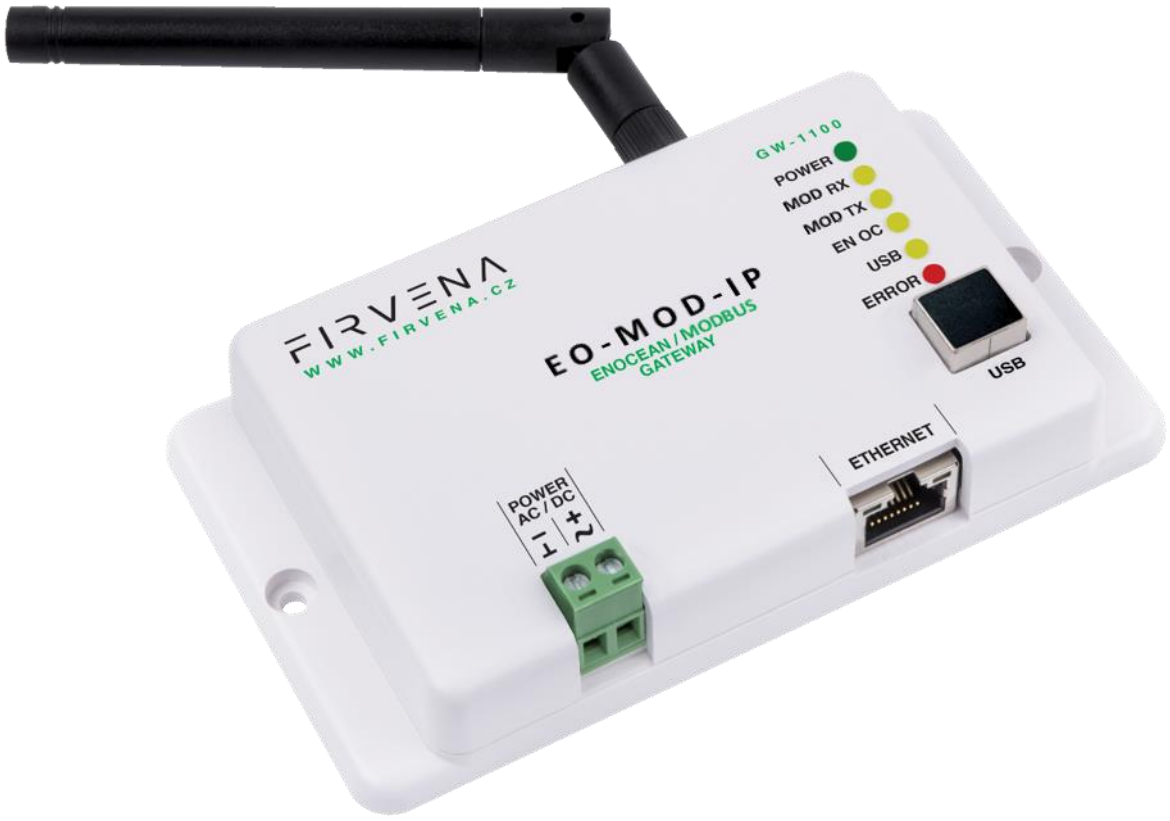
FIRVENA

EnOcean to Modbus/IP Gateway

EO-MOD-IP

User Manual V1.2

English



Contents

Terms and Abbreviations

Term/Abbr.	Explanation
Channel number (CH)	Identifier of EnOcean device within the gateway
DHCP	Dynamic Host Configuration Protocol
EEP	EnOcean Equipment Pofiles
EURID	EnOcean Unique Radio Identifier
Label	User-friendly name of EnOcean device
IP	Internet Protocol
PoE	Power over Ethernet
RX	Receive, reception
Teach-in	Pairing of EnOcean devices
Telegram	EnOcean message
TX	Transmit, transmission
UPnP	Universal Plug and Play
Value index	Identifier of a data unit within the channel

List of Figures and Tables

Figures

Tables

1 Introduction

The EO-MOD-IP device is a gateway between EnOcean and Modbus/IP communication protocols (Fig. 1.1). The EnOcean is a set of technologies and communication protocol that enables the use of wireless and batteryless sensors, switches and actuators. The Modbus is a communication protocol that is simple to implement and is widely used in building automation systems. The EO-MOD-IP gateway can receive data from up to 40 EnOcean devices, store it and provide it through the Modbus interface to other devices connected to the network.

The first part of this manual describes the hardware of the gateway. Chapter 4 describes the Modbus interface and explains how EnOcean devices are mapped to Modbus registers. Chapter 5 contains brief instructions for configuring the gateway, especially for connecting EnOcean devices. For firmware update procedure refer to Chapter 6.

