible to set only a subset of the possible objects and fields.

[…]/forcing

Use the Command topic `[base-topic]/command/forcing` for *Force object* data. The *Force object* can contain any of the following properties:

Property	Data type	Example values	Remarks
forcemode	boolean	true / false	Forcing Authority: on/off
digital	array (Table 28: Force object: Digital on page 90)		
iol	array (Table 29: Force object: IOL (IO-Link devices only) 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

Property	Data type	Example values	Remarks
port	integer	0, 1, 5	
output	array[integer]	[55, 88, 120]	
input	array[integer]		Input-Simulation

Table 29: Force object: IOL (IO-Link devices only)

[…]/config

Use the Command topic `[base-topic]/command/config` for *Config object* data. The *Config object* can contain any of the following properties:

Property	Data type	Example values	Remarks
portmode	array (Table 31: 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 30: Config object properties

For the *Config object* property `portmode`, there are several value specifications arrayed:

Property	Data type	Example values	Remarks
port	integer	2	
channelA*	string	"dio", "di", "do", "iol", "off"	
channelB*	string	"dio", "di", "do", "iol", "off", "aux"	
inlogicA	string	"no", "nc"	
inlogicB	string	"no", "nc"	
filterA	integer	3	input filter in ms
filterB	integer	3	input filter in ms
autorestartA	boolean		
autorestartB	boolean		
iolValidation	integer	0 = NoCheck 1 = Type 1.0 2 = Type 1.1 3 = Type 1.1 BR 4 = Type 1.1 RES	
iolDeviceID	integer		for validation
iolVendorID	integer		for validation

Table 31: 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 32: Reset object properties

[…]/publish

Use the Command topic `[base-topic]/command/publish` for *Publish object* data.

Trigger publish of all topics manually (can be used when auto publish is off or long interval is set).

11.1.3 MQTT configuration - Quick start guide

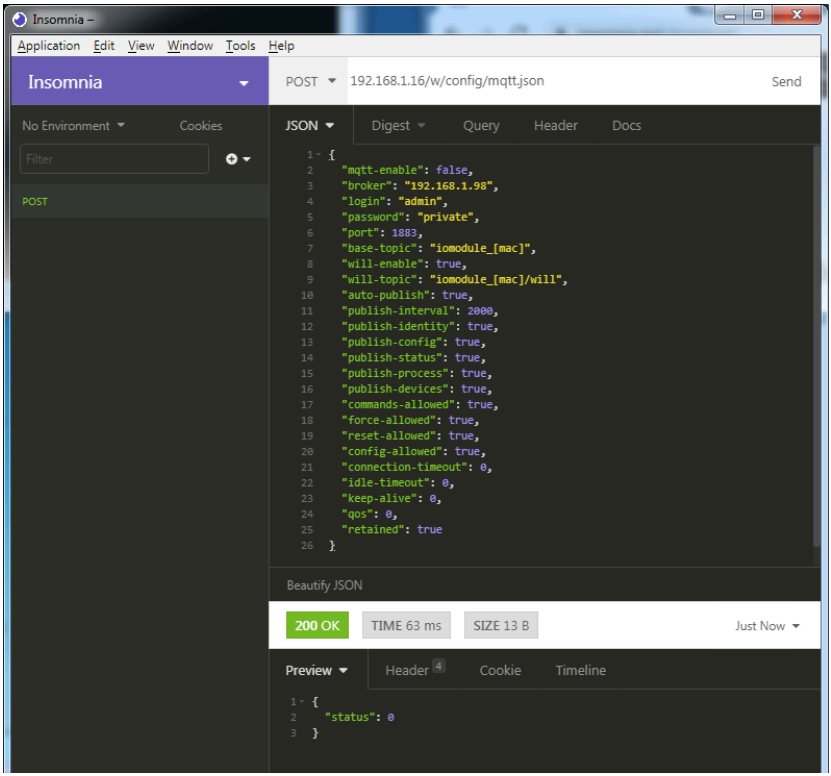
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

11.1.3.1 MQTT configuration via JSON

1. Depending on your application case, download and install *Insomnia* or a comparable application: https://insomnia.rest/download/

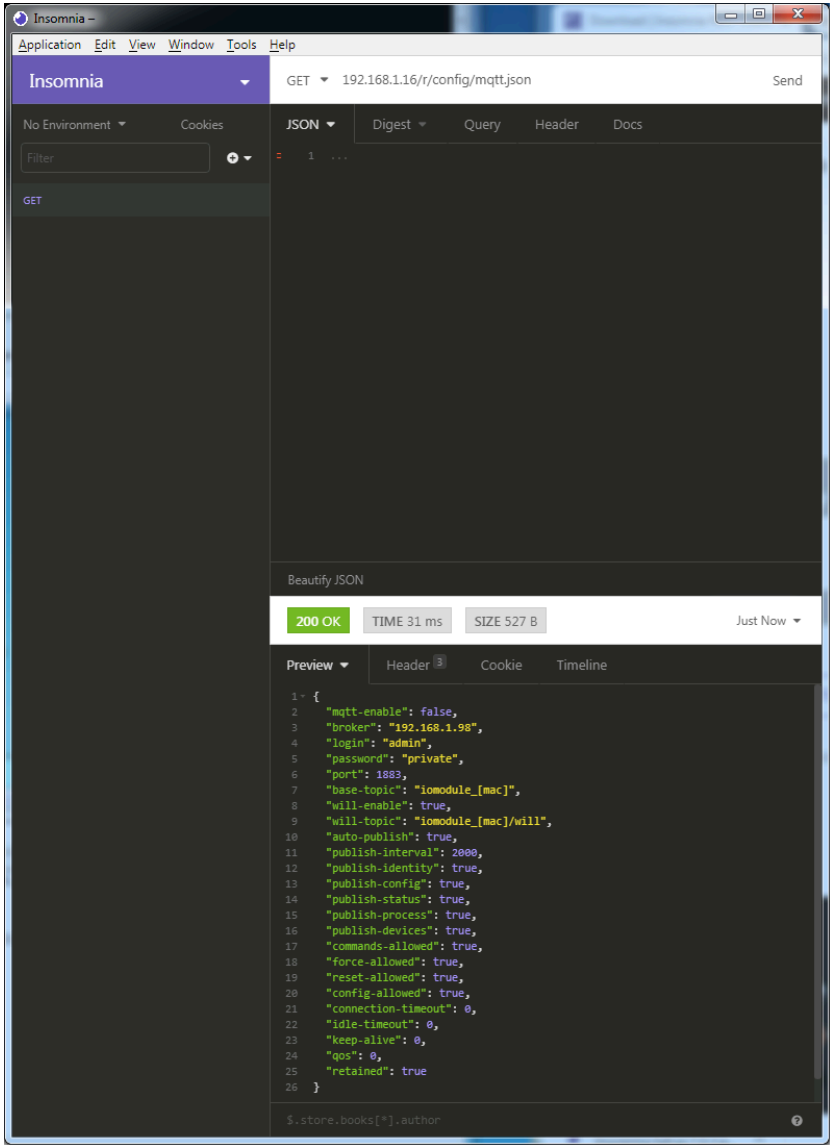
2. Configure MQTT:

POST: `[IP-address]/w/config/mqtt.json`



3. Read MQTT:

GET: `[IP-address]/r/config/mqtt.json`



11.2 OPC UA

OPC UA functions are **only** applicable for the following UNITRONIC® ACCESS variants:

- MP08IOLA08DIO

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. UNITRONIC® ACCESS provides an OPC UA server on field device level, with which an OPC UA client can connect for information exchange secure in transmission.

For OPC UA, we comply (apart from the exceptions listed below) with the IO-Link Companion Specification, which can be downloaded from https://lapp.com or directly from io-link.com.

Attention: When using OPC UA, the MQTT protocol must be disabled.

Feature	Support
Managing IODDs (chapter 6.1.6 in the specification)	Not supported
Mapping IODD information to OPC UA ObjectTypes (chapter 6.3 in the specification)	Not supported
IOLinkIODDDeviceType (chapters 7.2 ff. in the specification)	Not supported
ObjectTypes generated based on IODDs (chapters 7.3 ff. in the specification)	Not supported
Creation of Instances based on ObjectTypes generated out of IODDs (chapter 7.4 in the specification)	Not supported
IODDManagement Object (chapter 8.2 in the specification)	Not supported
RemoveIODD Method (chapter 8.3 in the specification)	Not supported

Table 33: Non-supported OPC UA features according to the IO-Link Companion Specification

11.2.1 OPC UA configuration

In **delivery state**, OPC UA functions are **disabled**. The OPC UA Server can be configured either using the Web interface or directly via a JSON Object sent in an HTTP/HTTPS request. For more information see OPC UA configuration - Quick start guide on page 99.

The configuration URL is:

`http://[ip-address]/w/config/opcua.json`

The configuration can also read back as a JSON file:

`http://[ip-address]/r/config/opcua.json`

The configuration is a JSON object. Each JSON member is a configuration element. The object must not contain all elements. Only the provided elements will be changed. All configuration changed applies only after a device restart.

There are the following configuration elements (default values in bold):

Element	Data type	Description	Example data
port	integer	Server port for the OPC UA server.	0, **4840**, 0xFFFF
opcua-enable	boolean	Master switch for the OPC UA server.	true / **false**
anon-allowed	boolean	If true, anonymous login is allowed.	**true** / false
commands-allowed	boolean	Master switch for OPC UA commands. If false there will be no writeable OPC UA objects.	true / **false**
force-allowed	boolean	If true, the device accepts force commands via OPC UA.	true / **false**
reset-allowed	boolean	If true, the device accepts restart and factory reset commands via OPC UA.	true / **false**
config-allowed	boolean	If true, the device accepts configuration changes via OPC UA.	true / **false**

Table 34: OPC UA Configuration

All configuration elements are optional and do not need a specific order. Not every element is required to be sent. This means that only configuration changes will be taken over.

Optional: The configuration parameters of OPC UA can be set directly via the Web interface. It is possible to download the Web interface for sharing with other devices.

Response:

The resulting response is a JSON object with a status field. Status should be "0" if no error occurred and "-1" if there is an error.

In case of an error, the response contains an error array.

The error array contains an error object for each error occurred. The object consists of a field "Element" which names the config element which caused the error, and a field "Message" for the error message.

Examples:

```
{"status": -1, "error": [{"Element": "upcua-enable", "Message": "Boolean
 expected"}]}

{"status": 0}

{"status": -1, "error": [{"Element": "root", "Message": "Not a JSON
 object"}]}
```

11.2.2 OPC UA address space

OPC UA provides different services on the UNITRONIC® ACCESS 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 UNITRONIC® ACCESS devices. The objects and information displayed depend on the device variant used.

11.2.3 OPC UA configuration - Quick start guide

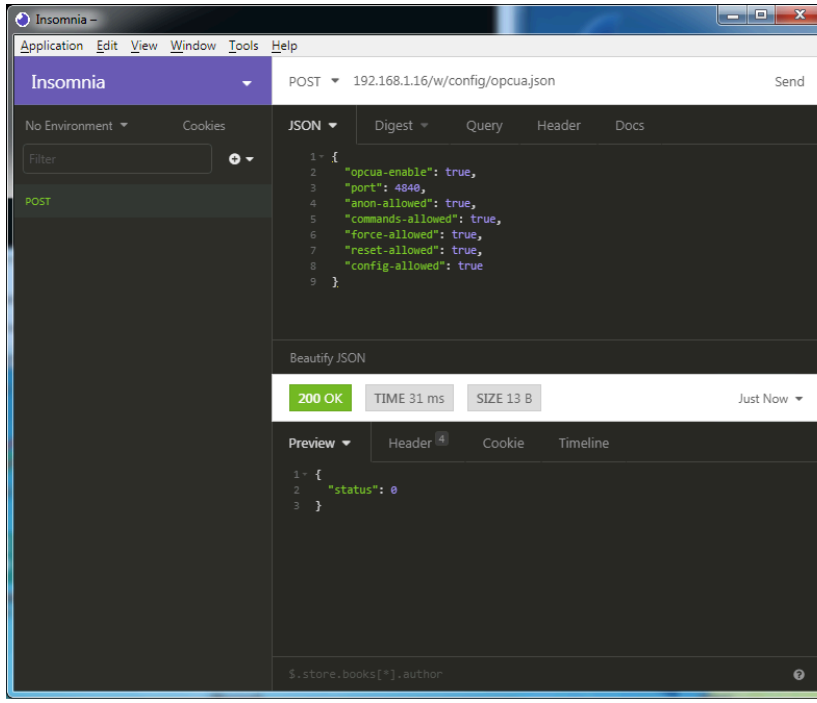
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

11.2.3.1 OPC UA configuration via JSON

1. Depending on your application case, download and install *Insomnia* or a comparable application: https://insomnia.rest/download/

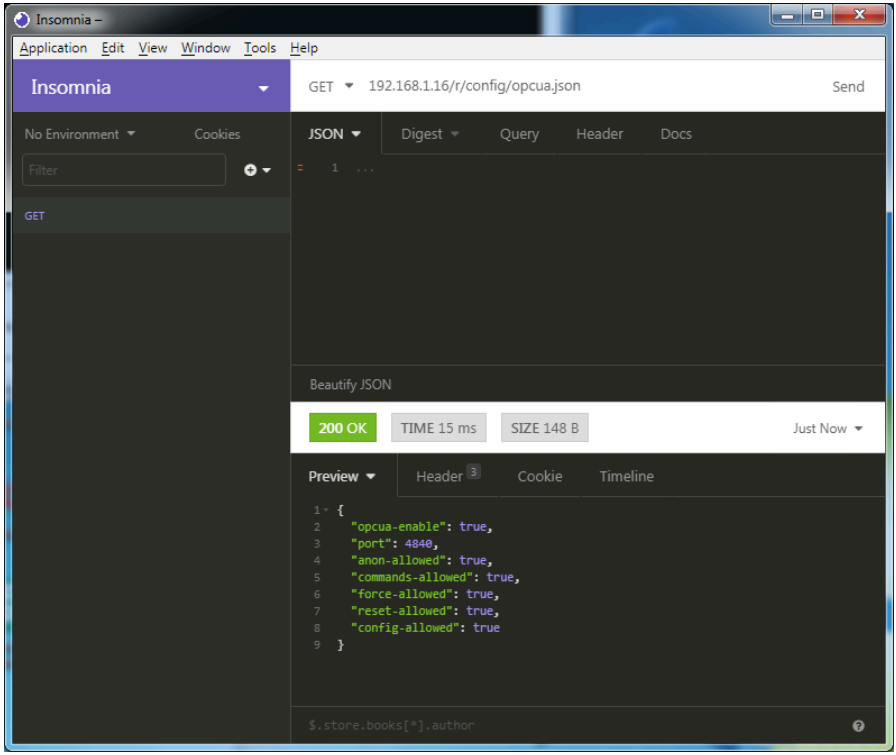
2. Configure OPC UA:

POST: `[IP-address]/w/config/opcua.json`



3. Read OPC UA:

GET: `[IP-address]/r/config/opcua.json`



11.3 REST API

The Representational State Transfer – Application Programming Interface (REST API) is a programmable interface which uses HTTP/HTTPS requests to GET and POST data. This enables the access to detailed device information.

For all UNITRONIC® ACCESS variants, the REST API can be used to read the device status. For the UNITRONIC® ACCESS multi-protocol variants, the REST API can also be used to write configuration and forcing data.

There are two different REST API standards you can use for the requests:

1. A standardized REST API that has been specified by the IO-Link Community and is described separately:

   `JSON_Integration_10222_V100_Mar20.pdf`

   Please download the file from https://lapp.com or directly from io-link.com.

   Attention: Consider the following table to get an overview of the supported features of the IO-Link specification:

Feature		Supported
Gateway	GET /identification	Yes
	GET /capabilities	Yes
	GET /configuration	Yes
	POST /configuration	Yes
	POST /reset	Yes
	POST /reboot	Yes
	GET /events	Yes
Master	GET /masters	Yes
	GET /capabilities	Yes
	GET /identification	Yes
	POST /identification	Yes

Feature		Supported
Port	GET /ports	Yes
	GET /capabilities	Yes
	GET /status	Yes
	GET /configuration	Yes
	POST /configuration	Yes
	GET /datastorage	Yes
	POST /datastorage	Yes
Devices	GET /devices	Yes
	GET /capabilities	Yes
	GET /identification	Yes
	POST /identification	Yes
	GET /processdata/value	Yes
	GET /processdata/getdata/value	Yes
	GET /processdata/setdata/value	Yes
	POST /processdata/value	Yes
	GET /parameters	Not supported
	GET /parameters/{index}/subindices	Not supported
	GET /parameters/{parameterName}/subindices	Not supported
	GET /parameters/{index}/value	Not supported
	GET /parameters/{index}/subindices/{subindex}/value	Not supported
	GET /parameters/{parameterName}/value	Not supported
	GET /parameters/{parameterName}/subindices/{subParameterName}/value	Not supported
	POST /parameters/{index}/value	Not supported
	POST /parameters/{parameterName}/value	Not supported
	POST /parameters/{index}/subindices/{subindex}/value	Not supported
	POST /parameters/{parameterName}/subindices/{subParameterName}/value	Not supported
	POST /blockparametrization	Yes
	GET /events	Yes

Feature		Supported
IODD	GET /iodds	Not supported
	POST /iodds/file	Not supported
	DELETE /iodds	Not supported
	GET /iodds/file	Not supported

Table 35: Support of REST API features according to the IO-Link specification

2. A customized LAPP REST API that is described in the following chapters.

11.3.1 Standard device information

Request method:	http GET
Request URL:	`<ip>/info.json`
Parameters	n.a.
Response format	JSON

The goal of the "Standard device information" request is to get a complete snapshot of the current device status. The format is JSON. For IO-Link devices, all ports with connected IO-Link device information are included.

11.3.2 Structure

Name	Data type	Description	Example
name	string	Device name	"MP08IOLA08DIO"
order-id	string	Ordering number	"381166717"
fw-version	string	Firmware version	"V.11.2.0.0 - 08.08.2024"
hw-version	string	Hardware version	"V.1.00"
mac	string	MAC address of the device	"7C F9 5C 4C CC CE"
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
iol	IOL Object	Contains all IO-Link related information such as events, port states, device parameters.	
iol/diagGateway	array of DIAG	Array of currently active device/ gateway related events	
iol/diagMaster	array of DIAG	Array of currently active IOL-Master related events	
iol/ports	array of PORT (8)	Contains one element for each IO-Link port	
CHANNEL Object			
name	string	Name of channel	
type	number	Hardware channel type as number: 0 = DIO 1 = Input 2 = Output 3 = Input/Output 4 = IO-Link 5 = IOL AUX 6 = IOL AUX with DO 7 = IOL AUX with DO. Can be deactivated. 8 = Channel not available	
type_text	string	Textual representation of the channel type	
config	number	Current configuration of the channel: 0 = DIO 1 = Input 2 = Output 3 = IO-Link 4 = Deactivated 5 = IOL AUX	
config_text	string	Textual representation of the current config	
inputState	boolean	Input data (producing data) bit to the PLC	
outputState	boolean	Output data bit to the physical output pin	

Name	Data type	Description	Example
forced	boolean	True, if the output pin of this channel is forced	
simulated	boolean	True, if the input value to the PLC of this channel is simulated	
actuatorDiag	boolean	True, if the output is in short circuit / overload condition	
sensorDiag	boolean	True, if the sensor supply (Pin 1) is in short circuit / overload condition	
maxOutputCurrent_mA	number	Maximum output current of the output in mA	
current_mA	number	Measured current of the output in mA (if current measurement is available)	
voltage_mV	number	Measured voltage of this output in mV (if voltage measurement is available)	
PORT Object			
port_type	string	Textual representation of the IO-Link port type	
iolink_mode	number	**Current port mode:** 0 = Inactive 1 = Digital output 2= Digital input 3 = SIO 4 = IO-Link	
iolink_text	string	Textual representation of the current port mode	"Digital Input"
aux_mode	number	Indicates the configured mode for the Pin 2: 0 = No AUX 1 = AUX output (always on) 2 = Digital output (can be controlled by cyclic data) 3 = Digital input	
aux_text	string	Textual representation of the current aux mode	"AUX Output"
cq_mode	number	Port mode according to IOL specification	
iq_mode	number	Pin2 mode according to IOL specification	

Name	Data type	Description	Example
port_status	number	Port status according to IOL specification	
ds_fault	number	Data storage error number	
ds_fault_text	string	Textual data storage error.	
device	DEVICE Object	IO-Link device parameters. → Null if no IO-Link communication active	
diag	array of DIAG (n)	Array of port related events	
DIAG Object			
error	number	Error code	
source	string	Source of the current error.	"device" "master"
eventcode	number	Event code according to IO-Link specification	
eventqualifier	number	Event qualifier according to IO-Link specification	
message	string	Error message	"Supply Voltage fault"
DEVICE Object		Standard parameters of the IOL-Device	
device_id	number		
vendor_id	number		
serial	string		
baudrate	string	Baudrate (COM1,2,3)	
cycle_time	number	Cycle time in microseconds	
input_len	array of numbers (n)	IOL input length in bytes	
output_len	array of numbers (n)	IOL output length in bytes	
input_data	array of numbers (n)	IOL input data	
output_data	array of numbers (n)	IOL output data	
pd_valid	number	"1", if IOL input data is valid	
pdout_valid	number	"1", if IOL output data is valid	
FORCING Object		Forcing information of the device	
forcingActive	boolean	Force mode is currently active	

Name	Data type	Description	Example
forcingPossible	boolean	True, if forcing is possible and force mode can be activated	
ownForcing	boolean	True, if forcing is performed by REST API at the moment	
forcingClient	string	Current forcing client identifier	
digitalOutForced	array of numbers (2)	The force values of all 16 digital output channels.	
digitalOutMask	array of numbers (2)	The forcing mask of all 16 digital output channels.	
digitalInForced	array of numbers (2)	The force values of all 16 digital input channels.	
digitalInMask	array of numbers (2)	The forcing mask of all 16 digital input channels.	

11.3.3 Configuration and forcing

Method: POST

URL: `<ip>/w/force.json`

Parameters: None

Post-Body: JSON Object

Property	Data type	Example values	Description
forcemode	boolean	true / false	Forcing authority on/off
portmode	array (Port mode object)		
digital	array (Digital object)		
iol	array (IOL object)		

Table 36: Root object

Property	Data type	Example values	Remarks
port	integer	0..7	
channel	string	"a","b"	optional default is "a"
direction	string	"dio","di","do","iol", "off", "aux"	
aux	string	"dio","di","do","iol", "off", "aux"	IOL only, but optional
inlogica	string	"no","nc"	
inlogicb	string	"no","nc"	
inputlatch	bool	true / false	enable/disable input latch, optional
inputext	integer	Depends on the fieldbus: ▶ eip: 0 (off) - 255 (ms) ▶ ethercat: 0 (off) - 255 (ms) ▶ pns: 0 (off) - 255 (ms) ▶ cclink: 0 (off) - 255 (ms) ▶ mbtcp: 0 (off) - 255 (ms)	set input extension, optional
inputfilter	integer	0 .. 255	set input filter, optional

Table 37: 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 38: Digital object

Property	Data type	Example values	Remarks
port	integer	0..7	
output	array[integer] or null to clear forcing	[55,88,120]	Output forcing
input	array[integer] or null to clear forcing	[20,0,88]	Input simulation to PLC

Table 39: IOL object

11.3.4 Reading and writing ISDU parameters

The *Indexed Service Data Unit* (ISDU) provides a highly flexible message format, which can contain single or multiple commands.

UNITRONIC® ACCESS IOL-Masters with IIoT support reading and writing ISDU parameters from connected IOL-Devices. It is possible to do this as a bulk transfer by reading and writing of multiple ISDU parameters via a single request.

11.3.4.1 Reading ISDU

Method: POST

URL: `<ip>/r/isdu.json`

Parameters: port (0-7)

Example:

```
192.168.1.20/r/isdu.json?port=5
```

Post-Body: JSON array of read ISDU object

Property	Data type	Example values	Remarks
ix	integer	0-INT16	Index to be read
subix	integer	0-INT8	Subindex to be read

Table 40: Read ISDU object

Property	Data type	Example values	Remarks
status	integer	0, -1	0 = no error, -1= an error occured
message	string		Error Message if error occured
data	array (Read ISDU data object)		data, if no error occured. otherweise null

Table 41: Read ISDU response object

Property	Data type	Example values	Remarks
ix	integer	0-INT16	Index that was read
subix	integer	0-INT8	Subindex that was read
status	integer	0, -1	0 = no error, -1= an error occured
eventcode	integer		IOL eventcode if status is -1
data	array[integer]		data, if no error occured. otherweise null

Table 42: Read ISDU data object

11.3.4.2 Writing ISDU

Method:	POST
URL:	`<ip>/w/isdu.json`
Parameters:	port (0-7)
Post-Body:	JSON array of write ISDU object

Property	Data type	Example values	Remarks
ix	integer	0-INT16	Index to be read
subix	integer	0-INT8	Subindex to be read
data	array[integer]		Data to be written

Table 43: Write ISDU object

Response: Write ISDU response object

Property	Data type	Example values	Remarks
status	integer	0, -1	0 = no error, -1= an error occured
message	string		Error Message if error occured
data	array (Write ISDU data object)		data, if no error occured. otherweise null

Table 44: Write ISDU response object

Property	Data type	Example values	Remarks
ix	integer	0-INT16	Index that was written
subix	integer	0-INT8	Subindex that was written
status	integer	0, -1	0 = no error, -1= an error occured
eventcode	integer		IOL eventcode if status is -1

Table 45: Write ISDU data object

Attention: For UNITRONIC® ACCESS device variants with HTTPS feature, https:// must be used in front of <ip> for every REST API.

11.3.5 Upload and process an IODD file

The REST API supports IODD file upload to the IO-Link Master.

Perform the following work steps:

1. Check file upload status

   Send request: GET file_upload

   Purpose: Get file upload status to check if there is another upload in progress.

   Expected response:

```
{
"status": 0,
"progress": 0,
"name": "",
"action": 0,
"upid": 0,
"errid": 0,
"errstr": "",
"pschr": 0
}
```

   Check the status ID. If status is '0', you can start a new iodd upload process. For reference, see tables Table 46: Status ID and meaning on page 119 and Table 47: Error ID and meaning on page 120. Proceed with the next step.

2. Initiate file upload

   Send request: POST file_upload

   Content-Type: application/json

   Purpose: Send details about the file to be uploaded.

   Expected response:

```
{"size":<total size>,"name":"<file name>",
"action":"iodd", "upid":
<upload id>}
{
"size": <total size>,
"name": "<file name>",
"action": "iodd",
"upid": <upload id>
}
```

The upload id (upid) is a number used by the backend to identify a specific upload and parsing process. It has to be used as a query parameter in the following steps.

The action will allways be iodd.

The size is the total size of the file in bytes.

The correct content type has to be set.

Note: Remember the upload ID (upid) for subsequent steps

3. Upload file content

   Send request: POST file_upload?upid=<value> → Use the upid value from Step 2.

   Content-Type: application/octet-stream → The correct content type has to be set.

   Purpose: Send file or file chunks (max chunk size: 64KB).

   Attention: Sending file chunks bigger than 64KB will result in unresponsive behavior.

4. Monitor upload status

   Send request: GET file_upload?upid=<value> → Use the upid value from Step 2.

   Purpose: Get the current file upload status.

   Expected response:

```
{
"status": <status id value>,
"progress": <percentage>,
"name": "<file name given in step 2>",
"action": "iodd",
"upid": <upload id chosen in step 2>,
"errid": <error id>,
"errstr": "",
"pschr": <count of parsed characters>
}
```

   Repeat this step until the status becomes 'idle'. For some states this request triggers the necessary transitions in the internal state machine.

Only after the backend is sure that the correct client identified by its upid received the action finished or error state it will transition to the next one, the idle state.

The fields now show values depending on what was sent in step 2 and on the current process status.

Status ID	Status
0	File upload idle. New upload can be triggered.
1	File upload started.
2	File upload in progress.
3	File upload finished.
4	Error during file upload.
5	File upload timeout.
6	IODD parsing started.
7	IODD parsing finished.
8	IODD parsing error.
9	IODD parsing canceled.

Table 46: Status ID and meaning

ID	Error
0	No error.
1	Json parsing error.
2	Json type error.
4	Upload error.
5	File opening error.
6	File writing error.
7	Thread creating error.
8	Error during file copy.
9	Upload timeout.
10	Upload size exceeded.
11	Unknown action.
12	No upload id.
13	IODD paasing error.
14	Internal error.
15	IODD store full. Delete an IODD before uploading a new one.
16	Internal error.
17	IODD file CRC error.
18	Standard IODD file crc error.
19	No available space for parsing.

Table 47: Error ID and meaning

11.3.6 Example: Reading ISDU

ISDU read request

```
[
 {"ix":5,"subix":0},
 {"ix":18,"subix":0},
 {"ix":19,"subix":0},
 {"ix":20,"subix":0}
]
```

Response

```
{
 "message":"OK",
 "data":
[
 {"ix":5,"subix":0,"status":-1,"eventcode":32785},
 {"ix":18,"subix":0,"data":[79,68,83,49,48,76,49,46,56,47,76,65,54,44,50,
        48,48,45,77,49,50],"status":0},
 {"ix":19,"subix":0,"data":[53,48,49,50,57,53,51,53],"status":0},
 {"ix":20,"subix":0,"data":[100,105,115,116,97,110,99,101,32,115,101,110,
        115,111,114],"status":0}
],
"status":0}
```

11.3.7 Example: Writing ISDU

ISDU write request

```
[
 {"ix":24,"subix":0,"data":[97,98,99,100,101,102]},
 {"ix":9,"subix":0,"data":[97,97,97,97,97,98]}
]
```

Response

```
{
 "message":"OK",
 "data":[
 {"ix":24,"subix":0,"status":0},
 {"ix":9,"subix":0,"eventcode":32785,"status":-1}
],
"status":0}
```

11.4 CoAP server

The CoAP server functions are **only** applicable for the following UNITRONIC® ACCESS variants:

- MP08IOLA08DIO

The **Co**nstrained **A**pplication **P**rotocol (CoAP) is a specialized Internet application protocol for constrained networks such as lossy or low power networks. CoAP is useful especially in M2M (Machine to Machine) communication and can be used to translate simplified HTTP/HTTPS requests of low speed networks.

CoAP is based on the Server-Client principle and a service layer protocol that lets nodes and machines communicate with each other. The UNITRONIC® ACCESS multi-protocol variants provide CoAP server functionalities via a REST API interface over UDP.

11.4.1 CoAP configuration

In delivery state, CoAP functions are *disabled*. The CoAP server can be configured either using the Web interface or directly via a JSON object sent in an HTTP/HTTPS request. For more information see chapter CoAP configuration - Quick start guide on page 126.

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 48: CoAP configuration

CoAP response:

The resulting response is a JSON object with a "status" field. Status should be "0" if no error occurred, and "-1" if there is an error.

In case of an error, the response contains an error array.

The error array contains an error object for each error occurred. The object consists of a field "Element" which names the config element that caused the error, and of a field "Message" for the error message.

Examples:

```
{"status": -1, "error": [{"Element": "upcua-enable", "Message": "Boolean
 expected"}]}

{"status": 0}

{"status": -1, "error": [{"Element": "root", "Message": "Not a JSON
 object"}]}
```

11.4.2 REST API access via CoAP

A connection to the CoAP server running on the UNITRONIC® ACCESS multi-protocol variants can be established via the following URL:

`coap://[ip-address]:[port]/[api]`

For UNITRONIC® ACCESS, the following REST API Requests (JSON format) can be accessed via a CoAP endpoint:

Type	API	Note
GET	/r/status.lr	
GET	/r/system.lr	
GET	/info.json"	
GET	/r/config/net.json	
GET	/r/config/mqtt.json	
GET	/r/config/opcua.json	
GET	/r/config/coapd.json	
GET	/r/config/syslog.json	
GET	/contact.json	
GET	/fwup_status	
GET	/iolink/v1/gateway/identification	
GET	/iolink/v1/gateway/capabilities	
GET	/iolink/v1/gateway/configuration	
GET	/iolink/v1/gateway/events	
GET	/iolink/v1/masters	
GET	/iolink/v1/masters/1/capabilities	
GET	/iolink/v1/masters/1/identification	
GET	/iolink/v1/masters/1/ports	
GET	/iolink/v1/masters/1/ports/{port_number}/capabilities	This API is available for all 8 ports. {port_number} should be between "1" and "8".
GET	/iolink/v1/masters/1/ports/{port_number}/status	This API is available for all 8 ports. {port_number} should be between "1" and "8".
GET	/iolink/v1/masters/1/ports/{port_number}/configuration	This API is available for all 8 ports. {port_number} should be between "1" and "8".
GET	/iolink/v1/devices/master1port{port_number}/identification	This API is available for all 8 ports. {port_number} should be between "1" and "8".

Type	API	Note
GET	/iolink/v1/devices/master1port{port_number}/capabilities	This API is available for all 8 ports. {port_number} should be between "1" and "8".
GET	/iolink/v1/devices/master1port{port_number}/processdata/getdata/value	This API is available for all 8 ports. {port_number} should be between "1" and "8".
GET	/iolink/v1/devices/master1port{port_number}/events	This API is available for all 8 ports. {port_number} should be between "1" and "8".

