Manual

PROFINET

UNITRONIC ACCESS Digital-I/O multi-protocol: MP08DIO08DIO (16 × Input/Output)

Technical Support
https://www.lapp.com

Contents

1 About this manual

1.1 General information

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

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

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

U.I. Lapp GmbH

Schulze-Delitzsch-Straße 25
D-70565 Stuttgart
Germany

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	09/2024	

Table 1: Overview of manual revisions

2 Safety instructions

2.1 Intended use

The products described in this manual are decentralized I/O Devices on an Industrial Ethernet Network.

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

The modules fulfill the requirements of the EMC guidelines (89/336/EEC, 93/68/EEC and 93/44/EEC) and the low voltage guideline (73/23/EEC).

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

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

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

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

Only install cables and accessories that fulfill the requirements and regulations for safety, electromagnetic compatibility, and, where applicable, telecommunication end devices, as well as the specification information. Information on which cables and accessories are permitted for the installation can be obtained from 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 this manual are authorized to install and maintain the devices described. These are persons who

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

Only 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 manual, can result in serious personal injury or damage to equipment.

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

3 Designations and synonyms

AOI	Add-On Instruction
API	Application Programming Interface
BF	Bus Fault LED
Big Endian	Data format with High-B on first place (PROFINET)
BUI	Back-Up Inconsistency (EIP diagnostics)
CC	CC-Link IE Field
Ch. A	Channel A (Pin 4) of I/O port
Ch. B	Channel B (Pin 2) of I/O port
CIP	Common Industrial Protocol (media independent protocol)
CoAP	Constrained Application Protocol
CSP+	Control & Communication System Profile Plus
DCP	Discovery and Configuration Protocol
DevCom	Device Comunicating (EIP diagnostics)
DevErr	Device Error (EIP diagnostics)
DI	Digital Input
DIA	Diagnostic LED
DO	Digital Output
DIO	Digital Input/Output
DTO	Device Temperature Overrun (EIP diagnostics)
DTU	Device Temperature Underrun (EIP diagnostics)
DUT	Device under test
EIP	EtherNet/IP
ERP	Enterprise Resource Planning system
ETH	ETHERNET
FE	Functional Earth
FME	Force Mode Enabled (EIP diagnostics)
FSU	Fast Start-Up

GSDML	General Station Description Markup Language
High-B	High-Byte
ICT	Invalid Cycle Time (EIP diagnostics)
IIoT	Industrial Internet of Things
ILE	Input process data Length Error (EIP diagnostics)
IME	Internal Module Error (EIP diagnostics)
I/O	Input / Output
I/O port	X1 .. X8
I/O port pin 2	Channel B of X1 .. X8
I/O port pin 4 (C/Q)	Channel A of X1 .. X8
IVE	IO-Link port Validation Error (EIP diagnostics)
I&M	Identification & Maintenance
JSON	JavaScript Object Notation (platform independent data format)
L+	I/O port pin 1, sensor power supply
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
OLE	Output process data Length Error (EIP diagnostics)
OPC UA	Open Platform Communications Unified Architecture (platform independent, service-oriented architecture)

PLC	Programmable Logic Controller
PN	PROFINET
PWR	Power
REST	REpresentational State Transfer
RFC	Request for Comments
RPI	Requested Packet Interval
RWr	Word data input as seen from the master station (CC-Link)
RWw	Word data output as seen from the master station (CC-Link)
RX	Bit data input as seen from the master station (CC-Link)
RY	Bit data output as seen from the master station (CC-Link)
SCA	Short Circuit Actuator/U_L/U_{AUX} (EIP diagnostics)
SCS	Short Circuit Sensor (EIP diagnostics)
SLMP	Seamless Message Protocol
SNMP	Simple Network Management Protocol
SP	Single Protocol (PROFINET, EtherNet/IP, EtherCAT®, Modbus TCP or CC-Link IE Field Basic)
SPE	Startup Parameterization Error (EIP diagnostics)
U_{AUX}	$U_{Auxiliary}$, supply voltage for the load circuit (Actuator supply on Class B ports)
UDP	User Datagram Protocol
UDT	User-Defined Data Types
UINT8	Byte in PLC (IB, QB)
UINT16	Unsigned integer with 16 bits or word in PLC (IW, QW)
U_L	U_{Load}, supply voltage for the load circuit (Actuator supply on Class A)
UL	Underwriters Laboratories Inc. (certification company)
UTC	Coordinated Universal Time (Temps Universel Coordonné)

Table 2: Designations and synonyms

4 System description

4.1 Device variants

The following Digital I/O device variants are available in the UNITRONIC® ACCESS family:

Article number	Product designation	Description	I/O port functionality
381166718	MP08DIO08DIO	UNITRONIC® ACCESS M12-60 mm, I/O Device Multi-protocol (PN, EIP, EC, MB, CC) Security	16 x Input/Output universal

Table 3: Overview of UNITRONIC® ACCESS Digital-I/O variants

4.2 I/O port overview

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

UNITRONIC® ACCESS 16DIO ports

Device variant	Port	Pin 1 U_S	Pin 4 / Ch. A (In/Out)		Pin 2 / Ch. B (In/Out)	
MP08DIO08DIO	**Info:**	–	Type 3	Supply by U_L	Type 3	Supply by U_L
	X8:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)
	X7:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)
	X6:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)
	X5:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)
	X4:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)
	X3:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)
	X2:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)
	X1:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)

Table 4: Port configuration of MP08DIO08DIO variant

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 I/O Device supports PROFINET RT (real-time). This allows the transmission of time sensitive process data between network components in real-time communication.

PROFINET specification V2.44, Conformance Class C

The UNITRONIC® ACCESS I/O Device complies with the PROFINET specification V2.44 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 a UNITRONIC® ACCESS I/O Device 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 500 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 Masters 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).

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

SNMPv1

The SNMPv1 protocol (according PROFINET standard V2.35) 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.

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 (integration for PROFINET, Edition 2):

- I&M0 … I&M3 for the interface module (access slot, sub-slot 0x8000)

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 Integrated Web server

Network parameter display

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

Displaying diagnostics

View diagnostics via the integrated Web server.

User management

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

5.3 Security features

Firmware signature

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

Syslog

The 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.

5.4 Other features

Interface protection

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

For more details, see section Port assignments on page 26.

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 Digital-I/O multi-protocol variants

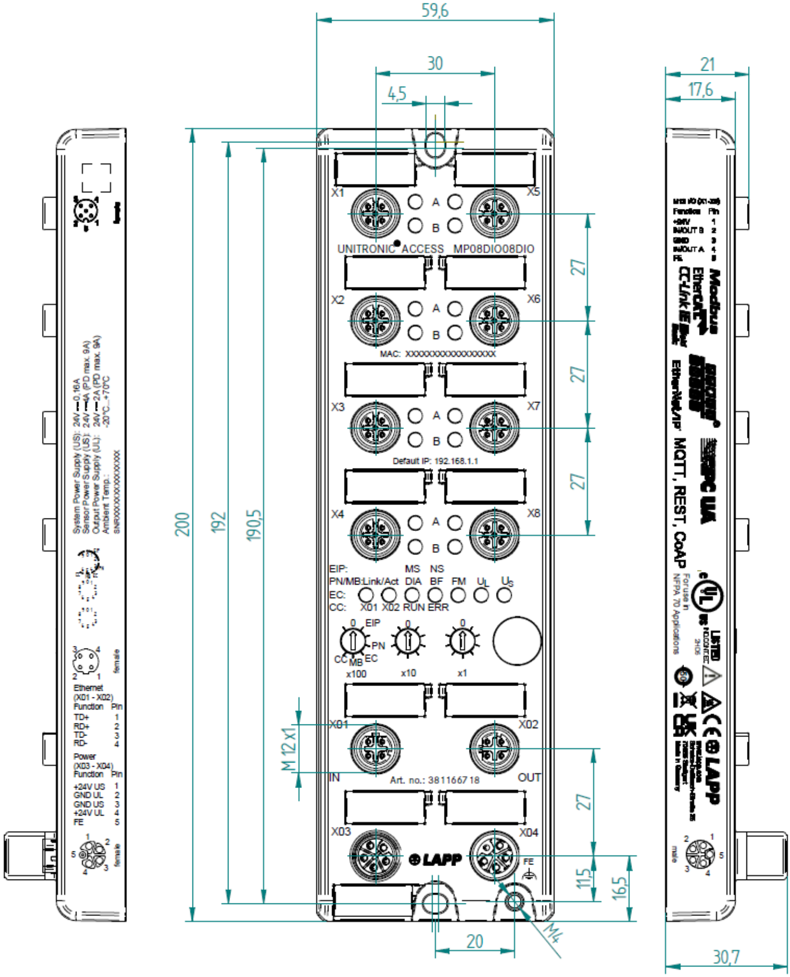


Figure 1: MP08DIO08DIO

6.2.2 Notifications

Attention:

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

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

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

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

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

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

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

6.3 Port assignments

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

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

Color coding: green

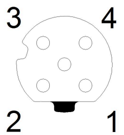


Figure 2: Schematic drawing, ports X01, X02

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

Table 5: Assignment of ports X01, X02

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

6.3.2 Power supply with M12 power L-coded

Color coding: gray

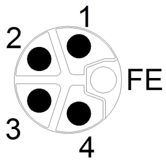


Figure 3: Schematic diagram of the M12 L-coding (connector X03 for Power In)

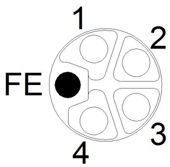


Figure 4: Schematic diagram of the M12 L-coding (socket X04 for Power Out)

Power supply	Pin	Signal	Function
	1	U_S (+24 V)	Sensor/system power supply
	2	GND_U_L	Ground/reference potential U_L
	3	GND_U_S	Ground/reference potential U_S
	4	U_L (+24 V)	Load supply Actuator supply
	5	FE	Functional ground

Table 6: Pin assignments ports X03 and X04

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

6.3.3 I/O ports as M12 sockets

Color coding: black

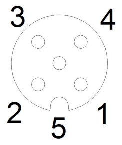


Figure 5: Schematic drawing I/O port as M12 socket

6.3.3.1 I/O ports

MP08DIO08DIO	Pin	Signal	Function
16DIO X1 .. X8	1	+24 V	power supply +24 V
	2	IN/OUT	Ch. B: Digital input or digital output
	3	GND	Ground/reference potential
	4	IN/OUT	Ch. A: Digital input or digital output
	5	FE	Functional ground

Table 7: Pin assignments I/O ports

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:	MP08DIO08DIO
Vendor ID:	0x18F
Device ID:	0x0608

7.4 Setting the rotary encoding switches

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

- MP08DIO08DIO

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.

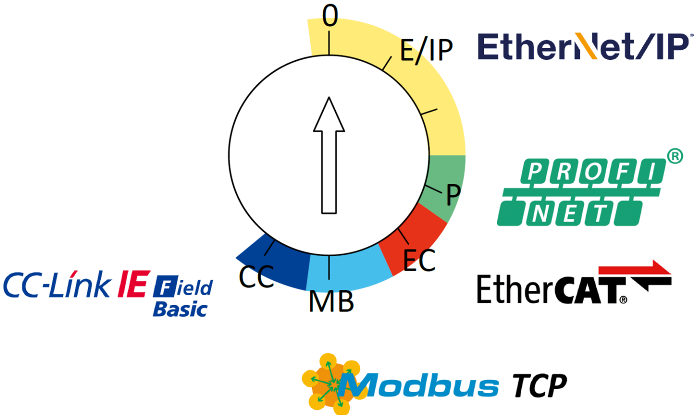
After selecting a protocol followed by a restart of the device, the new protocol is initialized. This can take up to 15 seconds. During this time the device is not usable and the LED indicators are out of function. When the protocol change is complete, the LED indicators return to normal operation and 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 Digital I/O devices 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 I/O Device in different environments.

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

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



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

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

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

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

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

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

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

To change the protocol, carry out a factory reset. In this way you restore the factory settings of the respective device. How you perform the factory reset for your device is described in chapter Factory reset on page 34.

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

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

	x100	x10	x1
Factory Reset	9	7	9

Follow the steps from section Setting the rotary encoding switches on page 31 again to select a new protocol.

For performing a factory reset via software configuration, see chapter OPC UA configuration on page 105 and the configuration section.

7.5 SNMPv1

The PROFINET IO device 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.

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:

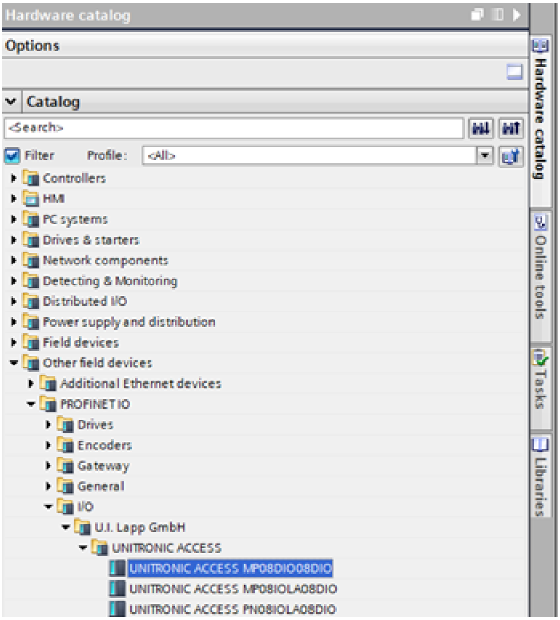


Figure 6: TIA Portal® Hardware catalog

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

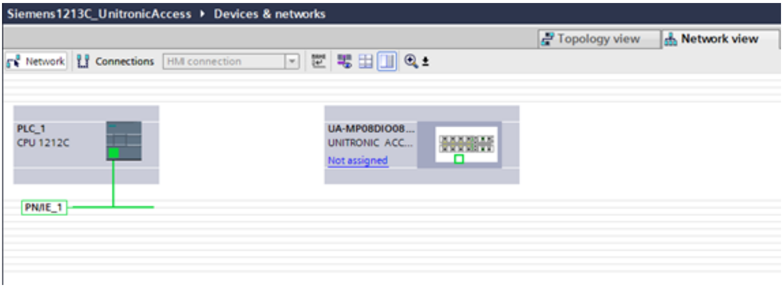


Figure 7: Network view

4. Assign the device to the PROFINET network:

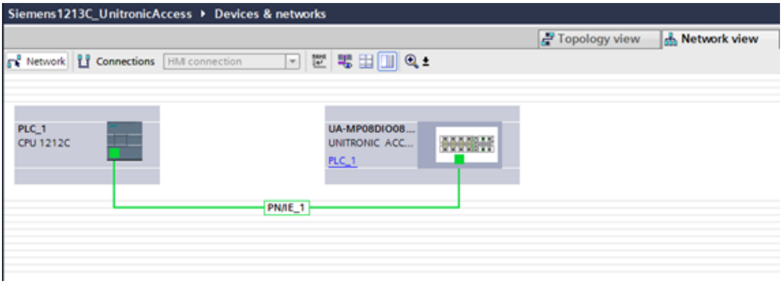


Figure 8: Assign device

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

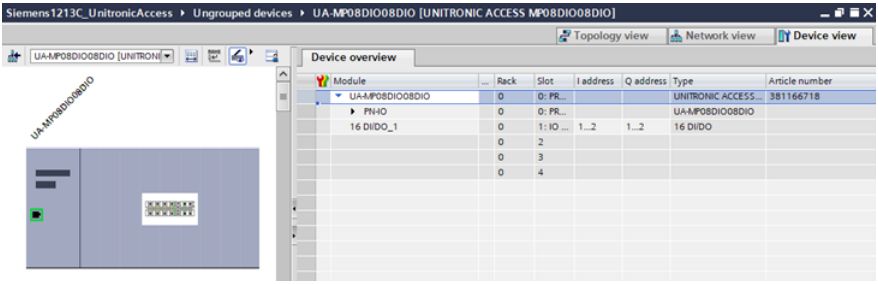


Figure 9: Device configuration

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**:

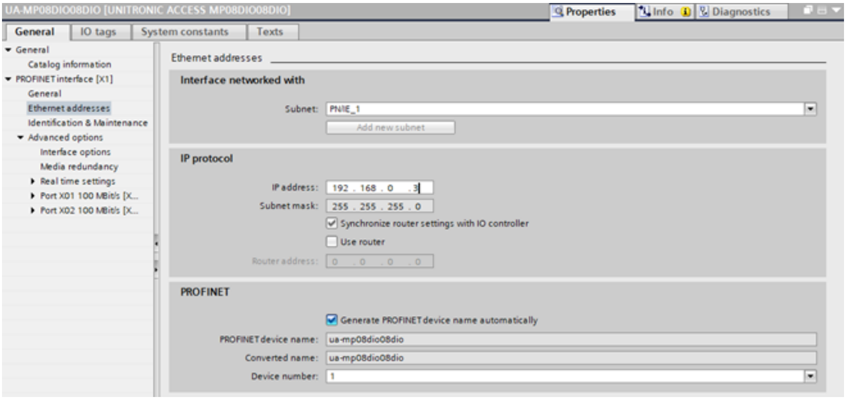


Figure 10: ETHERNET addresses

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:

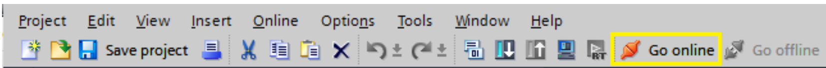


Figure 11: Go online

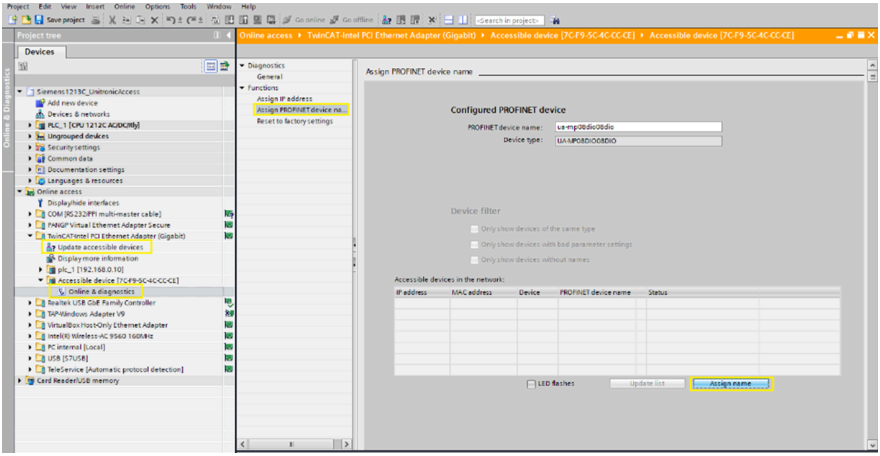


Figure 12: Online mode

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

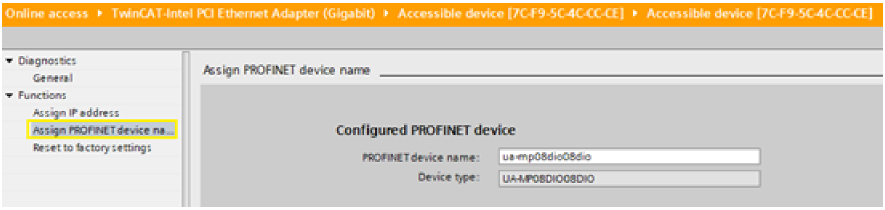


Figure 13: Assign device name

8.2 Configuring the I/O ports

For device MP08DIO08DIO, all I/O channels are pre-configured by default as 16 DI/DO.

This means you can attach a sensor or an actuator to each I/O channel without additional configuration of the channel direction (input or output). When you attach a sensor, do not activate the appropriate digital output via PLC.

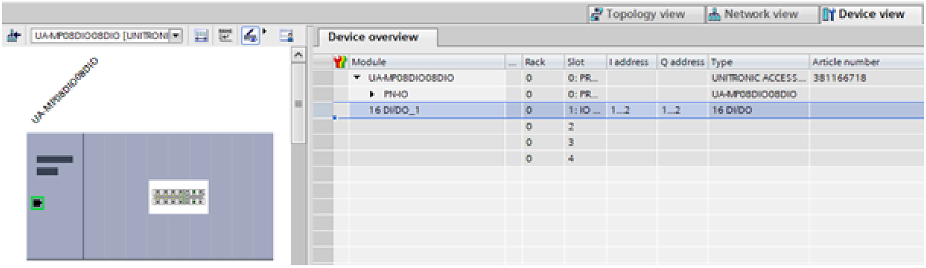


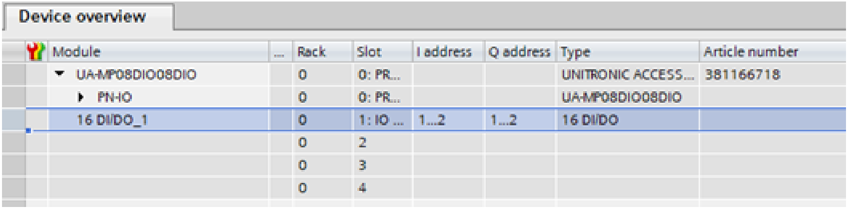
Figure 14: Channels pre-configuration

There are additional pre-configured channel setups available in the *Catalog* view. By removing the 16 DI/DO "Slot 1", you can set e.g. all I/O channels to 16 DO when choosing "Module 16 DO" from the *Catalog*.

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

8.2.1 Deleting the I/O configuration

1. To delete the current I/O configuration, select the respective slot in the *Device overview*:

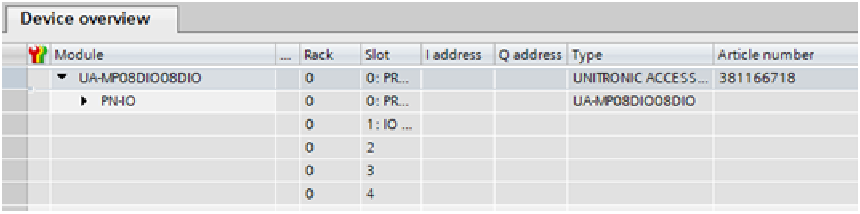


Device overview

Module	...	Rack	Slot	I address	Q address	Type	Article number
UA-MP08DIO08DIO		0	0: PR...			UNITRONIC ACCESS...	381166718
PN-IO		0	0: PR...			UA-MP08DIO08DIO	
16 DI/DO_1		0	1: IO ...	1...2	1...2	16 DI/DO	
		0	2				
		0	3				
		0	4				

Figure 15: Device overview

2. Right click on the slot and select option *Delete* in the appearing menu:



Device overview

Module	...	Rack	Slot	I address	Q address	Type	Article number
UA-MP08DIO08DIO		0	0: PR...			UNITRONIC ACCESS...	381166718
PN-IO		0	0: PR...			UA-MP08DIO08DIO	
		0	1: IO ...				
		0	2				
		0	3				
		0	4				

Figure 16: Empty I/O Slot 1

8.2.2 Changing the I/O configuration

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

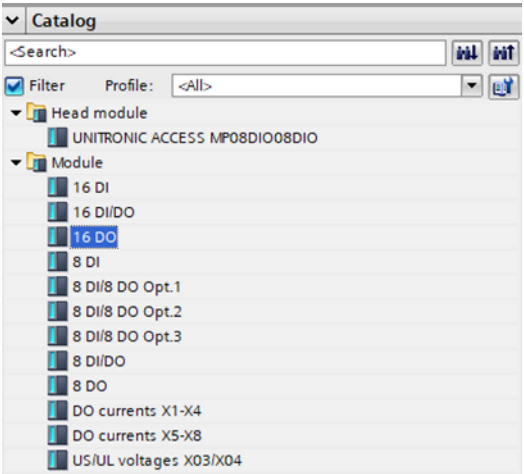
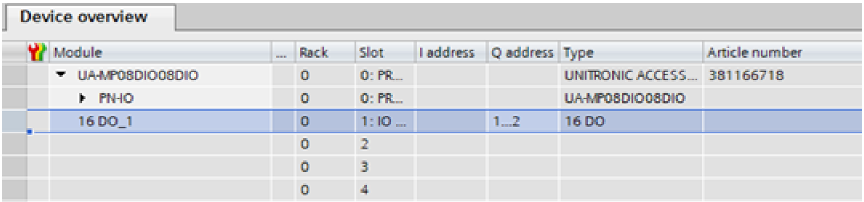


Figure 17: I/O channel configuration

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

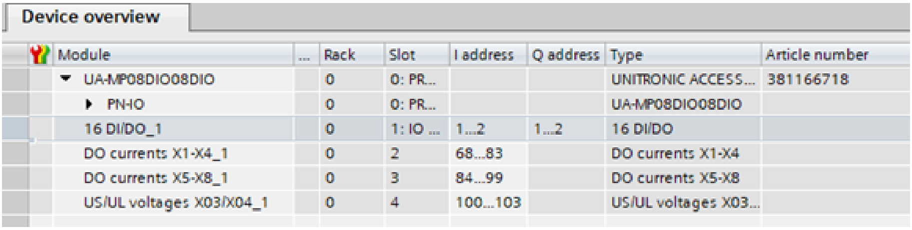


There are three additional slots (2 .. 4) for optional output current and voltage measurements:

Device overview

Module	...	Rack	Slot	I address	Q address	Type	Article number
UA-MP08DIO08DIO		0	0: PR...			UNITRONIC ACCESS...	381166718
PN-IO		0	0: PR...			UA-MP08DIO08DIO	
16 DO_1		0	1: IO ...		1...2	16 DO	
DO currents X1-X4_1		0	2	68...83		DO currents X1-X4	
DO currents X5-X8_1		0	3	84...99		DO currents X5-X8	
US/UL voltages X03/X04_1		0	4	100...103		US/UL voltages X03...	

8.3 Parameterization of the Status/Control Module



Device overview

Module	...	Rack	Slot	I address	Q address	Type	Article number
UA-MP08DIO08DIO		0	0: PR...			UNITRONIC ACCESS...	381166718
PN-IO		0	0: PR...			UA-MP08DIO08DIO	
16 DI/DO_1		0	1: IO ...	1...2	1...2	16 DI/DO	
DO currents X1-X4_1		0	2	68...83		DO currents X1-X4	
DO currents X5-X8_1		0	3	84...99		DO currents X5-X8	
US/UL voltages X03/X04_1		0	4	100...103		US/UL voltages X03...	

Figure 18: Status/Control Module

Parameters of the 16 DI/DO device variant:

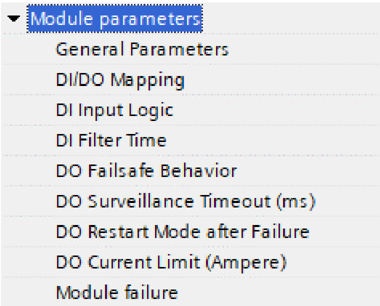


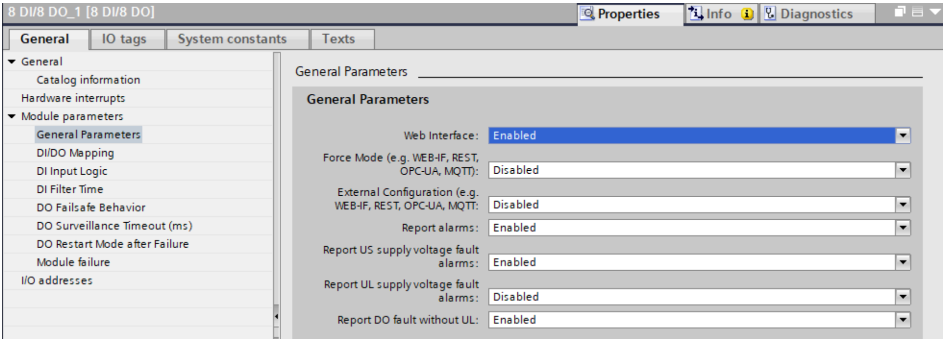
Figure 19: Parameters of the 16 DI/DO device variant

For variant MP08DIO08DIO, the parameter sub-sets can be different in dependency of the chosen I/O configuration, e.g. "8DI" instead of "16 DI/DO".

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

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

8.3.1 General Parameters



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). 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

Report Alarms

This is a global switch for enabling or disabling all PROFINET alarms.

Default: Enabled

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.

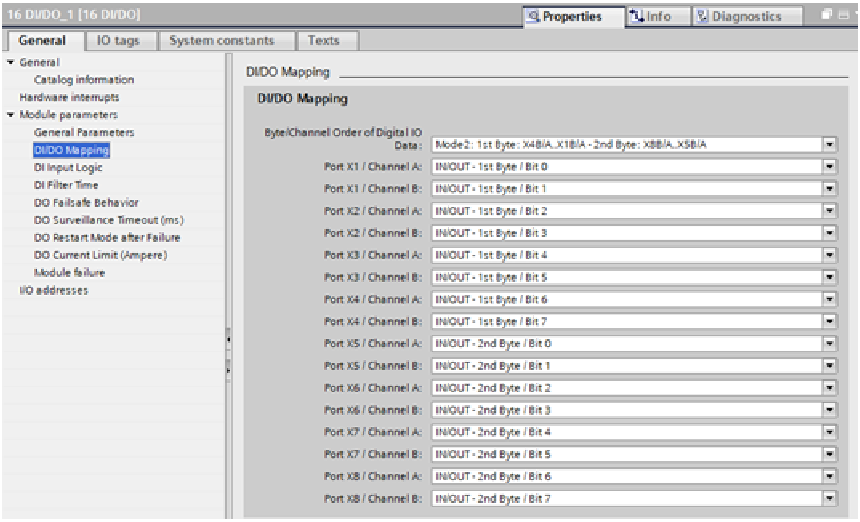
Report DO fault without U_L

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

8.3.2 DI/DO Mapping



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.

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 Slot 1 I/O data. It should be noticed that duplicate assignments are not possible here. If faulty parameterization is detected in the UNITRONIC® ACCESS device, 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 device.

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

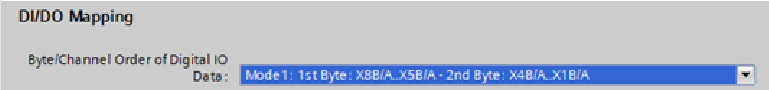
Key

1st Byte = low address byte in a Siemens PLC

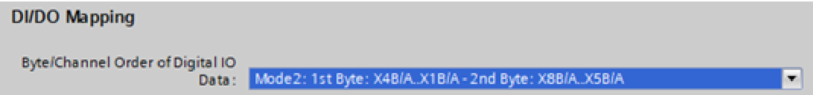
2nd Byte = high address byte in a Siemens PLC

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

Mode 1:



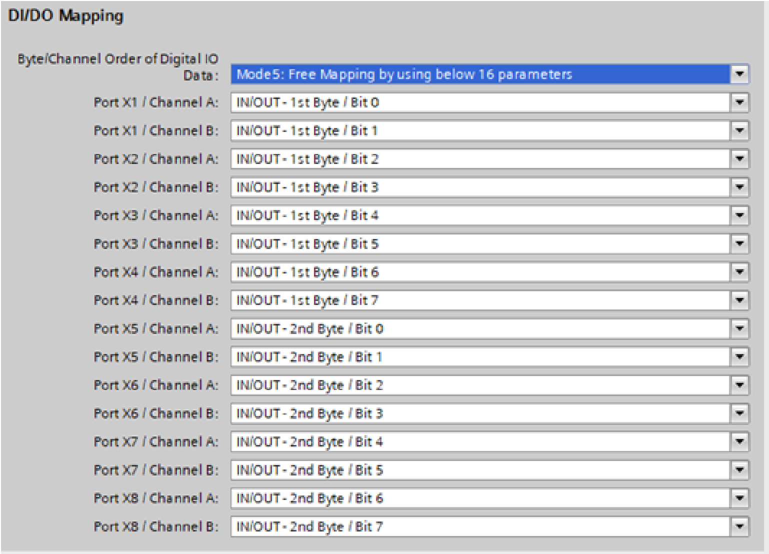
Mode 2:



Mode 3:

Mode 4:

Mode 5:

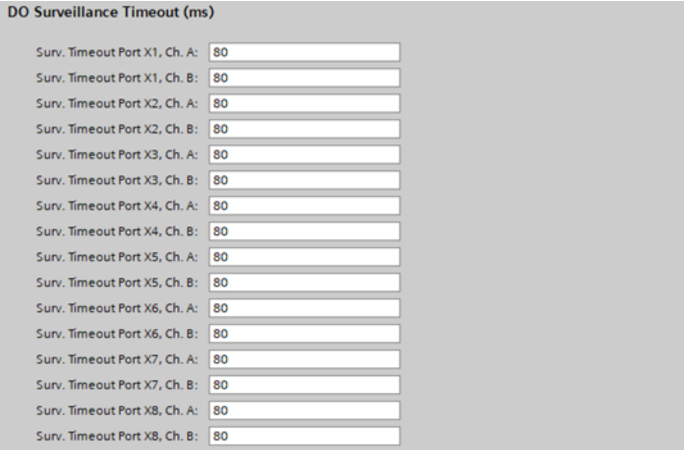


For detailed I/O mapping refer to chapter Process data assignment on page 66.

8.3.3 DO Surveillance Timeout (ms)

For channels configured as digital output, the firmware of the modules allows you 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.

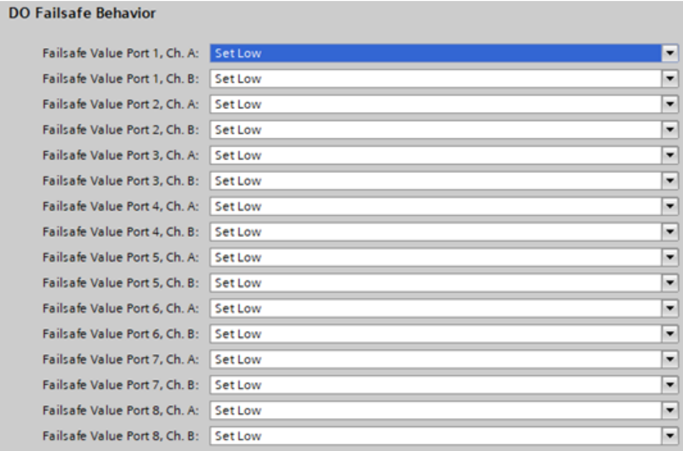


The *DO Surveillance-Timeout (ms)* parameter can be set from 0 to 255 ms. When an output channel is in static state, i.e., when the channel is permanently switched on or off, the typical filter value (not changeable) is 5 ms before a diagnostic message will be generated in case of a detected output error.

Default: 80 ms

8.3.4 DO Failsafe Behavior

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.



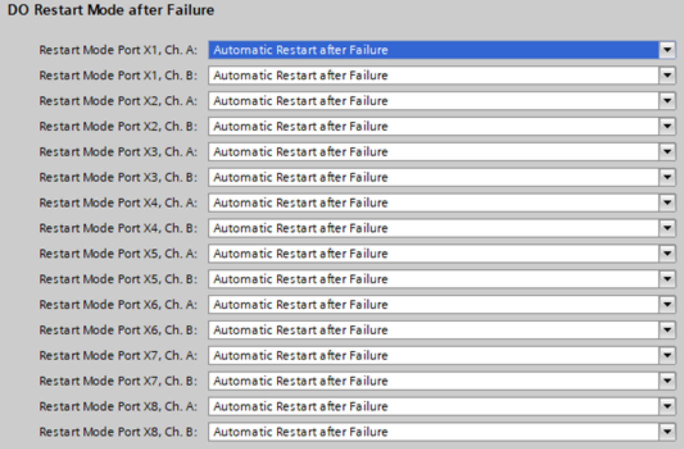
The following options can be selected:

- 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

8.3.5 DO Restart Mode after Failure

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. 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.

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

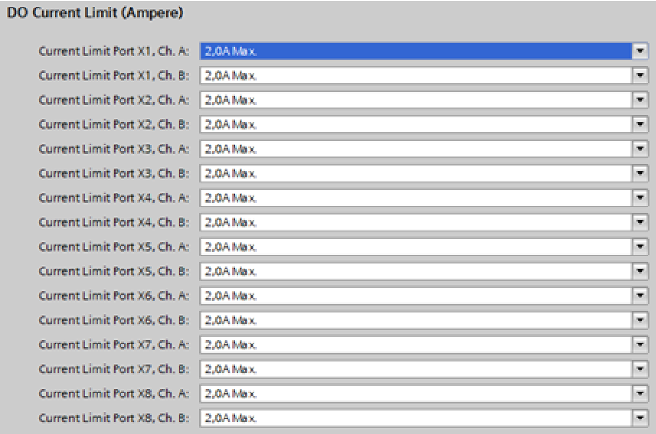
Default: Automatic Restart after Failure

8.3.6 DO Current Limit

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

- The following values are available: 0.5 A; 1.0 A; 1.5 A; 2,0 A; 2.0 A Max.

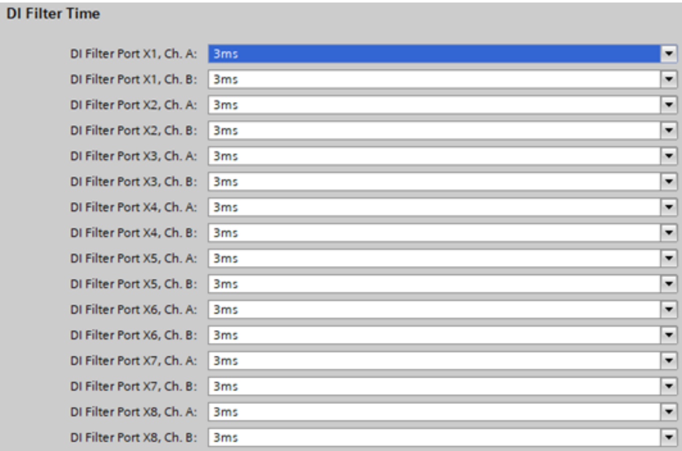
This means that the level for actuator overload diagnostic can be managed by this selection. *2.0 A Max.* means, that current limitation is **not** active and the maximum output current for this output is available.



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

8.3.7 DI Filter Time

With this parameter, the filter time of the digital input can be defined.



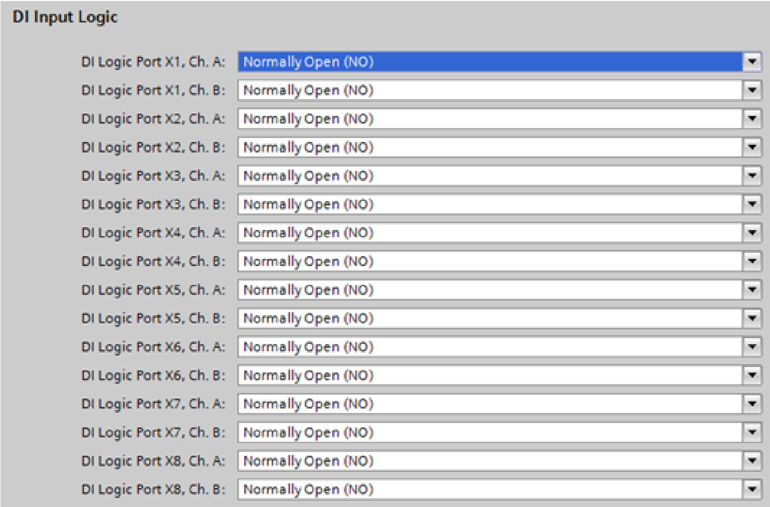
The following options are available:

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

Default: 3 ms

8.3.8 DI Input 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

8.4 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.

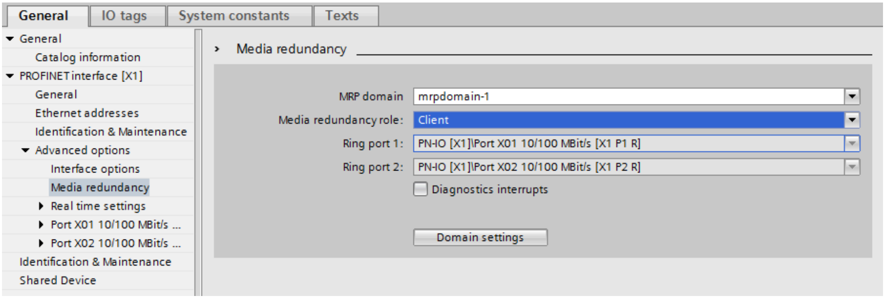


Figure 20: Example of setting up an MRP redundancy client in TIA Portal®

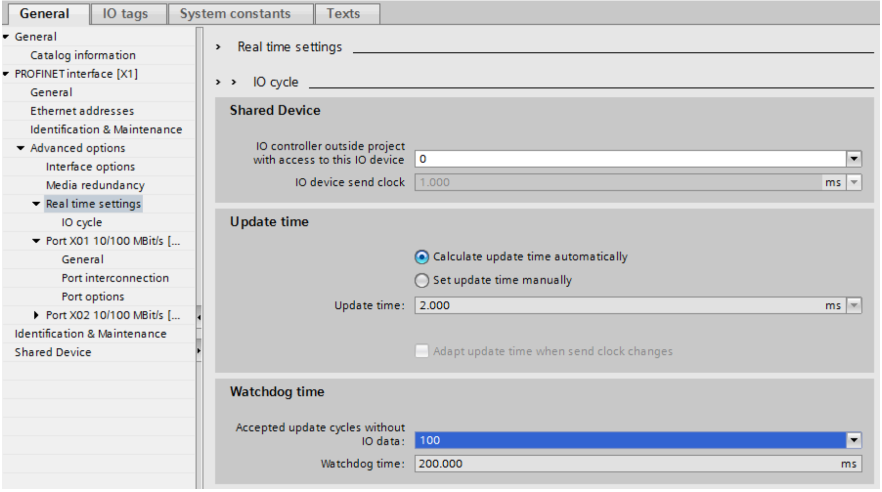


Figure 21: Example of setting up the Watchdog time monitoring in TIA Portal® for using MRP

8.5 Identification & maintenance (I&M)

The PROFINET IO device 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.

8.5.1 Supported I&M features

8.5.1.1 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:

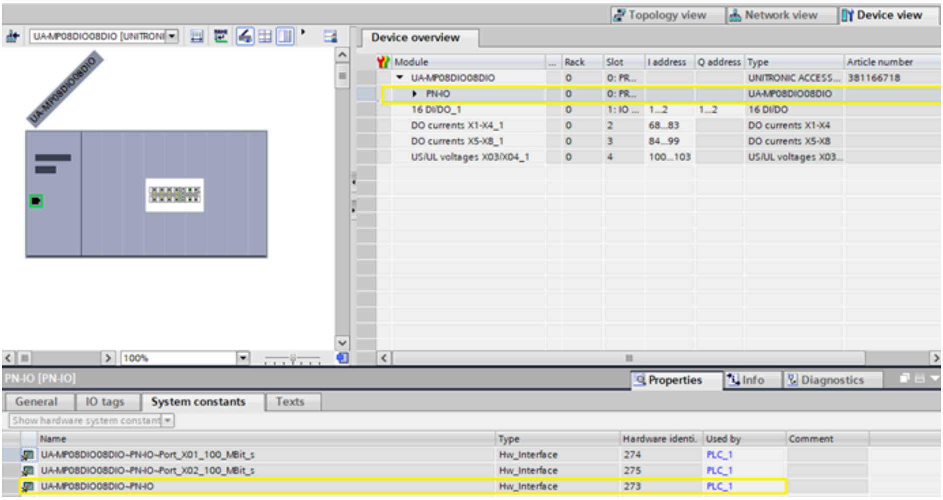


Figure 22: TIA Portal® hardware identifier of PROFINET interface for I&M 0-3 RDREC/WRREC

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[2]
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 the PROFINET IO device (e.g., device name or IP address)
PROFILE_ID	2	Read	0xF600 (Generic device)
PROFILE_SPECIFIC_TYPE	2	Read	0x0003 (I/O modules)
IM_VERSION	2	Read	0x0101 (I&M Version 1.1)
IM_SUPPORTED	2	Read	0x000E (I&M 1 .. 3 is supported)

Table 9: 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 10: I&M 1 (Slot 0: PROFINET Interface X1, Index 0xAFF1)

2 The serial number in the I&M data differs from the printed serial number on the housing. Printed serial number on housing: 9 characters article number + 9 characters ongoing number I&M0 serial number: 9 characters ongoing number (same last 9 characters as printed serial number on housing)

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 11: 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 12: I&M 3 (Slot 0: PROFINET Interface X1, Index 0xAFF3)

8.5.2 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 13: Data set with BlockHeader and I&M Record on page 60 shows the structure of a data set.

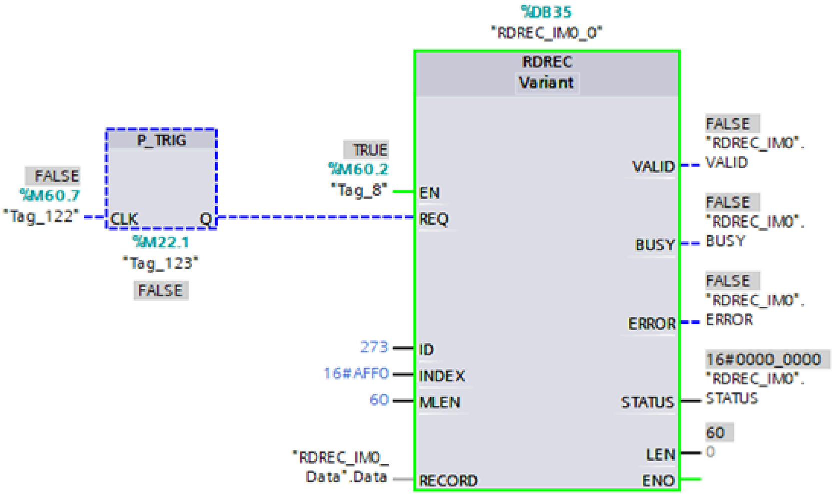
- For reading I&M 0..3, the RDREC block must be configured with `LEN = 6 Byte Block Header + I&M data length`.

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	BlockHeader
BlockLength	2	Word	I&M 0: 0x0038 I&M 1: 0x0038 I&M 2: 0x0012 I&M 3: 0x0038	
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	Byte		I&M Record

Table 13: Data set with BlockHeader and I&M Record

8.5.2.1 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. The return parameters show the length of the I&M data received and contain a status or error message.



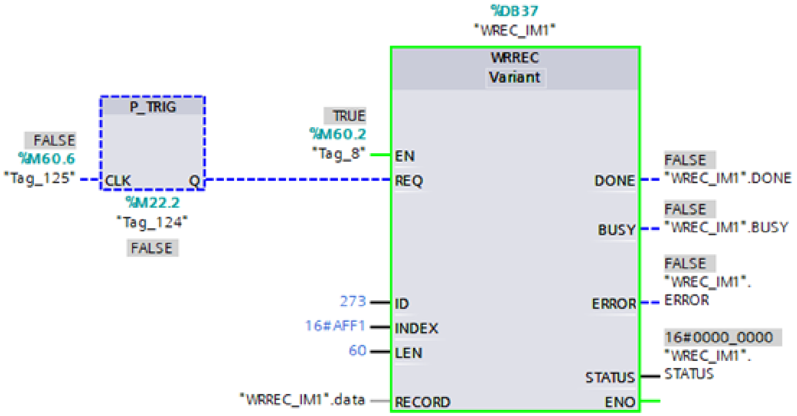
RDREC_IM0_Data

	Name	Data type	Start value	Monitor value	Retain	Comment
1	Static					
2	Data	Array[0..254] of Byte				
3	Data[0]	Byte	16#0	16#00		BlockType High: I&M0 = 0x0020
4	Data[1]	Byte	16#0	16#20		Block Type Low: I&M0 = 0x0020
5	Data[2]	Byte	16#0	16#00		BlockLength High: I&M0 = 0x0038
6	Data[3]	Byte	16#0	16#38		BlockLengthn Low: I&M0 = 0x0038
7	Data[4]	Byte	16#0	16#01		BlockVersion High: 1
8	Data[5]	Byte	16#0	16#00		BlockVersion Low: 0
9	Data[6]	Byte	16#0	16#01		Data: Vendor ID
10	Data[7]	Byte	16#0	16#8F		Data: Vendor ID
11	Data[8]	Byte	16#0	16#33		Data: Order ID 1
12	Data[9]	Byte	16#0	16#38		Data: Order ID
13	Data[10]	Byte	16#0	16#31		Data: Order ID
14	Data[11]	Byte	16#0	16#31		Data: Order ID
15	Data[12]	Byte	16#0	16#36		Data: Order ID
16	Data[13]	Byte	16#0	16#36		Data: Order ID
17	Data[14]	Byte	16#0	16#37		Data: Order ID
18	Data[15]	Byte	16#0	16#31		Data: Order ID
19	Data[16]	Byte	16#0	16#38		Data: Order ID
20	Data[17]	Byte	16#0	16#20		Data: Order ID
21	Data[18]	Byte	16#0	16#20		Data: Order ID
22	Data[19]	Byte	16#0	16#20		Data: Order ID
23	Data[20]	Byte	16#0	16#20		Data: Order ID

Figure 23: Read example I&M0 of PROFINET IO device

8.5.2.2 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.



