GB

Operating instructions

Remote I/O module 16 DI PROFINET IP65 / IP66 / IP67

AL4002

11425600 / 00 04 / 2022

Contents

1 Preliminary note

You will find instructions, technical data, approvals and further information using the QR code on the unit / packaging or at www.ifm.com.

2 Safety instructions

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

2.1 Cyber security

ATTENTION

Operating the machine in an unprotected network environment

▷ Unauthorised read or write access to data is possible.

▷ Unauthorised manipulation of the device function is possible.

▶ Check and restrict access options to the device.

3 Intended use

The unit may only be used for the following purposes:

- Gateway between digital sensors and a higher-level control system

The device is designed for use outside of a control cabinet.

4 Function

4.1 Visual indication

The device displays the following indications:

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

4.2 Parameter setting

The device can be configured using the following options:

- parameter setting software
  - ifm moneo
  - ifm moneo|configure SA
- ifm IoT Core
  - REST-API
  - IoT-Core Visualizer
- PROFINET projection software

4.3 Inputs

The device has 8 ports. Each port has 2 digital inputs.

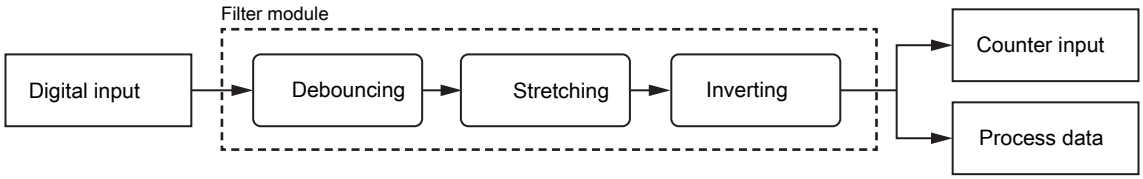
4.3.1 Sensor supply

The device has a total of 8 sensor supplies (1 sensor supply per port).

4.4 Digital input filters

The device supports preprocessing of the digital input signals. The filter result is forwarded as a process value. The following filters can be applied to the input signals in the sequence specified.

1. Debouncing
2. Stretching
3. Inverting



Each filter can be configured separately.

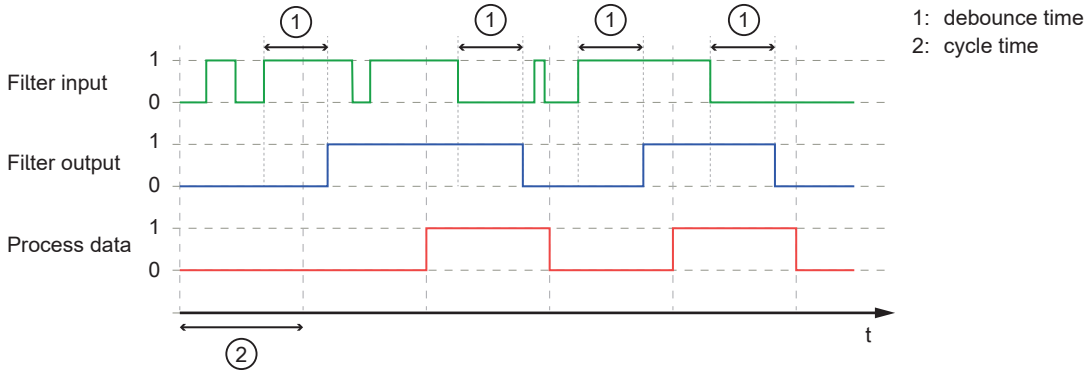
The device detects signals of a length of min. 0.23 ms. Shorter signals are not detected.

Periodic signals are only detected reliably if the signal period is at least twice as long as the cycle time.

4.4.1 Debouncing

The filter suppresses noise signals. The filter provides the input signals at the filter output with a delay (debounce time). All signals shorter than the set debounce time are ignored by the filter.

Time diagram debounce filter:



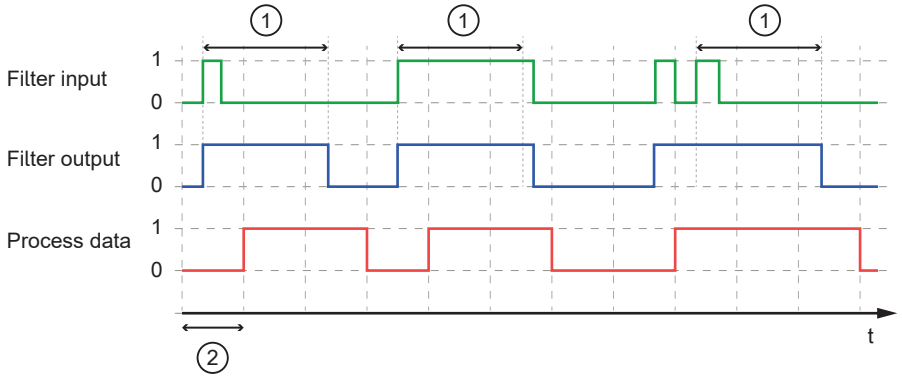
1: debounce time
2: cycle time

4.4.2 Holding

The filter prolongs short input pulses. Level changes that occur during a holding period are ignored. The filter is configured via the following parameters:

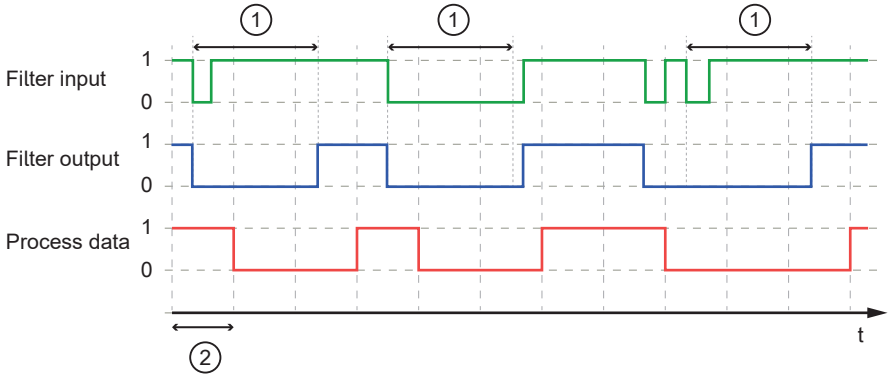
- Hold time: pulse duration to which short pulses are to be prolonged. Pulses that are present for a longer time than the hold time are not prolonged.
- Hold level: signal level to be prolonged (HIGH or LOW)

Time diagram hold filter (status HIGH):



1: hold time
2: cycle time

Time diagram hold filter (status LOW):



1: hold time
2: cycle time

4.4.3 Inverting

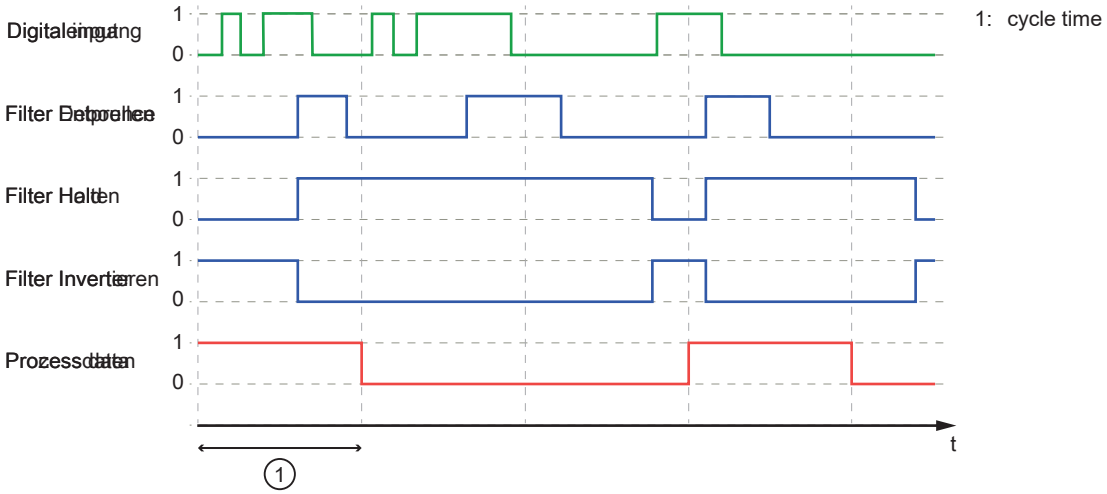
The filter inverts signals.

4.4.4 Filter combination

The filters can be combined.

Example: All 3 filters are activated

Time diagram:

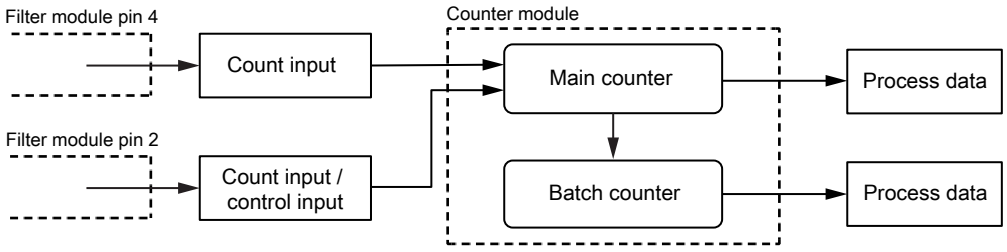


4.5 Counters

The device features one counter module per port.

A counter module consists of 2 separate counters:

- Main counter: The main counter counts the rising edges of the filtered digital input signals. The main counter has a value range that is defined by a threshold value. If the value range of the main counter is exceeded or not reached, an overflow or underflow signal is sent to the batch counter.
- Batch counter: The batch counter counts the overflow or underflow signals of the main counter.



A counter module can be operated in different operating modes. The following operating modes are available.

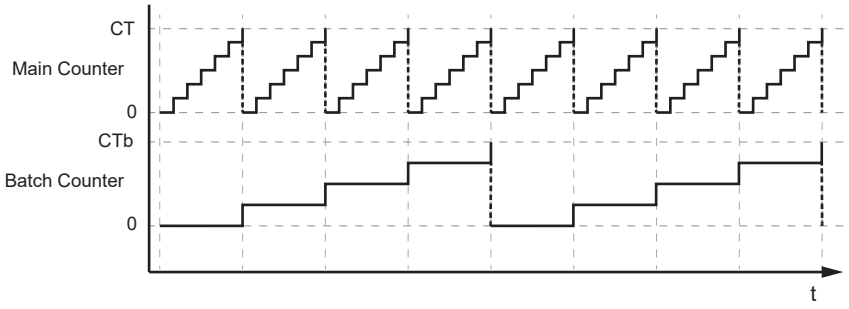
4.5.1 Counter mode CTU

In CTU (Count Up) mode, the counter module operates as an up counter with overflow detection and overflow counter.

Behaviour:

- The initial value of the main counter is m = 0. The initial value of the batch counter is b = 0. The main counter has a threshold value CT. The batch counter has a threshold value CTb.

- If the counter module detects a positive edge at pin 4 of the port, the value of the main counter is incremented (m = m+1).
- If the main counter reaches the threshold value CT (m = CT), the counter value is reset (m = 0). Due to the overflow detection, the value of the batch counter is incremented (b = b+1).
- If the batch counter reaches the threshold value CTb (b = CTb), the counter value is reset (b = 0).