Table 49: REST API access via CoAP

11.4.3 CoAP configuration - Quick start guide

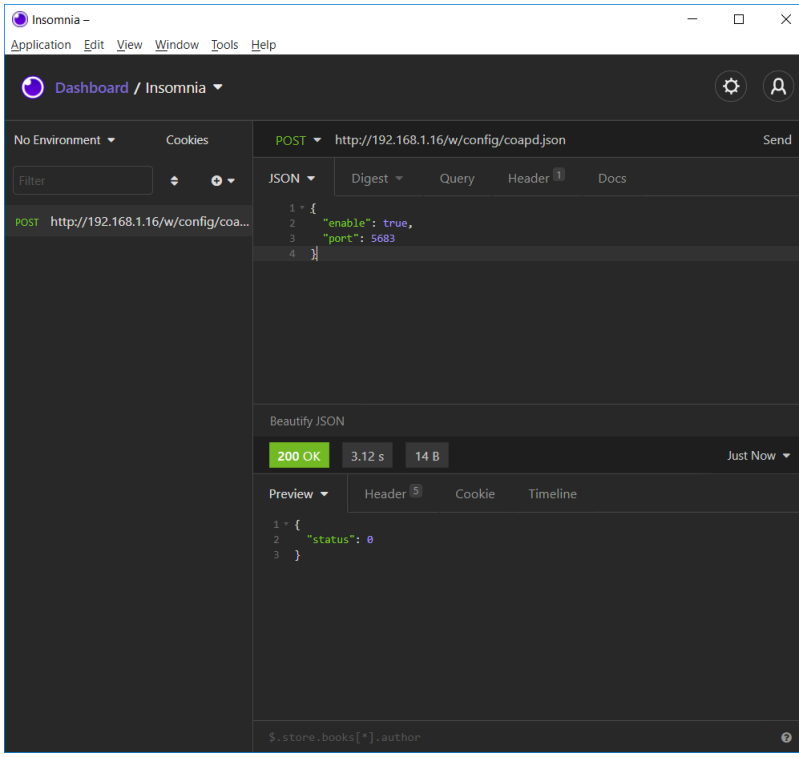
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

11.4.3.1 CoAP configuration via JSON

1. Depending on your application case, download and install *Insomnia* or a comparable application: https://insomnia.rest/download/

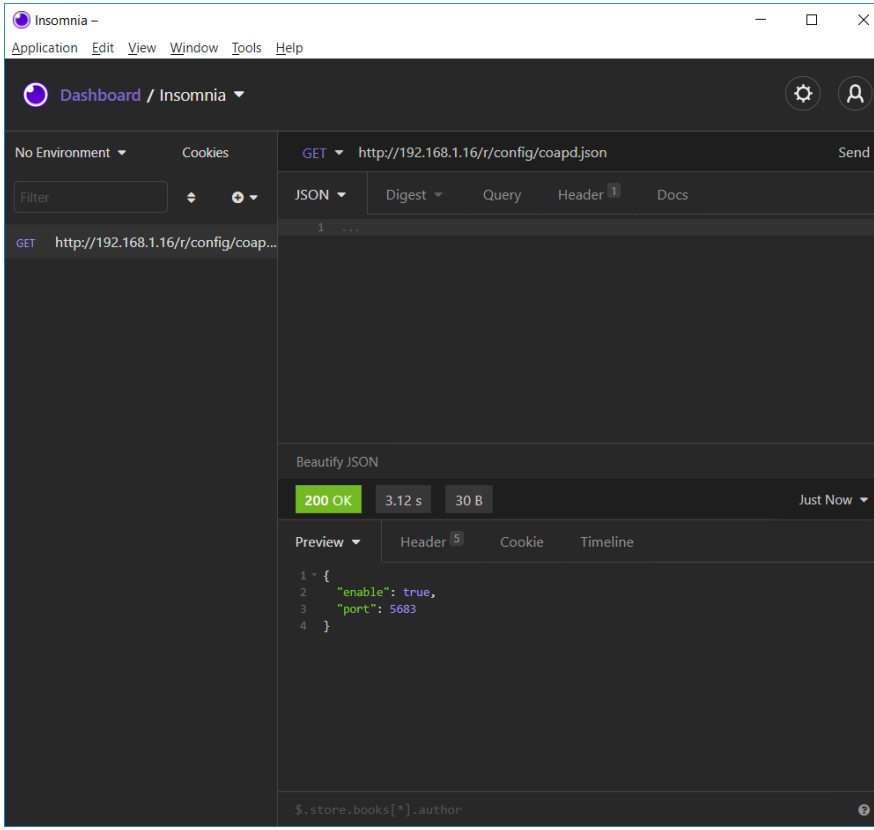
2. Configure CoAP:

POST: `[IP-address]/w/config/coapd.json`



3. Read CoAP configuration:

GET: `[IP-address]/r/config/coapd.json`



11.5 Syslog

Syslog functions are **only** applicable for the following UNITRONIC® ACCESS variants:

- MP08IOLA08DIO

The UNITRONIC® ACCESS multi-protocol variants provide a Syslog client which can connect with a configured Syslog server and is able to log messages.

Syslog is a platform-independent standard for logging messages. Each message contains a timestamp as well as information about the severity level and the subsystem. The Syslog protocol RFC5424 is based on the Server-Client principle and lets machines and devices send messages in the network and collect them centrally. (For more details on the used syslog standard, please refer to https://datatracker.ietf.org/doc/html/rfc5424.)

UNITRONIC® ACCESS supports the storage of 256 messages in a ring buffer which are sent to the configured Syslog server. When the ring is full with 256 messages, the oldest message is always replaced by the newly arriving messages. All messages can be saved on the Syslog server. The Syslog client of the IO-Link Master will not store any message permanently.

11.5.1 Syslog configuration

In **delivery state**, Syslog functions are **disabled**. The Syslog client can be configured either using the Web interface or directly via a JSON object sent in an HTTP/HTTPS request. For more information see chapter Syslog configuration - Quick start guide on page 131.

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 50: Syslog configuration

Syslog response:

The resulting response is a JSON object with a "status" field. Status should be "0" if no error occurred, and "-1" if there is an error.

In case of an error, the response contains an error array.

The error array contains an error object for each error occurred. The object consists of a field "Element" which names the config element that caused the error, and of a field "Message" for the error message.

Examples:

```
{"status": -1, "error": [{"Element": "upcua-enable", "Message": "Boolean
 expected"}]}

{"status": 0}

{"status": -1, "error": [{"Element": "root", "Message": "Not a JSON
 object"}]}
```

11.5.2 Syslog configuration - Quick start guide

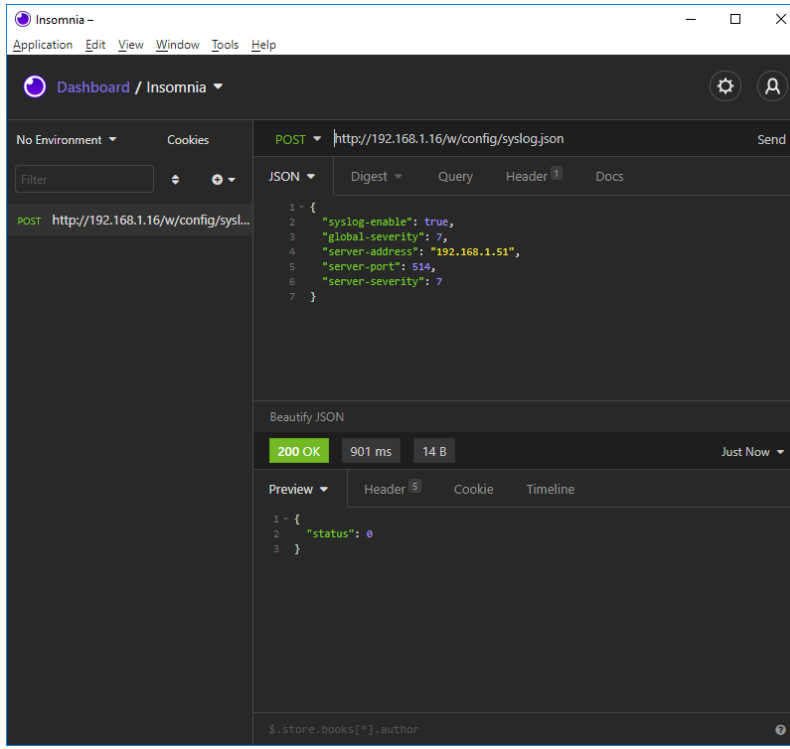
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

11.5.2.1 Syslog configuration via JSON

1. Depending on your application case, download and install *Insomnia* or a comparable application: https://insomnia.rest/download/

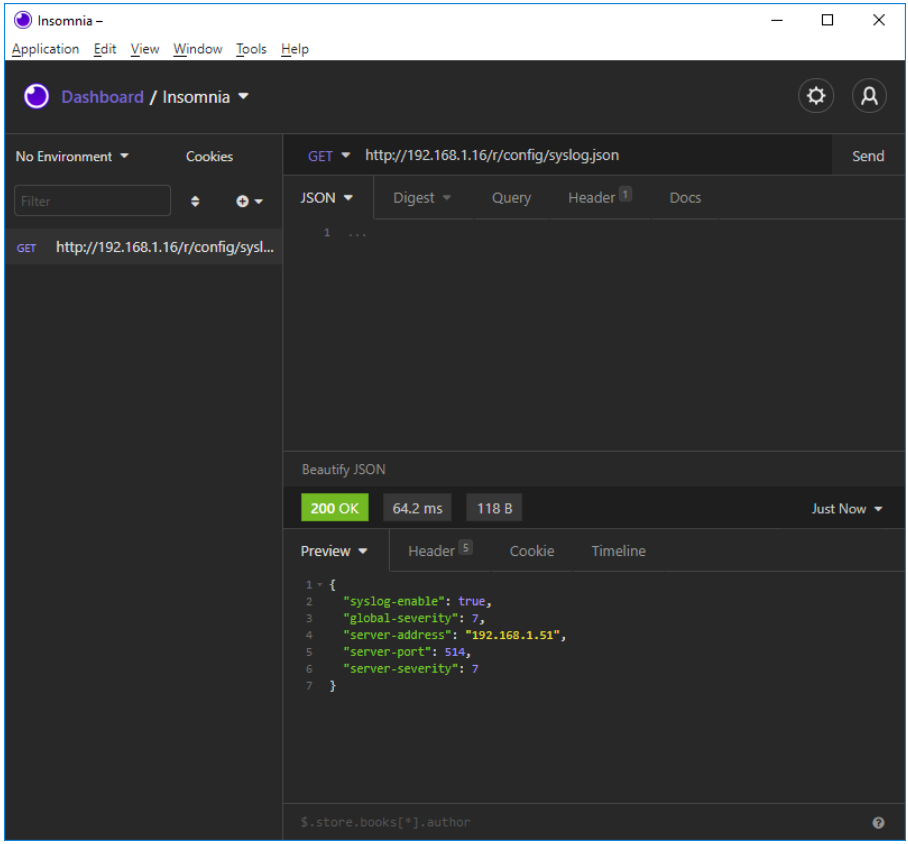
2. Configure Syslog:

POST: `[IP-address]/w/config/syslog.json`



3. Read Syslog configuration:

GET: `[IP-address]/r/config/syslog.json`



11.6 Network Time Protocol (NTP)

The NTP function is **only** applicable for the following UNITRONIC® ACCESS variant:

- MP08IOLA08DIO

The UNITRONIC® ACCESS multi-protocol variants provide an NTP client (version 3) which can connect with a configured NTP server and is able to synchronize the network time at a configurable interval.

NTP is a network protocol which uses UDP datagrams to send and receive timestamps in order to synchronize with a local clock. The NTP protocol RFC1305 is based on the Server-Client principle and exclusively supplies the synchronization with Coordinated Universal Time (UTC). (For more details on the used NTP standard, please refer to https://datatracker.ietf.org/doc/html/rfc1305.)

11.6.1 NTP configuration

In **delivery state**, the NTP client is **disabled**. The NTP client can be configured either using the Web interface or directly via a JSON object sent in an HTTP/HTTPS request. For more information see chapter NTP configuration - Quick start guide on page 135.

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 51: NTP configuration

NTP response:

The resulting response is a JSON object with a "status" field. Status should be "0" if no error occurred, and "-1" if there is an error.

In case of an error, the response contains an error array.

The error array contains an error object for each error occurred. The object consists of a field "Element" which names the configuration element that caused the error, and of a field "Message" for the error message.

Examples:

```
{"status": -1, "error": [{"Element": "ntpc-enable", "Message": "Boolean
 expected"}]}

{"status": 0}

{"status": -1, "error": [{"Element": "root", "Message": "Not a JSON
 object"}]}
```

11.6.2 NTP configuration - Quick start guide

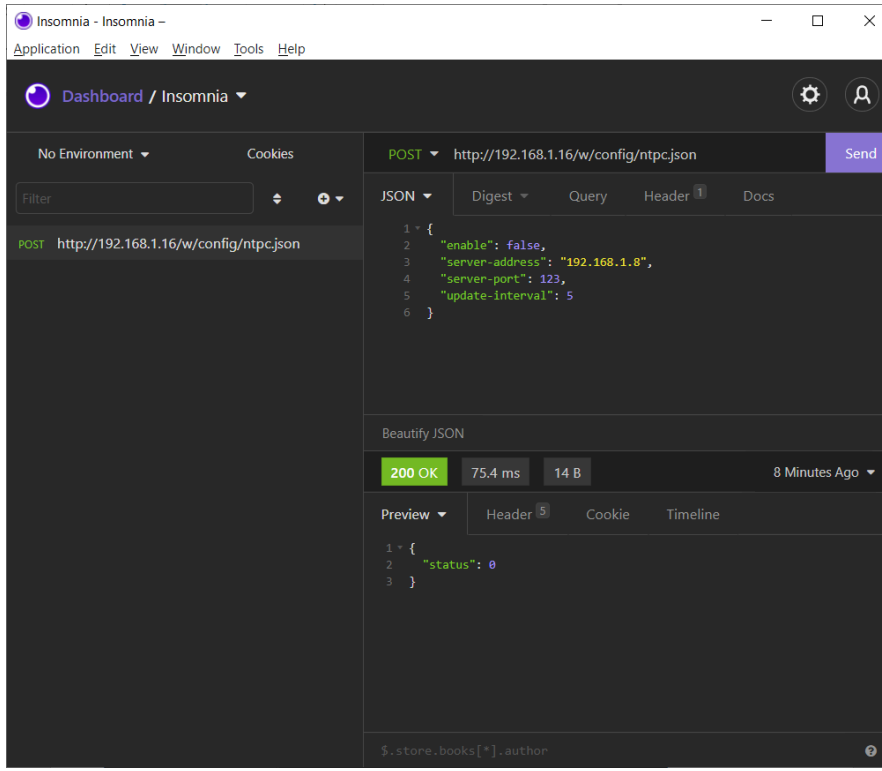
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

11.6.2.1 NTP configuration via JSON

1. Depending on your application case, download and install *Insomnia* or a comparable application: https://insomnia.rest/download/

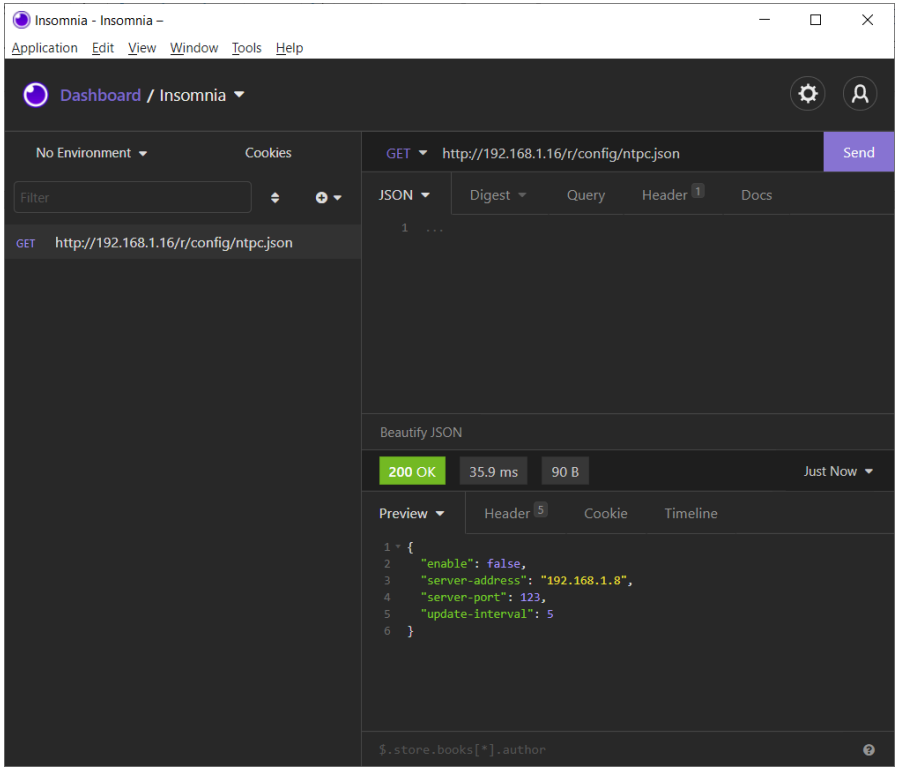
2. Configure NTP:

POST: `[IP-address]/w/config/ntpc.json`



3. Read NTP configuration:

GET: `[IP-address]/r/config/ntpc.json`



12 The integrated Web server

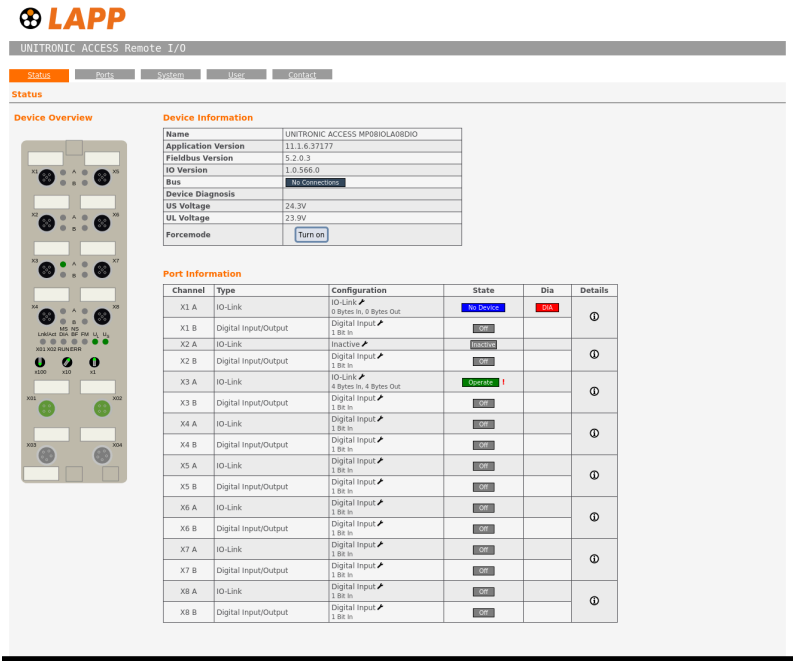
All device variants are equipped with an integrated Web server which makes functions for the device configuration and the display of status and diagnostic information available via a Web interface.

The Web interface provides an overview of the configuration and status of the device. It is also possible to use the Web interface to trigger a reboot, reset to the factory defaults, or perform a firmware update.

Enter "http://" or "https://" followed by the IP address, such as "http://192.168.1.5", in your Web browser’s address bar. If the status page of the device is not displayed, check your browser and firewall settings.

12.1 UNITRONIC® ACCESS MP08... variants

12.1.1 The Status page



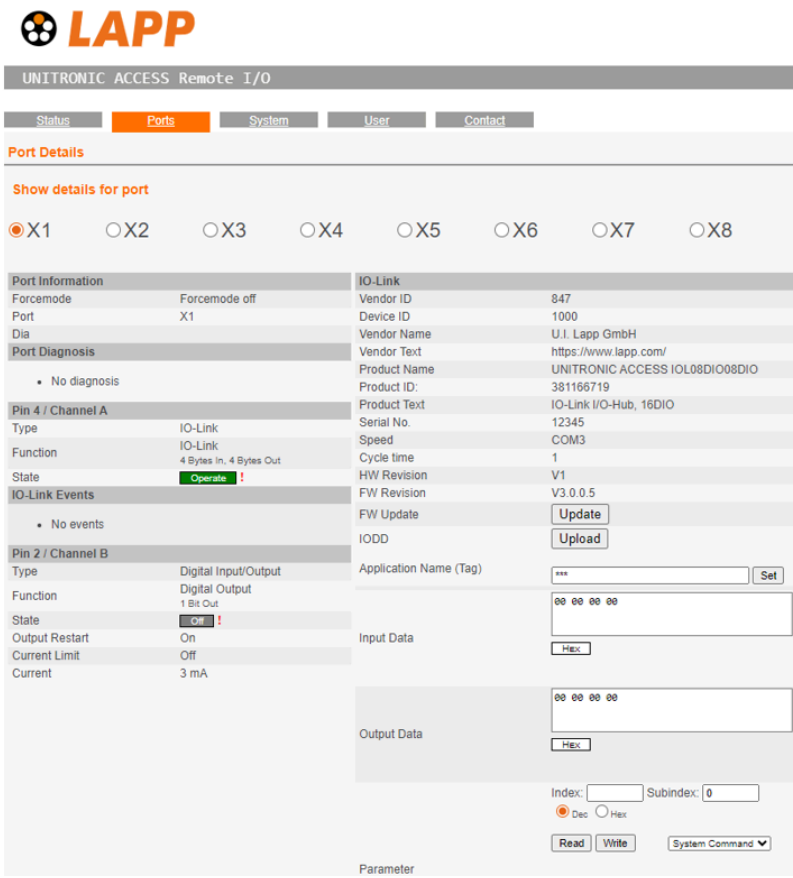
The status page provides a quick overview of the current state of the device.

The left side shows a graphical representation of the module with all its LEDs and the positions of the rotary encoding switches.

The right side shows the "Device Information" table with some basic data for the module; for example, the variant, the cyclic communication status and a diagnostic indicator. The indicator shows whether diagnostics for the module exist.

The "Port Information" table shows the configuration and state of the I/O ports.

12.1.2 The Ports page



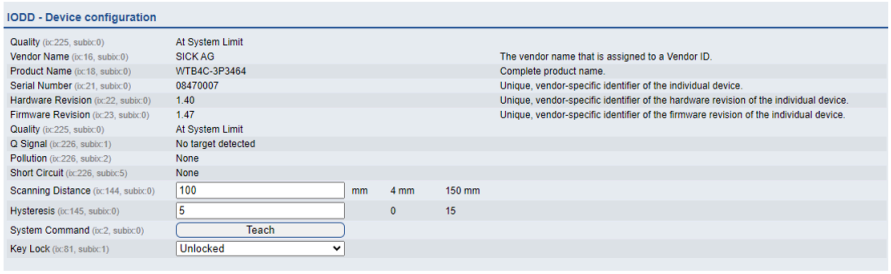
The page shows detailed port information. In the field **Port Diagnosis**, incoming and outgoing diagnostics are displayed as clear text. **Pin 2** and **Pin 4** contain information about the configuration and state of the port. For IO-Link ports, additional information relating to the connected sensor and the process data is displayed.

12.1.2.1 IODD upload

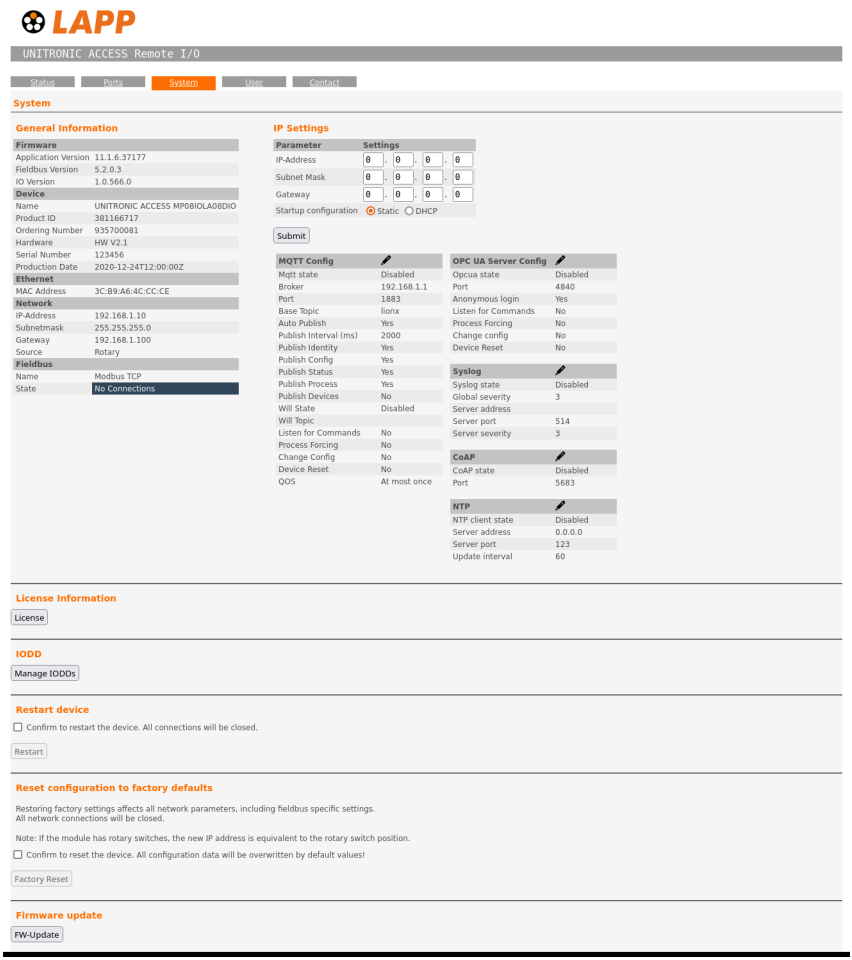
The button **UPLOAD** allows uploading an IODD file into the module, regardless of the device for which the IODD is designed.

The maximum number of IODDs is limited due to storage space. If there is no more space for a new IODD left, there will be a message of the detected error.

With the help of the IODD management page ("System" page), not used IODDs can be deleted. If there is already a matching IODD stored in the system for the connected IO-Link device, the button **CONFIGURE** is shown. By clicking this button, the page "IODD - Device configuration" will open, where the IO-Link device can be configured.



12.1.3 The System page



The System page shows the basic information for the module like Firmware version, Device information, Ethernet, Network and Fieldbus information.

IP Settings

Use this parameter to change the current IP address of the module.

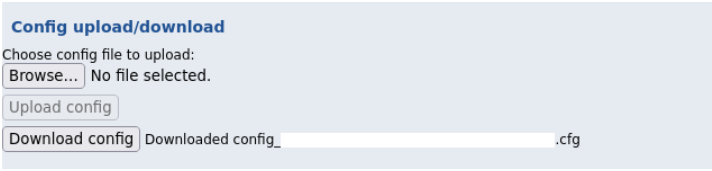
For PROFINET, this is only useful during commissioning. Normally, the PLC sets the IP address at start-up by detecting the PROFINET module via its device name.

12.1.3.1 License

This button opens a new window with Open Source Software information used in this product.

12.1.3.2 Config upload/download

With this feature, settings configured via the Web interface can be stored outside the I/O-Device (Download) for later Upload, e.g. after an I/O-Device change.



The following settings will be stored inside this file:

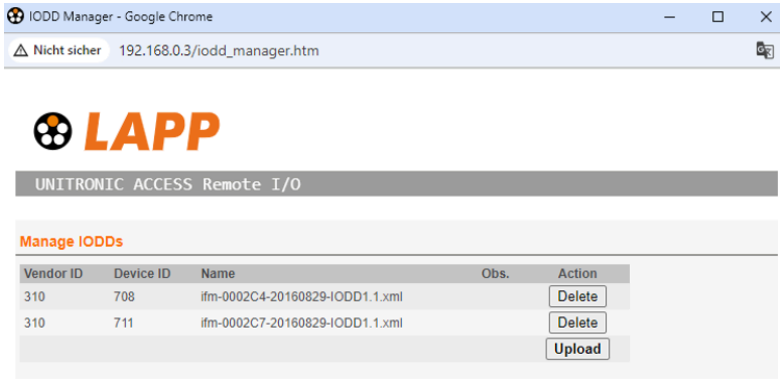
Scope	Type	Setting	Options	Details
Gateway		deviceID		To check device identity.
	iol	applicationSpecificTag		
	iol	functionTag		
	iol	locationTag		
		forcing		Enable/disable forcing
		channel_count		
		network configuration	ip	
			snMask	
			gw	
			source	1 - manual 2 - dhcp 3 - rotary 4 - dcp
Channel		index		channel index starting from 0
		channel configuration		0 - DIO 1 - IN 2 - OUT 3 - IOL 4 - AUX 5 - SAFIN 6 - SAFOUT
	iol	forced		
	iol	simulated		
	iol	force values		array
	iol	simulated		
	iol	sim values		array
	iol	validation	option	validation and backup
			vendorId	
			deviceId	
	digital	force		

Scope	Type	Setting	Options	Details
	digital	force value		
	digital	simulate		
	digital	sim value		
	digital	inputPolarity		
	digital	autorestart mode		
	digital	inputFilter100us		
	digital	currentLimit		
	digital	outputRestartMode		
	digital	failsafeMode		
	digital	surveillanceTimeouMs		
OPC UA		opcua	opcua-enable	
			port	
			anon-allowed	
			commands-allowed	
			force-allowed	
			reset-allowed	
			config-allowed	
	digital		dcu-allowed	
MQTT		mqtt	mqtt-enable	
			broker	
			login	
			password	
			port	
			base-topic	
			will-enable	
			will-topic	
			auto-publish	
			publish-interval	
			publish-identity	

Scope	Type	Setting	Options	Details
			publish-config	
			publish-status	
			publish-process	
	iol		publish-devices	
			commands-allowed	
			force-allowed	
			reset-allowed	
			config-allowed	
			qos	
SYSLOG		syslog	syslog-enable	
			global-severity	
			server-address	
			server-port	
			server-severity	
COAP		coap	enable	
			port	
NTP		ntpc	enable	
			server-address	
			server-port	
			update-interval	

12.1.3.3 IODD

The button **Manage IODDs** opens a new page for the IODD management on the I/O-Device. IODDs can be uploaded or deleted on this page, and all uploaded IODDs are listed here. For configuring connected IO-Link devices, open the related "Ports" page.



12.1.3.4 Restart device

The module initializes a software reset.

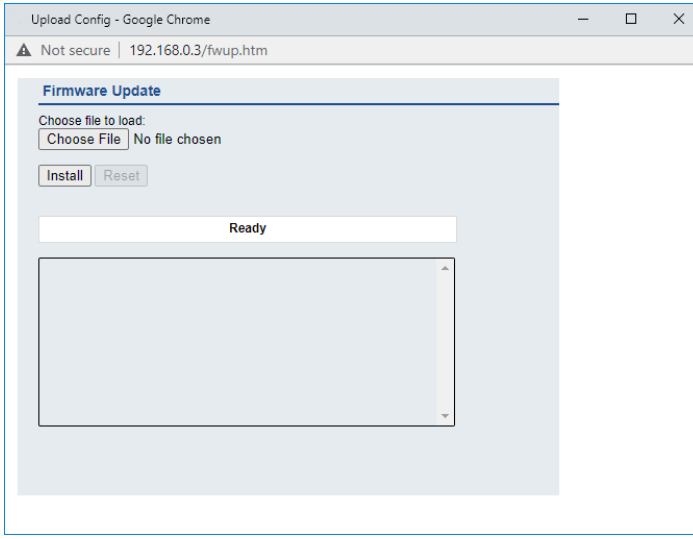
12.1.3.5 Reset configuration to factory defaults

The module restores to the default factory settings.

12.1.3.6 Firmware update

The module initializes a Firmware update.

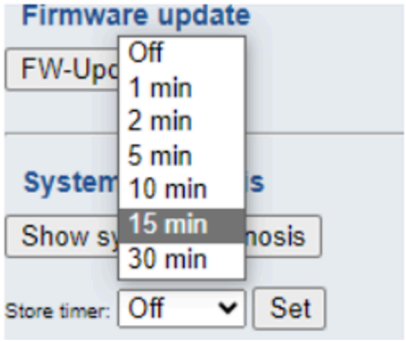
For a firmware update choose the **.ZIP* container available on our website or ask our support team. Afterwards follow the instructions shown on your screen.



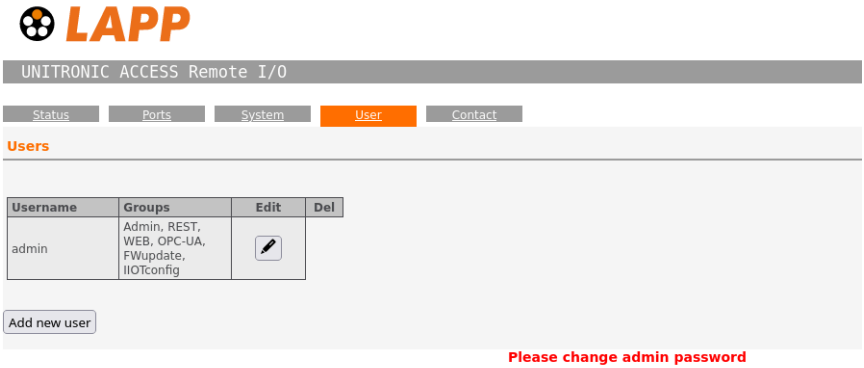
12.1.3.7 System diagnosis

All Syslog messages will be displayed in a ring buffer with 512 entries. By activation of the 'Store timer', the buffer content will be stored nonvolatile in the selected interval of 1, 2, 5, 10, 15 or 30 minutes.

The default value is 'Off' (no nonvolatile storage of system diagnosis ring buffer).



12.1.4 The User page



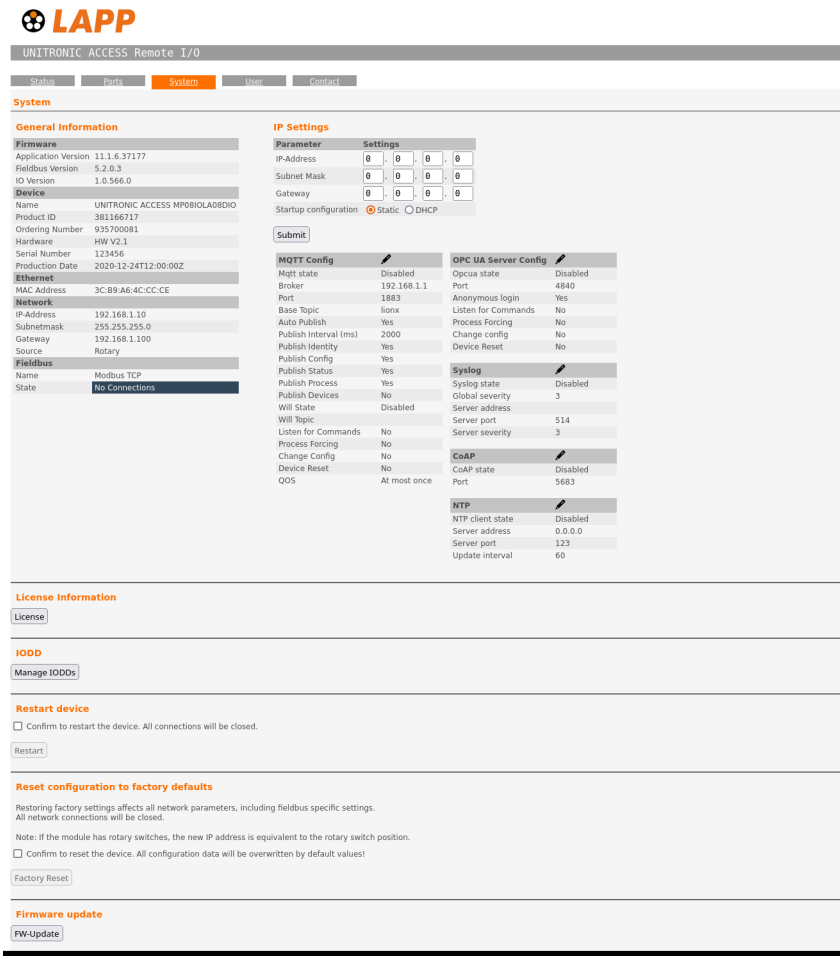
The User page provides the user management of the Web interface. New users with access rights "Admin" or "Write" can be added here. For security reasons please change the default admin password immediately after configuring the device.

Default user login data:

- User: admin
- Password: private

12.2 UNITRONIC® ACCESS single-protocol PN0.../EIP0.../EC0.../MTCP0.../CCL0... variants

12.2.1 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

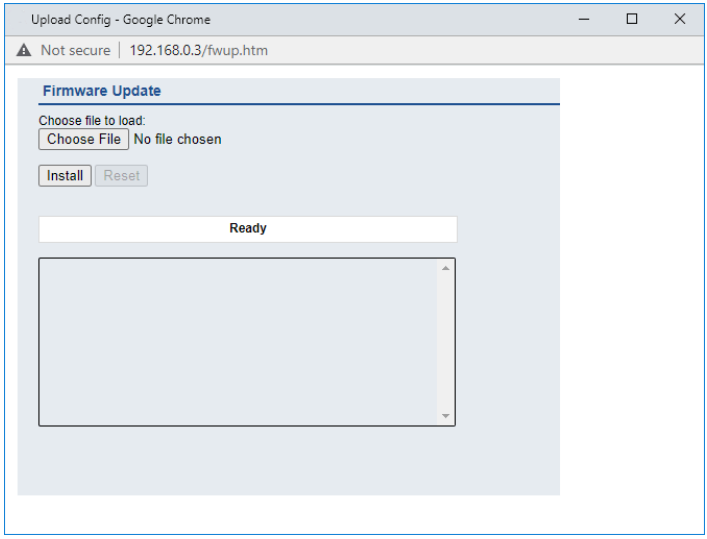
Use this parameter to change the current IP address of the module.