WRREC_IM1

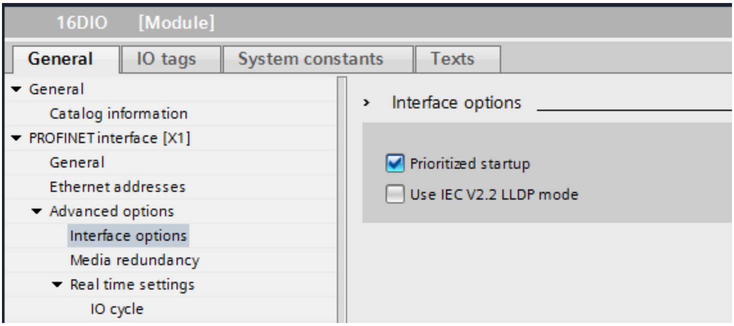
	Name	Data type	Start value	Monitor value	Retain	Comment
1	▼ Static					
2	▼ data	Array[0..254] of Byte				
3	data[0]	Byte	16#00	16#00		BlockType High I&M1 = 0x00
4	data[1]	Byte	16#21	16#21		BlockType Low I&M1 = 0x21
5	data[2]	Byte	16#00	16#00		BlockLength High 0x00 for I&M1
6	data[3]	Byte	16#38	16#38		BlockLength High 0x38 for I&M1
7	data[4]	Byte	16#01	16#01		Block Version High: 1
8	data[5]	Byte	16#00	16#00		Block Version Low: 0
9	data[6]	Byte	16#61	16#61		Data: 'a'
10	data[7]	Byte	16#62	16#62		Data: 'b'
11	data[8]	Byte	16#63	16#63		Data: 'c'
12	data[9]	Byte	16#64	16#64		Data: 'd'
13	data[10]	Byte	16#0	16#00		
14	data[11]	Byte	16#0	16#00		
15	data[12]	Byte	16#0	16#00		
16	data[13]	Byte	16#0	16#00		
17	data[14]	Byte	16#0	16#00		
18	data[15]	Byte	16#0	16#00		
19	data[16]	Byte	16#0	16#00		

Figure 24: Example of a completed I&M1 write action of a PROFINET IO device

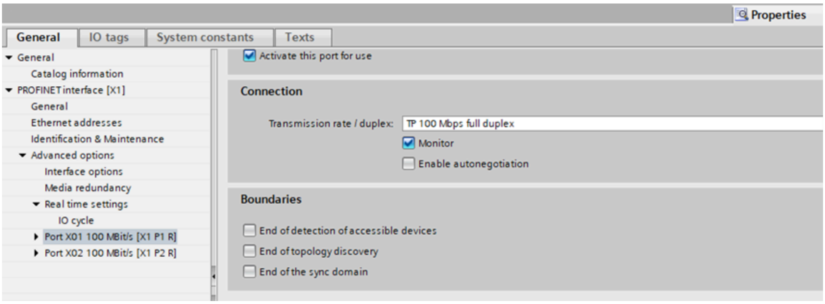
8.6 Fast Start Up (FSU) / Prioritized Startup

The UNITRONIC® ACCESS 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 for the UNITRONIC® ACCESS devices with **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)]

< 450 ms

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

< 500 ms

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

~5500 ms

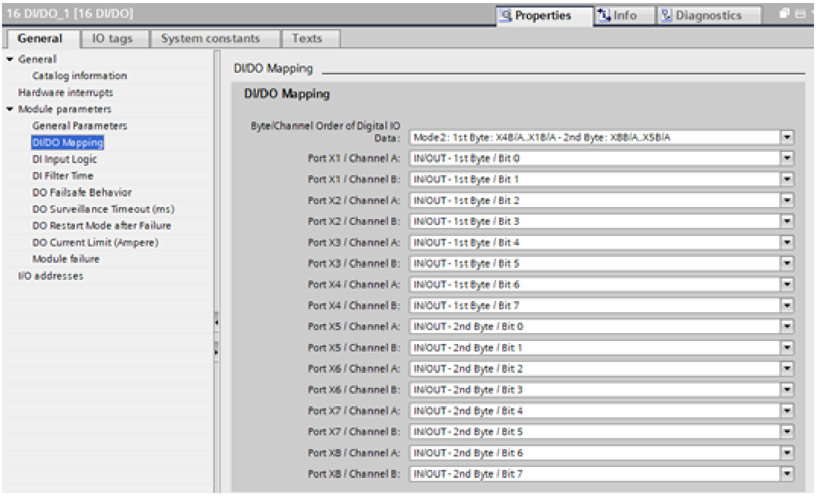
1) Measured according to specification: Internal switch is able to forward telegrams.

2) PLC reads one digital input and sets one digital output on I/O-Device after power-up of the DUT. The PLC is connected directly to DUT port X01 without any additional switch between PLC and DUT.

9 Process data assignment

This chapter describes the cyclic I/O data mapping between the PLC and the I/O device. The mapping depends on the device specific setting of parameter *DI/DO Mapping*.

For the DI/DO Mapping Mode configuration, see chapter DI/DO Mapping on page 47.



Key

X1A = Port 1, Channel A

1st Byte = low address byte in a Siemens PLC

2nd Byte = high address byte in a Siemens PLC

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

9.1 MP08DIO08DIO

9.1.1 16 DI/DO

9.1.1.1 Mapping Mode 1

Slot	Input/Output	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1st Byte	X8B	X8A	X7B	X7A	X6B	X6A	X5B	X5A
	2nd Byte	X4B	X4A	X3B	X3A	X2B	X2A	X1B	X1A

9.1.1.2 Mapping Mode 2

Default setting

Slot	Input/Output	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1st Byte	X4B	X4A	X3B	X3A	X2B	X2A	X1B	X1A
	2nd Byte	X8B	X8A	X7B	X7A	X6B	X6A	X5B	X5A

9.1.1.3 Mapping Mode 3

Slot	Input/Output	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1st Byte	X8B	X7B	X6B	X5B	X4B	X3B	X2B	X1B
	2nd Byte	X8A	X7A	X6A	X5A	X4A	X3A	X2A	X1A

9.1.1.4 Mapping Mode 4

Slot	Input/Output	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1st Byte	X8A	X7A	X6A	X5A	X4A	X3A	X2A	X1A
	2nd Byte	X8B	X7B	X6B	X5B	X4B	X3B	X2B	X1B

9.1.1.5 Mapping Mode 5

The mapping for this mode depends on the user settings.

9.1.2 16 DI

9.1.2.1 Mapping Mode 1

Slot	Input	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1st Byte	X8B	X8A	X7B	X7A	X6B	X6A	X5B	X5A
	2nd Byte	X4B	X4A	X3B	X3A	X2B	X2A	X1B	X1A

9.1.2.2 Mapping Mode 2

Default setting

Slot	Input	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1st Byte	X4B	X4A	X3B	X3A	X2B	X2A	X1B	X1A
	2nd Byte	X8B	X8A	X7B	X7A	X6B	X6A	X5B	X5A

9.1.2.3 Mapping Mode 3

Slot	Input	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1st Byte	X8B	X7B	X6B	X5B	X4B	X3B	X2B	X1B
	2nd Byte	X8A	X7A	X6A	X5A	X4A	X3A	X2A	X1A

9.1.2.4 Mapping Mode 4

Slot	Input	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1st Byte	X8A	X7A	X6A	X5A	X4A	X3A	X2A	X1A
	2nd Byte	X8B	X7B	X6B	X5B	X4B	X3B	X2B	X1B

9.1.2.5 Mapping Mode 5

The mapping for this mode depends on the user settings.

9.1.3 16 DO

9.1.3.1 Mapping Mode 1

Slot	Output	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1st Byte	X8B	X8A	X7B	X7A	X6B	X6A	X5B	X5A
	2nd Byte	X4B	X4A	X3B	X3A	X2B	X2A	X1B	X1A

9.1.3.2 Mapping Mode 2

Default setting

Slot	Output	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1st Byte	X4B	X4A	X3B	X3A	X2B	X2A	X1B	X1A
	2nd Byte	X8B	X8A	X7B	X7A	X6B	X6A	X5B	X5A

9.1.3.3 Mapping Mode 3

Slot	Output	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1st Byte	X8B	X7B	X6B	X5B	X4B	X3B	X2B	X1B
	2nd Byte	X8A	X7A	X6A	X5A	X4A	X3A	X2A	X1A

9.1.3.4 Mapping Mode 4

Slot	Output	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1st Byte	X8A	X7A	X6A	X5A	X4A	X3A	X2A	X1A
	2nd Byte	X8B	X7B	X6B	X5B	X4B	X3B	X2B	X1B

9.1.3.5 Mapping Mode 5

The mapping for this mode depends on the user settings.

9.1.4 8 DI

9.1.4.1 Mapping Mode 5

Default setting

All 16 inputs are physically available, but only 8 inputs can be mapped to one input byte.

Slot	Input	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1st Byte	X8A	X7A	X6A	X5A	X4A	X3A	X2A	X1A

9.1.5 8 DI/8 DO, Opt. 1

9.1.5.1 Mapping Mode 5

Default setting

All 16 inputs/outputs are physically available, but only 8 inputs and 8 outputs can be mapped to one input byte and one output byte.

Slot	Input	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1st Byte	X4B	X4A	X3B	X3A	X2B	X2A	X1B	X1A

Slot	Output	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1st Byte	X8B	X8A	X7B	X7A	X6B	X6A	X5B	X5A

9.1.6 8 DI/8 DO, Opt. 2

9.1.6.1 Mapping Mode 5

Default setting

All 16 inputs/outputs are physically available, but only 8 inputs and 8 outputs can be mapped to one input byte and one output byte.

Slot	Input	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1st Byte	X8A	X7A	X6A	X5A	X4A	X3A	X2A	X1A

Slot	Output	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1st Byte	X8B	X7B	X6B	X5B	X4B	X3B	X2B	X1B

9.1.7 8 DI/8 DO, Opt. 3

9.1.7.1 Mapping Mode 5

Default setting

All 16 inputs/outputs are physically available, but only 8 inputs and 8 outputs can be mapped to one input byte and one output byte.

Slot	Input	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1st Byte	X8B	X7B	X6B	X5B	X4B	X3B	X2B	X1B

Slot	Output	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1st Byte	X8A	X7A	X6A	X5A	X4A	X3A	X2A	X1A

9.1.8 8 DI

9.1.8.1 Mapping Mode 5

Default setting

All 16 outputs are physically available, but only 8 outputs can be mapped to one output byte.

Slot	Output	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1st Byte	X8A	X7A	X6A	X5A	X4A	X3A	X2A	X1A

9.2 DO currents X1 .. X4

This module can be configured optionally in slots 2, 3 or 4 when digital outputs are used. The content is four UINT16 of the actual measured output current.

A measured value of 1000mA will be transferred as "0x03E8 = b0000001111101000".

Slot	Input	Bit	Bit	Bit	Bit	Bit	Bit	Bit	Bit
2 .. 4	1st Byte X1	15	14	13	12	11	10	9	8
	2nd Byte X1	7	6	5	4	3	2	1	0
	3rd Byte X2	15	14	13	12	11	10	9	8
	4th Byte X2	7	6	5	4	3	2	1	0
	5th Byte X3	15	14	13	12	11	10	9	8
	6th Byte X3	7	6	5	4	3	2	1	0
	7th Byte X4	15	14	13	12	11	10	9	8
	8th Byte X4	7	6	5	4	3	2	1	0

9.3 DO currents X5 .. X8

This module can be optionally configured in slots 2, 3 or 4 when digital outputs are used. The content is four UINT16 of the actual measured output current.

A measured value of 1000mA will be transferred as "0x03E8 = b0000001111101000".

Slot	Input	Bit	Bit	Bit	Bit	Bit	Bit	Bit	Bit
2 .. 4	1st Byte X1	15	14	13	12	11	10	9	8
	2nd Byte X1	7	6	5	4	3	2	1	0
	3rd Byte X2	15	14	13	12	11	10	9	8
	4th Byte X2	7	6	5	4	3	2	1	0
	5th Byte X3	15	14	13	12	11	10	9	8
	6th Byte X3	7	6	5	4	3	2	1	0
	7th Byte X4	15	14	13	12	11	10	9	8
	8th Byte X4	7	6	5	4	3	2	1	0

9.4 U_S/U_L voltages X03/X04

This module can be optionally configured in slots 2, 3 or 4. The content is two UINT16 of the actual measured supply voltage U_S and U_L.

A measured value of 24 V will be transferred as "0x5DC0 = 0b0101110111000000".

Slot	Input	Bit	Bit	Bit	Bit	Bit	Bit	Bit	Bit
2 .. 4	1st Byte U_S	15	14	13	12	11	10	9	8
	2nd Byte U_S	7	6	5	4	3	2	1	0
	3rd Byte U_L	15	14	13	12	11	10	9	8
	4th Byte U_L	7	6	5	4	3	2	1	0

9.5 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 14: PROFINET channel diagnostics mapping

10 Diagnostics

10.1 Detailed diagnostics description

10.1.1 Error of the system/sensor power supply U_S

The voltage value for the incoming system/sensor power supply is monitored globally for the device. If the voltage drops below approx. 18 V, or exceeds approx. 30 V, an error message is generated.

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 device 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.
- 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

The voltage value for the incoming U_L power supply is monitored globally for the device. 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 device 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 (U`$_L$`)`
Extended description	`Check wire connection and U`$_L$` 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 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	`0x0102`
Channel related diagnostic code message	`Sensor short circuit`

- 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 (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	`0x0100`
Channel related diagnostic code message	`Actuator short circuit or supply error channel A`

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

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

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 Mod*e 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	`0x0101`
Channel related diagnostic code message	`Actuator short circuit or supply error channel B`

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

10.1.6 Generic parameter error

When a device parameter will be written to an invalid address (e.g. Sub-Slot / Index) or the parameter data content is detected as invalid for the device, the following device 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`

10.1.7 I/O mapping parameter error

The individual I/O data mapping parameter of the Status/Control data will be checked by the PROFINET IO device. 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`

10.1.8 Force mode diagnostic

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`

10.1.9 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 PROFINET diagnostic codes

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

Diagnostic code	Definition	Type
0x0000	Reserved	
0x0002	Undervoltage	Error
0x0009	Error	Error
0x000A	Simulation active	Error
0x0010	Parameter error	Error
0x0118	Low voltage of actuator power supply (U_L). Check power supply	Error
0x011A	I/O mapping configuration faulty	Error

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.

11.1 MQTT

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

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

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 request. For more information see chapter MQTT configuration - Quick start guide on page 103.

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
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 88.

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

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 88.

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

Variable	Description
mac	The MAC address of the device
name	The name of the device
order	The ordering number of the device
serial	The serial number of the device
ip0 ip1 ip2 ip3	IP address octets

Table 16: Base topic variables

Example:

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

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

`Base-Topic/domain/.….`

There are the following domains:

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

Table 17: Data domains

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

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

Table 18: Data model

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

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

Table 19: Use case examples

11.1.2.2 Publish topic

Overview of all publish JSON data for the defined topics:

Identity/gateway	
Key	**Data type**
product_name	json_string
ordering_number	json_string
device_type	json_string
serial_number	json_string
mac_address	json_string
production_date	json_string
fw_name	json_string
fw_date	json_string
fw_version	json_string
hw_version	json_string
family	json_string
location	json_string
country	json_string
fax	json_string
vendor_name	json_string
vendor_address	json_string
vendor_phone	json_string
vendor_email	json_string
vendor_techn_support	json_string
vendor_url	json_string
vendor_id	json_integer
device_id	json_integer

Table 20: Identity/gateway

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

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

Table 21: Config/gateway

Status/gateway				
Key	**Data type**	**Range**	**Default value**	**Remarks**
protocol	json_string	PROFINET: ▶ UNKNOWN ▶ OFFLINE ▶ STOP ▶ IDLE ▶ OPERATE EtherNet/IP: ▶ CONNECTED ▶ DISCONNECTED EtherCAT®: ▶ PREOP ▶ SAFEOP ▶ OP ▶ INIT ▶ UNKNOWN Modbus TCP: ▶ No Connections ▶ Connected CC-Link IE Feld Basic: ▶ ON ▶ STOP ▶ DISCONNECTED ▶ ERROR		
system_voltage_fault	json_boolean	true / false		
actuator_voltage_fault	json_boolean	true / false		
internal_module_error	json_boolean	true / false		
simulation_active_diag	json_boolean	true / false		
us_voltage	json_integer	0 .. 32		in Volts
ul_voltage	json_integer	0 .. 32		in Volts
forcemode_enabled	json_boolean	true / false		

Table 22: Status/gateway

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

Table 23: Process/gateway

Identity/port/1 .. 8				
Key	**Data type**	**Range**	**Default value**	**Remarks**
port	json_integer	1 .. 8		
type	json_string	Digital Input DIO Digital Output DIO Pin 4 Only DI Pin 4 Only DO Pin 4 Only Not available Unknown		
max_output_power_cha	json_string	2.0_mA 0.5_mA		
max_output_power_chb	json_string	2.0_mA 0.5_mA		
channel_cha	json_string	Digital Input Digital Output DIO Digital Input/Output Auxiliary Power Auxiliary with DO Not available Unknown		
channel_chb	json_string	Digital Input Digital Output DIO Digital Input/Output Auxiliary Power Auxiliary with DO Not available Unknown		

Table 24: Identity/port/1 .. 8

Config/port/1 .. 8				
Key	**Data type**	**Range**	**Default value**	**Remarks**
port	json_integer	1 .. 8		
direction_cha	json_string	Output Input Inactive Auxiliary Power DIO Unknown		
direction_chb	json_string	Output Input Inactive Auxiliary Power DIO Unknown		
restart_mode_cha	json_string	Manual Auto		
restart_mode_chb	json_string	Manual Auto		
input_polarity_cha	json_string	NO NC		
input_polarity_chb	json_string	NO NC		
input_filter_cha	json_integer			ms
input_filter_chb	json_integer			ms
do_auto_restart_cha	json_boolean	true / false		
do_auto_restart_chb	json_boolean	true / false		
failsafe_cha	json_string	set_low set_high hold_last	set_low	
failsafe_chb	json_string	set_low set_high hold_last	set_low	
surveillance_timeout_cha	json_integer	0 .. 255	80	

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

Table 25: Config/port/1 .. 8

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

Table 26: Status/port/1 .. 8

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 100)		

Table 27: Force object properties

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

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

Table 28: Force object: Digital

[…]/config

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

Property	Data type	Example values	Remarks
portmode	array (Table 30: Config object: Portmode on page 101)		
ip_address	string	"192.168.1.5"	
subnet_mask	string	"255.255.255.0"	
gateway	string	"192.168.1.100"	

Table 29: Config object properties

For the *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		

Table 30: Config object: Portmode

*channelA = Pin 4, channelB = Pin 2

[…]/reset

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

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

Table 31: Reset object properties

[…]/publish

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

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

11.1.3 MQTT configuration - Quick start guide

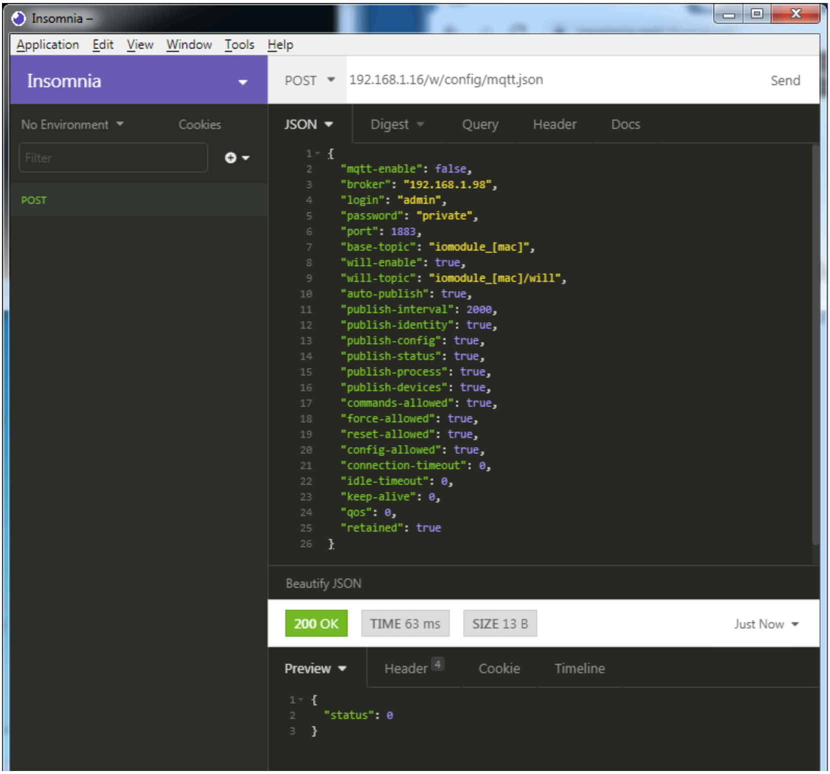
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

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/

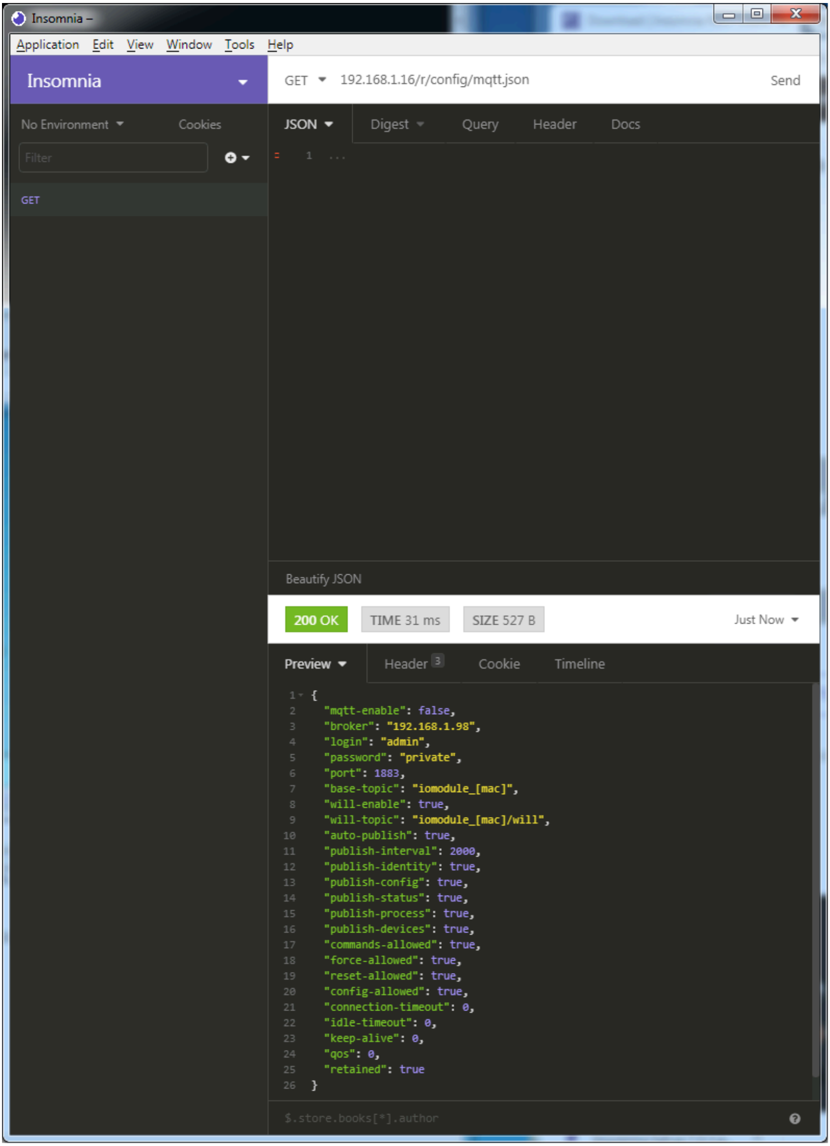
2. Configure MQTT:

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



3. Read MQTT:

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



11.2 OPC UA

OPC 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.

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 request.

The configuration URL is:

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

The configuration can also read back as a JSON file:

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

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

Tree overview of OPC UA objects:

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

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

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

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

Response:

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

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

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

Examples:

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

{"status": 0}

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

11.2.1.1 Gateway objects

Identity

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

Status (read)

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

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

Forcing (read)

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

Config (read + write)

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

Process (read)

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

Commands (write)

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

11.2.1.2 Ports objects

Identity

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

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

See details in Channel objects on page 112.

Status (read)

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

Config (read + write)

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

11.2.1.3 Channel objects

Identity (read)

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

Status (read)

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

Config (read + write)

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

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

Process (read)

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

Forcing (read + write)

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

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

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

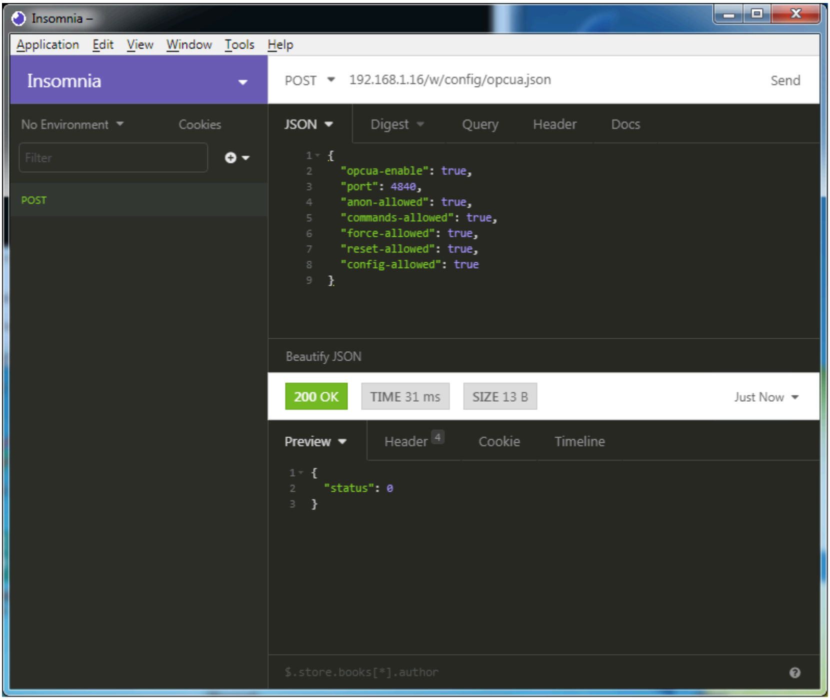
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

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/

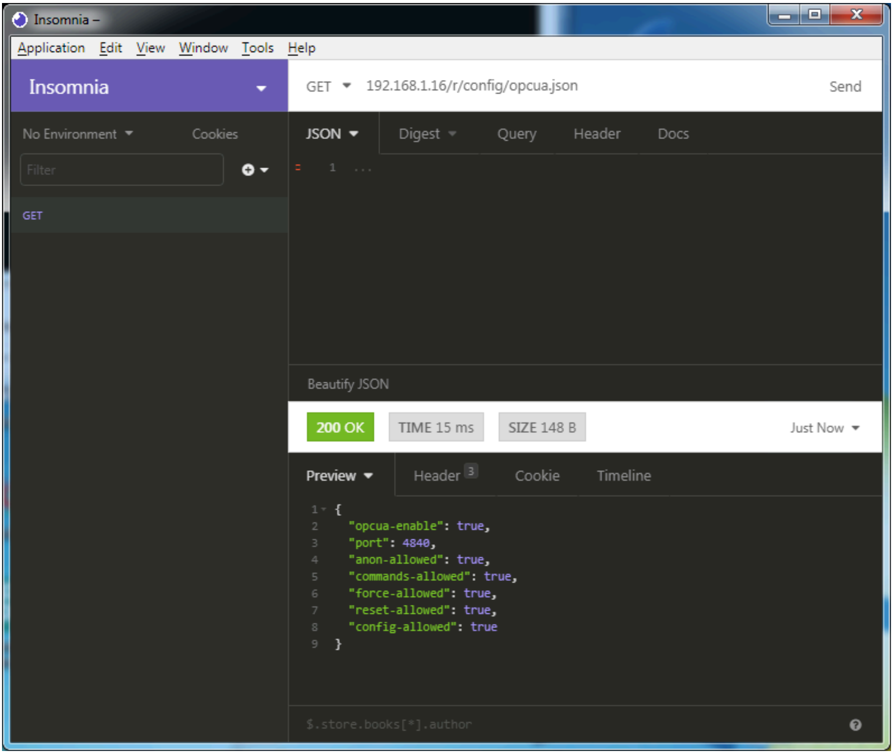
2. Configure OPC UA:

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 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.

The customized LAPP REST API 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.

11.3.2 Structure

Name	Data type	Description	Example
name	string	Device name	"MP08DIO08DIO "
order-id	string	Ordering number	"381166718"
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
CHANNEL Object			
name	string	Name of channel	
type	number	Hardware channel type as number: 0 = DIO 1 = Input 2 = Output 3 = Input/Output 4 = Channel not available 5 = Channel not available 6 = Channel not available 7 = Channel not available 8 = Channel not available	
type_text	string	Textual representation of the channel type	
config	number	Current configuration of the channel: 0 = DIO 1 = Input 2 = Output 3 = Channel not available 4 = Deactivated 5 = Channel not available	
config_text	string	Textual representation of the current config	
inputState	boolean	Input data (producing data) bit to the PLC	
outputState	boolean	Output data bit to the physical output pin	
forced	boolean	True, if the output pin of this channel is forced	
simulated	boolean	True, if the input value to the PLC of this channel is simulated	
actuatorDiag	boolean	True, if the output is in short circuit / overload condition	
sensorDiag	boolean	True, if the sensor supply (Pin 1) is in short circuit / overload condition	

Name	Data type	Description	Example
maxOutputCurrent_mA	number	Maximum output current of the output in mA	
current_mA	number	Measured current of the output in mA (if current measurement is available)	
voltage_mV	number	Measured voltage of this output in mV (if voltage measurement is available)	
PORT Object			
port_type	string	Textual representation of the port type	
aux_mode	number	Indicates the configured mode for the Pin 2: 0 = No AUX 1 = AUX output (always on) 2 = Digital output (can be controlled by cyclic data) 3 = Digital input	
aux_text	string	Textual representation of the current aux mode	"AUX Output"
ds_fault	number	Data storage error number	
ds_fault_text	string	Textual data storage error.	
diag	array of DIAG (n)	Array of port related events	
DIAG Object			
error	number	Error code	
source	string	Source of the current error.	"device" "master"
message	string	Error message	"Supply Voltage fault"
FORCING Object		Forcing information of the device	
forcingActive	boolean	Force mode is currently active	
forcingPossible	boolean	True, if forcing is possible and force mode can be activated	
AuthPossible	boolean	True, if the JSON Interface can obtain forcing autorization	
ownForcing	boolean	True, if forcing is performed by REST API at the moment	
currentClient	string	Current forcing client identifier	

Name	Data type	Description	Example
digitalOutForced	array of numbers (2)	The force values of all 16 digital output channels.	
digitalOutMask	array of numbers (2)	The forcing mask of all 16 digital output channels.	
digitalInForced	array of numbers (2)	The force values of all 16 digital input channels.	
digitalInMask	array of numbers (2)	The forcing mask of all 16 digital input channels.	

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)		

Table 32: 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", "off", "aux"	
inlogica	string	"no","nc"	
inlogicb	string	"no","nc"	

Table 33: Port mode object

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

Table 34: Digital object

11.4 CoAP server

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

CoAP is based on the Server-Client principle and a service layer protocol that lets nodes and machines communicate with each other. The 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 127.

The configuration URL is:

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

The configuration can also read back as a JSON file:

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

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

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

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

Table 35: CoAP configuration

CoAP response:

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

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

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

Examples:

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

{"status": 0}

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

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	

Table 36: REST API access via CoAP

11.4.3 CoAP configuration - Quick start guide

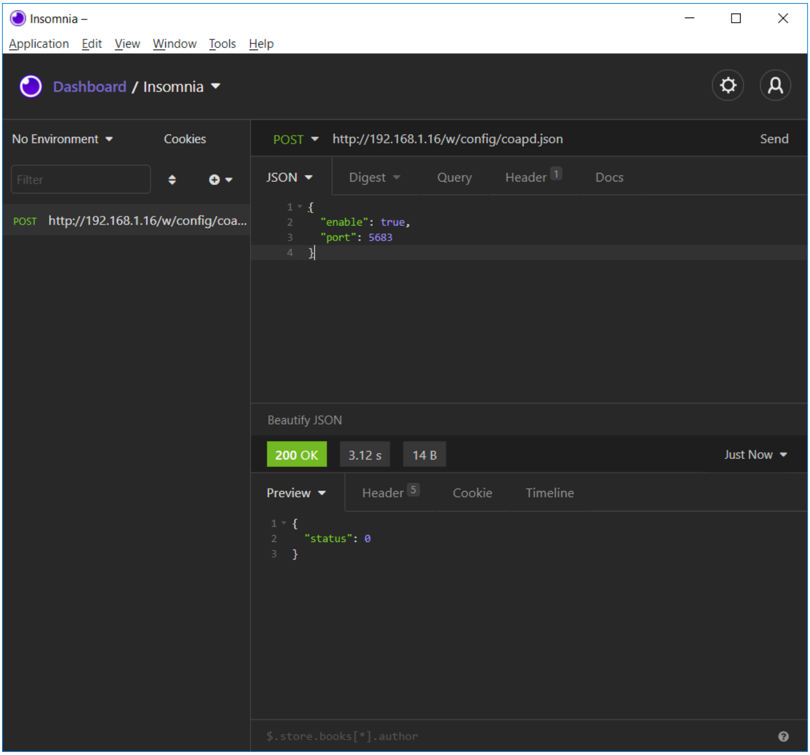
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

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/

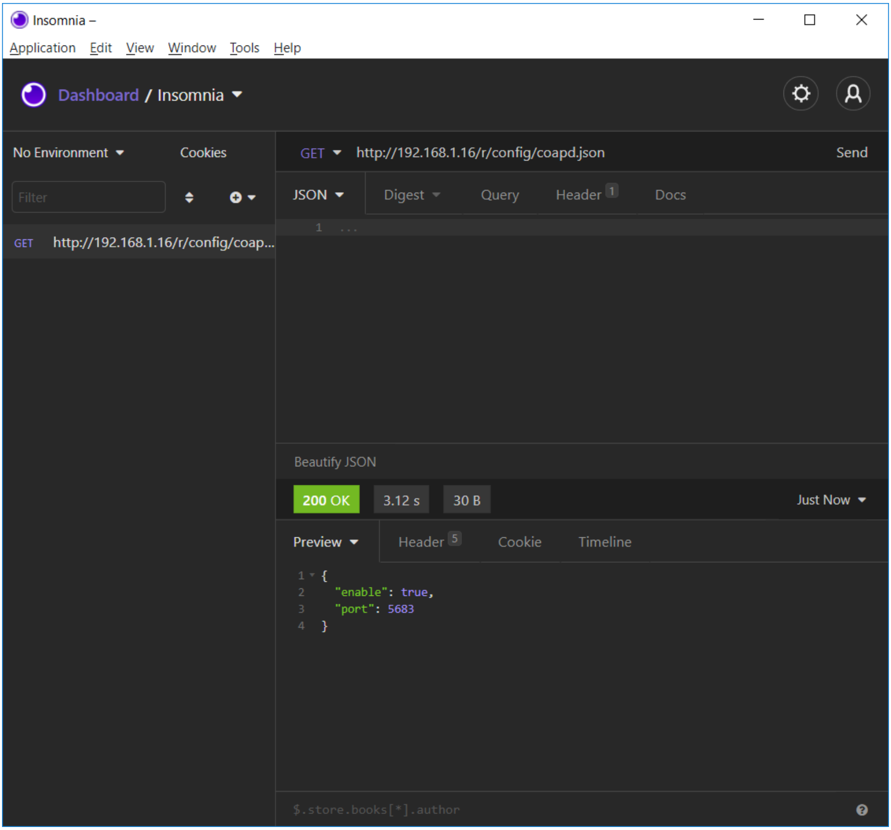
2. Configure CoAP:

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



3. Read CoAP configuration:

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



11.5 Syslog

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 will not store any message permanently.

11.5.1 Syslog configuration

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

The configuration URL is:

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

The configuration can also read back as a JSON file:

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

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

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

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

Table 37: Syslog configuration

Syslog response:

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

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

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

Examples:

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

{"status": 0}

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

11.5.2 Syslog configuration - Quick start guide

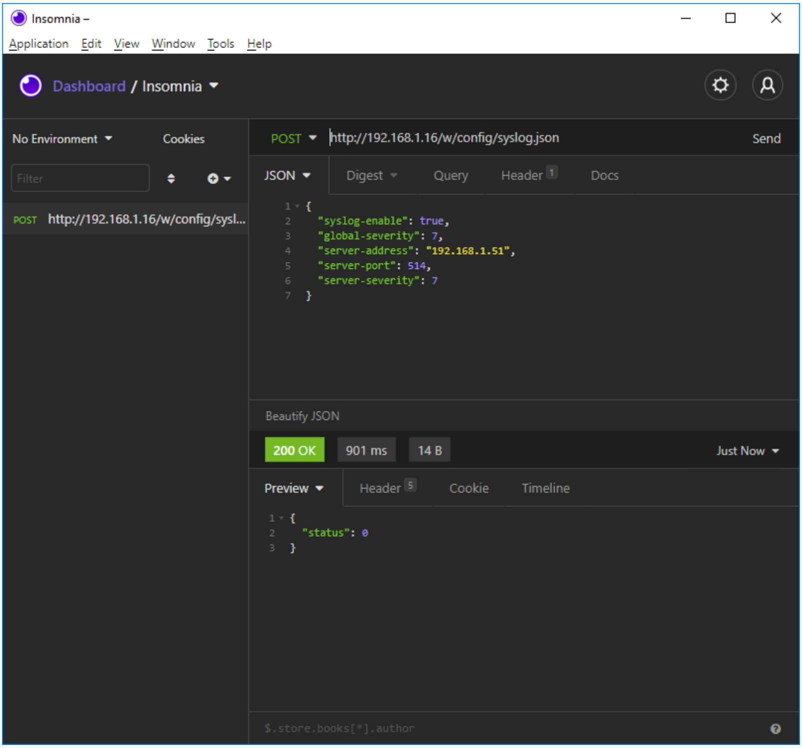
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

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/

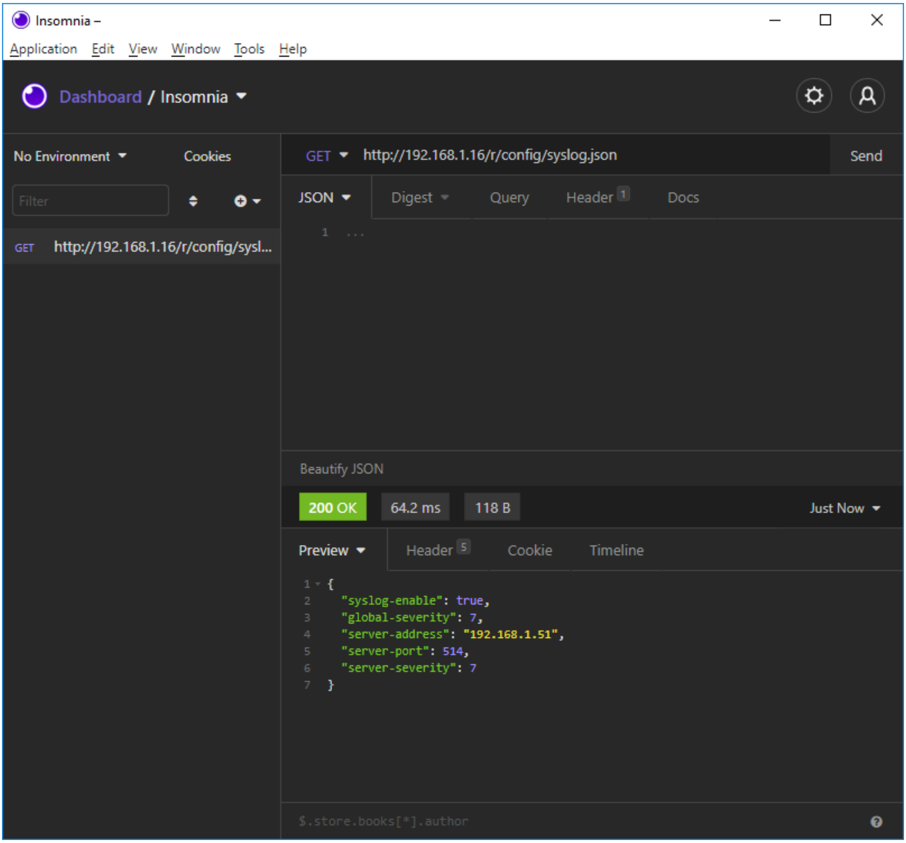
2. Configure Syslog:

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 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 136.

The configuration URL is:

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

The configuration can also read back as a JSON file:

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

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

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

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

Table 38: NTP configuration

NTP response:

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

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

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

Examples:

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

{"status": 0}

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

11.6.2 NTP configuration - Quick start guide

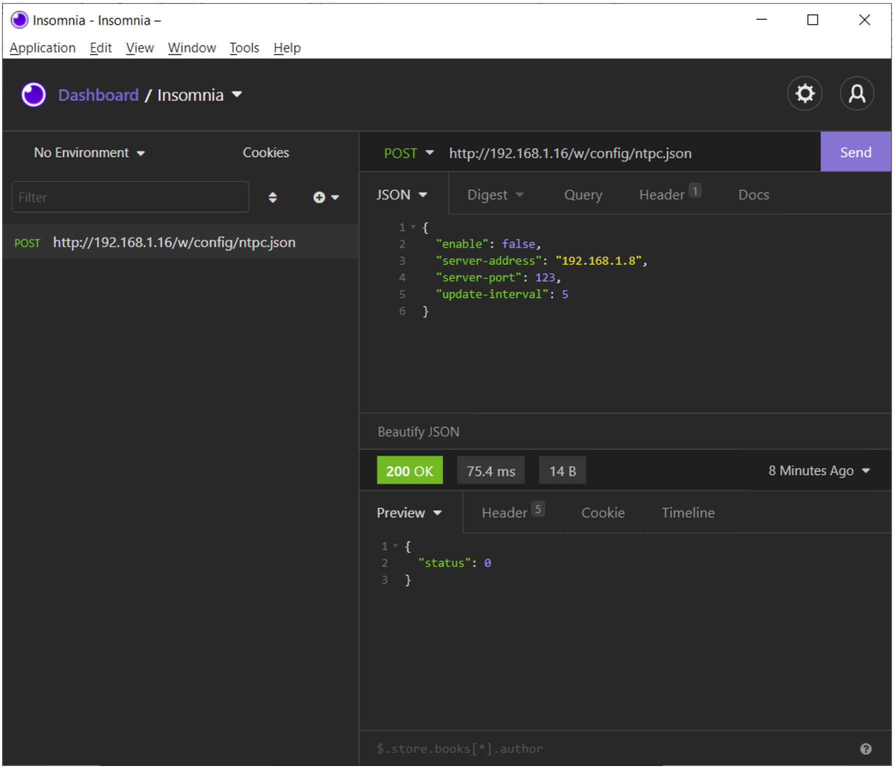
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

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/

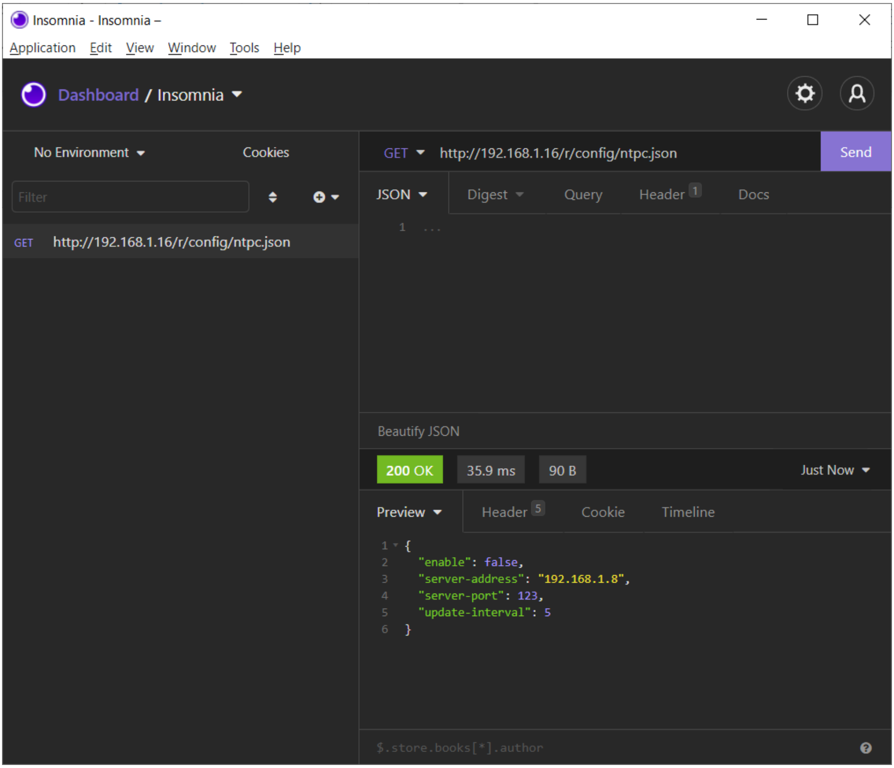
2. Configure NTP:

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



3. Read NTP configuration:

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



12 The integrated Web server

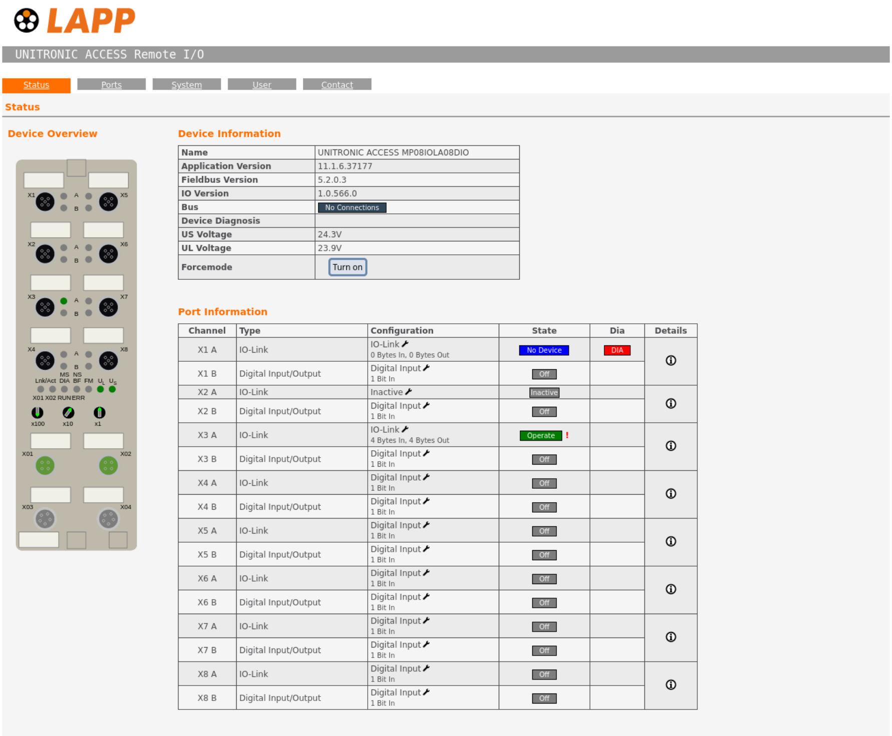
All device variants are equipped with an integrated Web server which makes functions for the device configuration and the display of status and diagnostic information available via a Web interface.

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://" 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


LAPP

UNITRONIC ACCESS Remote I/O

Status | Ports | System | User | Contact

Status

Device Overview

Device Information

Name	UNITRONIC ACCESS MP08IOLA08DIO
Application Version	11.1.6.37177
Fieldbus Version	5.2.0.3
IO Version	1.0.566.0
Bus	No Connections
Device Diagnosis	
US Voltage	24.3V
UL Voltage	23.9V
Forcemode	Turn on

Port Information

Channel	Type	Configuration	State	Dia	Details
X1 A	IO-Link	IO-Link 0 Bytes In, 0 Bytes Out	No Device	DIA	ⓘ
X1 B	Digital Input/Output	Digital Input 1 Bit In	Off		
X2 A	IO-Link	Inactive	Inactive		ⓘ
X2 B	Digital Input/Output	Digital Input 1 Bit In	Off		
X3 A	IO-Link	IO-Link 4 Bytes In, 4 Bytes Out	Operate !		ⓘ
X3 B	Digital Input/Output	Digital Input 1 Bit In	Off		
X4 A	IO-Link	Digital Input 1 Bit In	Off		ⓘ
X4 B	Digital Input/Output	Digital Input 1 Bit In	Off		
X5 A	IO-Link	Digital Input 1 Bit In	Off		ⓘ
X5 B	Digital Input/Output	Digital Input 1 Bit In	Off		
X6 A	IO-Link	Digital Input 1 Bit In	Off		ⓘ
X6 B	Digital Input/Output	Digital Input 1 Bit In	Off		
X7 A	IO-Link	Digital Input 1 Bit In	Off		ⓘ
X7 B	Digital Input/Output	Digital Input 1 Bit In	Off		
X8 A	IO-Link	Digital Input 1 Bit In	Off		ⓘ
X8 B	Digital Input/Output	Digital Input 1 Bit In	Off		

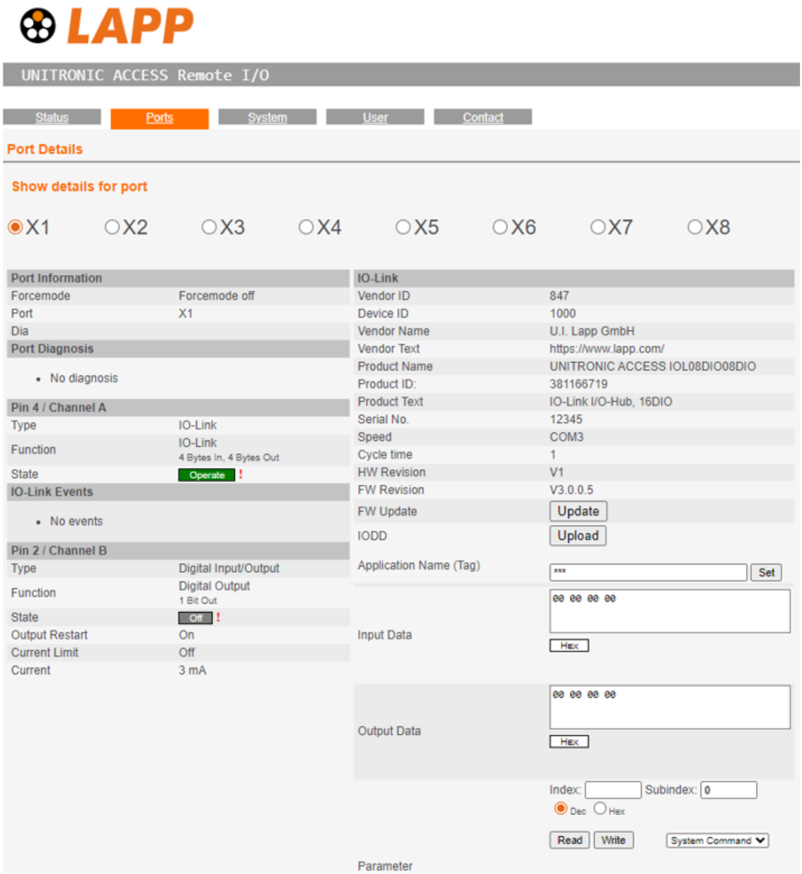
The status page provides a quick overview of the current state of the device.

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.

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.

Restart Device

The module initializes a software reset.

Reset to Factory Settings

The module restores to the default factory settings.

IP Settings

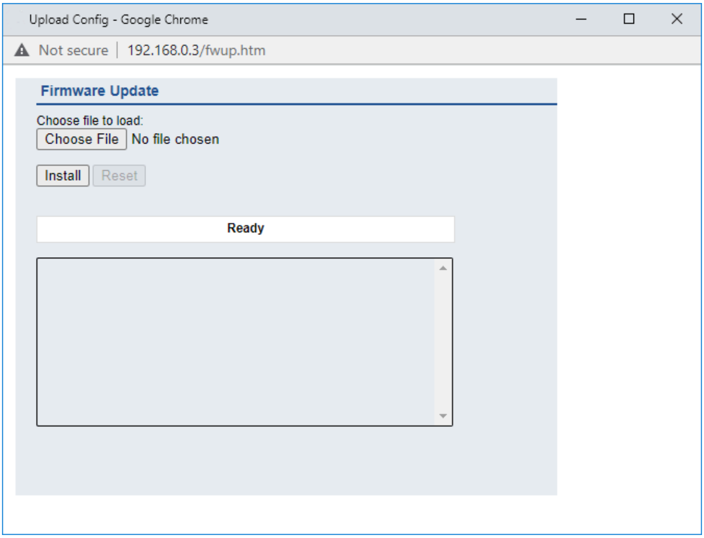
Use this parameter to change the current IP address of the module.