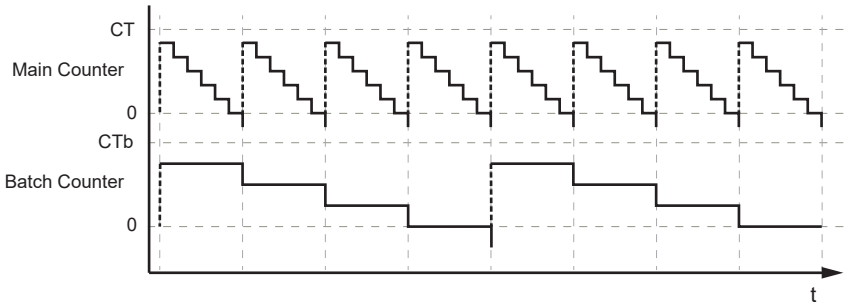


4.5.2 Counter mode CTD

In CTD (Count Down) mode, the counter module operates as a down counter with underflow detection and underflow counting.

Behaviour:

- The initial value of the main counter is m = 0. The initial value of the batch counter is b = 0. The main counter has a threshold value CT. The batch counter has a threshold value CTb.
- The first time a positive edge is detected at pin 4, the value of the main counter is set to the threshold value CT-1 (m = CT-1). At the same time, the value of the batch counter is set to the threshold value CTb-1 (b = CTb-1).
- If the counter module detects a positive edge at pin 4 of the port, the value of the main counter is decremented (m = m-1).
- If the main counter falls below 0, the counter value is reset to the threshold value (m = CT-1). Due to the underflow detection, the value of the batch counter is decremented (b = b-1).
- If the batch counter falls below 0, the counter value is reset to the threshold value (b = CTb-1).



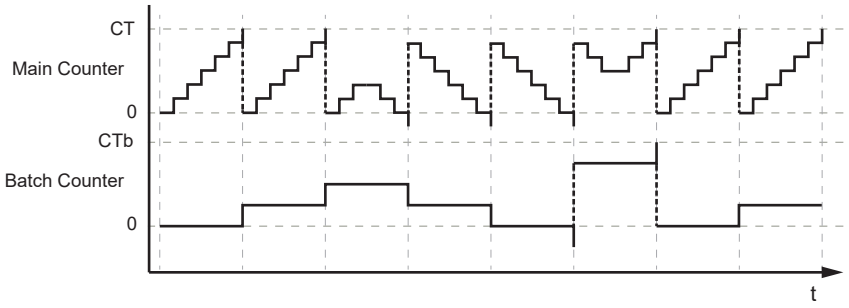
4.5.3 Counter mode CTUD

In CTUD (Count Up Down) mode, the counter operates as a simultaneous up and down counter with overflow and underflow detection.

Behaviour:

- The initial value of the main counter is m = 0. The initial value of the batch counter is b = 0. The main counter has a threshold value CT. The batch counter has a threshold value CTb.
- If the counter module detects a positive edge at pin 4 of the port, the value of the main counter is incremented (m = m+1).
- If the counter module detects a positive edge at pin 2 of the port, the value of the main counter is decremented (m = m-1).

- If the counter module simultaneously detects a positive edge at pin 4 and pin 2 of the port, the counter value of the main counter does not change.
- If the main counter reaches the threshold value CT (m = CT), the counter value is reset (m = 0). Due to the overflow detection, the value of the batch counter is incremented (b = b+1).
- If the main counter falls below 0, the counter value is reset to the threshold value (m = CT-1). Due to the underflow detection, the value of the batch counter is decremented (b = b-1).
- If the batch counter reaches the threshold value CTb (b = CTb), the counter value is reset (b = 0).
- If the batch counter falls below 0, the counter value is reset to the threshold value (b = CTb-1).

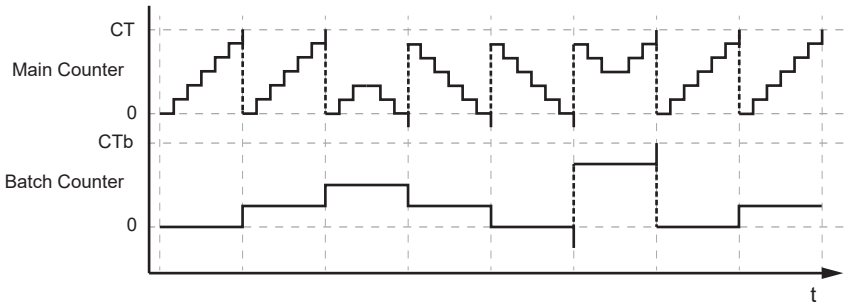


4.5.4 Counter mode CTDIR

In CTDIR (Count Direction) mode, the counter module operates either as an up counter with overflow detection or as a down counter with underflow detection. The counting direction can be set.

Behaviour:

- The initial value of the main counter is m = 0. The initial value of the batch counter is b = 0. The main counter has a threshold value CT. The batch counter has a threshold value CTb.
- The user can determine the counting direction. The counter module initially operates as an up counter with overflow detection.
- If the counter module detects a positive edge at pin 4 of the port and the counting direction of the port is set to “up”, the value of the main counter is incremented (m = m+1).
- If the main counter reaches the threshold value CT (m = CT), the counter value is reset (m = 0). Due to the overflow detection, the value of the batch counter is incremented (b = b+1).
- If the batch counter reaches the threshold value CTb (b = CTb), the counter value is reset (b = 0).
- If the counter module detects a positive edge at pin 4 of the port and the counting direction at pin 2 of the port is set to “down”, the value of the main counter is decremented (m = m-1).
- If the main counter falls below 0, the counter value is reset to the threshold value (m = CT-1). Due to the underflow detection, the value of the batch counter is decremented (b = b-1).
- If the batch counter falls below 0, the counter value is reset to the threshold value (b = CTb-1).



4.6 PROFINET

Supported PROFINET functions:

- Device profile: PROFINET IO device
- Fast Startup (FSU)
- Prioritized Startup
- Participation in network with activated IRT protocol
- System redundancy S2
- Dynamic reconfiguration
- Device description: GSD file

5 Installation

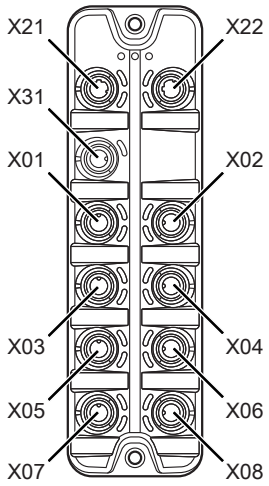
5.1 Install device

- ▶ Disconnect the power of the machine before installation.
- ▶ Use a flat mounting surface for installation.
- ▶ Please observe the maximum tightening torque.

▶ Fasten the module onto the mounting surface using M5 screws and washers (tightening torque: 1.8 Nm).

6 Electrical connection

6.1 Overview



X21: Ethernet 1 (PROFINET)
X22: Ethernet 2 (PROFINET)
X31: power IN
X01: process connection 1 (sensor)
X02: process connection 2 (sensor)
X03: process connection 3 (sensor)
X04: process connection 4 (sensor)
X05: process connection 5 (sensor)
X06: process connection 6 (sensor)
X07: process connection 7 (sensor)
X08: process connection 8 (sensor)

6.2 General wiring information

The unit must be connected by a qualified electrician.

▶ Observe the national and international regulations for the installation of electrical equipment.

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

This device contains components that may be damaged or destroyed by electrostatic discharge (ESD).

▶ Please observe the required precautions against electrostatic discharge.

The circuits are insulated from each other and from touchable surfaces of the device with basic insulation according to EN 61010-1.

The communication interfaces are insulated from each other and from touchable surfaces of the device with basic insulation according to EN 61010-1.

6.2.1 Connection technology

The threaded connections in the device correspond to the M12 standard. To ensure compliance with the specified protection rating, only cables that comply with this standard may be used. In the case of self-assembled cables, the system manufacturer is responsible for the protection rating.

▶ Use connectors with gold-plated contacts.

▶ During installation, place the connectors vertically so that the coupling nut will not damage the thread.

▶ Observe the coding of the connectors during installation.

▶ Cover unused connections with protective covers. Tightening torque: 0.3 ± 0.1 Nm

6.3 Ethernet

The device is connected to the PROFINET network via the Ethernet ports X21 / X22 (e. g. PROFINET control, additional PROFINET device). In addition, the device can be connected to an IT network via the Ethernet ports. Via the IT network, the user can access functions of the ifm IoT Core (configuration tools, REST API, IoT Core Visualizer).

▶ Connect the device to the PROFINET network via a free Ethernet port.

▶ Optional: Connect the device to the IT network via a free Ethernet port.

▶ For connection, use an M12 connector (with at least protection rating: IP65 / IP66 / IP67).

▶ Tighten the cable plug using 1.3 ± 0.1 Nm.

Wiring:

1: TX +
2: RX +
3: TX -
4: RX -
5: not used

6.4 Process connections

The sensors are connected to the device via the process connections.

The total current supply of the ports X01...X08 is limited to 3.6 A.

The ports feature short-circuit / overload detection.

▶ Connect the sensors to ports X01...X08.

▶ For connection, use M12 connectors (with at least protection rating: IP65 / IP66 / IP67; max. cable length: 30 m).

▶ Tighten the cable plug using 1.3 ± 0.1 Nm.

Wiring:

1: sensor supply L+ (US)
2: digital output 2 (DI2)
3: sensor supply L- (US)
4: digital output 1 (DI1)
5: functional earth (FE)

6.5 Voltage supply

The device is connected to the supply voltage US via the power IN port.

The US supply voltage supplies the device and the sensors connected to the ports X01...X08 with voltage.

Port X31 has overvoltage protection (US).

Port X31 has reverse polarity protection (US).

Port X31 has an inrush current limitation.

▶ Disconnect power!

▶ Connect the device via port X31 to 24 V DC (20…30 V SELV/PELV).

▶ For connection, use an L-coded M12 connector (with at least protection rating: IP65 / IP66 / IP67).

▶ Tighten the cable sockets according to the torque specifications indicated by the cable manufacturer. Maximum permissible tightening torque: 0.8 Nm

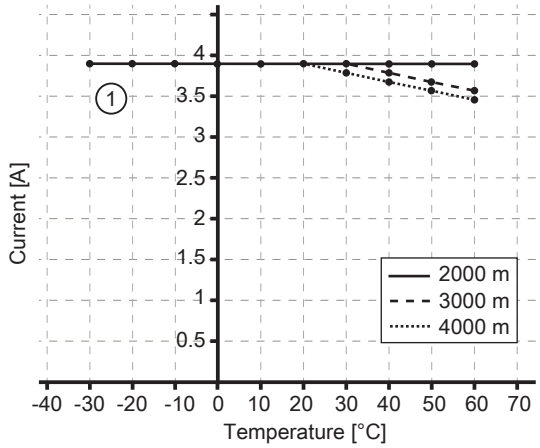
▶ Observe the derating behaviour of the device (→ Derating behaviour 🗋 17).

Wiring:

1: + 24 V DC (US)
2: not used
3: GND (US)
4: not used

6.5.1 Derating behaviour

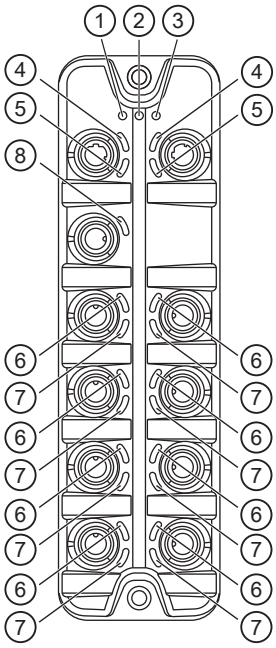
The current IUS available at ports X01...X08 depends on the ambient temperature of the device.