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.



13 IODD

IODD functions are **only** applicable for the following device variants:

- MP08IOLA08DIO

The **IO D**evice **D**escription (IODD) is a set of files formally describing an IO-Link Device. The IODD is created by the vendor and is mandatory for each IO-Link Device.

LAPP UNITRONIC® ACCESS IO-Link Masters with the "IODD on Module" functionality are ready to use IODDs in order to make the IO-Link Device configuration much more easier and the process data human readable in a better way. IODDs can be uploaded via the Web Interface and remanently stored on the IO-Link Master afterwards.

If a corresponding IO-Link Device is connected, the stored IODD is used to provide a user friendly configuration page, where all parameters of the device can be viewed and edited. Additionally, according to the IODD, the process data will also be formatted and displayed to the user.

13.1 IO-Link Device parameters and ISDU requests

Every IO-Link Device provides parameters that can be read and written via the special IO-Link service ISDU (**I**ndexed **S**ervice **D**ata **U**nit).

Every parameter is addressed by an index. Sub-indices are possible but optional. Some parameters (most of them read-only) are mandatory for IO-Link devices an can be found always on the same indices (See *Table B.8* in the *IO-Link Interface and System Specification*: https://io-link.com/share/Downloads/Package-2020/IOL-Interface-Spec_10002_V113_Jun19.pdf).

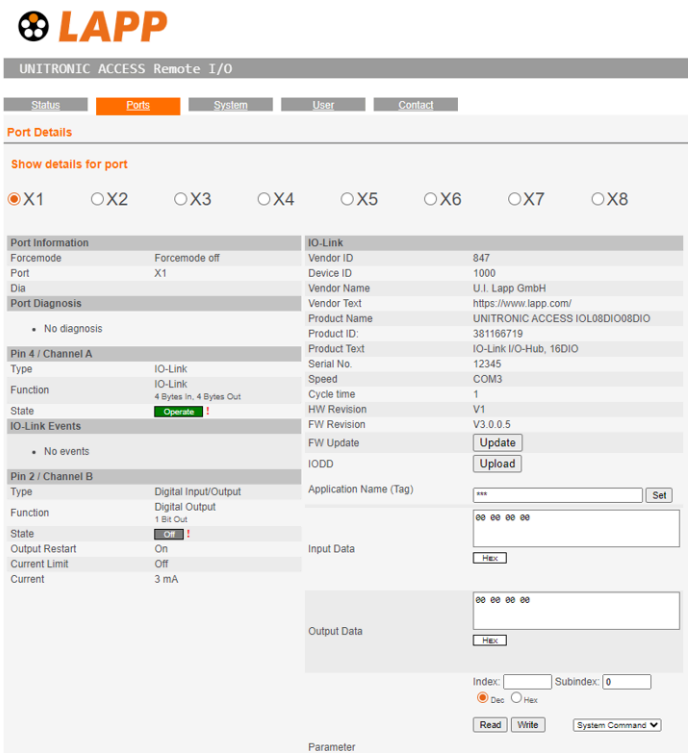
A vendor can use additional parameters and therefore more indices for their devices in order to provide additional configuration options. These vendor specific parameters can be described in an IODD. The "IODD on Module" feature of the UNITRONIC® ACCESS IO-Link Masters can read and parse this information out of an IODD and use it to provide the user viewing

and editing options for vendor specific parameters without any additional knowledge about the vendor specific device features.

13.2 Web GUI functionality

All of the "IODD on Module" features are accessible via the UNITRONIC® ACCESS Web interface.

13.2.1 Port Details page



The Port Details Page shows all information about the selected port. In the left column, all port and channel specific information is displayed. If the port is configured as IO-Link and there is an IO-Link Device connected, all IO-Link information for the connected device is displayed in the right column.

IODD buttons

The row called *IODD* provides access to the "IODD on Module" features. The button *UPLOAD* will let the user upload an IODD file into the module, regardless of the original device the IODD has been designed for.

The maximum number of IODDs is limited due to storage space. If there is no more space left for new IODDs, there will be an error message. In this case, navigate to the IODD Management page to delete IODDs which are no longer used.

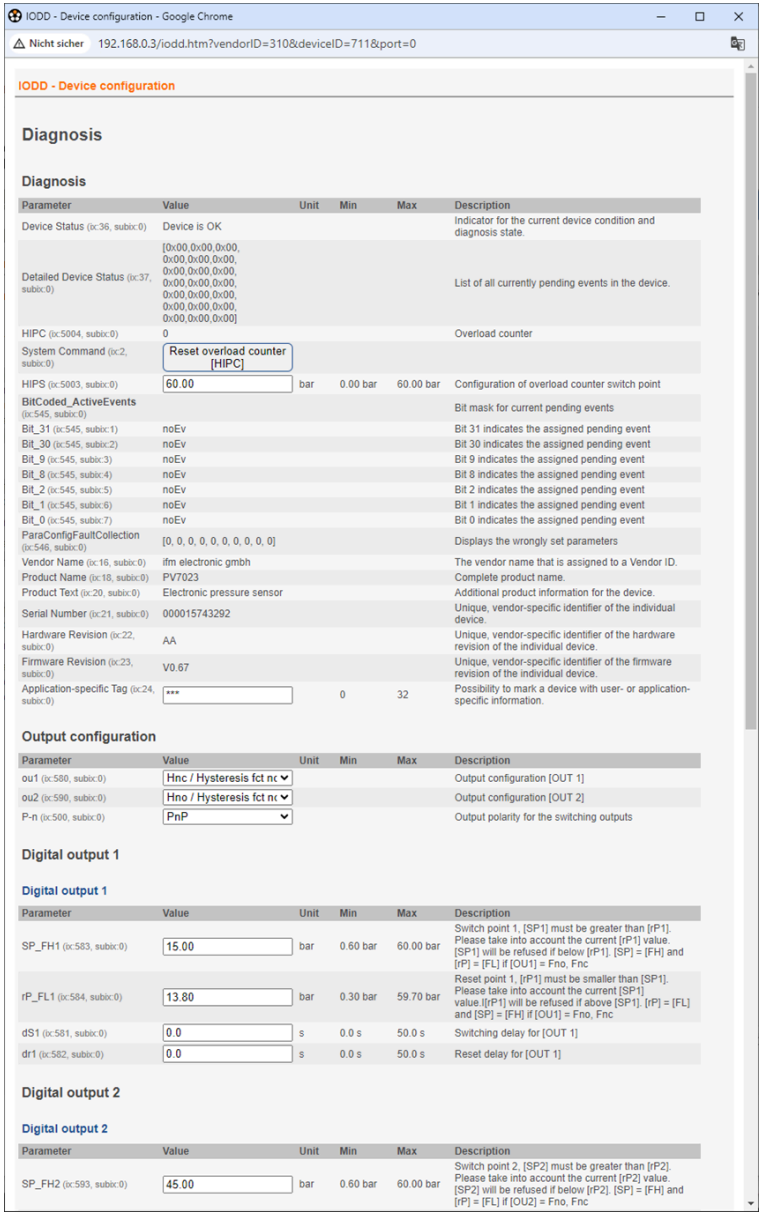
If there is a matching IODD for the currently connected device already stored in the system, the button *CONFIGURE* is shown in the interface. By clicking this button, the Parameter Page will open to configure the device.

Process data

For every connected IO-Link Device, raw process data for input and output direction (set of bytes) is on display.

If a matching IODD providing information about process data is already stored in the system, this data will also be displayed in a user-friendly format according to the IODD.

13.2.2 Parameters page



The parameters page "IODD – Device configuration" shows all parameters which are provided by the IODD of the device. That means the parameter set is variable and depends on the connected IO-Link Device.

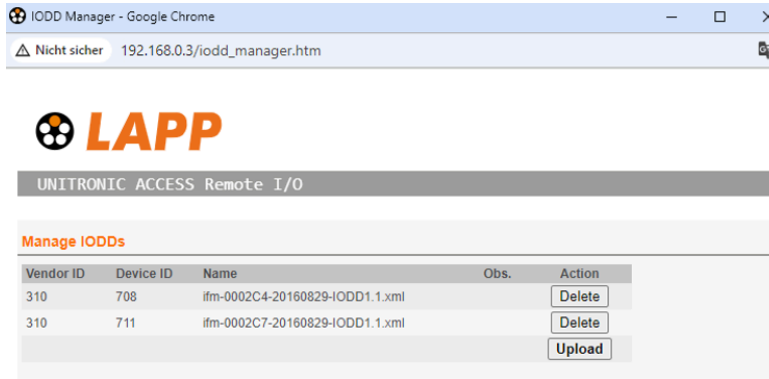
The stored IODD reads the parameter meta data, such as names, units, min/max values, descriptions etc. The values will be obtained directly from the connected device. For that reason it may take several seconds until the page is updated.

If not already saved into the browser, you will be asked for your credentials to continue. A valid user access with Web Interface group membership is needed in order to edit the device parameters. After the registration, enabled values can be changed. Disabled values cannot be changed and may be marked as "read-only" in the IODD. All values are directly written back to the device after any change.

Limitations

- Editing parameter values will directly change them inside the connected device. No parameter server action is triggered by that.
- There is a maximum size of the IODD in order to be uploaded into the system. This depends on several values, such as file size, parameter count, nesting levels etc.

13.2.3 IODD Management page



The IODD Management Page can be accessed via the System page displaying all IODDs that are currently stored in the system. All IODDs

matching connected devices are marked. On the IODD Management page, you can manually delete any IODD in the system.

14 Technical data

The following sections give an overview of the most important functional data needed to operate the device. For further information and detailed technical data, see the respective **Data Sheet** of your required product in the product specific download area on https://lapp.com.

14.1 General

Protection class (Only applies if the connectors are screwed together or if protective caps are used.)[2]	IP65 IP67 IP69K	
Ambient temperature (during operation and storage)	MP08...	-40 °C .. +70 °C (-40 °F .. +158 °F)
	PN08IOLA.../EIP08IOLA.../EC08IOLA.../ MTCP08IOLA.../CCL08IOLA...	-20 °C .. +60 °C (-4 °F .. +140 °F)
Weight	UNITRONIC® ACCESS 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 52: General information

2 Not under UL investigation.

14.2 Modbus TCP protocol

Protocol	Modbus TCP
Update cycle	1 ms
Transmission rate	100 Mbit/s, full duplex
Transmission procedure Autonegotiation	100BASE-TX supported
Product type	Modbus TCP server
Product code	MTCP08IOLA08DIO, 381166715
Supported Ethernet protocols	Ping ARP- HTTP / HTTPS TCP/IP DHCP/BOOTP
Switch functionality	Integrated
Modbus TCP interface Connections Autocrossing	2 M12 sockets, 4-pin, D-coded (see pin assignments) 2 M12 Hybrid male/female, 8-pin supported
Electrically isolated Ethernet ports -> FE	2000 V DC

Table 53: Modbus TCP protocol

14.3 Power supply of the module electronics/ sensors

Port X03, X04	M12-L-coded Power, connector/socket, 5-pole Pin 1 / Pin 3		
Nominal voltage U_S	24 V DC (SELV/PELV)		
Current U_S	Max. 16 A		
Voltage range	21 .. 30 V DC		
Power consumption of module electronics	Typically 160 mA (+/-20 % at U_S nominal voltage)		
Power supply interruption	Max. 10 ms		
Voltage ripple U_S	Max. 5 %		
Current consumption sensor system (L+ / Pin 1)	MP08...	Port X1 .. X8 (Pin 1)	max. 4 A per port (at $T_{ambient}$ = 30° C)
	PN08IOLA.../ EIP08IOLA.../ EC08IOLA.../ MTCP08IOLA.../ CCL08IOLA...	Port X1 .. X8 (Pin 1)	max. 2 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 54: Information on the power supply of the module electronics/ sensors

14.4 Power supply of the actuators

14.4.1 IO-Link Class A devices (U_L)

Nominal voltage U_L	24 V DC (SELV/PELV)
Voltage range	18 .. 30 V DC
Current U_L	Max. 16 A
Voltage ripple U_L	Max. 5 %
Reverse polarity protection	Yes
Operational indicator (U_L)	LED green: 18 V (+/- 1 V) < U_L LED red: U_L < 18 V (+/- 1 V) or U_L > 30 V (+/- 1 V) * if "Report U_L supply voltage fault" is enabled.
Port X03, X04	M12_L-coded Power, connector/socket, 5-pole Pin 2 / Pin 4

Table 55: Information on the power supply of the actuators

14.4.2 IO-Link Class A/B devices (U_{AUX})

Nominal voltage U_{AUX}	24 V DC (SELV/PELV)
Voltage range	18 .. 30 V DC
Current U_{AUX}	Max. 16 A
Voltage ripple U_{AUX}	Max. 5 %
Reverse polarity protection	Yes
Electric isolation $U_S \leftrightarrow U_{AUX}$	500 V
Operational indicator (U_{AUX})	LED green: 18 V (+/- 1 V) < U_{AUX} LED red: U_{AUX} < 18 V (+/- 1 V) or U_{AUX} > 30 V (+/- 1 V) * if "Report U_{AUX} supply voltage fault" is enabled.
Port X03, X04	M12_L-coded Power, connector/socket, 5-pole Pin 2 / Pin 4

Table 56: Information on the power supply of the actuators

14.5 I/O ports Channel A (Pin 4)

MP08...	Port X1 .. X8	Class A	IOL, DI, DO	M12 socket, 5-pin, Pin 4
PN08IOLA.../EIP08IOLA.../ EC08IOLA.../ MTCP08IOLA.../ CCL08IOLA...	Port X1 .. X8	Class A	IOL, DI, DO	

Table 57: IO-Link Master ports: Functional overview for Ch. A (Pin 4)

14.5.1 Configured as digital input, Ch. A (Pin 4)

Input connection	MP08... PN08IOLA.../EIP08IOLA.../EC08IOLA.../ MTCP08IOLA.../CCL08IOLA...		Type 1 as per IEC 61131-2
Nominal input voltage	24 V DC		
Input current	Typically 3 mA		
Channel type	Normally open, p-switching		
Number of digital inputs	MP08... PN08IOLA.../EIP08IOLA.../ EC08IOLA.../MTCP08IOLA.../ CCL08IOLA...	X1 .. X8	8
Status indicator	yellow LED		
Diagnostic indicator	red LED per channel		

Table 58: I/O ports Ch. A (Pin 4) configured as digital inputs

14.5.2 Configured as digital output, Ch. A (Pin 4)

Attention: For variant MP08IOLA08DIO, the digital outputs of Channel A is **supplied by the U_L power** when parameterized to "High-Side Switch" mode.

Attention: For variant PN08IOLA08DIO, the digital outputs of Channel A is **supplied by the U_S power**.

Output type	normally open, p-switching (parameterized to "High-Side Switch" mode)	
Nominal output voltage per channel Signal status “1” Signal status “0”	 min. (U_S -1 V) **or** min. (U_L -1 V) depending on the device variant max. 2 V	
Max. output current per device	MP08...	9 A (power supplied via U_L)
Max. output current per device	PN08IOLA.../EIP08IOLA.../ EC08IOLA.../MTCP08IOLA.../ CCL08IOLA...	4 A (power supplied via U_S)
Max. output current per channel[3]	MP08... (X1 .. X8)	2 A (power supplied via U_S)
	PN08IOLA.../EIP08IOLA.../ EC08IOLA.../MTCP08IOLA.../ CCL08IOLA... (X1 .. X8)	0.5 A (power supplied via U_S) 0.25 A for **UL applications**
Short-circuit/overload protected	yes/yes	
Behavior in case of short circuit or overload	deactivation with automatic power-on (parameterized)	
Number of digital outputs	MP08... (X1 .. X8) PN08IOLA.../EIP08IOLA.../ EC08IOLA.../MTCP08IOLA.../ CCL08IOLA... (X1 .. X8)	8
Status indicator	yellow LED per output	
Diagnostic indicator	red LED per channel	

Table 59: I/O ports Ch. A (Pin 4) configured as digital outputs

3 Max. 2.0 A per channel; for every port pair X1/X2, X3/X4, X5/X6, X7/X8 max. 6.5 A (for **UL applications** max. 5.0 A); for the whole port group X1 .. X8 max. 9.0 A in total (with derating).

14.5.3 Configured as IO-Link port in COM mode, Ch. A

IO-Link Master specification	v1.1.3, IEC 61131-9	
Communication rates	4.8 kbaud (COM 1) 38.4 kbaud (COM 2) 230.4 kbaud (COM 3)	
Line lengths in the IO-Link Device	max. 20 m	
Number of IO-Link ports	MP08... (X1 .. X8) PN08IOLA.../EIP08IOLA.../EC08IOLA.../ MTCP08IOLA.../CCL08IOLA... (X1 .. X8)	8
Min. IO-Link cycle time	400 µs	

Table 60: Configured as IO-Link port in COM mode

14.6 I/O ports Channel B (Pin 2)

MP08...	Port X1 .. X8	Class A	DI, DO	M12 socket, 5-pin, Pin 2
PN08IOLA.../EIP08IOLA.../ EC08IOLA.../ MTCP08IOLA.../ CCL08IOLA...	Port X1 .. X8	Class A	DI	

Table 61: IO-Link Master ports: Functional overview for Ch. B (Pin 2)

14.6.1 Configured as a digital input, Ch. B (Pin 2)

Input connection	MP08... PN08IOLA.../EIP08IOLA.../EC08IOLA.../ MTCP08IOLA.../CCL08IOLA...		Type 1 as per IEC 61131-2
Nominal input voltage	24 V DC		
Input current	Typically 3 mA		
Channel type	Normally open, p-switching		
Number of digital inputs	MP08...	X1 .. X8	8
Number of digital inputs	PN08IOLA.../EIP08IOLA.../ EC08IOLA.../MTCP08IOLA.../ CCL08IOLA...	X1 .. X8	8
Status indicator	white LED		
Diagnostic indicator	red LED per channel		

Table 62: I/O ports Ch. B (Pin 2) configured as digital inputs

14.6.2 Configured as a digital output, Ch. B (Pin 2)

Attention: For variant MP08IOLA08DIO, the digital outputs of Channel B is **supplied by the U_L power**.

Attention: For variant PN08IOLA08DIO, the digital outputs of Channel B is **supplied by the U_S power**.

Output type	normally open, p-switching	
Nominal output voltage per channel Signal status “1” Signal status “0”	min. (U_S -1 V) **or** min. (U_L -1 V) **or** min. (U_{AUX} -1 V) depending on the device variant max. 2 V	
Max. output current per device	MP08...	9 A (power supplied via U_L)
Max. output current per device	PN08IOLA.../ EIP08IOLA.../ EC08IOLA.../ MTCP08IOLA.../ CCL08IOLA...	4 A (power supplied via U_S)
Max. output current per channel [4]	MP08...	2 A (power supplied via U_S)
Max. output current per channel [4]	PN08IOLA.../ EIP08IOLA.../ EC08IOLA.../ MTCP08IOLA.../ CCL08IOLA...	0 A (no outputs on Ch. B)
Short-circuit/overload protected	yes/yes	
Behavior in case of short circuit or overload	deactivation with automatic power-on (parameterized)	
Number of digital outputs	MP08...	8
Number of digital outputs	PN08IOLA.../ EIP08IOLA.../ EC08IOLA.../ MTCP08IOLA.../ CCL08IOLA...	–
Status indicator	white LED per output	
Diagnostic indicator	red LED per channel	

Table 63: I/O ports Ch. B (Pin 2) configured as digital outputs

[4] For Class A devices: Max. 2.0 A per channel; for every port pair X1/X2, X3/X4, X5/X6, X7/X8 max. 6.5 A (for **UL applications** max. 5.0 A); for the whole port group X1 .. X8 max. 9.0 A in total (with derating).

14.7 LEDs

LED	Color	Description
U_L/U_{AUX}	Green	Auxiliary sensor/actuator voltage OK 18 V (+/- 1 V) < U_L/U_{AUX} < 30 V (+/- 1 V)
	Red*	Auxiliary sensor/actuator voltage LOW U_L/U_{AUX} < 18 V (+/- 1 V) or U_L/U_{AUX} > 30 V (+/- 1 V) * if "Report U_L/U_{AUX} supply voltage fault" is enabled.
	OFF	None of the above conditions.
U_S	Green	System/sensor voltage OK 18 V (+/- 1 V) < U_S < 30 V (+/- 1 V)
	Red	System/sensor voltage LOW U_S < 18 V (+/- 1 V) or U_S > 30 V (+/- 1 V)
	Red flashing	Device performs a factory reset (position of rotary encoding switches: 9-7-9)
	OFF	None of the above conditions.
X1 .. X8 A	Green	IO-Link COM Mode: IO-Link communication exists.
	Green flashing	IO-Link COM Mode: No IO-Link communication.
	Yellow	Standard-I/O Mode: Status of digital input or output on C/Q (pin 4) line "on".
	OFF	None of the above conditions
X1 .. X8 B	White	Status of digital input or digital output on pin 2 line "on".
	Red	Short circuit on pin 4 and pin 2 line. / All modes: Overload or short circuit on L+ (pin 1) line / communication error
	OFF	None of the above conditions.
P1 Lnk/Act P2 Lnk/Act	Green	Ethernet connection to another subscriber exists. Link detected.
	Yellow flashing	Data exchange with another subscriber.
	OFF	No connection to another subscriber. No link, no data exchange.

LED	Color	Description
BF	Red	Bus fault. No configuration, no or slow physical connection.
	Red flashing at 2 Hz	Link exists but no communication link to the Modbus TCP controller.
	OFF	Modbus TCP controller has established an active connection to the device.
DIA	Red	Modbus TCP module diagnostic alarm active.
	Red flashing at 1 Hz	Watchdog time-out; fail safe mode is active.
	Red flashing at 2 Hz, 3 sec	DCP signal service is initiated via the bus.
	Red double flash	Firmware update
	OFF	None of the above conditions.
MS	Green	Device is ready for operation.
	Green flashing	Device is ready but not configured yet.
	Red	Serious error that cannot be resolved.
	Red flashing	Minor error that can be resolved Example: An incorrect or contradictory configuration is classified as a minor error.
	Flashing alternately: Red / Green	The device is performing a self-test.
	OFF	The device is switched off.

LED	Color	Description
NS	Green	Connected: The device has at least one connection.
	Green flashing	No connection: The device has no connection. IP address exists.
	Red	Duplicate IP address: The device has detected that the assigned IP address is already being used by another device.
	Red flashing	Connection has exceeded time limit or connection interrupted.
	Flashing alternately: Red / Green	The device is performing a self-test.
	OFF	The device is switched off or has not been assigned an IP address.

Table 64: Information on the LED colors

14.8 Data transfer times

The following tables give an overview of the internal data transfer times of the UNITRONIC® ACCESS IO-Link Master with a connected IO-Link Device as digital I/O extension (LAPP article IOL08DIO08DIO 16DIO Hub with a minimum cycle time of 1 ms).

There are three measured data direction values for each use case:

- **PLC to DO:** Transfer of a changed PLC output data to IO-Link Device digital output.
- **DI to PLC:** Transfer of a changed digital input signal on IO-Link Device to PLC.
- **Round-trip time (RTT):** Transfer of a changed PLC output data to IO-Link Device digital output. The digital output is connected to an IO-Link Device digital input. Transfer of the changed digital input signal on IO-Link Device to PLC. RTT = [PLC to DO] + [DI to PLC].

The measured values are taken from the ethernet data transmission line. The values are therefore without PLC processing times and PLC cycle time.

The configurable digital input filter value on IOL08DIO08DIO was set to "off" (0 ms).

For calculation of user specific data transfer and round-trip times of possible input filters, PLC processing and cycles times must be taken into calculation.

The measured values are valid for a maximum of 48 bytes of IO-Link data for the IO-Link Master in each direction (Input/Output).

Use case 1:

IO-Link Master configuration with enabled Web interface and *disabled* IIoT protocols

Data direction	Data transfer time in ms		
	Minimum	Average	Maximum
PLC to DO	3.7	6.0	7.7
DI to PLC	1.1	3.0	4.3
RTT	6.1	8.9	11.1

Use case 2:

IO-Link Master configuration with enabled Web interface and *enabled* IIoT protocols

Data direction	Data transfer time in ms		
	Minimum	Average	Maximum
PLC to DO	7.7	10.0	13.4
DI to PLC	3.3	4.4	5.6
RTT	12.1	14.3	17.0

15 Accessories

In order to get access to various types of accessories, please visit our Web page:

https://www.lapp.com

Manual

CC-Link IE Field Basic

UNITRONIC ACCESS IO-Link Master

Multi-protocol:
MP08IOLA08DIO (8 × IO-Link Class A)

Single-protocol:
CCL08IOLA08DIO (8 × IO-Link Class A)



Contents

1 About this manual 8

1.1 General information 8
1.2 Explanation of symbols 9
1.2.1 Use of danger information 9
1.2.2 Use of general information 9
1.3 Version information 9

2 Safety instructions 10

2.1 Intended use 10
2.2 Qualified personnel 11

3 Designations and synonyms 12

4 System description 16

4.1 About UNITRONIC® ACCESS 16
4.2 Device variants 17
4.3 I/O port overview 19

5 Overview of product features 21

5.1 CC-Link IE Field Basic product features 21
5.2 I/O port features 22
5.3 Integrated Web server 23
5.4 Security features 24
5.5 Other features 25

6 Assembly and wiring 26

6.1 General information 26
6.2 Outer dimensions 27
6.2.1 UNITRONIC® ACCESS multi-protocol variants with CC-Link IE Field Basic 27
6.2.2 UNITRONIC® ACCESS single-protocol variants with CC-Link IE Field Basic 28
6.2.3 Notifications 29
6.3 Port assignments 30
6.3.1 Ethernet ports, M12 socket, 4-pin, D-coded 30
6.3.2 Power supply with M12 power L-coded 31
6.3.2.1 IO-Link Master with Class A ports 32
6.3.2.2 IO-Link Master with Class A/B ports 33
6.3.3 I/O ports as M12 sockets 33
6.3.3.1 IO-Link ports (Class A) 34

7 Starting operation 35

7.1 CSP+ file 35
7.2 MAC addresses 35
7.3 State on delivery 36
7.4 Setting the rotary encoding switches 37
7.4.1 CC-Link IE Field Basic 40
7.4.2 Factory reset 40
7.5 Setting network parameters 41

8 Configuration CC-Link IE Field Basic 42

8.1 General settings 43
8.1.1 Force mode lock 44
8.1.2 Web interface lock 44
8.1.3 Report U_L/U_{AUX} supply voltage fault 44

8.1.4 Report actuator fault without U_L/U_{AUX} voltage 44
8.1.5 Report U_S voltage fault 44
8.1.6 External configuration lock 44
8.2 Port configuration X1 .. X8 45
8.2.1 Port Mode 49
8.2.2 Validation and Backup 49
8.2.3 IQ Mode 53
8.2.4 Cycle Time 53
8.2.5 Vendor ID 54
8.2.6 Device ID 54
8.2.7 IOL Failsafe 54
8.2.8 IOL Failsafe values 55
8.2.9 Swapping Length 55
8.2.10 Swapping Offset 56
8.2.11 Swapping Count 56
8.2.12 Sensor Supply Disabled 56
8.2.13 Suppress all Diagnosis 56
8.2.14 DO Surveillance Timeout 56
8.2.15 DO Failsafe 56
8.2.16 DO Restart Mode 57
8.2.17 DI Logic 57
8.2.18 DI Filter 58
8.2.19 Error LED Disable 58
8.2.20 Level LED Disable 58
8.2.21 Use Push Pull 58
8.2.22 Current Limit 58
8.2.23 DI Latch 58
8.2.24 DI Extension 60

9 Process data assignment 62

9.1 Consuming data (output) 63
9.2 Producing data (input) 64

10 Diagnostics processing 65

10.1 Error of the system/sensor power supply 65
10.2 Error of the auxiliary/actuator power supply 66
10.3 Overload/short circuit of the digital outputs 66
10.4 Overload/short circuit of the actuator power supply P24 68
10.5 Overload/short-circuit of the I/O port sensor supply outputs 69

11 Configuration and operation with GxWorks3® 70

11.1 Integration of a CSP+ file 70
11.2 Network parameters 71
11.3 Parameter processing 73

12 IIoT functionality 76

12.1 MQTT 77
12.1.1 MQTT configuration 77
12.1.2 MQTT topics 80
12.1.2.1 Base topic 80
12.1.2.2 Publish topic 83
12.1.2.3 Command topic (MQTT Subscribe) 88
12.1.3 MQTT configuration - Quick start guide 92
12.1.3.1 MQTT configuration via JSON 92
12.2 OPC UA 94
12.2.1 OPC UA configuration 95
12.2.2 OPC UA address space 97
12.2.3 OPC UA configuration - Quick start guide 98
12.2.3.1 OPC UA configuration via JSON 98
12.3 REST API 100
12.3.1 Standard device information 102

12.3.2 Structure 103
12.3.3 Configuration and forcing 109
12.3.4 Reading and writing ISDU parameters 112
12.3.4.1 Reading ISDU 112
12.3.4.2 Writing ISDU 114
12.3.5 Upload and process an IODD file 116
12.3.6 Example: Reading ISDU 120
12.3.7 Example: Writing ISDU 120
12.4 CoAP server 121
12.4.1 CoAP configuration 121
12.4.2 REST API access via CoAP 122
12.4.3 CoAP configuration - Quick start guide 125
12.4.3.1 CoAP configuration via JSON 125
12.5 Syslog 127
12.5.1 Syslog configuration 127
12.5.2 Syslog configuration - Quick start guide 130
12.5.2.1 Syslog configuration via JSON 130
12.6 Network Time Protocol (NTP) 132
12.6.1 NTP configuration 132
12.6.2 NTP configuration - Quick start guide 134
12.6.2.1 NTP configuration via JSON 134

13 The integrated Web server 136

13.1 UNITRONIC® ACCESS MP08... variants 137
13.1.1 The Status page 137
13.1.2 The Ports page 138
13.1.2.1 IODD upload 138
13.1.3 The System page 140
13.1.3.1 License 141
13.1.3.2 Config upload/download 141
13.1.3.3 IODD 144
13.1.3.4 Restart device 145
13.1.3.5 Reset configuration to factory defaults 145
13.1.3.6 Firmware update 145

13.1.3.7 System diagnosis 146
13.1.4 The User page 147
13.2 UNITRONIC® ACCESS single-protocol PN0.../EIP0.../EC0.../MTCP0.../CCL0... variants 148
13.2.1 The System page 148

14 IODD 150

14.1 IO-Link Device parameters and ISDU requests 150
14.2 Web GUI functionality 151
14.2.1 Port Details page 151
14.2.2 Parameters page 153
14.2.3 IODD Management page 154

15 Technical data 156

15.1 General 157
15.2 CC-Link IE Field Basic protocol 158
15.3 Power supply of the module electronics/sensors 159
15.4 Power supply of the actuators 160
15.4.1 IO-Link Class A devices (U_L) 160
15.4.2 IO-Link Class A/B devices (U_{AUX}) 160
15.5 I/O ports Channel A (Pin 4) 161
15.5.1 Configured as digital input, Ch. A (Pin 4) 161
15.5.2 Configured as digital output, Ch. A (Pin 4) 162
15.5.3 Configured as IO-Link port in COM mode, Ch. A 164
15.6 I/O ports Channel B (Pin 2) 165
15.6.1 Configured as a digital input, Ch. B (Pin 2) 165
15.6.2 Configured as a digital output, Ch. B (Pin 2) 166
15.7 LEDs 168
15.8 Data transfer times 171

16 Accessories 173

1 About this manual

1.1 General information

Read the assembly and operating instructions on the following pages carefully before starting up the modules. Keep this information where it is accessible to all users.

The texts, figures, diagrams, and examples used in this document are exclusively used to explain how to operate and apply the modules.

Please contact us if you have any detailed questions on installing and starting up the devices.

U.I. Lapp GmbH

Schulze-Delitzsch-Straße 25
D-70565 Stuttgart
Germany

mailto:info@lapp.com
https://www.lapp.com

U.I. Lapp GmbH reserves the right to make technical changes or changes to this document 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.

1.3 Version information

Index	Created	Changes
1.0	10/2024	

Table 1: Overview of manual revisions

2 Safety instructions

2.1 Intended use

The products described in this manual are decentralized IO-Link Masters on an Industrial Ethernet Network.

We adhere to all safety standards when developing, producing, testing, and documenting our products. When you adhere to the handling specifications and safety instructions described for the configuration, assembly, and correct operation, there should not normally be any risks for people or equipment.

The modules fulfill the requirements of the EMC guidelines (2014/30/EU) and the low voltage guideline (2014/35/EU).

The IO-Link Masters are designed to be used in the industrial sector. The industrial environment is distinguished by the fact that the consumer is not connected directly to the public low voltage network. Additional measures are required for use in residential areas or in business and commercial sectors.

Attention: This equipment may cause radio interference in residential areas. In this case the operator may be requested to carry out appropriate measures.

The proper and safe operation of this product depends on proper transportation, storage, assembly, and installation, and careful operation.

A completely assembled device housing is required for the proper operation of the IO-Link Masters. Only connect devices that fulfill the requirements of EN 61558-2-4 and EN 61558-2-6 to the IO-Link Masters.

During the configuration, installation, start-up, maintenance, and testing of the devices, adhere to the safety and accident-prevention guidelines for the specific application.

Only install cables and accessories that fulfill the requirements and regulations for safety, electromagnetic compatibility, and, where applicable, telecommunication end devices, as well as the specification information.

Information on which cables and accessories are permitted for the installation can be obtained from U.I. Lapp GmbH or is contained in this manual.

2.2 Qualified personnel

The configuration, installation, start-up, maintenance, and testing of the devices may only be performed by a qualified electrician who is familiar with the safety standards of the automation technology.

The personnel requirements are based on the requirement profiles described by ZVEI, VDMA, or equivalent organizations.

Only electricians who are familiar with the content of all provided device documentation are authorized to install and maintain the devices described. These are persons who

- based on their technical training, knowledge, and experience, and their knowledge of the pertinent standards, can evaluate the work to be carried out and identify any potential risks or
- based on working for several years in a related sector, have the same level of knowledge as they would have from the relevant technical training.

Only U.I. Lapp GmbH is permitted to make changes to the hardware or software of the products that go beyond the scope of this manual.



Warning: Making unqualified changes to the hardware or software, or non-adherence to the warning information contained in this document, can result in serious personal injury or damage to equipment.