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.

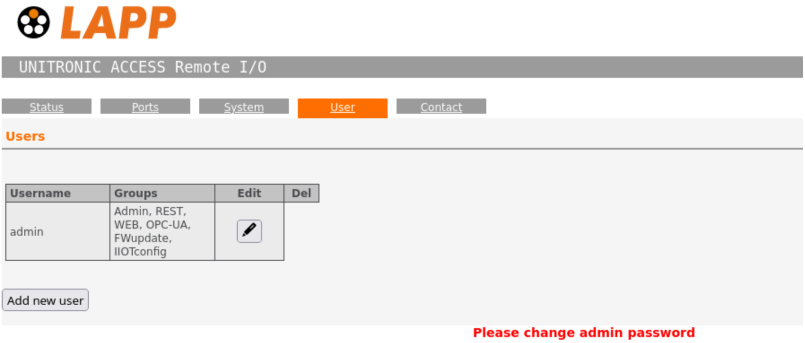
Firmware Update

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.



12.1.4 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 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.

13.1 General

Protection class (Only applies if the connectors are screwed together or if protective caps are used.)[3]	IP65 IP67 IP69K	
Ambient temperature (during operation and storage)	MP08DIO08DIO	-40 °C .. +70 °C (-40 °F .. +158 °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 39: General information

3 Not under UL investigation.

13.2 PROFINET protocol

Protocol	PROFINET IO device V2.44
Conformance Class	C
Netload Class	III
Update cycle	1 ms
GSDML file	GSDML-V2.44-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 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 40: PROFINET protocol

13.3 Power supply of the module electronics/ sensors

Port X03, X04	M12-L-coded Power, connector/socket, 5-pole Pin 1 / Pin 3		
Nominal voltage U_S	24 V DC (SELV/PELV)		
Current U_S	Max. 16 A		
Voltage range	21 .. 30 V DC		
Power consumption of module electronics	Typically 160 mA (+/-20 % at U_S nominal voltage)		
Power supply interruption	Max. 10 ms		
Voltage ripple U_S	Max. 5 %		
Current consumption sensor system (Pin 1)	MP08DIO08DIO	Port X1 .. X8 (Pin 1)	max. 4 A per port (at $T_{ambient}$ = 30° C)
Voltage level of the sensor power supply	Min. (U_S – 1.5 V)		
Short circuit/overload protection of sensor supply	Yes, per port		
Reverse polarity protection	Yes		
Operational indicator (U_S)	LED green:	18 V (+/- 1 V) < U_S	
	LED red:	U_S < 18 V (+/- 1 V)	

Table 41: Information on the power supply of the module electronics/ sensors

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

- Max. 4.0 A per port
- Max. 5.0 A for each port pair X1/X2, X3/X4, X5/X6, X7/X8

- Max. 9.0 A in total for the whole port group X1 .. X8

  Pay attention to the derating!

13.4 Power supply of the actuators

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

Table 42: Information on the power supply of the actuators

13.5 I/O ports

MP08DIO08DIO	Ports X1 .. X8	DI, DO	M12 socket, 5-pin

Table 43: I/O ports: Overview of functions

13.5.1 Digital inputs

Input connection	MP08DIO08DIO		Type 3 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	MP08DIO08DIO	X1 .. X8	16
Status indicator	yellow LED for Channel A (Pin 4) white LED for Channel B (Pin 2)		
Diagnostic indicator	red LED per port		

Table 44: I/O ports configured as digital input

13.5.2 Digital outputs

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

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

Pay attention to the derating!

Output type	normally open, p-switching	
Nominal output voltage per channel	Signal status “1” Signal status “0”	min. (U_L -1 V) max. 2 V
Max. output current per device	MP08DIO08DIO	9 A
Max. output current per channel	MP08DIO08DIO (X1 .. X8)	2 A
Short-circuit/overload protected	yes/yes	
Behavior in case of short circuit or overload	deactivation with automatic power-on (parameterized)	
Number of digital outputs	MP08DIO08DIO (X1 .. X8)	16
Status indicator	yellow LED per output Channel A (Pin 4) white LED per output Channel B (Pin 2)	
Diagnostic indicator	red LED per channel	

Table 45: I/O ports configured as digital output

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

13.6 LEDs

LED	Color	Description
U_L	Green	Auxiliary sensor/actuator voltage OK 18 V (+/- 1 V) < U_L < 30 V (+/- 1 V)
	Red*	Auxiliary sensor/actuator voltage LOW U_L < 18 V (+/- 1 V) or U_L > 30 V (+/- 1 V) * if "Report U_L supply voltage fault" is enabled.
	OFF	None of the above conditions.
U_S	Green	System/sensor voltage OK 18 V (+/- 1 V) < U_S < 30 V (+/- 1 V)
	Red	System/sensor voltage LOW U_S < 18 V (+/- 1 V) or U_S > 30 V (+/- 1 V)
	Red flashing	Device performs a factory reset (position of rotary encoding switches: 9-7-9)
	OFF	None of the above conditions.
X1 .. X8 A	Yellow	Status of digital input or digital output on pin 4 line "on".
	Red	Short circuit on pin 4 line. / Overload or short circuit on L+ (pin 1) line / communication error
	OFF	None of the above conditions.
X1 .. X8 B	White	Status of digital input or digital output on pin 2 line "on".
	Red	Short circuit on pin 2 line. / 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.

Table 46: Information on the LED colors

13.7 Data transfer times

The following tables give an overview of the internal data transfer times of UNITRONIC® ACCESS.

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

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

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

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

Use case 1:

UNITRONIC® ACCESS Digital-I/O configuration with enabled Web interface and *disabled* IIoT protocols

16DIO variant (MP08DIO08DIO):

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

Use case 2:

UNITRONIC® ACCESS Digital-I/O configuration with enabled Web interface and *enabled* IIoT protocols

16DIO variant (MP08DIO08DIO):

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

14 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 Digital-I/O multi-protocol: MP08DIO08DIO (16 × Input/Output)

Technical Support
https://www.lapp.com

Contents

1 About this manual

1.1 General information

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

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

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

U.I. Lapp GmbH

Schulze-Delitzsch-Straße 25
D-70565 Stuttgart
Germany

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	09/2024	

Table 1: Overview of manual revisions

2 Safety instructions

2.1 Intended use

The products described in this manual are decentralized I/O Devices on an Industrial Ethernet Network.

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

The modules fulfill the requirements of the EMC guidelines (89/336/EEC, 93/68/EEC and 93/44/EEC) and the low voltage guideline (73/23/EEC).

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

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

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

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

Only install cables and accessories that fulfill the requirements and regulations for safety, electromagnetic compatibility, and, where applicable, telecommunication end devices, as well as the specification information. Information on which cables and accessories are permitted for the installation can be obtained from 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 this manual are authorized to install and maintain the devices described. These are persons who

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

Only 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 manual, can result in serious personal injury or damage to equipment.

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

3 Designations and synonyms

AOI	Add-On Instruction
API	Application Programming Interface
BF	Bus Fault LED
Big Endian	Data format with High-B on first place (PROFINET)
BUI	Back-Up Inconsistency (EIP diagnostics)
CC	CC-Link IE Field
Ch. A	Channel A (Pin 4) of I/O port
Ch. B	Channel B (Pin 2) of I/O port
CIP	Common Industrial Protocol (media independent protocol)
CoAP	Constrained Application Protocol
CSP+	Control & Communication System Profile Plus
DCP	Discovery and Configuration Protocol
DevCom	Device Comunicating (EIP diagnostics)
DevErr	Device Error (EIP diagnostics)
DI	Digital Input
DIA	Diagnostic LED
DO	Digital Output
DIO	Digital Input/Output
DTO	Device Temperature Overrun (EIP diagnostics)
DTU	Device Temperature Underrun (EIP diagnostics)
DUT	Device under test
EIP	EtherNet/IP
ERP	Enterprise Resource Planning system
ETH	ETHERNET
FE	Functional Earth
FME	Force Mode Enabled (EIP diagnostics)
FSU	Fast Start-Up

GSDML	General Station Description Markup Language
High-B	High-Byte
ICT	Invalid Cycle Time (EIP diagnostics)
IIoT	Industrial Internet of Things
ILE	Input process data Length Error (EIP diagnostics)
IME	Internal Module Error (EIP diagnostics)
I/O	Input / Output
I/O port	X1 .. X8
I/O port pin 2	Channel B of X1 .. X8
I/O port pin 4 (C/Q)	Channel A of X1 .. X8
IVE	IO-Link port Validation Error (EIP diagnostics)
I&M	Identification & Maintenance
JSON	JavaScript Object Notation (platform independent data format)
L+	I/O port pin 1, sensor power supply
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
OLE	Output process data Length Error (EIP diagnostics)
OPC UA	Open Platform Communications Unified Architecture (platform independent, service-oriented architecture)

PLC	Programmable Logic Controller
PN	PROFINET
PWR	Power
REST	REpresentational State Transfer
RFC	Request for Comments
RPI	Requested Packet Interval
RWr	Word data input as seen from the master station (CC-Link)
RWw	Word data output as seen from the master station (CC-Link)
RX	Bit data input as seen from the master station (CC-Link)
RY	Bit data output as seen from the master station (CC-Link)
SCA	Short Circuit Actuator/U_L/U_{AUX} (EIP diagnostics)
SCS	Short Circuit Sensor (EIP diagnostics)
SLMP	Seamless Message Protocol
SNMP	Simple Network Management Protocol
SP	Single Protocol (PROFINET, EtherNet/IP, EtherCAT®, Modbus TCP or CC-Link IE Field Basic)
SPE	Startup Parameterization Error (EIP diagnostics)
U_{AUX}	$U_{Auxiliary}$, supply voltage for the load circuit (Actuator supply on Class B ports)
UDP	User Datagram Protocol
UDT	User-Defined Data Types
UINT8	Byte in PLC (IB, QB)
UINT16	Unsigned integer with 16 bits or word in PLC (IW, QW)
U_L	U_{Load}, supply voltage for the load circuit (Actuator supply on Class A)
UL	Underwriters Laboratories Inc. (certification company)
UTC	Coordinated Universal Time (Temps Universel Coordonné)

Table 2: Designations and synonyms

4 System description

4.1 Device variants

The following Digital I/O device variants are available in the UNITRONIC® ACCESS family:

Article number	Product designation	Description	I/O port functionality
381166718	MP08DIO08DIO	UNITRONIC® ACCESS M12-60 mm, I/O Device Multi-protocol (PN, EIP, EC, MB, CC) Security	16 x Input/Output universal

Table 3: Overview of UNITRONIC® ACCESS Digital-I/O variants

4.2 I/O port overview

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

UNITRONIC® ACCESS 16DIO ports

Device variant	Port	Pin 1 U_S	Pin 4 / Ch. A (In/Out)		Pin 2 / Ch. B (In/Out)	
MP08DIO08DIO	**Info:**	–	Type 3	Supply by U_L	Type 3	Supply by U_L
	X8:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)
	X7:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)
	X6:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)
	X5:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)
	X4:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)
	X3:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)
	X2:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)
	X1:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)

Table 4: Port configuration of MP08DIO08DIO variant

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 Digital-I/O modules 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 Digita-I/O modules 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.

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 Integrated Web server

Network parameter display

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

Displaying diagnostics

View diagnostics via the integrated Web server.

User management

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

5.3 Security features

Firmware signature

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

Syslog

The 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.

5.4 Other features

Interface protection

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

For more details, see section Port assignments on page 25.

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 Digital-I/O multi-protocol variants

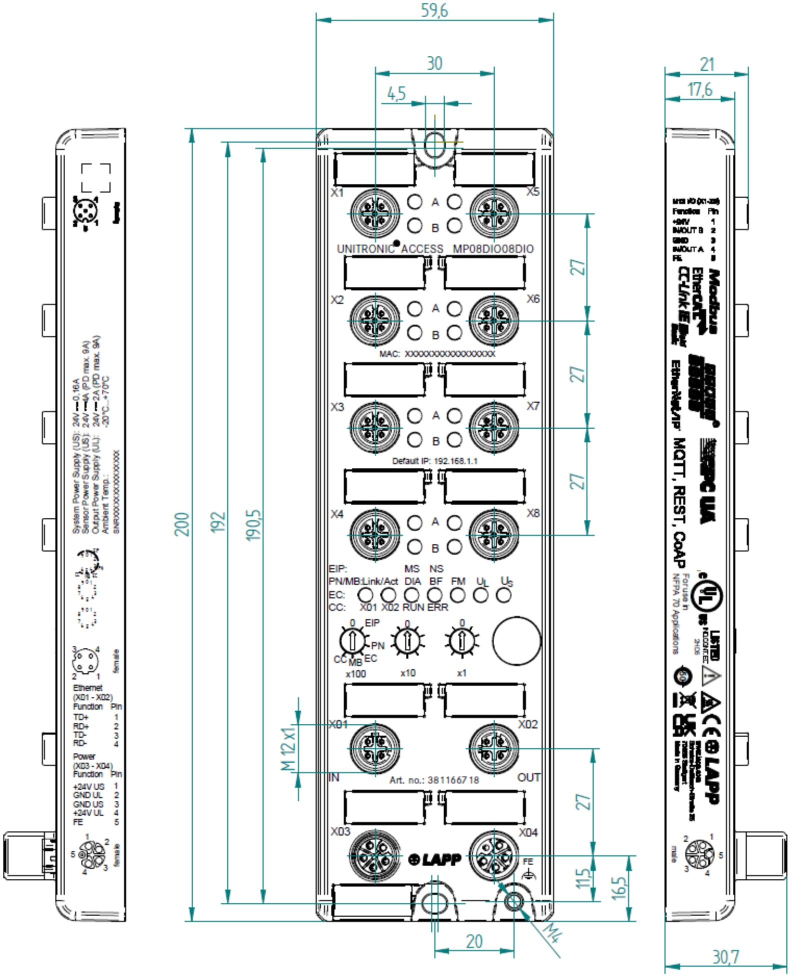


Figure 1: MP08DIO08DIO

6.2.2 Notifications

Attention:

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

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

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

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

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

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

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

6.3 Port assignments

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

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

Color coding: green

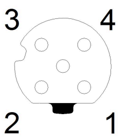


Figure 2: Schematic drawing, ports X01, X02

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

Table 5: Assignment of ports X01, X02

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

6.3.2 Power supply with M12 power L-coded

Color coding: gray

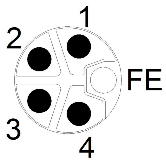


Figure 3: Schematic diagram of the M12 L-coding (connector X03 for Power In)

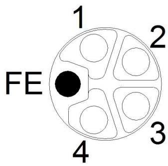


Figure 4: Schematic diagram of the M12 L-coding (socket X04 for Power Out)

Power supply	Pin	Signal	Function
	1	U_S (+24 V)	Sensor/system power supply
	2	GND_U_L	Ground/reference potential U_L
	3	GND_U_S	Ground/reference potential U_S
	4	U_L (+24 V)	Load supply Actuator supply
	5	FE	Functional ground

Table 6: Pin assignments ports X03 and X04

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

6.3.3 I/O ports as M12 sockets

Color coding: black

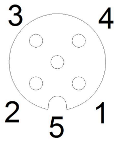


Figure 5: Schematic drawing I/O port as M12 socket

6.3.3.1 I/O ports

MP08DIO08DIO	Pin	Signal	Function
16DIO X1 .. X8	1	+24 V	power supply +24 V
	2	IN/OUT	Ch. B: Digital input or digital output
	3	GND	Ground/reference potential
	4	IN/OUT	Ch. A: Digital input or digital output
	5	FE	Functional ground

Table 7: Pin assignments I/O ports

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:	MP08DIO08DIO
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.

7.5 Setting the rotary encoding switches

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

- MP08DIO08DIO

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.

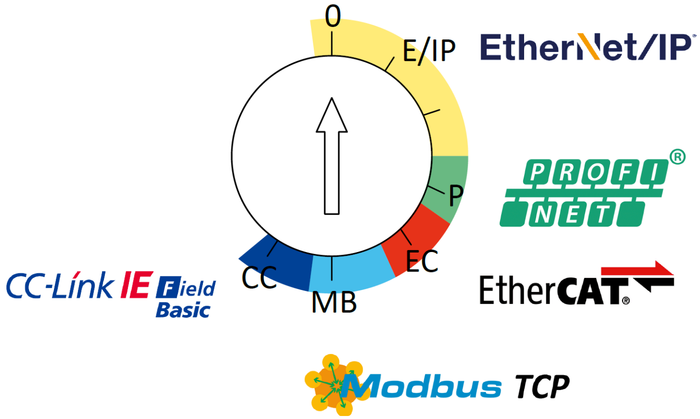
After selecting a protocol followed by a restart of the device, the new protocol is initialized. This can take up to 15 seconds. During this time the device is not usable and the LED indicators are out of function. When the protocol change is complete, the LED indicators return to normal operation and 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 Digital I/O devices 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 I/O Device in different environments.

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

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



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

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

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

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

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

Once you have set the protocol using the rotary encoding switches, the device stores this setting when it starts in cyclic communication. Changing the protocol using the rotary encoding switch is no longer possible after this point. The device will always start using the stored protocol from that point on. The IP address can be changed depending on the selected protocol.

To change the protocol, carry out a factory reset. In this way you restore the factory settings of the respective device. How you perform the factory reset for your device is described in chapter Factory reset on page 34.

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

7.5.1 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 9: 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 is blinking red. After the internal memory write processes have finished, the U_S LED returns to display static green or red light, in dependency of the actual U_S voltage.

	x100	x10	x1
Factory Reset	9	7	9

Follow the steps from section Setting the rotary encoding switches on page 30 again to select a new protocol.

For performing a factory reset via software configuration, see chapter OPC UA configuration on page 109 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 61.

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	4 Byte (16DIO, 8DI/DO) 0 Byte (16DI)	Consuming Data Image
131	Input Connection Point Assembly	8 Byte (16DIO, 8DI/DO) 6 Byte (16DI)	Producing Data Image
132	Input Connection Point Assembly with extended diagnosis	36 Byte (16DIO) 20 Byte (8DI/DO) 2 Byte (16DI)	Producing Data Image with extended diagnosis
140	Configuration Assembly	208 Byte	Module Configuration Data

The *Consuming Data Image* and the *Producing Data Image* have fixed sizes which depend on the connection. The general input and output process data sizes of each connection can be configured in the engineering tool.

The contents of the *Consuming Data Image* and the *Producing Data Image* are specified in chapter Process data assignment on page 49.

The *Module Configuration Data* is defined in chapter Configuration parameters on page 40.

8.2 Connections

The UNITRONIC® ACCESS I/O modules support four different connection types which are defined as follows:

Connect-ion name	Connect-ion type	Output connect-ion point assembly	Output data size	Input connect-ion point assembly	Input data size	Configu-ration assembly	Configu-ration data size
16 DI/DO (Exclusive Owner)	Exclusive Owner	130	4 Byte	131	8 Byte	140	0 or 208 Byte
16 DI (Input Only)	Input Only	193	0 Byte	131	8 Byte	140	0 or 208 Byte
16 DI (Listen Only)	Listen Only	192	0 Byte	131	8 Byte	n/a	0 Byte
Extended Diagnoses (Input Only)	Input Only	192	0 Byte	132	36 Byte	140	0 or 208 Byte

The general input and output process data sizes of each connection are fixed.

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 16 DI/DO (Exclusive Owner) parameters

Connection properties	
Connection name	16 DI/DO (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	4 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	8 Byte
Data type	INT (2 Byte)

8.2.2 16 DI (Input Only) parameters

Connection properties	
Connection name	16 DI (Input Only)
Application type	Input Only
Trigger mode	Cyclic
RPI	min. 1 ms

Connection parameters (O->T)	
Real time transfer format	Heartbeat
Connection type	POINT2POINT
Assembly ID	193
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	8 Byte
Data type	INT (2 Byte)

8.2.3 16 DI (Listen Only) parameters

Connection properties	
Connection name	16 DI (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	8 Byte
Data type	INT (2 Byte)

8.2.4 Extended Diagnoses (Input Only) parameters

Connection properties	
Connection name	Extended Diagnoses (Input Only)
Application type	Input 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	8 Byte
Data type	INT (36 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.

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

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

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
Quick connect	0	SINT	**0: Disable** 1: Enable	Attribute 1
Force Mode Lock	1	SINT	**0: Disable** 1: Enable	Attribute 2
Web Interface Lock	2	SINT	**0: Disable** 1: Enable	Attribute 3
Reserved	3	SINT	–	Attribute 4
Report U_L/U_{Aux} Supply Voltage Fault	4	SINT	**0: Disable** 1: Enable	Attribute 5
Report DO Fault without U_L/U_{Aux}	5	SINT	0: Disable **1: Enable**	Attribute 6
CIP object configuration lock	24	SINT	**0: Disable** 1: Enable	Attribute 25
External configuration lock	25	SINT	**0: Disable** 1: Enable	Attribute 26

9.1.1 QuickConnect

QuickConnect (QC) enables the module to perform the start-up process faster. With the activation of this parameter, a particularly quick start-up of EtherNet/IP communication is possible.

If you enable QuickConnect, the UNITRONIC® ACCESS module accepts a TCP connection within 350 ms after being switched on. Then the control system establishes a connection. The UNITRONIC® ACCESS Digital I/O module achieves a start-up time of 400 to 500 ms.

To use QuickConnect, the network must be set up in a star or line topology and the UNITRONIC® ACCESS Digital I/O module must have a static IP address. Ring topologies and DHCP/BOOTP are not supported. Please note that there is no automatic check performed for IP addresses that are assigned more than once inside the same network.

If QuickConnect is activated, the following fix parameters for the Ethernet interface of the UNITRONIC® ACCESS Digital I/O module are set:

- 100 Mbit/s transmission speed
- Full duplex connection
- Auto-negotiation and auto-MDIX deactivated

Attention: The prerequisite for the use of QuickConnect is the adherence to a strictly prescribed procedure. The UNITRONIC® ACCESS Digital I/O modules must be notified before switch-off (inhibit instruction) and switch-on (uninhibit instruction). A hard disconnect during operation is not permitted. Details of this procedure can be found in Rockwell Automation's document "ENET-AT001C-ENP".

9.1.2 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.

9.1.3 Web interface lock

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

9.1.4 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.

9.1.5 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.

9.1.6 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.

9.1.7 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.

9.2 Channel settings

Configuration parameter	Byte offset config. assembly	Data type	Valid values	CIP object class 0xA1, Instance 1 .. 16
IO Mapping (Ch1 ..1 16)	32	SINT[16]	**01 ..1 15**: Bit number of 16 channel process data 16: Inactive	Attribute 1
DO Surveillance Timeout (Ch11 ..1 16)	48	INT[16]	01 ..1 255 **(80)**	Attribute 2
DO Failsafe (Ch11 ..1 16)	80	SINT[16]	**0: Set Low** 1: Set High 2: Hold Last	Attribute 3
DO Restart Mode (Ch1 .. 16)	96	SINT[16]	0: Disable **1: Enable**	Attribute 4
DO Current Limit (Ch1 .. 16)	112	SINT[16]	0: 0.5 A 1: 1.5 A 2: 1.5 A 3: 2.0 A **4: 2.0 A Max.**	Attribute 5
DI Logic (Ch1 .. 16)	128	SINT[16]	**0: Normally Open** 1: Normally Close	Attribute 6
DI Filter (Ch1 .. 16)	144	SINT[16]	0: Disabled 1: 1 ms 2: 2 ms **3: 3 ms** 4: 6 ms 5: 10 ms 6: 15 ms	Attribute 7

Configuration parameter	**Byte offset** **config. assembly**	**Data type**	**Valid values**	**CIP object class 0xA1, Instance 1 .. 16**
Channel Mode (Ch1 .. 16)	192	SINT[16]	0: Digital Input/Digital Output 1: Digital Output 2: Digital Input 3: Inactive **The supported Channel Mode and the default value depend on the device variant.**	Attribute 10

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.

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 Current Limit (Ch1 .. 16)

Only applicable for the following device variants:

- MP08DIO08DIO

With this parameter you can configure the current limitations for the digital outputs by selecting a DO Current limit. Output switch mode:

- 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 16 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 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 16DIO, 16DI or 8DI/8DO).

Digital Input/Digital Output:

In this mode, the channel operates as digital input/output. The channel can be controlled by the *Digital Output Channel Control* (first two bytes of the output data) and 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)

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.

Inactive:

This mode should be selected when the channel is not in use.

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. The producing data in this context is defined as the process input data which contains the physical inputs, standard diagnostics and optional extended diagnostics.

The following sections 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)	Digital output data
Consuming data size	2 Byte, INT	2 Byte, INT	4 Byte, INT

The complete *Output data frame* has a variable size of 4 Bytes. In general, a 4 Byte Run/Idle Header precedes, resulting in up to 8 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.

10.2 Producing data image (input)

Input data frame	Digital input channel status	General diagnostics	Sensor diagnostics	Actuator/U_{Aux} diagnostics
Producing data size	2 Byte, INT	2 Byte, INT	2 Byte, INT	2 Byte, INT

The complete *Input data frame* has a fixed size of 8 Bytes (6 Bytes for the 16DI variant).

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	Reserved	Reserved	SCA	SCS	LVA	LVS
	Byte 1	0	0	0	0	0	0	0	0

LVS	Low Voltage System/Sensor Supply
LVA	Low Voltage Actuator Supply
SCS	Short Circuit Sensor
SCA	Short Circuit Actuator/U_L/U_{Aux}

FME	Force Mode Enabled
IME	Internal Module 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

10.3 Producing data image (Extended diagnosis)

Input data frame	U_S voltage	U_L voltage	DO current port (X1 .. X4)	DO current port (X5 .. X8)
Producing data size	2 Byte, INT	2 Byte, INT	16 Byte, INT	16 Byte, INT

The complete *Input data frame* has a fixed size of 36 Bytes for 16DIO variants, 20 Bytes for 8DI/8DO variants and 2 Bytes for 16DI variants.

The following chapters describe the bit assignment.

Byte offset	Input data
0	U_S voltage (2 Bytes)
2	U_L voltage (2 Bytes)
4	DO current port (X1 .. X4) (16 Bytes)
20	DO current port (X5 .. X8) (16 Bytes)

10.4 Sample applications

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.

10.4.1 Process data images – default configuration

The default configuration of the digital input and output data sizes are fixed in the EDS files. This means the user gets all data of each digital channel. 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	4
Input data size	8

Byte offset	Output data
0	Digital output channel control (2 bytes)
2	Reserved (2 bytes)

Table 10: 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)

Table 11: Default input process data

10.4.2 Process data images with modified data sizes

The digital input and output data sizes are fixed. Additionally, the Extended diagnosis function can be added to the input process data. 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 as well as the byte offsets for input and output data:

Connection parameters

Output data size	4
Input data size	44

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	–	Sensor diagnostics (2 Bytes)
6	–	Actuator diagnostics (2 Bytes)
8	–	U_S voltage (2 Bytes)
10	–	U_L Voltage (2 Bytes)
12	–	DO current port X1 Ch. A (2 Bytes)
14	–	DO current port X1 Ch. B (2 Bytes)
16	–	DO current port X2 Ch. A (2 Bytes)
18	–	DO current port X2 Ch. B (2 Bytes)
20	–	DO current port X3 Ch. A (2 Bytes)
22	–	DO current port X3 Ch. B (2 Bytes)
24	–	DO current port X4 Ch. A (2 Bytes)
26	–	DO current port X4 Ch. B (2 Bytes)
28	–	DO current port X5 Ch. A (2 Bytes)
30	–	DO current port X5 Ch. B (2 Bytes)
32	–	DO current port X6 Ch. A (2 Bytes)
34	–	DO current port X6 Ch. B (2 Bytes)
36	–	DO current port X7 Ch. A (2 Bytes)
38	–	DO current port X7 Ch. B (2 Bytes)
40	–	DO current port X8 Ch. A (2 Bytes)
42	–	DO current port X8 Ch. B (2 Bytes)

Table 12: Modified process data

11 Configuration and operation with Rockwell Automation Studio 5000®

The configuration and start-up of the UNITRONIC® ACCESS 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®.

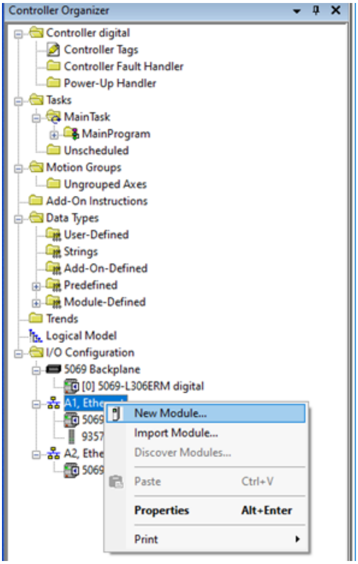
2. Select the correct controller.

3. When no integrated EtherNet/IP interface is available, add the proper communication interface to your backplane under **Controller Organizer** > **I/O-Configuration**.

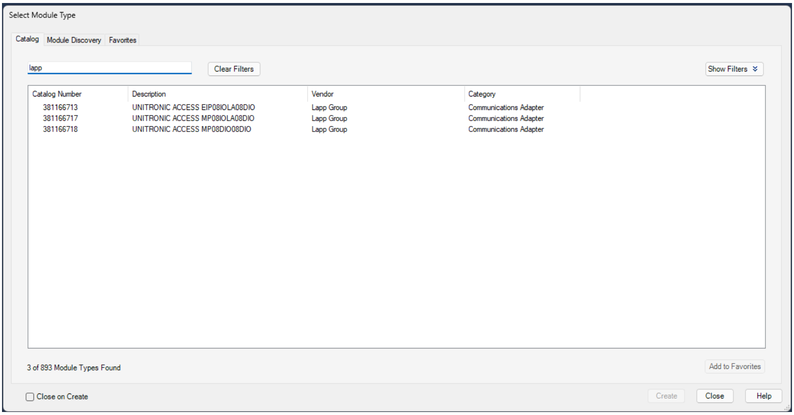
4. Set a communication path to enable the project download.

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.

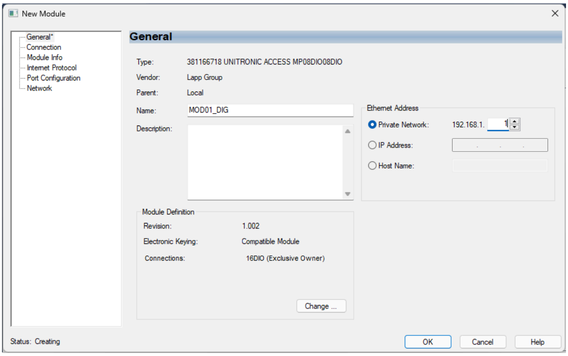


7. Select **New Module** in the menu. The following selection window opens:



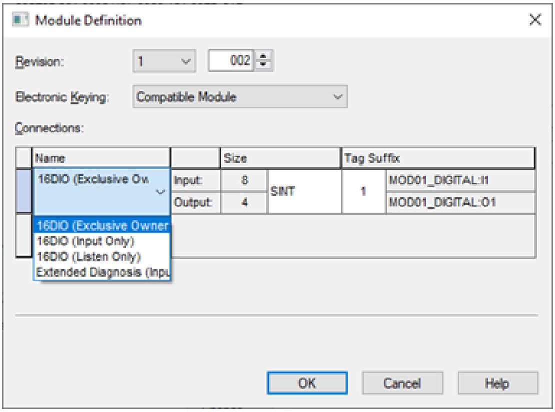
8. Use the **Module Type Vendor Filter** on the right side to display all installed devices of U.I. Lapp GmbH.

9. Select the device you wish to add and click on **Create**.



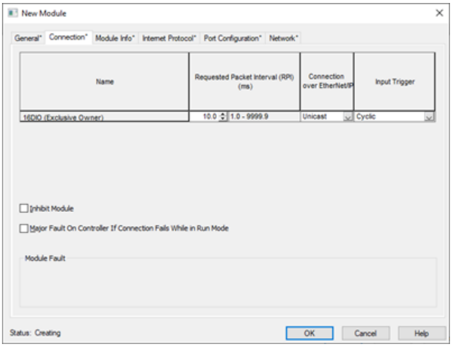
10. Enter a name for the device and set the chosen IP address. In this example, the name is **MOD01_DIGITAL** and the IP address is **192.168.1.1**.

11. Click on **Change** in order to change the settings for the device revision, electronic keying and connection type.

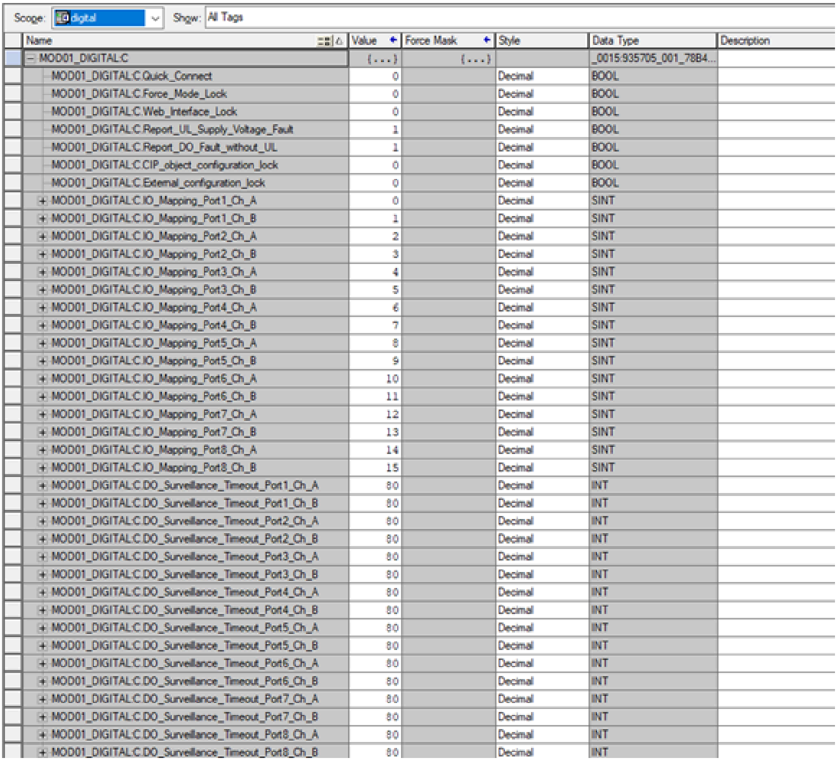


12. Select the connection type and configure the total sizes of the input and output process data. The sizes depend on the number of connected devices and their data lengths of both directions. Each device input and output data size must also be set later in the port configuration. The selection of the 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.

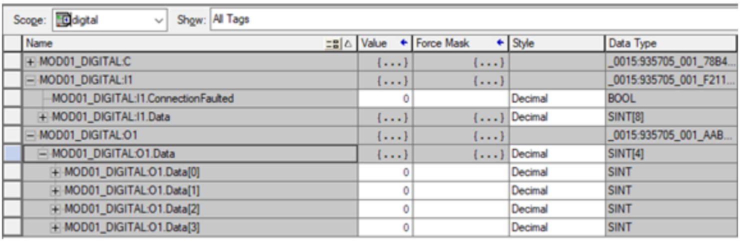


14. Move to **Controller-Tags** in **Controller Organizer**. The controller tags for the configuration parameters contain the name of the device, followed by a "**:C**". The configuration parameters can be set under **Value** and are described in chapter Configuration parameters on page 40.


Scope: digital Show: All Tags

Name	Value	Force Mask	Style	Data Type	Description
MOD01_DIGITAL:C	{...}	{...}		_0015:935705_001_78B4...	
MOD01_DIGITAL:C.Quick_Connect	0		Decimal	BOOL	
MOD01_DIGITAL:C.Force_Mode_Lock	0		Decimal	BOOL	
MOD01_DIGITAL:C.Web_Interface_Lock	0		Decimal	BOOL	
MOD01_DIGITAL:C.Report_UL_Supply_Voltage_Fault	1		Decimal	BOOL	
MOD01_DIGITAL:C.Report_DO_Fault_without_UL	1		Decimal	BOOL	
MOD01_DIGITAL:C.CIP_object_configuration_lock	0		Decimal	BOOL	
MOD01_DIGITAL:C.External_configuration_lock	0		Decimal	BOOL	
MOD01_DIGITAL:C.IO_Mapping_Port1_Ch_A	0		Decimal	SINT	
MOD01_DIGITAL:C.IO_Mapping_Port1_Ch_B	1		Decimal	SINT	
MOD01_DIGITAL:C.IO_Mapping_Port2_Ch_A	2		Decimal	SINT	
MOD01_DIGITAL:C.IO_Mapping_Port2_Ch_B	3		Decimal	SINT	
MOD01_DIGITAL:C.IO_Mapping_Port3_Ch_A	4		Decimal	SINT	
MOD01_DIGITAL:C.IO_Mapping_Port3_Ch_B	5		Decimal	SINT	
MOD01_DIGITAL:C.IO_Mapping_Port4_Ch_A	6		Decimal	SINT	
MOD01_DIGITAL:C.IO_Mapping_Port4_Ch_B	7		Decimal	SINT	
MOD01_DIGITAL:C.IO_Mapping_Port5_Ch_A	8		Decimal	SINT	
MOD01_DIGITAL:C.IO_Mapping_Port5_Ch_B	9		Decimal	SINT	
MOD01_DIGITAL:C.IO_Mapping_Port6_Ch_A	10		Decimal	SINT	
MOD01_DIGITAL:C.IO_Mapping_Port6_Ch_B	11		Decimal	SINT	
MOD01_DIGITAL:C.IO_Mapping_Port7_Ch_A	12		Decimal	SINT	
MOD01_DIGITAL:C.IO_Mapping_Port7_Ch_B	13		Decimal	SINT	
MOD01_DIGITAL:C.IO_Mapping_Port8_Ch_A	14		Decimal	SINT	
MOD01_DIGITAL:C.IO_Mapping_Port8_Ch_B	15		Decimal	SINT	
MOD01_DIGITAL:C.DO_Surveillance_Timeout_Port1_Ch_A	80		Decimal	INT	
MOD01_DIGITAL:C.DO_Surveillance_Timeout_Port1_Ch_B	80		Decimal	INT	
MOD01_DIGITAL:C.DO_Surveillance_Timeout_Port2_Ch_A	80		Decimal	INT	
MOD01_DIGITAL:C.DO_Surveillance_Timeout_Port2_Ch_B	80		Decimal	INT	
MOD01_DIGITAL:C.DO_Surveillance_Timeout_Port3_Ch_A	80		Decimal	INT	
MOD01_DIGITAL:C.DO_Surveillance_Timeout_Port3_Ch_B	80		Decimal	INT	
MOD01_DIGITAL:C.DO_Surveillance_Timeout_Port4_Ch_A	80		Decimal	INT	
MOD01_DIGITAL:C.DO_Surveillance_Timeout_Port4_Ch_B	80		Decimal	INT	
MOD01_DIGITAL:C.DO_Surveillance_Timeout_Port5_Ch_A	80		Decimal	INT	
MOD01_DIGITAL:C.DO_Surveillance_Timeout_Port5_Ch_B	80		Decimal	INT	
MOD01_DIGITAL:C.DO_Surveillance_Timeout_Port6_Ch_A	80		Decimal	INT	
MOD01_DIGITAL:C.DO_Surveillance_Timeout_Port6_Ch_B	80		Decimal	INT	
MOD01_DIGITAL:C.DO_Surveillance_Timeout_Port7_Ch_A	80		Decimal	INT	
MOD01_DIGITAL:C.DO_Surveillance_Timeout_Port7_Ch_B	80		Decimal	INT	
MOD01_DIGITAL:C.DO_Surveillance_Timeout_Port8_Ch_A	80		Decimal	INT	
MOD01_DIGITAL:C.DO_Surveillance_Timeout_Port8_Ch_B	80		Decimal	INT	

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 49.


Scope: digital Show: All Tags

Name	Value	Force Mask	Style	Data Type
MOD01_DIGITAL:C	{...}	{...}		_0015:935705_001_78B4...
MOD01_DIGITAL:I1	{...}	{...}		_0015:935705_001_F211...
MOD01_DIGITAL:I1.ConnectionFaulted	0		Decimal	BOOL
MOD01_DIGITAL:I1.Data	{...}	{...}	Decimal	SINT[8]
MOD01_DIGITAL:O1	{...}	{...}		_0015:935705_001_AAB...
MOD01_DIGITAL:O1.Data	{...}	{...}	Decimal	SINT[4]
MOD01_DIGITAL:O1.Data[0]	0		Decimal	SINT
MOD01_DIGITAL:O1.Data[1]	0		Decimal	SINT
MOD01_DIGITAL:O1.Data[2]	0		Decimal	SINT
MOD01_DIGITAL:O1.Data[3]	0		Decimal	SINT

16. When the configuration is completed, the parameters can be downloaded to the EtherNet/IP controller.

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	*Not supported on UNITRONIC® ACCESS Digital I/O module*s
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 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*

*) The available instances depend on the number of digital channels of the device variant. Up to 16 digital channels 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
1	Quick Connect	Get, Set	BOOL	0: Disable 1: Enable
2	Force Mode Lock	Get, Set	BOOL	0: Disable 1: Enable
3	Web Interface Lock	Get, Set	BOOL	0: Disable 1: Enable
4	Reserved	Get	SINT	–
5	Report UL/UAux Supply Voltage Fault	Get, Set	BOOL	0: Disable 1: Enable
6	Report DO Fault without UL/UAux	Get, Set	BOOL	0: Disable 1: Enable
7.. 24	Reserved	Get	SINT	–
25	CIP object configuration lock	Get, Set	BOOL	0: Disable 1: Enable
26	External configuration lock	Get, Set	BOOL	0: Disable 1: Enable
27.. 32	Reserved	Get	SINT	–

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	I/O 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 Current Limit	Get, Set	SINT	0: High-Side (U_L, 0.5 A) 1: High-Side (U_L, 1.0 A) 2: High-Side (U_L, 1.5 A) 3: High-Side (U_L, 2.0 A) 4: High-Side (U_L, 2.0 A max)
6**	DI Logic	Get, Set	SINT	0: Normally Open 1: Normally Close

Attribute	Name	Access	Data type	Description
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 .. 9	Reserved			
10	Channel Mode	Get, Set	SINT	0: Digital Input/Output 1: Digital Output 2: Digital Input 3: Inactive **The supported Channel Mode depends on the device variant.**

* Only available for channels supporting DO.

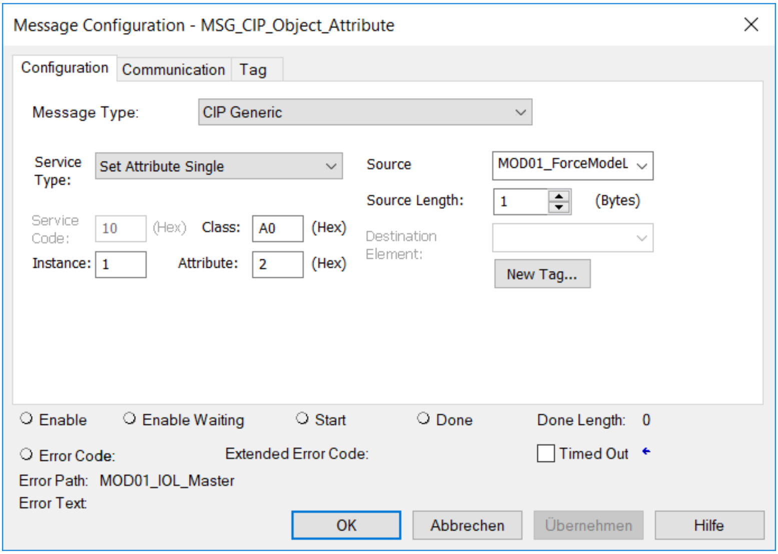
** Only available for channels supporting DI.

For information on the ports see chapter I/O port overview on page 16.

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 attributes can be defined as *Get* or *Set* in 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*:



The channels as in the *Channel Settings Object* are each assigned in ascending order to an instance ID.

Assignment of the 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

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. At least 21 V of U_S supply voltage for the Digital I/O module are required to minimize the risk of internal voltage drops in module.

The green U_S indicator LED 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	Reserved	Reserved	SCA	SCS	LVA	**LVS**
	Byte 1	0	0	0	0	0	0	0	0

LVS	Low Voltage System/Sensor Supply
LVA	Low Voltage Actuator Supply
SCS	Short Circuit Sensor
SCA	Short Circuit Actuator/U_L/U_{Aux}

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.

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:

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*:

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.

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.

14.1 MQTT

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

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

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 request. For more information see chapter MQTT configuration - Quick start guide on page 107.

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
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 92.

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

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 14: Base topic variables on page 92.

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

Variable	Description
mac	The MAC address of the device
name	The name of the device
order	The ordering number of the device
serial	The serial number of the device
ip0 ip1 ip2 ip3	IP address octets

Table 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, Device status and data.
process	All process data which is produced and consumed by the device itself or by attached devices.	Digital inputs, digital outputs, cyclic data.

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

Table 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

14.1.2.2 Publish topic

Overview of all publish JSON data for the defined topics:

Identity/gateway	
Key	**Data type**
product_name	json_string
ordering_number	json_string
device_type	json_string
serial_number	json_string
mac_address	json_string
production_date	json_string
fw_name	json_string
fw_date	json_string
fw_version	json_string
hw_version	json_string
family	json_string
location	json_string
country	json_string
fax	json_string
vendor_name	json_string
vendor_address	json_string
vendor_phone	json_string
vendor_email	json_string
vendor_techn_support	json_string
vendor_url	json_string
vendor_id	json_integer
device_id	json_integer

Table 18: Identity/gateway

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

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

Table 19: Config/gateway

Status/gateway				
Key	**Data type**	**Range**	**Default value**	**Remarks**
protocol	json_string	PROFINET: ▶ UNKNOWN ▶ OFFLINE ▶ STOP ▶ IDLE ▶ OPERATE EtherNet/IP: ▶ CONNECTED ▶ DISCONNECTED EtherCAT®: ▶ PREOP ▶ SAFEOP ▶ OP ▶ INIT ▶ UNKNOWN Modbus TCP: ▶ No Connections ▶ Connected CC-Link IE Feld Basic: ▶ ON ▶ STOP ▶ DISCONNECTED ▶ ERROR		
system_voltage_fault	json_boolean	true / false		
actuator_voltage_fault	json_boolean	true / false		
internal_module_error	json_boolean	true / false		
simulation_active_diag	json_boolean	true / false		
us_voltage	json_integer	0 .. 32		in Volts
ul_voltage	json_integer	0 .. 32		in Volts
forcemode_enabled	json_boolean	true / false		

Table 20: Status/gateway

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

Table 21: Process/gateway

Identity/port/1 .. 8				
Key	**Data type**	**Range**	**Default value**	**Remarks**
port	json_integer	1 .. 8		
type	json_string	Digital Input DIO Digital Output DIO Pin 4 Only DI Pin 4 Only DO Pin 4 Only Not available Unknown		
max_output_power_cha	json_string	2.0_mA 0.5_mA		
max_output_power_chb	json_string	2.0_mA 0.5_mA		
channel_cha	json_string	Digital Input Digital Output DIO Digital Input/Output Auxiliary Power Auxiliary with DO Not available Unknown		
channel_chb	json_string	Digital Input Digital Output DIO Digital Input/Output Auxiliary Power Auxiliary with DO Not available Unknown		

Table 22: Identity/port/1 .. 8

Config/port/1 .. 8				
Key	**Data type**	**Range**	**Default value**	**Remarks**
port	json_integer	1 .. 8		
direction_cha	json_string	Output Input Inactive Auxiliary Power DIO Unknown		
direction_chb	json_string	Output Input Inactive Auxiliary Power DIO Unknown		
restart_mode_cha	json_string	Manual Auto		
restart_mode_chb	json_string	Manual Auto		
input_polarity_cha	json_string	NO NC		
input_polarity_chb	json_string	NO NC		
input_filter_cha	json_integer			ms
input_filter_chb	json_integer			ms
do_auto_restart_cha	json_boolean	true / false		
do_auto_restart_chb	json_boolean	true / false		
failsafe_cha	json_string	set_low set_high hold_last	set_low	
failsafe_chb	json_string	set_low set_high hold_last	set_low	
surveillance_timeout_cha	json_integer	0 .. 255	80	

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

Table 23: Config/port/1 .. 8

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

Table 24: 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 26: Force object: Digital on page 104)		

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

[…]/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 28: Config object: Portmode on page 105)		
ip_address	string	"192.168.1.5"	
subnet_mask	string	"255.255.255.0"	
gateway	string	"192.168.1.100"	

Table 27: 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		

Table 28: 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 29: 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

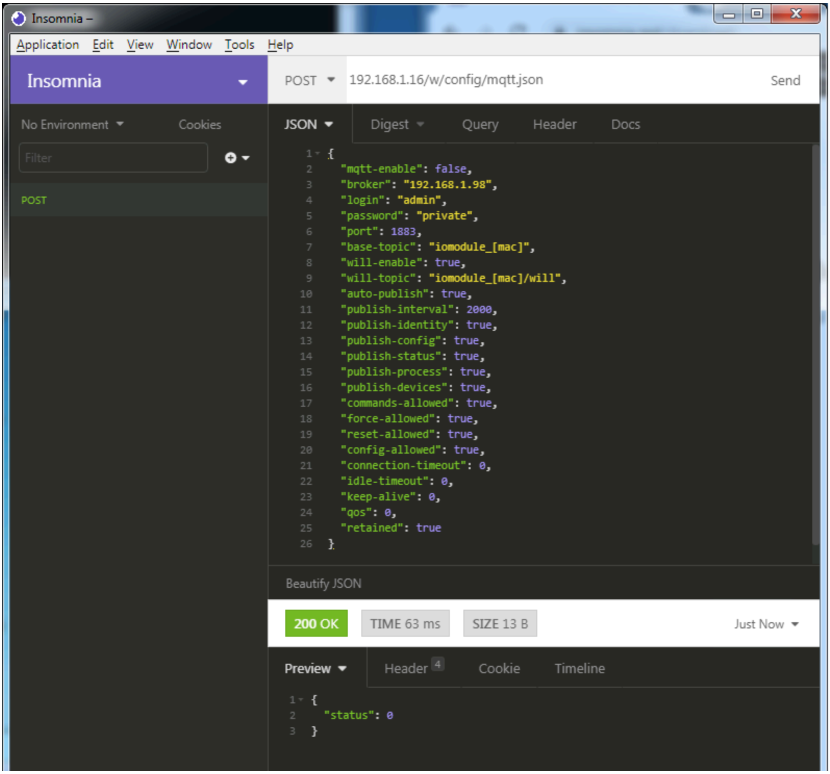
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

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/

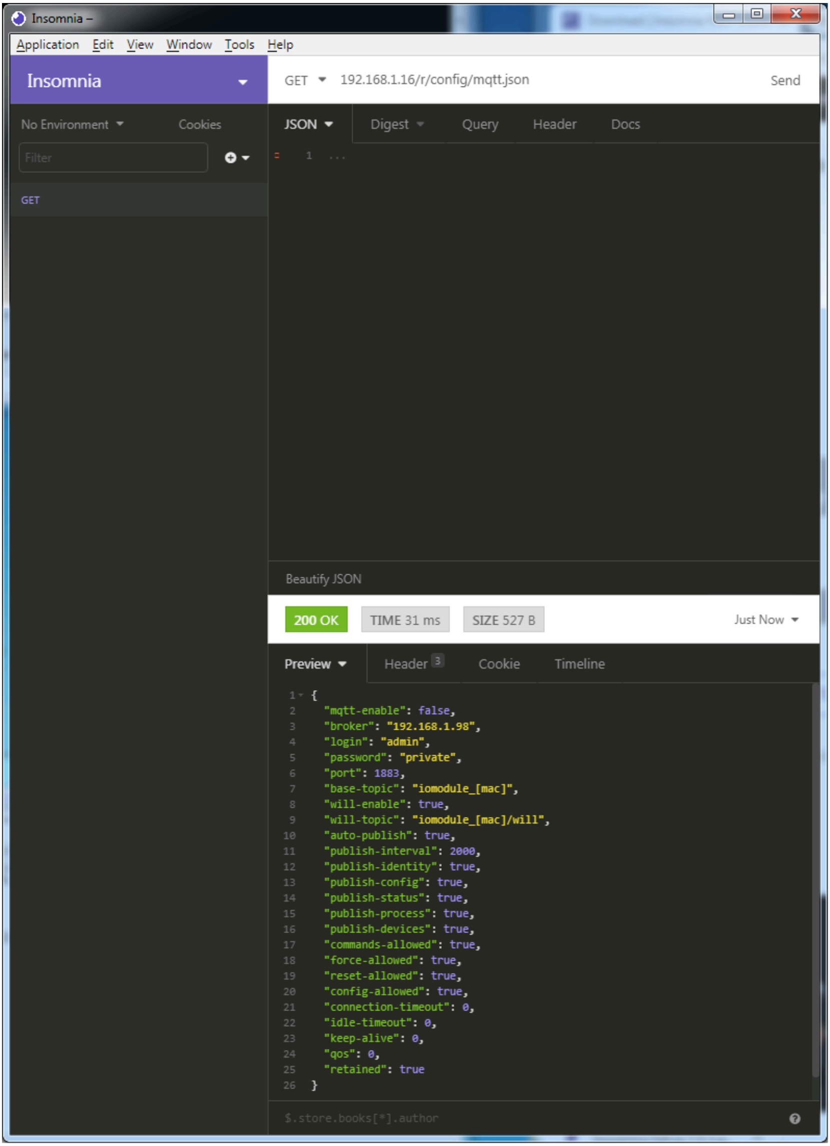
2. Configure MQTT:

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



3. Read MQTT:

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



14.2 OPC UA

OPC 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.