1: I_{US} at ports X01...X08

7 Operating and display elements

7.1 LEDs



1: status: RDY
2: status: BF
3: status: SF
4: PROFINET: LNK
5: PROFINET: ACT
6: process connection: DI1
7: Process connection: DI2
8: voltage supply: US

7.1.1 Status

LED	Description	Colour	State	Description
RDY	Gateway status	-	Off	Not active or reboots
		Green	Flashes 3 s (1 Hz)	DCP signalling service initiated via fieldbus
			Flashes (5 Hz)	Error
			Flashes (200 ms on, 800 ms off)	Firmware update running
			On	OK
		Red	On	Error during firmware update (e.g. firmware not compatible)
BF	PROFINET connection status (bus failure)	Red	Off	No error
			Flashes (2 Hz)	No data transmission
			On	• No configuration or • physical connection with low speed or • no physical connection
SF	System status (system failure)	Red	Off	No error
			Flashes 3 s (1 Hz)	DCP signalling service initiated via fieldbus
			On	• Watchdog expired or • channel diagnosis, general or extended diagnosis available or • system error

7.1.2 Ethernet

LED	Description	Colour	State	Description
LNK	Status of the connection	Green	Off	no Ethernet connection

LED	Description	Colour	State	Description
LNK	Status of the connection	Green	On	Ethernet connection established
ACT	Status of the data transmission	Yellow	Off	no data transmission
			Flashes	Data transmission

7.1.3 Voltage supply

LED	Description	Colour	State	Description
US	Voltage supply status	-	Off	No supply voltage is applied or the applied supply voltage is too low
		Green	On	Supply voltage applied
		Red	On	Overvoltage, undervoltage, short circuit at sensor supply

7.1.4 Process connections

LED	Description	Colour	State	Description
DI1	Digital input signal level (pin 4)	Yellow	Off	Digital input - pin 4: LOW
			On	Digital input - pin 4: HIGH
DI2	Digital input signal level (pin 2)	Yellow	Off	Digital input - pin 2: LOW
			On	Digital input - pin 2: HIGH

8 Set-up

▶ Install the unit correctly.

▶ Establish a correct electrical connection with the device.

▷ Once connected to the supply voltage, the unit will start.

▷ The LEDs show status and error conditions.

▷ The unit is ready for operation.

▷ The device can be configured.

9 Settings

9.1 Parameter setting software

9.1.1 Supported parameter setting software

The device can be configured with the following parameter setting software:

- ifm moneo
- ifm moneo|configure SA

▶ Install the desired parameter setting software.

▶ Activate the licences required for operation.

▷ Parameter setting software can be used for parameter setting of the device.

The configuration created with ifm moneo is overwritten when a connection is established between the device and the PROFINET PLC.

9.1.2 Getting started

Requirements:

✓ The parameter setting software is correctly installed on the laptop / PC.

✓ The laptop / PC is connected to a free Ethernet port of the device.

▶ Start the parameter setting software.

▶ Scan the network for devices.

▷ The parameter setting software recognises the device.

▶ Optional: Configure the PROFINET interface (→ 🗋 21)

▶ Establish a connection to the device.

▷ The parameter setting software can access the device parameters.

9.1.2.1 Configure the PROFINET interface

The PROFINET interface of the device can only be set via DCP-capable parameter setting software, e.g. PROFINET projection software.

With the parameter setting software ifm moneo or ifm moneo|configure (SA), the IP parameters of the PROFINET interface can only be set during the network scan. In the editor view of the device, the configuration of the PROFINET interface is read-only.

▶ Start the parameter setting software.

▶ Configure the PROFINET interface of the device.

▷ The device has a configured PROFINET interface.

9.1.3 Fieldbus: Read the interface configuration

▶ Observe the notes on the configuration of the PROFINET interface: Configure the PROFINET interface (→ 🗋 21)

Available parameters:

Name	Description	Value range	Access
[ipaddress]	IP address of the PROFINET interface	e.g. 192.100.0.10 0.0.0.0 (default)	ro [1]
[subnetmask]	Subnet mask of the network segment	e.g. 255.255.255.0 0.0.0.0 (default)	ro [1]
[ipdefaultgateway]	IP address of the network gateway	e.g. 192.100.0.1 0.0.0.0 (default)	ro [1]
[macaddress]	MAC address of the Ethernet interface	e.g. 00:02:01:0E:10:7F	ro [1]
[hostname]	Name of the device in the PROFINET network	e.g. al4x0x	ro [1]

[1] read only

Requirements:

✓ The parameter setting software has been started.

✓ The detailed view of the device is active.

▶ Select the [fieldbussetup] > [network] menu.

▷ The menu page displays the current configuration of the PROFINET interface.

9.1.4 Fieldbus: Read the connection status

Available information:

Name	Description	Value range	Access
[connectionstatus]	Status of the PROFINET connection	• Disconnected: not connected • Connected: connected	ro [1]
[fieldbusfirmware]	Firmware version of the PROFINET stack	e.g. 5.4.0.3 (PROFINET IO Device)	ro [1]

[1] read only

Requirements:

✓ The parameter setting software has been started.

✓ The detailed view of the device is active.

▶ Select the [fieldbussetup] menu.

▷ The menu page displays the status of the PROFINET connection.

9.1.5 Ports: Configure input filters

▶ Observe the notes on input filters: Digital input filters (→ 🗋 8)

Available parameters:

Name	Description	Value range	Access
[pin2]/[debounce_time]	Pin 2: debounce time (= value * 0.1 ms)	• 0: 0 ms (default) … • 500: 50 ms	rw [1]

Name	Description	Value range	Access
[pin2]/[hold_time]	Pin 2: hold time (= value * 0.1 ms)	• 0: 0 ms (default) … • 60000: 6000 ms	rw [1]
[pin2]/[hold_level]	Pin 2: hold level	• 0: hold LOW • 1: hold HIGH (default)	rw [1]
[pin2]/[invert]	Pin 2: inversion	• 0: do not invert (default) • 1: invert	rw [1]
[pin4]/[debounce_time]	Pin 4: debounce time (= value * 0.1 ms)	• 0: 0 ms … • 500: 50 ms	rw [1]
[pin4]/[hold_time]	Pin 4: hold time (= value * 0.1 ms)	• 0: 0 ms (default) … • 60000: 6000 ms	rw [1]
[pin4]/[hold_level]	Pin 4: hold level	• 0: hold LOW • 1: hold HIGH (default)	rw [1]
[pin4]/[invert]	Pin 4: inversion	• 0: do not invert (default) • 1: invert	rw [1]

[1] read and write; can only be changed if no connection to the fieldbus controller is active

▶ Select the menu option [io] > [port[n]] (n: 1…8).

▷ The menu page displays the available parameters.

▶ Set the parameters.

▶ Write the changed parameter values to the device.

▷ The digital input filters have been configured.

9.1.6 Ports: Read digital input data

Available information:

Name	Description	Value range	Access
[pin2]/[digital]	Process value digital input - pin 2 (after filtering)	• LOW: off • HIGH: on	ro [1]
[pin4]/[digital]	Process value digital input - pin 4 (after filtering)	• LOW: Off • HIGH: on	ro [1]

[1] read only

Requirements:

✓ The parameter setting software has been started.

✓ The detailed view of the device is active.

▶ Select the menu option [io] > [port[n]] (n: 1…8).

▷ The menu page displays the current process values of the port's digital inputs.

The displayed process values are the filtered input data.

9.1.7 Counters: Configure counter modules

▶ Observe the notes on counter modules: Counters (→ 🗋 10)

If the operating mode of a counter module is changed, the current counter values will be reset and any active events will be deleted.

For the parameters [pin2_function] and [count_direction_selection] all shown parameter values can be selected. It is not checked whether these make sense. For each counter operating mode (parameter [mode]), the table below indicates the valid value ranges (✓: valid setting; ✕: invalid setting):

[mode]	[pin2_function]					[count_direction_selection]	
	N/C	Counter Edge Input Pin2	Count Direction	Reset (Main & Batch Counter)	Disable (Main & Batch Counter)	Pin 2 Count Direction	IoT / PLC Count Direction
CTU	✓	✕	✕	✓	✓	✕	✕
CTD	✓	✕	✕	✓	✓	✕	✕
CTUD	✕	✓	✕	✕	✕	✕	✕
CTDIR	✕	✕	✓	✕	✕	✓	✕
CTDIR	✓	✕	✕	✓	✓	✕	✓

Available parameters:

Name	Description	Value range	Access
[mode]	Operating mode of the counter module	• CTU (up counter): up counter (default) • CTD (down counter): down counter • CTUD (up counter / down counter): up and down counter • CTDIR (direction counter): up or down counter	rw [1]
[pin2_function]	Pin 2 function of the port (→ Observe note!)	• N/C: no function (default) • Counter Edge Input 2: counting pulse (rising edge) • Count Direction: counting direction • Reset (Main & Batch Counter): reset main counter and batch counter • Disable (Main & Batch Counter): disable main counter and batch counter	rw [1]
[count_direction_selection]	Control instance for selecting the counting direction (→ Observe note!)	• Pin 2 Count Direction: pin 2 of the port (default) • IoT / PLC Count Direction: Fieldbus PLC	rw [1]
[main_threshold]	Main counter threshold (CT)	• 1 … • 4294967295 (default)	rw [1]
[batch_threshold]	Batch counter threshold (CTb)	• 1 … • 65535 (default)	rw [1]

[1] read and write; can only be changed if no connection to the fieldbus controller is active

Requirements:

✓ The parameter setting software has been started.

✓ The detailed view of the device is active.

▶ Select the menu option [io] > [counter[n]] (n: 1…8).

▷ The menu page displays the configuration options of the counter.

▶ Configure the counter module.

▶ Optional: Configure additional counter modules.

▶ Write the changed values to the device.

▷ The counter modules are configured.

9.1.8 Counters: Read counter values

Available parameters:

Name	Description	Value range	Access
[maincounter_value]	Main counter value	0…4294967294	ro [1]
[batchcounter_value]	Batch counter value	0…65534	ro [1]

[1] read only

Requirements:

✓ The parameter setting software has been started.

✓ The detailed view of the device is active.

▶ Select the menu option [io] > [port[n]] (n: 1…8).

▷ The menu page displays the current counter values of the main and batch counter.

9.1.9 Counters: Control counter modules

Available parameters:

Name	Description	Value range	Access
[disable]	Disable main counter and batch counter	• 0: counter module is active (default) • 1: counter module is not active	rw [1]
[reset]	Reset main counter, batch counter and CT and CTb thresholds to initial values	• 0: no action (default) • 1: reset	rw [1]
[direction] [2]	Set counting direction for main and batch counter	• 0: up (default) • 1: down	rw [1]

[1] read and write; can only be changed if no connection to the fieldbus controller is active

[2] only effective if operating mode of counter module = `CTDIR`

Requirements:

✓ The parameter setting software has been started.

✓ The detailed view of the device is active.

▶ Select the menu option [io] > [counter[n]] (n: 1…8).

▷ The menu page displays the available parameters.

▶ Optional: disable counter module.

▶ Optional: reset counter module.

▶ Optional: set counting direction of counter module.

▶ Write the changed parameter values to the device.

▷ Selected actions are executed.

9.1.10 Gateway: Read identification information

Available information:

Name	Description	Value range	Access
[productcode]	Article number	AL4002	ro [1]
[devicefamily]	Device family	Ethernet modules	ro [1]
[vendor]	Manufacturer	ifm electronic gmbh	ro [1]
[swrevision]	Firmware revision	e.g. AL4x0x_fw_pn_v1.4.0.137	ro [1]
[hwrevision]	Hardware revision (status)	e.g. AA	ro [1]

Name	Description	Value range	Access
[bootloaderrevision]	Bootloader version	e.g. AL4xxx_bl_v1.2.0.35	ro [1]
[serialnumber]	Serial number	e.g. 0002043100003	ro [1]
[fieldbustype]	Fieldbus	PROFINET	ro [1]

[1] read only

Requirements:

✓ The parameter setting software has been started.