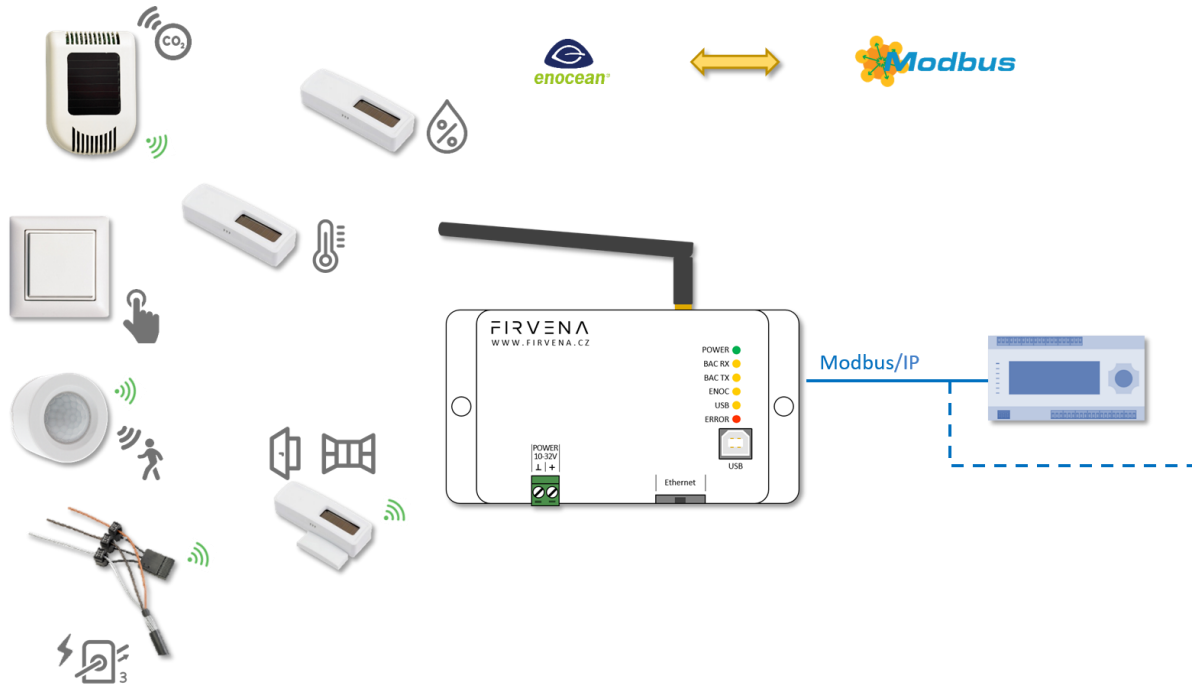


Fig. 1.1 Functional overview

2 Hardware Overview

There is a power connector, RJ45 Ethernet connector, type B USB connector, antenna connector and LED indicators on the front panel (Fig. 2.1). The LEDs indicates the connection of power supply (POWER), traffic in the Modbus network (MOD RX, MOD TX), traffic in the EnOcean wireless network (ENOC), communication through the USB interface (USB) and error states (ERROR). The USB connector is used for configuration and firmware update, it is used by the *EO-BAC Tool* configuration application. The green POWER connector is removable, which simplifies device handling if wires are connected.

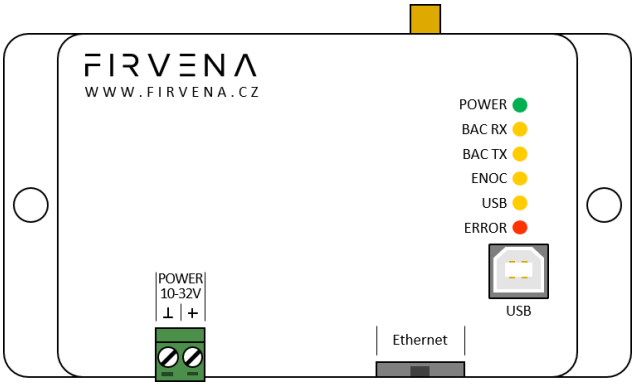


Fig. 2.1 Gateway overview – front side

There is a switch to the right of the POWER connector that is accessible after pulling out the POWER connector (Fig. 2.2). It sets the USB interface mode:

- **the "HID" position (default)** allows the *EO-BAC Tool* application to be connected
- **the "MSC" position** activates the firmware update mode (see Ch. 6.2)