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 request.

The configuration URL is:

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

The configuration can also read back as a JSON file:

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

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

Tree overview of OPC UA objects:

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

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

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

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

Response:

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

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

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

Examples:

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

{"status": 0}

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

14.2.1.1 Gateway objects

Identity

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

Status (read)

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

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

Forcing (read)

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

Config (read + write)

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

Process (read)

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

Commands (write)

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

14.2.1.2 Ports objects

Identity

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

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

See details in Channel objects on page 116.

Status (read)

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

Config (read + write)

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

14.2.1.3 Channel objects

Identity (read)

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

Status (read)

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

Config (read + write)

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

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

Process (read)

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

Forcing (read + write)

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

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

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

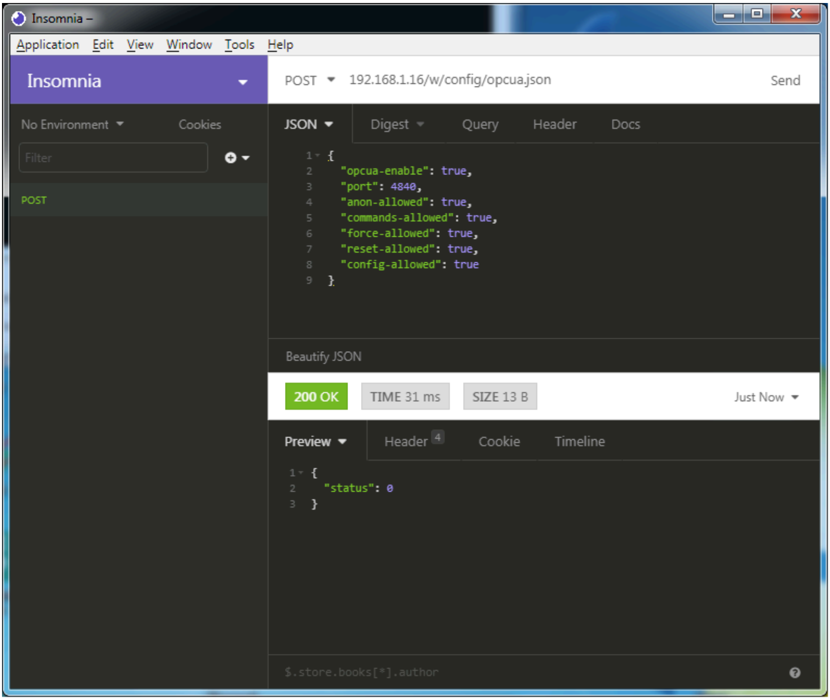
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

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/

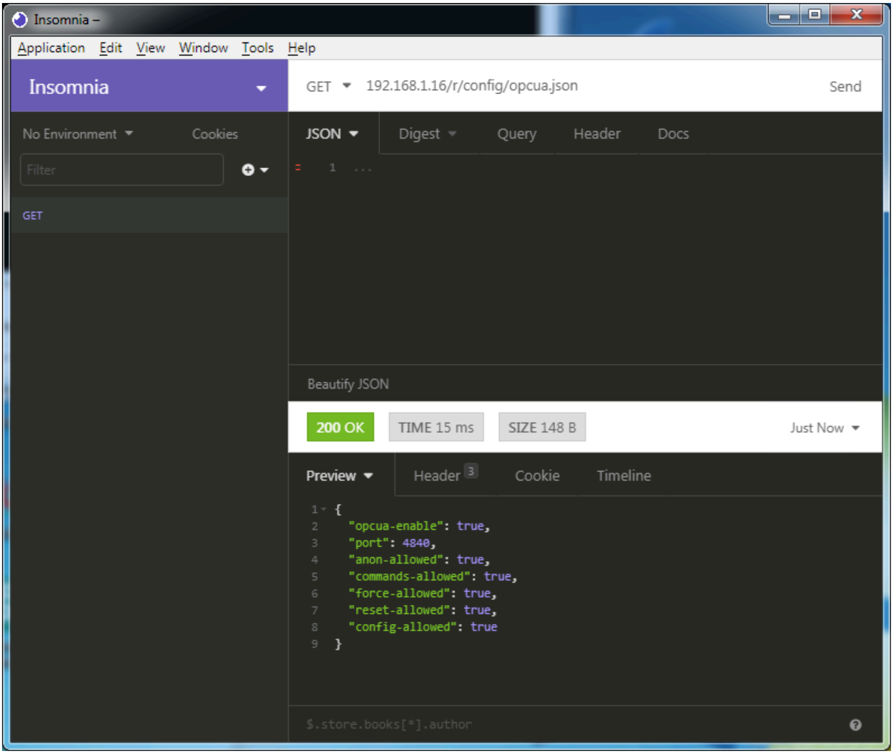
2. Configure OPC UA:

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 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.

The customized LAPP REST API 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.

14.3.2 Structure

Name	Data type	Description	Example
name	string	Device name	"MP08DIO08DIO "
order-id	string	Ordering number	"381166718"
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
CHANNEL Object			
name	string	Name of channel	
type	number	Hardware channel type as number: 0 = DIO 1 = Input 2 = Output 3 = Input/Output 4 = Channel not available 5 = Channel not available 6 = Channel not available 7 = Channel not available 8 = Channel not available	
type_text	string	Textual representation of the channel type	
config	number	Current configuration of the channel: 0 = DIO 1 = Input 2 = Output 3 = Channel not available 4 = Deactivated 5 = Channel not available	
config_text	string	Textual representation of the current config	
inputState	boolean	Input data (producing data) bit to the PLC	
outputState	boolean	Output data bit to the physical output pin	
forced	boolean	True, if the output pin of this channel is forced	
simulated	boolean	True, if the input value to the PLC of this channel is simulated	
actuatorDiag	boolean	True, if the output is in short circuit / overload condition	
sensorDiag	boolean	True, if the sensor supply (Pin 1) is in short circuit / overload condition	

Name	Data type	Description	Example
maxOutputCurrent_mA	number	Maximum output current of the output in mA	
current_mA	number	Measured current of the output in mA (if current measurement is available)	
voltage_mV	number	Measured voltage of this output in mV (if voltage measurement is available)	
PORT Object			
port_type	string	Textual representation of the port type	
aux_mode	number	Indicates the configured mode for the Pin 2: 0 = No AUX 1 = AUX output (always on) 2 = Digital output (can be controlled by cyclic data) 3 = Digital input	
aux_text	string	Textual representation of the current aux mode	"AUX Output"
ds_fault	number	Data storage error number	
ds_fault_text	string	Textual data storage error.	
diag	array of DIAG (n)	Array of port related events	
DIAG Object			
error	number	Error code	
source	string	Source of the current error.	"device" "master"
message	string	Error message	"Supply Voltage fault"
FORCING Object		Forcing information of the device	
forcingActive	boolean	Force mode is currently active	
forcingPossible	boolean	True, if forcing is possible and force mode can be activated	
AuthPossible	boolean	True, if the JSON Interface can obtain forcing autorization	
ownForcing	boolean	True, if forcing is performed by REST API at the moment	
currentClient	string	Current forcing client identifier	

Name	Data type	Description	Example
digitalOutForced	array of numbers (2)	The force values of all 16 digital output channels.	
digitalOutMask	array of numbers (2)	The forcing mask of all 16 digital output channels.	
digitalInForced	array of numbers (2)	The force values of all 16 digital input channels.	
digitalInMask	array of numbers (2)	The forcing mask of all 16 digital input channels.	

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)		

Table 30: 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", "off", "aux"	
inlogica	string	"no","nc"	
inlogicb	string	"no","nc"	

Table 31: 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 32: Digital object

14.4 CoAP server

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

CoAP is based on the Server-Client principle and a service layer protocol that lets nodes and machines communicate with each other. The 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 131.

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 33: 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	

Table 34: REST API access via CoAP

14.4.3 CoAP configuration - Quick start guide

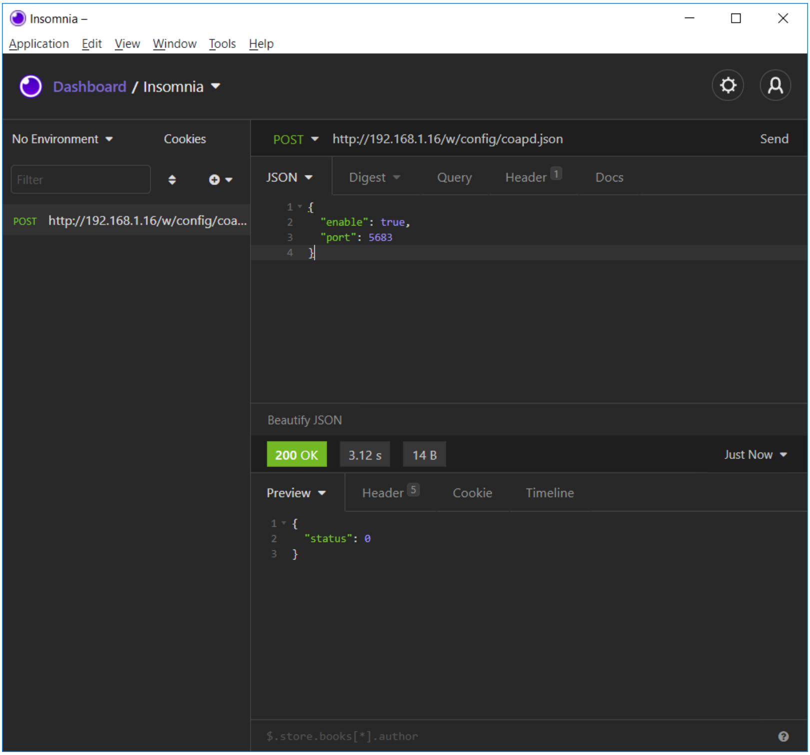
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

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/

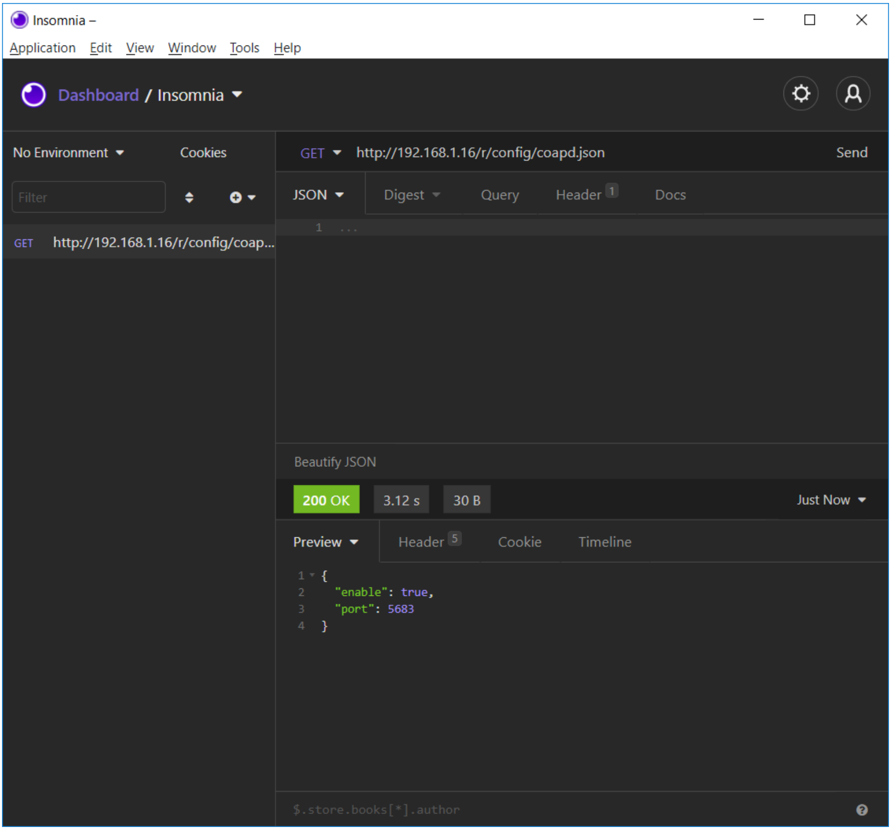
2. Configure CoAP:

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



3. Read CoAP configuration:

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



14.5 Syslog

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 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 136.

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 35: 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

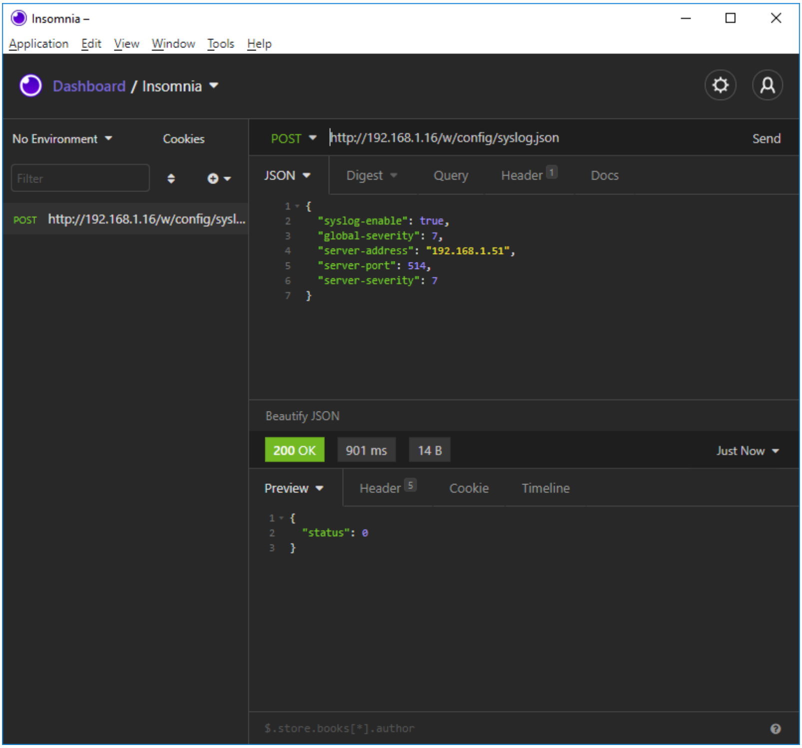
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

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/

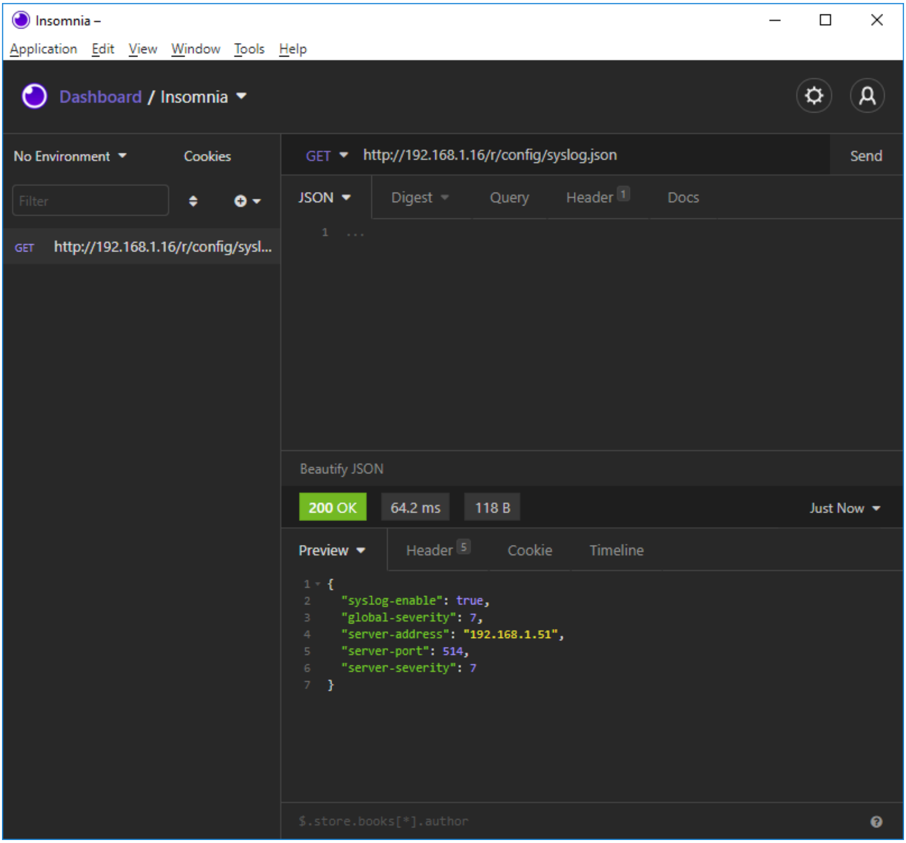
2. Configure Syslog:

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 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 140.

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 36: 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

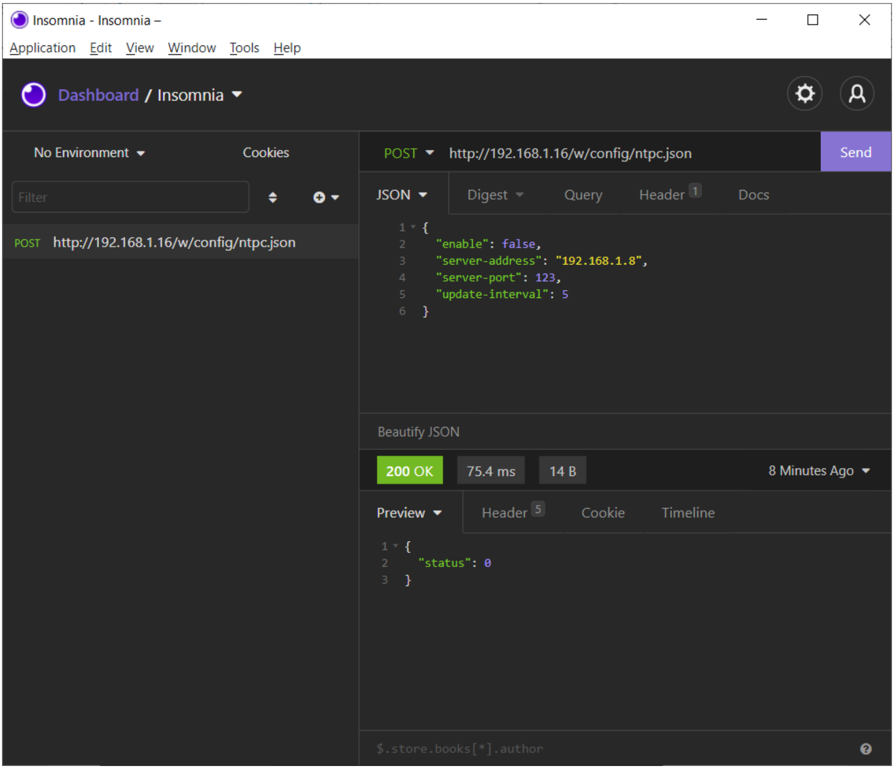
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

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/

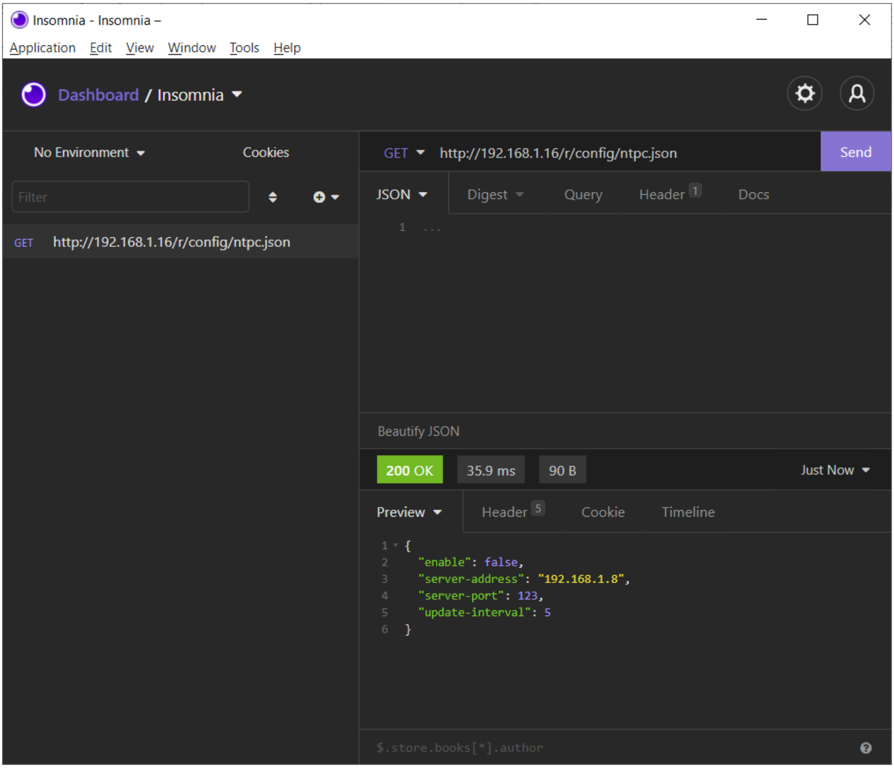
2. Configure NTP:

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



3. Read NTP configuration:

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



15 The integrated Web server

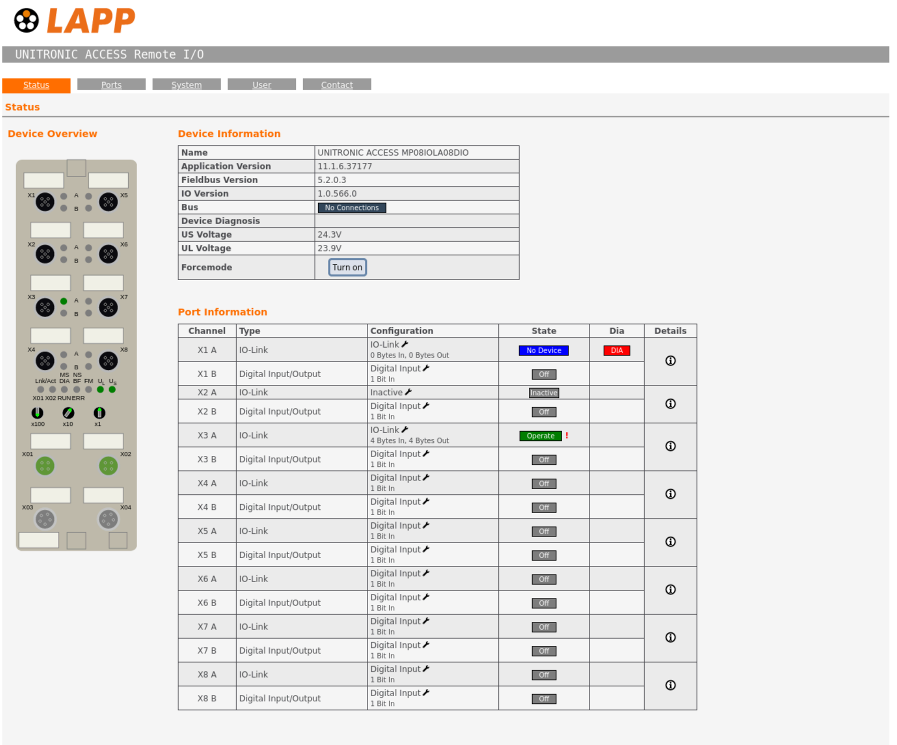
All device variants are equipped with an integrated Web server which makes functions for the device configuration and the display of status and diagnostic information available via a Web interface.

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://" 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


LAPP

UNITRONIC ACCESS Remote I/O

Status | Ports | System | User | Contact

Status

Device Overview

Device Information

Name	UNITRONIC ACCESS MP08IOLA08DIO
Application Version	11.1.6.37177
Fieldbus Version	5.2.0.3
IO Version	1.0.566.0
Bus	No Connections
Device Diagnosis	
US Voltage	24.3V
UL Voltage	23.9V
Forcemode	Turn on

Port Information

Channel	Type	Configuration	State	Dia	Details
X1 A	IO-Link	IO-Link 0 Bytes In, 0 Bytes Out	No Device	DIA	ⓘ
X1 B	Digital Input/Output	Digital Input 1 Bit In	Off		
X2 A	IO-Link	Inactive	Inactive		ⓘ
X2 B	Digital Input/Output	Digital Input 1 Bit In	Off		
X3 A	IO-Link	IO-Link 4 Bytes In, 4 Bytes Out	Operate !		ⓘ
X3 B	Digital Input/Output	Digital Input 1 Bit In	Off		
X4 A	IO-Link	Digital Input 1 Bit In	Off		ⓘ
X4 B	Digital Input/Output	Digital Input 1 Bit In	Off		
X5 A	IO-Link	Digital Input 1 Bit In	Off		ⓘ
X5 B	Digital Input/Output	Digital Input 1 Bit In	Off		
X6 A	IO-Link	Digital Input 1 Bit In	Off		ⓘ
X6 B	Digital Input/Output	Digital Input 1 Bit In	Off		
X7 A	IO-Link	Digital Input 1 Bit In	Off		ⓘ
X7 B	Digital Input/Output	Digital Input 1 Bit In	Off		
X8 A	IO-Link	Digital Input 1 Bit In	Off		ⓘ
X8 B	Digital Input/Output	Digital Input 1 Bit In	Off		

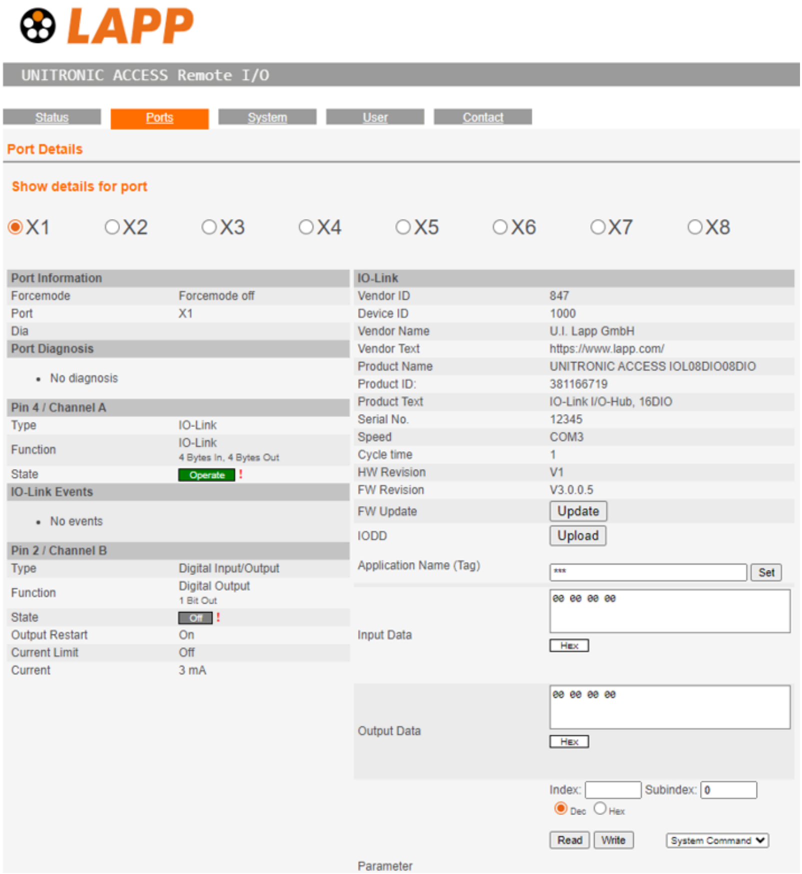
The status page provides a quick overview of the current state of the device.

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.

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.

Restart Device

The module initializes a software reset.

Reset to Factory Settings

The module restores to the default factory settings.

IP Settings

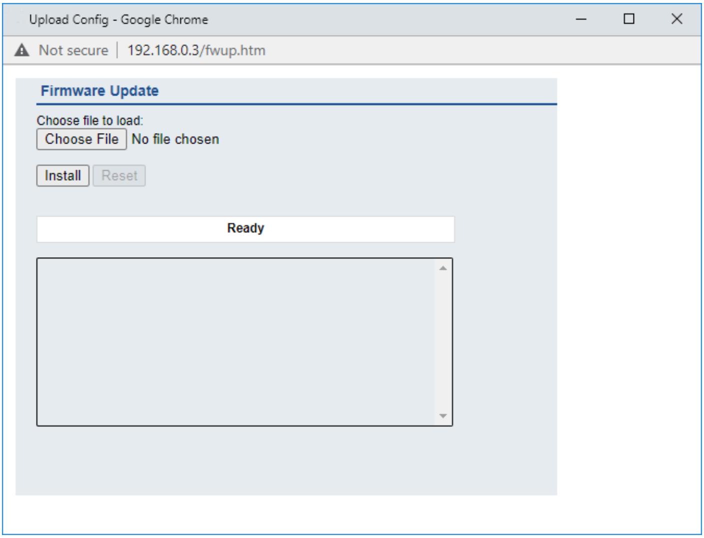
Use this parameter to change the current IP address of the module.

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.

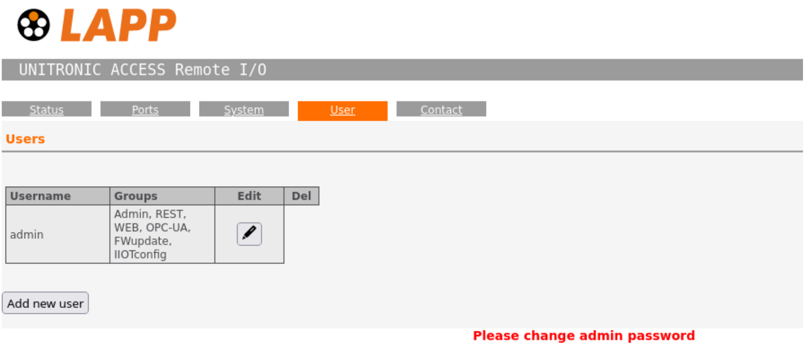
Firmware Update

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.



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

16 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.

16.1 General

Protection class (Only applies if the connectors are screwed together or if protective caps are used.)[1]	IP65 IP67 IP69K	
Ambient temperature (during operation and storage)	MP08DIO08DIO	-40 °C .. +70 °C (-40 °F .. +158 °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 37: General information

1 Not under UL investigation.

16.2 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	41000 (MP08DIO08DIO, 381166718)
Supported Ethernet protocols	Ping ARP HTTP 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 38: EtherNet/IP protocol

16.3 Power supply of the module electronics/ sensors

Port X03, X04	M12-L-coded Power, connector/socket, 5-pole Pin 1 / Pin 3		
Nominal voltage U_S	24 V DC (SELV/PELV)		
Current U_S	Max. 16 A		
Voltage range	21 .. 30 V DC		
Power consumption of module electronics	Typically 160 mA (+/-20 % at U_S nominal voltage)		
Power supply interruption	Max. 10 ms		
Voltage ripple U_S	Max. 5 %		
Current consumption sensor system (Pin 1)	MP08DIO08DIO	Port X1 .. X8 (Pin 1)	max. 4 A per port (at $T_{ambient}$ = 30° C)
Voltage level of the sensor power supply	Min. (U_S – 1.5 V)		
Short circuit/overload protection of sensor supply	Yes, per port		
Reverse polarity protection	Yes		
Operational indicator (U_S)	LED green:	18 V (+/- 1 V) < U_S	
	LED red:	U_S < 18 V (+/- 1 V)	

Table 39: Information on the power supply of the module electronics/ sensors

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

- Max. 4.0 A per port
- Max. 5.0 A for each port pair X1/X2, X3/X4, X5/X6, X7/X8

- Max. 9.0 A in total for the whole port group X1 .. X8

  Pay attention to the derating!

16.4 Power supply of the actuators

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

Table 40: Information on the power supply of the actuators

16.5 I/O ports

MP08DIO08DIO	Ports X1 .. X8	DI, DO	M12 socket, 5-pin

Table 41: I/O ports: Overview of functions

16.5.1 Digital inputs

Input connection	MP08DIO08DIO		Type 3 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	MP08DIO08DIO	X1 .. X8	16
Status indicator	yellow LED for Channel A (Pin 4) white LED for Channel B (Pin 2)		
Diagnostic indicator	red LED per port		

Table 42: I/O ports configured as digital input

16.5.2 Digital outputs

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

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

Pay attention to the derating!

Output type	normally open, p-switching	
Nominal output voltage per channel	Signal status “1” Signal status “0”	min. (U_L -1 V) max. 2 V
Max. output current per device	MP08DIO08DIO	9 A
Max. output current per channel	MP08DIO08DIO (X1 .. X8)	2 A
Short-circuit/overload protected	yes/yes	
Behavior in case of short circuit or overload	deactivation with automatic power-on (parameterized)	
Number of digital outputs	MP08DIO08DIO (X1 .. X8)	16
Status indicator	yellow LED per output Channel A (Pin 4) white LED per output Channel B (Pin 2)	
Diagnostic indicator	red LED per channel	

Table 43: I/O ports configured as digital output

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

16.6 LEDs

LED	Color	Description
U_L	Green	Auxiliary sensor/actuator voltage OK 18 V (+/- 1 V) < U_L < 30 V (+/- 1 V)
	Red*	Auxiliary sensor/actuator voltage LOW U_L < 18 V (+/- 1 V) or U_L > 30 V (+/- 1 V) * if "Report U_L supply voltage fault" is enabled.
	OFF	None of the above conditions.
U_S	Green	System/sensor voltage OK 18 V (+/- 1 V) < U_S < 30 V (+/- 1 V)
	Red	System/sensor voltage LOW U_S < 18 V (+/- 1 V) or U_S > 30 V (+/- 1 V)
	Red flashing	Device performs a factory reset (position of rotary encoding switches: 9-7-9)
	OFF	None of the above conditions.
X1 .. X8 A	Yellow	Status of digital input or digital output on pin 4 line "on".
	Red	Short circuit on pin 4 line. / Overload or short circuit on L+ (pin 1) line / communication error
	OFF	None of the above conditions.
X1 .. X8 B	White	Status of digital input or digital output on pin 2 line "on".
	Red	Short circuit on pin 2 line. / 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 double flash	Firmware update
	OFF	None of the above conditions.

Table 44: Information on the LED colors

16.7 Data transfer times

The following tables give an overview of the internal data transfer times of UNITRONIC® ACCESS.

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

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

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

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

Use case 1:

UNITRONIC® ACCESS Digital-I/O configuration with enabled Web interface and *disabled* IIoT protocols

16DIO variant (MP08DIO08DIO):

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

Use case 2:

UNITRONIC® ACCESS Digital-I/O configuration with enabled Web interface and *enabled* IIoT protocols

16DIO variant (MP08DIO08DIO):

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

17 Accessories

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

https://www.lapp.com

Manual

EtherCAT®

UNITRONIC ACCESS Digital-I/O multi-protocol: MP08DIO08DIO (16 × Input/Output)

Contents

1 About this manual

1.1 General information

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

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

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

U.I. Lapp GmbH

Schulze-Delitzsch-Straße 25
D-70565 Stuttgart
Germany

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	09/2024	

Table 1: Overview of manual revisions

2 Safety instructions

2.1 Intended use

The products described in this manual are decentralized I/O Devices on an Industrial Ethernet Network.

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

The modules fulfill the requirements of the EMC guidelines (89/336/EEC, 93/68/EEC and 93/44/EEC) and the low voltage guideline (73/23/EEC).

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

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

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

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

Only install cables and accessories that fulfill the requirements and regulations for safety, electromagnetic compatibility, and, where applicable, telecommunication end devices, as well as the specification information. Information on which cables and accessories are permitted for the installation can be obtained from 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 this manual are authorized to install and maintain the devices described. These are persons who

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

Only 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 manual, can result in serious personal injury or damage to equipment.

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

3 Designations and synonyms

AOI	Add-On Instruction
API	Application Programming Interface
BF	Bus Fault LED
Big Endian	Data format with High-B on first place (PROFINET)
BUI	Back-Up Inconsistency (EIP diagnostics)
CC	CC-Link IE Field
Ch. A	Channel A (Pin 4) of I/O port
Ch. B	Channel B (Pin 2) of I/O port
CIP	Common Industrial Protocol (media independent protocol)
CoAP	Constrained Application Protocol
CSP+	Control & Communication System Profile Plus
DCP	Discovery and Configuration Protocol
DevCom	Device Comunicating (EIP diagnostics)
DevErr	Device Error (EIP diagnostics)
DI	Digital Input
DIA	Diagnostic LED
DO	Digital Output
DIO	Digital Input/Output
DTO	Device Temperature Overrun (EIP diagnostics)
DTU	Device Temperature Underrun (EIP diagnostics)
DUT	Device under test
EIP	EtherNet/IP
ERP	Enterprise Resource Planning system
ETH	ETHERNET
FE	Functional Earth
FME	Force Mode Enabled (EIP diagnostics)
FSU	Fast Start-Up

GSDML	General Station Description Markup Language
High-B	High-Byte
ICT	Invalid Cycle Time (EIP diagnostics)
IIoT	Industrial Internet of Things
ILE	Input process data Length Error (EIP diagnostics)
IME	Internal Module Error (EIP diagnostics)
I/O	Input / Output
I/O port	X1 .. X8
I/O port pin 2	Channel B of X1 .. X8
I/O port pin 4 (C/Q)	Channel A of X1 .. X8
IVE	IO-Link port Validation Error (EIP diagnostics)
I&M	Identification & Maintenance
JSON	JavaScript Object Notation (platform independent data format)
L+	I/O port pin 1, sensor power supply
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
OLE	Output process data Length Error (EIP diagnostics)
OPC UA	Open Platform Communications Unified Architecture (platform independent, service-oriented architecture)

PLC	Programmable Logic Controller
PN	PROFINET
PWR	Power
REST	REpresentational State Transfer
RFC	Request for Comments
RPI	Requested Packet Interval
RWr	Word data input as seen from the master station (CC-Link)
RWw	Word data output as seen from the master station (CC-Link)
RX	Bit data input as seen from the master station (CC-Link)
RY	Bit data output as seen from the master station (CC-Link)
SCA	Short Circuit Actuator/U_L/U_{AUX} (EIP diagnostics)
SCS	Short Circuit Sensor (EIP diagnostics)
SLMP	Seamless Message Protocol
SNMP	Simple Network Management Protocol
SP	Single Protocol (PROFINET, EtherNet/IP, EtherCAT®, Modbus TCP or CC-Link IE Field Basic)
SPE	Startup Parameterization Error (EIP diagnostics)
U_{AUX}	$U_{Auxiliary}$, supply voltage for the load circuit (Actuator supply on Class B ports)
UDP	User Datagram Protocol
UDT	User-Defined Data Types
UINT8	Byte in PLC (IB, QB)
UINT16	Unsigned integer with 16 bits or word in PLC (IW, QW)
U_L	U_{Load}, supply voltage for the load circuit (Actuator supply on Class A)
UL	Underwriters Laboratories Inc. (certification company)
UTC	Coordinated Universal Time (Temps Universel Coordonné)

Table 2: Designations and synonyms

4 System description

4.1 Device variants

The following Digital I/O device variants are available in the UNITRONIC® ACCESS family:

Article number	Product designation	Description	I/O port functionality
381166718	MP08DIO08DIO	UNITRONIC® ACCESS M12-60 mm, I/O Device Multi-protocol (PN, EIP, EC, MB, CC) Security	16 x Input/Output universal

Table 3: Overview of UNITRONIC® ACCESS Digital-I/O variants

4.2 I/O port overview

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

UNITRONIC® ACCESS 16DIO ports

Device variant	Port	Pin 1 U_S	Pin 4 / Ch. A (In/Out)		Pin 2 / Ch. B (In/Out)	
MP08DIO08DIO	**Info:**	–	Type 3	Supply by U_L	Type 3	Supply by U_L
	X8:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)
	X7:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)
	X6:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)
	X5:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)
	X4:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)
	X3:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)
	X2:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)
	X1:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)

Table 4: Port configuration of MP08DIO08DIO variant

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

Alarm and diagnostic messages

The devices support extended EtherCAT® diagnostic emergency messages.

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

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

5.2 Integrated Web server

Network parameter display

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

Displaying diagnostics

View diagnostics via the integrated Web server.

User management

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

5.3 Security features

Firmware signature

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

Syslog

The 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.

5.4 Other features

Interface protection

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

For more details, see section Port assignments on page 24.

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 Digital-I/O multi-protocol variants

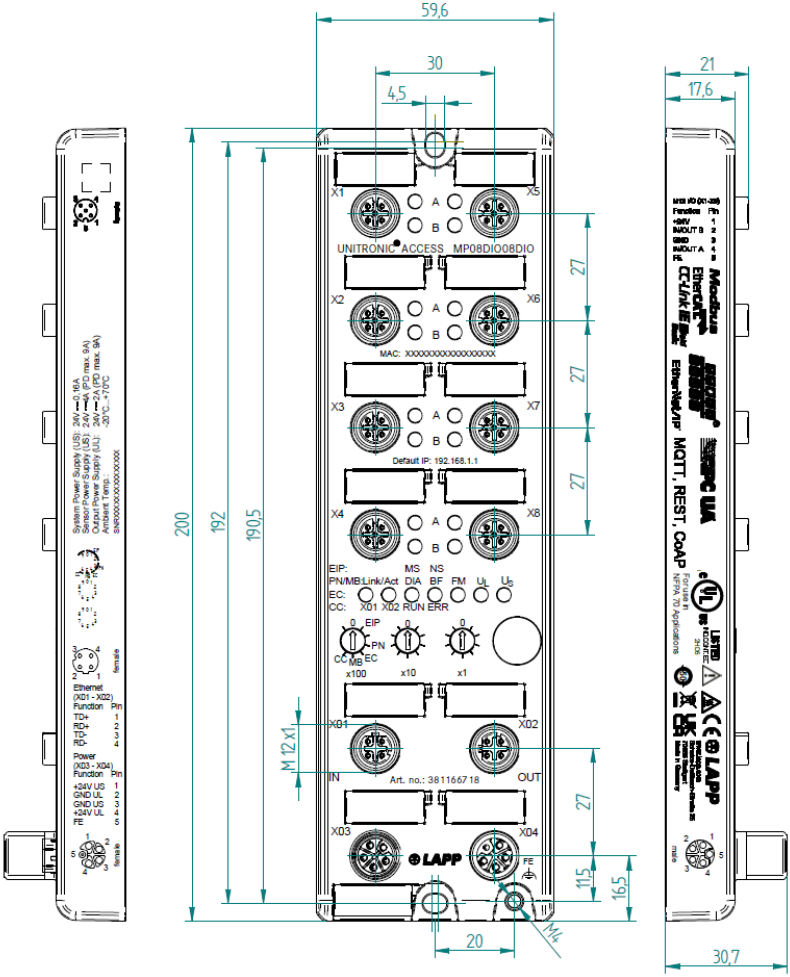


Figure 1: MP08DIO08DIO

6.2.2 Notifications

Attention:

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

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

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

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

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

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

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

6.3 Port assignments

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

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

Color coding: green

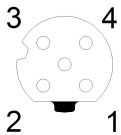


Figure 2: Schematic drawing, ports X01, X02

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

Table 5: Assignment of ports X01, X02

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

6.3.2 Power supply with M12 power L-coded

Color coding: gray

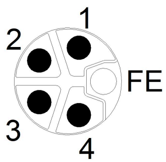


Figure 3: Schematic diagram of the M12 L-coding (connector X03 for Power In)

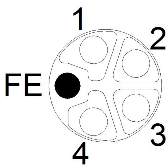


Figure 4: Schematic diagram of the M12 L-coding (socket X04 for Power Out)

Power supply	Pin	Signal	Function
	1	U_S (+24 V)	Sensor/system power supply
	2	GND_U_L	Ground/reference potential U_L
	3	GND_U_S	Ground/reference potential U_S
	4	U_L (+24 V)	Load supply Actuator supply
	5	FE	Functional ground

Table 6: Pin assignments ports X03 and X04

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

6.3.3 I/O ports as M12 sockets

Color coding: black

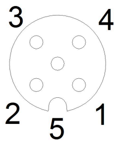


Figure 5: Schematic drawing I/O port as M12 socket

6.3.3.1 I/O ports

MP08DIO08DIO	Pin	Signal	Function
16DIO X1 .. X8	1	+24 V	power supply +24 V
	2	IN/OUT	Ch. B: Digital input or digital output
	3	GND	Ground/reference potential
	4	IN/OUT	Ch. A: Digital input or digital output
	5	FE	Functional ground

Table 7: Pin assignments I/O ports

7 Starting operation

7.1 ESI file

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

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

The ESI file is named **U.I.Lapp-UnitronicAccess-Digital-IO.xml**.

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

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

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

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

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

7.2 MAC addresses

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

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

7.3 Setting the rotary encoding switches

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

- MP08DIO08DIO

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.

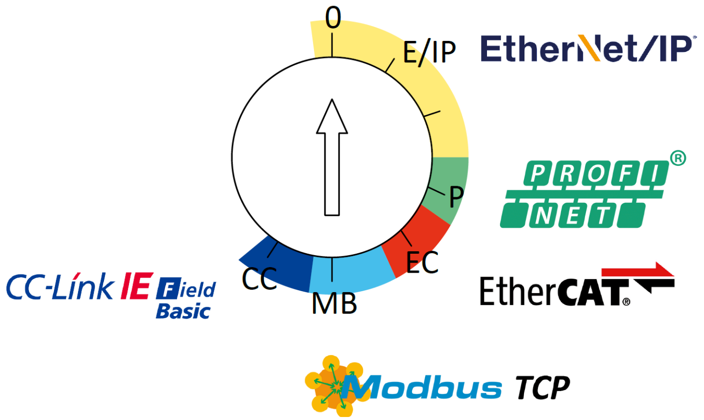
After selecting a protocol followed by a restart of the device, the new protocol is initialized. This can take up to 15 seconds. During this time the device is not usable and the LED indicators are out of function. When the protocol change is complete, the LED indicators return to normal operation and 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 Digital I/O devices 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 I/O Device in different environments.

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

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



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

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

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

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

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

Once you have set the protocol using the rotary encoding switches, the device stores this setting when it starts in cyclic communication. Changing the protocol using the rotary encoding switch is no longer possible after this point. The device will always start using the stored protocol from that point on. The IP address can be changed depending on the selected protocol.

To change the protocol, carry out a factory reset. In this way you restore the factory settings of the respective device. How you perform the factory reset for your device is described in chapter Factory reset on page 31.

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

7.3.1 Factory reset

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

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

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

	x100	**x10**	**x1**
Factory Reset	9	7	9

Follow the steps from section Setting the rotary encoding switches on page 28 again to select a new protocol.

For performing a factory reset via software configuration, see chapter OPC UA configuration on page 87 and the configuration section.

8 Configuration and operation in TwinCAT® 3

8.1 PDO assignments

Only applicable for variant MP08DIO08DIO (16 x Input/Output).

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

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

8.1.1 Input data

PDO 0x1A00 (TxPDO Mapping 2 Byte)

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

PDO 0x1A01 (TxPDO Mapping 16 Bits)

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

PDO 0x1A10 (TxPDO Mapping 1 Byte)

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

PDO 0x1A11 (TxPDO Mapping 8 Bits)

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

PDO 0x1A04 (TxPDO Error Register)

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

PDO 0x1A05 (TxPDO Diagnostic Register)

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

PDO 0x1A81 (TxPDO U_S/U_L measurements)

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

PDO 0x1A82 (TxPDO current measurements)

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

8.1.2 Output data

PDO 0x1600 (RxPDO Mapping 2 Byte)

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

PDO 0x1601 (RxPDO Mapping 16 Bits)

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

PDO 0x1610 (RxPDO Mapping 1 Byte)

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

PDO 0x1611 (RxPDO Mapping 8 Bits)

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

8.1.3 Modular slots

Only applicable for variant MP08DIO08DIO (16 x Input/Output).