✓ The detailed view of the device is active.

▶ Select the menu option [deviceinfo].

▷ The menu page displays the identification information of the device.

9.1.11 Gateway: Read status and diagnostic information

Available information:

Parameter	Description	Value range	Access
[temperature]	Temperature of the device (value in °C)	-30…80	ro [1]
[voltage_us]	Present voltage value of the device supply US (value in mV)	0…40000	ro [1]
[supervisionstatus_us]	Status of the device supply US	• 0: no error • 1: error	ro [1]
[current_us]	Present current value of the device supply US (value in mA)	0…40000	ro [1]

[1] read only

Requirements:

✓ The parameter setting software has been started.

✓ The detailed view of the device is active.

▶ Select the menu option [Processdatamaster].

▷ The menu page displays the diagnostic and status information.

9.1.12 Gateway: Set the application tag

Available parameters:

Parameter	Description	Value range	Access
[applicationtag]	Application-specific identifier of the device in moneo	e.g. plant 1 machine 3	rw [1]

[1] read and write

Requirements:

✓ The parameter setting software has been started.

✓ The detailed view of the device is active.

▶ Select the menu option [devicetag].

▶ Enter the application identifier.

▶ Write the changed values to the device.

▷ The device can be identified by the selected application tag.

9.1.13 Firmware: Read firmware version

Available information:

Parameter	Description	Value range	Access
[version]	Firmware version	e.g. AL4x0x_fw_pn_1.4.0.137	ro [1]

[1] read only

Requirements:

✓ The parameter setting software has been started.

✓ The detailed view of the device is active.

▶ Select the [Firmware] menu.

▷ The menu page displays the firmware version of the device.

9.1.14 Firmware: Reset the device

Requirements:

✓ The parameter setting software has been started.

✓ The detailed view of the device is active.

▶ Select the [Firmware] menu.

▶ Click on [factoryreset].

▷ The device will be reset to the factory settings.

▷ All parameters are set to their default values.

9.1.15 Firmware: Restart the device

Requirements:

✓ The parameter setting software has been started.

✓ The detailed view of the device is active.

▶ Select the [Firmware] menu.

▶ Click on [Reboot].

▷ The device will be restarted.

▷ All set parameter values will be retained.

9.2 ifm IoT Core

9.2.1 ifm IoT Core: General information

The device has the ifm IoT Core. The ifm IoT Core represents the functionality of a device. Each device is represented by a number of objects, services and events. The elements of the ifm IoT Core are arranged in a JSON object in a hierarchical tree structure. The ifm IoT Core makes these elements available to the outside world via standard interfaces. This allows the user and other devices to access data (parameters, process data, events) and functions (services) of the ifm IoT Core.

9.2.1.1 Accessing the ifm IoT Core

An element of the ifm IoT Core is accessed via its address (e.g. `root/port1/pin2`). The address is composed of the path leading to the element (`root/port1`) and the identifier of the element (`pin2`).

The user can access the ifm IoT Core via HTTP requests. The following methods are supported:

GET method

Access: reading

Syntax of the request:

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

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

Syntax of the response:

```
{
  "cid":id,
  "data":{"value":"resp_data"},
  "adr":"data_point/service",
  "code":diag_code
}
```

Field	Parameter	Description
cid	id	Correlation ID for the assignment of request and reply
data	resp_data	Value of the data point; depending on the data type of the data point
adr	data_point	Data point accessed
	service	Service that accessed the data point
code	diag_code	Diagnostic code Diagnostic codes

Example: GET request

- Request:

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

- Response:

```
{
  "cid":-1,
  "data":{"value":"factory 2 plant 1"},
  "adr":"devicetag/applicationtag/getdata",
  "code":200
}
```

POST method

Access: reading, writing

Syntax of the request:

```
{
  "code":"code_id",
  "cid":id,
  "adr":"data_point/service",
  "data":{req_data},
}
```

Field	Parameter	Description
code	code_id	Service class • request: Request • transaction: Transaction • event: Event
cid	id	Correlation ID for the assignment in pairs of request and return; identifier freely selectable by the user
adr	data_point	Data point which is to be accessed
	service	Service to access the data point
data [1]	req_data	Data sent to the ifm IoT Core (e.g. new values); syntax depending on the service

[1] optional; only required for services that send data to the ifm IoT Core (e.g. setdata)

Syntax of the response:

```
{
  "cid":id,
  "data":{resp_data},
  "adr":"data_point/service",
  "code":diag_code
}
```

Field	Parameter	Description
cid	id	Correlation ID for the assignment of request and return (→ Request)
data [1]	resp_data	Values returned by the ifm IoT Core; syntax depending on the service
adr	data_point	Data point accessed
	service	Service that accessed the data point
code	diag_code	Diagnostic code

[1] optional; only available for services that receive data from the ifm IoT Core (e.g. getdata)

Example: POST request

- Request:

```
{
  "code":"request",
  "cid":-1,
  "adr":"devicetag/applicationtag/getdata"
}
```

- Response:

```
{
  "cid":-1,
  "data":{"value":"Do not use"},
  "adr":"devicetag/applicationtag/getdata",
  "code":200
}
```

9.2.2 Getting started

To register the device description:

▶ Send the following POST request to the ifm IoT Core:
`{"code":"request", "cid":-1, "adr":"gettree"}`

▷ ifm IoT Core returns the device description as a structured JSON object.

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

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

9.2.2.1 Notes on configuration

The configuration created via the IoT Core (API, IoT Core Visualizer) is overwritten when a connection is established between the device and the fieldbus PLC.

9.2.3 General functions

The device has the type “device” (→ Types 🗋 48). The following services can be applied to the root element of the device tree:

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

The following services can be applied to elements of the type `data` depending on its access rights:

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

9.2.3.1 Example: Outputting the subtree

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

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

- Request:

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

- Response:

```
{
  "cid":4711,
  "data":{
    "identifier":"firmware",
    "type":"structure",
    "profiles":["software","software/uploadablesoftware","devicereset"],
    "subs":[
      {"identifier":"version","type":"data","profiles":
["parameter"],"profiles":["parameter"],"format":
{"type":"string","namespace":"json","encoding":"UTF-8"}},
      {"identifier":"type","type":"data","profiles":["parameter"],"format":
{"type":"string","namespace":"json","encoding":"UTF-8"}},
      {"identifier":"factoryreset","type":"service"},
      {"identifier":"install","type":"service"},
      {"identifier":"signal","type":"service"},
      {"identifier":"container","type":"data","profiles":["blob"],"format":
{"type":"binary","namespace":"json","encoding":"base64"}},
      {"identifier":"reboot","type":"service"}
      ]
    },
  "adr":"gettree",
  "code":200
}
```

9.2.3.2 Example: Reading several elements sequentially

Task: The following current values of the device are to be read consecutively: Temperature, serial number

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

- Request:

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

- Response:

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

9.2.3.3 Example: Changing a parameter value

Task: The Application Tag parameter of the device is to be written with the value “Do not use”. The new value is only supposed to be valid until the next reboot of the device.

Solution: Write the new value of the `/devicetage/applicationtag` element with the `setdata` service. To keep the new value only until the next restart of the device, pass on the `duration` option with the `uptime` value.

- Request:

```
{
  "code":"request",
  "cid":4711,
  "adr":"/devicetag/applicationtag/setdata",
  "data":{"duration":"uptime","newvalue":"Do not use"}
}
```

- Response:

```
{
  "cid":4711,
  "adr":"/devicetag/applicationtag/setdata",
  "code":200,
}
```

9.2.4 Fieldbus: Read the interface configuration

Substructure: `fieldbussetup`

Available data points:

Name	Description	Values	Access
../network/ipaddress	IP address of the Ethernet interface:	e.g. 192.200.0.100 • 0.0.0.0 (default)	ro [1]
../network/subnetmask	Subnet mask of the network segment	e.g. 255.255.192.0 • 0.0.0.0 (default)	ro [1]
../network/ipdefaultgateway	IP address of the network gateway	e.g. 192.200.63.1 • 0.0.0.0 (default)	ro [1]
../network/hostname	Name of device in PROFINET project	e.g. al4x0x	ro [1]
../network/macaddress	MAC address of the Ethernet interface	e.g. 00:02:01:0E:10:7C	ro [1]
../fieldbusfirmware	Version of the PROFINET firmware of the device	e.g. 5.4.0.3 (PROFINET IO Device)	ro [1]
../connectionstatus	Status of the connection to the PROFINET network	• 0: not connected • 1: connected	ro [1]

[1] read only

9.2.5 Ports: Configure input filters

▶ Observe the notes on input filters: Digital input filters (→ 🗋 8)

Substructure: `io/port[n]` (n: 1…8)

Available data points:

Name	Description	Values	Access
../pin2/debounce_time	Pin 2: debounce time (= value * 0.1 ms)	• 0: 0 ms (default) … • 500: 50 ms	rw [1]
../pin2/hold_time	Pin 2: hold time (= value * 0.1 ms)	• 0: 0 ms (default) … • 60000: 6000 ms	rw [1]
../pin2/hold_level	Pin 2: hold level	• 0: hold LOW • 1: hold HIGH (default)	rw [1]
../pin2/invert	Pin 2: inversion	• 0: do not invert (default) • 1: invert	rw [1]
../pin4/debounce_time	Pin 4: debounce time (= value * 0.1 ms)	• 0: 0 ms (default) … • 500: 50 ms	rw [1]
../pin4/hold_time	Pin 4: hold time (= value * 0.1 ms)	• 0: 0 ms (default) … • 60000: 6000 ms	rw [1]
../pin4/hold_level	Pin 4: hold level	• 0: hold LOW • 1: hold HIGH (default)	rw [1]
../pin4/invert	Pin 4: inversion	• 0: do not invert (default) • 1: invert	rw [1]

[1] read and write; can only be changed if no connection to the fieldbus controller is active

9.2.6 Ports: Read digital input data

Substructure: `io/port[n]` (n: 1…8)

Available data points:

Name	Description	Values	Access
../pin2/digital	Process value digital input - pin 2 (after filtering)	• 0: LOW • 1: HIGH	ro [1]
../pin4/digital	Process value digital input - pin 4 (after filtering)	• 0: LOW • 1: HIGH	ro [1]

[1] read only

9.2.7 Counters: Configure counter modules

▶ Observe the notes on counter modules: Counters (→ 🗋 10)

If the operating mode of a counter module is changed, the current counter values will be reset and any active events will be deleted.