Attention: U.I. Lapp GmbH 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 and IO-Link)
BUI	Back-Up Inconsistency (EIP diagnostics)
CC	CC-Link IE Field
C/Q	I/O port pin 4 mode, IO-Link communication/switching signal
Ch. A	Channel A (Pin 4) of I/O port
Ch. B	Channel B (Pin 2) of I/O port
CIP	Common Industrial Protocol (media independent protocol)
CIP Safety™	Common Industrial Protocol for Safety applications, CIP Safety™ is a registered trademark of ODVA
Class A	IO-Link port specification (Class A)
Class B	IO-Link port specification (Class B)
CoAP	Constrained Application Protocol
CSP+	Control & Communication System Profile Plus
DAT	Device Acknowledgement Time
DCP	Discovery and Configuration Protocol
DevCom	Device Comunicating (EIP diagnostics)
DevErr	Device Error (EIP diagnostics)
DI	Digital Input
DIA	Diagnostic LED
DO	Digital Output
DIO	Digital Input/Output
DTO	Device Temperature Overrun (EIP diagnostics)
DTU	Device Temperature Underrun (EIP diagnostics)
DUT	Device under test

EIP	EtherNet/IP™ is a registered trademark of ODVA
ERP	Enterprise Resource Planning system
ETH	ETHERNET
FE	Functional Earth
FME	Force Mode Enabled (EIP diagnostics)
FS	Functional Safety
FSU	Fast Start-Up
GSDML	General Station Description Markup Language
High-B	High-Byte
HTTPS	Hyper Text Transfer Protocol Secure
ICE	IO-Link port COM Error (EIP diagnostics)
ICT	Invalid Cycle Time (EIP diagnostics)
IDE	IO-Link port Device Error (EIP diagnostics)
IDN	IO-Link port Device Notification (EIP diagnostics)
IDW	IO-Link port Device Warning (EIP diagnostics)
IIoT	Industrial Internet of Things
ILE	Input process data Length Error (EIP diagnostics)
IME	Internal Module Error (EIP diagnostics)
I/O	Input / Output
I/O port	X1 .. X8
I/O port pin 2	Channel B of I/O ports
I/O port pin 4 (C/Q)	Channel A of I/O ports
IODD	I/O Device Description
IOL or IO-L	IO-Link
I/Q	I/O port pin 2 mode, Digital Input/switching signal
ISDU	Indexed Service Data Unit
IVE	IO-Link port Validation Error (EIP diagnostics)
I&M	Identification & Maintenance
JSON	JavaScript Object Notation (platform independent data format)
L+	I/O port pin 1, sensor power supply

UNITRONIC® ACCESS 60	UNITRONIC® ACCESS variants with a width of 60mm
Little Endian	Data format with Low-B on first place (EtherNet/IP)
LLDP	Link Layer Discovery Protocol
Low-B	Low-Byte
LSB	Least Significant Bit
LVA	Low Voltage Actuator Supply (EIP diagnostics)
LVS	Low Voltage System/Sensor Supply (EIP diagnostics)
MIB	Management Information Base
MP	Multi-protocol: PROFINET + EtherNet/IP + EtherCAT® + Modbus TCP (+ CC-Link IE Field Basic)
MQTT	Message Queuing Telemetry Transport (open networking protocol)
MSB	Most Significant Bit
M12	Metric thread according to DIN 13-1 with 12 mm diameter
NTP	Network Time Protocol
OFDT	One Fault Delay Time
OLE	Output process data Length Error (EIP diagnostics)
OPC UA	Open Platform Communications Unified Architecture (platform independent, service-oriented architecture)
PFH	Probability of dangerous Failure per Hour [h -1]
PD	Process Data
PDCT	Port and Device Configuration Tool
PLC	Programmable Logic Controller
PN	PROFINET
PWR	Power
Qualifier	Validity on a process value. Valid = "1"
REST	REpresentational State Transfer
RFC	Request for Comments
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)
SFRT	Safety Function Response Time
SIO mode	Standard Input Output mode
SLMP	Seamless Message Protocol
SNMP	Simple Network Management Protocol
SP	Single Protocol (PROFINET, EtherNet/IP, EtherCAT®, Modbus TCP or CC-Link IE Field Basic)
SPE	Startup Parameterization Error (EIP diagnostics)
T-B	Test Channel B
T-A	Test Channel A
U_{AUX}	$U_{Auxiliary}$, supply voltage for the load circuit (Actuator supply on Class B ports of Class A/B IO-Link Master)
UDP	User Datagram Protocol
UDT	User-Defined Data Types
UINT8	Byte in PLC (IB, QB)
UINT16	Unsigned integer with 16 bits or word in PLC (IW, QW)
U_L	U_{Load}, supply voltage for the load circuit (Actuator supply on Class A IO-Link-Master)
UL	Underwriters Laboratories Inc. (certification company)
UTC	Coordinated Universal Time (Temps Universel Coordonné)
WCDT	Worst Case Delay Time

Table 2: Designations and synonyms

4 System description

The Lapp U.I. UNITRONIC® ACCESS modules function as the interface in an industrial Ethernet system: A central controller on the management level is able to communicate with the decentralized sensors and actuators on the field level. The line or ring topologies for which UNITRONIC® modules can be used ensure not only reliable data communication but also significantly reduce the number of cables required and thus also the costs for installation and maintenance. They additionally enable easy and quick extension.

4.1 About UNITRONIC® ACCESS

The UNITRONIC® ACCESS device variants convert standard input, standard output or IO-Link signals from sensors & actuators into an industrial Ethernet protocol (PROFINET, EtherNet/IP, EtherCAT®, Modbus TCP, CC-Link IE Field Basic) and/or into a cloud protocol (REST API, CoAP, OPC UA, MQTT). For the first time, there is now Syslog on board. The robust 8 port housing design allows the use even in harsh environments where e.g. weld field immunity, high temperature ranges or protection class IP67 & IP69K are needed.

4.2 Device variants

The following variants are available in the UNITRONIC® ACCESS Multi-protocol and the UNITRONIC® ACCESS single-protocol family:

Article number	Product designation	Description	I/O port functionality
381166717	MP08IOLA08DIO	UNITRONIC® ACCESS M12-60 mm, IO-Link Master Multi-protocol (PN, EIP, EC, MB, CC) Security	8 x IO-Link Class A
381166712	PN08IOLA08DIO	UNITRONIC® ACCESS single-protocol M12-60 mm, IO-Link Master PROFINET	8 x IO-Link Class A
381166713	EIP08IOLA08DIO	UNITRONIC® ACCESS single-protocol M12-60 mm, IO-Link Master EtherNet/IP	8 x IO-Link Class A
381166714	EC08IOLA08DIO	UNITRONIC® ACCESS single-protocol M12-60 mm, IO-Link Master EtherCAT®	8 x IO-Link Class A
381166715	MTCP08IOLA08DIO	UNITRONIC® ACCESS single-protocol M12-60 mm, IO-Link Master Modbus TCP	8 x IO-Link Class A
381166716	CCL08IOLA08DIO	UNITRONIC® ACCESS single-protocol M12-60 mm, IO-Link Master CC-Link IE Field Basic	8 x IO-Link Class A

Table 3: Overview of UNITRONIC® ACCESS Multi-protocol and UNITRONIC® ACCESS single-protocol variants

4.3 I/O port overview

The following tables show the main I/O port differences of the UNITRONIC® ACCESS IO-Link-Master family. Pin 4 and Pin 2 of the I/O ports can be configured partly to IO-Link, Digital Input or Digital Output.

UNITRONIC® ACCESS Class A IO-Link ports

Device variant	Port	Pin 1 U_S	Pin 4 / Ch. A (C/Q)				Pin 2 / Ch. B (I/Q)	
MP08...	**Info:**	–	Class A	Type 1	Supply by U_S [1)]	Supply by U_L [2)]	Type 1	Supply by U_L [2)]
	X8:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)
	X7:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)
	X6:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)
	X5:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)
	X4:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)
	X3:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)
	X2:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)
	X1:	Out (4 A)	IOL	DI	DO (0.5 A)	DO (2 A)	DI	DO (2 A)

Table 4: Port configuration of MP08... variants

[1)] DO switch mode configured as "Push-Pull" (description in the configuration chapters).

[2)] DO switch mode configured as "High-Side" (description in the configuration chapters).

UNITRONIC® ACCESS single-protocol Class A IO-Link ports

Device variant	Port	Pin 1 U_S	Pin 4 / Ch. A (C/Q)			Pin 2 / Ch. B (I/Q)
PN08IOLA.../ EIP08IOLA.../ EC08IOLA.../ MTCP08IOLA.../ CCL08IOLA...	**Info:**	–	Class A	Type 1	Supply by U_S [1)]	Type 1
	X8:	Out (2 A)	IOL	DI	DO (0.5 A*)	DI
	X7:	Out (2 A)	IOL	DI	DO (0.5 A*)	DI
	X6:	Out (2 A)	IOL	DI	DO (0.5 A*)	DI
	X5:	Out (2 A)	IOL	DI	DO (0.5 A*)	DI
	X4:	Out (2 A)	IOL	DI	DO (0.5 A*)	DI
	X3:	Out (2 A)	IOL	DI	DO (0.5 A*)	DI
	X2:	Out (2 A)	IOL	DI	DO (0.5 A*)	DI
	X1:	Out (2 A)	IOL	DI	DO (0.5 A*)	DI

Table 5: Port configuration of PN08IOLA.../EIP08IOLA.../EC08IOLA.../ MTCP08IOLA.../CCL08IOLA... variants

[1)] With DO Switch Mode configured as "Push-Pull" (see description in the configuration chapters).

* For **UL applications**: Max. 0.25 A DO.

5 Overview of product features

5.1 CC-Link IE Field Basic product features

CC-Link IE Field Basic network

- Number of stations: 4 .
- RX – 64 bits (per station)
- RY – 64 bits (per station)
- RWw – 32 words (per station)
- RWr – 32 words (per station)

Data connection

The connection option provided by UNITRONIC® ACCESS is the widely-used M12 connector with D-coding for the CC-Link IE Field Basic network.

The connectors are also color-coded to prevent the ports from being mixed up.

Data transmission rates

Featuring a transmission rate of up to 100 MBit/s, the CC-Link IE Field Basic devices can handle both fast transmission of I/O data and transmission of larger volumes of data.

Diagnostic data

The devices support diagnosis flags and extended diagnostic data that can be appended to the I/O data.

5.2 I/O port features

IO-Link specification.

UNITRONIC® ACCESS is ready for IO-Link specification v1.1.3.

8 x IO-Link Master ports

Depending on the device variant, the device has 4 IO-Link Class A ports, 4 IO-Link Class A ports and 4 IO-Link Class B ports, or 8 IO-Link Class A ports with an additional digital input and optional output on pin 2 of the I/O port. For detailed information see chapter I/O port overview on page 19.



Warning: If modules with electric isolation and modules without electric isolation are used within the same system, the electric isolation of all connected modules is annulled.

IO-Link port connections

The IO-Link port connection option provided by the module series is the 5-pin M12 connector.

Validation & Backup

The Validation & Backup function checks if the right device is connected and stores / monitors the parameters of the IO-Link Device. The function thus gives you an easy option for replacing the IO-Link Device.

This is possible as of IO-Link specification V1.1 and only if the IO-Link Device **and** the IO-Link Master support the function.

LED

You can see the status of a port by the color of the matching LEDs and their flash pattern. For details on the meanings of the LED colors, please see section LEDs on page 168.

5.3 Integrated Web server

Network parameter display

Get an overview of network parameters such as the IP address, subnet mask and gateway.

Displaying diagnostics

View diagnostics via the integrated Web server.

User management

Use the integrated Web server for convenient management of all users.

IO-Link Device parameters

Reading and writing of IO-Link Device parameters is supported. The system command Store parameters is needed after parameter writing, to take over the changed parameter into the IO-Link Master backup memory when enabled.

5.4 Security features

Firmware signature

The official firmware update packages contain a signature which helps prevent the system against manipulated firmware updates.

Syslog

The UNITRONIC® ACCESS multi-protocol variants support the traceability of messages centrally managed and logged via Syslog.

User manager

The Web server provides a user manager to help protect the Web interface against unauthorized access. You can manage the users by groups with different access levels “Admin” or “Write”.

Default user settings:

User: admin

Password: private

Attention: Change the default settings to help protect the device against unauthorized access.

5.5 Other features

Interface protection

The devices have reverse polarity, short-circuit and overload protection for each interface.

For more details, see section Port assignments on page 30.

Failsafe

The devices support a failsafe function. This allows you to define the behavior of every single channel configured as an output in the case of invalid PLC data (e.g. PLC in STOP) or of lost PLC communication.

Industrial Internet of Things

UNITRONIC® ACCESS 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 UNITRONIC® ACCESS 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 UNITRONIC® ACCESS multi-protocol variants with CC-Link IE Field Basic

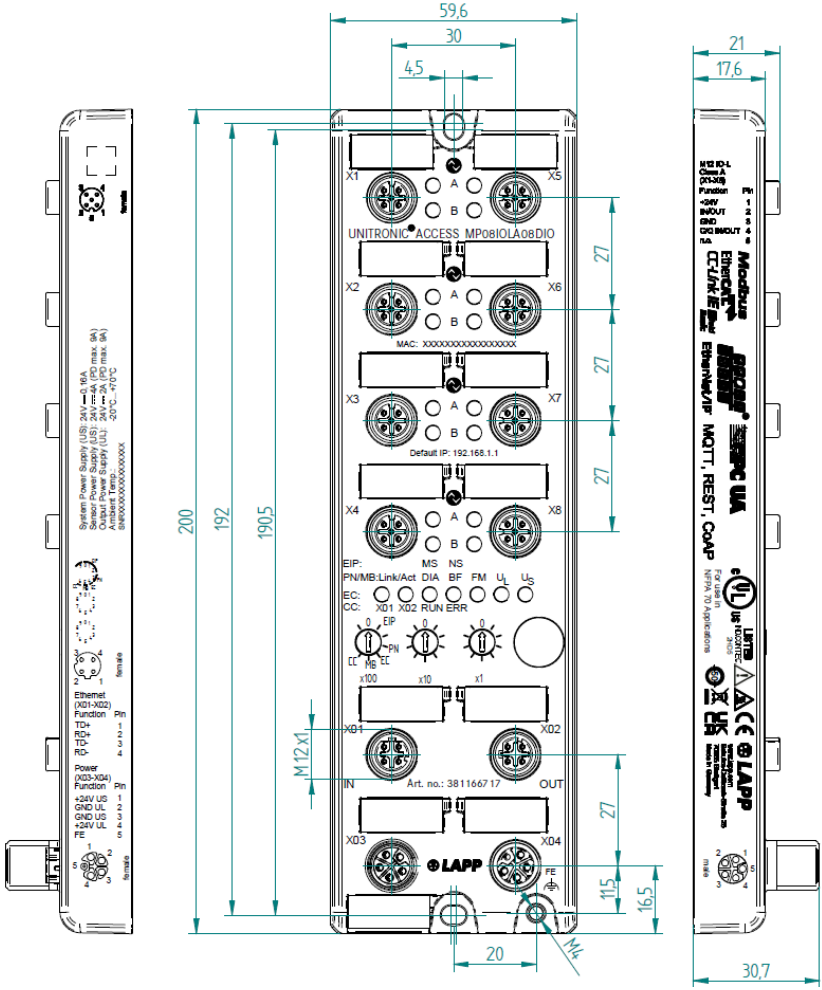


Figure 1: MP08IOLA08DIO

6.2.2 UNITRONIC® ACCESS single-protocol variants with CC-Link IE Field Basic

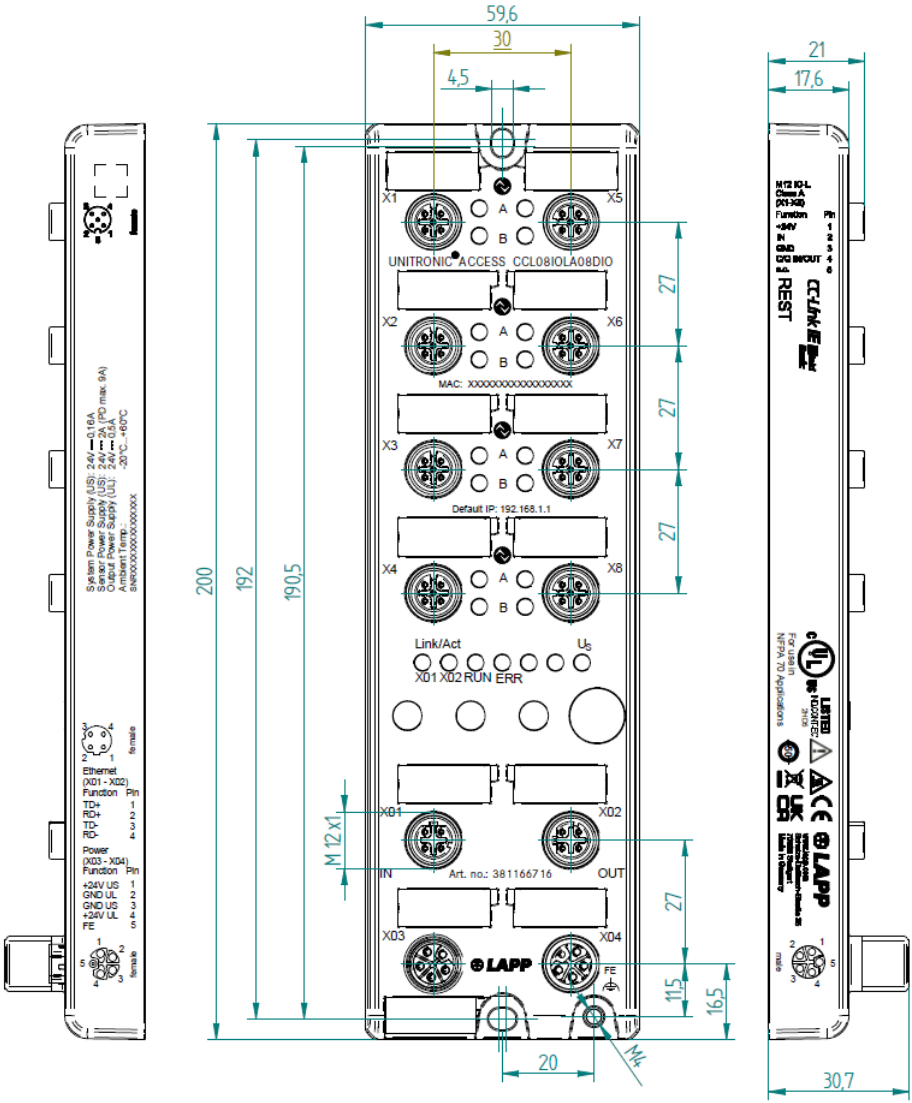


Figure 2: CCL08IOLA08DIO

6.2.3 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 device variants.

Warning: Observe the following maximum output power for the sensor supply of Class A devices:

Max. 4.0 A per port; for **UL applications** max. 5 A for every port pair X1/X2, X3/X4, X5/X6, X7/X8; max. 9.0 A in total (with derating) for the whole port group X1 .. X8.

6.3 Port assignments

All the contact arrangements shown in this chapter show the frontal view of the connection area for the connectors.

6.3.1 Ethernet ports, M12 socket, 4-pin, D-coded

Color coding: green

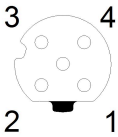


Figure 3: Schematic drawing, ports X01, X02

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 6: 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

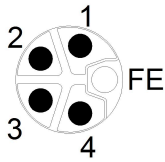


Figure 4: Schematic diagram of the M12 L-coding (connector X03 for Power In)

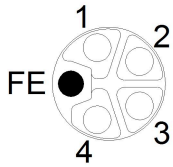


Figure 5: Schematic diagram of the M12 L-coding (socket X04 for Power Out)

6.3.2.1 IO-Link Master with Class A ports

Power supply	Pin	Signal	Function
	1	U_S (+24 V)	Sensor/system power supply
	2	GND_U_L	Ground/reference potential U_L
	3	GND_U_S	Ground/reference potential U_S[1]
	4	U_L (+24 V)	Load supply (**NOT** electrically isolated to U_S internally in device)
	5	FE	Functional ground

Table 7: Power supply with M12-Power Class A

Attention: Only use power supply units for the system/sensor and actuator supply that correspond to PELV (Protective Extra Low Voltage) or SELV (Safety Extra Low Voltage). Power supplies according to EN 61558-2-6 (transformers) or EN 60950-1 (switching power supply units) fulfill these requirements.

[1] U_L and U_S ground connected in device

6.3.2.2 IO-Link Master with Class A/B ports

Power supply	Pin	Signal	Function
Mixed IO-Link (Class A/B) I/O ports	1	U_S (+24 V)	Sensor/system power supply
	2	GND_U_{AUX}	Ground/reference potential U_{AUX} (electrically **isolated** to GND_U_S internally in device)
	3	GND_U_S	Ground/reference potential U_S
	4	U_{AUX} (+24 V)	Auxiliary supply (electrically **isolated** to U_S internally in device)
	5	FE	Functional ground

Table 8: Power supply with M12-Power Class A/B

Attention: Only use power supply units for the system/sensor and actuator supply that correspond to PELV (Protective Extra Low Voltage) or SELV (Safety Extra Low Voltage). Power supplies according to EN 61558-2-6 (transformers) or EN 60950-1 (switching power supply units) fulfill these requirements.

6.3.3 I/O ports as M12 sockets

Color coding: black

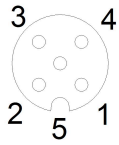


Figure 6: Schematic drawing I/O port as M12 socket IO-Link

6.3.3.1 IO-Link ports (Class A)

MP08...	Pin	Signal	Function
IO-Link Class A, ports X1 .. X8	1	+24 V	power supply +24 V
	2	IN/OUT	Ch. B: Digital input or digital output
	3	GND	Ground/reference potential
	4	C/Q	Ch. A: IO-Link data communication, digital input or digital output
	5	n.c.	not connected

PN08IOLA.../ EIP08IOLA.../ EC08IOLA.../ MTCP08IOLA.../ CCL08IOLA...	Pin	Signal	Function
IO-Link Class A, ports X1 .. X8	1	+24 V	power supply +24 V
	2	IN	Ch. B: Digital input
	3	GND	Ground/reference potential
	4	C/Q	Ch. A: IO-Link data communication, digital input or digital output
	5	n.c.	not connected

Table 9: I/O ports as IO-Link Class A

Used signal names compared to the IO-Link specification conventions:

Pin	UNITRONIC® ACCESS	IO-Link specification	Comment
1	+24 V	L+	Supplied by U_S
2	IN/OUT	I/Q	
3	GND	L-	
4	C/Q IN/OUT	C/Q	

7 Starting operation

7.1 CSP+ file

A CSP+ file describes the information of a CC-Link device and is required to configure the UNITRONIC® ACCESS multi-protocol and UNITRONIC® ACCESS single-protocol variants in an engineering tool. Each device variant requires its own CSP+ file. The file can be downloaded from the product pages on our online catalog: https://lapp.com

On request, the CSP+ file is also sent by the support team.

The CSP+ file and the associated icons are grouped together in an archive file named "0x0670_UNITRONIC ACCESS MP08IOLA08DIO_1.0_en.cspp.zip / 0x0670_UNITRONIC ACCESS CCL08IOLA08DIO_1.0_en.cspp.zip".

"MP08IOLA08DIO" is the model number of the UNITRONIC® ACCESS variant.

Download this file and install it as described in chapter Configuration and operation with GxWorks3 on page 70.

Install the CSP+ file for the respective device variant by using the hardware or network configuration tool of your controller manufacturer.

In GxWorks®, install the files with the CSP+ Hardware Installation Tool.

The UNITRONIC® ACCESS multi-protocol and UNITRONIC® ACCESS single-protocol variants are then available in the hardware catalog as Communications Adapter.

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.

7.3 State on delivery

CC-Link IE Field Basic parameters in state on delivery or after a factory reset:

Network mode:	Static
Static IP address:	192.168.3.XXX (XXX = rotary switch position or last stored data)
Subnet mask:	255.255.255.0
Gateway address	192.168.3.100
Device designations:	MP08IOLA08DIO CCL08IOLA08DIO
Product type:	CC-Link IE Field Basic Slave Station

7.4 Setting the rotary encoding switches

The following UNITRONIC® ACCESS IO-Link Master variants additionally provide the protocol CC-Link IE Field Basic (CC):

- MP08IOLA08DIO

Caution: Risk of device damage due to corrupt device memory

Any interruption of the power supply to the device during and after protocol selection can lead to a corrupt device memory.

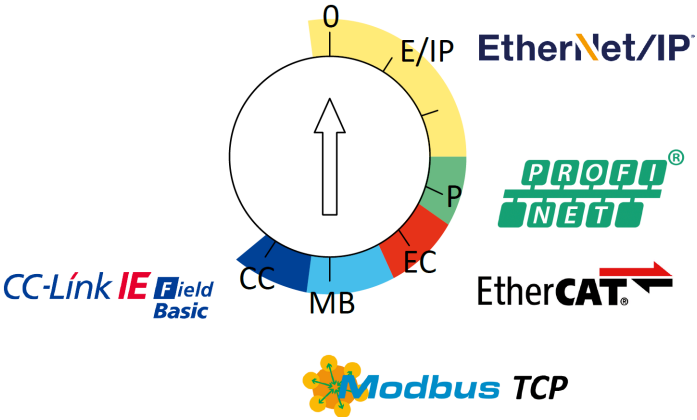
After selecting a protocol followed by a restart of the device, the new protocol is initialized. This can take up to 15 seconds. During this time the device is not usable and the LED indicators are out of function. When the protocol change is complete, the LED indicators return to normal operation and the device can be used again.

- Make sure that the power supply is maintained during the entire process.

The UNITRONIC® ACCESS multi-protocol variants allow you to select different protocols for communication within an industrial Ethernet system. In this way the IO-Link Masters with multi-protocol function can be integrated into different networks without it being necessary to purchase products specific for each protocol. This technology also gives you the option to use the same IOL-Master in different environments.

Using rotary encoding switches at the lower front of the devices, you can easily and conveniently set both the protocol and the address of the device, if the protocol to be used supports this. Once you have made a protocol selection and started the cyclical communication, the device stores this setting permanently and uses the selected protocol from this point on. To use another supported protocol with this device, perform a factory reset.

The multi-protocol devices have a total of three rotary encoding switches. With the first rotary encoding switch (x100) you set the protocol by using the corresponding switch position. Additionally, x100 is used to set the third last digit of the IP address for EIP.



With the other rotary encoding switches (x10 / x1), you set the last two digits of the IP address when you are using EtherNet/IP, Modbus TCP or CC-Link IE Field Basic.

Protocol	x100	x10	x1
EtherNet/IP	0-2	0-9	0-9
PROFINET	P	–	–
EtherCAT®	EC	–	–
Modbus TCP	MB	0-9	0-9
CC-Link IE Field	CC	0-9	0-9

Table 10: 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 40.

If you position the rotary encoding switch in a manner that is invalid, the device signals this to you with a blink code (the RUN LED blinks continuously).

7.4.1 CC-Link IE Field Basic

If you decide to use CC-Link IE Field Basic as a protocol, use the first rotary encoding switch to select the protocol. The second rotary encoding switch (x10) can be used to configure the 10 position of the last octet of the IP address, and the third rotary encoding switch (x1) allows you to configure the 1 position. Values between 0 and 9 can be selected for the second and third switches. The first three octets of the IP address are set by default to 192.168.3.

For example, the rotary encoding switch setting 6(x100), 1(x10) and 0(x1) gives you an IP address of 192.168.3.10 for CC-Link IE Field Basic. It is only possible to assign IP addresses between 192.168.3.1 and 192.168.3.99 for CC-Link IE Field Basic via the rotary switches.

Rotary switch setting	Function
600 (network parameters already saved)	The network parameters last saved are used (IP address, subnet mask, gateway address).
600 ... 699	The last 2 digits of the saved or preset IP address are overwritten by the setting of the rotary switch.
979	The device performs a reset to the factory settings. The network parameters are also reset to the default values. Communication is not possible in this operation mode.

Table 11: Setting options of the rotary encoding switches for CC-Link IE Field Basic

7.4.2 Factory reset

A factory reset restores the original factory settings and thus resets the changes and settings you have made up to that point. It also resets the protocol selection. To perform a factory reset, set the first rotary encoding switch (x100) to 9, the second (x10) to 7, and the third (x1) also to 9.

Afterwards perform a power cycle and wait 10 seconds due to internal memory write processes.

During the factory reset, the U_S LED lights up red. As soon as the factory settings have been restored, the U_S LED is flashing green.

	x100	x10	x1
Factory Reset	9	7	9

Follow the steps from section Setting the rotary encoding switches again to select a new protocol.

For performing a factory reset via software configuration, see chapter OPC UA configuration on page 95 and the configuration section.

7.5 Setting network parameters

Use the two right-hand rotary switches (x10 and x1) on the front of the device to set the last octet of the static IP address. Each rotary switch in the range of CC-Link IE Field Basic is assigned to one decimal digit, so that you can configure a number between **0 – 99**. During startup, the position of the rotary switches is typically read within one time cycle.

The complete IP address, the subnet mask, the gateway address and the network mode can be configured and stored via the Web server or any other available configuration interfaces. New configuration interfaces can only be applied to after a restart of the device.

For additional information, see chapter Setting the rotary encoding switches.

8 Configuration CC-Link IE Field Basic

Parameters of the UNITRONIC® ACCESS device can be configured via SNMP, the Web server or IIoT protocols. Acyclic messages over SNMP are sent to read and write the configuration. When sending, all existing parameters will be overwritten by this data and the content of the SNMP messages has the highest valence.

To avoid parameter overwriting by the Web server or IIoT protocols during operation, some lock parameters can be enabled in the PLC configuration respectively in the configuration assembly.

The following chapters represent different setting groups with its configuration parameters. The default values are highlighted.

8.1 General settings

Setting	Description	Default value
Suppress U_{Aux} Diagnosis Mode	Report U_L/U_{Aux} supply voltage fault 0 = Report U_L/U_{Aux} supply voltage fault enabled 1 = Report U_L/U_{Aux} supply voltage fault disabled 2 = Auto	0
Suppress Actuator Diagnosis without U_L	Report actuator fault without U_L/U_{Aux} voltage 0 = Report actuator fault without U_L/U_{Aux} voltage enabled 1 = Report actuator fault without U_L/U_{Aux} voltage disabled	0
Suppress U_S Diagnosis	Report U_S voltage fault 0 = Diagnosis disabled 1 = Diagnosis enabled	0
Reserved	Reserved	0
Output Auto Restart	Output auto restart 0 = Output auto restart disabled 1 = Output auto restart enabled	0
Web Interface Lock	Web interface lock 0 = Web interface lock disabled 1 = Web interface lock enabled	0
Forcing Lock	Force mode lock 0 = Force lock disabled 1 = Force lock enabled	0
External Configuration Lock	External configuration lock 0 = External configuration lock disabled 1 = External configuration lock enabled	1

8.1.1 Force mode lock

The input and output process data can be forced via different interfaces (e.g. Web interface, REST, OPC UA, MQTT). The support of interfaces depends on the available software features. If the *Force mode lock* is enabled, it is no longer possible to force input and output process data through these interfaces.

Danger: Risk of physical injury or death! Unattended forcing can lead to unexpected signals and uncontrolled machine movements.

8.1.2 Web interface lock

The Web interface access can be configured. If *Web interface lock* is enabled, the Web pages are no longer reachable.

8.1.3 Report U_L/U_{AUX} supply voltage fault

During commissioning, it is possible that no power supply is connected to the U_L/U_{AUX} pins. Therefore it can be helpful to suppress and disable the *U_L/U_{AUX} supply voltage fault* diagnosis.

8.1.4 Report actuator fault without U_L/U_{AUX} voltage

During commissioning, it is possible that no power supply is connected to the U_L/U_{AUX} pins. Therefore it can be helpful to suppress and disable the *Report actuator fault without U_L/U_{AUX} voltage* diagnosis.

8.1.5 Report U_S voltage fault

During commissioning, it is possible that no power supply is connected to the U_S pins. Therefore it can be helpful to suppress and disable the *Report U_S voltage fault* diagnosis.

8.1.6 External configuration lock

Configuration parameters can be set via different alternative interfaces (e.g. Web interface, REST, OPC UA, MQTT). An external configuration can only be done, if no cyclic PLC connection is active. Every new PLC configuration overwrites the external configuration settings.

8.2 Port configuration X1 .. X8

Setting	Description	Default value
Port Mode	Port Mode 0: Deactivated 1: IO-Link Manual 2: IO-Link Auto 3: Digital Input 4: Digital Output	3
Validation Check	Validation Option 0: No device check and clear (no data storage) 1: Type compatible V1.0 device (no data storage) 2: Type compatible V1.1 device (no data storage) 3: Type compatible V1.1 device with Backup & Restore (download + upload) 4: Type compatible V1.1 device with Restore (download master to device)	0
IQ (Pin 2) Mode	IQ Mode 0: Deactivated 1: Digital Input 2: Digital Output 5: Aux* *exclusively applicable for Class A/B Mixmodules	1
Cycle Time	Cycle Time 0: As fast as possible 1: 1.6 ms 2: 3.2 ms 3: 4.8 ms 4: 8.0 ms 5: 20.8 ms 6: 40.0 ms 7: 80.0 ms 8: 120.0 ms	0
Vendor ID	Vendor ID 0 .. 65535 ("0")	0
Device ID	Device ID 0 .. 16777215 ("0")	0

Setting	Description	Default value
IO-Link Failsafe Mode	Failsafe Mode 0: Set Low 1: Set High 2: Hold Last 3: Replacement Value 4: IO-Link Master Command	0
IO-Link Failsafe Value 0..31	IOL Failsafe replacement values Between 0 .. 255	0
Swap Length Consuming	Swap Length (Consuming data) 0: DWORD 1: WORD	0
Offset Consuming	Swap Offset (Consuming data) 0 .. 30 Byte	0
Swap Count Consuming	Swap Count (Consuming data) 0 .. 30 Byte	0
Swap Length Producing	Swap Length (Producing data) 0: DWORD 1: WORD	0
Offset Producing	Swap Offset (Producing data) 0 .. 30 Byte	0
Swap Count Producing	Swap Count (Producing data) 0 .. 30 Byte	0
Sensor Supply Disabled	Sensor Supply Disabled 0: Supply electric voltage to sensor 1: Do not supply electric voltage to sensor	0
Suppress all Diagnosis	Suppress all Diagnosis 0: Generate diagnosis on this channel 1: Do not generate any diagnosis on this channel	0
Surv. Timeout (Pin 2)	DO Surveillance Timeout for Pin 2 (IQ) Valid values: 0 .. 255	80
Surv. Timeout (Pin 4)	DO Surveillance Timeout for Pin 4 (CQ) Valid values: 0 .. 255	80

Setting	Description	Default value
Failsafe Mode SIO (Pin 2)	DO Failsafe for Pin 2 (IQ) 0: Set Low 1: Set High 2: Hold Last	0
Failsafe Mode SIO (Pin 4)	DO Failsafe for Pin 4 (CQ) 0: Set Low 1: Set High 2: Hold Last	0
DI Filter (Pin 2)	DI Filter for Pin 2 (IQ) 0: Disabled 1: 10 ms 2: 20 ms 3: 30 ms 4: 60 ms 5: 100 ms 6: 150 ms	0
DI Filter (Pin 4)	DI Filter for Pin 4 (CQ) 0: Disabled 1: 10 ms 2: 20 ms 3: 30 ms 4: 60 ms 5: 100 ms 6: 150 ms	0
DI Logic (Pin 2)	DI Logic for Pin 2 (IQ) 0: Normally Open 1: Normally Close	0
DI Logic (Pin 4)	DI Logic for Pin 4 (CQ) 0: Normally Open 1: Normally Close	0
DO Restart (Pin 2)	DO Restart for Pin 2 (IQ) 0: Disable 1: Enable	0
DO Restart (Pin 4)	DO Restart for Pin 4 (CQ) 0: Disable 1: Enable	0

Setting	Description	Default value
Error LED Disable (Pin 2)	Disable Pin 2 Error LED 0: Enable LED on channel B 1: Disable LED on channel B	0
Error LED Disable (Pin 4)	Disable Pin 4 Error LED 0: Enable LED on channel A 1: Disable LED on channel A	0
Level LED Disable (Pin 2)	Disable Pin 2 Level LED 0: Enable LED on channel B 1: Disable LED on channel B	0
Level LED Disable (Pin 4)	Disable Pin 4 Level LED 0: Enable LED on channel A 1: Disable LED on channel A	0
Use Push-Pull (Pin 4)	Use Push-Pull for Pin 4 0: Use High-Side switches 1: Use Push-Pull	0
Current limit (Pin 2)	Pin 2 current limit (maximum current limit till Pin 2 is turned off) 0 .. 65535	65535
Current limit (Pin 4)	Pin 4 current limit (maximum current limit till Pin 4 is turned off) 0 .. 65535	65535
DI Latch (Pin 4)	Enable Input Latch for Pin 4 0: Disable 1: Enable	0
DI Latch (Pin 2)	Enable Input Latch for Pin 2 0: Disable 1: Enable	0
DI Extension (Pin 4)	Set Input Extension for Pin 4 0 .. 255 ms	0
DI Extension (Pin 2)	Set Input Extension for Pin 2 0 .. 255 ms	0

8.2.1 Port Mode

The *Port Mode* describes how the IO-Link Master handles the presence of an IO-Link device at the port.