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

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

8.2 Device parameters

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

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

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

8.2.1 Failsafe mode for the digital output

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

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

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

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

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

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

8.2.2 General device settings

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

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

8.2.3 Surveillance timeout

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

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

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

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

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

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

8.2.4 Digital input logic

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

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

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

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

8.2.5 Digital input filter

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

Example: Value "100" = 10 ms

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

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

8.2.6 Digital Output restart

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

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

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

8.2.7 I/O mapping configuration

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

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

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

8.2.8 Output current limit

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

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

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

8.3 Configuration example with TwinCAT® 3

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

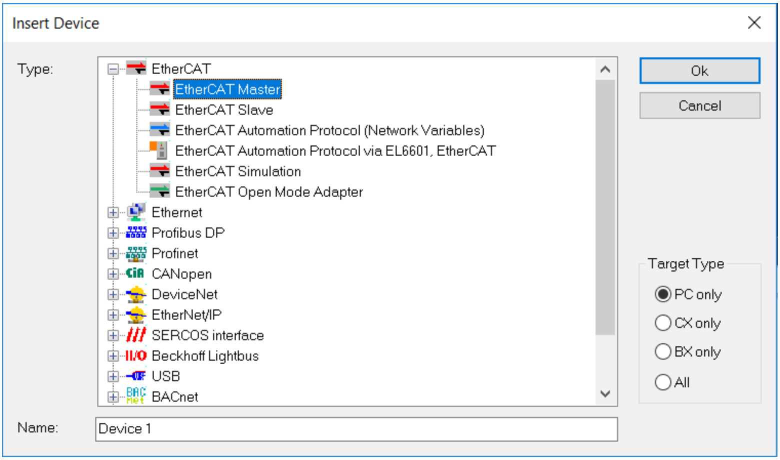
1. Install the ESI file of the device family in TwinCAT®. For TwinCAT®, the ESI file must be copied into the installation folder, e.g.: `C:\TwinCAT\3.1\Config\Io\EtherCAT`.

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

The devices are now available in the hardware catalog.

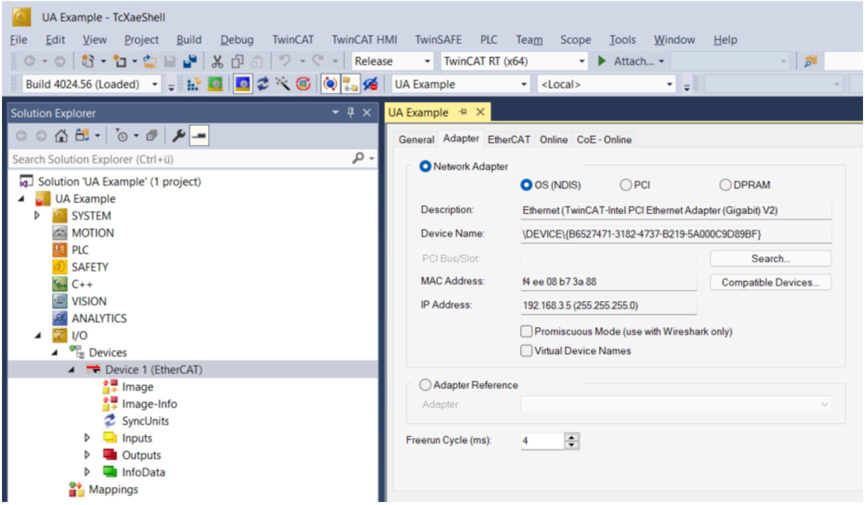
3. Start TwinCAT® and open a new project.

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



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

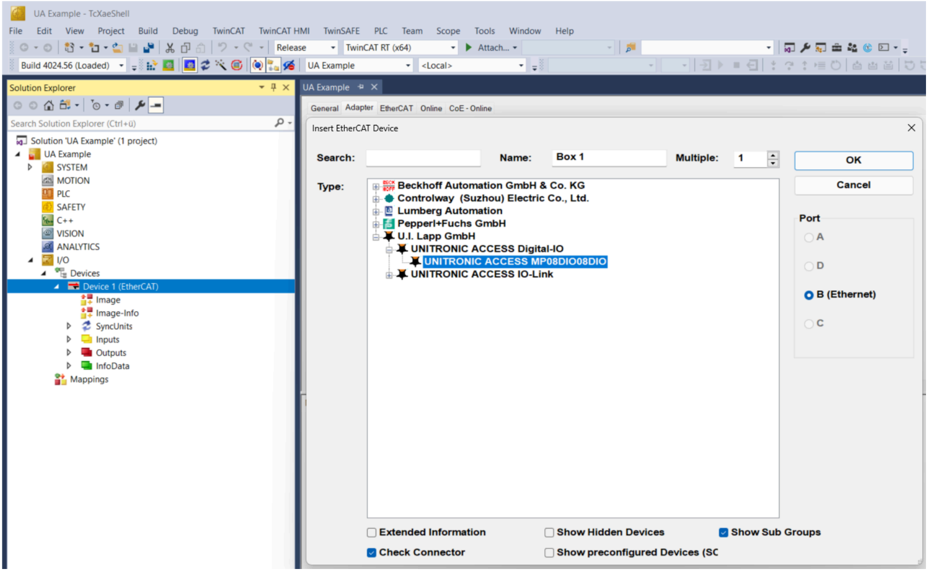
Browse to **Adapter** in the right workspace window and click on **Compatible Devices...** to choose the driver and start the installation.



8.3.1 Configuration of MP08DIO08DIO devices

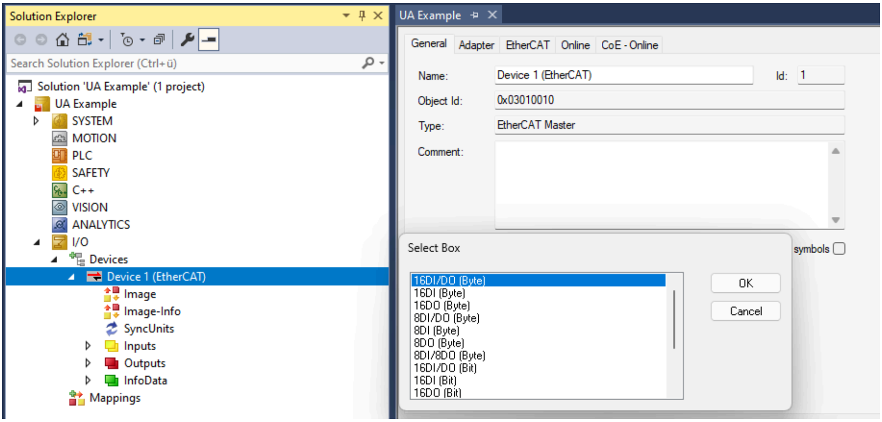
1. Select the I/O device from the hardware catalog:

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



2. Configure the “Slots”:

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



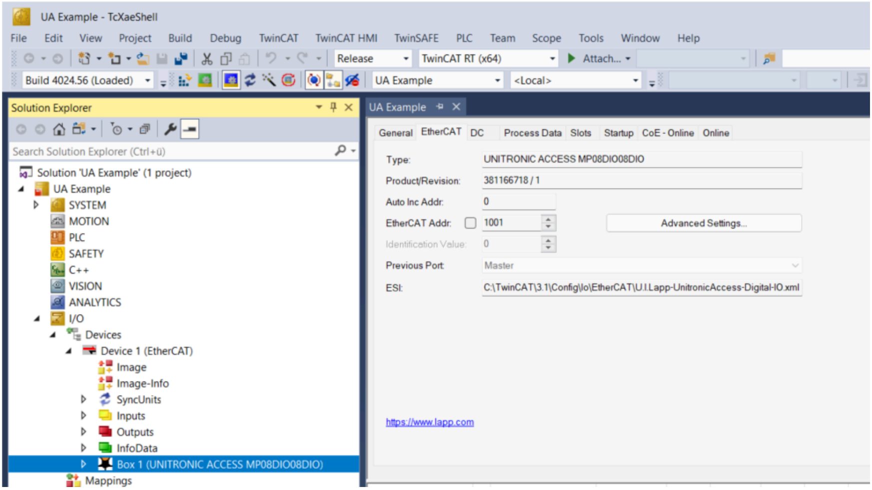
3. Configure the Process Data:

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

8.3.2 EoE IP address

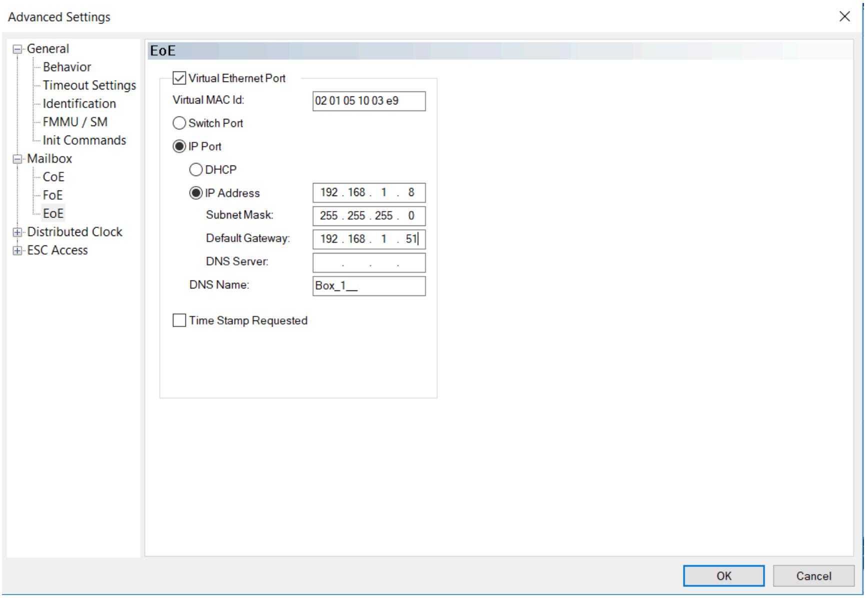
1. Set the IP address for the EoE (Ethernet over EtherCAT®) protocol:

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



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

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



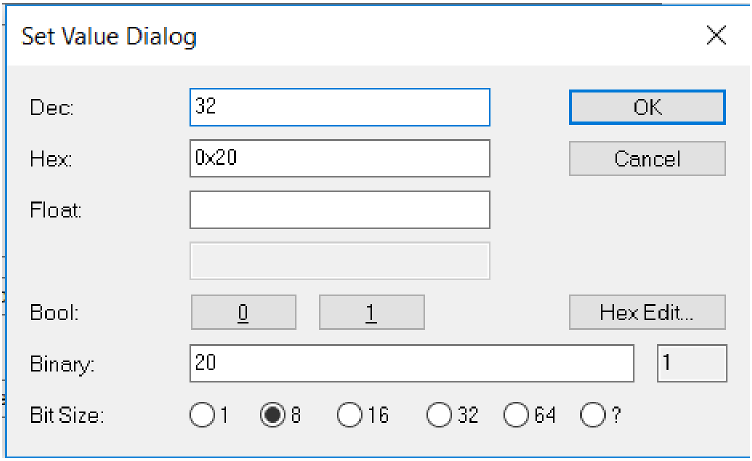
8.3.3 Activate configuration

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

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

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

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



9 Diagnostics processing

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

9.1 Channel error

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

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

Table 9: Interpretation of channel errors

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

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

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

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

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

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

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

9.2.1 Error of the actuator power supply U_L

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

- MP08DIO08DIO

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

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

The following LED behavior is visible:

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

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

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

In case of an overload or a short circuit between pin 1 and pin 3 on the ports (X1 .. X8), the following LED behavior is visible:

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

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

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

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

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

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

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

9.2.4 Internal module error

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

9.3 Emergency messages

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

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

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

Table 10: Byte content of the emergency message

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

Table 11: Content of the error register (CoE register 0x1001)

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.

10.1 MQTT

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

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

10.1.1 MQTT configuration

In **delivery state**, MQTT functions are **disabled**. The MQTT client can be configured either using the Web interface or directly via a JSON object sent in an HTTP request. For more information see chapter MQTT configuration - Quick start guide on page 85.

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
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 12: 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 70.

10.1.2 MQTT topics

MQTT mainly relates to topics. All messages are attached to a topic which adds context to the message itself. Topics may consist of a string and they are allowed to contain slashes (`/`). In topic filters, there also wildcard symbols like e.g. (#) allowed.

10.1.2.1 Base topic

For all 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 13: Base topic variables on page 70.

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

Variable	Description
mac	The MAC address of the device
name	The name of the device
order	The ordering number of the device
serial	The serial number of the device
ip0 ip1 ip2 ip3	IP address octets

Table 13: Base topic variables

Example:

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

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

```
Base-Topic/domain/.….
```

There are the following domains:

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

Table 14: Data domains

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

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

Table 15: 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 16: Use case examples

10.1.2.2 Publish topic

Overview of all publish JSON data for the defined topics:

Identity/gateway	
Key	**Data type**
product_name	json_string
ordering_number	json_string
device_type	json_string
serial_number	json_string
mac_address	json_string
production_date	json_string
fw_name	json_string
fw_date	json_string
fw_version	json_string
hw_version	json_string
family	json_string
location	json_string
country	json_string
fax	json_string
vendor_name	json_string
vendor_address	json_string
vendor_phone	json_string
vendor_email	json_string
vendor_techn_support	json_string
vendor_url	json_string
vendor_id	json_integer
device_id	json_integer

Table 17: Identity/gateway

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

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

Table 18: Config/gateway

Status/gateway				
Key	**Data type**	**Range**	**Default value**	**Remarks**
protocol	json_string	PROFINET: ▶ UNKNOWN ▶ OFFLINE ▶ STOP ▶ IDLE ▶ OPERATE EtherNet/IP: ▶ CONNECTED ▶ DISCONNECTED EtherCAT®: ▶ PREOP ▶ SAFEOP ▶ OP ▶ INIT ▶ UNKNOWN Modbus TCP: ▶ No Connections ▶ Connected CC-Link IE Feld Basic: ▶ ON ▶ STOP ▶ DISCONNECTED ▶ ERROR		
system_voltage_fault	json_boolean	true / false		
actuator_voltage_fault	json_boolean	true / false		
internal_module_error	json_boolean	true / false		
simulation_active_diag	json_boolean	true / false		
us_voltage	json_integer	0 .. 32		in Volts
ul_voltage	json_integer	0 .. 32		in Volts
forcemode_enabled	json_boolean	true / false		

Table 19: Status/gateway

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

Table 20: Process/gateway

Identity/port/1 .. 8				
Key	**Data type**	**Range**	**Default value**	**Remarks**
port	json_integer	1 .. 8		
type	json_string	Digital Input DIO Digital Output DIO Pin 4 Only DI Pin 4 Only DO Pin 4 Only Not available Unknown		
max_output_power_cha	json_string	2.0_mA 0.5_mA		
max_output_power_chb	json_string	2.0_mA 0.5_mA		
channel_cha	json_string	Digital Input Digital Output DIO Digital Input/Output Auxiliary Power Auxiliary with DO Not available Unknown		
channel_chb	json_string	Digital Input Digital Output DIO Digital Input/Output Auxiliary Power Auxiliary with DO Not available Unknown		

Table 21: Identity/port/1 .. 8

Config/port/1 .. 8				
Key	**Data type**	**Range**	**Default value**	**Remarks**
port	json_integer	1 .. 8		
direction_cha	json_string	Output Input Inactive Auxiliary Power DIO Unknown		
direction_chb	json_string	Output Input Inactive Auxiliary Power DIO Unknown		
restart_mode_cha	json_string	Manual Auto		
restart_mode_chb	json_string	Manual Auto		
input_polarity_cha	json_string	NO NC		
input_polarity_chb	json_string	NO NC		
input_filter_cha	json_integer			ms
input_filter_chb	json_integer			ms
do_auto_restart_cha	json_boolean	true / false		
do_auto_restart_chb	json_boolean	true / false		
failsafe_cha	json_string	set_low set_high hold_last	set_low	
failsafe_chb	json_string	set_low set_high hold_last	set_low	
surveillance_timeout_cha	json_integer	0 .. 255	80	

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

Table 22: Config/port/1 .. 8

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

Table 23: 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 25: Force object: Digital on page 82)		

Table 24: 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 25: Force object: Digital

[…]/config

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

Property	Data type	Example values	Remarks
portmode	array (Table 27: Config object: Portmode on page 83)		
ip_address	string	"192.168.1.5"	
subnet_mask	string	"255.255.255.0"	
gateway	string	"192.168.1.100"	

Table 26: 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		

Table 27: 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 28: 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

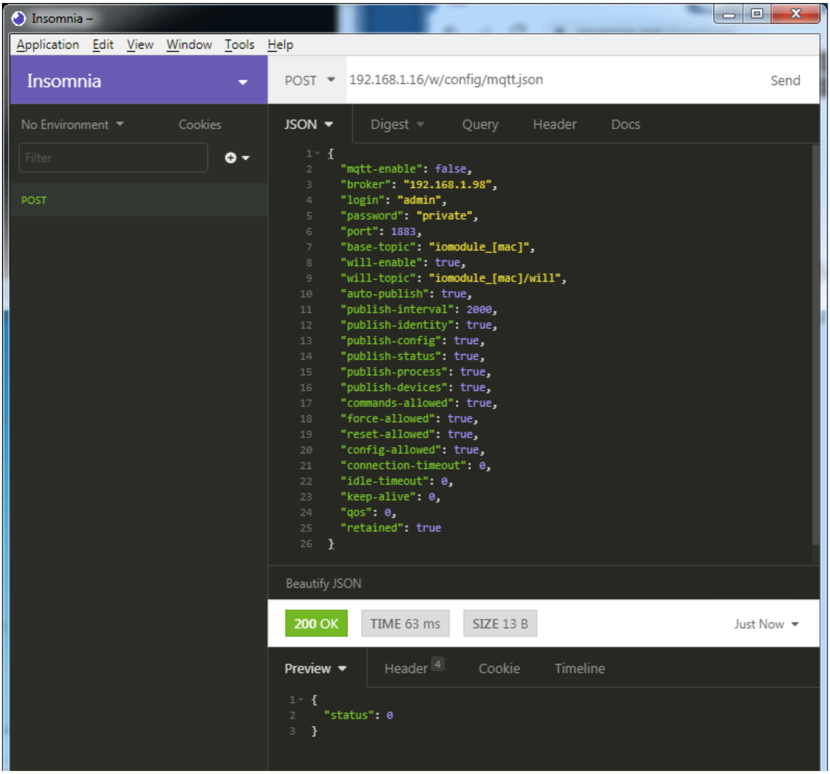
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

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/

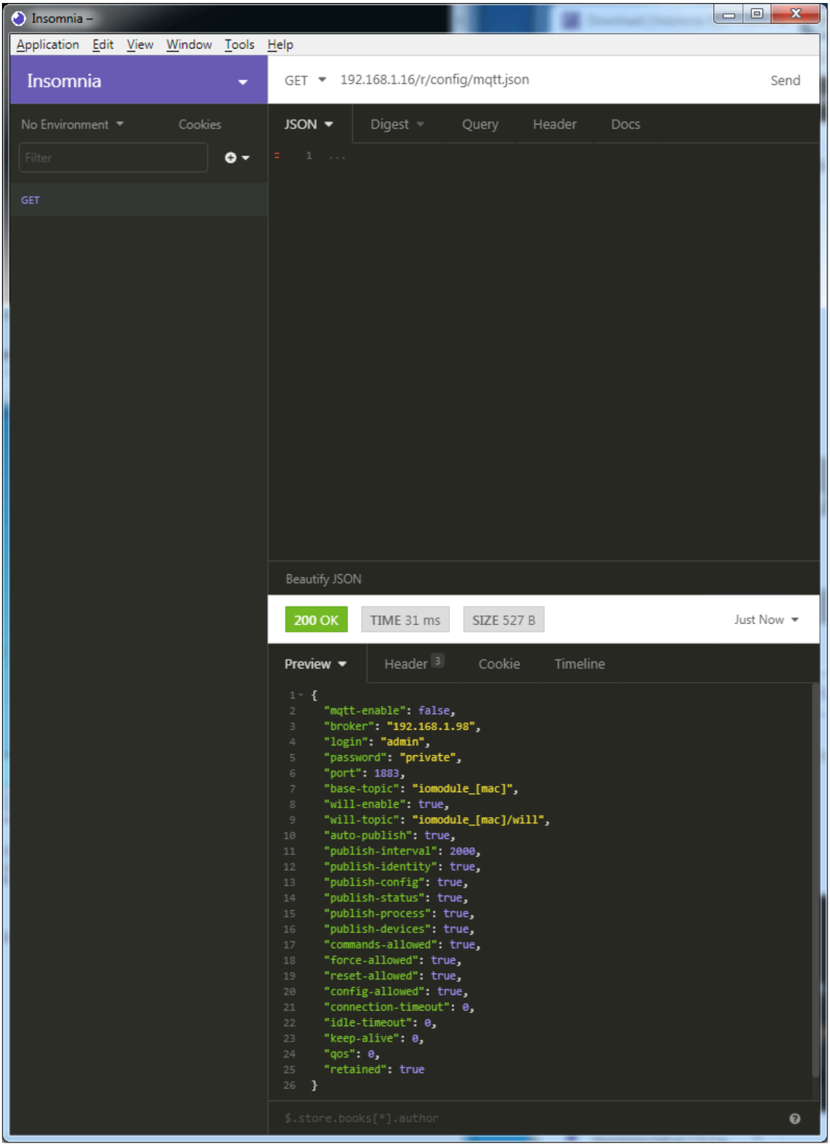
2. Configure MQTT:

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



3. Read MQTT:

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



10.2 OPC UA

OPC 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.

10.2.1 OPC UA configuration

In **delivery state**, OPC UA functions are **disabled**. The OPC UA Server can be configured either using the Web interface or directly via a JSON Object sent in an HTTP request.

The configuration URL is:

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

The configuration can also read back as a JSON file:

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

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

Tree overview of OPC UA objects:

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

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

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

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

Response:

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

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

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

Examples:

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

{"status": 0}

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

10.2.1.1 Gateway objects

Identity

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

Status (read)

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

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

Forcing (read)

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

Config (read + write)

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

Process (read)

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

Commands (write)

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

10.2.1.2 Ports objects

Identity

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

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

See details in Channel objects on page 94.

Status (read)

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

Config (read + write)

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

10.2.1.3 Channel objects

Identity (read)

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

Status (read)

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

Config (read + write)

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

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

Process (read)

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

Forcing (read + write)

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

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

10.2.2 OPC UA address space

OPC UA provides different services on the 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

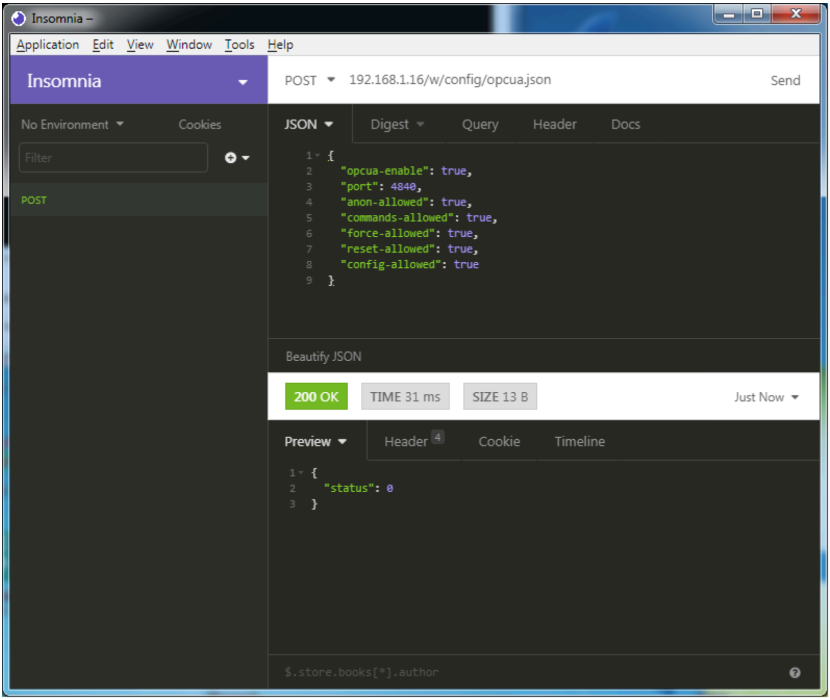
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

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/

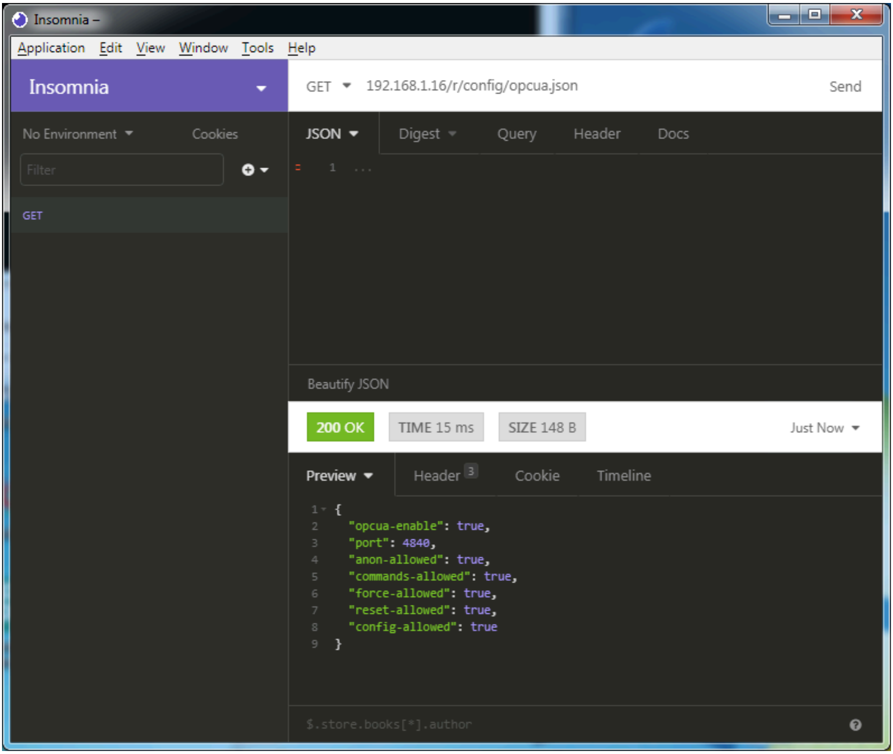
2. Configure OPC UA:

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 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.

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

10.3.1 Standard device information

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

The goal of the "Standard device information" request is to get a complete snapshot of the current device status. The format is JSON.

10.3.2 Structure

Name	Data type	Description	Example
name	string	Device name	"MP08DIO08DIO "
order-id	string	Ordering number	"381166718"
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
CHANNEL Object			
name	string	Name of channel	
type	number	Hardware channel type as number: 0 = DIO 1 = Input 2 = Output 3 = Input/Output 4 = Channel not available 5 = Channel not available 6 = Channel not available 7 = Channel not available 8 = Channel not available	
type_text	string	Textual representation of the channel type	
config	number	Current configuration of the channel: 0 = DIO 1 = Input 2 = Output 3 = Channel not available 4 = Deactivated 5 = Channel not available	
config_text	string	Textual representation of the current config	
inputState	boolean	Input data (producing data) bit to the PLC	
outputState	boolean	Output data bit to the physical output pin	
forced	boolean	True, if the output pin of this channel is forced	
simulated	boolean	True, if the input value to the PLC of this channel is simulated	
actuatorDiag	boolean	True, if the output is in short circuit / overload condition	
sensorDiag	boolean	True, if the sensor supply (Pin 1) is in short circuit / overload condition	

Name	Data type	Description	Example
maxOutputCurrent_mA	number	Maximum output current of the output in mA	
current_mA	number	Measured current of the output in mA (if current measurement is available)	
voltage_mV	number	Measured voltage of this output in mV (if voltage measurement is available)	
PORT Object			
port_type	string	Textual representation of the port type	
aux_mode	number	Indicates the configured mode for the Pin 2: 0 = No AUX 1 = AUX output (always on) 2 = Digital output (can be controlled by cyclic data) 3 = Digital input	
aux_text	string	Textual representation of the current aux mode	"AUX Output"
ds_fault	number	Data storage error number	
ds_fault_text	string	Textual data storage error.	
diag	array of DIAG (n)	Array of port related events	
DIAG Object			
error	number	Error code	
source	string	Source of the current error.	"device" "master"
message	string	Error message	"Supply Voltage fault"
FORCING Object		Forcing information of the device	
forcingActive	boolean	Force mode is currently active	
forcingPossible	boolean	True, if forcing is possible and force mode can be activated	
AuthPossible	boolean	True, if the JSON Interface can obtain forcing autorization	
ownForcing	boolean	True, if forcing is performed by REST API at the moment	
currentClient	string	Current forcing client identifier	

Name	Data type	Description	Example
digitalOutForced	array of numbers (2)	The force values of all 16 digital output channels.	
digitalOutMask	array of numbers (2)	The forcing mask of all 16 digital output channels.	
digitalInForced	array of numbers (2)	The force values of all 16 digital input channels.	
digitalInMask	array of numbers (2)	The forcing mask of all 16 digital input channels.	

10.3.3 Configuration and forcing

Method: POST

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

Parameters: None

Post-Body: JSON Object

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

Table 29: 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", "off", "aux"	
inlogica	string	"no","nc"	
inlogicb	string	"no","nc"	

Table 30: 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 31: Digital object

10.4 CoAP server

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

CoAP is based on the Server-Client principle and a service layer protocol that lets nodes and machines communicate with each other. The 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 109.

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 32: 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	

Table 33: REST API access via CoAP

10.4.3 CoAP configuration - Quick start guide

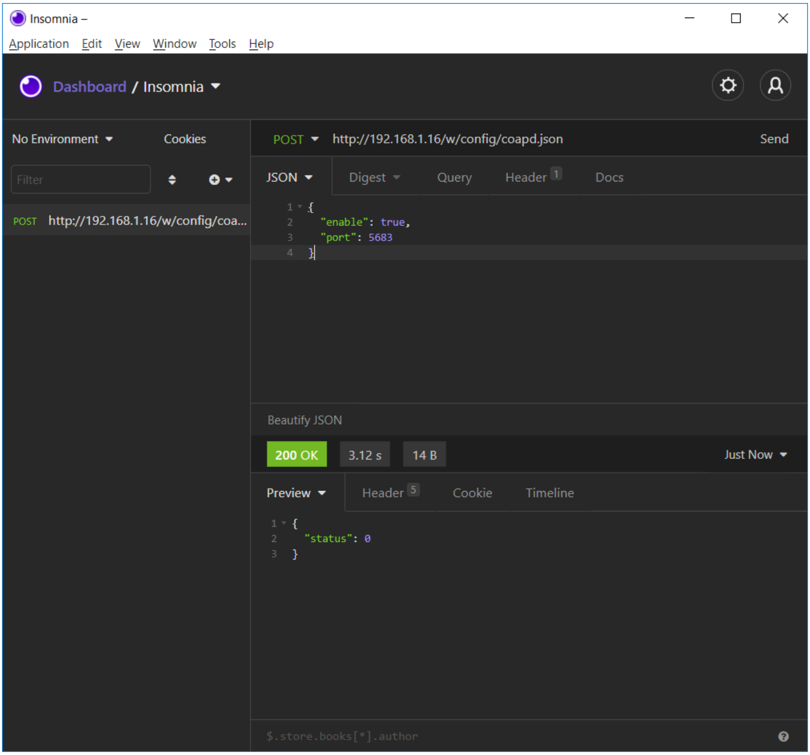
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

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/

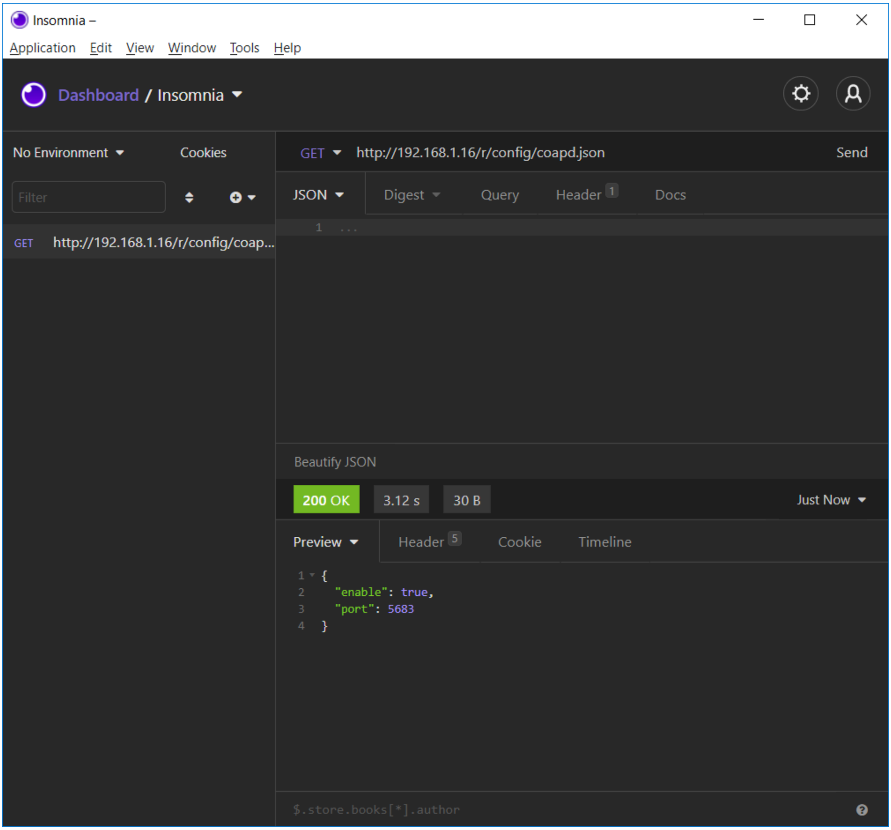
2. Configure CoAP:

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



3. Read CoAP configuration:

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



10.5 Syslog

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 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 114.

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 34: 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

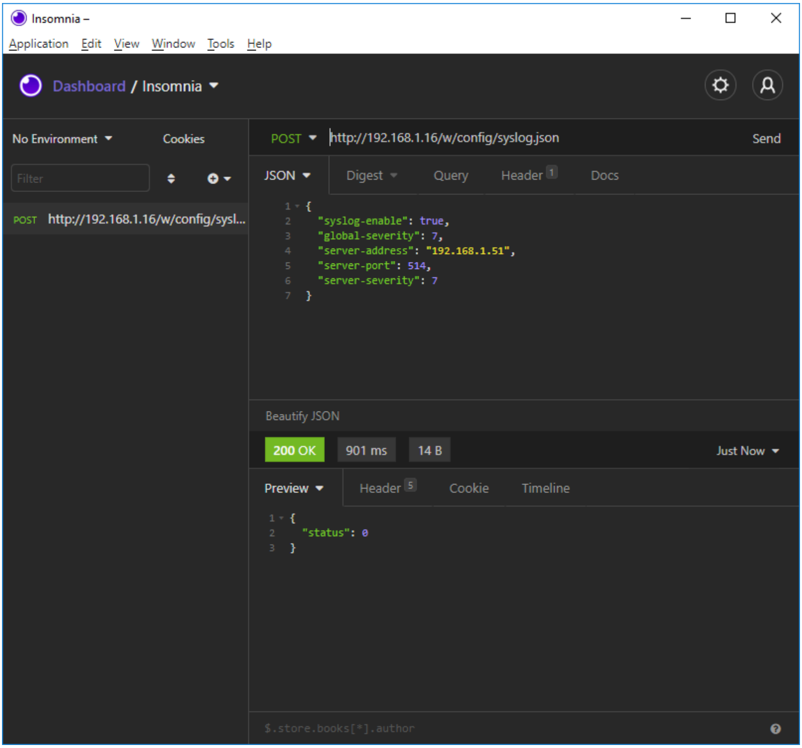
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

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/

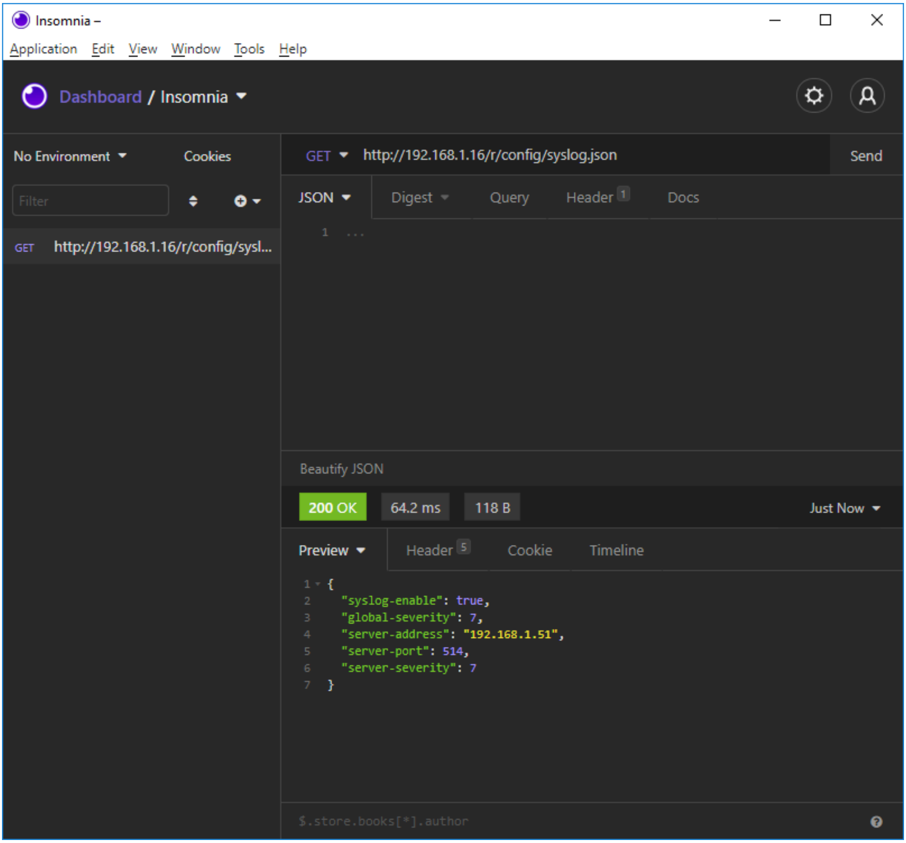
2. Configure Syslog:

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 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 118.

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 35: 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

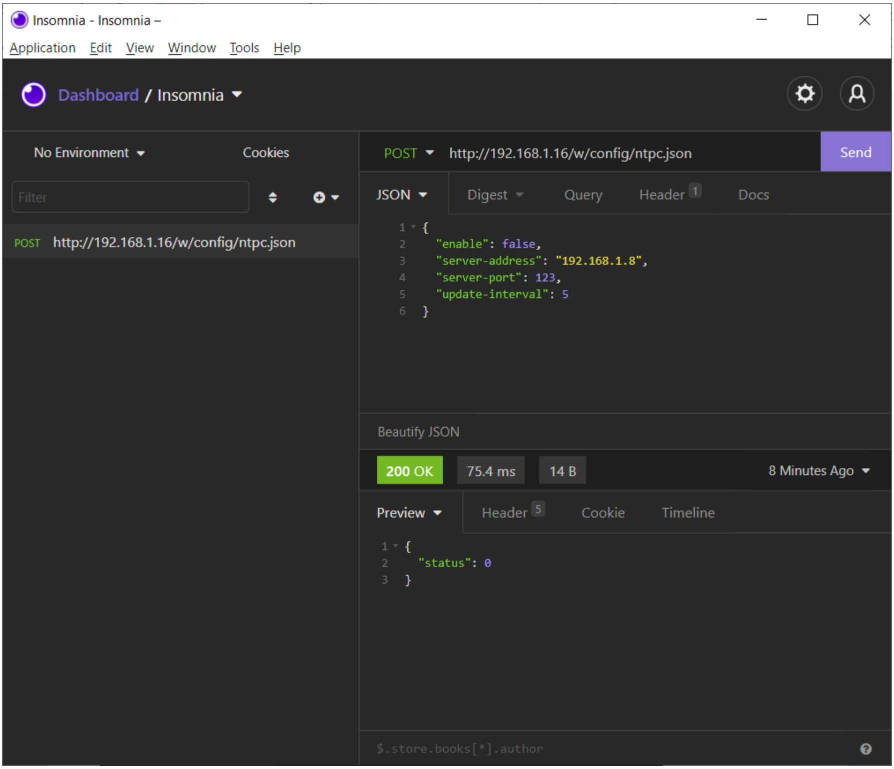
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

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/

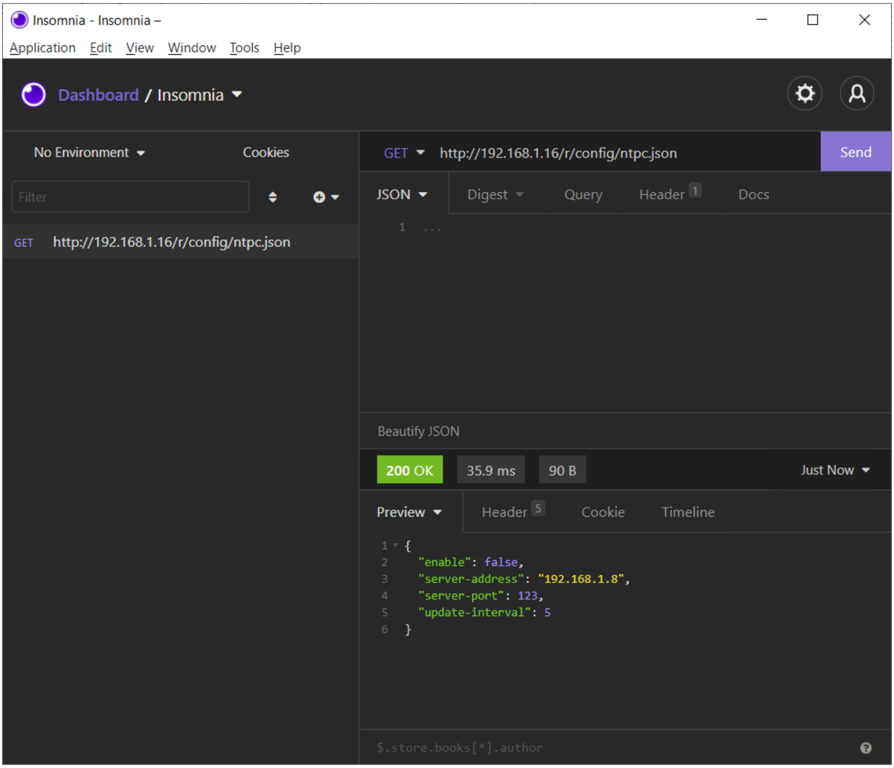
2. Configure NTP:

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



3. Read NTP configuration:

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



11 The integrated Web server

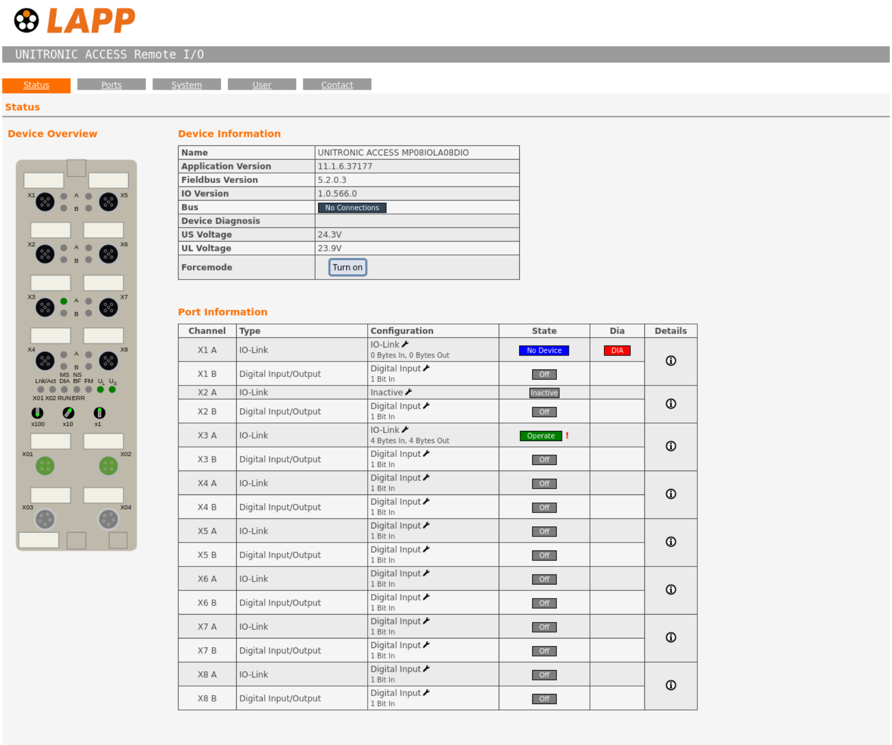
All device variants are equipped with an integrated Web server which makes functions for the device configuration and the display of status and diagnostic information available via a Web interface.

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://" 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


LAPP

UNITRONIC ACCESS Remote I/O

Status Ports System User Contact

Status

Device Overview

Device Information

Name	UNITRONIC ACCESS MP08IOLA08DIO
Application Version	11.1.6.37177
Fieldbus Version	5.2.0.3
IO Version	1.0.566.0
Bus	No Connections
Device Diagnosis	
US Voltage	24.3V
UL Voltage	23.9V
Forcemode	Turn on

Port Information

Channel	Type	Configuration	State	Dia	Details
X1 A	IO-Link	IO-Link 0 Bytes In, 0 Bytes Out	No Device	DIA	ⓘ
X1 B	Digital Input/Output	Digital Input 1 Bit In	Off		
X2 A	IO-Link	Inactive	Inactive		ⓘ
X2 B	Digital Input/Output	Digital Input 1 Bit In	Off		
X3 A	IO-Link	IO-Link 4 Bytes In, 4 Bytes Out	Operate !		ⓘ
X3 B	Digital Input/Output	Digital Input 1 Bit In	Off		
X4 A	IO-Link	Digital Input 1 Bit In	Off		ⓘ
X4 B	Digital Input/Output	Digital Input 1 Bit In	Off		
X5 A	IO-Link	Digital Input 1 Bit In	Off		ⓘ
X5 B	Digital Input/Output	Digital Input 1 Bit In	Off		
X6 A	IO-Link	Digital Input 1 Bit In	Off		ⓘ
X6 B	Digital Input/Output	Digital Input 1 Bit In	Off		
X7 A	IO-Link	Digital Input 1 Bit In	Off		ⓘ
X7 B	Digital Input/Output	Digital Input 1 Bit In	Off		
X8 A	IO-Link	Digital Input 1 Bit In	Off		ⓘ
X8 B	Digital Input/Output	Digital Input 1 Bit In	Off		

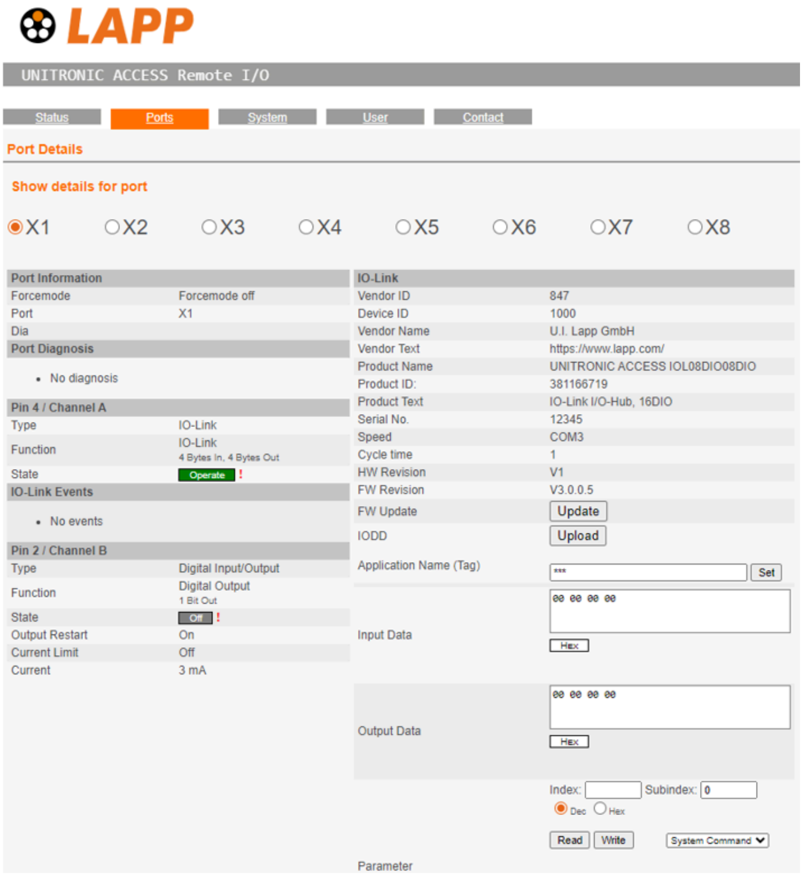
The status page provides a quick overview of the current state of the device.

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

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

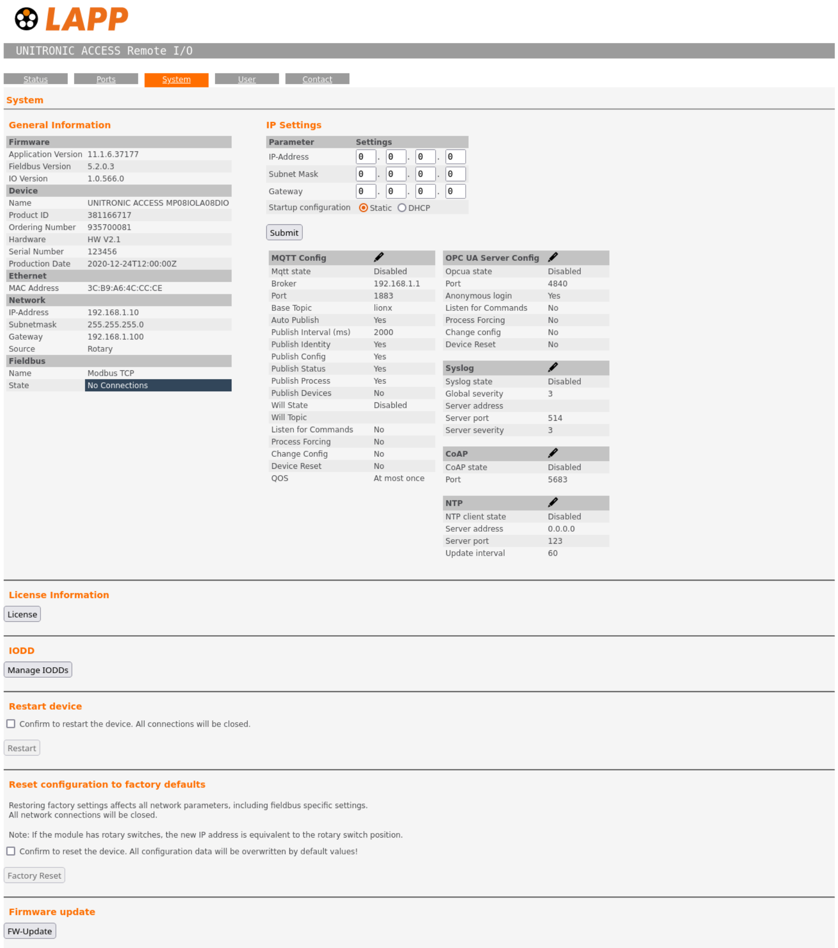
The "Port Information" table shows the configuration and state of the I/O ports.