For the parameters [pin2_function] and [count_direction_selection] all shown parameter values can be selected. It is not checked whether these make sense. For each counter operating mode (parameter [mode]), the table below indicates the valid value ranges (✓: valid setting; ✕: invalid setting):

[mode]	[pin2_function]					[count_direction_selection]	
	No function	Counting pulse	Counting direction	Reset main and batch counter	Disable main and batch counter	Pin 2	Fieldbus PLC
CTU	✓	×	×	✓	✓	×	×
CTD	✓	×	×	✓	✓	×	×
CTUD	×	✓	×	×	×	×	×
CTDIR	×	×	✓	×	×	✓	×
CTDIR	✓	×	×	✓	✓	×	✓

Substructure: `io/counter[n]` (n: 1…8)

Available data points:

Name	Description	Values	Access
../mode	Operating mode of the counter module	• 0: CTU – up counter (default) • 1: CTD – down counter • 2: CTUD – up and down counter • 3: CTDIR – up and down counter with selectable counting direction	rw [1]
../pin2_function	Pin 2 function of the port (→ Observe note!)	• 0: no function (default) • 1: counting pulse (rising edge) • 2: counting direction • 3: reset main counter and batch counter • 4: disable main counter and batch counter	rw [1]
../count_direction_selection	Control instance for selecting the counting direction (→ Observe note!)	• 0: Pin 2 (default) • 1: Fieldbus PLC	rw [1]
../main_threshold	Main counter threshold (CT)	• 1 … • 4294967295 (default)	rw [1]
../batch_threshold	Batch counter threshold (CTb)	• 1 … • 65535 (default)	rw [1]

[1] read and write; can only be changed if no connection to the fieldbus controller is active

9.2.8 Counters: Control counters

Substructure: `io/counter[n]` (n: 1…8)

Available data points:

Name	Description	Values	Access
../reset	Reset counter module (reset counter and threshold values to default values)	• 0: no action (default) • 1: reset	rw [1]
../disable	Disable counter module	• 0: enable counter (default) • 1: disable counter	rw [1]
../direction [2]	Set counting direction for main and batch counter	• 0: up (default) • 1: down	rw [1]

[1] read and write; can only be changed if no connection to the fieldbus controller is active

[2] only effective if operating mode of counter module = `CTDIR`

9.2.9 Counters: Read and write counter values

Substructure: `io/counter[n]` (n: 1…8)

Available data points:

Name	Description	Values	Access
../maincounter_value	Current main counter value	0…4294967295	ro [1]
../batchcounter_value	Current batch counter value	0…65535	ro [1]

[1] read only

Applicable services:

Name	Description
../force_counter_value	Write counter values of main and batch counter

9.2.9.1 Example: Write counter values

Task: The counter values of the counter module of port 2 are to be changed (main counter = 100, batch counter = 10).

Solution: Write the new values to the structure `io/counter[2]` with the service `force_counter_value`.

- Request:

```
{
  "code":"request",
  "cid":4711,
  "adr":"io/counter[2]/force_counter_value",
  "data": {"maincounter_value":100,"batchcounter_value":10}
}
```

- Response:

```
{
  "cid":4711,
  "adr":"io/counter[2]/force_counter_value",
  "code":200
}
```

9.2.10 Gateway: Read device information

Substructure: `deviceinfo`

Available data points:

Name	Description	Values	Access
../productcode	Article number	AL4002	ro [1]
../vendor	Manufacturer	ifm electronic	ro [1]
../devicefamily	Device family	Remote IO	ro [1]
../serialnumber	Serial number (12 digits)	e.g. 000174210161	ro [1]
../hwrevision	Hardware revision	e.g. AA	ro [1]
../swrevision	Firmware version	e.g. AL4x0x_fw_pn_v1.4.0.137	ro [1]
../bootloaderrevision	Bootloader version	e.g. AL4xxx_bl_v1.2.0.35	ro [1]
../fieldbustype	Fieldbus	PROFINET	ro [1]

[1] read only

9.2.11 Gateway: Read status and diagnostic information

Substructure: `processdatamaster`

Available data points:

Name	Description	Values	Access
../temperature	Temperature of the device (value in °C)	e.g. 52	ro [1]
../voltage_us	Present voltage value of the device supply US (value in mV)	e.g. 25236	ro [1]
../current_us	Present current value of the device supply US (value in mA)	e.g. 82	ro [1]
../supervisionstatus_us	Status of the device supply US	• 0: no error • 1: Error	ro [1]

[1] read only

9.2.12 Gateway: Set the application tag

Substructure: `devicetag`

Available data points:

Name	Description	Values	Access
../applicationtag	Name of the device in moneo or LR SMARTOBSERVER	e.g. “factory 2 plant 1”	rw [1]

[1] read and write

32 bytes are available on the device for storing the applicationtag parameter. If the memory range is exceeded, the device will abort the write process (diagnostic code 400).

▶ When writing the applicationtag parameter, note the different memory requirements of the individual UTF-8 characters (characters 0-127: 1 byte per character; character >127: more than 1 byte per character).

9.2.13 Gateway: Update firmware

Substructure: `firmware`

Available data points:

Name	Description	Values	Access
../version	Firmware version	AL4x0x_fw_pn_1.4.0.137	ro [1]
../type	Software type	Firmware	ro [1]
../container	Structure for firmware (BLOB)	-	wo [2]
../container/maxsize	Container size (in bytes)	E.g. 4194304	ro [1]
../container/chunksize	Size of a data segment (in bytes)	E.g. 4096	ro [1]
../container/size	Size of firmware file in container (in bytes)	E.g. 634523	ro [1]

[1] read only

[2] write only

Applicable services:

Name	Description
../install	Install firmware
../container/stream_set	Transfer an individual data segment
../container/start_stream_set	Start sequential transmission of several data segments

ifm recommends using the IoT Core Visualizer (→ IoT-Core Visualizer 🗋 37) to update the firmware.

9.2.14 IoT-Core Visualizer

The IoT Core Visualizer provides a graphical user interface to access the functions of the ifm IoT Core.

9.2.14.1 Start the ifm IoT Core Visualizer

Requirements:

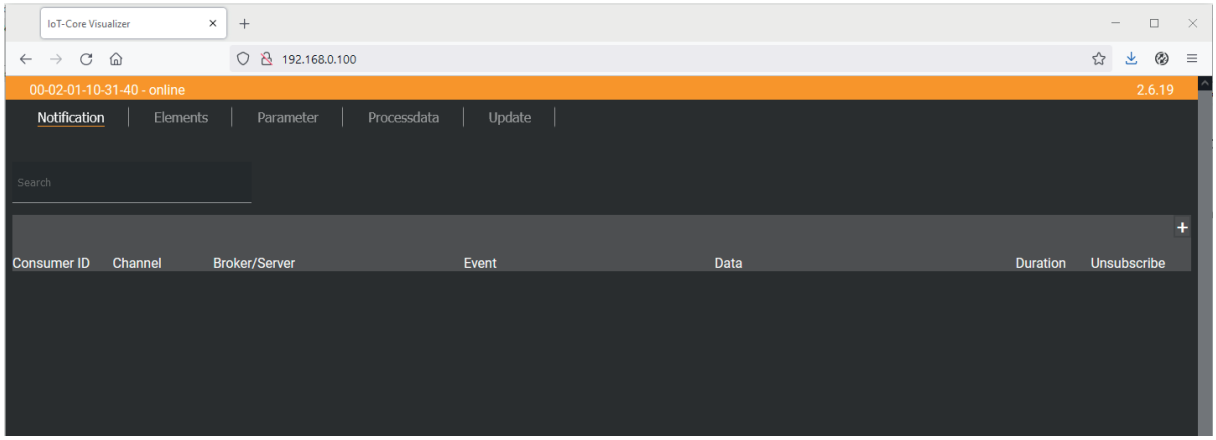
✓ The PC is connected to the Ethernet interface of the device.

✓ Ethernet interface has been configured correctly.

▶ Start web browser.

▶ Go to the following URL: http://<ip-address> (e.g. `http://192.168.0.10`)

▷ The web browser displays the start page of the IoT Core Visualiser.



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

- [Notification]: no function
- [Elements]: Search for elements in the device description (→ 🗋 37)
- [Parameter]: Configure the device (→ 🗋 38)
- [Processdata]: Access process data (→ 🗋 38)
- [Update]: Update firmware (→ 🗋 38)

9.2.14.2 Search for elements in the device description

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

Requirements:

✓ The ifm IoT Core Visualizer has been started.

▶ Click on [Elements].

▷ The menu page to search for elements appears.

▷ The input mask appears.

▶ Select the search criteria of the required element in the selection lists `identifier`, `profile` and `type`.

▶ Click on [Search for ...].

▷ The ifm IoT Core Visualizer searches the device description for elements with the selected search criteria.

▷ The result list shows all elements found.

9.2.14.3 Configure the device

The [Parameter] menu page allows you to configure the device.

Requirements:

✓ The ifm IoT Core Visualizer has been started.

▶ Click on [Parameter].

▷ The menu page displays the available parameters of the device.

▷ Current parameter values are displayed.

▶ Navigate to the desired parameter.

▶ Change the parameter value.

▶ Click on ✎ to save the changes.

▷ The changed parameter value is written to the device.

▷ The changed parameter value is active.

▶ Optional: Repeat the procedure to change further parameter values.

▷ The device has been configured.

9.2.14.4 Access process data

The [Processdata] menu page makes it possible to read and write the process data of the device and the connected sensors.

Requirements:

✓ The ifm IoT Core Visualizer has been started.

▶ Click on [Processdata].

▷ Menu page shows the substructures of the device description that contains the process data.

▷ The current process values are displayed.

▶ Optional: Activate the [Polling] option and change the update interval.

▷ The process values will be updated with the set interval.

▶ Optional: Click on ↻ next to an element to manually update the process value.

To change the value of a process date:

▶ Navigate to the required process date.

▶ Change the process value.

▶ Click on ✎ to save the changes.

▷ The changed process value is written to the device.

▷ The changed process value is active.

9.2.14.5 Update firmware

The [Update] menu page allows you to update the firmware of the device:

Requirements:

✓ The ifm IoT Core Visualizer has been started.

▶ Click on [Update].

▷ The menu page displays information about the current firmware version.

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

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

▷ The firmware of the device is updated.

▷ The area shows the progress of the update process.

▷ After successful update: The device reboots automatically.

9.3 PROFINET

9.3.1 Install the GSD file

To map the device in a PROFINET projection software, ifm provides an GSD file. The device description file contains identification information, supported parameters and process data. The user can download the GSD file via documentation.ifm.com.

To add the device to the hardware catalogue of the PROFINET projection software:

▶ Download the GSD file of the device.

▶ Launch the PROFINET projection software.

▶ Install the GSD file.

▷ The device is added to the hardware catalogue of the PROFINET projection software.

▷ The PROFINET projection software can access the device functions and data.

9.3.2 Integrate the device into a PROFINET project

Using the installed device description in the hardware catalogue, the device can be added to a PROFINET project.

Requirements:

✓ The GSD file of the device is installed.

▶ Create a new project or open an existing one.

▶ Open the [Device & networks] view.

▶ Add necessary components of the automation network (e.g. PROFINET PLC).

▶ Select the device in the hardware catalogue and add it to the network using drag & drop.

▶ Establish a logical PROFINET IO connection between the device and the PROFINET PLC.

▶ Set the IP configuration of the device's PROFINET interface.

▷ The device has been integrated in the PROFINET project.

9.3.2.1 Use S2 redundancy

The device supports the S2 redundancy. S2 redundancy provides a solution to build a redundant system for implementation of fail-safe systems. The following constraints apply:

- Both Application Relations (SR-AR) use the same configuration (submodule settings, connection parameters).
- Only one AR can act as a primary SR-AR.
- Diagnostic messages and alarms are only reported to the primary SR-AR.
- Data records can be read via any SR-AR.
- Data records can only be written via the following SR-AR:
  - primary SR-AR
  - SR-AR that first accesses the device while establishing the connection

9.3.2.2 Use Configuration-in-Run

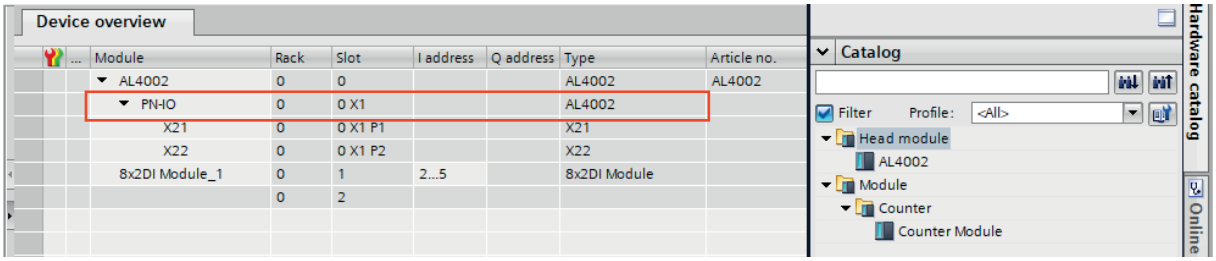
The device supports the Configuration-in-Run (CiR) function. CiR enables the user to load changes and extensions of the hardware configuration in the PROFINET configuration software to the PLC without needing to put the PLC into the “stop” state first.