Deactivated:

The IO-Link port is deactivated but can be configured for later use. No diagnostics are generated if the IO-Link device is not connected.

IO-Link Manual:

The IO-Link port is activated and explicit port configuration can be done for the parameters *Validation and Backup* (Inspection Level), *Vendor ID*, *Device ID* and *Cycle Time*.

IO-Link Auto:

The IO-Link port is activated and no explicit port configuration is needed. Configurations such as *Validation and Backup* (Inspection Level), *Vendor ID*, *Device ID* and *Cycle Time* are not required.

Digital Input:

In this mode, the channel operates as digital input. The channel state can be seen in the *Digital Input Channel* status of the cyclic process data.

Digital Output:

In this mode, the channel operates as digital output. The channel can be controlled by the *Digital Output Channel Control* (first two bytes of the output data) or by the *IO-Link Output Data* (first byte of each IO-Link device output data) of the cyclic process data. This depends on the *Digital Output Control* parameter in the general settings.

8.2.2 Validation and Backup

With this parameter, the user can set the behavior of the IO-Link ports regarding the type compatibility and data storage mechanism of the connected IO-Link Device.

The precondition for using *Validation and Backup* is that you configure the *Port Mode* to "IO-Link Manual".

The IO-Link Master has a backup memory which can be used for storing the device parameters and for restoring them on the device. This backup memory can be deleted by the following events:

- IO-Link Master factory reset
- *Channel Mode* reconfiguration, e.g. from “Digital-Input” to “IO-Link”
- *Validation and Backup* reconfiguration, e.g. from “No device check” to “Type compatible V1.1 device with Backup & Restore”

For further information refer to the ‘IO-Link Interface and System Specification’ version 1.1.3 which can be downloaded from https://io-link.com/

No device check (no data storage):

No check of connected Vendor ID or Device ID and no "Backup and Restore" support of the IO-Link Master parameter server.

Type compatible V1.0 device (no data storage):

Type compatible according IO-Link specification V1.0 which includes validation of Vendor ID and Device ID. The IO-Link specification V1.0 does not support IO-Link Master parameter server.

Type compatible V1.1 device (no data storage):

Type compatible according IO-Link specification V1.1 which includes validation of Vendor ID and Device ID. "Backup and Restore" is disabled.

Type compatible V1.1 device with Backup + Restore

(upload + download):

Type compatible according IO-Link specification V1.1 which includes validation of Vendor ID and Device ID. "Backup and Restore" is enabled.

Pay attention to the following explanations regarding *Backup and Restore* conditions:

- Backup (Device to Master):

  A Backup (upload from IOL-Device to IOL-Master) is performed when an IO-Link Device is connected and the Master does not have any valid parameter data. The read parameter data are permanently stored on the IO-Link Master.

  An upload will also be performed, when the IO-Link Device has set the DS_UPLOAD_FLAG (Data Storage Upload Flag). This IOL-Device flag can be set in two ways:

  - Parameters written to IOL-Device in *Block Parameter* mode: An IO-Link Device sets the DS_UPLOAD_FLAG self-dependent, if the parameters were written in block parameter mode to the IO-Link Device with the last system command ParamDownloadStore (e.g. by a third party USB IO-Link Master for commissioning).
  - Parameters written to IOL-Device in *Single Parameter* mode: If single parameter data is changed on the IOL-Device during runtime, the stored device parameters on the IOL-Master can be updated using the ParamDownloadStore (index 0x0002, subindex 0x00, value 0x05) command. This command sets the DS_UPLOAD_REQ flag on the IOL-Device and thus the IO-Link Master executes an upload procedure from the IO-Link Device.

- Restore (Master to Device):

  A Restore (download from IOL-Master to IOL-Device) is performed when an IO-Link Device is connected and the IO-Link Master has valid parameter data stored which are usable for the IOL-Device and not equal compared to the device parameters.

  The restore procedure can be blocked by the IO-Link Device via the *Device Access Locks* parameter when supported by the IO-Link Device (Index 0x000C, refer to vendor specific IO-Link Device documentation).

Type compatible V1.1 device with Restore (download Master to Device):

Type compatible according IO-Link specification V1.1 which includes validation of Vendor ID and Device ID. Only "Restore" is enabled.

Pay attention to the following explanations regarding *Restore* conditions:

- Restore (Download / IOL-Master to IOL-Device):

  A Restore (download from IOL-Master to IOL-Device) is performed when an IO-Link Device is connected and the IO-Link Master has valid parameter data stored which are usable for the IOL-Device and not equal compared to the device parameters.

  In the *Restore* mode no change of the IOL-Device parameters will be stored permanently on the IOL-Master. When the IOL-Device sets the DS_UPLOAD_FLAG in this mode, the device parameters will be restored by the IOL-Master.

  The restore procedure can be blocked by the IO-Link Device via the *Device Access Locks* parameter when supported by the IO-Link Device (Index 0x000C, refer to vendor specific IO-Link Device documentation).

8.2.3 IQ Mode

The operating mode of Pin 2 (Channel B) of the respective IO-Link channel can be configured via this parameter.

Digital Output:

In this mode, the channel operates as digital output. The channel can be controlled by the *Digital Output Channel Control* (first two bytes of the output data) or by the *IO-Link Output Data* (first byte of each IO-Link device output data) of the cyclic process data. This depends on the *Digital Output Control* parameter in the general settings.

Digital Input:

In this mode, the channel operates as digital input. The channel state can be seen in the *Digital Input Channel* status of the cyclic process data.

AUX:

This option is exclusively applicable for Class A/B Mixmodules.

In this mode, Pin 2 of the IO-Link port acts as an auxiliary voltage output. The auxiliary voltage is fed by the U_{AUX} supply input. The auxiliary voltage output cannot be controlled.

8.2.4 Cycle Time

The IO-Link cycle time can be configured by this parameter.

The precondition for using *Cycle Time* is that you configure *Port Mode* to "IO-Link Manual".

As fast as possible:

The IO-Link port uses the max. supported IO-Link Device and Master update cycle time for the cyclic I/O data update between IO-Link Master and IO-Link Device.

1.6 ms, 3.2 ms, 4.8 ms, 8.0 ms, 20.8 ms, 40.0 ms, 80.0 ms, 120.0 ms:

The cycle time can be set manually to the provided options. This option can be used e.g. for IO-Link devices which are connected over inductive couplers. Inductive couplers are normally the bottleneck in the update cycle

time between IO-Link Master and IO-Link Device. In this case, please refer to the data sheet of the inductive coupler.

8.2.5 Vendor ID

The *Vendor ID* is needed for the validation of the IO-Link device and can be configured with this parameter.

Precondition for using the *Vendor ID* is that you configure *Port Mode* to "IO-Link Manual". *Validation and Backup* must be set to a type compatible V1.X device.

8.2.6 Device ID

The *Device ID* is needed for the validation of the IO-Link device and can be configured with this parameter.

Precondition for using the *Device ID* is that you configure *Port Mode* to "IO-Link Manual". *Validation and Backup* must be set to a type compatible V1.X device.

8.2.7 IOL Failsafe

The UNITRONIC® ACCESS devices support a failsafe function for the output data of the IO-Link channels. In case of an internal device error, the PLC is in STOP state and cannot provide valid process data, the connection is interrupted or the communication is lost: The output data of the IO-Link channels is controlled by the configured failsafe values.

Set Low:

If failsafe is active, all bits of the IO-Link output data are set to low ("0").

Set High:

If failsafe is active, all bits of the IO-Link output data are set to high ("1").

Hold Last:

If failsafe is active, all bits of the IO-Link output data arel hold the last valid process data state ("0" or "1").

Replacement Value:

A replacement value can be set via the *IO-Link Failsafe* parameter object for every IO-Link device. If failsafe is active, these replacement values are transmitted to the IO-Link device. Take into account that in the case of an error the replacement values are sent instead of the output process data so that a configured *Swapping Mode* has influence on the byte order.

IO-Link Master Command:

If failsafe is active, an IO-Link-specific mechanism for valid/invalid output process data is used and the IO-Link device determines the behavior itself.

8.2.8 IOL Failsafe values

IOL failsafe values represent byte-wise 32 replacement values. If failsafe is active, these values are transmitted to the IO-Link Device.

8.2.9 Swapping Length

The byte order of IO-Link is big endian. For setting output data in the correct format, the parameters *Swapping Mode* and *Swapping Offset* support the user. There can be selected up to 16 words or up to 8 double words for converting the output data.

Raw IO-Link Data:

No byte swap

Data type DWORD:

Data byte order: Byte 1, Byte 2, Byte 3, Byte 4

Order after Swap: Byte 4, Byte 3, Byte 2, Byte 1

Data type WORD:

Data byte order: Byte 1, Byte 2

Order after Swap: Byte 2, Byte 1

8.2.10 Swapping Offset

The *Swapping Offset* describes the start point in the process data for using the configured *Swapping Length*. Both parameters are dependent on the configured input or output data size.

8.2.11 Swapping Count

The *Swapping Count* describes the number of bytes in the process data to be swapped using the configured *Swapping Length*.

8.2.12 Sensor Supply Disabled

The supply of sensors can be disabled when *Sensor Supply Disabled* is set. The IO-Link Master will disable the supply for the respective IO-Link port.

8.2.13 Suppress all Diagnosis

By default, the IO-Link Master generates all possible diagnostics and will send reports via cyclic and cyclic data. All diagnostics can be suppressed by setting *Suppress all Diagnosis*.

8.2.14 DO Surveillance Timeout

The digital output channels are monitored during runtime. The error states are detected and reported as a diagnosis. To avoid error states during the switching of output channels, the surveillance timeout can be configured as a delay with deactivated monitoring.

The delay time begins with a rising edge of the output control bit. After delay time has elapsed, the output is monitored and error states are reported by diagnosis. When the channel is permanently switched on or off, the typical filter value (not changeable) is 5 ms.

8.2.15 DO Failsafe

The UNITRONIC® ACCESS devices support a failsafe function for the channels used as digital outputs. In case of an internal device error, the PLC is in STOP state and cannot provide valid process data. The connection is interrupted or the communication is lost. The outputs are controlled according to the configured failsafe values.

Set Low:

If failsafe is active, the physical output pin of the channel is set to low ("0").

Set High:

If failsafe is active, the physical output pin of the channel is set to high ("1").

Hold Last:

If failsafe is active, the physical output pin of the channel holds the last valid process data state ("0" or "1").

8.2.16 DO Restart Mode

In case of a short circuit or overload at an output channel, a diagnosis is reported and the output is switched to "off".

If *DO Restart Mode* is disabled, the output channel is not automatically turned on again. It can be turned on after a logical reset of the process output data of the channel.

If *DO Restart Mode* for this channel is enabled, the output will automatically be turned on again after a fix time delay for checking if the overload or short circuit condition is still active. When it is active, the channel is switched off again.

8.2.17 DI Logic

The logical state of an input channel can be configured via these parameters. If a channel is set to "Normally Open", a low signal ("0") is transferred to the process input data (e.g. if a non-damped sensor has an open switching output).

If a channel is set to "Normally Close", a high signal ("0") is transferred to the process input data (e.g. if a non-damped sensor has a closed switching output).

The channel LED shows, independent of these settings, the physical input state of the port pin.

8.2.18 DI Filter

A filter time for every digital input channel can be configured by these parameters. When there is no need for a filter it can be disabled.

8.2.19 Error LED Disable

Every channel of the ports X1 .. X8 has an error LED. The error LED can be disabled by enabling the parameter *Error LED Disable*. When this parameter is enabled, the LED status will not be "ON" in the case of an error on the port.

8.2.20 Level LED Disable

Every channel of the ports X1 .. X8 has a level LED. The level LED can be disabled by enabling the parameter *Level LED Disable*. When this parameter is enabled, the LED status will not be "ON" in case the input or output are high.

8.2.21 Use Push Pull

If *Use Push Pull* is enabled, the output will be switched active to high or low. In low state, the output can be a current sink. The digital output is supplied by U_S with a maximum current of 0.5 A.

When this option is not enabled, it will use option "High-Side switch" and the current limit is set according to the parameter *Current limit*. This option is not available for the Channel B of any port.

8.2.22 Current Limit

With this parameter you can configure the current limitations for the digital outputs. You can choose between different current limit options.

In low state, the output has a high impedance. The digital output is supplied by U_L or U_{AUX}, depending on the device variant, and has a selectable current limit. This means that the output is turned off and the actuator channel error diagnosis is reported when this limit is exceeded. If you set the level to *2.0 A max.*, the current limitation is not active and the maximum output current is available.

8.2.23 DI Latch

With this parameter enabled, a rising edge at digital input is held high in the input status data (latched) as this is acknowledged by the PLC.

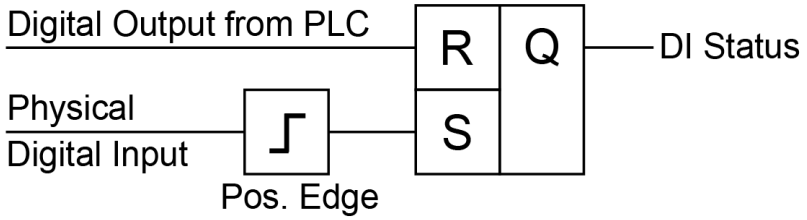


Figure 7: Input latch

R	S	Q
0	0	x (hold last)
0	1	1
1	0	0
1	1	0

Table 12: Truth table for input latch

When the input latch is enabled for a specific channel:

- The corresponding physical input of the channel is logically connected to the set input of a latch via an edge detector.
- The corresponding PLC output control bit (consuming data of I/O device) is logically connected to the reset input of the latch.
- The latch output is connected to the PLC input status data (producing data of the I/O device) of the corresponding channel.
- The latch will work according to the truth table above.
- It is not possible to directly read the physical input status data of that input channel, as the latch output is mapped to the PLC input status data.

The behavior in detail:

- A rising edge on the digital input will trigger the latch and set the latch output to '1'.
- The output will remain '1' until it is reset by the PLC program.
- A logical '1' on the corresponding PLC output control bit for that channel will reset the latch and set the latch output to '0', regardless of the set input or physical input state.

- If the input logic is inverted in the channel configuration, the inverted input logic is connected to the latch. Therefore, it will trigger on the falling edge regarding to a physical input signal.
- If the input is already high during enabling the latch, the latch will set to '1' (Q).

These DI Latch settings only work for channels that have been set to 'digital input mode'. It is recommended to always reset the latch before usage.

Default: Disabled

8.2.24 DI Extension

This parameter extends the duration of the digital input status after a state change at the physical input, when the input state change is faster than the extension time set.

The extension time will be applied on 'high' to 'low' and 'low' to 'high' input transitions. This setting only works for channels that have been set to 'digital input'.

Example:

The DI extension parameter is set to 16 ms, the physical input signal has low status => a high signal is detected for 8 ms.

In this case, the DI channel reports a high-status signal for 16 ms, regardless of other physical input signal transitions during this time.

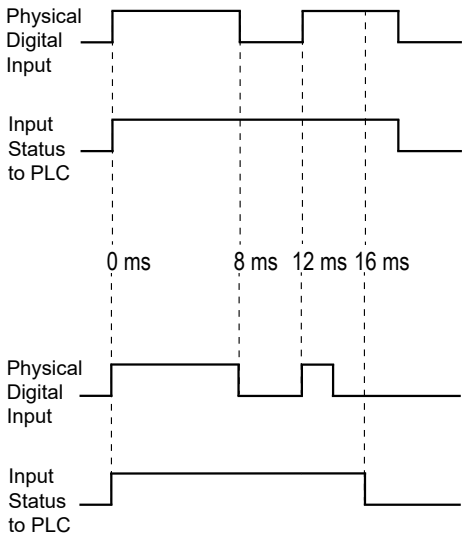


Figure 8: DI Extension

Available values: Off; 8 ms; 16 ms; 64 ms

Default: Off

9 Process data assignment

The UNITRONIC® ACCESS devices in general support process data communication in both directions. The consuming data in this context is defined as the process output data which controls physical outputs and IO-Link output data. The producing data in this context is defined as the process input data which contains the physical inputs, diagnostics and IO-Link input data with optional extended status and event data.

The following chapters describe the data images for the consuming and producing data direction which are assigned to the output and input assemblies.

9.1 Consuming data (output)

Attention: Depending on the engineering tool used, the numbering of the registers is displayed as octal, decimal or hexadecimal.

Port No.	Pin	Register for DO	Register for IO-Link	Access
X1	4	Y0	RWw00 – RWw0F	RW ("Read/Write")
	2	Y1	–	RW
X2	4	Y2	RWw10 – RWw1F	RW
	2	Y3	–	RW
X3	4	Y4	RWw20 – RWw2F	RW
	2	Y5	–	RW
X4	4	Y6	RWw30 – RWw3F	RW
	2	Y7	–	RW
X5	4	Y8	RWw40 – RWw4F	RW
	2	Y9	–	RW
X6	4	YA	RWw50 – RWw5F	RW
	2	YB	–	RW
X7	4	YC	RWw60 – RWw6F	RW
	2	YD	–	RW
X8	4	YE	RWw70 – RWw7F	RW
	2	YF	–	RW

Register for DO = single bit

Register for IO-Link = WORD

9.2 Producing data (input)

Attention: Depending on the engineering tool used, the numbering of the registers is displayed as octal, decimal or hexadecimal.

Port No.	Pin	Register for DI	Register for IO-Link	Access
X1	4	X0	RWr00 – RWr0F	R ("Read Only")
	2	X1	–	R
X2	4	X2	RWr10 – RWr1F	R
	2	X3	–	R
X3	4	X4	RWr20 – RWr2F	R
	2	X5	–	R
X4	4	X6	RWr30 – RWr3F	R
	2	X7	–	R
X5	4	X8	RWr40 – RWr4F	R
	2	X9	–	R
X6	4	XA	RWr50 – RWr5F	R
	2	XB	–	R
X7	4	XC	RWr60 – RWr6F	R
	2	XD	–	R
X8	4	XE	RWr70 – RWr7F	R
	2	XF	–	R

Register for DI = single bit

Register for IO-Link = WORD

10 Diagnostics processing

Port No.	Register for Diagnosis	Description	Access
X1	X20	X1 IO-Link data valid	R ("Read only")
X2	X21	X2 IO-Link data valid	R
X3	X22	X3 IO-Link data valid	R
X4	X23	X4 IO-Link data valid	R
X5	X24	X5 IO-Link data valid	R
X6	X25	X6 IO-Link data valid	R
X7	X26	X7 IO-Link data valid	R
X8	X27	X8 IO-Link data valid	R

Sr. No.	Register for Diagnosis	Description	Access
1	X38	U_S supply present	R ("Read only")
2	X39	U_S supply Fault	R
3	X3A	U_L supply present	R
4	X3B	U_L supply Fault	R
5	X3C	Internal module error	R
6	X3D	Force mode diagnosis	R

10.1 Error of the system/sensor power supply

The voltage value for the incoming system/sensor power supply is also monitored globally. If the voltage drops below approx. 18 V, or exceeds approx. 30 V, an error diagnosis is generated. The IO-Link specification requires at least 20 V at the L+ (pin1) output supply of the I/O ports. At least 21 V of U_S supply voltage for the IO-Link Master are required to minimize the risk of internal voltage drops in the IO-Link Master.

The green U_S indicator is off.

The error diagnosis has no effect on the outputs.

Caution: It must definitely be ensured that the supply voltage, measured at the most remote participant is not below 21 V DC from the perspective of the system power supply.

10.2 Error of the auxiliary/actuator power supply

The voltage value for the incoming auxiliary/actuator power supply is also monitored globally. If *Report U_L/U_{Aux} Supply Voltage Fault* is enabled, an error message is generated when the voltage drops below approx. 18 V or exceeds approx. 30 V. The U_L/U_{Aux} indicator shows red.

If output channels are set to *High State* and *Report DO Fault without U_L/U_{Aux}*, additional error diagnostics, caused by the voltage failure, are generated on the channels.

If *Report U_L/U_{Aux} Supply Voltage Fault* is disabled, no U_L/U_{Aux} or channel diagnostics appear.

10.3 Overload/short circuit of the digital outputs

In case of an overload or a short circuit of an output channel, the following channel-specific diagnostics are generated in the producing data image.

Port No.	Pin	Register for Diagnosis	Description	Access
X1	4	X10	Short circuit X1 Channel A	R ("Read only")
	2	X11	Short circuit X1 Channel B	R
X2	4	X12	Short circuit X2 Channel A	R
	2	X13	Short circuit X2 Channel B	R
X3	4	X14	Short circuit X3 Channel A	R
	2	X15	Short circuit X3 Channel B	R
X4	4	X16	Short circuit X4 Channel A	R
	2	X17	Short circuit X4 Channel B	R
X5	4	X18	Short circuit X5 Channel A	R
	2	X19	Short circuit X5 Channel B	R
X6	4	X1A	Short circuit X6 Channel A	R
	2	X1B	Short circuit X6 Channel B	R
X7	4	X1C	Short circuit X7 Channel A	R
	2	X1D	Short circuit X7 Channel B	R
X8	4	X1E	Short circuit X8 Channel A	R
	2	X1F	Short circuit X8 Channel B	R

A channel error is determined by comparing the target value set of a controller to the physical value of an output channel.

When an output channel is activated (rising edge of the channel state), the channel errors are filtered for the period that is set by the "Surveillance-Timeout" parameter via the configuration of the device. The value of this parameter can range from 0 to 255 ms; the default setting is 80 ms.

The filter is used to avoid premature error messages during the activation of a capacitive load, the deactivation of an inductive load or during any other voltage peak when changing a status.

10.4 Overload/short circuit of the actuator power supply P24

In case of an overload or a short circuit of the actuator power supply P24 (Class B) on the ports (X5 .. X8), the following channel-specific diagnostics in the producing data image are generated.

Port No.	Register for Diagnosis	Description	Access
X1	X30	reserved	R ("Read only")
X2	X31	reserved	R
X3	X32	reserved	R
X4	X33	reserved	R
X5	X34	X5 actuator supply P24 short circuit	R
X6	X35	X6 actuator supply P24 short circuit	R
X7	X36	X7 actuator supply P24 short circuit	R
X8	X37	X8 actuator supply P24 short circuit	R

10.5 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 channel-specific diagnostics in the producing data image are generated.

Port No.	Register for Diagnosis	Description	Access
X1	X28	X1 sensor short circuit	R ("Read only")
X2	X29	X2 sensor short circuit	R
X3	X2A	X3 sensor short circuit	R
X4	X2B	X4 sensor short circuit	R
X5	X2C	X5 sensor short circuit	R
X6	X2D	X6 sensor short circuit	R
X7	X2E	X7 sensor short circuit	R
X8	X2F	X8 sensor short circuit	R

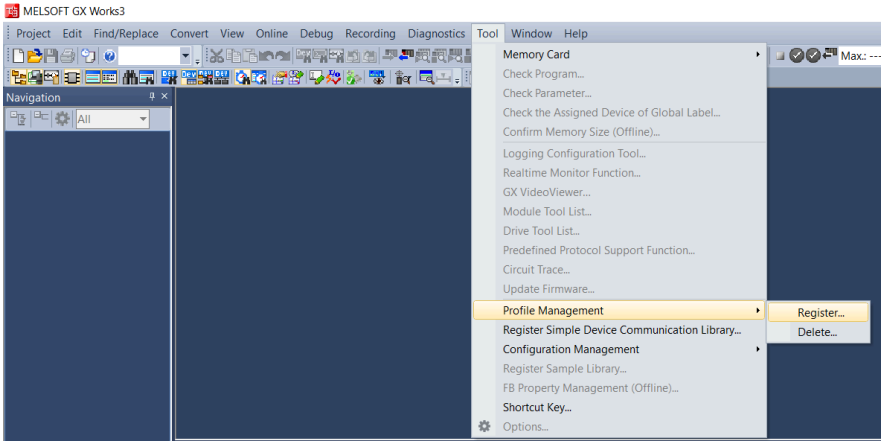
11 Configuration and operation with GxWorks3®

The configuration and start-up of UNITRONIC® ACCESS devices described in this chapter refers to the Mitsubishi Engineering Tool GxWorks®, V2. If you are using an engineering tool from another provider, please consider the related documentation.

11.1 Integration of a CSP+ file

Perform the following work steps to integrate a CSP+ file in GxWorks3®:

1. Open GxWorks3® and navigate to **Tool** > **Profile Management** > **Register**.



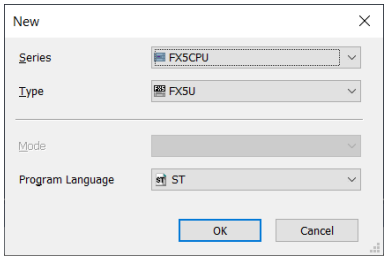
2. Select *0x0670_UNITRONIC ACCESS ...08DIO_1.0_en.cspp.zip* and the CSP+ file will be registered.

11.2 Network parameters

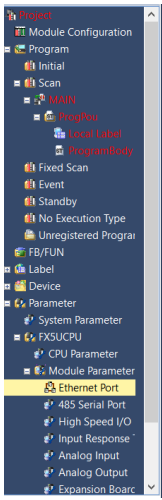
Perform the following work steps to change the Network parameters:

1. Open GxWorks3® and create a new project.

2. Select the series and the type of the used PLC.

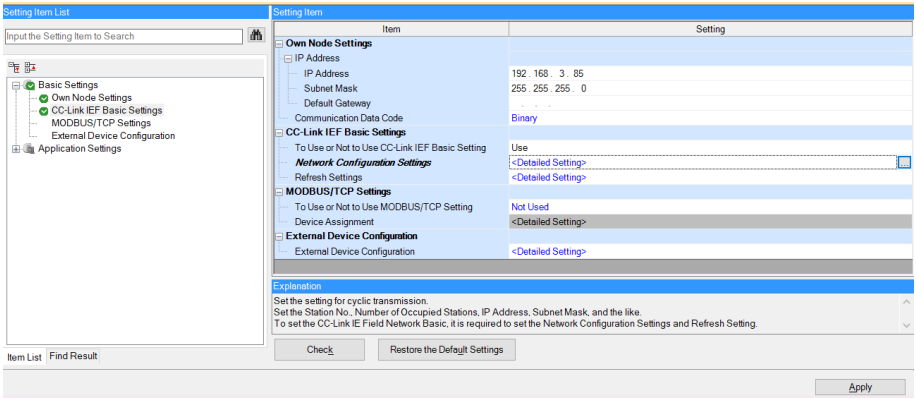


3. To open the setting window, navigate to **Project** > **Parameter** > **"the selected CPU module"** > **Module Parameter**



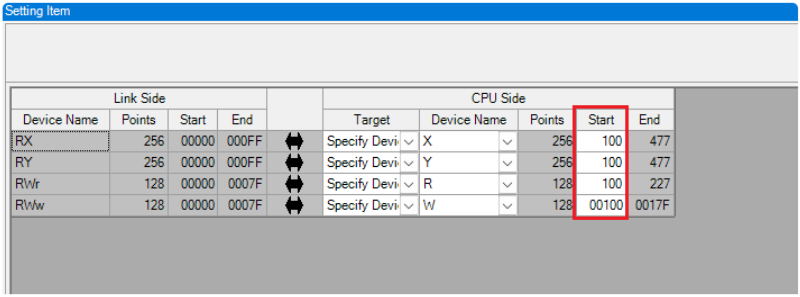
In the appearing window, the CC-Link IE Field Basic Master station can be configured.

4. Navigate to *Own Node Settings* to configure the PLC or Master station.



5. Under **CC-Link IEF Basic settings** > **To Use or Not to Use CC-Link IEF Basic Setting** select "Use".

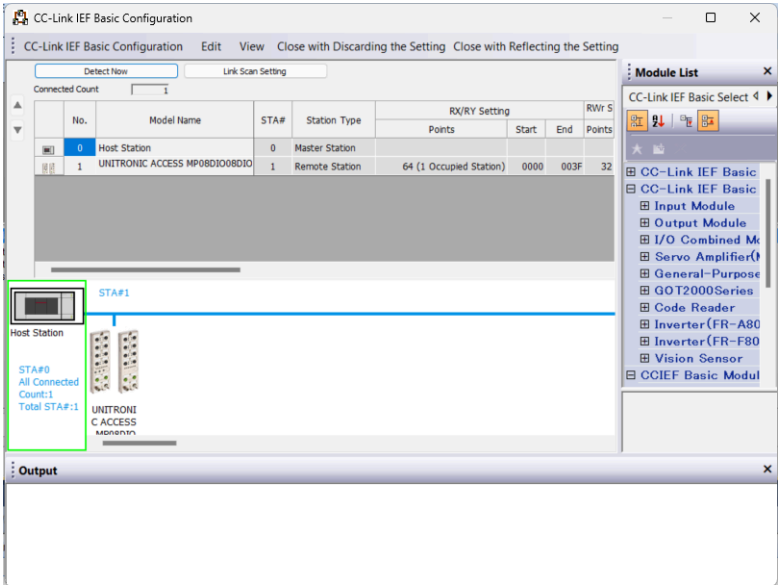
- The option *Network Configuration Settings* allows you to configure a CC-Link IE Field Basic Master, connected stations, a Network, parameters and many more.
- Settings under *Refresh Settings* are necessary for the automatic data transfer between Link side and CPU side:



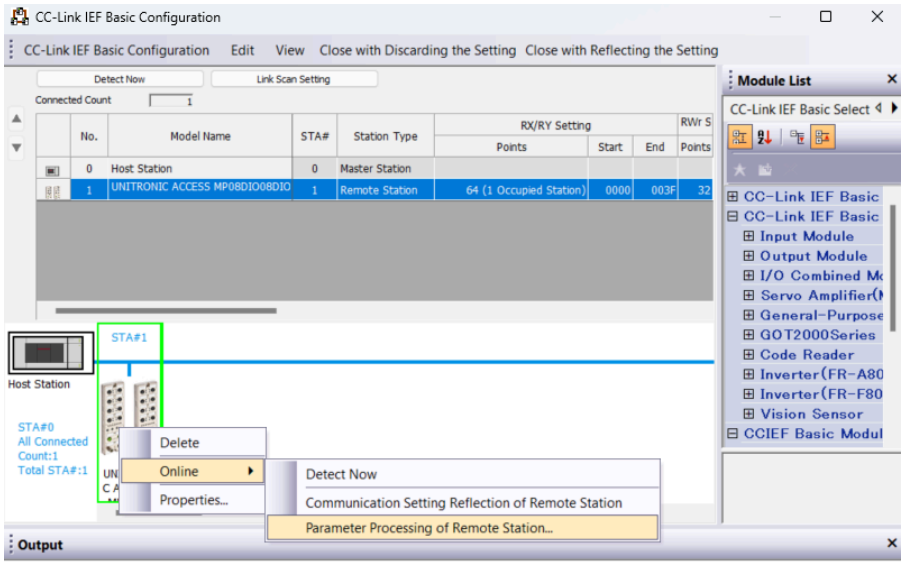
11.3 Parameter processing

Under *Network Configuration Settings*, individual stations can be configured. Perform the following work steps to configure a UNITRONIC® ACCESS device:

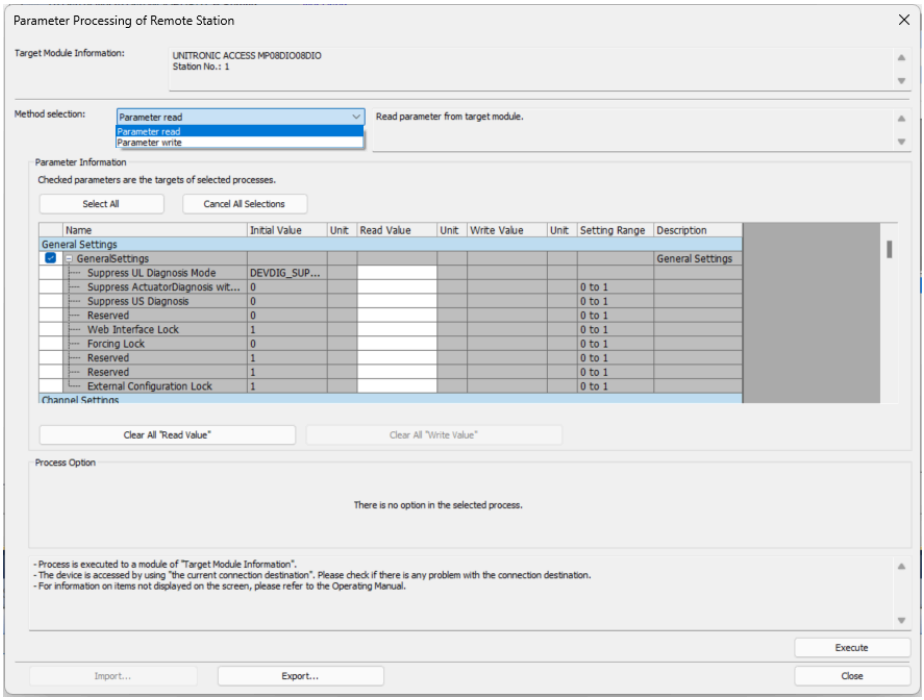
1. Select the UNITRONIC® ACCESS device from the *Module List*. Alternatively, click the button **Detect Now** for automatic detection of devices.



2. Right-click on "Slave Station" and select **Online** > **Parameter Processing of Slave Station...**.



3. In the next window under *Method selection*, choose “Parameter read” or “Parameter write”, depending on which method you want to configure for the UNITRONIC® ACCESS device. For details on the different parameters please refer to chapter Configuration CC-Link IE Field Basic on page 42.



4. After having adjusted the parameters, click on **Communication Setting Reflection of Slave Station** to apply the changes to the respective module.

12 IIoT functionality

The UNITRONIC® ACCESS 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 UNITRONIC® ACCESS 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 UNITRONIC® ACCESS variants provide user administration, which is also applicable for accessing and configuring the IIoT protocols. This allows you to manage all modification options for the device settings via personalized user authorizations.

All IIoT protocols can be used and configured independently of the field bus. It is also possible to use the devices completely without the help of a PLC and control them via IIoT protocols.

Attention: When using the IIoT functionality, a protected local network environment without direct access to the Internet is recommended.

Attention: Only activate one of the IIoT protocols at a time. Exclusively use MQTT **or** OPC UA.

12.1 MQTT

MQTT functions are **only** applicable for the following UNITRONIC® ACCESS variants:

- MP08IOLA08DIO

The MQTT (Message Queuing Telemetry Transport) protocol is an open network protocol for machine-to-machine communication, which provides the transmission of telemetric data messages between devices. The integrated MQTT client allows the device to publish a specific set of information to an MQTT broker.

The publishing of messages can either occur periodically or be triggered manually.

Attention: When using MQTT, the OPC UA protocol must be disabled.

12.1.1 MQTT configuration

In **delivery state**, MQTT functions are **disabled**. The MQTT client can be configured either using the Web interface or directly via a JSON object sent in an HTTP/HTTPS request. For more information see chapter MQTT configuration - Quick start guide on page 92.