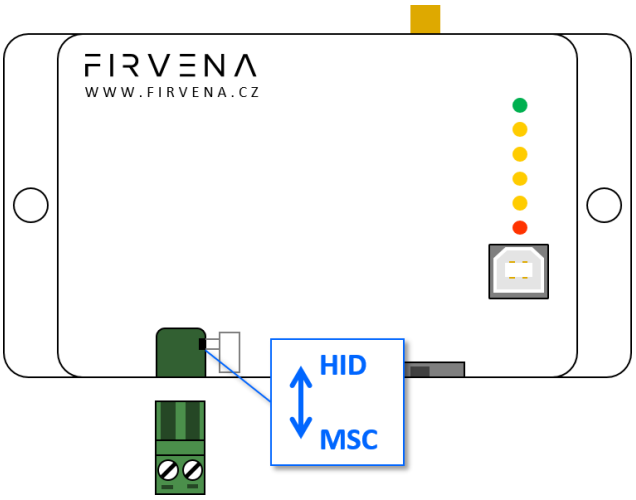


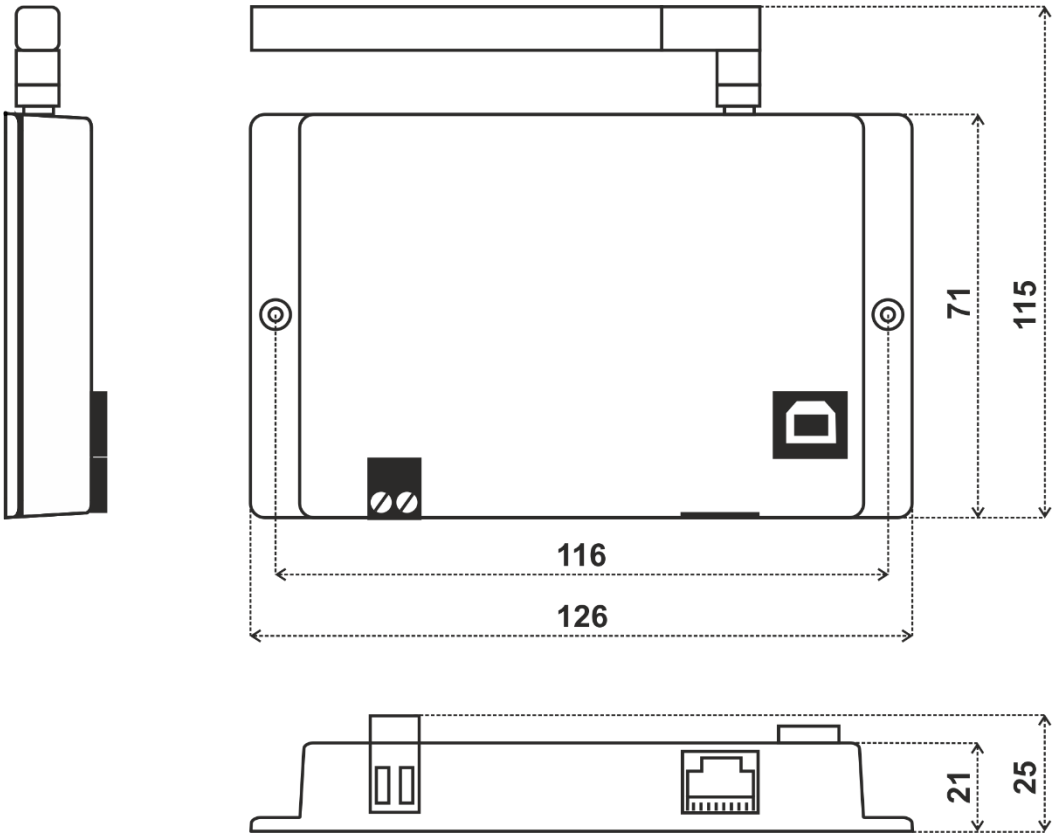
Fig. 2.2 Gateway overview – meaning of manual switches