9.3.2.3 Use Isochronous Realtime (IRT)

The unit supports participation in a network with activated IRT protocol.

9.3.3 Configure the device

The device is configured via slot 0, subslot X1. The module [PN-IO] is permanently assigned to the slot. The assignment cannot be changed.



Requirements:

✓ Device is integrated in PROFINET project.

▶ Open the [Device view].

▷ The [Device overview] tab displays the device structure.

▶ Click on the module [PN_IO] in the slot [0 X1].

▷ The [Properties] view appears.

▷ The [General] tab displays the available configuration options of the device.

▶ Set the parameters.

▷ The device has been configured.

9.3.3.1 Use Prioritized Startup

The device supports the Prioritized Startup function.

To activate the Prioritized Startup function:

Requirements:

✓ Device is integrated in PROFINET project.

▶ Open the [Device view].

▶ Click on the module [PN_IO] (slot 1, subslot X1).

▷ The [General] tab displays general configuration options.

▶ Select [Advanced options] > [Interface options].

▶ Activate the [Prioritized startup] option.

▷ The Fast Startup function has been activated.

9.3.3.2 Use Fast Startup

The device supports the Fast Startup function (FSU).

Achieving the guaranteed fast startup time depends on the complexity and elements used in the PROFINET network.

To activate the Fast Startup function:

Requirements:

✓ Device is integrated in PROFINET project.

✓ The Fast Startup function has been activated.

▶ Open the [Device view].

▷ The [Device overview] tab displays the device structure.

▶ Click on the module [PN_IO] (slot 1, subslot X1).

▷ The [General] tab displays general configuration options.

▶ Select [Advanced options] > [X21 [X1 P1 R]] > [Port options].

▶ In the list [Transmission rate / duplex], select the value [TP 100Mbps full duplex]. The selected value must match the transmission rate of the network partner.

▶ Disable the option [Enable autonegotiation].

▷ The transmission rate of Ethernet port 1 is permanently set.

▶ Repeat steps for Ethernet port 2 ([X22 [X1 P2 R]]).

▷ The transmission rate of Ethernet port 2 is permanently set.

▷ The Fast Startup function is activated for PROFINET IO ports.

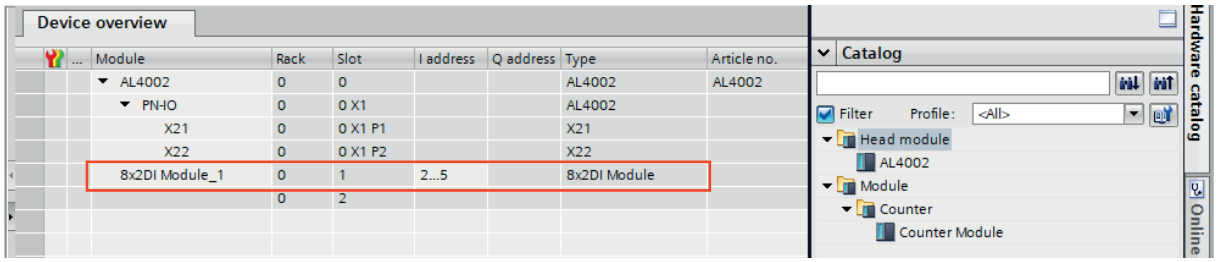
9.3.4 Configure input filters

▶ Observe the notes on input filters: Digital input filters (→ 🗋 8)

The input filters are configured via the following module:

- Modules: 8x2DI + Qualifier (→ 🗋 54)

The module is permanently assigned to slot 1.



Each digital input on pin 2 and pin 4 of the ports has a filter. Each of these filters can be configured separately.

Available parameters per input filter:

- debounce time
- hold time
- hold level
- signal inversion

Requirements:

✓ Device is integrated in PROFINET project.

▶ Open the [Device view].

▷ The [Device overview] tab displays the device structure.

▶ Click on the module [8x2 DI + Qualifier] (slot [1]).

▷ The [Properties] view appears.

▷ The [General] tab displays the properties of the module.

▷ [Module parameters] provides access to parameters of the digital inputs.

▶ Set the parameters of the input filters.

▶ Save the project.

▷ The filters of the digital inputs are configured.

▷ The changed configuration will be activated the next time the application is downloaded to the device.

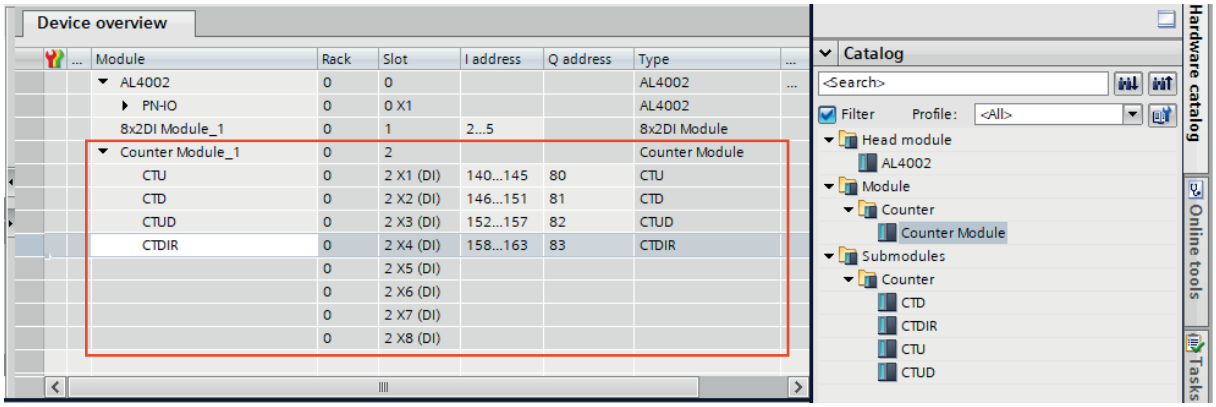
9.3.5 Configure counter modules

▶ Observe the notes on counter modules: Counters (→ 🗋 10)

The counter modules are configured via the module [Counter Module] and its submodules:

- Submodule: CTU (→ 🗋 55)
- Submodule: CTD (→ 🗋 56)
- Submodule: CTUD (→ 🗋 57)
- Submodule: CTDIR (→ 🗋 58)

The module [Counter Module] needs to be assigned to slot 2 manually. Subsequently, a counter submodule can be assigned to each port via the respective subslot. The counter submodule determines the operating mode of the counter.



Each counter module can be configured separately.

Available parameters of the counter submodules:

- event messages of the main counter
- event messages of the batch counter
- threshold CT of the main counter
- threshold CTb of the batch counter
- function of pin 2 of the port
- instance for selecting the counting direction (only with CTDIR operating mode)

Requirements:

✓ Device is integrated in PROFINET project.

▶ Open the [Device view].

▷ The [Device overview] tab displays the device structure.

▶ In the hardware catalogue under [Modul] > [Counter], select the module [Counter Module] and move it to slot 2 using drag & drop.

▷ The counter module is assigned to the device.

▷ The device overview displays the port designations assigned to the slot.

▷ The hardware catalogue displays the available submodules.

▶ In the hardware catalogue under [Submodule] > [Counter], select the desired counter submodule and move it to a free subslot of the port using drag & drop.

▷ The counter submodule is assigned to the port.

▶ Click on the subslot with the assigned counter submodule.

▷ The [Properties] view appears.

▷ The [Properties] tab shows the properties of the counter submodule.

▷ [Module parameters] provides access to parameters of the counter submodule.

▶ Set the parameters.

▶ Optional: Repeat the procedure to assign and configure further counter submodules.

▷ Counter submodules are assigned to the ports and configured.

9.3.6 Read process data of the ports

The digital input data of the ports and the associated validity indicators are transmitted in the cyclic process data: Modules: 8x2DI + Qualifier (→ 🗋 59)

When the device is integrated into the PROFINET project, IEC addresses are generated automatically for the digital input data on pin 2 and pin 4 of the ports X01...X08.

Requirements:

✓ Device is integrated in PROFINET project.

▶ Connect the IEC addresses of the subslots with symbols in the global variable list.

▷ The user can access the digital input data of the ports via symbol names in the application.

9.3.7 Read counter values

The current counter values are accessed via the cyclical input data:

- Submodule: CTU (→ 🗋 60)
- Submodule: CTD (→ 🗋 61)
- Submodule: CTUD (→ 🗋 62)
- Submodule: CTDIR (→ 🗋 63)

When configuring the counter submodules, IEC addresses are generated automatically for the counter values of the main and batch counter.

Requirements:

✓ Device is integrated in PROFINET project.

✓ The counter submodules are configured.

▶ Connect the IEC addresses of the subslots with symbols in the global variable list.

▷ The user can access the counter values via symbol names in the application.

9.3.8 Control counters

The counter submodules assigned to the ports can be controlled separately. The following control signals are available per counter:

- Reset counter module
- Disable counter module
- Set counting direction (only for CTUD counter operating mode)

The control signals are accessed via the cyclical output data:

- Submodule: CTU (→ 🗋 60)
- Submodule: CTD (→ 🗋 61)

- Submodule: CTUD (→ 🗋 62)
- Submodule: CTDIR (→ 🗋 63)

To control the counter modules:

Requirements:

✓ Device is integrated in PROFINET project.

✓ The counter submodules are configured.

✓ The IEC addresses of the cyclic process data are linked to variables.

▶ Set the control signals in the output data of the counter submodules.

▷ The control signals are sent to the counter submodule.

9.3.9 Acyclic access

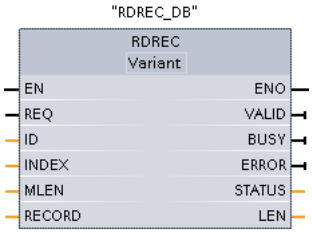
The user can access configuration and process data acyclically via data records.

The device supports the following data records:

- Data record: Filter configuration (→ 🗋 64)
- Data record: Counter configuration (→ 🗋 65)
- Data record: Counter values (→ 🗋 66)

Note: Read data record

▶ Read the data record with the function block RDREC.



Specific parameters:

- Input ID: HW_ID of the subslot (project-specific); properties of the subslot
- Input INDEX: index of the data record

9.3.10 Read counter events

The counter modules generate events when any of the following conditions occur:

- overflow main counter
- underflow main counter
- overflow batch counter
- underflow batch counter

The events triggered by the device are transmitted via the alarm mechanism of PROFINET IO. The overflow and underflow events of the counter modules are transmitted as process alarms. Process alarms are stored in an AINFO array. The relevant information is stored in bytes 16...35.

Structure of the AINFO array:

Bytes	Content	Description
16…19	Module ID	Module identification • 0x40000000: 8x2 DI Module • 0x41000000: Counter module
20…23	Submodule ID	Submodule identification • 0x41000010: CTU Submodule • 0x41000011: CTD Submodule • 0x41000012: CTUD Submodule • 0x41000013: CTDIR Submodule

Bytes	Content	Description
24…25	Alarm Specifier	Alarm • 0x0000: fixed
26…27	Format Identifier	Identification of the format • 0x8320: user-specific structure
28…29	Channel Number	Channel number • 0x8000: whole device
30…31	reserved	Reserved
32…33	Event Info	Event • 0x0100: overflow main counter • 0x0101: underflow main counter • 0x0110: overflow batch counter • 0x0111: underflow batch counter
34…35	reserved	Reserved

Requirements:

✓ The counter modules are configured.