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: "**unitronic**")
will-enable	boolean	If true, the device provides a last will message to the broker	true / **false**
will-topic	string	The topic for the last will message.	(Default: **null**)
auto-publish	boolean	If true, all enabled domains will be published automatically in the specified interval.	**true** / false
publish-interval	number	The publish interval in ms if auto-publish is enabled. Minimum is 250 ms.	**2000**
publish-identity	boolean	If true, all identity domain data will be published	**true** / false
publish-config	boolean	If true, all config domain data will be published	**true** / false
publish-status	boolean	If true, all status domain data will be published	**true** / false
publish-process	boolean	If true, all process domain data will be published	**true** / false
publish-devices	boolean	If true, all IO-Link Device domain data will be published	true / **false**
commands-allowed	boolean	Master switch for MQTT commands. If false, the device will not subscribe to any command topic, even if specific command topics are activated below.	true / **false**
force-allowed	boolean	If true, the device accepts force commands via MQTT.	true / **false**
reset-allowed	boolean	If true, the device accepts restart and factory reset commands via MQTT.	true / **false**
config-allowed	boolean	If true, the device accepts configuration changes via MQTT.	true / **false**

Element	Data type	Description	Example data
qos	number	Selects the "Quality of Service" status for all published messages.	**0 = At most once** 1 = At least once 2 = Exactly once

Table 13: MQTT configuration

MQTT response:

The resulting response is a JSON object with a "status" field. Status should be "0" if no error occurred and "-1" if there is an error.

In case of an error, the response contains an error array.

The error array contains an error object for each error occurred. The object consists of a field "Element" which names the config element which caused the error, and a field "Message" for the error message.

- A malformed JSON object produces an error.
- Not existing parameters produce an error.
- Parameters with a wrong data type produce an error.

It is not allowed to write all available parameters at once. You may write only one or a limited number of parameters.

Examples:

```
{"status": -1, "error": [{"Element": "publish-interval", "Message": "Integer
 expected"}]}

{"status": 0}

{"status": -1, "error": [{"Element": "root", "Message": "Not a JSON
 object"}]}
```

For more information see chapter MQTT topics on page 80.

12.1.2 MQTT topics

MQTT mainly relates to topics. All messages are attached to a topic which adds context to the message itself. Topics may consist of any string and they are allowed to contain slashes (`/`) as well as wildcard symbols (`*, #`).

12.1.2.1 Base topic

For all UNITRONIC® ACCESS variants there is a configurable Base topic which is the prefix for all topics. The Base topic can be chosen freely by the user. The Base topic can also contain selected variables as shown in Table 14: Base topic variables on page 80.

Variables in the Base topic have to be written in brackets ("[ ]"). The following variables are possible:

Variable	Description
mac	The MAC address of the device
name	The name of the device
order	The ordering number of the device
serial	The serial number of the device

Table 14: Base topic variables

Example:

The Base topic `"io_[mac]"` translates to `"io_A3B6F3F0F2F1"`.

All data is organized in domains. The domain name is the first level in the topic after the Base topic. Note the following notation:

`Base-Topic/domain/….`

There are the following domains:

Domain name	Definition	Example content
identity	All fixed data which is defined by the used hardware and which cannot be changed by configuration or at runtime.	Device name, ordering number, MAC address, port types, port capabilites and more.
config	Configuration data which is commonly loaded once at startup, mostly by a PLC.	IP address, port modes, input logic, failsafe values and more.
status	All (non-process) data which changes quite often in normal operation.	Bus state, diagnostic information, IO-Link Device status and data.
process	All process data which is produced and consumed by the device itself or by attached devices.	Digital inputs, digital outputs, cyclic IO-Link data.
iold	IO-Link Device parameters according to the IO-Link specification.	Vendor name, product name, serial number, hardware revision, software revision and more.

Table 15: Data domains

There is often one topic used for all gateway related information and topics for each port. All identity topics are published just once at start-up, because this information should never change. All other topics are published either in a fixed interval or just triggered manually, according to the configuration.

Topic	Content examples	Total publish count	Publish interval
[base-topic]/identity/gateway	Name, ordering number, MAC, vendor, I&M etc.	1	Startup
[base-topic]/identity/port/n	Port name, port type	8	Startup
[base-topic]/config/gateway	Configuration parameters, ip address etc.	1	Interval
[base-topic]/config/port/n	Port mode, data storage, mapping, direction	8	Interval
[base-topic]/status/gateway	Bus state, device diagnosis, master events	1	Interval
[base-topic]/status/port/n	Port or channel diagnosis, IO-Link state, IO-Link Device events	8	Interval
[base-topic]/process/gateway	All Digital IN/OUT	1	Interval
[base-topic]/process/port/n	Digital IN/OUT per port, IOL-data, pdValid	8	Interval
[base-topic]/iold/port/n	IO-Link Device parameter	8	Interval

Table 16: Data model

An MQTT client which wants to subscribe to one or more of these topics can also use wildcards.

Full topic	Description
[base-topic]/identity/gateway	Receive only indentity objects for the gateway
[base-topic]/identity/#	Receive all data related to the identity domain
[base-topic]/status/port/5	Receive only status information for port number 5
[base-topic]/+/port/2	Receive information of all domains for port number 2
[base-topic]/process/port/#	Receive only process data for all ports
[base-topic]/config/#	Receive config data for the gateway and all ports.

Table 17: Use case examples

12.1.2.2 Publish topic

Overview of all publish JSON data for the defined topics:

Key	Data type
product_name	json_string
ordering_number	json_string
device_type	json_string
serial_number	json_string
mac_address	json_string
production_date	json_string
fw_name	json_string
fw_date	json_string
fw_version	json_string
hw_version	json_string
vendor_name	json_string
vendor_address	json_string
vendor_phone	json_string
vendor_email	json_string
vendor_techn_support	json_string
vendor_url	json_string
vendor_id	json_integer
device_id	json_integer

Table 18: Identity/gateway

Key	Data type	Range	Default value	Remarks
fieldbus_protocol	json_string	PROFINET, EtherNet/IP, EtherCAT®		
ip_address	json_string		192.168.1.1	
subnet_mask	json_string		255.255.255.0	
report_alarms	json_boolean		0.0.0.0	
report_ul_alarm	json_boolean	true / false	true	
report_do_fault_without_ul	json_boolean	true / false	false	
force_mode_lock	json_boolean	true / false	false	
web_interface_lock	json_boolean	true / false	false	
do_auto_restart	json_boolean	true / false	true	
fast_startup	json_boolean	true / false	false	PROFINET and EIP only

Table 19: Config/gateway

Key	Data type	Range	Default value	Remarks
protocol	json_string	wait_for_io_system wait_for_io_Connection failsafe connected error		
ethernet_port1	json_string	100_mbit/s_full 100_mbit/s 10_mbit/s_full 100_mbit/s		
ethernet_port2	json_string	100_mbit/s_full 100_mbit/s 10_mbit/s_full 100_mbit/s		
module_restarts	json_integer	0 .. 4294967295		
channel_diagnosis	json_boolean	true / false		
failsafe_active	json_boolean	true / false		
system_voltage_fault	json_boolean	true / false		
actuator_voltage_fault	json_boolean	true / false		
internal_module_error	json_boolean	true / false		
simulation_active_diag	json_boolean	true / false		
us_voltage	json_integer	0 .. 32		in Volts
ul_voltage	json_integer	0 .. 32		in Volts
forcemode_enabled	json_boolean	true / false		

Table 20: Status/gateway

Key	Data type	Range	Default value	Remarks
Input_data	json_integer[]			
output_data	json_integer[]			

Table 21: Process/gateway

Key	Data type	Range	Default value	Remarks
port	json_integer	1 .. 8		
type	json_string	digital_universal digital_input digital_Output io_link		
max_output_power_cha	json_string	2.0_mA 0.5_mA		
max_output_power_chb	json_string	2.0_mA 0.5_mA		
channel_cha	json_string	input/output input output io_link aux		
channel_chb	json_string	input/output input output io_link aux		

Table 22: Identity/port/1 .. 8

Key	Data type	Range	Default value	Remarks
port	json_integer	1 .. 8		
direction_cha	json_string	input/output input output		
restart_mode_cha	json_string	Manual Auto		
restart_mode_chb	json_string	Manual Auto		
input_polarity_cha	json_string	NO NC		
input_polarity_chb	json_string	NO NC		
input_filter_cha	json_integer			ms
input_filter_chb	json_integer			ms
do_auto_restart_cha	json_boolean	true / false		
do_auto_restart_chb	json_boolean	true / false		

Table 23: Config/port/1 .. 8

Key	Data type	Range	Default value	Remarks
port	json_integer	1 .. 8		
physical_state_cha	json_integer	0 .. 1		
physical_state_chb	json_integer	0 .. 1		
actuator_short_circuit_cha	json_boolean	true / false		
actuator_short_circuit_chb	json_boolean	true / false		
sensor_short_circuit	json_boolean	true / false		
current_cha	json_integer			mA
current_chb	json_integer			mA
current_pin1	json_integer			mA

Table 24: Status/port/1 .. 8

12.1.2.3 Command topic (MQTT Subscribe)

The main purpose of MQTT is to publish data from the device to a broker. This data can then be received by any subscriber who is interested in this data. But also the other way round is possible. The device can subscribe to a topic on the broker and is then able to receive data. This data can contain configuration or forcing data. This allows the user to fully control a device via MQTT only, without using other ways of communication like Web or REST.

If the configuration allows commands in general, the device subscribes to special Command topics on which it can receive commands from other MQTT clients. The Command topic is based upon the Base topic. It always has the following form:

`[base-topic]/command`

After the Command topic, there are fixed topics for different writeable objects. The data format of the MQTT payload is always JSON. It is possible to set only a subset of the possible objects and fields.

[…]/forcing

Use the Command topic `[base-topic]/command/forcing` for *Force object* data. The *Force object* can contain any of the following properties:

Property	Data type	Example values	Remarks
forcemode	boolean	true / false	Forcing Authority: on/off
digital	array (Table 26: Force object: Digital on page 89)		
iol	array (Table 27: Force object: IOL (IO-Link devices only) on page 89)		

Table 25: Force object properties

For the *Force object* properties `digital` and `iol`, there are several value specifications arrayed:

Property	Data type	Example values	Remarks
port	integer	1, 2, 5	
channel	string	"a", "b"	
force_dir	string	"out", "in", "clear"	
force_value	integer	0, 1	

Table 26: Force object: Digital

Property	Data type	Example values	Remarks
port	integer	0, 1, 5	
output	array[integer]	[55, 88, 120]	
input	array[integer]		Input-Simulation

Table 27: Force object: IOL (IO-Link devices only)

[…]/config

Use the Command topic `[base-topic]/command/config` for *Config object* data. The *Config object* can contain any of the following properties:

Property	Data type	Example values	Remarks
portmode	array (Table 29: Config object: Portmode on page 90)		
ip_address	string	"192.168.1.5"	
subnet_mask	string	"255.255.255.0"	
gateway	string	"192.168.1.100"	

Table 28: Config object properties

For the *Config object* property `portmode`, there are several value specifications arrayed:

Property	Data type	Example values	Remarks
port	integer	2	
channelA*	string	"dio", "di", "do", "iol", "off"	
channelB*	string	"dio", "di", "do", "iol", "off", "aux"	
inlogicA	string	"no", "nc"	
inlogicB	string	"no", "nc"	
filterA	integer	3	input filter in ms
filterB	integer	3	input filter in ms
autorestartA	boolean		
autorestartB	boolean		
iolValidation	integer	0 = NoCheck 1 = Type 1.0 2 = Type 1.1 3 = Type 1.1 BR 4 = Type 1.1 RES	
iolDeviceID	integer		for validation
iolVendorID	integer		for validation

Table 29: Config object: Portmode

*channelA = Pin 4, channelB = Pin 2

[…]/reset

Use the Command topic `[base-topic]/command/reset` for *Reset object* data about restart and factory reset issues. The *Reset object* can contain any of the following properties:

Property	Data type	Example values	Remarks
factory_reset	boolean	true / false	
system_reset	boolean	true / false	

Table 30: Reset object properties

[…]/publish

Use the Command topic `[base-topic]/command/publish` for *Publish object* data.

Trigger publish of all topics manually (can be used when auto publish is off or long interval is set).

12.1.3 MQTT configuration - Quick start guide

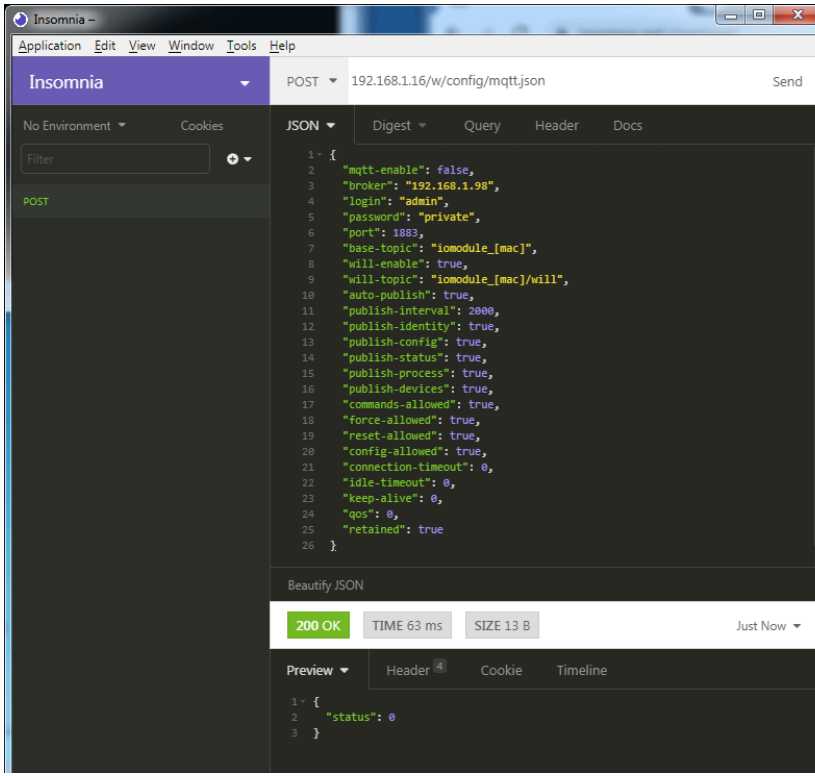
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

12.1.3.1 MQTT configuration via JSON

1. Depending on your application case, download and install *Insomnia* or a comparable application: https://insomnia.rest/download/

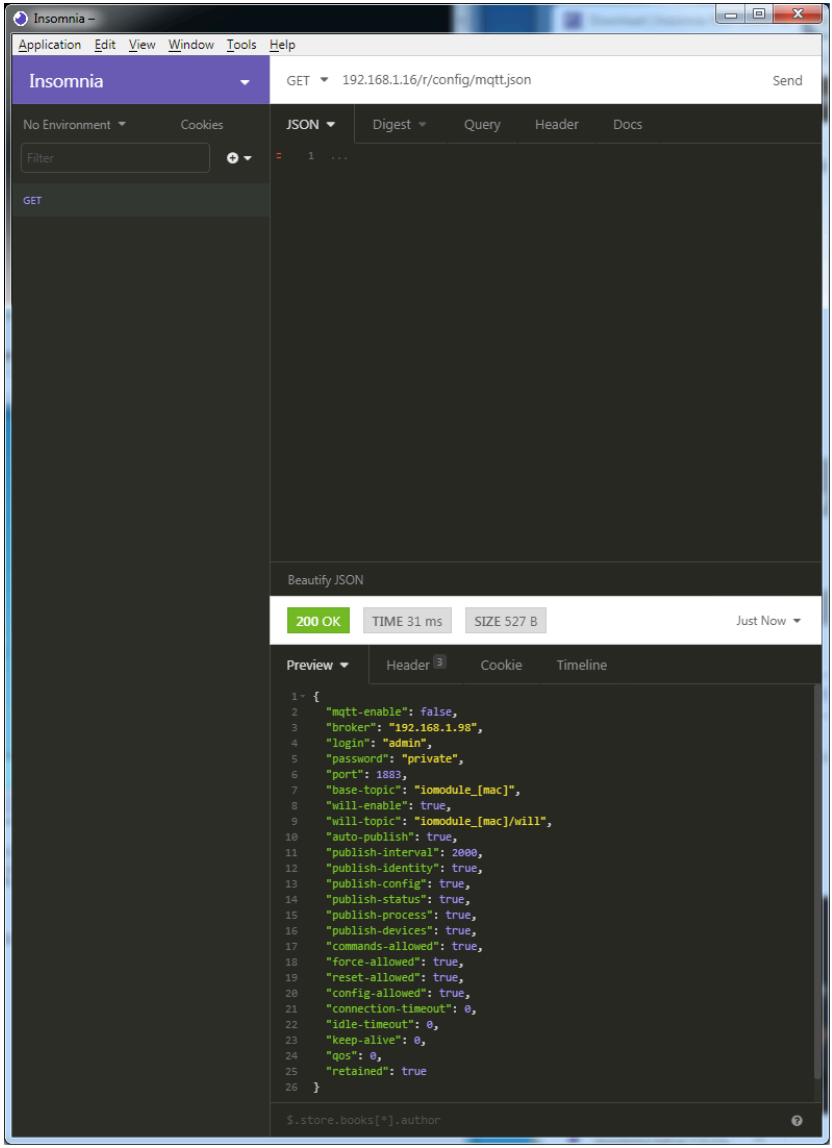
2. Configure MQTT:

POST: `[IP-address]/w/config/mqtt.json`



3. Read MQTT:

GET: `[IP-address]/r/config/mqtt.json`



12.2 OPC UA

OPC UA functions are **only** applicable for the following UNITRONIC® ACCESS variants:

- MP08IOLA08DIO

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. UNITRONIC® ACCESS provides an OPC UA server on field device level, with which an OPC UA client can connect for information exchange secure in transmission.

For OPC UA, we comply (apart from the exceptions listed below) with the IO-Link Companion Specification, which can be downloaded from https://lapp.com or directly from io-link.com.

Attention: When using OPC UA, the MQTT protocol must be disabled.

Feature	Support
Managing IODDs (chapter 6.1.6 in the specification)	Not supported
Mapping IODD information to OPC UA ObjectTypes (chapter 6.3 in the specification)	Not supported
IOLinkIODDDeviceType (chapters 7.2 ff. in the specification)	Not supported
ObjectTypes generated based on IODDs (chapters 7.3 ff. in the specification)	Not supported
Creation of Instances based on ObjectTypes generated out of IODDs (chapter 7.4 in the specification)	Not supported
IODDManagement Object (chapter 8.2 in the specification)	Not supported
RemoveIODD Method (chapter 8.3 in the specification)	Not supported

Table 31: Non-supported OPC UA features according to the IO-Link Companion Specification

12.2.1 OPC UA configuration

In **delivery state**, OPC UA functions are **disabled**. The OPC UA Server can be configured either using the Web interface or directly via a JSON Object sent in an HTTP/HTTPS request. For more information see OPC UA configuration - Quick start guide on page 98.

The configuration URL is:

`http://[ip-address]/w/config/opcua.json`

The configuration can also read back as a JSON file:

`http://[ip-address]/r/config/opcua.json`

The configuration is a JSON object. Each JSON member is a configuration element. The object must not contain all elements. Only the provided elements will be changed. All configuration changed applies only after a device restart.

There are the following configuration elements (default values in bold):

Element	Data type	Description	Example data
port	integer	Server port for the OPC UA server.	0, **4840**, 0xFFFF
opcua-enable	boolean	Master switch for the OPC UA server.	true / **false**
anon-allowed	boolean	If true, anonymous login is allowed.	**true** / false
commands-allowed	boolean	Master switch for OPC UA commands. If false there will be no writeable OPC UA objects.	true / **false**
force-allowed	boolean	If true, the device accepts force commands via OPC UA.	true / **false**
reset-allowed	boolean	If true, the device accepts restart and factory reset commands via OPC UA.	true / **false**
config-allowed	boolean	If true, the device accepts configuration changes via OPC UA.	true / **false**

Table 32: OPC UA Configuration

All configuration elements are optional and do not need a specific order. Not every element is required to be sent. This means that only configuration changes will be taken over.

Optional: The configuration parameters of OPC UA can be set directly via the Web interface. It is possible to download the Web interface for sharing with other devices.

Response:

The resulting response is a JSON object with a status field. Status should be "0" if no error occurred and "-1" if there is an error.

In case of an error, the response contains an error array.

The error array contains an error object for each error occurred. The object consists of a field "Element" which names the config element which caused the error, and a field "Message" for the error message.

Examples:

```
{"status": -1, "error": [{"Element": "upcua-enable", "Message": "Boolean
 expected"}]}

{"status": 0}

{"status": -1, "error": [{"Element": "root", "Message": "Not a JSON
 object"}]}
```

12.2.2 OPC UA address space

OPC UA provides different services on the UNITRONIC® ACCESS 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 UNITRONIC® ACCESS devices. The objects and information displayed depend on the device variant used.

12.2.3 OPC UA configuration - Quick start guide

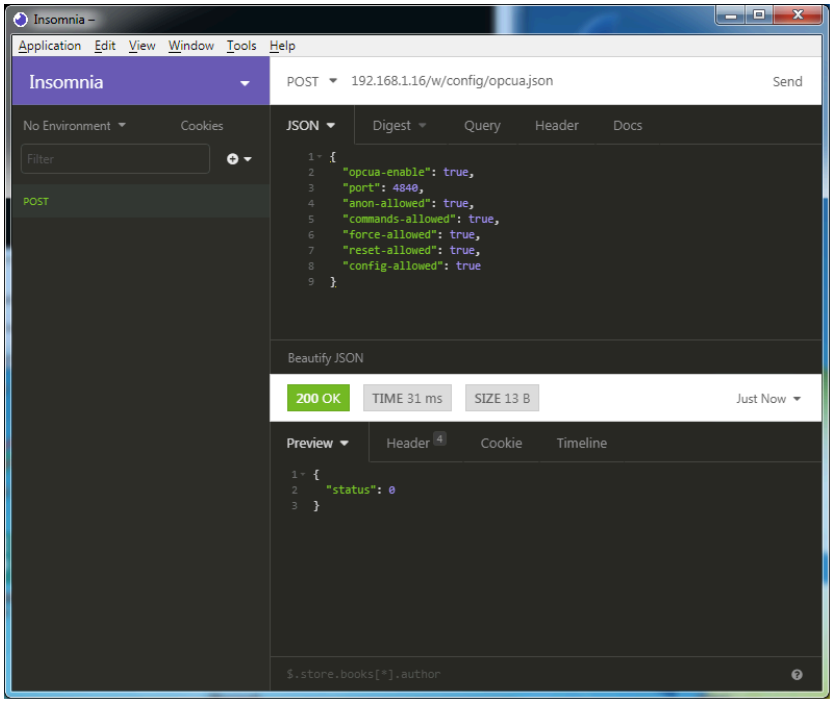
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

12.2.3.1 OPC UA configuration via JSON

1. Depending on your application case, download and install *Insomnia* or a comparable application: https://insomnia.rest/download/

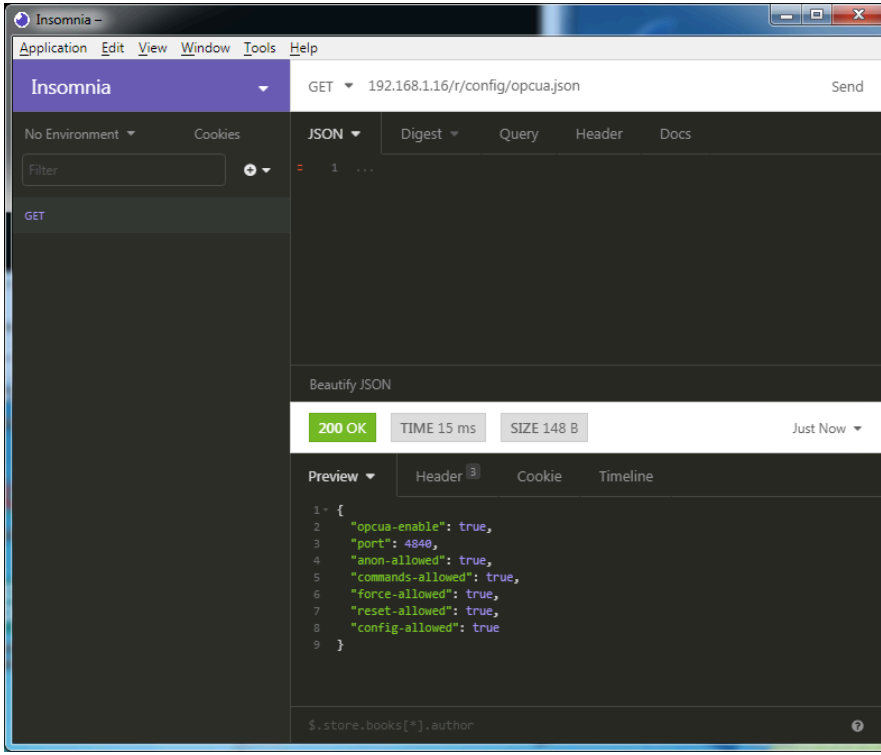
2. Configure OPC UA:

POST: `[IP-address]/w/config/opcua.json`



3. Read OPC UA:

GET: `[IP-address]/r/config/opcua.json`



12.3 REST API

The Representational State Transfer – Application Programming Interface (REST API) is a programmable interface which uses HTTP/HTTPS requests to GET and POST data. This enables the access to detailed device information.

For all UNITRONIC® ACCESS variants, the REST API can be used to read the device status. For the UNITRONIC® ACCESS multi-protocol variants, the REST API can also be used to write configuration and forcing data.

There are two different REST API standards you can use for the requests:

1. A standardized REST API that has been specified by the IO-Link Community and is described separately:

   `JSON_Integration_10222_V100_Mar20.pdf`

   Please download the file from https://lapp.com or directly from io-link.com.

   Attention: Consider the following table to get an overview of the supported features of the IO-Link specification:

Feature		Supported
Gateway	GET /identification	Yes
	GET /capabilities	Yes
	GET /configuration	Yes
	POST /configuration	Yes
	POST /reset	Yes
	POST /reboot	Yes
	GET /events	Yes
Master	GET /masters	Yes
	GET /capabilities	Yes
	GET /identification	Yes
	POST /identification	Yes

Feature		Supported
Port	GET /ports	Yes
	GET /capabilities	Yes
	GET /status	Yes
	GET /configuration	Yes
	POST /configuration	Yes
	GET /datastorage	Yes
	POST /datastorage	Yes
Devices	GET /devices	Yes
	GET /capabilities	Yes
	GET /identification	Yes
	POST /identification	Yes
	GET /processdata/value	Yes
	GET /processdata/getdata/value	Yes
	GET /processdata/setdata/value	Yes
	POST /processdata/value	Yes
	GET /parameters	Not supported
	GET /parameters/{index}/subindices	Not supported
	GET /parameters/{parameterName}/subindices	Not supported
	GET /parameters/{index}/value	Not supported
	GET /parameters/{index}/subindices/{subindex}/value	Not supported
	GET /parameters/{parameterName}/value	Not supported
	GET /parameters/{parameterName}/subindices/{subParameterName}/value	Not supported
	POST /parameters/{index}/value	Not supported
	POST /parameters/{parameterName}/value	Not supported
	POST /parameters/{index}/subindices/{subindex}/value	Not supported
	POST /parameters/{parameterName}/subindices/{subParameterName}/value	Not supported
	POST /blockparametrization	Yes
	GET /events	Yes

Feature		Supported
IODD	GET /iodds	Not supported
	POST /iodds/file	Not supported
	DELETE /iodds	Not supported
	GET /iodds/file	Not supported

Table 33: Support of REST API features according to the IO-Link specification

2. A customized LAPP REST API that is described in the following chapters.

12.3.1 Standard device information

Request method:	http GET
Request URL:	`<ip>/info.json`
Parameters	n.a.
Response format	JSON

The goal of the "Standard device information" request is to get a complete snapshot of the current device status. The format is JSON. For IO-Link devices, all ports with connected IO-Link device information are included.

12.3.2 Structure

Name	Data type	Description	Example
name	string	Device name	"MP08IOLA08DIO"
order-id	string	Ordering number	"381166717"
fw-version	string	Firmware version	"V.11.2.0.0 - 08.08.2024"
hw-version	string	Hardware version	"V.1.00"
mac	string	MAC address of the device	"7C F9 5C 4C CC CE"
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
iol	IOL Object	Contains all IO-Link related information such as events, port states, device parameters.	
iol/diagGateway	array of DIAG	Array of currently active device/ gateway related events	
iol/diagMaster	array of DIAG	Array of currently active IOL-Master related events	
iol/ports	array of PORT (8)	Contains one element for each IO-Link port	
CHANNEL Object			
name	string	Name of channel	
type	number	Hardware channel type as number: 0 = DIO 1 = Input 2 = Output 3 = Input/Output 4 = IO-Link 5 = IOL AUX 6 = IOL AUX with DO 7 = IOL AUX with DO. Can be deactivated. 8 = Channel not available	
type_text	string	Textual representation of the channel type	
config	number	Current configuration of the channel: 0 = DIO 1 = Input 2 = Output 3 = IO-Link 4 = Deactivated 5 = IOL AUX	
config_text	string	Textual representation of the current config	
inputState	boolean	Input data (producing data) bit to the PLC	
outputState	boolean	Output data bit to the physical output pin	

Name	Data type	Description	Example
forced	boolean	True, if the output pin of this channel is forced	
simulated	boolean	True, if the input value to the PLC of this channel is simulated	
actuatorDiag	boolean	True, if the output is in short circuit / overload condition	
sensorDiag	boolean	True, if the sensor supply (Pin 1) is in short circuit / overload condition	
maxOutputCurrent_mA	number	Maximum output current of the output in mA	
current_mA	number	Measured current of the output in mA (if current measurement is available)	
voltage_mV	number	Measured voltage of this output in mV (if voltage measurement is available)	
PORT Object			
port_type	string	Textual representation of the IO-Link port type	
iolink_mode	number	**Current port mode:** 0 = Inactive 1 = Digital output 2= Digital input 3 = SIO 4 = IO-Link	
iolink_text	string	Textual representation of the current port mode	"Digital Input"
aux_mode	number	Indicates the configured mode for the Pin 2: 0 = No AUX 1 = AUX output (always on) 2 = Digital output (can be controlled by cyclic data) 3 = Digital input	
aux_text	string	Textual representation of the current aux mode	"AUX Output"
cq_mode	number	Port mode according to IOL specification	
iq_mode	number	Pin2 mode according to IOL specification	

Name	Data type	Description	Example
port_status	number	Port status according to IOL specification	
ds_fault	number	Data storage error number	
ds_fault_text	string	Textual data storage error.	
device	DEVICE Object	IO-Link device parameters. → Null if no IO-Link communication active	
diag	array of DIAG (n)	Array of port related events	
DIAG Object			
error	number	Error code	
source	string	Source of the current error.	"device" "master"
eventcode	number	Event code according to IO-Link specification	
eventqualifier	number	Event qualifier according to IO-Link specification	
message	string	Error message	"Supply Voltage fault"
DEVICE Object		Standard parameters of the IOL-Device	
device_id	number		
vendor_id	number		
serial	string		
baudrate	string	Baudrate (COM1,2,3)	
cycle_time	number	Cycle time in microseconds	
input_len	array of numbers (n)	IOL input length in bytes	
output_len	array of numbers (n)	IOL output length in bytes	
input_data	array of numbers (n)	IOL input data	
output_data	array of numbers (n)	IOL output data	
pd_valid	number	"1", if IOL input data is valid	
pdout_valid	number	"1", if IOL output data is valid	
FORCING Object		Forcing information of the device	
forcingActive	boolean	Force mode is currently active	

Name	Data type	Description	Example
forcingPossible	boolean	True, if forcing is possible and force mode can be activated	
ownForcing	boolean	True, if forcing is performed by REST API at the moment	
forcingClient	string	Current forcing client identifier	
digitalOutForced	array of numbers (2)	The force values of all 16 digital output channels.	
digitalOutMask	array of numbers (2)	The forcing mask of all 16 digital output channels.	
digitalInForced	array of numbers (2)	The force values of all 16 digital input channels.	
digitalInMask	array of numbers (2)	The forcing mask of all 16 digital input channels.	

12.3.3 Configuration and forcing

Method: POST

URL: `<ip>/w/force.json`

Parameters: None

Post-Body: JSON Object

Property	Data type	Example values	Description
forcemode	boolean	true / false	Forcing authority on/off
portmode	array (Port mode object)		
digital	array (Digital object)		
iol	array (IOL object)		

Table 34: Root object

Property	Data type	Example values	Remarks
port	integer	0..7	
channel	string	"a","b"	optional default is "a"
direction	string	"dio","di","do","iol", "off", "aux"	
aux	string	"dio","di","do","iol", "off", "aux"	IOL only, but optional
inlogica	string	"no","nc"	
inlogicb	string	"no","nc"	
inputlatch	bool	true / false	enable/disable input latch, optional
inputext	integer	Depends on the fieldbus: ▶ eip: 0 (off) - 255 (ms) ▶ ethercat: 0 (off) - 255 (ms) ▶ pns: 0 (off) - 255 (ms) ▶ cclink: 0 (off) - 255 (ms) ▶ mbtcp: 0 (off) - 255 (ms)	set input extension, optional
inputfilter	integer	0 .. 255	set input filter, optional

Table 35: Port mode object

Property	Data type	Example values	Remarks
port	integer	0..7	
channel	string	"a","b"	
force_dir	string	"phys_out","plc_in","clear"	optional default is "phys_out"
force_value	integer	0,1	

Table 36: Digital object

Property	Data type	Example values	Remarks
port	integer	0..7	
output	array[integer] or null to clear forcing	[55,88,120]	Output forcing
input	array[integer] or null to clear forcing	[20,0,88]	Input simulation to PLC

Table 37: IOL object

12.3.4 Reading and writing ISDU parameters

The *Indexed Service Data Unit* (ISDU) provides a highly flexible message format, which can contain single or multiple commands.

UNITRONIC® ACCESS IOL-Masters with IIoT support reading and writing ISDU parameters from connected IOL-Devices. It is possible to do this as a bulk transfer by reading and writing of multiple ISDU parameters via a single request.

12.3.4.1 Reading ISDU

Method: POST

URL: `<ip>/r/isdu.json`

Parameters: port (0-7)

Example:

```
192.168.1.20/r/isdu.json?port=5
```

Post-Body: JSON array of read ISDU object

Property	Data type	Example values	Remarks
ix	integer	0-INT16	Index to be read
subix	integer	0-INT8	Subindex to be read

Table 38: Read ISDU object

Property	Data type	Example values	Remarks
status	integer	0, -1	0 = no error, -1= an error occured
message	string		Error Message if error occured
data	array (Read ISDU data object)		data, if no error occured. otherweise null

Table 39: Read ISDU response object

Property	Data type	Example values	Remarks
ix	integer	0-INT16	Index that was read
subix	integer	0-INT8	Subindex that was read
status	integer	0, -1	0 = no error, -1= an error occured
eventcode	integer		IOL eventcode if status is -1
data	array[integer]		data, if no error occured. otherweise null

Table 40: Read ISDU data object

12.3.4.2 Writing ISDU

Method: POST

URL: `<ip>/w/isdu.json`

Parameters: port (0-7)

Post-Body: JSON array of write ISDU object

Property	Data type	Example values	Remarks
ix	integer	0-INT16	Index to be read
subix	integer	0-INT8	Subindex to be read
data	array[integer]		Data to be written

Table 41: Write ISDU object

Response: Write ISDU response object

Property	Data type	Example values	Remarks
status	integer	0, -1	0 = no error, -1= an error occured
message	string		Error Message if error occured
data	array (Write ISDU data object)		data, if no error occured. otherweise null

Table 42: Write ISDU response object

Property	Data type	Example values	Remarks
ix	integer	0-INT16	Index that was written
subix	integer	0-INT8	Subindex that was written
status	integer	0, -1	0 = no error, -1= an error occured
eventcode	integer		IOL eventcode if status is -1

Table 43: Write ISDU data object

Attention: For UNITRONIC® ACCESS device variants with HTTPS feature, https:// must be used in front of <ip> for every REST API.