11.1.2 The Ports page



The page shows detailed port information. In the field **Port Diagnosis**, incoming and outgoing diagnostics are displayed as clear text. **Pin 2** and **Pin 4** contain information about the configuration and state of the port.

11.1.3 The System page



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

Restart Device

The module initializes a software reset.

Reset to Factory Settings

The module restores to the default factory settings.

IP Settings

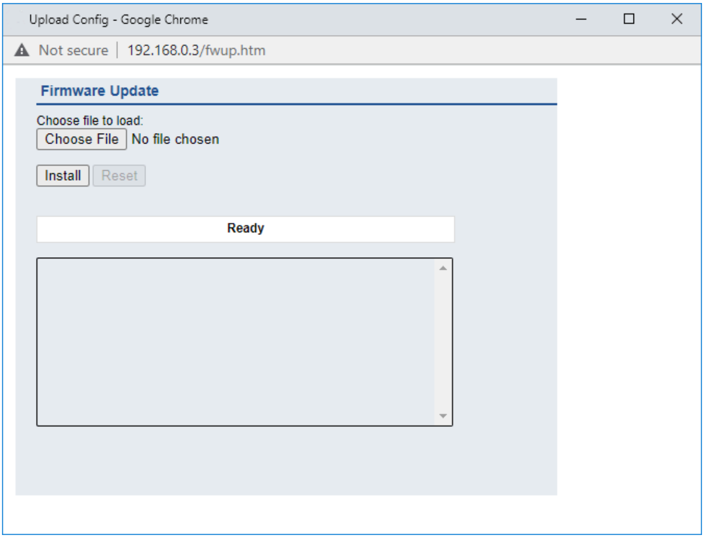
Use this parameter to change the current IP address of the module.

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

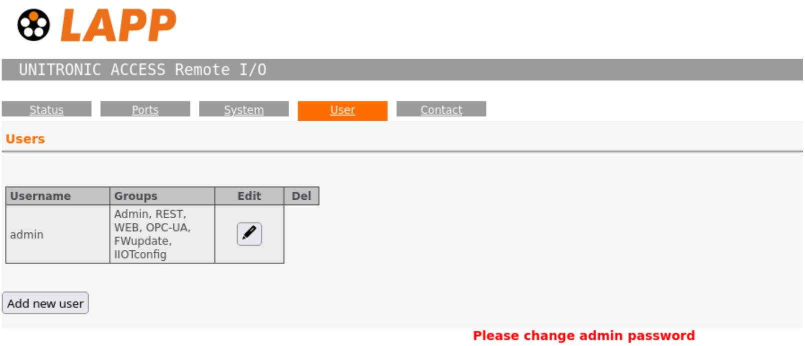
Firmware Update

The module initializes a Firmware update.

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



11.1.4 The User page



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

Default user login data:

- User: admin
- Password: private

12 Technical data

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

12.1 General

Protection class (Only applies if the connectors are screwed together or if protective caps are used.)[1]	IP65 IP67 IP69K	
Ambient temperature (during operation and storage)	MP08DIO08DIO	-40 °C .. +70 °C (-40 °F .. +158 °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 36: General information

1 Not under UL investigation.

12.2 EtherCAT® protocol

Protocol	EtherCAT® (ETG.1000 V1.2)
ESI file	U.I.Lapp-UnitronicAccess-Digital-IO.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 TCP/IP
Switch functionality	Integrated
EtherCAT® interface Port	2x M12 sockets 4-pin, D-coded (see pin assignments)

Table 37: EtherCAT® protocol

12.3 Power supply of the module electronics/ sensors

Port X03, X04	M12-L-coded Power, connector/socket, 5-pole Pin 1 / Pin 3		
Nominal voltage U_S	24 V DC (SELV/PELV)		
Current U_S	Max. 16 A		
Voltage range	21 .. 30 V DC		
Power consumption of module electronics	Typically 160 mA (+/-20 % at U_S nominal voltage)		
Power supply interruption	Max. 10 ms		
Voltage ripple U_S	Max. 5 %		
Current consumption sensor system (Pin 1)	MP08DIO08DIO	Port X1 .. X8 (Pin 1)	max. 4 A per port (at $T_{ambient}$ = 30° C)
Voltage level of the sensor power supply	Min. (U_S – 1.5 V)		
Short circuit/overload protection of sensor supply	Yes, per port		
Reverse polarity protection	Yes		
Operational indicator (U_S)	LED green:	18 V (+/- 1 V) < U_S	
	LED red:	U_S < 18 V (+/- 1 V)	

Table 38: Information on the power supply of the module electronics/ sensors

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

- Max. 4.0 A per port
- Max. 5.0 A for each port pair X1/X2, X3/X4, X5/X6, X7/X8

- Max. 9.0 A in total for the whole port group X1 .. X8

  Pay attention to the derating!

12.4 Power supply of the actuators

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

Table 39: Information on the power supply of the actuators

12.5 I/O ports

MP08DIO08DIO	Ports X1 .. X8	DI, DO	M12 socket, 5-pin

Table 40: I/O ports: Overview of functions

12.5.1 Digital inputs

Input connection	MP08DIO08DIO		Type 3 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	MP08DIO08DIO	X1 .. X8	16
Status indicator	yellow LED for Channel A (Pin 4) white LED for Channel B (Pin 2)		
Diagnostic indicator	red LED per port		

Table 41: I/O ports configured as digital input

12.5.2 Digital outputs

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

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

Pay attention to the derating!

Output type	normally open, p-switching	
Nominal output voltage per channel	Signal status “1” Signal status “0”	min. (U_L -1 V) max. 2 V
Max. output current per device	MP08DIO08DIO	9 A
Max. output current per channel	MP08DIO08DIO (X1 .. X8)	2 A
Short-circuit/overload protected	yes/yes	
Behavior in case of short circuit or overload	deactivation with automatic power-on (parameterized)	
Number of digital outputs	MP08DIO08DIO (X1 .. X8)	16
Status indicator	yellow LED per output Channel A (Pin 4) white LED per output Channel B (Pin 2)	
Diagnostic indicator	red LED per channel	

Table 42: I/O ports configured as digital output

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

12.6 LEDs

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

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

Table 43: Information on the LED colors

12.7 Data transfer times

The following tables give an overview of the internal data transfer times of UNITRONIC® ACCESS.

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

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

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

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

Use case 1:

UNITRONIC® ACCESS Digital-I/O configuration with enabled Web interface and *disabled* IIoT protocols

16DIO variant (MP08DIO08DIO):

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

Use case 2:

UNITRONIC® ACCESS Digital-I/O configuration with enabled Web interface and *enabled* IIoT protocols

16DIO variant (MP08DIO08DIO):

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

13 Accessories

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

https://www.lapp.com

Manual

Modbus TCP

UNITRONIC ACCESS Digital-I/O multi-protocol: MP08DIO08DIO (16 × Input/Output)

Technical Support
https://www.lapp.com

Contents

1 About this manual

1.1 General information

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

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

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

U.I. Lapp GmbH

Schulze-Delitzsch-Straße 25
D-70565 Stuttgart
Germany

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	09/2024	

Table 1: Overview of manual revisions

2 Safety instructions

2.1 Intended use

The products described in this manual are decentralized I/O Devices on an Industrial Ethernet Network.

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

The modules fulfill the requirements of the EMC guidelines (89/336/EEC, 93/68/EEC and 93/44/EEC) and the low voltage guideline (73/23/EEC).

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

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

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

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

Only install cables and accessories that fulfill the requirements and regulations for safety, electromagnetic compatibility, and, where applicable, telecommunication end devices, as well as the specification information. Information on which cables and accessories are permitted for the installation can be obtained from 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 this manual are authorized to install and maintain the devices described. These are persons who

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

Only 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 manual, can result in serious personal injury or damage to equipment.

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

3 Designations and synonyms

AOI	Add-On Instruction
API	Application Programming Interface
BF	Bus Fault LED
Big Endian	Data format with High-B on first place (PROFINET)
BUI	Back-Up Inconsistency (EIP diagnostics)
CC	CC-Link IE Field
Ch. A	Channel A (Pin 4) of I/O port
Ch. B	Channel B (Pin 2) of I/O port
CIP	Common Industrial Protocol (media independent protocol)
CoAP	Constrained Application Protocol
CSP+	Control & Communication System Profile Plus
DCP	Discovery and Configuration Protocol
DevCom	Device Comunicating (EIP diagnostics)
DevErr	Device Error (EIP diagnostics)
DI	Digital Input
DIA	Diagnostic LED
DO	Digital Output
DIO	Digital Input/Output
DTO	Device Temperature Overrun (EIP diagnostics)
DTU	Device Temperature Underrun (EIP diagnostics)
DUT	Device under test
EIP	EtherNet/IP
ERP	Enterprise Resource Planning system
ETH	ETHERNET
FE	Functional Earth
FME	Force Mode Enabled (EIP diagnostics)
FSU	Fast Start-Up

GSDML	General Station Description Markup Language
High-B	High-Byte
ICT	Invalid Cycle Time (EIP diagnostics)
IIoT	Industrial Internet of Things
ILE	Input process data Length Error (EIP diagnostics)
IME	Internal Module Error (EIP diagnostics)
I/O	Input / Output
I/O port	X1 .. X8
I/O port pin 2	Channel B of X1 .. X8
I/O port pin 4 (C/Q)	Channel A of X1 .. X8
IVE	IO-Link port Validation Error (EIP diagnostics)
I&M	Identification & Maintenance
JSON	JavaScript Object Notation (platform independent data format)
L+	I/O port pin 1, sensor power supply
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
OLE	Output process data Length Error (EIP diagnostics)
OPC UA	Open Platform Communications Unified Architecture (platform independent, service-oriented architecture)

PLC	Programmable Logic Controller
PN	PROFINET
PWR	Power
REST	REpresentational State Transfer
RFC	Request for Comments
RPI	Requested Packet Interval
RWr	Word data input as seen from the master station (CC-Link)
RWw	Word data output as seen from the master station (CC-Link)
RX	Bit data input as seen from the master station (CC-Link)
RY	Bit data output as seen from the master station (CC-Link)
SCA	Short Circuit Actuator/U_L/U_{AUX} (EIP diagnostics)
SCS	Short Circuit Sensor (EIP diagnostics)
SLMP	Seamless Message Protocol
SNMP	Simple Network Management Protocol
SP	Single Protocol (PROFINET, EtherNet/IP, EtherCAT®, Modbus TCP or CC-Link IE Field Basic)
SPE	Startup Parameterization Error (EIP diagnostics)
U_{AUX}	$U_{Auxiliary}$, supply voltage for the load circuit (Actuator supply on Class B ports)
UDP	User Datagram Protocol
UDT	User-Defined Data Types
UINT8	Byte in PLC (IB, QB)
UINT16	Unsigned integer with 16 bits or word in PLC (IW, QW)
U_L	U_{Load}, supply voltage for the load circuit (Actuator supply on Class A)
UL	Underwriters Laboratories Inc. (certification company)
UTC	Coordinated Universal Time (Temps Universel Coordonné)

Table 2: Designations and synonyms

4 System description

4.1 Device variants

The following Digital I/O device variants are available in the UNITRONIC® ACCESS family:

Article number	Product designation	Description	I/O port functionality
381166718	MP08DIO08DIO	UNITRONIC® ACCESS M12-60 mm, I/O Device Multi-protocol (PN, EIP, EC, MB, CC) Security	16 x Input/Output universal

Table 3: Overview of UNITRONIC® ACCESS Digital-I/O variants

4.2 I/O port overview

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

UNITRONIC® ACCESS 16DIO ports

Device variant	Port	Pin 1 U_S	Pin 4 / Ch. A (In/Out)		Pin 2 / Ch. B (In/Out)	
MP08DIO08DIO	**Info:**	–	Type 3	Supply by U_L	Type 3	Supply by U_L
	X8:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)
	X7:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)
	X6:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)
	X5:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)
	X4:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)
	X3:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)
	X2:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)
	X1:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)

Table 4: Port configuration of MP08DIO08DIO variant

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 Integrated Web server

Network parameter display

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

Displaying diagnostics

View diagnostics via the integrated Web server.

User management

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

5.3 Security features

Firmware signature

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

Syslog

The 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.

5.4 Other features

Interface protection

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

For more details, see section Port assignments on page 22.

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 Digital-I/O multi-protocol variants

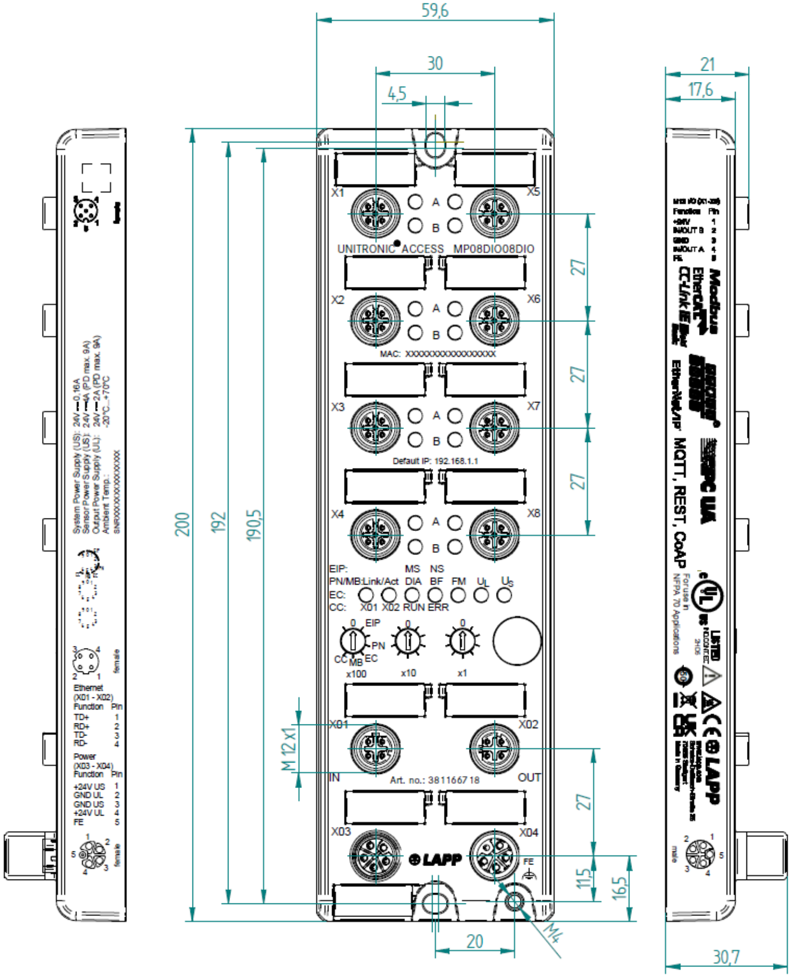


Figure 1: MP08DIO08DIO

6.2.2 Notifications

Attention:

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

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

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

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

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

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

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

6.3 Port assignments

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

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

Color coding: green

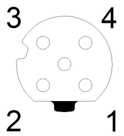


Figure 2: Schematic drawing, ports X01, X02

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

Table 5: Assignment of ports X01, X02

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

6.3.2 Power supply with M12 power L-coded

Color coding: gray

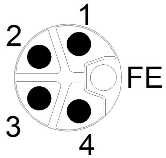


Figure 3: Schematic diagram of the M12 L-coding (connector X03 for Power In)

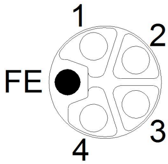


Figure 4: Schematic diagram of the M12 L-coding (socket X04 for Power Out)

Power supply	Pin	Signal	Function
	1	U_S (+24 V)	Sensor/system power supply
	2	GND_U_L	Ground/reference potential U_L
	3	GND_U_S	Ground/reference potential U_S
	4	U_L (+24 V)	Load supply Actuator supply
	5	FE	Functional ground

Table 6: Pin assignments ports X03 and X04

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

6.3.3 I/O ports as M12 sockets

Color coding: black

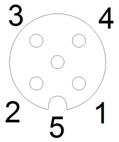


Figure 5: Schematic drawing I/O port as M12 socket

6.3.3.1 I/O ports

MP08DIO08DIO	Pin	Signal	Function
16DIO X1 .. X8	1	+24 V	power supply +24 V
	2	IN/OUT	Ch. B: Digital input or digital output
	3	GND	Ground/reference potential
	4	IN/OUT	Ch. A: Digital input or digital output
	5	FE	Functional ground

Table 7: Pin assignments I/O ports

7 Starting operation

7.1 Device information

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 accessible device information via the register addresses 1024-1053.

Register address[1]	Name	Access	Register length	Size (bytes)	Description
1024	Firmware Version minor	RO (Read Only)	1	2	MBTCP Digital device Firmware Version minor
1025	Firmware Version major	RO	1	2	MBTCP Digital device Firmware Version major
1026-1041	Model Number/ Device name	RO	16	32	Name of device
1042	IP address Source	RO	1	2	0: DHCP 1: Static
1043-1046	IP address	RO	4	8	IP Address of Device
1047-1052	MAC address	RO	6	12	MAC Address of device
1053	Current TCP connections	RO	1	2	Active TCP connections

1 Given numbers are addresses starting from "0".

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:	MP08DIO08DIO
Product type:	Modbus TCP server

7.5 Setting the rotary encoding switches

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

- MP08DIO08DIO

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.

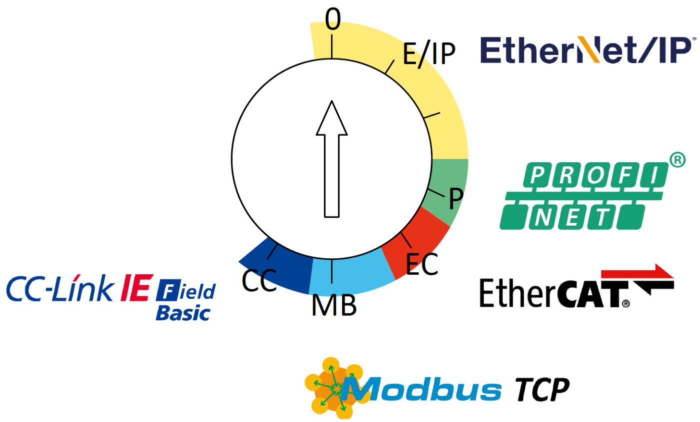
After selecting a protocol followed by a restart of the device, the new protocol is initialized. This can take up to 15 seconds. During this time the device is not usable and the LED indicators are out of function. When the protocol change is complete, the LED indicators return to normal operation and 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 Digital I/O devices 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 I/O Device in different environments.

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

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



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

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

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

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

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

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

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

To change the protocol, carry out a factory reset. In this way you restore the factory settings of the respective device. How you perform the factory reset for your device is described in chapter Factory reset on page 31.

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

7.5.1 Modbus TCP

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

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

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

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

7.5.2 Factory reset

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

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

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

	x100	x10	x1
Factory Reset	9	7	9

Follow the steps from section Setting the rotary encoding switches on page 28 again to select a new protocol.

For performing a factory reset via software configuration, see chapter OPC UA configuration on page 71 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 on page 28.

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.

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

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

8.1 Outputs / Coils

Register address*	Coil No.*	Name	Access	Length (Boolean)	Size (bits)
0	1	Output X1.A	RW (Read/ Write)	1	1
1	2	Output X1.B	RW	1	1
2	3	Output X2.A	RW	1	1
3	4	Output X2.B	RW	1	1
4	5	Output X3.A	RW	1	1
5	6	Output X3.B	RW	1	1
6	7	Output X4.A	RW	1	1
7	8	Output X4.B	RW	1	1
8	9	Output X5.A	RW	1	1
9	10	Output X5.B	RW	1	1
10	11	Output X6.A	RW	1	1
11	12	Output X6.B	RW	1	1
12	13	Output X7.A	RW	1	1
13	14	Output X7.B	RW	1	1
14	15	Output X8.A	RW	1	1
15	16	Output X8.B	RW	1	1

* Depending on *Coils* or *Register* selection in Flashlabel. Default: *Coils*.

8.2 Inputs

Register address*	Input No.*	Name	Access	Length (Boolean)	Size (bits)
16	1	Input X1.A	RO (Read Only)	1	1
17	2	Input X1.B	RO	1	1
18	3	Input X2.A	RO	1	1
19	4	Input X2.B	RO	1	1
20	5	Input X3.A	RO	1	1
21	6	Input X3.B	RO	1	1
22	7	Input X4.A	RO	1	1
23	8	Input X4.B	RO	1	1
24	9	Input X5.A	RO	1	1
25	10	Input X5.B	RO	1	1
26	11	Input X6.A	RO	1	1
27	12	Input X6.B	RO	1	1
28	13	Input X7.A	RO	1	1
29	14	Input X7.B	RO	1	1
30	15	Input X8.A	RO	1	1
31	16	Input X8.B	RO	1	1

* Depending on *Inputs* or *Register* selection in Flashlabel. Default: *Inputs*.

8.3 Registers

Given numbers are addresses starting from "0".

8.3.1 Latch register

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

8.3.2 Global configuration

Register address	Name	Access	Register length	Size (bytes)	Description
+ 128 (640)					
640	Report U_L Voltage Fault	RW (Read/ Write)	1	2	A missing or out of range U_L supply voltage is ignored. Valid Values: "0" = Reporting disabled "1" = Reporting enabled "2" = Auto (Reporting active after U_L was present) Default Value: "1"
641	Report Actuator Fault without U_L	RW	1	2	An actuator diagnosis which is caused by a missing U_L voltage is suppressed. Valid Values: "0" = Reporting disabled "1" = Reporting enabled Default Value: "1"
642	Report U_S Voltage Fault	RW	1	2	An out of range U_S supply voltage is ignored. Valid Values: "0" = Reporting disabled "1" = Reporting enabled Default Value: "1"

Register address	Name	Access	Register length	Size (bytes)	Description
644	Web Interface Lock	RW	1	2	If enabled, the Web interface is not accessible. Valid Values: "0" = No lock "1" = Web interface locked Default Value: "0"
645	Forcing Lock	RW	1	2	If locked, outputs can not be forced from Web interface or any IIOT protocol. Additionally, there is no input simulation possible in this case. Valid Values: "0" = No Lock "1" = Forcing locked Default Value: "0"
646	Reserved	RW	1	2	
647	Reserved	RW	1	2	
648	External Configuration Lock	RW	1	2	If locked, no port / channel reconfiguration via Web interface or IIOT protocol is possible. Valid Values: "0" = No External Lock "1" = External config is locked Default Value: "0"

Register address	Name	Access	Register length	Size (bytes)	Description
649	Output Source	RW	1	2	Specifies the control source for the digital outputs. If "Coils" is selected, the outputs can be controlled via Coils 1-16. If "Holding Register" is selected, the outputs are controlled via Holding Register 00. The 16 Bit value in this single register directly maps to the max. 16 output channels. "0" = Coils "1" = Holding Register

8.3.3 Surveillance Timeout

Register address	Name	Access	Register length	Size (bytes)	Description
+0 (512)			16	32	
	Surv Timeout Port X1.A	RW (Read/ Write)	1	2	Surveillance timeout in ms 0 .. 255
	Surv Timeout Port X1.B	RW	1	2	
	Surv Timeout Port X2.A	RW	1	2	
	Surv Timeout Port X2.B	RW	1	2	
	Surv Timeout Port X3.A	RW	1	2	
	Surv Timeout Port X3.B	RW	1	2	
	Surv Timeout Port X4.A	RW	1	2	
	Surv Timeout Port X4.B	RW	1	2	
	Surv Timeout Port X5.A	RW	1	2	
	Surv Timeout Port X5.B	RW	1	2	
	Surv Timeout Port X6.A	RW	1	2	
	Surv Timeout Port X6.B	RW	1	2	
	Surv Timeout Port X7.A	RW	1	2	
	Surv Timeout Port X7.B	RW	1	2	
	Surv Timeout Port X8.A	RW	1	2	
	Surv Timeout Port X8.B	RW	1	2	

8.3.4 Failsafe

Register address	Name	Access	Register length	Size (bytes)	Description
+16 (528)			16	32	
	Failsafe mode Port X1.A	RW (Read/ Write)	1	2	Valid Values : "0" = Set Low "1" = Set High "2" = Hold Last Default Value : "0"
	Failsafe mode Port X1.B	RW	1	2	
	Failsafe mode Port X2.A	RW	1	2	
	Failsafe mode Port X2.B	RW	1	2	
	Failsafe mode Port X3.A	RW	1	2	
	Failsafe mode Port X3.B	RW	1	2	
	Failsafe mode Port X4.A	RW	1	2	
	Failsafe mode Port X4.B	RW	1	2	
	Failsafe mode Port X5.A	RW	1	2	
	Failsafe mode Port X5.B	RW	1	2	
	Failsafe mode Port X6.A	RW	1	2	
	Failsafe mode Port X6.B	RW	1	2	
	Failsafe mode Port X7.A	RW	1	2	
	Failsafe mode Port X7.B	RW	1	2	
	Failsafe mode Port X8.A	RW	1	2	
	Failsafe mode Port X8.B	RW	1	2	

8.3.5 Digital Input Filter

Register address	Name	Access	Register length	Size (bytes)	Description
+32 (544)			16	32	
	DI Filter Port X1.A	RW (Read/ Write)	1	2	Input Filter in 100µS (10 = 1 ms). Max. value "255" = 25.5 ms.
	DI Filter Port X1.B	RW	1	2	
	DI Filter Port X2.A	RW	1	2	
	DI Filter Port X2.B	RW	1	2	
	DI Filter Port X3.A	RW	1	2	
	DI Filter Port X3.B	RW	1	2	
	DI Filter Port X4.A	RW	1	2	
	DI Filter Port X4.B	RW	1	2	
	DI Filter Port X5.A	RW	1	2	
	DI Filter Port X5.B	RW	1	2	
	DI Filter Port X6.A	RW	1	2	
	DI Filter Port X6.B	RW	1	2	
	DI Filter Port X7.A	RW	1	2	
	DI Filter Port X7.B	RW	1	2	
	DI Filter Port X8.A	RW	1	2	
	DI Filter Port X8.B	RW	1	2	

8.3.6 Digital Input Logic

Register address	Name	Access	Register length	Size (bytes)	Description
+48 (560)			16	32	
	Digital input logic Port X1.A	RW (Read/ Write)	1	2	Valid Values : "0": Normally Open "1": Normally Close Default Value : "0"
	Digital input logic Port X1.B	RW	1	2	
	Digital input logic Port X2.A	RW	1	2	
	Digital input logic Port X2.B	RW	1	2	
	Digital input logic Port X3.A	RW	1	2	
	Digital input logic Port X3.B	RW	1	2	
	Digital input logic Port X4.A	RW	1	2	
	Digital input logic Port X4.B	RW	1	2	
	Digital input logic Port X5.A	RW	1	2	
	Digital input logic Port X5.B	RW	1	2	
	Digital input logic Port X6.A	RW	1	2	
	Digital input logic Port X6.B	RW	1	2	
	Digital input logic Port X7.A	RW	1	2	
	Digital input logic Port X7.B	RW	1	2	
	Digital input logic Port X8.A	RW	1	2	
	Digital input logic Port X8.B	RW	1	2	

8.3.7 Digital Output Auto Restart Mode

Register address	Name	Access	Register length	Size (bytes)	Description
+64 (576)			16	32	
	DO Restart Port X1.A	RW (Read/ Write)	1	2	Valid Values : "0": Disable "1": Enable Default Value : "1"
	DO Restart Port X1.B	RW	1	2	
	DO Restart Port X2.A	RW	1	2	
	DO Restart Port X2.B	RW	1	2	
	DO Restart Port X3.A	RW	1	2	
	DO Restart Port X3.B	RW	1	2	
	DO Restart Port X4.A	RW	1	2	
	DO Restart Port X4.B	RW	1	2	
	DO Restart Port X5.A	RW	1	2	
	DO Restart Port X5.B	RW	1	2	
	DO Restart Port X6.A	RW	1	2	
	DO Restart Port X6.B	RW	1	2	
	DO Restart Port X7.A	RW	1	2	
	DO Restart Port X7.B	RW	1	2	
	DO Restart Port X8.A	RW	1	2	
	DO Restart Port X8.B	RW	1	2	

8.3.8 Channel Direction

Register address	Name	Access	Register length	Size (bytes)	Description
+96 (608)			16	32	
	Channel_Direction Port X1.A	RW (Read/ Write)	1	2	**Note:** The possible value options depend on the used device variant. Valid Values : "0": Input/Output "1": Output "2": Input "3": Inactive Default Value : "0"
	Channel_Direction Port X1.B	RW	1	2	
	Channel_Direction Port X2.A	RW	1	2	
	Channel_Direction Port X2.B	RW	1	2	
	Channel_Direction Port X3.A	RW	1	2	
	Channel_Direction Port X3.B	RW	1	2	
	Channel_Direction Port X4.A	RW	1	2	
	Channel_Direction Port X4.B	RW	1	2	
	Channel_Direction Port X5.A	RW	1	2	
	Channel_Direction Port X5.B	RW	1	2	
	Channel_Direction Port X6.A	RW	1	2	
	Channel_Direction Port X6.B	RW	1	2	
	Channel_Direction Port X7.A	RW	1	2	
	Channel_Direction Port X7.B	RW	1	2	
	Channel_Direction Port X8.A	RW	1	2	
	Channel_Direction Port X8.B	RW	1	2	

8.3.9 Digital Output Current Limit

Register address	Name	Access	Register length	Size (bytes)	Description
+112 (624)			16	32	
	DO Current Limit Port X1.A	RW (Read/ Write)	1	2	Current limit in mA. "0" = Current limit off Max. value = 4000 mA
	DO Current Limit Port X1.B	RW	1	2	
	DO Current Limit Port X2.A	RW	1	2	
	DO Current Limit Port X2.B	RW	1	2	
	DO Current Limit Port X3.A	RW	1	2	
	DO Current Limit Port X3.B	RW	1	2	
	DO Current Limit Port X4.A	RW	1	2	
	DO Current Limit Port X4.B	RW	1	2	
	DO Current Limit Port X5.A	RW	1	2	
	DO Current Limit Port X5.B	RW	1	2	
	DO Current Limit Port X6.A	RW	1	2	
	DO Current Limit Port X6.B	RW	1	2	
	DO Current Limit Port X7.A	RW	1	2	
	DO Current Limit Port X7.B	RW	1	2	
	DO Current Limit Port X8.A	RW	1	2	
	DO Current Limit Port X8.B	RW	1	2	

9 Diagnostics

Diagnostics base register: 400

Register address	Name	Access	Length	Size (bits)	Description
400	Sensor Diagnosis	RO (Read Only)		8	
	b0: Port X1	RO	1	1	
	b1: Port X2	RO	1	1	
	b2: Port X3	RO	1	1	
	b3: Port X4	RO	1	1	
	b4: Port X5	RO	1	1	
	b5: Port X6	RO	1	1	
	b6: Port X7	RO	1	1	
	b7: Port X8	RO	1	1	

Register address	Name	Access	Length	Size (bits)	Description
401	Actuator Diagnosis	RO (Read Only)		16	
	b0: Port X1.A	RO	1	1	
	b1: Port X1.B	RO	1	1	
	b2: Port X2.A	RO	1	1	
	b3: Port X2.B	RO	1	1	
	b4: Port X3.A	RO	1	1	
	b5: Port X3.B	RO	1	1	
	b6: Port X4.A	RO	1	1	
	b7: Port X4.B	RO	1	1	
	b8: Port X5.A	RO	1	1	
	b9: Port X5.B	RO	1	1	
	b10: Port X6.A	RO	1	1	
	b11: Port X6.B	RO	1	1	
	b12: Port X7.A	RO	1	1	
	b13: Port X7.B	RO	1	1	
	b14: Port X8.A	RO	1	1	
	b15: Port X8.B	RO	1	1	

Register address	Name	Access	Length	Size (byte)	Description
402	US Supply Present	RO (Read Only)	1	2	System Supply Voltage valid range
403	US Supply Diag	RO	1	2	System Supply Voltage diagnosis
404	UL Supply Present	RO	1	2	Actuator Supply valid range
405	UL Supply Diag	RO	1	2	Actuator Supply diagnosis
406	Internal Module Error	RO	1	2	I/O data not reliable due to internal error
407	Forcemode active	RO	1	2	Forcemode enabled. Outputs can differ from Modbus output values.
408	US Voltage	RO	16	2	System Supply Voltage in mV
409	UL Voltage	RO	16	2	Actuator Supply Voltage in mV
410 .. 425	Output Currents	RO	16	2	Output Currents in mA for X1 .. X8 Pin 4 + Pin 2
426 .. 433	Sensor Currents	RO	16	2	Sensor Currents in mA for X1 .. X8 Pin 1

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.

10.1 MQTT

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

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

10.1.1 MQTT configuration

In **delivery state**, MQTT functions are **disabled**. The MQTT client can be configured either using the Web interface or directly via a JSON object sent in an HTTP request. For more information see chapter MQTT configuration - Quick start guide on page 69.

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
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 10: 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 54.

10.1.2 MQTT topics

MQTT mainly relates to topics. All messages are attached to a topic which adds context to the message itself. Topics may consist of a string and they are allowed to contain slashes (`/`). In topic filters, there also wildcard symbols like e.g. (#) allowed.

10.1.2.1 Base topic

For all 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 11: Base topic variables on page 54.

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

Variable	Description
mac	The MAC address of the device
name	The name of the device
order	The ordering number of the device
serial	The serial number of the device
ip0 ip1 ip2 ip3	IP address octets

Table 11: Base topic variables

Example:

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

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

```
Base-Topic/domain/.….
```

There are the following domains:

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

Table 12: Data domains

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

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

Table 13: 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 14: Use case examples

10.1.2.2 Publish topic

Overview of all publish JSON data for the defined topics:

Identity/gateway	
Key	**Data type**
product_name	json_string
ordering_number	json_string
device_type	json_string
serial_number	json_string
mac_address	json_string
production_date	json_string
fw_name	json_string
fw_date	json_string
fw_version	json_string
hw_version	json_string
family	json_string
location	json_string
country	json_string
fax	json_string
vendor_name	json_string
vendor_address	json_string
vendor_phone	json_string
vendor_email	json_string
vendor_techn_support	json_string
vendor_url	json_string
vendor_id	json_integer
device_id	json_integer

Table 15: Identity/gateway

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

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

Table 16: Config/gateway

Status/gateway				
Key	**Data type**	**Range**	**Default value**	**Remarks**
protocol	json_string	PROFINET: ▶ UNKNOWN ▶ OFFLINE ▶ STOP ▶ IDLE ▶ OPERATE EtherNet/IP: ▶ CONNECTED ▶ DISCONNECTED EtherCAT®: ▶ PREOP ▶ SAFEOP ▶ OP ▶ INIT ▶ UNKNOWN Modbus TCP: ▶ No Connections ▶ Connected CC-Link IE Feld Basic: ▶ ON ▶ STOP ▶ DISCONNECTED ▶ ERROR		
system_voltage_fault	json_boolean	true / false		
actuator_voltage_fault	json_boolean	true / false		
internal_module_error	json_boolean	true / false		
simulation_active_diag	json_boolean	true / false		
us_voltage	json_integer	0 .. 32		in Volts
ul_voltage	json_integer	0 .. 32		in Volts
forcemode_enabled	json_boolean	true / false		

Table 17: Status/gateway

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

Table 18: Process/gateway

Identity/port/1 .. 8				
Key	**Data type**	**Range**	**Default value**	**Remarks**
port	json_integer	1 .. 8		
type	json_string	Digital Input DIO Digital Output DIO Pin 4 Only DI Pin 4 Only DO Pin 4 Only Not available Unknown		
max_output_power_cha	json_string	2.0_mA 0.5_mA		
max_output_power_chb	json_string	2.0_mA 0.5_mA		
channel_cha	json_string	Digital Input Digital Output DIO Digital Input/Output Auxiliary Power Auxiliary with DO Not available Unknown		
channel_chb	json_string	Digital Input Digital Output DIO Digital Input/Output Auxiliary Power Auxiliary with DO Not available Unknown		

Table 19: Identity/port/1 .. 8

Config/port/1 .. 8				
Key	**Data type**	**Range**	**Default value**	**Remarks**
port	json_integer	1 .. 8		
direction_cha	json_string	Output Input Inactive Auxiliary Power DIO Unknown		
direction_chb	json_string	Output Input Inactive Auxiliary Power DIO Unknown		
restart_mode_cha	json_string	Manual Auto		
restart_mode_chb	json_string	Manual Auto		
input_polarity_cha	json_string	NO NC		
input_polarity_chb	json_string	NO NC		
input_filter_cha	json_integer			ms
input_filter_chb	json_integer			ms
do_auto_restart_cha	json_boolean	true / false		
do_auto_restart_chb	json_boolean	true / false		
failsafe_cha	json_string	set_low set_high hold_last	set_low	
failsafe_chb	json_string	set_low set_high hold_last	set_low	
surveillance_timeout_cha	json_integer	0 .. 255	80	

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

Table 20: Config/port/1 .. 8

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

Table 21: 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 23: Force object: Digital on page 66)		

Table 22: 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 23: Force object: Digital

[…]/config

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

Property	Data type	Example values	Remarks
portmode	array (Table 25: Config object: Portmode on page 67)		
ip_address	string	"192.168.1.5"	
subnet_mask	string	"255.255.255.0"	
gateway	string	"192.168.1.100"	

Table 24: 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		

Table 25: 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 26: 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

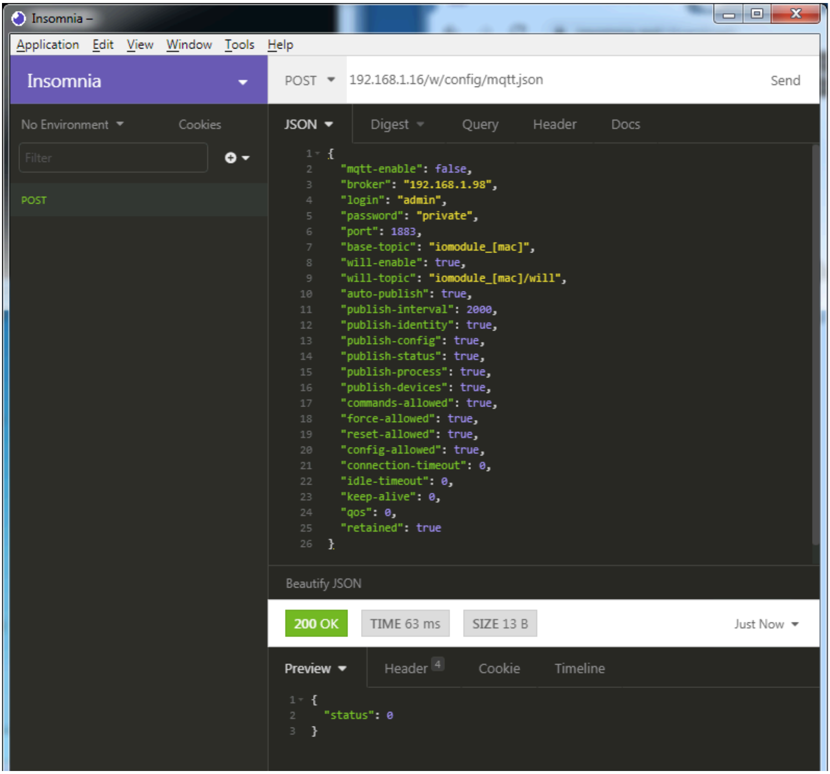
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

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/

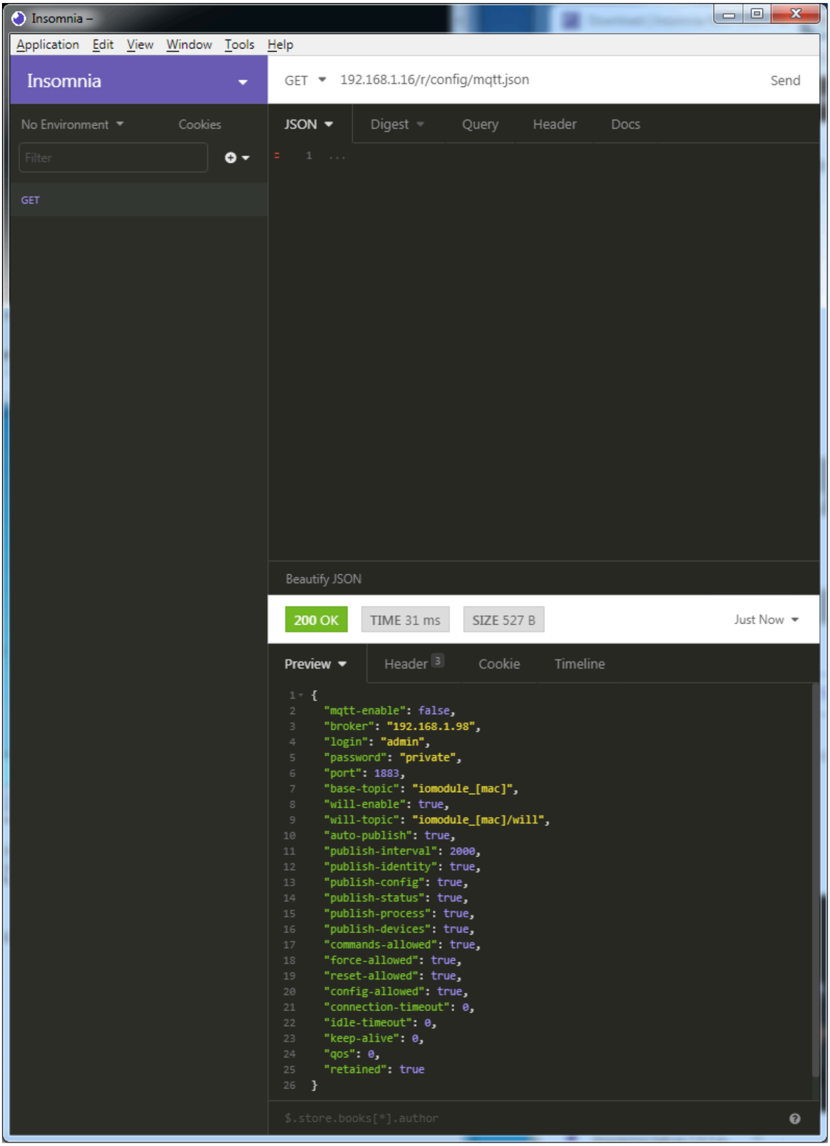
2. Configure MQTT:

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



3. Read MQTT:

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



10.2 OPC UA

OPC 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.

10.2.1 OPC UA configuration

In **delivery state**, OPC UA functions are **disabled**. The OPC UA Server can be configured either using the Web interface or directly via a JSON Object sent in an HTTP request.

The configuration URL is:

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

The configuration can also read back as a JSON file:

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

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

Tree overview of OPC UA objects:

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

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

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

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

Response:

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

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

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

Examples:

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

{"status": 0}

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

10.2.1.1 Gateway objects

Identity

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

Status (read)

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

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

Forcing (read)

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

Config (read + write)

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

Process (read)

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

Commands (write)

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

10.2.1.2 Ports objects

Identity

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

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

See details in Channel objects on page 78.

Status (read)

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

Config (read + write)

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

10.2.1.3 Channel objects

Identity (read)

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

Status (read)

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

Config (read + write)

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

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

Process (read)

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

Forcing (read + write)

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

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

10.2.2 OPC UA address space

OPC UA provides different services on the 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

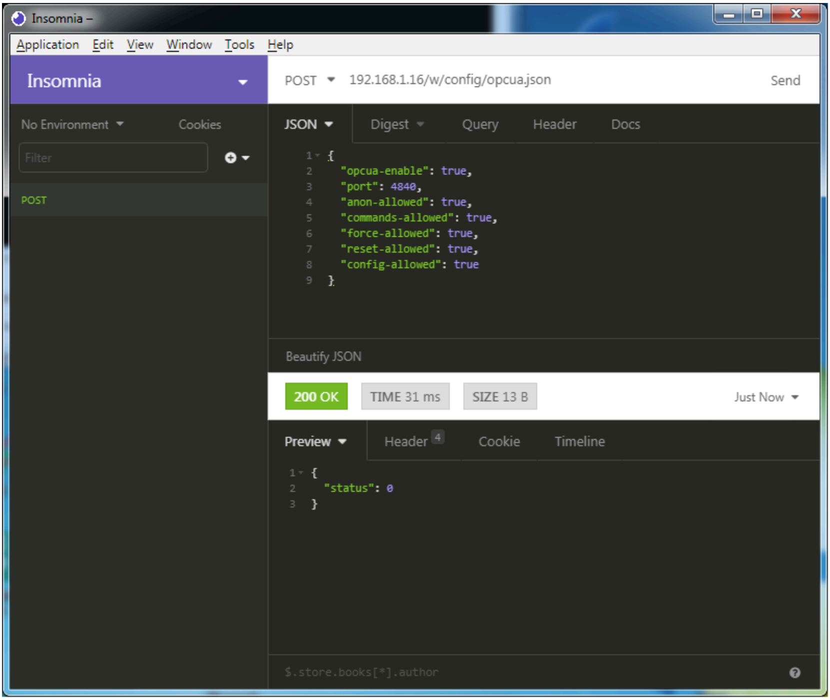
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

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/

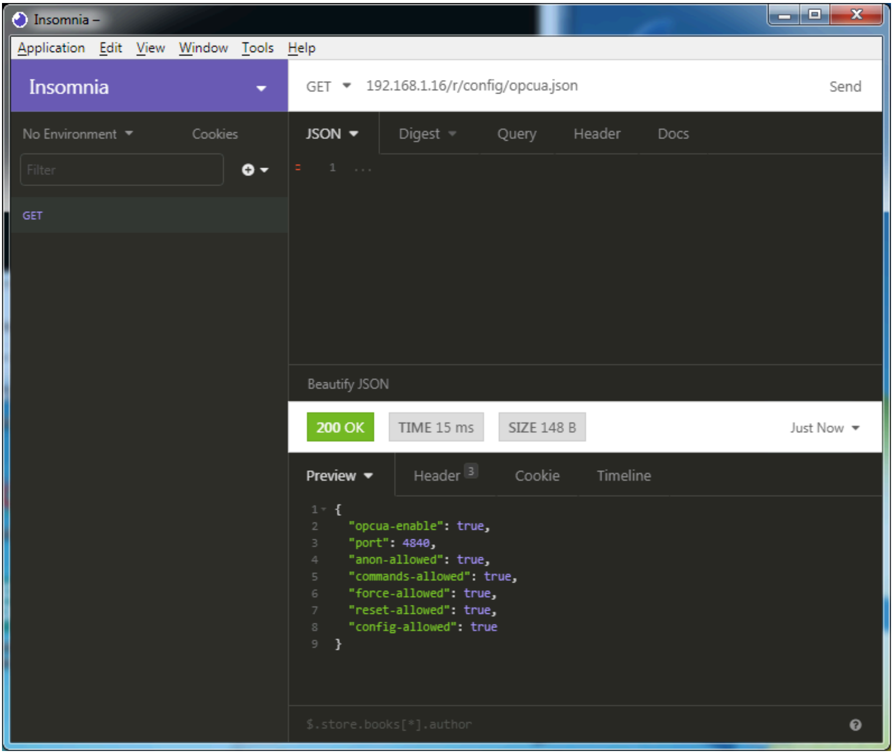
2. Configure OPC UA:

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 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.

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

10.3.1 Standard device information

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

The goal of the "Standard device information" request is to get a complete snapshot of the current device status. The format is JSON.

10.3.2 Structure

Name	Data type	Description	Example
name	string	Device name	"MP08DIO08DIO "
order-id	string	Ordering number	"381166718"
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
CHANNEL Object			
name	string	Name of channel	
type	number	Hardware channel type as number: 0 = DIO 1 = Input 2 = Output 3 = Input/Output 4 = Channel not available 5 = Channel not available 6 = Channel not available 7 = Channel not available 8 = Channel not available	
type_text	string	Textual representation of the channel type	
config	number	Current configuration of the channel: 0 = DIO 1 = Input 2 = Output 3 = Channel not available 4 = Deactivated 5 = Channel not available	
config_text	string	Textual representation of the current config	
inputState	boolean	Input data (producing data) bit to the PLC	
outputState	boolean	Output data bit to the physical output pin	
forced	boolean	True, if the output pin of this channel is forced	
simulated	boolean	True, if the input value to the PLC of this channel is simulated	
actuatorDiag	boolean	True, if the output is in short circuit / overload condition	
sensorDiag	boolean	True, if the sensor supply (Pin 1) is in short circuit / overload condition	

Name	Data type	Description	Example
maxOutputCurrent_mA	number	Maximum output current of the output in mA	
current_mA	number	Measured current of the output in mA (if current measurement is available)	
voltage_mV	number	Measured voltage of this output in mV (if voltage measurement is available)	
PORT Object			
port_type	string	Textual representation of the port type	
aux_mode	number	Indicates the configured mode for the Pin 2: 0 = No AUX 1 = AUX output (always on) 2 = Digital output (can be controlled by cyclic data) 3 = Digital input	
aux_text	string	Textual representation of the current aux mode	"AUX Output"
ds_fault	number	Data storage error number	
ds_fault_text	string	Textual data storage error.	
diag	array of DIAG (n)	Array of port related events	
DIAG Object			
error	number	Error code	
source	string	Source of the current error.	"device" "master"
message	string	Error message	"Supply Voltage fault"
FORCING Object		Forcing information of the device	
forcingActive	boolean	Force mode is currently active	
forcingPossible	boolean	True, if forcing is possible and force mode can be activated	
AuthPossible	boolean	True, if the JSON Interface can obtain forcing autorization	
ownForcing	boolean	True, if forcing is performed by REST API at the moment	
currentClient	string	Current forcing client identifier	

Name	Data type	Description	Example
digitalOutForced	array of numbers (2)	The force values of all 16 digital output channels.	
digitalOutMask	array of numbers (2)	The forcing mask of all 16 digital output channels.	
digitalInForced	array of numbers (2)	The force values of all 16 digital input channels.	
digitalInMask	array of numbers (2)	The forcing mask of all 16 digital input channels.	

10.3.3 Configuration and forcing

Method:	POST
URL:	`<ip>/w/force.json`
Parameters:	None
Post-Body:	JSON Object

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

Table 27: 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", "off", "aux"	
inlogica	string	"no","nc"	
inlogicb	string	"no","nc"	

Table 28: 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 29: Digital object

10.4 CoAP server

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

CoAP is based on the Server-Client principle and a service layer protocol that lets nodes and machines communicate with each other. The 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 93.