✓ Events for main counter are activated.

✓ Events for batch counter are activated.

▶ Process the process alarms with OB40.

▶ Use the RALRM instruction to read the occurred event in the AINFO array.

9.3.11 Use I&M data records

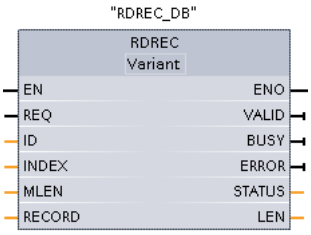
The unit supports the I&M data records I&M0 to I&M3 (→ I&M data 67).

- The data record I&M0 contains ID information. The I&M0 data record is available for each module / submodule of the device.
- In the data records I&M1...I&M3, the user can store application and device-specific information.

I&M data records can be accessed acyclically. Access to the I&M data records takes place index-based.

Note: Read data record

▶ Read the data record with the function block RDREC.



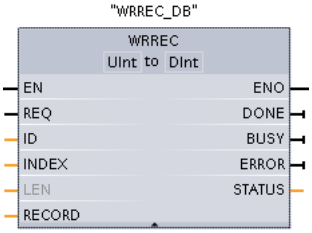
Specific parameters:

- Input ID: HW_ID of the subslot (project-specific); properties of the subslot
- Input INDEX: index of the data record

Note: Write data record

▶ Write the data record with the function block WRREC.

▶ Observe read and write permissions!



Specific parameters:

- Input ID: HW-ID of the slot (project-specific); properties of the slot
- Input INDEX: index of the data record

10 Maintenance, repair and disposal

The operation of the unit is maintenance-free.

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

10.1 Cleaning

▶ Disconnect the unit from the voltage supply.

▶ Clean the device from dirt using a soft, chemically untreated and dry cloth.

▶ In case of severe soiling, use a damp cloth.

▶ Do not use any caustic cleaning agents for this!

10.2 Update firmware

The system software of the device can be updated using the following options:

- IoT-Core Visualizer
- IoT-Core REST API

11 Appendix

11.1 ifm IoT Core

11.1.1 Profiles

Profile	Description
blob	Binary Large Object
deviceinfo	Identification information of a device
devicetag	Device-specific identification
devicereset	Restart and reset to factory settings
network	Network
parameter	Parameter
processdata	Process data
service	Service
software	Software
software/uploadablesoft-ware	Upgradeable software

11.1.2 Types

Type	Description
structure	Structural element (e.g. a folder in the file system)
service	Service that can be addressed from the network
event	An event that can be started by the firmware and sends messages.
data	Data point
device	Root element a device represents

11.1.3 Services

11.1.3.1 Service: factoryreset

Name: `factoryreset`

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

Request ("data” field): none

Return ("data" field): none

11.1.3.2 Service: force_counter_values

Name: `force_counter_values`

Description: The service writes the values of the main counter and batch counter. The service can only be executed if there is no connection to the fieldbus controller.

Request ("data” field):

Parameter	Mandatory field	Data type	Description
maincounter_value	Optional	INT	Main counter target value

Parameter	Mandatory field	Data type	Description
batchcounter_value	Optional	INT	Batch counter target value

Return (“data” field): none

11.1.3.3 Service: getblobdata

Name: `getblobdata`

Description: The service reads a Binary Large Object (blob).

Request (“data“ field):

Data field	Mandatory field	Data type	Description
pos	mandatory	NUMBER	Byte position
length	mandatory	NUMBER	Size of the object (number of bytes)

Return ("data" field):

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

11.1.3.4 Service: getdata

Name: `getdata`

Description: The service reads the value of a data point and outputs it.

Request ("data” field): none

Return data ("data" field):

Parameter	Mandatory field	Data type	Description
value	mandatory	STRING	Value of the data point

11.1.3.5 Service: getdatamulti

Name: `getdatamulti`

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

Request ("data" field):

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

Return ("data" field):

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

11.1.3.6 Service: getelementinfo

Name: `getelementinfo`

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

Request ("data" field):

Parameter	Mandatory field	Data type	Description
adr	mandatory	STRING	URL of the element whose proerties are to be changed

Return ("data" field):

Parameter	Mandatory field	Data type	Description
identifier	mandatory	STRING	Identifier of the element
type	mandatory	STRING	Type of the element
format	optional	JSON object	Format of the data or of the service content
uid	optional	STRING	
profiles	optional	JSON-ARRAY	Element profiles
hash	optional	STRING	

11.1.3.7 Service: getidentity

Name: `getidentity`

Description: The service reads device information and outputs it.

Request (field „data“): none

Response (field „data“):

Parameter	Mandatory field	Data type	Description
iot		device	Device description as JSON object
iot.name	mandatory	STRING	Type of the element
iot.uid	optional	STRING	
iot.version	mandatory	STRING	
iot.catalogue	optional	ARRAY OF OBJECTS	
iot.deviceclass	optional	ARRAY OF STRING	Device class
iot.serverlist	optional	ARRAY OF OBJECTS	
device	optional		Article nummer
device.serialnumber	optional		Serial number
device.hwrevision	optional		Hardware version
device.swrevision	optional		Software version
device.custom	optional		

11.1.3.8 Service: gettree

Name: `gettree`

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

Request (“data” field):

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

Return (“data” field)

Parameter	Mandatory field	Data type	Description
identifier	Mandatory	STRING	Identifier of the root element
type	Mandatory	STRING	Type of the element
format	Optional	JSON object	Format of the data content
uid	Optional	STRING	
profiles	Optional	JSON array	
subs	Mandatory	JSON array	Sub-elements
hash	Optional	STRING	
adr	Mandatory	STRING	Root element of the subtree

11.1.3.9 Service: install

Name: `install`

Description: The service installs the firmware stored in a memory area of the unit.

Request ("data" field): none

Return ("data" field): none

11.1.3.10 Service: querytree

Name: `querytree`

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

Request ("data" field):

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

Return ("data" field):

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

11.1.3.11 Service: reboot

Name: `reboot`

Description: The service reboots the device.

Request ("data" field): none

Return ("data" field): none

11.1.3.12 Service: setblock

Name: `setblock`

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

Request ("data" field):

Parameter	Mandatory field	Data type	Description
datatoset	mandatory	ARRAY OF OBJECTS	List of data points and their new values; Data points must support the `setdata` service
consitent	optional	BOOL	IO-Link subindex of the parameter

Return ("data" field): none

11.1.3.13 Service: setdata

Name: `setdata`

Description: The service sets the value of the data point.

Request ("data" field):

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

Return ("data" field): none

11.1.3.14 Service: signal

Name: `signal`

Description: The service triggers the flashing of the status LEDs of the unit.

Request ("data" field): none

Return ("data" field): none

11.1.3.15 Service: start_stream_set

Name: `start_stream_set`

Description: The service starts the sequential transmission of several data fragments.

Request ("data" field):

Parameter	Mandatory field	Data type	Description
size	mandatory	STRING	Overal length of the data to be transmitted (number of bytes)

Return ("data" field): none

11.1.3.16 Service: stream_set

Name: `stream_set`

Description: The service transfers a data segment.

Request ("data" field):

Parameter	Mandatory field	Data type	Description
value	mandatory	BIN (BASE64)	Segment of the binary data (BASE64-coded)

Return ("data" field): none

11.2 PROFINET

11.2.1 Parameters

11.2.1.1 Modules: 8x2DI + Qualifier

Parameter	Description	Values	Access
Debounce Time	Debounce time (value * 0.1 ms)	• 0: 0 ms (default) … • 500: 50 ms	rw [1]
Hold time	Hold time (value * 0.1 ms)	• 0: 0 ms (default) … • 60000: 6000 ms	rw [1]
Hold level	Hold level	• 0: Low: hold LOW • 1: High: hold HIGH (default)	rw [1]
Input Inverter	signal inversion	• 0: Signal not inverted: do not invert signal (default) • 1: Signal inverted: invert signal	rw [1]

[1] read and write

11.2.1.2 Modules: Counter module

Submodule: CTU

Parameter	Description	Values	Access
Enable Main counter event	Notifications for main counter	• ☐: disable • ☑: enable (default)	rw [1]
Enable Main counter event	Notifications for batch counter	• ☐: disable • ☑: enable (default)	rw [1]
Pin 2 function	function of pin 2 of the port	• Not used: no function (default) • Reset main & batch counter: Reset main counter and batch counter • Disable main & batch counter: Disable main counter and batch counter	rw [1]
Main Threshold	Threshold CT of the main counter	• 1 … • 4294967295 (default)	rw [1]
Batch Threshold	Threshold CTb of the batch counter	• 1 … • 65535 (default)	rw [1]

[1] read and write

Submodule: CTD

Parameter	Description	Values	Access
Enable Main counter event	Notifications for main counter	• ☐: disable • ☑: enable (default)	rw [1]
Enable Main counter event	Notifications for batch counter	• ☐: disable • ☑: enable (default)	rw [1]
Pin 2 function	function of pin 2 of the port	• Not used: no function (default) • Reset main & batch counter: Reset main counter and batch counter • Disable main & batch counter: Disable main counter and batch counter	rw [1]
Main Threshold	Threshold CT of the main counter	• 1 … • 4294967295 (default)	rw [1]
Batch Threshold	Threshold CTb of the batch counter	• 1 … • 65535 (default)	rw [1]

[1] read and write

Submodule: CTUD

Parameter	Description	Values	Access
Enable Main counter event	Notifications for main counter	• ☐: disable • ☑: enable (default)	rw [1]
Enable Main counter event	Notifications for batch counter	• ☐: disable • ☑: enable (default)	rw [1]
Pin 2 function	function of pin 2 of the port	• Counter edge input 2: counting input (default)	rw [1]
Main Threshold	Threshold CT of the main counter	• 1 … • 4294967295 (default)	rw [1]
Batch Threshold	Threshold CTb of the batch counter	• 1 … • 65535 (default)	rw [1]

[1] read and write

Submodule: CTDIR

Parameter	Description	Values	Access
Enable Main counter event	Notifications for main counter	• ☐: disable • ☑: enable (default)	rw [1]
Enable Main counter event	Notifications for batch counter	• ☐: disable • ☑: enable (default)	rw [1]
Pin2 function / Count direction selection	Pin 2 function of the port and selection of the control instance for selecting the counting direction	• Pin2 Count direction: signal at pin 2 controls counting direction • Pin2 Not used & Count Direction by PLC: pin 2 without function; selection of counting direction via PLC • Pin2 Reset Counter & Count Direction by PLC: signal at pin 2 resets main counter and batch counter; selection of counting direction via PLC • Pin2 Reset Counter & Count Direction by PLC: signal at pin 2 deactivates main counter and batch counter; selection of counting direction via PLC	rw [1]
Main Threshold	Threshold CT of the main counter	• 1 … • 4294967295 (default)	rw [1]
Batch Threshold	Threshold CTb of the batch counter	• 1 … • 65535 (default)	rw [1]