12.3.5 Upload and process an IODD file

The REST API supports IODD file upload to the IO-Link Master.

Perform the following work steps:

1. Check file upload status

   Send request: GET file_upload

   Purpose: Get file upload status to check if there is another upload in progress.

   Expected response:

```
{
"status": 0,
"progress": 0,
"name": "",
"action": 0,
"upid": 0,
"errid": 0,
"errstr": "",
"pschr": 0
}
```

   Check the status ID. If status is '0', you can start a new iodd upload process. For reference, see tables Table 44: Status ID and meaning on page 118 and Table 45: Error ID and meaning on page 119. Proceed with the next step.

2. Initiate file upload

   Send request: POST file_upload

   Content-Type: application/json

   Purpose: Send details about the file to be uploaded.

   Expected response:

```
{"size":<total size>,"name":"<file name>",
"action":"iodd", "upid":
<upload id>}
{
"size": <total size>,
"name": "<file name>",
"action": "iodd",
"upid": <upload id>
}
```

The upload id (upid) is a number used by the backend to identify a specific upload and parsing process. It has to be used as a query parameter in the following steps.

The action will allways be iodd.

The size is the total size of the file in bytes.

The correct content type has to be set.

Note: Remember the upload ID (upid) for subsequent steps

3. Upload file content

   Send request: POST file_upload?upid=<value> → Use the upid value from Step 2.

   Content-Type: application/octet-stream → The correct content type has to be set.

   Purpose: Send file or file chunks (max chunk size: 64KB).

   Attention: Sending file chunks bigger than 64KB will result in unresponsive behavior.

4. Monitor upload status

   Send request: GET file_upload?upid=<value> → Use the upid value from Step 2.

   Purpose: Get the current file upload status.

   Expected response:

```
{
"status": <status id value>,
"progress": <percentage>,
"name": "<file name given in step 2>",
"action": "iodd",
"upid": <upload id chosen in step 2>,
"errid": <error id>,
"errstr": "",
"pschr": <count of parsed characters>
}
```

   Repeat this step until the status becomes 'idle'. For some states this request triggers the necessary transitions in the internal state machine.

Only after the backend is sure that the correct client identified by its upid received the action finished or error state it will transition to the next one, the idle state.

The fields now show values depending on what was sent in step 2 and on the current process status.

Status ID	Status
0	File upload idle. New upload can be triggered.
1	File upload started.
2	File upload in progress.
3	File upload finished.
4	Error during file upload.
5	File upload timeout.
6	IODD parsing started.
7	IODD parsing finished.
8	IODD parsing error.
9	IODD parsing canceled.

Table 44: Status ID and meaning

ID	Error
0	No error.
1	Json parsing error.
2	Json type error.
4	Upload error.
5	File opening error.
6	File writing error.
7	Thread creating error.
8	Error during file copy.
9	Upload timeout.
10	Upload size exceeded.
11	Unknown action.
12	No upload id.
13	IODD paasing error.
14	Internal error.
15	IODD store full. Delete an IODD before uploading a new one.
16	Internal error.
17	IODD file CRC error.
18	Standard IODD file crc error.
19	No available space for parsing.

Table 45: Error ID and meaning

12.3.6 Example: Reading ISDU

ISDU read request

```
[
 {"ix":5,"subix":0},
 {"ix":18,"subix":0},
 {"ix":19,"subix":0},
 {"ix":20,"subix":0}
]
```

Response

```
{
 "message":"OK",
 "data":
[
 {"ix":5,"subix":0,"status":-1,"eventcode":32785},
 {"ix":18,"subix":0,"data":[79,68,83,49,48,76,49,46,56,47,76,65,54,44,50,
        48,48,45,77,49,50],"status":0},
 {"ix":19,"subix":0,"data":[53,48,49,50,57,53,51,53],"status":0},
 {"ix":20,"subix":0,"data":[100,105,115,116,97,110,99,101,32,115,101,110,
        115,111,114],"status":0}
],
"status":0}
```

12.3.7 Example: Writing ISDU

ISDU write request

```
[
 {"ix":24,"subix":0,"data":[97,98,99,100,101,102]},
 {"ix":9,"subix":0,"data":[97,97,97,97,97,98]}
]
```

Response

```
{
 "message":"OK",
 "data":[
 {"ix":24,"subix":0,"status":0},
 {"ix":9,"subix":0,"eventcode":32785,"status":-1}
],
"status":0}
```

12.4 CoAP server

The CoAP server functions are **only** applicable for the following UNITRONIC® ACCESS variants:

- MP08IOLA08DIO

The **Co**nstrained **A**pplication **P**rotocol (CoAP) is a specialized Internet application protocol for constrained networks such as lossy or low power networks. CoAP is useful especially in M2M (Machine to Machine) communication and can be used to translate simplified HTTP/HTTPS requests of low speed networks.

CoAP is based on the Server-Client principle and a service layer protocol that lets nodes and machines communicate with each other. The UNITRONIC® ACCESS multi-protocol variants provide CoAP server functionalities via a REST API interface over UDP.

12.4.1 CoAP configuration

In delivery state, CoAP functions are *disabled*. The CoAP server can be configured either using the Web interface or directly via a JSON object sent in an HTTP/HTTPS request. For more information see chapter CoAP configuration - Quick start guide on page 125.

The configuration URL is:

`http://[ip-address]/w/config/coapd.json`

The configuration can also read back as a JSON file:

`http://[ip-address]/r/config/coapd.json`

The configuration is a JSON object. Each JSON member is a configuration element. The object must not contain all elements. Only the provided elements will be changed. The configuration changes apply only after a device restart.

The following configuration elements are available (default values in bold):

Element	Data type	Description	Example data
enable	boolean	Master switch for the CoAP server	true / **false**
port	integer (0 to 65535)	Port of the CoAP server	**5683**

Table 46: CoAP configuration

CoAP response:

The resulting response is a JSON object with a "status" field. Status should be "0" if no error occurred, and "-1" if there is an error.

In case of an error, the response contains an error array.

The error array contains an error object for each error occurred. The object consists of a field "Element" which names the config element that caused the error, and of a field "Message" for the error message.

Examples:

```
{"status": -1, "error": [{"Element": "upcua-enable", "Message": "Boolean
 expected"}]}

{"status": 0}

{"status": -1, "error": [{"Element": "root", "Message": "Not a JSON
 object"}]}
```

12.4.2 REST API access via CoAP

A connection to the CoAP server running on the UNITRONIC® ACCESS multi-protocol variants can be established via the following URL:

`coap://[ip-address]:[port]/[api]`

For UNITRONIC® ACCESS, the following REST API Requests (JSON format) can be accessed via a CoAP endpoint:

Type	API	Note
GET	/r/status.lr	
GET	/r/system.lr	
GET	/info.json"	
GET	/r/config/net.json	
GET	/r/config/mqtt.json	
GET	/r/config/opcua.json	
GET	/r/config/coapd.json	
GET	/r/config/syslog.json	
GET	/contact.json	
GET	/fwup_status	
GET	/iolink/v1/gateway/identification	
GET	/iolink/v1/gateway/capabilities	
GET	/iolink/v1/gateway/configuration	
GET	/iolink/v1/gateway/events	
GET	/iolink/v1/masters	
GET	/iolink/v1/masters/1/capabilities	
GET	/iolink/v1/masters/1/identification	
GET	/iolink/v1/masters/1/ports	
GET	/iolink/v1/masters/1/ports/{port_number}/capabilities	This API is available for all 8 ports. {port_number} should be between "1" and "8".
GET	/iolink/v1/masters/1/ports/{port_number}/status	This API is available for all 8 ports. {port_number} should be between "1" and "8".
GET	/iolink/v1/masters/1/ports/{port_number}/configuration	This API is available for all 8 ports. {port_number} should be between "1" and "8".
GET	/iolink/v1/devices/master1port{port_number}/identification	This API is available for all 8 ports. {port_number} should be between "1" and "8".

Type	API	Note
GET	/iolink/v1/devices/master1port{port_number}/capabilities	This API is available for all 8 ports. {port_number} should be between "1" and "8".
GET	/iolink/v1/devices/master1port{port_number}/processdata/getdata/value	This API is available for all 8 ports. {port_number} should be between "1" and "8".
GET	/iolink/v1/devices/master1port{port_number}/events	This API is available for all 8 ports. {port_number} should be between "1" and "8".

Table 47: REST API access via CoAP

12.4.3 CoAP configuration - Quick start guide

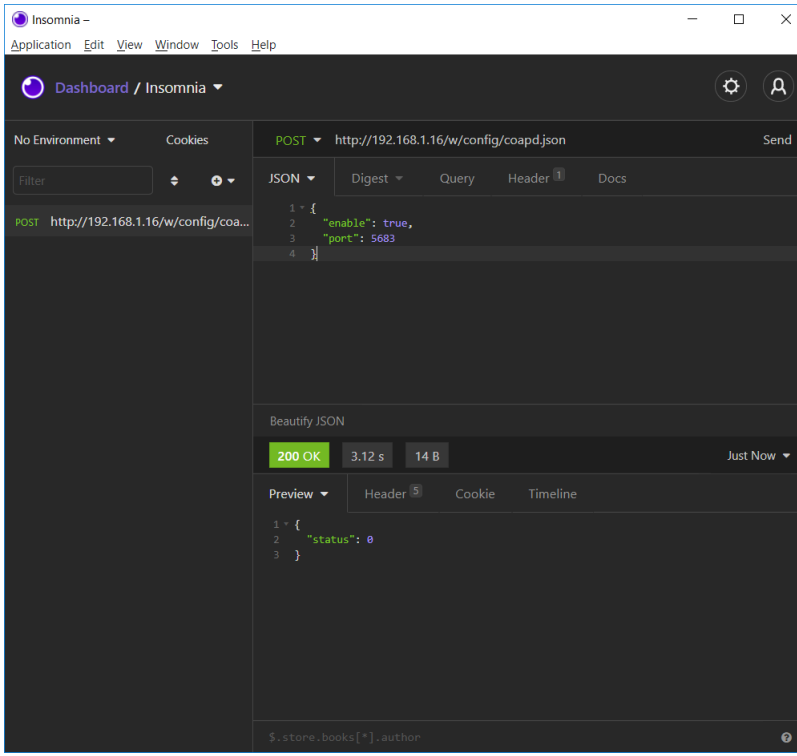
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

12.4.3.1 CoAP configuration via JSON

1. Depending on your application case, download and install *Insomnia* or a comparable application: https://insomnia.rest/download/

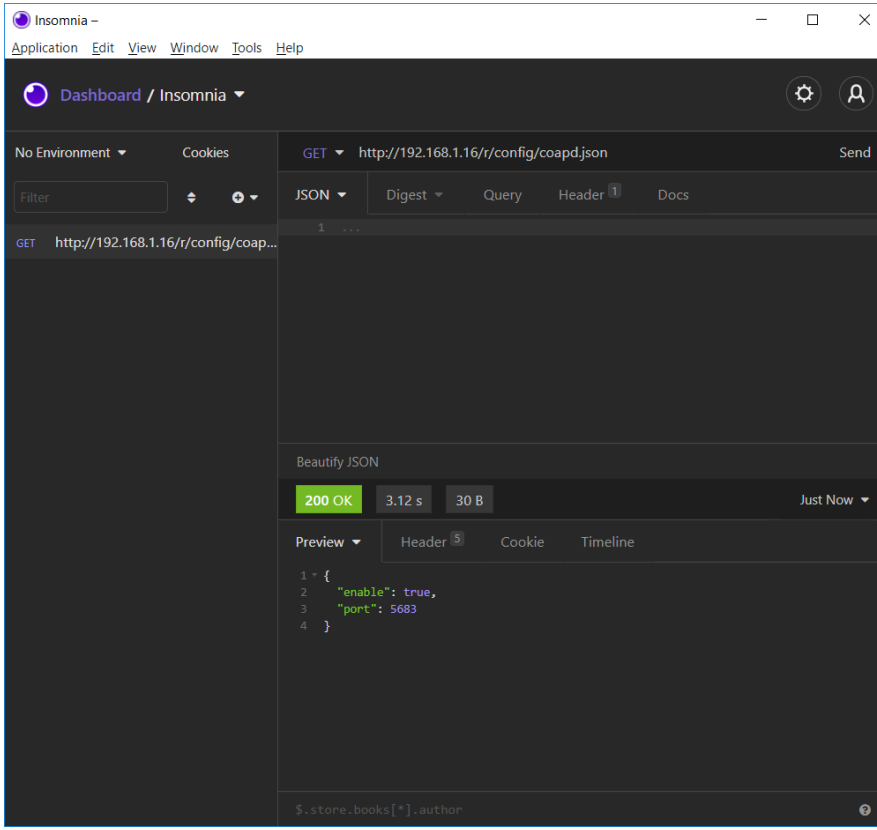
2. Configure CoAP:

POST: `[IP-address]/w/config/coapd.json`



3. Read CoAP configuration:

GET: `[IP-address]/r/config/coapd.json`



12.5 Syslog

Syslog functions are **only** applicable for the following UNITRONIC® ACCESS variants:

- MP08IOLA08DIO

The UNITRONIC® ACCESS multi-protocol variants provide a Syslog client which can connect with a configured Syslog server and is able to log messages.

Syslog is a platform-independent standard for logging messages. Each message contains a timestamp as well as information about the severity level and the subsystem. The Syslog protocol RFC5424 is based on the Server-Client principle and lets machines and devices send messages in the network and collect them centrally. (For more details on the used syslog standard, please refer to https://datatracker.ietf.org/doc/html/rfc5424.)

UNITRONIC® ACCESS supports the storage of 256 messages in a ring buffer which are sent to the configured Syslog server. When the ring is full with 256 messages, the oldest message is always replaced by the newly arriving messages. All messages can be saved on the Syslog server. The Syslog client of the IO-Link Master will not store any message permanently.

12.5.1 Syslog configuration

In **delivery state**, Syslog functions are **disabled**. The Syslog client can be configured either using the Web interface or directly via a JSON object sent in an HTTP/HTTPS request. For more information see chapter Syslog configuration - Quick start guide on page 130.

The configuration URL is:

```
http://[ip-address]/w/config/syslog.json
```

The configuration can also read back as a JSON file:

```
http://[ip-address]/r/config/syslog.json
```

The configuration is a JSON object. Each JSON member is a configuration element. The object must not contain all elements. Only the provided

elements will be changed. The configuration changes apply only after a device restart.

The following configuration elements are available (default values in bold):

Element	Data type	Description	Example data
syslog-enable	boolean	Master switch for the Syslog client	true / **false**
global-severity	integer	Severity level of Syslog client 0 – Emergency 1 – Alert 2 – Critical **3 – Error** 4 – Warning 5 – Notice 6 – Info 7 – Debug The client will log all messages of severity according to the setting, including all below levels.	0/1/2/**3**/4/5/6/7
server-address	string (IP address)	IP address of the Syslog server	192.168.0.51 (Default: **null**)
server-port	integer (0 to 65535)	Server port of the Syslog server	**514**
server-severity	integer (0 to 7)	Severity level of Syslog server 0 – Emergency 1 – Alert 2 – Critical **3 – Error** 4 – Warning 5 – Notice 6 – Info 7 – Debug	0/1/2/**3**/4/5/6/7

Table 48: Syslog configuration

Syslog response:

The resulting response is a JSON object with a "status" field. Status should be "0" if no error occurred, and "-1" if there is an error.

In case of an error, the response contains an error array.

The error array contains an error object for each error occurred. The object consists of a field "Element" which names the config element that caused the error, and of a field "Message" for the error message.

Examples:

```
{"status": -1, "error": [{"Element": "upcua-enable", "Message": "Boolean
 expected"}]}

{"status": 0}

{"status": -1, "error": [{"Element": "root", "Message": "Not a JSON
 object"}]}
```

12.5.2 Syslog configuration - Quick start guide

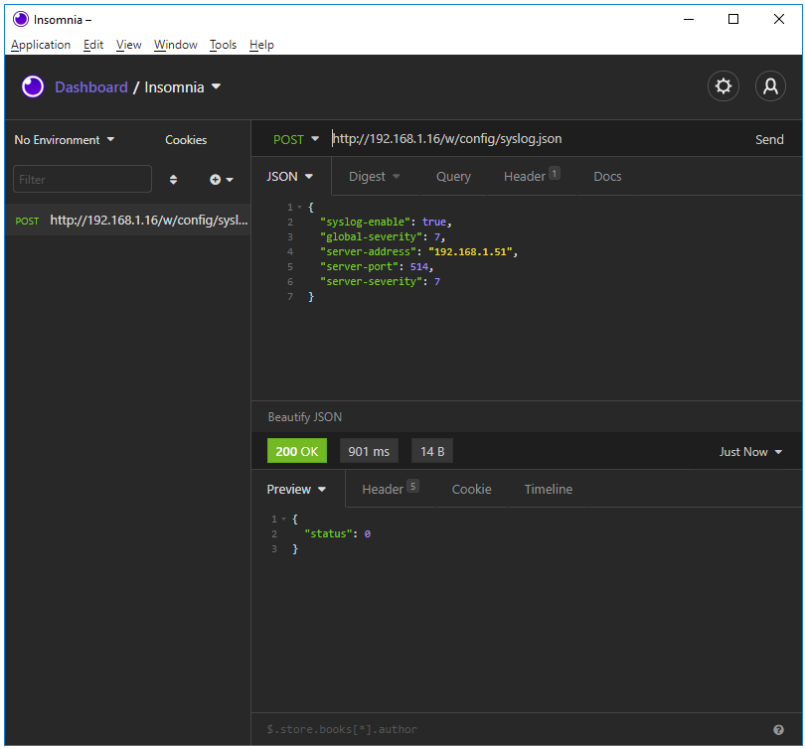
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

12.5.2.1 Syslog configuration via JSON

1. Depending on your application case, download and install *Insomnia* or a comparable application: https://insomnia.rest/download/

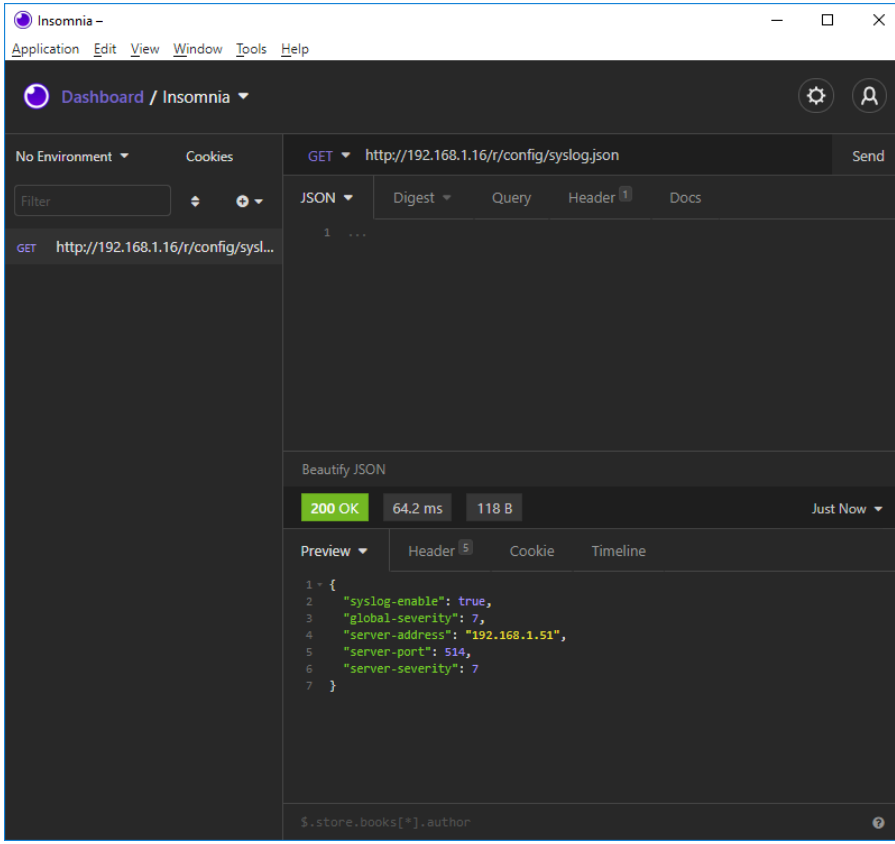
2. Configure Syslog:

POST: `[IP-address]/w/config/syslog.json`



3. Read Syslog configuration:

GET: `[IP-address]/r/config/syslog.json`



12.6 Network Time Protocol (NTP)

The NTP function is **only** applicable for the following UNITRONIC® ACCESS variant:

- MP08IOLA08DIO

The UNITRONIC® ACCESS multi-protocol variants provide an NTP client (version 3) which can connect with a configured NTP server and is able to synchronize the network time at a configurable interval.

NTP is a network protocol which uses UDP datagrams to send and receive timestamps in order to synchronize with a local clock. The NTP protocol RFC1305 is based on the Server-Client principle and exclusively supplies the synchronization with Coordinated Universal Time (UTC). (For more details on the used NTP standard, please refer to https://datatracker.ietf.org/doc/html/rfc1305.)

12.6.1 NTP configuration

In **delivery state**, the NTP client is **disabled**. The NTP client can be configured either using the Web interface or directly via a JSON object sent in an HTTP/HTTPS request. For more information see chapter NTP configuration - Quick start guide on page 134.

The configuration URL is:

`http://[ip-address]/w/config/ntpc.json`

The configuration can also read back as a JSON file:

`http://[ip-address]/r/config/ntpc.json`

The configuration is a JSON object. Each JSON member is a configuration element. The object must not contain all elements. Only the provided elements will be changed. The configuration changes apply only after a device restart.

The following configuration elements are available (default values in bold):

Element	Data type	Description	Example data
NTP client state	boolean	Master switch for the NTP client	true / **false**
Server address	string	IP address of the NTP server	192.168.1.50
Server port	integer	Port of the NTP server	**123**
Update interval	integer	Interval at which the client will connect with the configured NTP server (see table row "Server address"). **Note: This value is in seconds.**	1/2/10/**60**

Table 49: NTP configuration

NTP response:

The resulting response is a JSON object with a "status" field. Status should be "0" if no error occurred, and "-1" if there is an error.

In case of an error, the response contains an error array.

The error array contains an error object for each error occurred. The object consists of a field "Element" which names the configuration element that caused the error, and of a field "Message" for the error message.

Examples:

```
{"status": -1, "error": [{"Element": "ntpc-enable", "Message": "Boolean
 expected"}]}

{"status": 0}

{"status": -1, "error": [{"Element": "root", "Message": "Not a JSON
 object"}]}
```

12.6.2 NTP configuration - Quick start guide

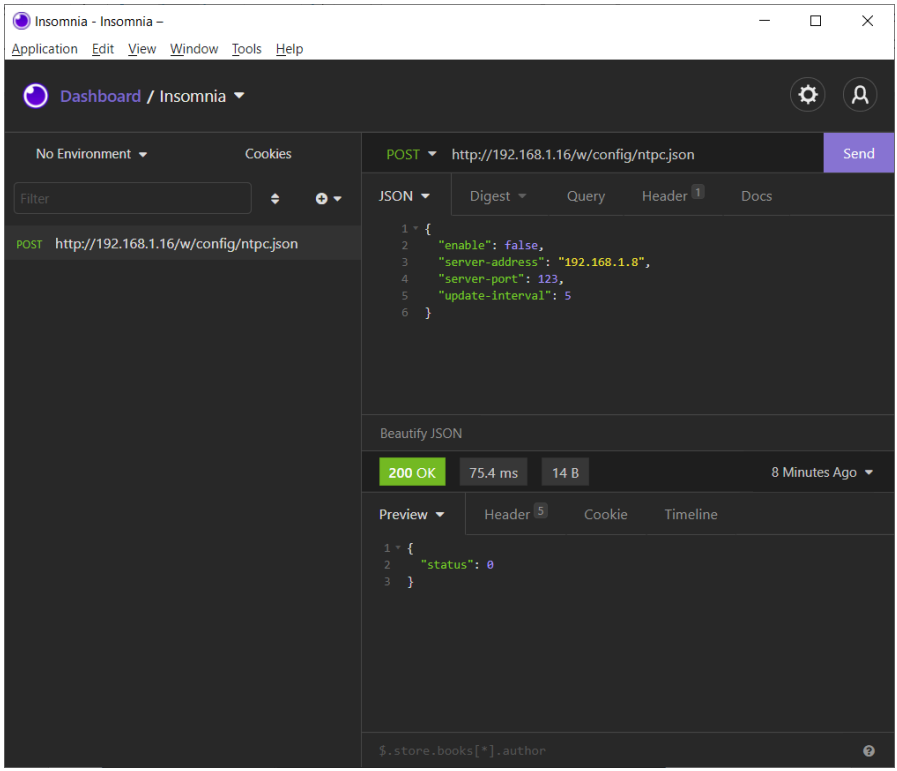
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

12.6.2.1 NTP configuration via JSON

1. Depending on your application case, download and install *Insomnia* or a comparable application: https://insomnia.rest/download/

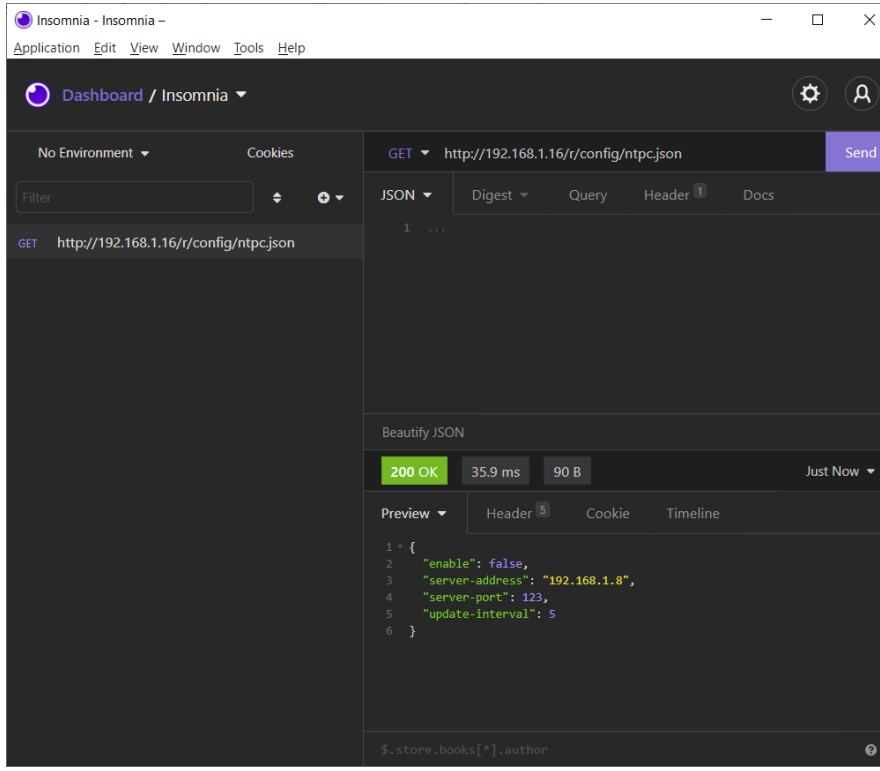
2. Configure NTP:

POST: `[IP-address]/w/config/ntpc.json`



3. Read NTP configuration:

GET: `[IP-address]/r/config/ntpc.json`



13 The integrated Web server

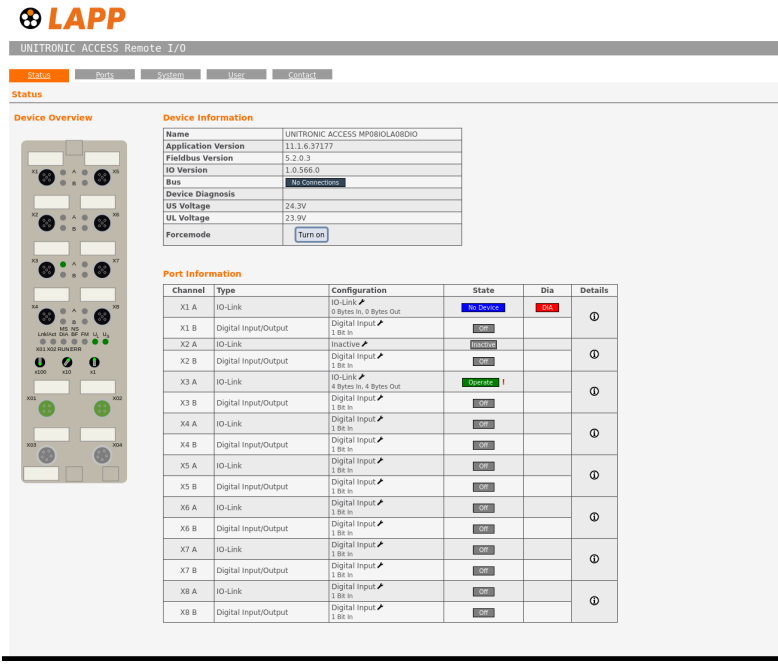
All device variants are equipped with an integrated Web server which makes functions for the device configuration and the display of status and diagnostic information available via a Web interface.

The Web interface provides an overview of the configuration and status of the device. It is also possible to use the Web interface to trigger a reboot, reset to the factory defaults, or perform a firmware update.

Enter "http://" or "https://" followed by the IP address, such as "http://192.168.1.5", in your Web browser’s address bar. If the status page of the device is not displayed, check your browser and firewall settings.

13.1 UNITRONIC® ACCESS MP08... variants

13.1.1 The Status page



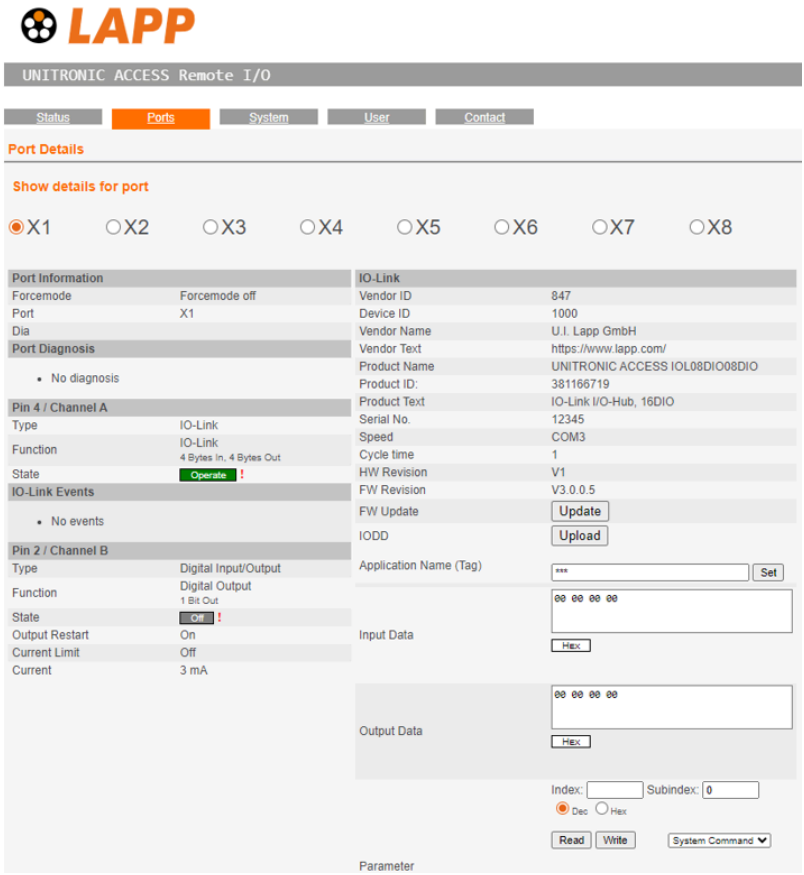
The status page provides a quick overview of the current state of the device.

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.

13.1.2 The Ports page



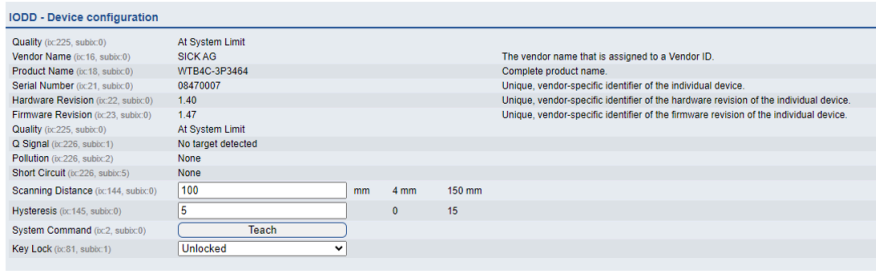
The page shows detailed port information. In the field **Port Diagnosis**, incoming and outgoing diagnostics are displayed as clear text. **Pin 2** and **Pin 4** contain information about the configuration and state of the port. For IO-Link ports, additional information relating to the connected sensor and the process data is displayed.

13.1.2.1 IODD upload

The button **UPLOAD** allows uploading an IODD file into the module, regardless of the device for which the IODD is designed.

The maximum number of IODDs is limited due to storage space. If there is no more space for a new IODD left, there will be a message of the detected error.

With the help of the IODD management page ("System" page), not used IODDs can be deleted. If there is already a matching IODD stored in the system for the connected IO-Link device, the button **CONFIGURE** is shown. By clicking this button, the page "IODD - Device configuration" will open, where the IO-Link device can be configured.



13.1.3 The System page



The System page shows the basic information for the module like Firmware version, Device information, Ethernet, Network and Fieldbus information.

IP Settings

Use this parameter to change the current IP address of the module.

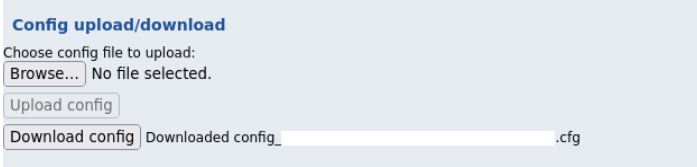
For PROFINET, this is only useful during commissioning. Normally, the PLC sets the IP address at start-up by detecting the PROFINET module via its device name.

13.1.3.1 License

This button opens a new window with Open Source Software information used in this product.

13.1.3.2 Config upload/download

With this feature, settings configured via the Web interface can be stored outside the I/O-Device (Download) for later Upload, e.g. after an I/O-Device change.