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 30: 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	

Table 31: REST API access via CoAP

10.4.3 CoAP configuration - Quick start guide

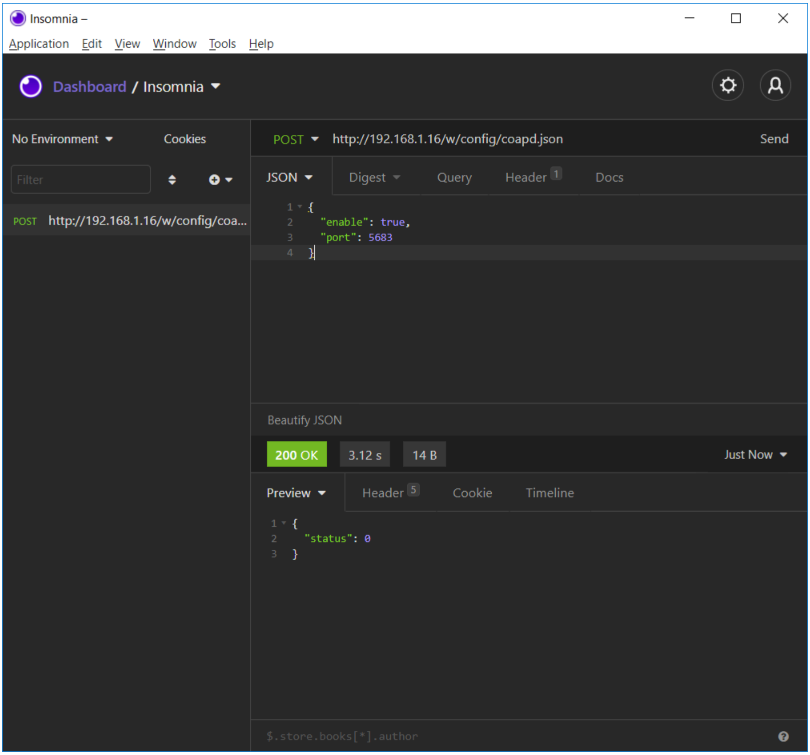
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

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/

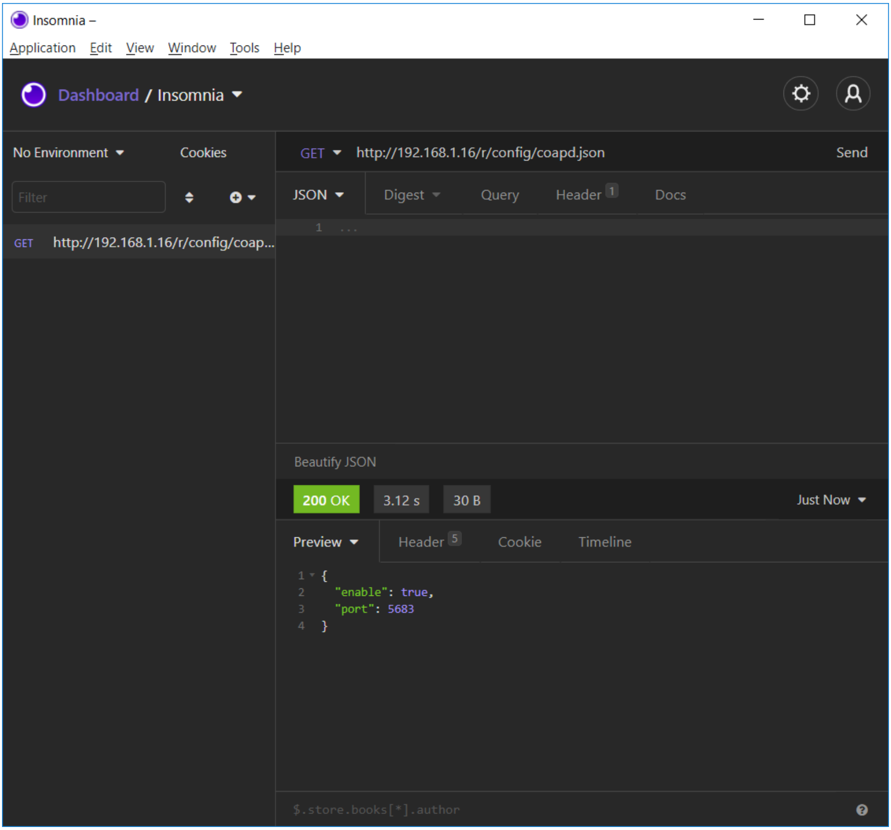
2. Configure CoAP:

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



3. Read CoAP configuration:

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



10.5 Syslog

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 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 98.

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 32: 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

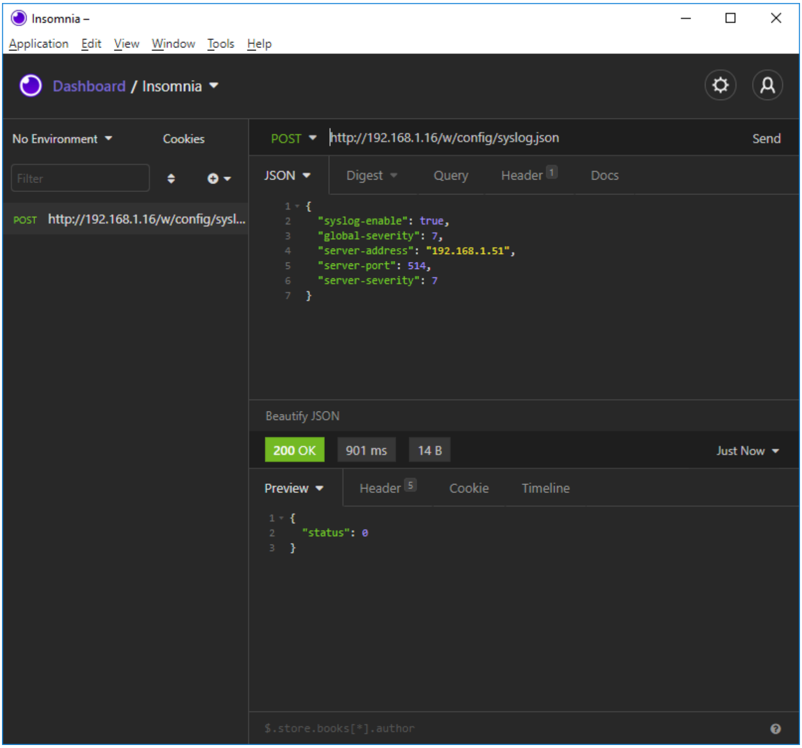
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

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/

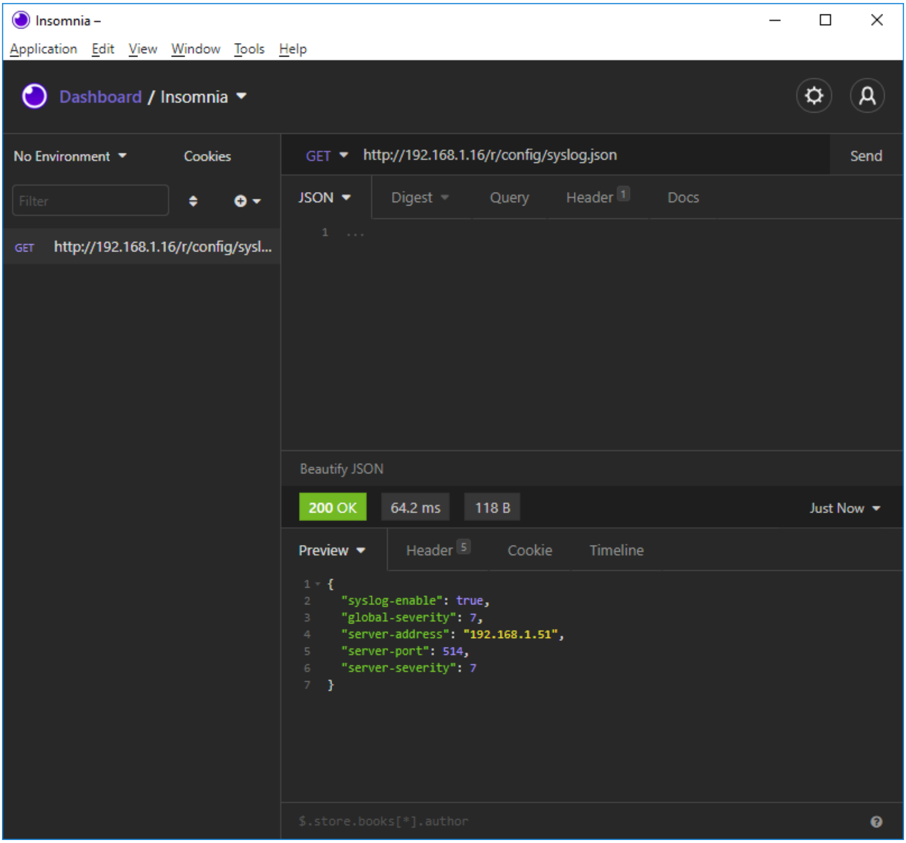
2. Configure Syslog:

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 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 102.

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 33: 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

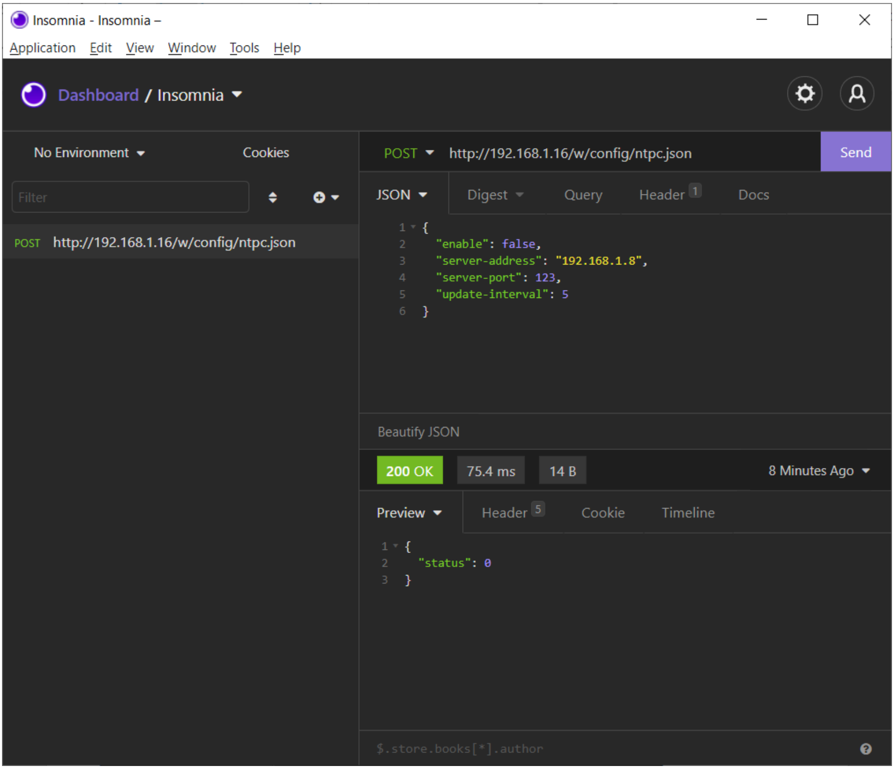
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

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/

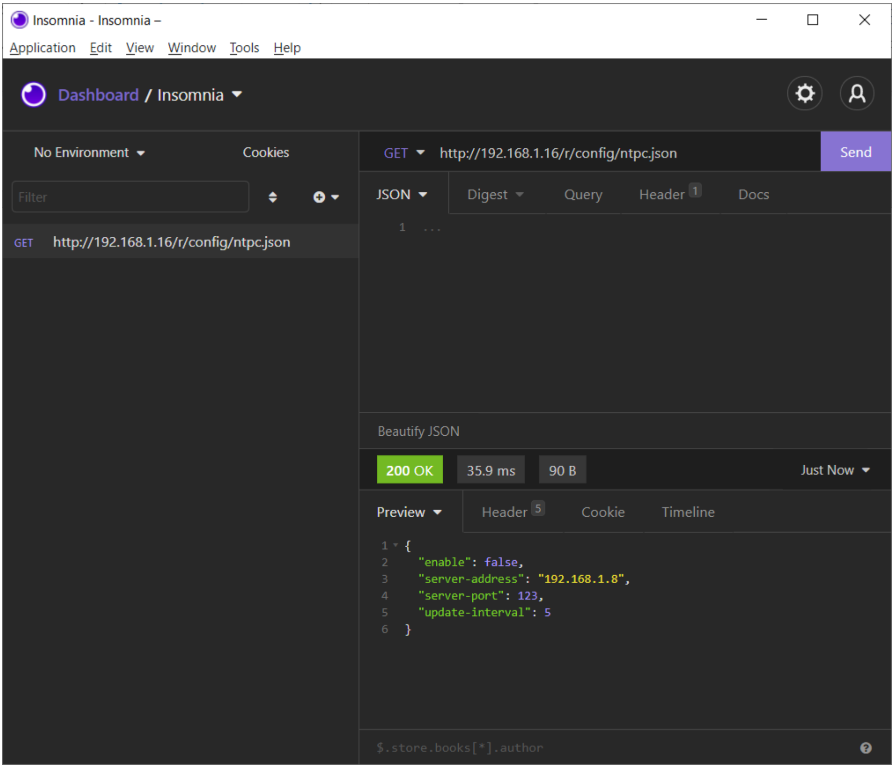
2. Configure NTP:

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



3. Read NTP configuration:

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



11 The integrated Web server

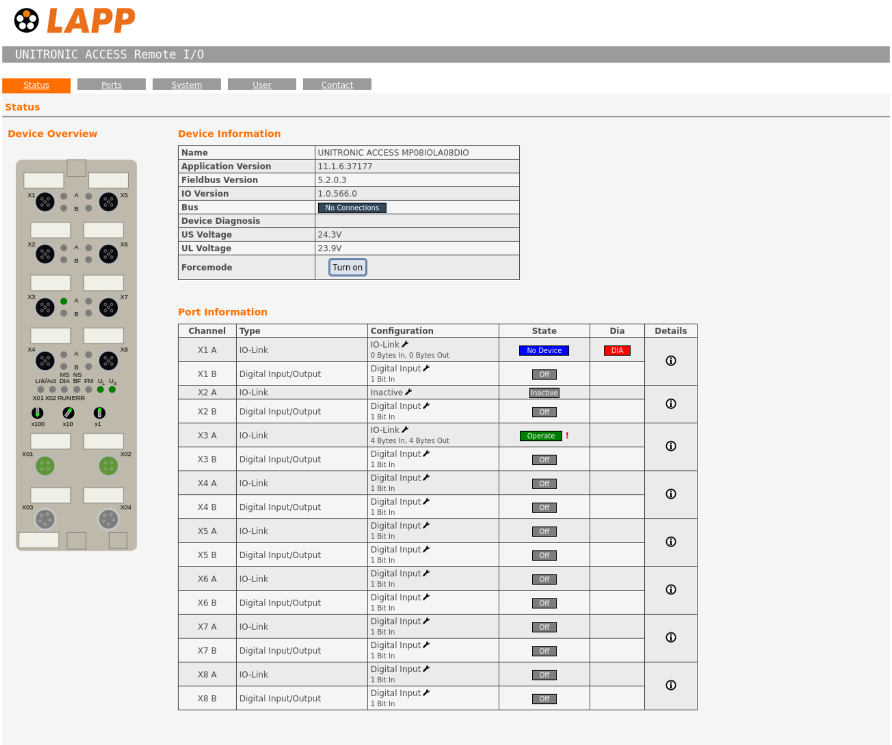
All device variants are equipped with an integrated Web server which makes functions for the device configuration and the display of status and diagnostic information available via a Web interface.

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://" 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


UNITRONIC ACCESS Remote I/O

Status | Ports | System | User | Contact

Status

Device Overview

Device Information

Name	UNITRONIC ACCESS MP08IOLA08DIO
Application Version	11.1.6.37177
Fieldbus Version	5.2.0.3
IO Version	1.0.566.0
Bus	No Connections
Device Diagnosis	
US Voltage	24.3V
UL Voltage	23.9V
Forcemode	Turn on

Port Information

Channel	Type	Configuration	State	Dia	Details
X1 A	IO-Link	IO-Link 0 Bytes In, 0 Bytes Out	No Device	DIA	ⓘ
X1 B	Digital Input/Output	Digital Input 1 Bit In	Off		
X2 A	IO-Link	Inactive	Inactive		ⓘ
X2 B	Digital Input/Output	Digital Input 1 Bit In	Off		
X3 A	IO-Link	IO-Link 4 Bytes In, 4 Bytes Out	Operate !		ⓘ
X3 B	Digital Input/Output	Digital Input 1 Bit In	Off		
X4 A	IO-Link	Digital Input 1 Bit In	Off		ⓘ
X4 B	Digital Input/Output	Digital Input 1 Bit In	Off		
X5 A	IO-Link	Digital Input 1 Bit In	Off		ⓘ
X5 B	Digital Input/Output	Digital Input 1 Bit In	Off		
X6 A	IO-Link	Digital Input 1 Bit In	Off		ⓘ
X6 B	Digital Input/Output	Digital Input 1 Bit In	Off		
X7 A	IO-Link	Digital Input 1 Bit In	Off		ⓘ
X7 B	Digital Input/Output	Digital Input 1 Bit In	Off		
X8 A	IO-Link	Digital Input 1 Bit In	Off		ⓘ
X8 B	Digital Input/Output	Digital Input 1 Bit In	Off		

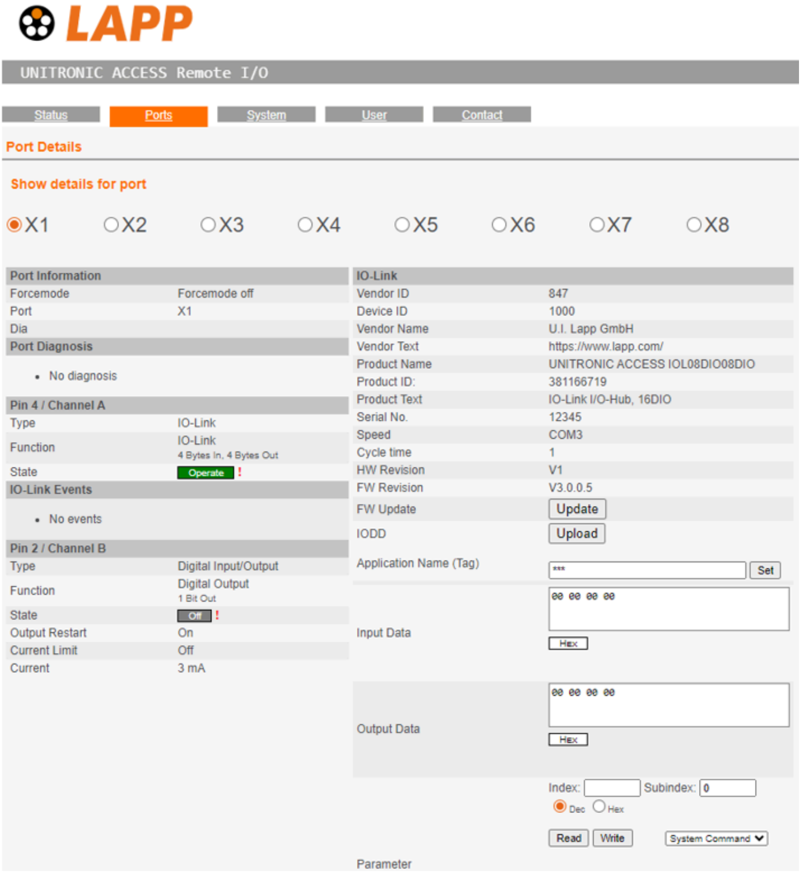
The status page provides a quick overview of the current state of the device.

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

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

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

11.1.2 The Ports page



The page shows detailed port information. In the field **Port Diagnosis**, incoming and outgoing diagnostics are displayed as clear text. **Pin 2** and **Pin 4** contain information about the configuration and state of the port.

11.1.3 The System page

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

Restart Device

The module initializes a software reset.

Reset to Factory Settings

The module restores to the default factory settings.

IP Settings

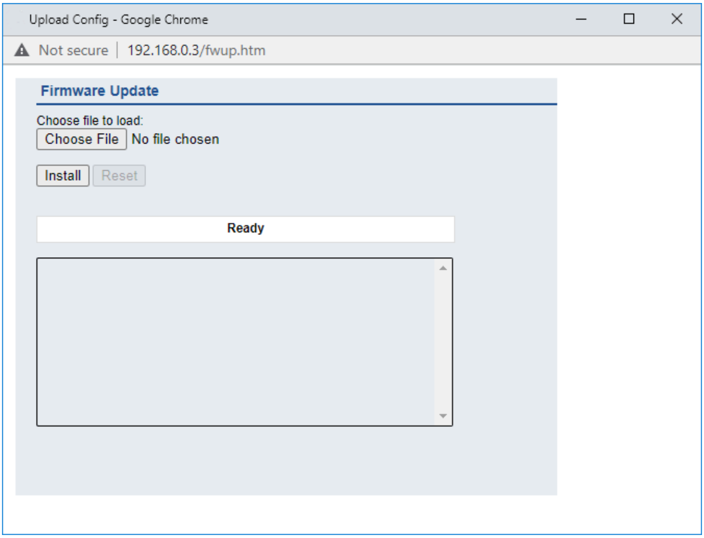
Use this parameter to change the current IP address of the module.

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

Firmware Update

The module initializes a Firmware update.

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



11.1.4 The User page

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

Default user login data:

- User: admin
- Password: private

12 Technical data

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

12.1 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)	MP08DIO08DIO	-40 °C .. +70 °C (-40 °F .. +158 °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 34: General information

2 Not under UL investigation.

12.2 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	41000 (MP08DIO08DIO, 381166718)
Supported Ethernet protocols	Ping ARP HTTP 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 35: Modbus TCP protocol

12.3 Power supply of the module electronics/ sensors

<table>
<tr><td>Port X03, X04</td><td colspan="3">M12-L-coded Power, connector/socket, 5-pole
Pin 1 / Pin 3</td></tr>
<tr><td>Nominal voltage U_S</td><td colspan="3">24 V DC (SELV/PELV)</td></tr>
<tr><td>Current U_S</td><td colspan="3">Max. 16 A</td></tr>
<tr><td>Voltage range</td><td colspan="3">21 .. 30 V DC</td></tr>
<tr><td>Power consumption of module electronics</td><td colspan="3">Typically 160 mA (+/-20 % at U_S nominal voltage)</td></tr>
<tr><td>Power supply interruption</td><td colspan="3">Max. 10 ms</td></tr>
<tr><td>Voltage ripple U_S</td><td colspan="3">Max. 5 %</td></tr>
<tr><td>Current consumption sensor system
(Pin 1)</td><td>MP08DIO08DIO</td><td>Port X1 .. X8
(Pin 1)</td><td>max. 4 A per port
(at $T_{ambient}$ = 30° C)</td></tr>
<tr><td>Voltage level of the sensor power supply</td><td colspan="3">Min. (U_S – 1.5 V)</td></tr>
<tr><td>Short circuit/overload protection of sensor supply</td><td colspan="3">Yes, per port</td></tr>
<tr><td>Reverse polarity protection</td><td colspan="3">Yes</td></tr>
<tr><td rowspan="2">Operational indicator (U_S)</td><td>LED green:</td><td colspan="2">18 V (+/- 1 V) < U_S</td></tr>
<tr><td>LED red:</td><td colspan="2">U_S < 18 V (+/- 1 V)</td></tr>
</table>

Table 36: Information on the power supply of the module electronics/ sensors

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

- Max. 4.0 A per port
- Max. 5.0 A for each port pair X1/X2, X3/X4, X5/X6, X7/X8

▶ Max. 9.0 A in total for the whole port group X1 .. X8

Pay attention to the derating!

12.4 Power supply of the actuators

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

Table 37: Information on the power supply of the actuators

12.5 I/O ports

MP08DIO08DIO	Ports X1 .. X8	DI, DO	M12 socket, 5-pin

Table 38: I/O ports: Overview of functions

12.5.1 Digital inputs

Input connection	MP08DIO08DIO		Type 3 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	MP08DIO08DIO	X1 .. X8	16
Status indicator	yellow LED for Channel A (Pin 4) white LED for Channel B (Pin 2)		
Diagnostic indicator	red LED per port		

Table 39: I/O ports configured as digital input

12.5.2 Digital outputs

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

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

Pay attention to the derating!

Output type	normally open, p-switching	
Nominal output voltage per channel	Signal status “1” Signal status “0”	min. (U_L -1 V) max. 2 V
Max. output current per device	MP08DIO08DIO	9 A
Max. output current per channel	MP08DIO08DIO (X1 .. X8)	2 A
Short-circuit/overload protected	yes/yes	
Behavior in case of short circuit or overload	deactivation with automatic power-on (parameterized)	
Number of digital outputs	MP08DIO08DIO (X1 .. X8)	16
Status indicator	yellow LED per output Channel A (Pin 4) white LED per output Channel B (Pin 2)	
Diagnostic indicator	red LED per channel	

Table 40: I/O ports configured as digital output

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

12.6 LEDs

LED	Color	Description
U_L	Green	Auxiliary sensor/actuator voltage OK 18 V (+/- 1 V) < U_L < 30 V (+/- 1 V)
	Red*	Auxiliary sensor/actuator voltage LOW U_L < 18 V (+/- 1 V) or U_L > 30 V (+/- 1 V) * if "Report U_L supply voltage fault" is enabled.
	OFF	None of the above conditions.
U_S	Green	System/sensor voltage OK 18 V (+/- 1 V) < U_S < 30 V (+/- 1 V)
	Red	System/sensor voltage LOW U_S < 18 V (+/- 1 V) or U_S > 30 V (+/- 1 V)
	Red flashing	Device performs a factory reset (position of rotary encoding switches: 9-7-9)
	OFF	None of the above conditions.
X1 .. X8 A	Yellow	Status of digital input or digital output on pin 4 line "on".
	Red	Short circuit on pin 4 line. / Overload or short circuit on L+ (pin 1) line / communication error
	OFF	None of the above conditions.
X1 .. X8 B	White	Status of digital input or digital output on pin 2 line "on".
	Red	Short circuit on pin 2 line. / 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 double flash	Firmware update
	OFF	None of the above conditions.

Table 41: Information on the LED colors

12.7 Data transfer times

The following tables give an overview of the internal data transfer times of UNITRONIC® ACCESS.

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

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

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

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

Use case 1:

UNITRONIC® ACCESS Digital-I/O configuration with enabled Web interface and *disabled* IIoT protocols

16DIO variant (MP08DIO08DIO):

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

Use case 2:

UNITRONIC® ACCESS Digital-I/O configuration with enabled Web interface and *enabled* IIoT protocols

16DIO variant (MP08DIO08DIO):

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

13 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 Digital-I/O multi-protocol: MP08DIO08DIO (16 × Input/Output)

Technical Support
https://www.lapp.com

Contents

1 About this manual

1.1 General information

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

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

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

U.I. Lapp GmbH

Schulze-Delitzsch-Straße 25
D-70565 Stuttgart
Germany

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	09/2024	

Table 1: Overview of manual revisions

2 Safety instructions

2.1 Intended use

The products described in this manual are decentralized I/O Devices on an Industrial Ethernet Network.

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

The modules fulfill the requirements of the EMC guidelines (89/336/EEC, 93/68/EEC and 93/44/EEC) and the low voltage guideline (73/23/EEC).

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

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

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

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

Only install cables and accessories that fulfill the requirements and regulations for safety, electromagnetic compatibility, and, where applicable, telecommunication end devices, as well as the specification information. Information on which cables and accessories are permitted for the installation can be obtained from 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 this manual are authorized to install and maintain the devices described. These are persons who

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

Only 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 manual, can result in serious personal injury or damage to equipment.

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

3 Designations and synonyms

AOI	Add-On Instruction
API	Application Programming Interface
BF	Bus Fault LED
Big Endian	Data format with High-B on first place (PROFINET)
BUI	Back-Up Inconsistency (EIP diagnostics)
CC	CC-Link IE Field
Ch. A	Channel A (Pin 4) of I/O port
Ch. B	Channel B (Pin 2) of I/O port
CIP	Common Industrial Protocol (media independent protocol)
CoAP	Constrained Application Protocol
CSP+	Control & Communication System Profile Plus
DCP	Discovery and Configuration Protocol
DevCom	Device Comunicating (EIP diagnostics)
DevErr	Device Error (EIP diagnostics)
DI	Digital Input
DIA	Diagnostic LED
DO	Digital Output
DIO	Digital Input/Output
DTO	Device Temperature Overrun (EIP diagnostics)
DTU	Device Temperature Underrun (EIP diagnostics)
DUT	Device under test
EIP	EtherNet/IP
ERP	Enterprise Resource Planning system
ETH	ETHERNET
FE	Functional Earth
FME	Force Mode Enabled (EIP diagnostics)
FSU	Fast Start-Up

GSDML	General Station Description Markup Language
High-B	High-Byte
ICT	Invalid Cycle Time (EIP diagnostics)
IIoT	Industrial Internet of Things
ILE	Input process data Length Error (EIP diagnostics)
IME	Internal Module Error (EIP diagnostics)
I/O	Input / Output
I/O port	X1 .. X8
I/O port pin 2	Channel B of X1 .. X8
I/O port pin 4 (C/Q)	Channel A of X1 .. X8
IVE	IO-Link port Validation Error (EIP diagnostics)
I&M	Identification & Maintenance
JSON	JavaScript Object Notation (platform independent data format)
L+	I/O port pin 1, sensor power supply
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
OLE	Output process data Length Error (EIP diagnostics)
OPC UA	Open Platform Communications Unified Architecture (platform independent, service-oriented architecture)

PLC	Programmable Logic Controller
PN	PROFINET
PWR	Power
REST	REpresentational State Transfer
RFC	Request for Comments
RPI	Requested Packet Interval
RWr	Word data input as seen from the master station (CC-Link)
RWw	Word data output as seen from the master station (CC-Link)
RX	Bit data input as seen from the master station (CC-Link)
RY	Bit data output as seen from the master station (CC-Link)
SCA	Short Circuit Actuator/U_L/U_{AUX} (EIP diagnostics)
SCS	Short Circuit Sensor (EIP diagnostics)
SLMP	Seamless Message Protocol
SNMP	Simple Network Management Protocol
SP	Single Protocol (PROFINET, EtherNet/IP, EtherCAT®, Modbus TCP or CC-Link IE Field Basic)
SPE	Startup Parameterization Error (EIP diagnostics)
U_{AUX}	$U_{Auxiliary}$, supply voltage for the load circuit (Actuator supply on Class B ports)
UDP	User Datagram Protocol
UDT	User-Defined Data Types
UINT8	Byte in PLC (IB, QB)
UINT16	Unsigned integer with 16 bits or word in PLC (IW, QW)
U_L	U_{Load}, supply voltage for the load circuit (Actuator supply on Class A)
UL	Underwriters Laboratories Inc. (certification company)
UTC	Coordinated Universal Time (Temps Universel Coordonné)

Table 2: Designations and synonyms

4 System description

4.1 Device variants

The following Digital I/O device variants are available in the UNITRONIC® ACCESS family:

Article number	Product designation	Description	I/O port functionality
381166718	MP08DIO08DIO	UNITRONIC® ACCESS M12-60 mm, I/O Device Multi-protocol (PN, EIP, EC, MB, CC) Security	16 x Input/Output universal

Table 3: Overview of UNITRONIC® ACCESS Digital-I/O variants

4.2 I/O port overview

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

UNITRONIC® ACCESS 16DIO ports

Device variant	Port	Pin 1 U_S	Pin 4 / Ch. A (In/Out)		Pin 2 / Ch. B (In/Out)	
MP08DIO08DIO	**Info:**	–	Type 3	Supply by U_L	Type 3	Supply by U_L
	X8:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)
	X7:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)
	X6:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)
	X5:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)
	X4:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)
	X3:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)
	X2:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)
	X1:	U_S (4 A)	DI	DO (2 A)	DI	DO (2 A)

Table 4: Port configuration of MP08DIO08DIO variant

5 Overview of product features

5.1 CC-Link IE Field Basic product features

CC-Link IE Field Basic network

- Number of stations: 1
- 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 Integrated Web server

Network parameter display

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

Displaying diagnostics

View diagnostics via the integrated Web server.

User management

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

5.3 Security features

Firmware signature

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

Syslog

The 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.

5.4 Other features

Interface protection

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

For more details, see section Port assignments on page 23.

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 Digital-I/O multi-protocol variants

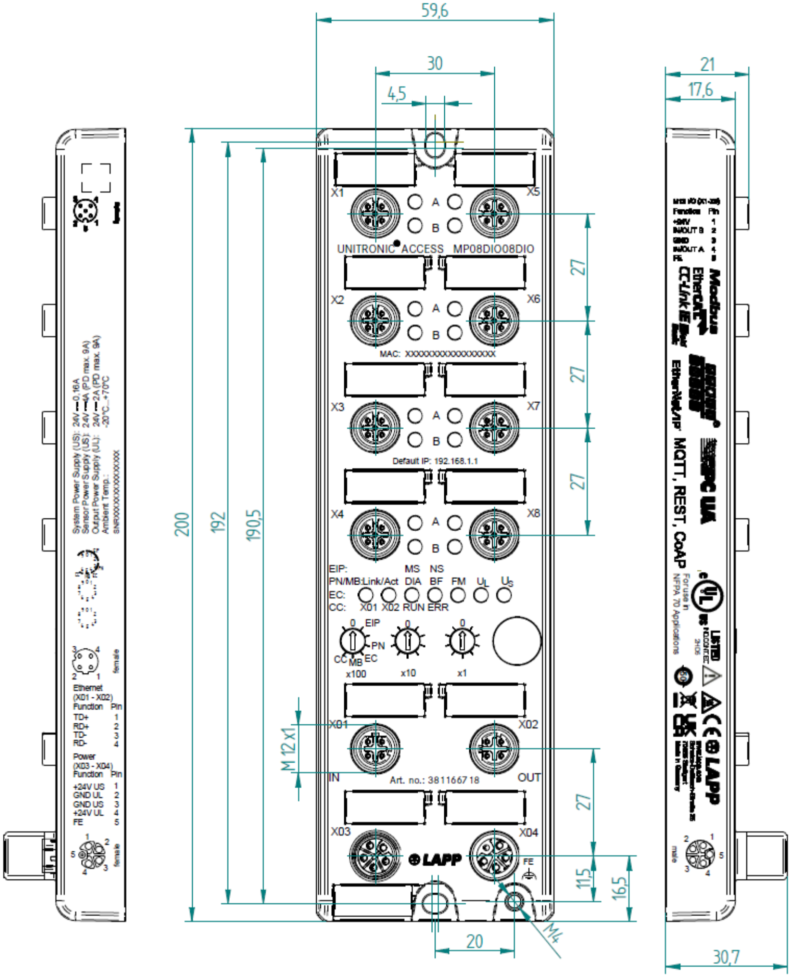


Figure 1: MP08DIO08DIO

6.2.2 Notifications

Attention:

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

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

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

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

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

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

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

6.3 Port assignments

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

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

Color coding: green

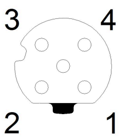


Figure 2: Schematic drawing, ports X01, X02

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

Table 5: Assignment of ports X01, X02

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

6.3.2 Power supply with M12 power L-coded

Color coding: gray

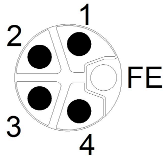


Figure 3: Schematic diagram of the M12 L-coding (connector X03 for Power In)

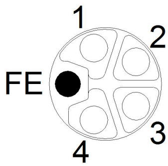


Figure 4: Schematic diagram of the M12 L-coding (socket X04 for Power Out)

Power supply	Pin	Signal	Function
	1	U_S (+24 V)	Sensor/system power supply
	2	GND_U_L	Ground/reference potential U_L
	3	GND_U_S	Ground/reference potential U_S
	4	U_L (+24 V)	Load supply Actuator supply
	5	FE	Functional ground

Table 6: Pin assignments ports X03 and X04

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

6.3.3 I/O ports as M12 sockets

Color coding: black

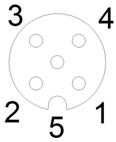


Figure 5: Schematic drawing I/O port as M12 socket

6.3.3.1 I/O ports

MP08DIO08DIO	Pin	Signal	Function
16DIO X1 .. X8	1	+24 V	power supply +24 V
	2	IN/OUT	Ch. B: Digital input or digital output
	3	GND	Ground/reference potential
	4	IN/OUT	Ch. A: Digital input or digital output
	5	FE	Functional ground

Table 7: Pin assignments I/O ports

7 Starting operation

7.1 CSP+ file

The CSP+ file describes the information of a CC-Link device and can be installed in the engineering tool to configure the UNITRONIC® ACCESS variants. Each of the UNITRONIC® ACCESS device variants requires its own CSP+ file.

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

The CSP+ file and the required icons are grouped together in an archive file named "0x0670_UNITRONIC ACCESS MP08DIO08DIO_1.0_en.cspp.zip".

"0x4338" stands for the vendor ID of U.I. Lapp GmbH, "MP08DIO08DIO" is the model number of the UNITRONIC® ACCESS variant.

Download the CSP+ file and install it 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. See chapter Configuration and operation with GxWorks3 on page 47.

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:	MP08DIO08DIO
Vendor code:	1247 (Hexadecimal: 17208)
Product type:	CC-Link IE Field Basic Slave Station

7.4 Setting the rotary encoding switches

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

- MP08DIO08DIO

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.

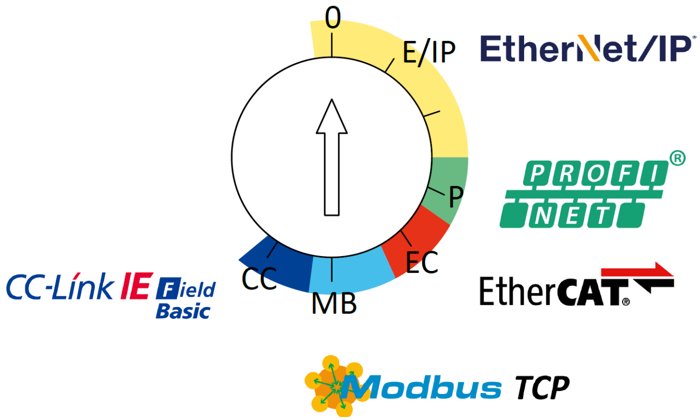
After selecting a protocol followed by a restart of the device, the new protocol is initialized. This can take up to 15 seconds. During this time the device is not usable and the LED indicators are out of function. When the protocol change is complete, the LED indicators return to normal operation and 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 Digital I/O devices 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 I/O Device in different environments.

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

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



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

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

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

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

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

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

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

To change the protocol, carry out a factory reset. In this way you restore the factory settings of the respective device. How you perform the factory reset for your device is described in chapter Factory reset on page 31.

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

7.4.1 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 9: 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 is blinking red. After the internal memory write processes have finished, the U_S LED returns to display static green or red light, in dependency of the actual U_S voltage.

	x100	x10	x1
Factory Reset	9	7	9

Follow the steps from section Setting the rotary encoding switches on page 28 again to select a new protocol.

For performing a factory reset via software configuration, see chapter OPC UA configuration on page 74 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 SLMP, the Web server or IIoT protocols. Acyclic messages over SLMP are sent to read and write the configuration. When sending, all existing parameters will be overwritten by this data. Therefore the content of the SLMP 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.

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

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

The following chapters represent different setting groups with its configuration parameters.

8.1 General settings

Setting	Description	Default value
Suppress U_L Diagnosis Mode	0 = Diagnosis enabled 1 = Diagnosis suppressed 2 = Auto	0
Suppress Actuator Diagnosis without U_L	0 = Diagnosis enabled 1 = Diagnosis suppressed	0
Suppress U_S Diagnosis	0 = Diagnosis enabled 1 = Diagnosis suppressed	0
Reserved	Reserved	0
Web Interface Lock	0 = Web Interface enabled 1 = Web Interface locked	0
Forcing Lock	0 = Forcing Lock disabled 1 = Forcing Lock enabled	0
Reserved	Reserved	0
Reserved	Reserved	0
External Configuration Lock	0 = External configuration enabled 1 = External configuration locked	0

8.1.1 Report U_L supply voltage fault

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

8.1.2 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.

8.1.3 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.

8.1.4 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.

8.1.5 Web interface lock

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

8.1.6 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
Surv. Timeout X1.A .. X8.B	DO Surveillance Timeout Valid values: 0 .. 255	80
Failsafe Mode X1.A .. X8.B	Failsafe mode 0: Set Low 1: Set High 2: Hold Last	0
Direction X1.A .. X8.B	Port mode 0: Digital DIO 1: Digital Output 2: Digital Input 3: Deactivated	3
Current limit X1.A .. X8.B	Current limit at Pin 4 (Channel A) or Pin 2 (Channel B) in mA. If a current higher than the given limit (in mA) is measured, the output will be turned off and a diagnosis will be generated. 0 .. 65535	65535 (= unlimited current)
Output Auto Restart X1.A .. X8.B	DO Restart 0: Disable 1: Enable	0
Digital Input Filter X1.A .. X8.B	Digital Input Filter 0: Disabled 10: 1 ms 20: 2 ms 30: 3 ms 60: 6 ms 100: 10 ms 150: 15 ms	0
Digital Input Logic X1.A .. X8.B	Digital Input Logic 0: Normally Open 1: Normally Close	0
I/O Map channel X1.A .. X8.B	I/O Map channel	X1 .. X16

8.2.1 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.

8.2.2 Failsafe Mode

The UNITRONIC® ACCESS devices support a failsafe function for the output data of the digital 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 digital channels is controlled by the configured failsafe values.

Set Low:

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

Set High:

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

Hold Last:

If failsafe is active, all bits of the digital output data are holding the last valid process data state ("0" or "1").

8.2.3 Channel Direction

Digital Input/Output (DIO):

In this mode, the channel operates as digital input/output. The channel can be controlled by the *Digital Output Channel Control* (first two bytes of the output data) and 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).

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.

Deactivated:

The channel is deactivated but can be configured for later use. No diagnostics are generated.

8.2.4 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.

8.2.5 Output Auto Restart

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

If *Output Auto Restart* 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 *Output Auto Restart* 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.

8.2.6 Digital Input 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.

8.2.7 Digital Input 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.

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. The producing data in this context is defined as the process input data which contains the physical inputs, standard diagnostics and optional extended diagnostics.

The following sections 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)

Channel No.	Register for DO (single bit)	Description	Access
X1	Y0	Digital Output control for X1	RW ("Read/Write")
X2	Y1	Digital Output control for X2	RW
X3	Y2	Digital Output control for X3	RW
X4	Y3	Digital Output control for X4	RW
X5	Y4	Digital Output control for X5	RW
X6	Y5	Digital Output control for X6	RW
X7	Y6	Digital Output control for X7	RW
X8	Y7	Digital Output control for X8	RW
X9	Y8	Digital Output control for X9	RW
X10	Y9	Digital Output control for X10	RW
X11	YA	Digital Output control for X11	RW
X12	YB	Digital Output control for X12	RW
X13	YC	Digital Output control for X13	RW
X14	YD	Digital Output control for X14	RW
X15	YE	Digital Output control for X15	RW
X16	YF	Digital Output control for X16	RW

9.2 Producing data (input)

Channel No.	Register for DI (single bit)	Description	Access
X1	X0	Digital Input for X1	R ("Read Only")
X2	X1	Digital Input for X2	R
X3	X2	Digital Input for X3	R
X4	X3	Digital Input for X4	R
X5	X4	Digital Input for X5	R
X6	X5	Digital Input for X6	R
X7	X6	Digital Input for X7	R
X8	X7	Digital Input for X8	R
X9	X8	Digital Input for X9	R
X10	X9	Digital Input for X10	R
X11	XA	Digital Input for X11	R
X12	XB	Digital Input for X12	R
X13	XC	Digital Input for X13	R
X14	XD	Digital Input for X14	R
X15	XE	Digital Input for X15	R
X16	XF	Digital Input for X16	R

10 Diagnostics processing

Sr. No.	Register for Diagnosis	Name	Description	Access
1	X28	U_S supply fault	System supply voltage diagnosis	R ("Read only")
2	X29	U_L supply present	Actuator supply valid range	R
3	X2A	U_L supply fault	Actuator supply fault diagnosis	R
4	X3C	Internal module error	I/O data not reliable due to internal error	R
5	X3D	Force mode diagnosis	Force mode enabled or disabled. Outputs can differ from CC-Link IE Field Basic output values.	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 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 18 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.	Register for Diagnosis	Description	Access
X1	X10	Short circuit X1 Channel A	R ("Read only")
X2	X11	Short circuit X1 Channel B	R
X3	X12	Short circuit X2 Channel A	R
X4	X13	Short circuit X2 Channel B	R
X5	X14	Short circuit X3 Channel A	R
X6	X15	Short circuit X3 Channel B	R
X7	X16	Short circuit X4 Channel A	R
X8	X17	Short circuit X4 Channel B	R
X9	X18	Short circuit X5 Channel A	R
X10	X19	Short circuit X5 Channel B	R
X11	X1A	Short circuit X6 Channel A	R
X12	X1B	Short circuit X6 Channel B	R
X13	X1C	Short circuit X7 Channel A	R
X14	X1D	Short circuit X7 Channel B	R
X15	X1E	Short circuit X8 Channel A	R
X16	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 when a capacitive load is activated or an inductive load is deactivated, and during other voltage peaks when a status changes.

10.4 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 diagnostics in the producing data image are generated.

Port No.	Register for Diagnosis	Description	Access
X1	X20	X1 sensor short circuit	R ("Read only")
X2	X21	X2 sensor short circuit	R
X3	X22	X3 sensor short circuit	R
X4	X23	X4 sensor short circuit	R
X5	X24	X5 sensor short circuit	R
X6	X25	X6 sensor short circuit	R
X7	X26	X7 sensor short circuit	R
X8	X27	X8 sensor short circuit	R

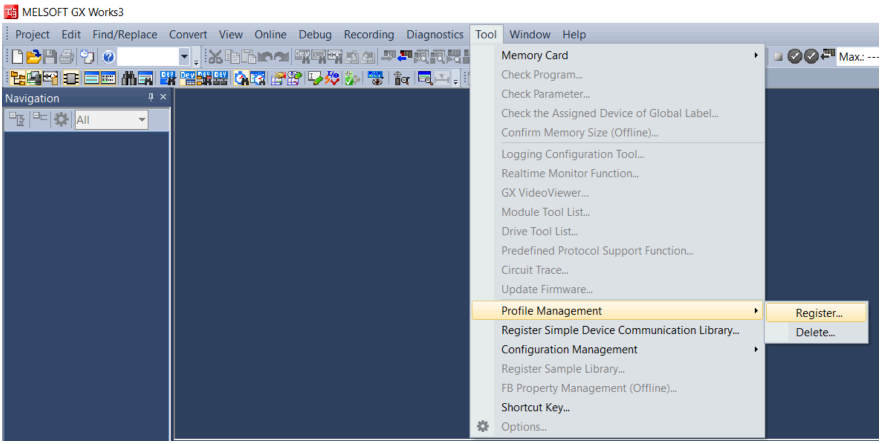
11 Configuration and operation with GxWorks3®

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**.



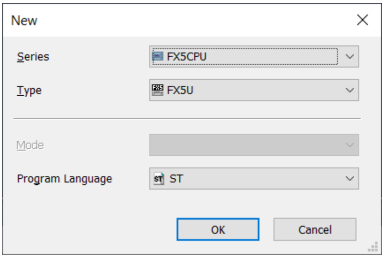
2. Select *0x0670_UNITRONIC ACCESS ...08DIO_1.0_en.cspp.zip* and the CSP+ file will be registered.

11.2 Network parameters

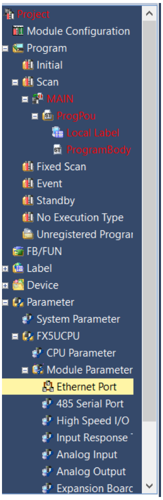
Perform the following work steps to change the Network parameters:

1. Open GxWorks3® and create a new project.

2. Select the series and the type of the used PLC.

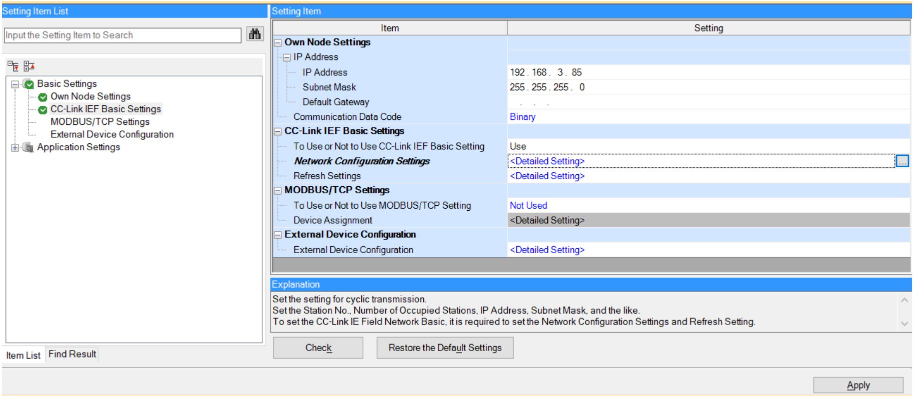


3. To open the setting window, navigate to **Project** > **Parameter** > **"the selected CPU module"** > **Module Parameter**



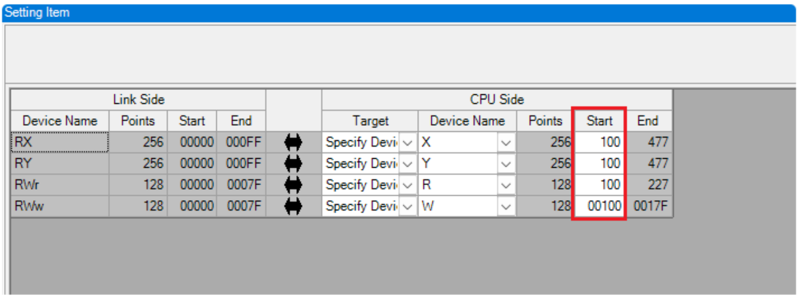
In the appearing window, the CC-Link IE Field Basic Master station can be configured.

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".