[1] read and write

11.2.2 Cyclic data

11.2.2.1 Modules: 8x2DI + Qualifier

Input data: 4 bytes

Byte (offset)	Bit							
	7	6	5	4	3	2	1	0
n	X4 (pin 2): DI	X4 (pin 4): DI	X3 (pin 2): DI	X3 (pin 4): DI	X2 (pin 2): DI	X2 (pin 4): DI	X1 (pin 2): DI	X1 (pin 4): DI
n+1	X8 (pin 2): DI	X8 (pin 4): DI	X7 (pin 2): DI	X7 (pin 4): DI	X6 (pin 2): DI	X6 (pin 4): DI	X5 (pin 2): DI	X5 (pin 4): DI
n+2	X4 (pin 2): QDI	X4 (pin 4): DQI	X3 (pin 2): QDI	X3 (pin 4): DQI	X2 (pin 2): QDI	X2 (pin 4): DQI	X1 (pin 2): QDI	X1 (pin 4): DQI
n+3	X8 (pin 2): QDI	X8 (pin 4): DQI	X7 (pin 2): QDI	X7 (pin 4): DQI	X6 (pin 2): QDI	X6 (pin 4): DQI	X5 (pin 2): QDI	X5 (pin 4): DQI

Legend:

• DI	Signal level of the digital input	1 bit	• 0: LOW • 1: HIGH
• QDI	Validity of the process value of the digital input	1 bit	• 0: invalid • 1: valid

Output data: none

11.2.2.2 Submodule: CTU

Input data: 6 bytes

Byte (offset)	Bit							
	7	6	5	4	3	2	1	0
0…3	Main Counter Value							
4…5	Batch Counter Value							

Legend:

• Main Counter Value	Current main counter value Main Counter Value = 0x12345678 • Main Counter Value[0] = 0x12 • Main Counter Value[1] = 0x34 • Main Counter Value[2] = 0x56 • Main Counter Value[3] = 0x78	UINT32	• 0x00000000: 0 … • 0xFFFFFFFE: 4294967294
• Batch Counter Value	Current batch counter value Batch Counter Value = 0x1234 • Batch Counter Value[0] = 0x12 • Batch Counter Value[1] = 0x34	UINT16	• 0x0000: 0 … • 0xFFFE: 65534

Output data: 1 byte

Byte (offset)	Bit							
	7	6	5	4	3	2	1	0
0	Reserved						Disable Counter	Reset Counter

Legend:

• Reset Counter	Reset main counter and batch counter to initial value	1 bit	• 0x0: no action • 0x1: reset
• Disable Counter	Disable main counter and batch counter	1 bit	• 0x0: no action • 0x1: disable

11.2.2.3 Submodule: CTD

Input data: 6 bytes

Byte (offset)	Bit							
	7	6	5	4	3	2	1	0
0…3	Main Counter Value							
4…5	Batch Counter Value							

Legend:

• Main Counter Value	Current main counter value Main Counter Value = 0x12345678 • Main Counter Value[0] = 0x12 • Main Counter Value[1] = 0x34 • Main Counter Value[2] = 0x56 • Main Counter Value[3] = 0x78	UINT32	• 0x00000000: 0 … • 0xFFFFFFFE: 4294967294
• Batch Counter Value	Current batch counter value Batch Counter Value = 0x1234 • Batch Counter Value[0] = 0x12 • Batch Counter Value[1] = 0x34	UINT16	• 0x0000: 0 … • 0xFFFE: 65534

Output data: 1 byte

Byte (offset)	Bit							
	7	6	5	4	3	2	1	0
0	Reserved						Disable Counter	Reset Counter

Legend:

• Reset Counter	Reset main counter and batch counter to initial value	1 bit	• 0x0: no action • 0x1: reset
• Disable Counter	Disable main counter and batch counter	1 bit	• 0x0: no action • 0x1: disable

11.2.2.4 Submodule: CTUD

Input data: 6 bytes

Byte (offset)	Bit							
	7	6	5	4	3	2	1	0
0…3	Main Counter Value							
4…5	Batch Counter Value							

Legend:

• Main Counter Value	Current main counter value Main Counter Value = 0x12345678 • Main Counter Value[0] = 0x12 • Main Counter Value[1] = 0x34 • Main Counter Value[2] = 0x56 • Main Counter Value[3] = 0x78	UINT32	• 0x00000000: 0 … • 0xFFFFFFFE: 4294967294
• Batch Counter Value	Current batch counter value Batch Counter Value = 0x1234 • Batch Counter Value[0] = 0x12 • Batch Counter Value[1] = 0x34	UINT16	• 0x0000: 0 … • 0xFFFE: 65534

Output data: 1 byte

Byte (offset)	Bit							
	7	6	5	4	3	2	1	0
0	Reserved						Disable Counter	Reset Counter

Legend:

• Reset Counter	Reset main counter and batch counter to initial value	1 bit	• 0x0: no action • 0x1: reset
• Disable Counter	Disable main counter and batch counter	1 bit	• 0x0: no action • 0x1: disable

11.2.2.5 Submodule: CTDIR

Input data: 6 bytes

Byte (offset)	Bit							
	7	6	5	4	3	2	1	0
0…3	Main Counter Value							
4…5	Batch Counter Value							

Legend:

• Main Counter Value	Current main counter value Main Counter Value = 0x12345678 • Main Counter Value[0] = 0x12 • Main Counter Value[1] = 0x34 • Main Counter Value[2] = 0x56 • Main Counter Value[3] = 0x78	UINT32	• 0x00000000: 0 … • 0xFFFFFFFE: 4294967294
• Batch Counter Value	Current batch counter value Batch Counter Value = 0x1234 • Batch Counter Value[0] = 0x12 • Batch Counter Value[1] = 0x34	UINT16	• 0x0000: 0 … • 0xFFFE: 65534

Output data: 1 byte

Byte (offset)	Bit							
	7	6	5	4	3	2	1	0
0	Reserved					Counter Direction	Disable Counter	Reset Counter

Legend:

• Reset Counter	Reset main counter and batch counter to initial value	1 bit	• 0x0: no action • 0x1: reset
• Disable Counter	Disable main counter and batch counter	1 bit	• 0x0: no action • 0x1: disable
• Counter Direction	Set counting direction (only effective if parameter [Count direction selection] = [PLC])	1 bit	• 0x0: up • 0x1: down

11.2.3 Acyclical data

11.2.3.1 Data record: Filter configuration

Index	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315
Port	1	1	2	2	3	3	4	4	5	5	6	6	7	7	8	8
Pin	4	2	4	2	4	2	4	2	4	2	4	2	4	2	4	2

Per index:

Byte (offset)	Bit							
	7	6	5	4	3	2	1	0
0…1	Debounce Time							
2…3	Hold Time							
4	res.	res.	res.	res.	res.	res.	Hold Level	Input Inverter

Legend:

• Debounce Time	Debounce time (= value * 0.1 ms) Debounce Time = 0x0123: • Debounce time[0] = 0x01 • Debounce time[1] = 0x23	UINT16 / rc	• 0x0000: 0 ms … • 0x01F4: 50 ms
• Hold Time	Hold time (= value * 0.1 ms) Hold Time = 0x1234: • Hold time[0] = 0x12 • Hold time[1] = 0x34	UINT16 / rc	• 0x0000: 0 ms … • 0xEA60: 6000 ms
• Input Inverter	Inversion	1 bit / rc	• 0x0: do not invert • 0x1: invert
• Hold Level	Hold level	1 bit / rc	• 0x0: hold LOW • 0x1: hold HIGH

11.2.3.2 Data record: Counter configuration

Index: 500

Byte (offset)	Bit							
	7	6	5	4	3	2	1	0
0	Counter mode							
1	Reserved		Batch Event enable	Main Event enable	Count directionData record: Counter configuration (→ 65) | Pin 2 function / Count directionData record: Counter configuration (→ 65)			
2..5	Main Treshold							
6…7	Batch Threshold							

Legend:

• Counter mode	Operating mode counter module	UINT8 / rc	• 0x0: CTU – up counter • 0x1: CTD – down counter • 0x2: CTUD – up and down counter • 0x3: CTDIR – up and down counter with selectable counting direction
• Pin 2 function	Pin 2 function of the port	4 bits / rc	For [CTU] and [CTD]: • 0x00: no function • 0x03: reset counter module • 0x04: disable counter module For [CTUD]: • 0x01: count input
• Pin 2 function / Count direction	Pin 2 function of the port and desired counting direction	4 bits / rc	• 0x02: pin 2 determines counting direction • 0x08: pin 2 not used & PLC determines counting direction • 0x0B: pin 2 resets counter module & PLC determines counting direction • 0x0C: pin 2 disables counter module & PLC determines counting direction
• Main Event enable	Enable overflow/underflow event of the main counter	1 bit / rc	• 0x0: disable • 0x1: activate
• Batch Event enable	Enable overflow/underflow event of the batch counter	1 bit / rc	• 0x0: disable • 0x1: activate
• Main Threshold	threshold CT of the main counter	UINT32 /rc	• 0x00000001:1 … • 0xFFFFFFF: 4294967295
• Batch Threshold	threshold CTb of the batch counter	UINT16 /rc	• 0x0001: 1 … • 0xFFFF: 65535

only available for operating mode [CTDIR]

only available for operating modes [CTU], [CTD] and [CTUD]

11.2.3.3 Data record: Counter values

Index: 501

Byte (offset)	Bit							
	7	6	5	4	3	2	1	0
0…3	Main Counter value							
4…5	Batch Counter value							

Legend:

• Main Counter value	Counter value of the main counter	UINT32 / rw	0x00000000: 0 … 0xFFFFFFFE: 4294967294
• Batch Counter value	Counter value of the batch counter	UINT16 / rw	0x0000: 0 … 0xFFFE: 65534

11.2.3.4 I&M data

I&M0

Index: 0xAFF0

Variable	Description	Value	Bytes
MANUFACTURER_ID	Manufacturer ID	0x136	2
ORDER_ID	Order ID (ASCII, separated by spaces)	AL4xx2	20
SERIAL_NUMBER	Serial number (ASCII, separated by spaces)		16
HARDWARE_REVISION	Hardware revision	e.g. AA	2
SOFTWARE_REVISION	Software revision • Byte 0: software type (V: release) • Byte 1: main version (uint8) • Byte 2: subversion (uint8) • byte 3: build version (uint8)	e.g. V1.0.3	4
REVISION_COUNTER	Revision counter; counter is incremented with every parameter change	0x0000 … 0xFFFF	2
PROFILE_ID	Profile ID • 0x0000: unspecific	0x0000	2
PROFILE_SPECIFIC_TYPE	Profile type • 0x0000: unused	0x0000	2
IM_VERSION	I&M version • 0x0101: V1.1	0x0101	2
IM_SUPPORTED	Supported I&M data records • 0x000: I&M0 is supported • 0x00E: I&M0-3 are supported	• DAP: 0x000E • Submodule: 0x000	2

I&M1

Index: 0xAFF1

Variable	Description	Value	Bytes
TAG_FUNCTION	Identifier for function of the submodule • 0x20: empty	0x20	32
TAG_LOCATION	Identifier for location of the submodule • 0x20: empty	0x20	22

I&M2

Index: 0xAFF2

Variable	Description	Value	Bytes
INSTALLATION_DATE	Installation date of the submodule (ASCII, separated by spaces) • 0x20: empty	0x20	16
RESERVED	Reserved	0x00	38

I&M3

Index: 0xAFF3

Variable	Description	Value	Bytes
DESCRIPTOR	Description of the submodule (ASCII, separated by spaces) • 0x20: empty	0x20	54

I&M0 filter

Index: 0xAFF4

Variable	Description	Value	Bytes
API	API of the submodule		4
SLOT	Slot of the submodule		2
SUBSLOT	Subslot of the submodule		2
FLAGS	Flags: • 0x01: submodule has own I&M data • 0x02: I&M data of the submodule represent I&M data of the module • 0x04: I&M data of the submodule represent I&M data of the device		4