The following settings will be stored inside this file:

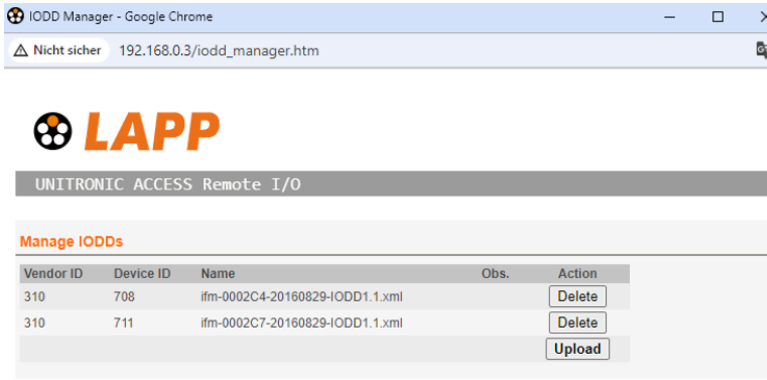
Scope	Type	Setting	Options	Details
Gateway		deviceID		To check device identity.
	iol	applicationSpecificTag		
	iol	functionTag		
	iol	locationTag		
		forcing		Enable/disable forcing
		channel_count		
		network configuration	ip	
			snMask	
			gw	
			source	1 - manual 2 - dhcp 3 - rotary 4 - dcp
Channel		index		channel index starting from 0
		channel configuration		0 - DIO 1 - IN 2 - OUT 3 - IOL 4 - AUX 5 - SAFIN 6 - SAFOUT
	iol	forced		
	iol	simulated		
	iol	force values		array
	iol	simulated		
	iol	sim values		array
	iol	validation	option	validation and backup
			vendorId	
			deviceId	
	digital	force		

Scope	Type	Setting	Options	Details
	digital	force value		
	digital	simulate		
	digital	sim value		
	digital	inputPolarity		
	digital	autorestart mode		
	digital	inputFilter100us		
	digital	currentLimit		
	digital	outputRestartMode		
	digital	failsafeMode		
	digital	surveillanceTimeouMs		
OPC UA		opcua	opcua-enable	
			port	
			anon-allowed	
			commands-allowed	
			force-allowed	
			reset-allowed	
			config-allowed	
	digital		dcu-allowed	
MQTT		mqtt	mqtt-enable	
			broker	
			login	
			password	
			port	
			base-topic	
			will-enable	
			will-topic	
			auto-publish	
			publish-interval	
			publish-identity	

Scope	Type	Setting	Options	Details
			publish-config	
			publish-status	
			publish-process	
	iol		publish-devices	
			commands-allowed	
			force-allowed	
			reset-allowed	
			config-allowed	
			qos	
SYSLOG		syslog	syslog-enable	
			global-severity	
			server-address	
			server-port	
			server-severity	
COAP		coap	enable	
			port	
NTP		ntpc	enable	
			server-address	
			server-port	
			update-interval	

13.1.3.3 IODD

The button **Manage IODDs** opens a new page for the IODD management on the I/O-Device. IODDs can be uploaded or deleted on this page, and all uploaded IODDs are listed here. For configuring connected IO-Link devices, open the related "Ports" page.



13.1.3.4 Restart device

The module initializes a software reset.

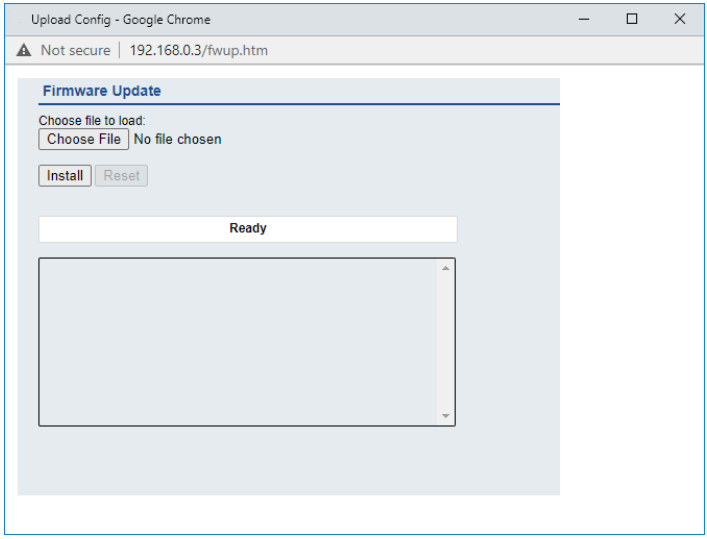
13.1.3.5 Reset configuration to factory defaults

The module restores to the default factory settings.

13.1.3.6 Firmware update

The module initializes a Firmware update.

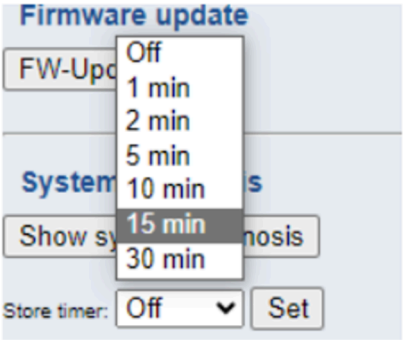
For a firmware update choose the **.ZIP* container available on our website or ask our support team. Afterwards follow the instructions shown on your screen.



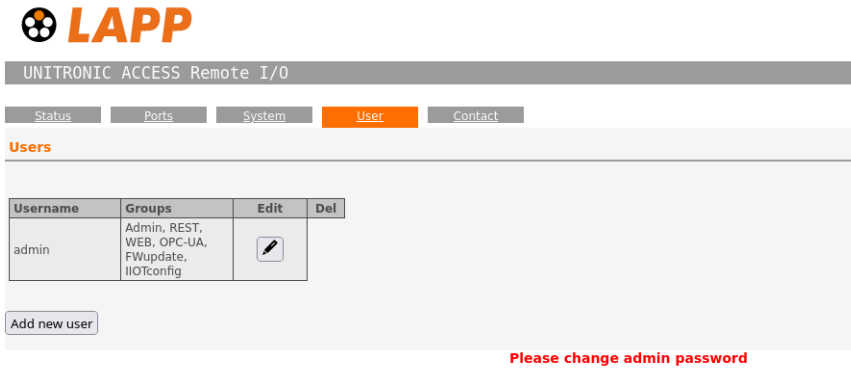
13.1.3.7 System diagnosis

All Syslog messages will be displayed in a ring buffer with 512 entries. By activation of the 'Store timer', the buffer content will be stored nonvolatile in the selected interval of 1, 2, 5, 10, 15 or 30 minutes.

The default value is 'Off' (no nonvolatile storage of system diagnosis ring buffer).



13.1.4 The User page



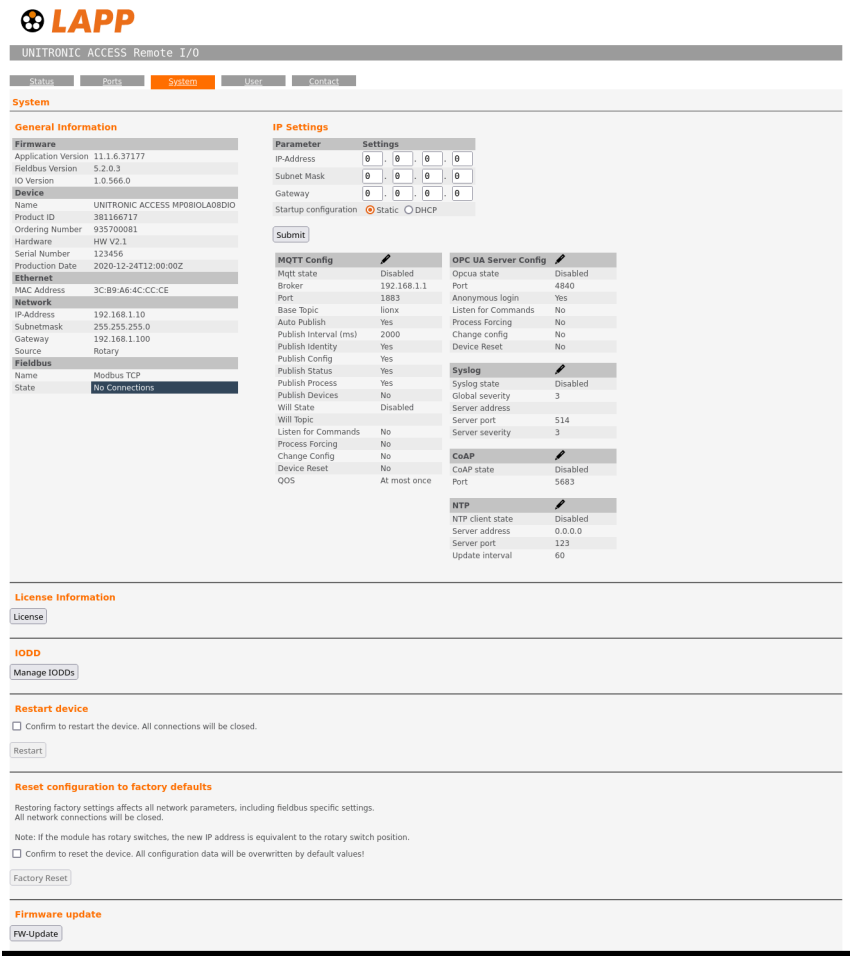
The User page provides the user management of the Web interface. New users with access rights "Admin" or "Write" can be added here. For security reasons please change the default admin password immediately after configuring the device.

Default user login data:

- User: admin
- Password: private

13.2 UNITRONIC® ACCESS single-protocol PN0.../EIP0.../EC0.../MTCP0.../CCL0... variants

13.2.1 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

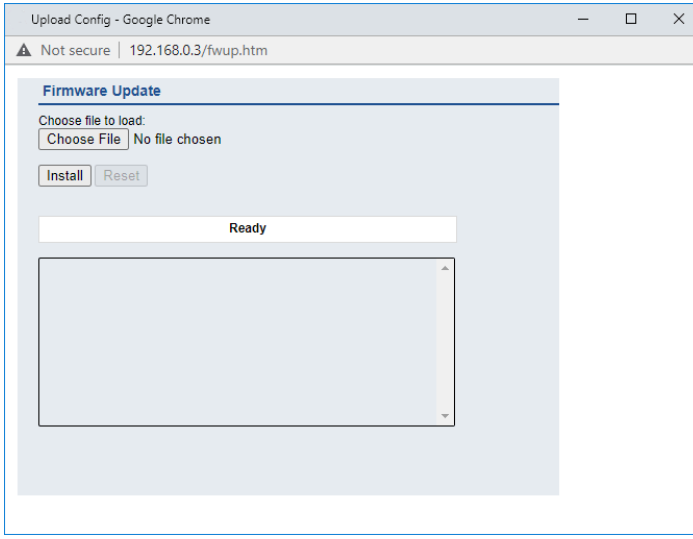
Use this parameter to change the current IP address of the module.

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.



14 IODD

IODD functions are **only** applicable for the following device variants:

- MP08IOLA08DIO

The **IO D**evice **D**escription (IODD) is a set of files formally describing an IO-Link Device. The IODD is created by the vendor and is mandatory for each IO-Link Device.

LAPP UNITRONIC® ACCESS IO-Link Masters with the "IODD on Module" functionality are ready to use IODDs in order to make the IO-Link Device configuration much more easier and the process data human readable in a better way. IODDs can be uploaded via the Web Interface and remanently stored on the IO-Link Master afterwards.

If a corresponding IO-Link Device is connected, the stored IODD is used to provide a user friendly configuration page, where all parameters of the device can be viewed and edited. Additionally, according to the IODD, the process data will also be formatted and displayed to the user.

14.1 IO-Link Device parameters and ISDU requests

Every IO-Link Device provides parameters that can be read and written via the special IO-Link service ISDU (**I**ndexed **S**ervice **D**ata **U**nit).

Every parameter is addressed by an index. Sub-indices are possible but optional. Some parameters (most of them read-only) are mandatory for IO-Link devices an can be found always on the same indices (See *Table B.8* in the *IO-Link Interface and System Specification*: https://io-link.com/share/Downloads/Package-2020/IOL-Interface-Spec_10002_V113_Jun19.pdf).

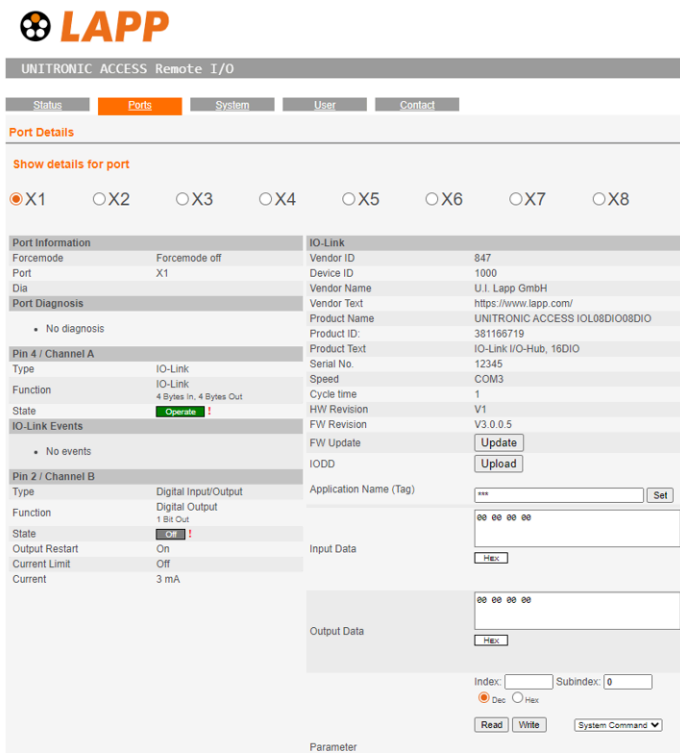
A vendor can use additional parameters and therefore more indices for their devices in order to provide additional configuration options. These vendor specific parameters can be described in an IODD. The "IODD on Module" feature of the UNITRONIC® ACCESS IO-Link Masters can read and parse this information out of an IODD and use it to provide the user viewing

and editing options for vendor specific parameters without any additional knowledge about the vendor specific device features.

14.2 Web GUI functionality

All of the "IODD on Module" features are accessible via the UNITRONIC® ACCESS Web interface.

14.2.1 Port Details page



The Port Details Page shows all information about the selected port. In the left column, all port and channel specific information is displayed. If the port is configured as IO-Link and there is an IO-Link Device connected, all IO-Link information for the connected device is displayed in the right column.

IODD buttons

The row called *IODD* provides access to the "IODD on Module" features. The button *UPLOAD* will let the user upload an IODD file into the module, regardless of the original device the IODD has been designed for.

The maximum number of IODDs is limited due to storage space. If there is no more space left for new IODDs, there will be an error message. In this case, navigate to the IODD Management page to delete IODDs which are no longer used.

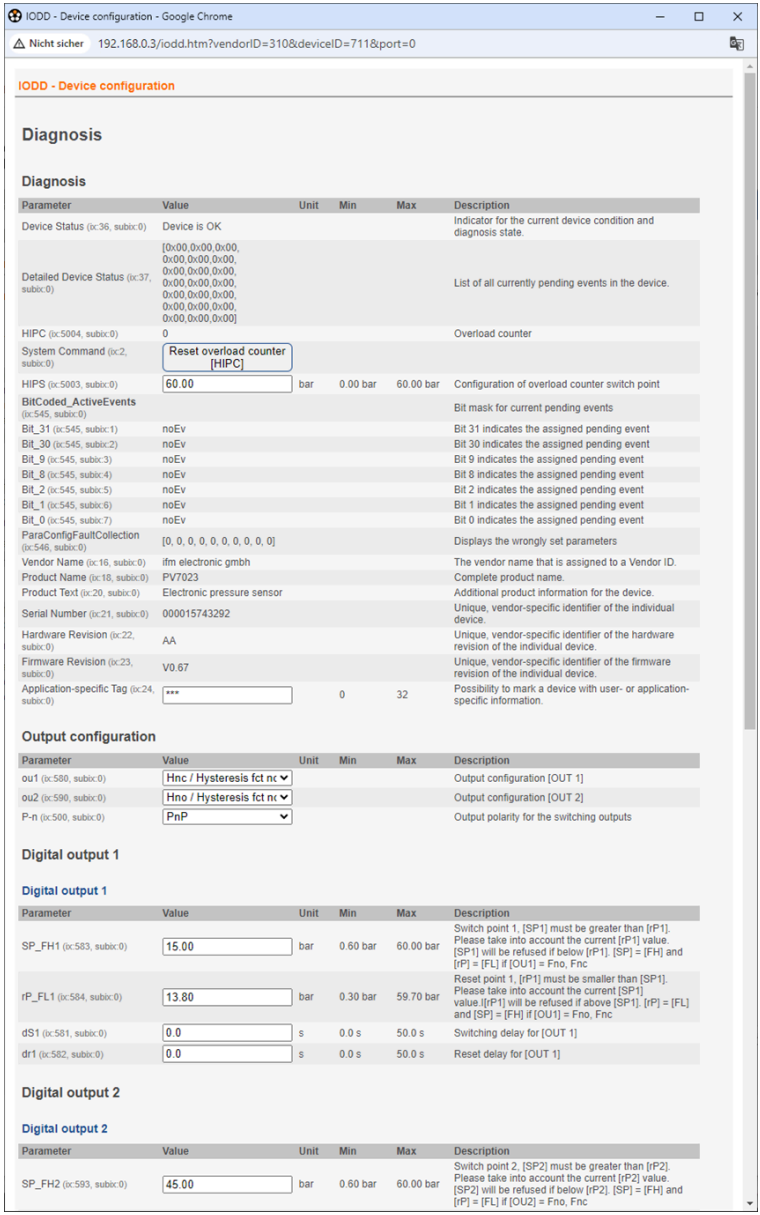
If there is a matching IODD for the currently connected device already stored in the system, the button *CONFIGURE* is shown in the interface. By clicking this button, the Parameter Page will open to configure the device.

Process data

For every connected IO-Link Device, raw process data for input and output direction (set of bytes) is on display.

If a matching IODD providing information about process data is already stored in the system, this data will also be displayed in a user-friendly format according to the IODD.

14.2.2 Parameters page



The parameters page "IODD – Device configuration" shows all parameters which are provided by the IODD of the device. That means the parameter set is variable and depends on the connected IO-Link Device.

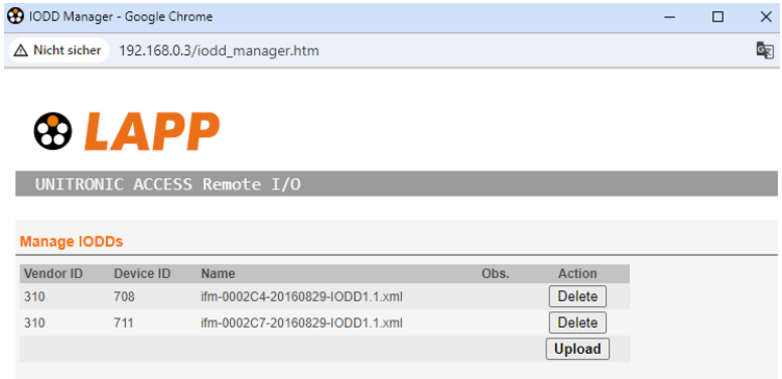
The stored IODD reads the parameter meta data, such as names, units, min/max values, descriptions etc. The values will be obtained directly from the connected device. For that reason it may take several seconds until the page is updated.

If not already saved into the browser, you will be asked for your credentials to continue. A valid user access with Web Interface group membership is needed in order to edit the device parameters. After the registration, enabled values can be changed. Disabled values cannot be changed and may be marked as "read-only" in the IODD. All values are directly written back to the device after any change.

Limitations

- Editing parameter values will directly change them inside the connected device. No parameter server action is triggered by that.
- There is a maximum size of the IODD in order to be uploaded into the system. This depends on several values, such as file size, parameter count, nesting levels etc.

14.2.3 IODD Management page



The IODD Management Page can be accessed via the System page displaying all IODDs that are currently stored in the system. All IODDs

matching connected devices are marked. On the IODD Management page, you can manually delete any IODD in the system.

15 Technical data

The following sections give an overview of the most important functional data needed to operate the device. For further information and detailed technical data, see the respective **Data Sheet** of your required product in the product specific download area on https://lapp.com.

15.1 General

Protection class (Only applies if the connectors are screwed together or if protective caps are used.)[2]	IP65 IP67 IP69K	
Ambient temperature (during operation and storage)	MP08...	-40 °C .. +70 °C (-40 °F .. +158 °F)
	PN08IOLA.../EIP08IOLA.../EC08IOLA.../ MTCP08IOLA.../CCL08IOLA...	-20 °C .. +60 °C (-4 °F .. +140 °F)
Weight	UNITRONIC® ACCESS 60 mm	approx. 500 gr. (17.6 oz)
Ambient moisture	Max. 98% RH (For UL applications: Max. 80% RH)	
Housing material	Die-cast zinc	
Surface finish	Frosted nickel	
Flammability class	UL 94 (IEC 61010)	
Vibration resistance (oscillation) DIN EN 60068-2-6 (2008-11)	15 g / 5-500 Hz	
Shock resistance DIN EN 60068-2-27 (2010-02)	50 g / 11 ms +/- X,Y,Z	
Fastening torques	M4 fixing screws	1 Nm
	M4 ground connection	1 Nm
	M12 connector	0.5 Nm
Permitted cables	Ethernet cables according to IEEE 802.3, min. CAT 5 (shielded) Max. length of 100 m, not routed out of facility (= local network)	

Table 50: General information

2 Not under UL investigation.

15.2 CC-Link IE Field Basic protocol

Protocol	CC-Link IE Field Basic
Update cycle	1 ms
Transmission rate	100 Mbit/s, full duplex
Transmission procedure Autonegotiation	100BASE-TX supported
Product type	0x001F IO-Link Master
Product code	MP08IOLA08DIO, 381166717 CCL08IOLA08DIO, 381166716
Supported Ethernet protocols	ICMP ARP HTTP / HTTPS SNMP
Switch functionality	Integrated
CC-Link IE Field Basic interface Connections Autocrossing	2 M12 sockets, 4-pin, D-coded (see pin assignments) 2 M12 Hybrid male/female, 8-pin supported
Electrically isolated Ethernet ports -> FE	2000 V DC

Table 51: CC-Link IE Field Basic protocol

15.3 Power supply of the module electronics/ sensors

Port X03, X04	M12-L-coded Power, connector/socket, 5-pole Pin 1 / Pin 3		
Nominal voltage U_S	24 V DC (SELV/PELV)		
Current U_S	Max. 16 A		
Voltage range	21 .. 30 V DC		
Power consumption of module electronics	Typically 160 mA (+/-20 % at U_S nominal voltage)		
Power supply interruption	Max. 10 ms		
Voltage ripple U_S	Max. 5 %		
Current consumption sensor system (L+ / Pin 1)	MP08...	Port X1 .. X8 (Pin 1)	max. 4 A per port (at $T_{ambient}$ = 30° C)
	PN08IOLA.../ EIP08IOLA.../ EC08IOLA.../ MTCP08IOLA.../ CCL08IOLA...	Port X1 .. X8 (Pin 1)	max. 2 A per port (at $T_{ambient}$ = 30° C)
Voltage level of the sensor power supply	Min. (U_S – 1.5 V)		
Short circuit/overload protection of sensor supply	Yes, per port		
Reverse polarity protection	Yes		
Operational indicator (U_S)	LED green:	18 V (+/- 1 V) < U_S	
	LED red:	U_S < 18 V (+/- 1 V)	

Table 52: Information on the power supply of the module electronics/ sensors

15.4 Power supply of the actuators

15.4.1 IO-Link Class A devices (U_L)

Nominal voltage U_L	24 V DC (SELV/PELV)
Voltage range	18 .. 30 V DC
Current U_L	Max. 16 A
Voltage ripple U_L	Max. 5 %
Reverse polarity protection	Yes
Operational indicator (U_L)	LED green: 18 V (+/- 1 V) < U_L LED red: U_L < 18 V (+/- 1 V) or U_L > 30 V (+/- 1 V) * if "Report U_L supply voltage fault" is enabled.
Port X03, X04	M12_L-coded Power, connector/socket, 5-pole Pin 2 / Pin 4

Table 53: Information on the power supply of the actuators

15.4.2 IO-Link Class A/B devices (U_{AUX})

Nominal voltage U_{AUX}	24 V DC (SELV/PELV)
Voltage range	18 .. 30 V DC
Current U_{AUX}	Max. 16 A
Voltage ripple U_{AUX}	Max. 5 %
Reverse polarity protection	Yes
Electric isolation $U_S \leftrightarrow U_{AUX}$	500 V
Operational indicator (U_{AUX})	LED green: 18 V (+/- 1 V) < U_{AUX} LED red: U_{AUX} < 18 V (+/- 1 V) or U_{AUX} > 30 V (+/- 1 V) * if "Report U_{AUX} supply voltage fault" is enabled.
Port X03, X04	M12_L-coded Power, connector/socket, 5-pole Pin 2 / Pin 4

Table 54: Information on the power supply of the actuators

15.5 I/O ports Channel A (Pin 4)

MP08...	Port X1 .. X8	Class A	IOL, DI, DO	M12 socket, 5-pin, Pin 4
PN08IOLA.../EIP08IOLA.../ EC08IOLA.../ MTCP08IOLA.../ CCL08IOLA...	Port X1 .. X8	Class A	IOL, DI, DO	

Table 55: IO-Link Master ports: Functional overview for Ch. A (Pin 4)

15.5.1 Configured as digital input, Ch. A (Pin 4)

Input connection	MP08... PN08IOLA.../EIP08IOLA.../EC08IOLA.../ MTCP08IOLA.../CCL08IOLA...		Type 1 as per IEC 61131-2
Nominal input voltage	24 V DC		
Input current	Typically 3 mA		
Channel type	Normally open, p-switching		
Number of digital inputs	MP08... PN08IOLA.../EIP08IOLA.../ EC08IOLA.../MTCP08IOLA.../ CCL08IOLA...	X1 .. X8	8
Status indicator	yellow LED		
Diagnostic indicator	red LED per channel		

Table 56: I/O ports Ch. A (Pin 4) configured as digital inputs

15.5.2 Configured as digital output, Ch. A (Pin 4)

Attention: For variant MP08IOLA08DIO, the digital outputs of Channel A is **supplied by the U_L power** when parameterized to "High-Side Switch" mode.

Attention: For variant PN08IOLA08DIO, the digital outputs of Channel A is **supplied by the U_S power**.

Output type	normally open, p-switching (parameterized to "High-Side Switch" mode)	
Nominal output voltage per channel Signal status “1” Signal status “0”	 min. (U_S -1 V) **or** min. (U_L -1 V) depending on the device variant max. 2 V	
Max. output current per device	MP08...	9 A (power supplied via U_L)
Max. output current per device	PN08IOLA.../EIP08IOLA.../ EC08IOLA.../MTCP08IOLA.../ CCL08IOLA...	4 A (power supplied via U_S)
Max. output current per channel[3]	MP08... (X1 .. X8)	2 A (power supplied via U_S)
	PN08IOLA.../EIP08IOLA.../ EC08IOLA.../MTCP08IOLA.../ CCL08IOLA... (X1 .. X8)	0.5 A (power supplied via U_S) 0.25 A for **UL applications**
Short-circuit/overload protected	yes/yes	
Behavior in case of short circuit or overload	deactivation with automatic power-on (parameterized)	
Number of digital outputs	MP08... (X1 .. X8) PN08IOLA.../EIP08IOLA.../ EC08IOLA.../MTCP08IOLA.../ CCL08IOLA... (X1 .. X8)	8
Status indicator	yellow LED per output	
Diagnostic indicator	red LED per channel	

Table 57: I/O ports Ch. A (Pin 4) configured as digital outputs

3 Max. 2.0 A per channel; for every port pair X1/X2, X3/X4, X5/X6, X7/X8 max. 6.5 A (for **UL applications** max. 5.0 A); for the whole port group X1 .. X8 max. 9.0 A in total (with derating).

15.5.3 Configured as IO-Link port in COM mode, Ch. A

IO-Link Master specification	v1.1.3, IEC 61131-9	
Communication rates	4.8 kbaud (COM 1) 38.4 kbaud (COM 2) 230.4 kbaud (COM 3)	
Line lengths in the IO-Link Device	max. 20 m	
Number of IO-Link ports	MP08... (X1 .. X8) PN08IOLA.../EIP08IOLA.../EC08IOLA.../ MTCP08IOLA.../CCL08IOLA... (X1 .. X8)	8
Min. IO-Link cycle time	400 µs	

Table 58: Configured as IO-Link port in COM mode

15.6 I/O ports Channel B (Pin 2)

MP08...	Port X1 .. X8	Class A	DI, DO	M12 socket, 5-pin, Pin 2
PN08IOLA.../EIP08IOLA.../ EC08IOLA.../ MTCP08IOLA.../ CCL08IOLA...	Port X1 .. X8	Class A	DI	

Table 59: IO-Link Master ports: Functional overview for Ch. B (Pin 2)

15.6.1 Configured as a digital input, Ch. B (Pin 2)

Input connection	MP08... PN08IOLA.../EIP08IOLA.../EC08IOLA.../ MTCP08IOLA.../CCL08IOLA...		Type 1 as per IEC 61131-2
Nominal input voltage	24 V DC		
Input current	Typically 3 mA		
Channel type	Normally open, p-switching		
Number of digital inputs	MP08...	X1 .. X8	8
Number of digital inputs	PN08IOLA.../EIP08IOLA.../ EC08IOLA.../MTCP08IOLA.../ CCL08IOLA...	X1 .. X8	8
Status indicator	white LED		
Diagnostic indicator	red LED per channel		

Table 60: I/O ports Ch. B (Pin 2) configured as digital inputs

15.6.2 Configured as a digital output, Ch. B (Pin 2)

Attention: For variant MP08IOLA08DIO, the digital outputs of Channel B is **supplied by the U_L power**.

Attention: For variant PN08IOLA08DIO, the digital outputs of Channel B is **supplied by the U_S power**.

Output type	normally open, p-switching	
Nominal output voltage per channel Signal status “1” Signal status “0”	min. (U_S -1 V) **or** min. (U_L -1 V) **or** min. (U_{AUX} -1 V) depending on the device variant max. 2 V	
Max. output current per device	MP08...	9 A (power supplied via U_L)
Max. output current per device	PN08IOLA.../ EIP08IOLA.../ EC08IOLA.../ MTCP08IOLA.../ CCL08IOLA...	4 A (power supplied via U_S)
Max. output current per channel [4]	MP08...	2 A (power supplied via U_S)
Max. output current per channel [4]	PN08IOLA.../ EIP08IOLA.../ EC08IOLA.../ MTCP08IOLA.../ CCL08IOLA...	0 A (no outputs on Ch. B)
Short-circuit/overload protected	yes/yes	
Behavior in case of short circuit or overload	deactivation with automatic power-on (parameterized)	
Number of digital outputs	MP08...	8
Number of digital outputs	PN08IOLA.../ EIP08IOLA.../ EC08IOLA.../ MTCP08IOLA.../ CCL08IOLA...	–
Status indicator	white LED per output	
Diagnostic indicator	red LED per channel	

Table 61: I/O ports Ch. B (Pin 2) configured as digital outputs

4 For Class A devices: Max. 2.0 A per channel; for every port pair X1/X2, X3/X4, X5/X6, X7/X8 max. 6.5 A (for **UL applications** max. 5.0 A); for the whole port group X1 .. X8 max. 9.0 A in total (with derating).

15.7 LEDs

LED	Color	Description
U_L/U_{AUX}	Green	Auxiliary sensor/actuator voltage OK 18 V (+/- 1 V) < U_L/U_{AUX} < 30 V (+/- 1 V)
	Red*	Auxiliary sensor/actuator voltage LOW U_L/U_{AUX} < 18 V (+/- 1 V) or U_L/U_{AUX} > 30 V (+/- 1 V) * if "Report U_L/U_{AUX} supply voltage fault" is enabled.
	OFF	None of the above conditions.
U_S	Green	System/sensor voltage OK 18 V (+/- 1 V) < U_S < 30 V (+/- 1 V)
	Red	System/sensor voltage LOW U_S < 18 V (+/- 1 V) or U_S > 30 V (+/- 1 V)
	Red flashing	Device performs a factory reset (position of rotary encoding switches: 9-7-9)
	OFF	None of the above conditions.
X1 .. X8 A	Green	IO-Link COM Mode: IO-Link communication exists.
	Green flashing	IO-Link COM Mode: No IO-Link communication.
	Yellow	Standard-I/O Mode: Status of digital input or output on C/Q (pin 4) line "on".
	OFF	None of the above conditions
X1 .. X8 B	White	Status of digital input or digital output on pin 2 line "on".
	Red	Short circuit on pin 4 and pin 2 line. / All modes: Overload or short circuit on L+ (pin 1) line / communication error
	OFF	None of the above conditions.
P1 Lnk/Act P2 Lnk/Act	Green	Ethernet connection to another subscriber exists. Link detected.
	Yellow flashing	Data exchange with another subscriber.
	OFF	No connection to another subscriber. No link, no data exchange.

LED	Color	Description
BF	Red	Bus fault. No configuration, no or slow physical connection.
	Red flashing at 2 Hz	Link exists but no communication link to the CC-Link IE controller.
	OFF	CC-Link IE controller has established an active connection to the device.
DIA	Red	CC-Link IE module diagnostic alarm active.
	Red flashing at 1 Hz	Watchdog time-out; fail safe mode is active.
	Red flashing at 2 Hz, 3 sec	DCP signal service is initiated via the bus.
	Red double flash	Firmware update
	OFF	None of the above conditions.
MS	Green	Device is ready for operation.
	Green flashing	Device is ready but not configured yet.
	Red	Serious error that cannot be resolved.
	Red flashing	Minor error that can be resolved Example: An incorrect or contradictory configuration is classified as a minor error.
	Flashing alternately:	The device is performing a self-test.
	Red / Green	
	OFF	The device is switched off.

LED	Color	Description
NS	Green	Connected: The device has at least one connection.
	Green flashing	No connection: The device has no connection. IP address exists.
	Red	Duplicate IP address: The device has detected that the assigned IP address is already being used by another device.
	Red flashing	Connection has exceeded time limit or connection interrupted.
	Flashing alternately: Red / Green	The device is performing a self-test.
	OFF	The device is switched off or has not been assigned an IP address.

Table 62: Information on the LED colors

15.8 Data transfer times

The following tables give an overview of the internal data transfer times of the UNITRONIC® ACCESS IO-Link Master with a connected IO-Link Device as digital I/O extension (LAPP article IOL08DIO08DIO 16DIO Hub with a minimum cycle time of 1 ms).

There are three measured data direction values for each use case:

- **PLC to DO:** Transfer of a changed PLC output data to IO-Link Device digital output.
- **DI to PLC:** Transfer of a changed digital input signal on IO-Link Device to PLC.
- **Round-trip time (RTT):** Transfer of a changed PLC output data to IO-Link Device digital output. The digital output is connected to an IO-Link Device digital input. Transfer of the changed digital input signal on IO-Link Device to PLC. RTT = [PLC to DO] + [DI to PLC].

The measured values are taken from the ethernet data transmission line. The values are therefore without PLC processing times and PLC cycle time.

The configurable digital input filter value on IOL08DIO08DIO was set to "off" (0 ms).

For calculation of user specific data transfer and round-trip times of possible input filters, PLC processing and cycles times must be taken into calculation.

The measured values are valid for a maximum of 48 bytes of IO-Link data for the IO-Link Master in each direction (Input/Output).

Use case 1:

IO-Link Master configuration with enabled Web interface and *disabled* IIoT protocols

Data direction	Data transfer time in ms		
	Minimum	Average	Maximum
PLC to DO	3.7	6.0	7.7
DI to PLC	1.1	3.0	4.3
RTT	6.1	8.9	11.1

Use case 2:

IO-Link Master configuration with enabled Web interface and *enabled* IIoT protocols

Data direction	Data transfer time in ms		
	Minimum	Average	Maximum
PLC to DO	7.7	10.0	13.4
DI to PLC	3.3	4.4	5.6
RTT	12.1	14.3	17.0

16 Accessories

In order to get access to various types of accessories, please visit our Web page:

https://www.lapp.com