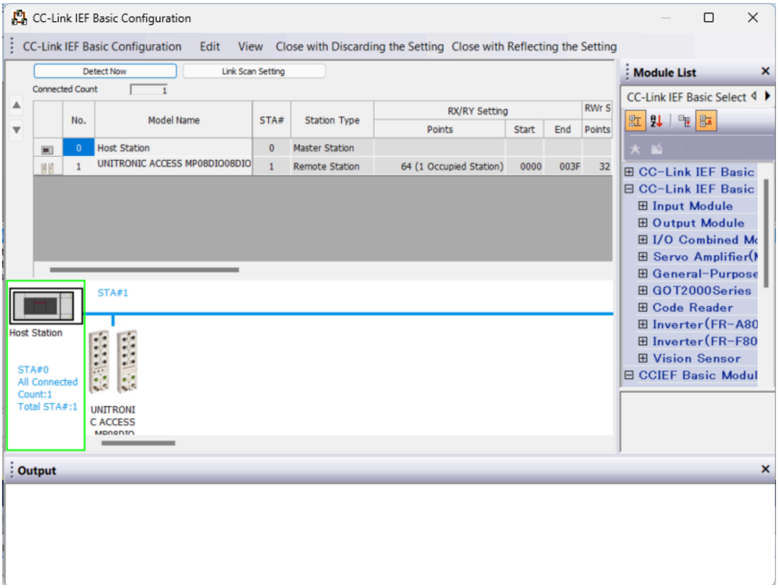
- The option *Network Configuration Settings* allows you to configure a CC-Link IE Field Basic Master, connected stations, a Network, parameters and many more.
- Settings under *Refresh Settings* are necessary for the automatic data transfer between Link side and CPU side:



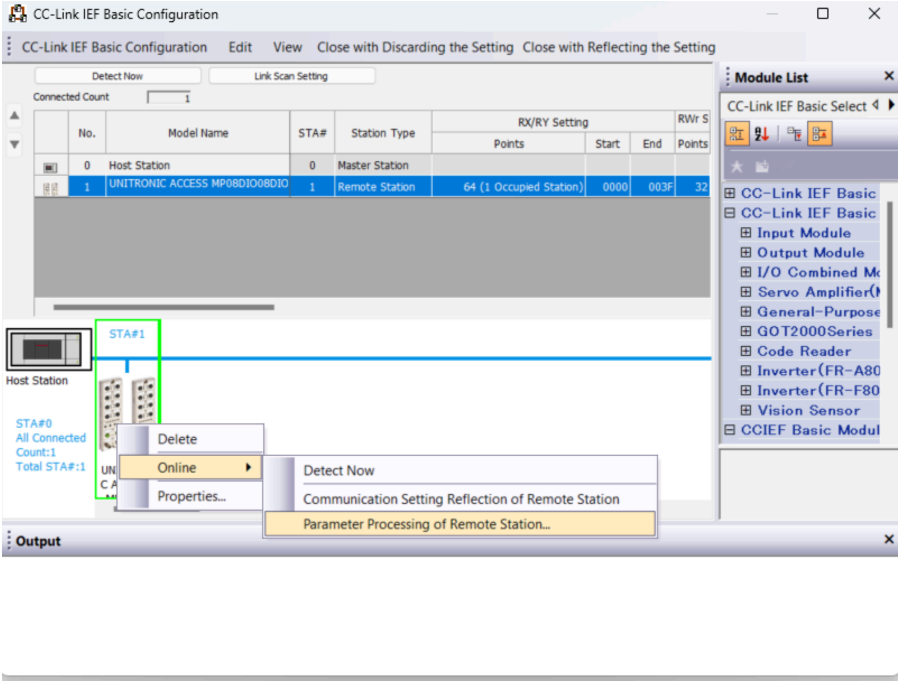
11.3 Parameter processing

Under *Network Configuration Settings*, individual stations can be configured. Perform the following work steps to configure a UNITRONIC® ACCESS device:

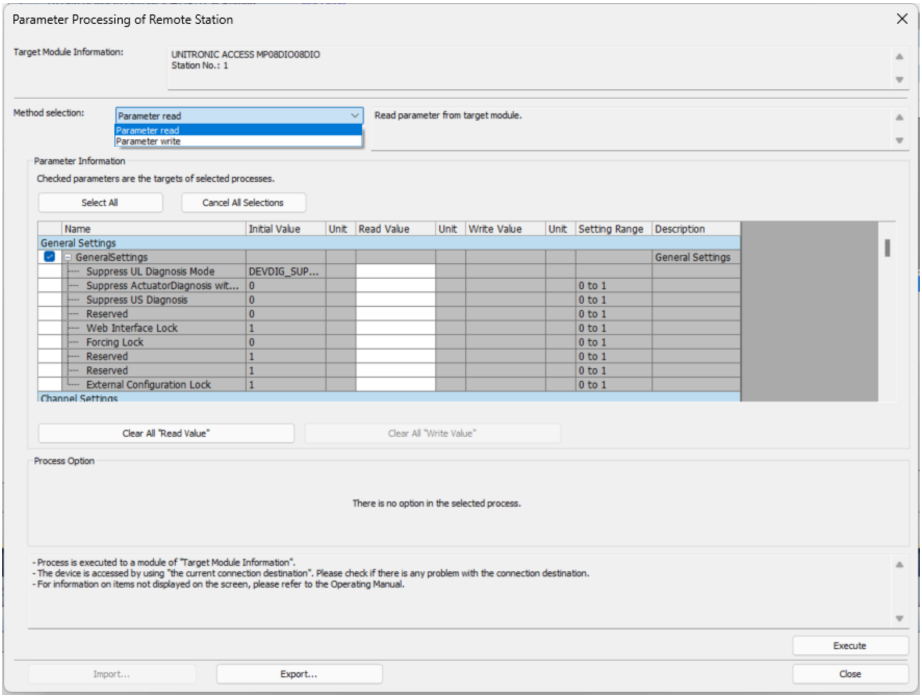
1. Select the UNITRONIC® ACCESS device from the *Module List*. Alternatively, click the button **Detect Now** for automatic detection of devices.



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.



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.

12.1 MQTT

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

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

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 request. For more information see chapter MQTT configuration - Quick start guide on page 72.

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
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 10: 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 57.

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

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 11: Base topic variables on page 57.

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

Variable	Description
mac	The MAC address of the device
name	The name of the device
order	The ordering number of the device
serial	The serial number of the device
ip0 ip1 ip2 ip3	IP address octets

Table 11: Base topic variables

Example:

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

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

`Base-Topic/domain/.….`

There are the following domains:

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

Table 12: Data domains

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

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

Table 13: 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 14: Use case examples

12.1.2.2 Publish topic

Overview of all publish JSON data for the defined topics:

Identity/gateway	
Key	**Data type**
product_name	json_string
ordering_number	json_string
device_type	json_string
serial_number	json_string
mac_address	json_string
production_date	json_string
fw_name	json_string
fw_date	json_string
fw_version	json_string
hw_version	json_string
family	json_string
location	json_string
country	json_string
fax	json_string
vendor_name	json_string
vendor_address	json_string
vendor_phone	json_string
vendor_email	json_string
vendor_techn_support	json_string
vendor_url	json_string
vendor_id	json_integer
device_id	json_integer

Table 15: Identity/gateway

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

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

Table 16: Config/gateway

Status/gateway				
Key	**Data type**	**Range**	**Default value**	**Remarks**
protocol	json_string	PROFINET: ▶ UNKNOWN ▶ OFFLINE ▶ STOP ▶ IDLE ▶ OPERATE EtherNet/IP: ▶ CONNECTED ▶ DISCONNECTED EtherCAT®: ▶ PREOP ▶ SAFEOP ▶ OP ▶ INIT ▶ UNKNOWN Modbus TCP: ▶ No Connections ▶ Connected CC-Link IE Feld Basic: ▶ ON ▶ STOP ▶ DISCONNECTED ▶ ERROR		
system_voltage_fault	json_boolean	true / false		
actuator_voltage_fault	json_boolean	true / false		
internal_module_error	json_boolean	true / false		
simulation_active_diag	json_boolean	true / false		
us_voltage	json_integer	0 .. 32		in Volts
ul_voltage	json_integer	0 .. 32		in Volts
forcemode_enabled	json_boolean	true / false		

Table 17: Status/gateway

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

Table 18: Process/gateway

Identity/port/1 .. 8				
Key	**Data type**	**Range**	**Default value**	**Remarks**
port	json_integer	1 .. 8		
type	json_string	Digital Input DIO Digital Output DIO Pin 4 Only DI Pin 4 Only DO Pin 4 Only Not available Unknown		
max_output_power_cha	json_string	2.0_mA 0.5_mA		
max_output_power_chb	json_string	2.0_mA 0.5_mA		
channel_cha	json_string	Digital Input Digital Output DIO Digital Input/Output Auxiliary Power Auxiliary with DO Not available Unknown		
channel_chb	json_string	Digital Input Digital Output DIO Digital Input/Output Auxiliary Power Auxiliary with DO Not available Unknown		

Table 19: Identity/port/1 .. 8

Config/port/1 .. 8				
Key	**Data type**	**Range**	**Default value**	**Remarks**
port	json_integer	1 .. 8		
direction_cha	json_string	Output Input Inactive Auxiliary Power DIO Unknown		
direction_chb	json_string	Output Input Inactive Auxiliary Power DIO Unknown		
restart_mode_cha	json_string	Manual Auto		
restart_mode_chb	json_string	Manual Auto		
input_polarity_cha	json_string	NO NC		
input_polarity_chb	json_string	NO NC		
input_filter_cha	json_integer			ms
input_filter_chb	json_integer			ms
do_auto_restart_cha	json_boolean	true / false		
do_auto_restart_chb	json_boolean	true / false		
failsafe_cha	json_string	set_low set_high hold_last	set_low	
failsafe_chb	json_string	set_low set_high hold_last	set_low	
surveillance_timeout_cha	json_integer	0 .. 255	80	

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

Table 20: Config/port/1 .. 8

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

Table 21: 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 23: Force object: Digital on page 69)		

Table 22: 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 23: Force object: Digital

[…]/config

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

Property	Data type	Example values	Remarks
portmode	array (Table 25: Config object: Portmode on page 70)		
ip_address	string	"192.168.1.5"	
subnet_mask	string	"255.255.255.0"	
gateway	string	"192.168.1.100"	

Table 24: 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		

Table 25: 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 26: 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

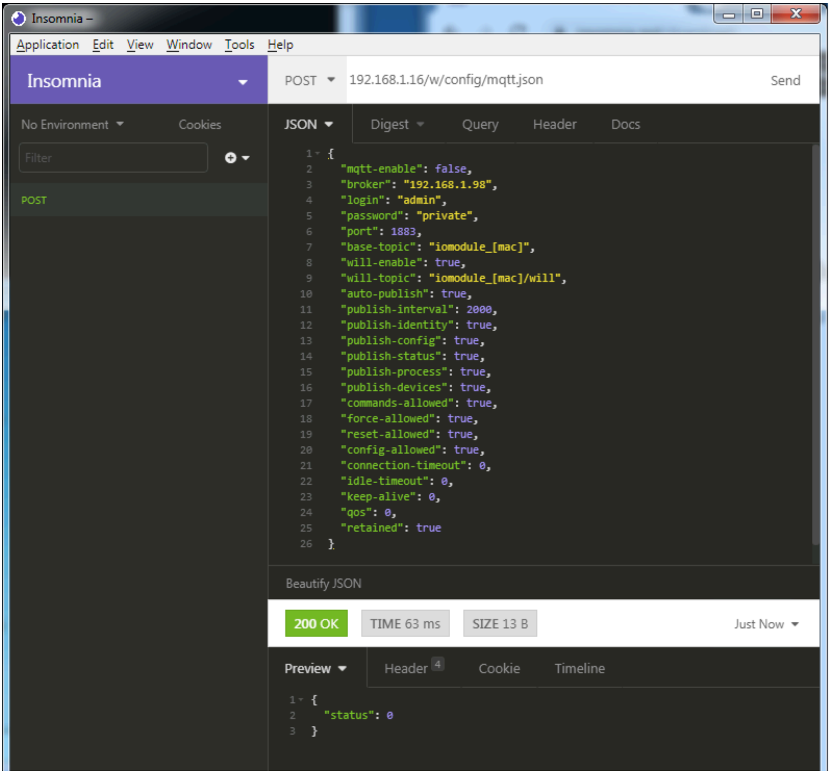
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

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/

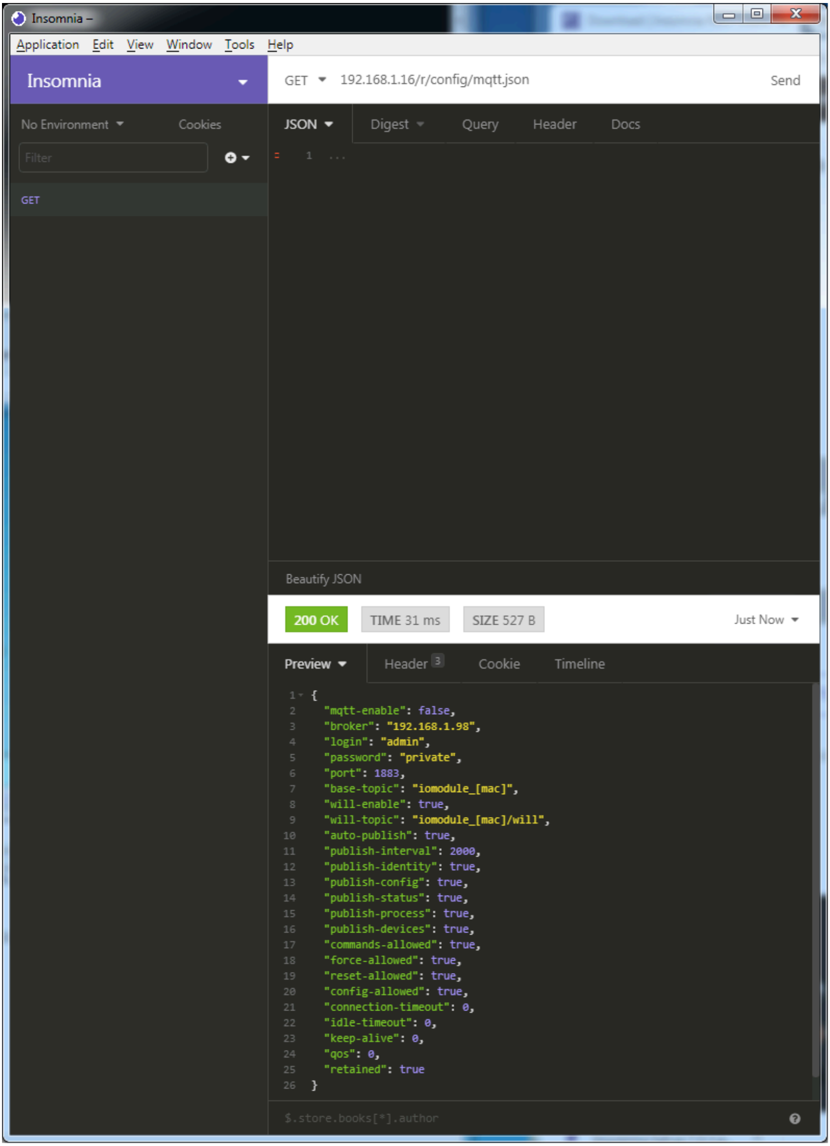
2. Configure MQTT:

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



3. Read MQTT:

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



12.2 OPC UA

OPC 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.

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 request.

The configuration URL is:

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

The configuration can also read back as a JSON file:

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

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

Tree overview of OPC UA objects:

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

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

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

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

Response:

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

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

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

Examples:

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

{"status": 0}

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

12.2.1.1 Gateway objects

Identity

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

Status (read)

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

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

Forcing (read)

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

Config (read + write)

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

Process (read)

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

Commands (write)

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

12.2.1.2 Ports objects

Identity

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

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

See details in Channel objects on page 81.

Status (read)

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

Config (read + write)

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

12.2.1.3 Channel objects

Identity (read)

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

Status (read)

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

Config (read + write)

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

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

Process (read)

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

Forcing (read + write)

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

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

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

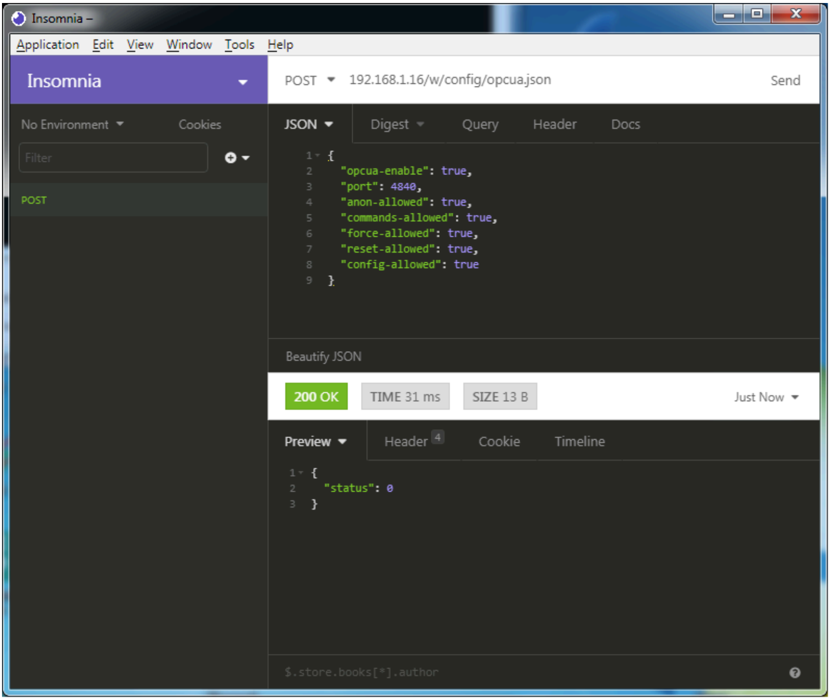
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

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/

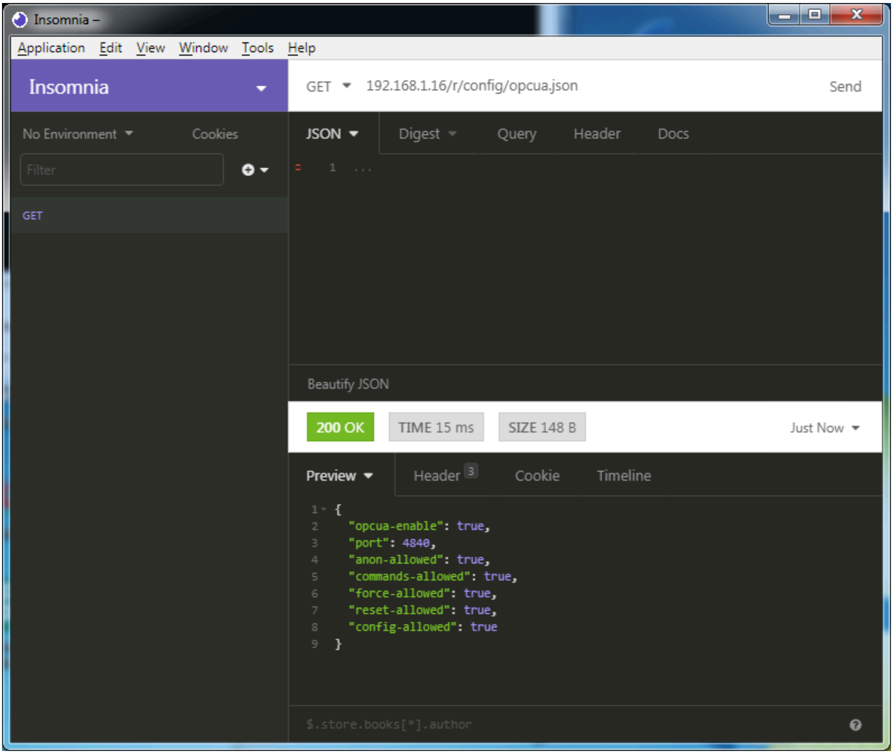
2. Configure OPC UA:

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 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.

The customized LAPP REST API 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.

12.3.2 Structure

Name	Data type	Description	Example
name	string	Device name	"MP08DIO08DIO "
order-id	string	Ordering number	"381166718"
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
CHANNEL Object			
name	string	Name of channel	
type	number	Hardware channel type as number: 0 = DIO 1 = Input 2 = Output 3 = Input/Output 4 = Channel not available 5 = Channel not available 6 = Channel not available 7 = Channel not available 8 = Channel not available	
type_text	string	Textual representation of the channel type	
config	number	Current configuration of the channel: 0 = DIO 1 = Input 2 = Output 3 = Channel not available 4 = Deactivated 5 = Channel not available	
config_text	string	Textual representation of the current config	
inputState	boolean	Input data (producing data) bit to the PLC	
outputState	boolean	Output data bit to the physical output pin	
forced	boolean	True, if the output pin of this channel is forced	
simulated	boolean	True, if the input value to the PLC of this channel is simulated	
actuatorDiag	boolean	True, if the output is in short circuit / overload condition	
sensorDiag	boolean	True, if the sensor supply (Pin 1) is in short circuit / overload condition	

Name	Data type	Description	Example
maxOutputCurrent_mA	number	Maximum output current of the output in mA	
current_mA	number	Measured current of the output in mA (if current measurement is available)	
voltage_mV	number	Measured voltage of this output in mV (if voltage measurement is available)	
PORT Object			
port_type	string	Textual representation of the port type	
aux_mode	number	Indicates the configured mode for the Pin 2: 0 = No AUX 1 = AUX output (always on) 2 = Digital output (can be controlled by cyclic data) 3 = Digital input	
aux_text	string	Textual representation of the current aux mode	"AUX Output"
ds_fault	number	Data storage error number	
ds_fault_text	string	Textual data storage error.	
diag	array of DIAG (n)	Array of port related events	
DIAG Object			
error	number	Error code	
source	string	Source of the current error.	"device" "master"
message	string	Error message	"Supply Voltage fault"
FORCING Object		Forcing information of the device	
forcingActive	boolean	Force mode is currently active	
forcingPossible	boolean	True, if forcing is possible and force mode can be activated	
AuthPossible	boolean	True, if the JSON Interface can obtain forcing autorization	
ownForcing	boolean	True, if forcing is performed by REST API at the moment	
currentClient	string	Current forcing client identifier	

Name	Data type	Description	Example
digitalOutForced	array of numbers (2)	The force values of all 16 digital output channels.	
digitalOutMask	array of numbers (2)	The forcing mask of all 16 digital output channels.	
digitalInForced	array of numbers (2)	The force values of all 16 digital input channels.	
digitalInMask	array of numbers (2)	The forcing mask of all 16 digital input channels.	

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)		

Table 27: 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", "off", "aux"	
inlogica	string	"no","nc"	
inlogicb	string	"no","nc"	

Table 28: 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 29: Digital object

12.4 CoAP server

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

CoAP is based on the Server-Client principle and a service layer protocol that lets nodes and machines communicate with each other. The 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 96.

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 30: 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	

Table 31: REST API access via CoAP

12.4.3 CoAP configuration - Quick start guide

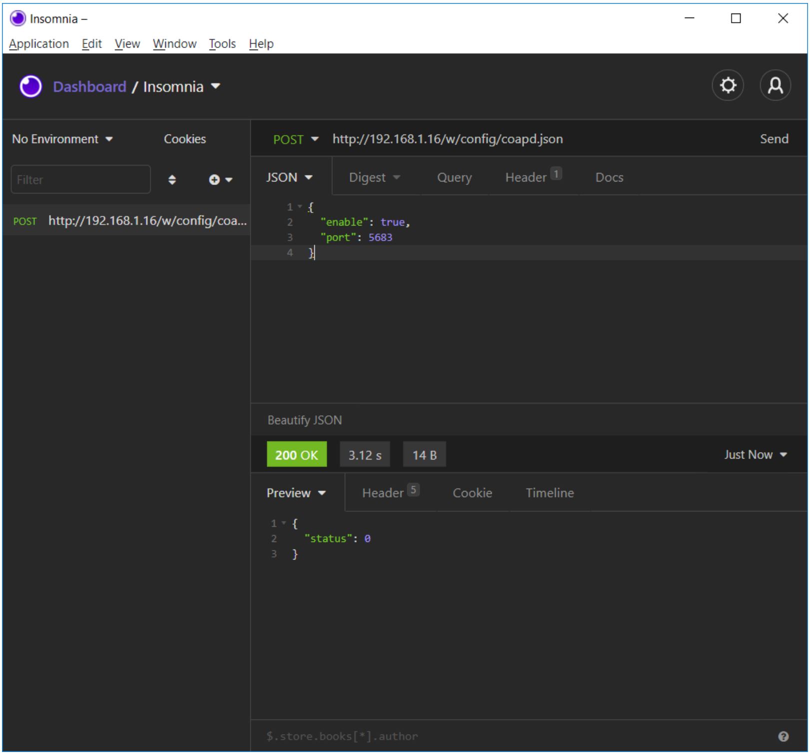
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

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/

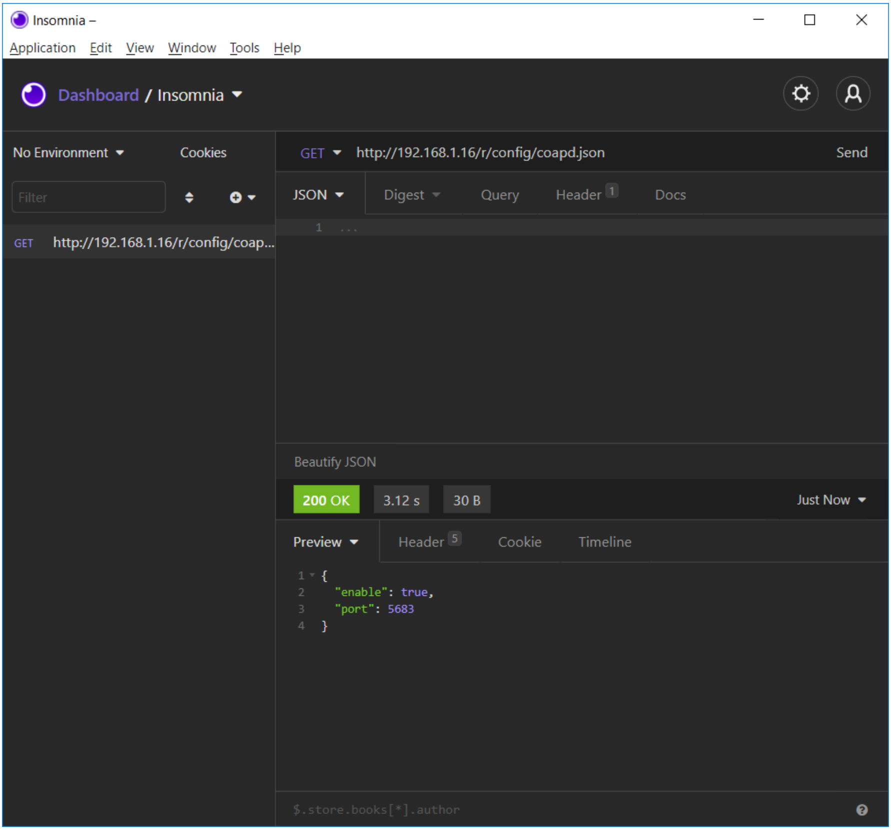
2. Configure CoAP:

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



3. Read CoAP configuration:

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



12.5 Syslog

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 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 101.

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 32: 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

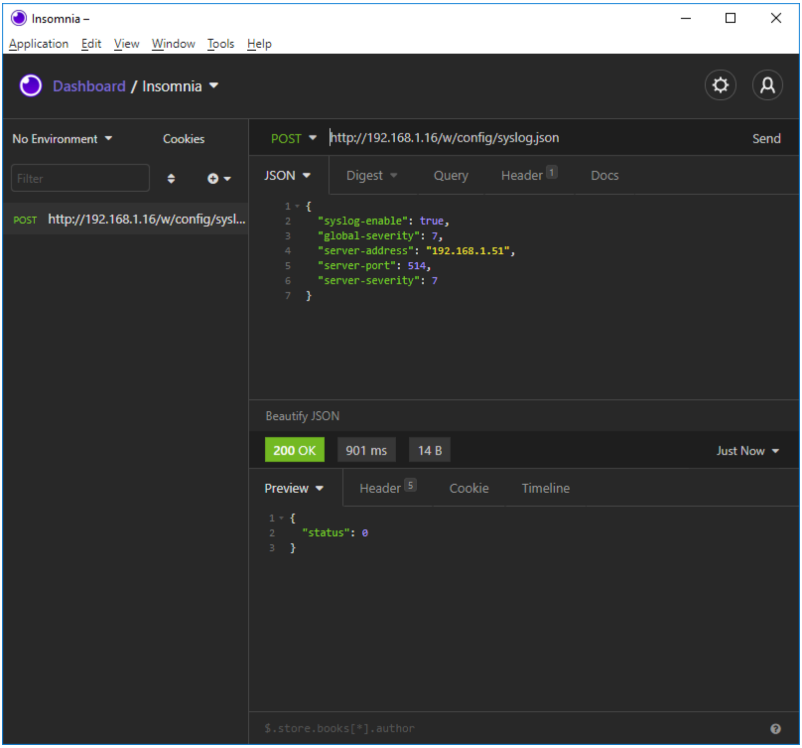
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

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/

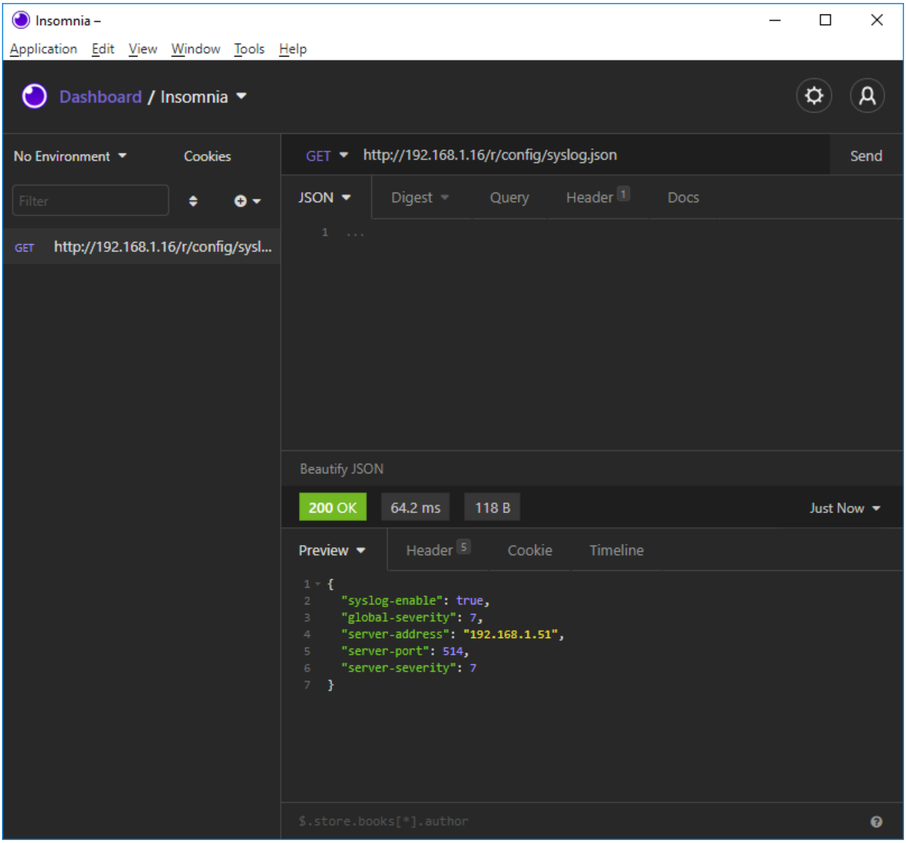
2. Configure Syslog:

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 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 105.

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 33: 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

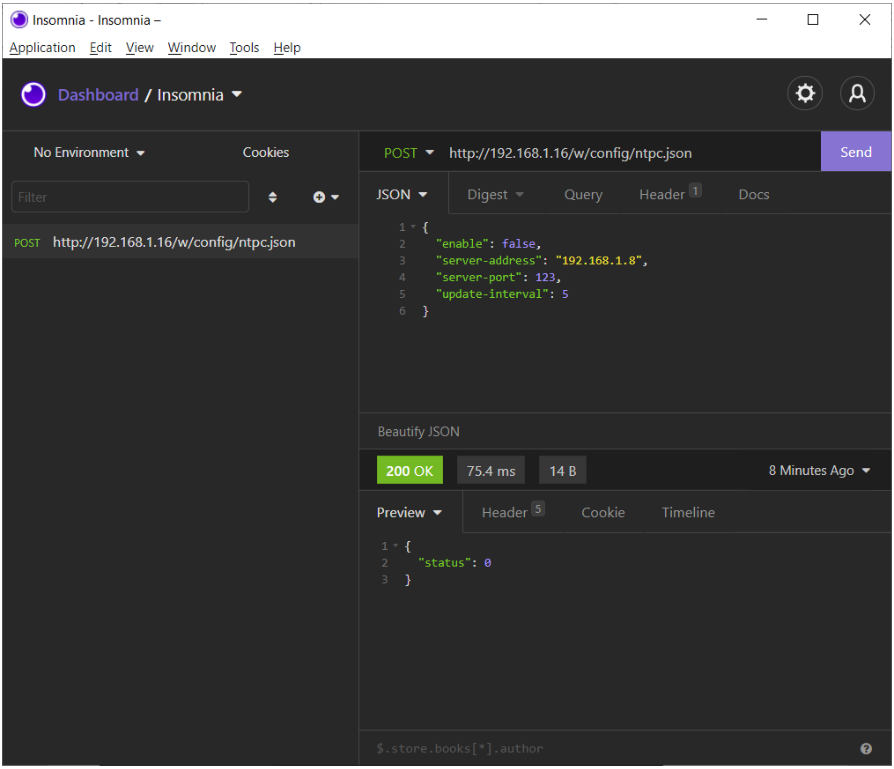
Attention: U.I. Lapp GmbH is not responsible for any content of the referenced Web pages and provides no warranty for any functionality of the named third party software.

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/

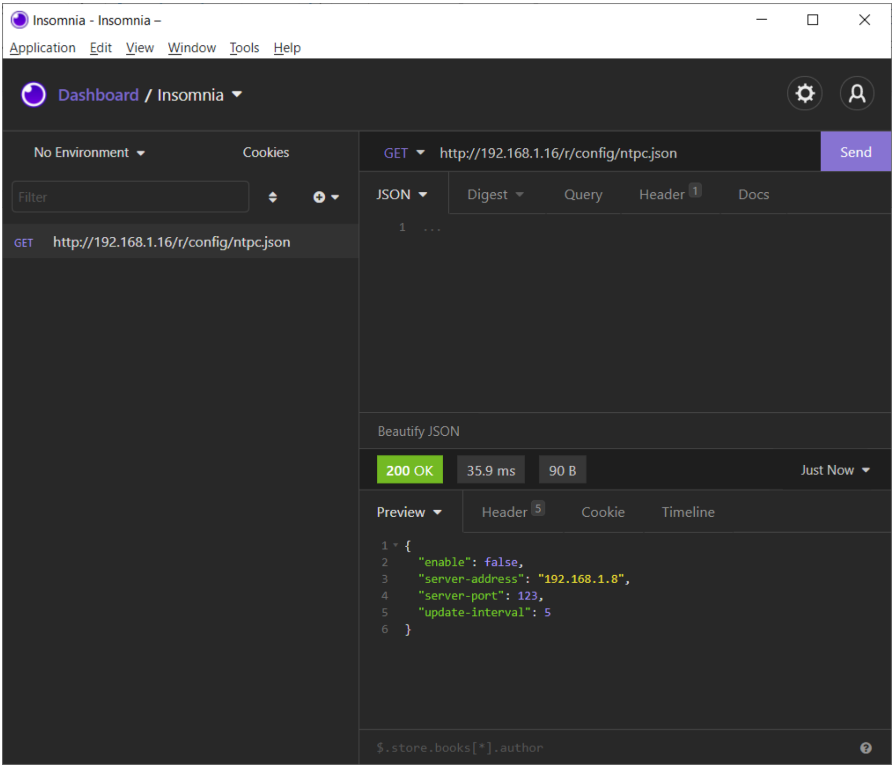
2. Configure NTP:

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



3. Read NTP configuration:

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



13 The integrated Web server

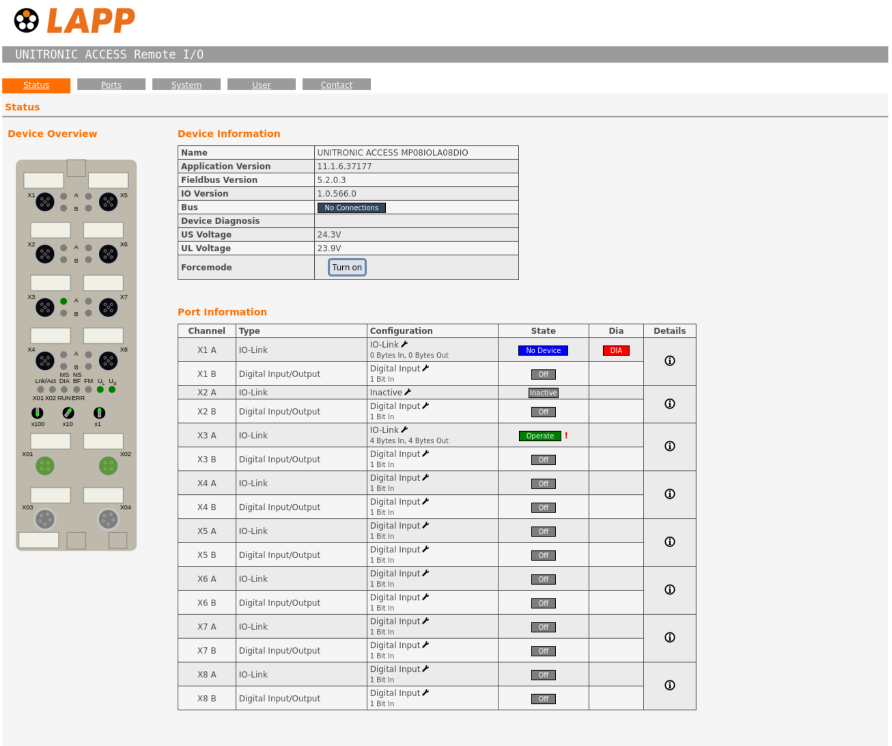
All device variants are equipped with an integrated Web server which makes functions for the device configuration and the display of status and diagnostic information available via a Web interface.

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://" 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



Name	UNITRONIC ACCESS MP08IOLA08DIO
Application Version	11.1.6.37177
Fieldbus Version	5.2.0.3
IO Version	1.0.566.0
Bus	No Connections
Device Diagnosis	
US Voltage	24.3V
UL Voltage	23.9V
Forcemode	Turn on

Channel	Type	Configuration	State	Dia	Details
X1 A	IO-Link	IO-Link 0 Bytes In, 0 Bytes Out	No Device	DIA	
X1 B	Digital Input/Output	Digital Input 1 Bit In	Off		
X2 A	IO-Link	Inactive	Inactive		
X2 B	Digital Input/Output	Digital Input 1 Bit In	Off		
X3 A	IO-Link	IO-Link 4 Bytes In, 4 Bytes Out	Operate !		
X3 B	Digital Input/Output	Digital Input 1 Bit In	Off		
X4 A	IO-Link	Digital Input 1 Bit In	Off		
X4 B	Digital Input/Output	Digital Input 1 Bit In	Off		
X5 A	IO-Link	Digital Input 1 Bit In	Off		
X5 B	Digital Input/Output	Digital Input 1 Bit In	Off		
X6 A	IO-Link	Digital Input 1 Bit In	Off		
X6 B	Digital Input/Output	Digital Input 1 Bit In	Off		
X7 A	IO-Link	Digital Input 1 Bit In	Off		
X7 B	Digital Input/Output	Digital Input 1 Bit In	Off		
X8 A	IO-Link	Digital Input 1 Bit In	Off		
X8 B	Digital Input/Output	Digital Input 1 Bit In	Off		

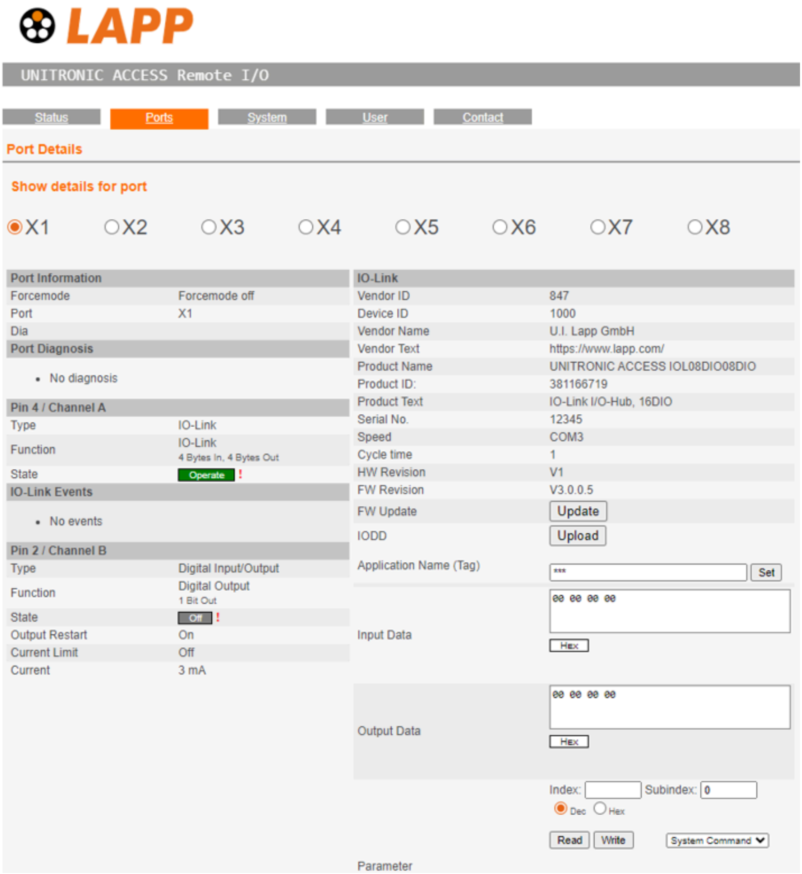
The status page provides a quick overview of the current state of the device.

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.

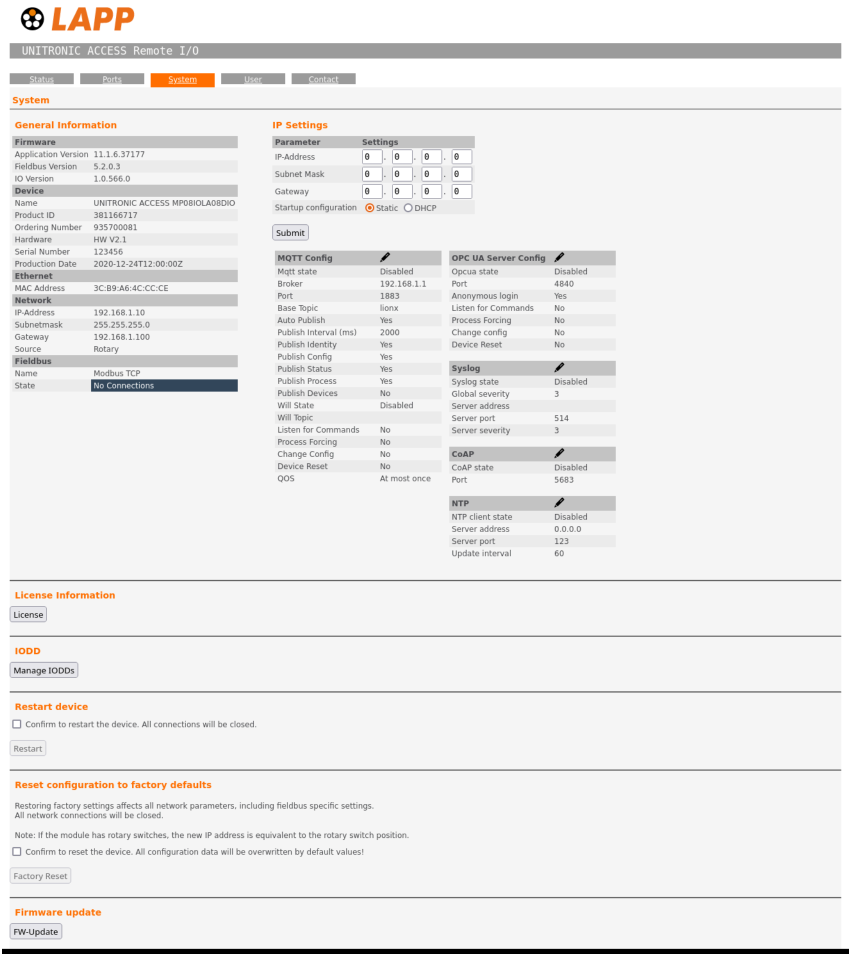
The "Port Information" table shows the configuration and state of the I/O ports.

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.

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.

Restart Device

The module initializes a software reset.

Reset to Factory Settings

The module restores to the default factory settings.

IP Settings

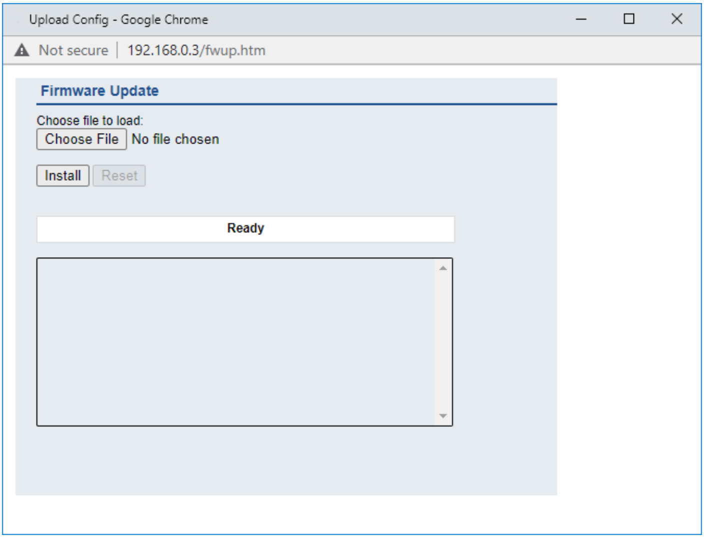
Use this parameter to change the current IP address of the module.

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.

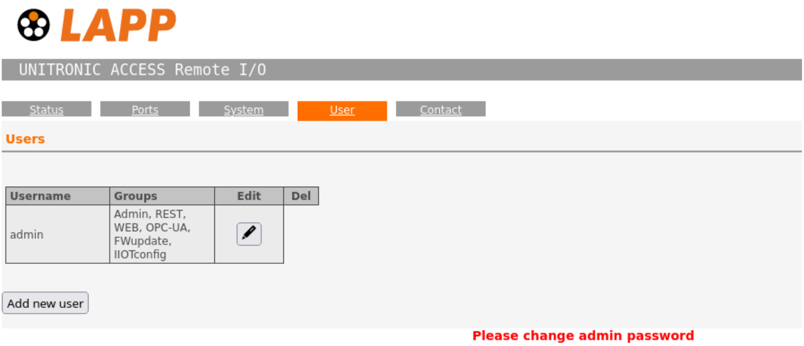
Firmware Update

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.



13.1.4 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

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.)[1]	IP65 IP67 IP69K	
Ambient temperature (during operation and storage)	MP08DIO08DIO	-40 °C .. +70 °C (-40 °F .. +158 °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 34: General information

1 Not under UL investigation.

14.2 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	12 (Communications Adapter)
Product code	41000 (MP08DIO08DIO, 381166718)
Supported Ethernet protocols	Ping ARP HTTP TCP/IP
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 35: CC-Link IE Field Basic 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 (Pin 1)	MP08DIO08DIO	Port X1 .. X8 (Pin 1)	max. 4 A per port (at $T_{ambient}$ = 30° C)
Voltage level of the sensor power supply	Min. (U_S – 1.5 V)		
Short circuit/overload protection of sensor supply	Yes, per port		
Reverse polarity protection	Yes		
Operational indicator (U_S)	LED green:	18 V (+/- 1 V) < U_S	
	LED red:	U_S < 18 V (+/- 1 V)	

Table 36: Information on the power supply of the module electronics/ sensors

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

- Max. 4.0 A per port
- Max. 5.0 A for each port pair X1/X2, X3/X4, X5/X6, X7/X8

- Max. 9.0 A in total for the whole port group X1 .. X8

  Pay attention to the derating!

14.4 Power supply of the actuators

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

Table 37: Information on the power supply of the actuators

14.5 I/O ports

MP08DIO08DIO	Ports X1 .. X8	DI, DO	M12 socket, 5-pin

Table 38: I/O ports: Overview of functions

14.5.1 Digital inputs

Input connection	MP08DIO08DIO		Type 3 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	MP08DIO08DIO	X1 .. X8	16
Status indicator	yellow LED for Channel A (Pin 4) white LED for Channel B (Pin 2)		
Diagnostic indicator	red LED per port		

Table 39: I/O ports configured as digital input

14.5.2 Digital outputs

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

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

Pay attention to the derating!

Output type	normally open, p-switching	
Nominal output voltage per channel	Signal status “1” Signal status “0”	min. (U_L -1 V) max. 2 V
Max. output current per device	MP08DIO08DIO	9 A
Max. output current per channel	MP08DIO08DIO (X1 .. X8)	2 A
Short-circuit/overload protected	yes/yes	
Behavior in case of short circuit or overload	deactivation with automatic power-on (parameterized)	
Number of digital outputs	MP08DIO08DIO (X1 .. X8)	16
Status indicator	yellow LED per output Channel A (Pin 4) white LED per output Channel B (Pin 2)	
Diagnostic indicator	red LED per channel	

Table 40: I/O ports configured as digital output

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

14.6 LEDs

LED	Color	Description
U_L	Green	Auxiliary sensor/actuator voltage OK 18 V (+/- 1 V) < U_L < 30 V (+/- 1 V)
	Red*	Auxiliary sensor/actuator voltage LOW U_L < 18 V (+/- 1 V) or U_L > 30 V (+/- 1 V) * if "Report U_L supply voltage fault" is enabled.
	OFF	None of the above conditions.
U_S	Green	System/sensor voltage OK 18 V (+/- 1 V) < U_S < 30 V (+/- 1 V)
	Red	System/sensor voltage LOW U_S < 18 V (+/- 1 V) or U_S > 30 V (+/- 1 V)
	Red flashing	Device performs a factory reset (position of rotary encoding switches: 9-7-9)
	OFF	None of the above conditions.
X1 .. X8 A	Yellow	Status of digital input or digital output on pin 4 line "on".
	Red	Short circuit on pin 4 line. / Overload or short circuit on L+ (pin 1) line / communication error
	OFF	None of the above conditions.
X1 .. X8 B	White	Status of digital input or digital output on pin 2 line "on".
	Red	Short circuit on pin 2 line. / 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 double flash	Firmware update
	OFF	None of the above conditions.

Table 41: Information on the LED colors

14.7 Data transfer times

The following tables give an overview of the internal data transfer times of UNITRONIC® ACCESS.

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

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

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

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

Use case 1:

UNITRONIC® ACCESS Digital-I/O configuration with enabled Web interface and *disabled* IIoT protocols

16DIO variant (MP08DIO08DIO):

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

Use case 2:

UNITRONIC® ACCESS Digital-I/O configuration with enabled Web interface and *enabled* IIoT protocols

16DIO variant (MP08DIO08DIO):

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

15 Accessories

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

https://www.lapp.com