3 Technical Data

Tab. 3.1 Technical data

Category	Parameter	Value
Product	Product name	EO-MOD-IP
	Product title	EnOcean to Modbus/IP Gateway
	Product ID	11.2
	Vendor name	FIRVENA s.r.o.
Electrical data	Rated supply voltage	24 V DC / 24 V AC
	Supply voltage range	10–32 V DC / 24 V AC (± 10 %)
	Rated input current	50 mA
	Rated input power	1.2 W
Ethernet (Modbus/IP)	Speed	10, 100 Mbit/s
	Connector	RJ45
	PoE	✓ (pins 4, 5, 7, 8)
	PoE supply powering	According to standard 802.3af
EnOcean	Frequency	868 MHz
	Maximum number of handled devices	40 (max. 20 with SmartACK)
	Repeater	✓
USB	Device class	Custom HID or Mass Storage
	Connector	Type B
Operating conditions	IP Code	IP20
	Operating temperature	−20 to +70 °C
	Relative humidity	max. 80 %
Dimensions in mm	Dimensions without antenna	Width=126, Height=71, Depth=25
Weight	Weight without antenna	115 g
Box material		ABS, white
EMC	In accordance with the directive	2014/53/EU, 2011/65/EU RoHS
EMC	Approvals tests	ČSN EN 55032, ČSN EN 55035, ČSN EN 6100-4-2, ČSN EN 6100-4-3, ČSN EN 6100-4-4, ČSN EN 6100-4-5, ČSN EN 6100-4-6, ČSN EN 6100-4-11, ČSN EN IEC 6100-6-2

Dimensions in mm:



4 Modbus Interface

4.1 Mapping of EnOcean Devices

EnOcean devices are mapped as a set of Modbus registers. The gateway can handle up to 40 EnOcean devices. To assign an EnOcean device, the teach-in procedure has to be carried out (see Ch. 5.2). The assigned EnOcean devices are identified by *Channel* (CH1…40) within the gateway, the *Channel* is selected by user during teach-in procedure.

Data fields received in a telegram are divided into individual Modbus registers so that they can be accessed using Modbus standard functions from the network – see Fig. 4.1. The gateway contains a database of supported EnOcean products, objects are created depending on the type of EnOcean device (EEP) that is assigned to the channel during teach-in procedure.

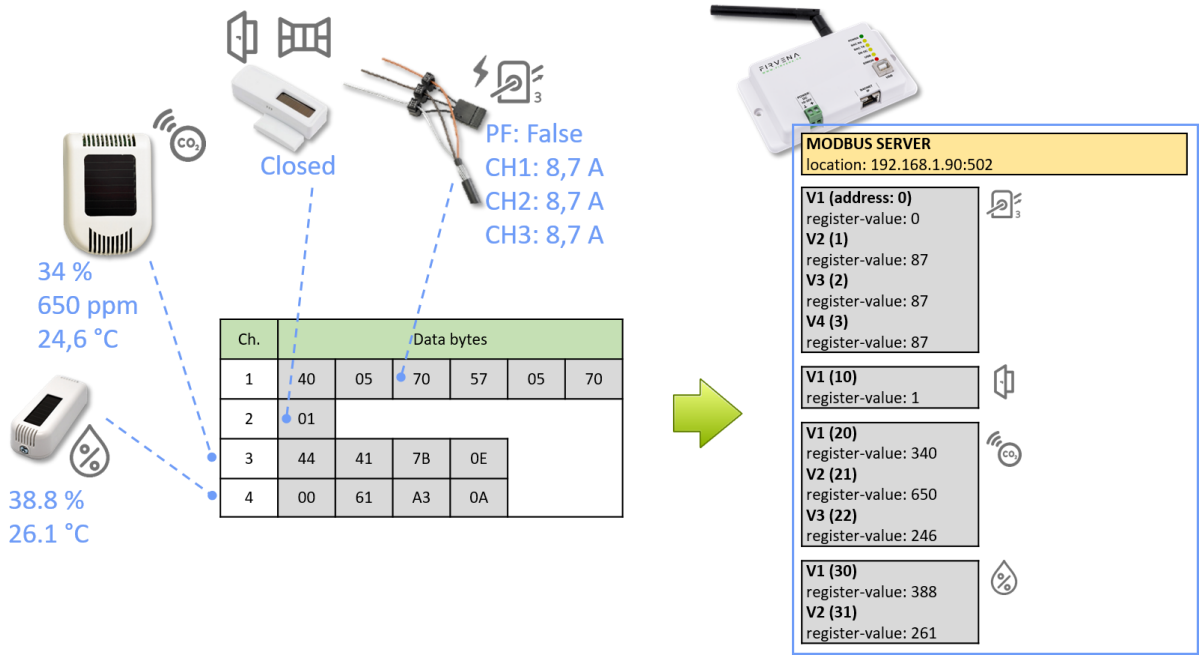


Fig. 4.1 Conversion of data

Three types of data fields are distinguished:

- **Numeric value:** usually a measured physical quantity such as temperature, humidity, etc.
- **Enumeration:** defined as a list of items that express a state or configuration of the device
- **Boolean:** two-state enumeration whose items can be interpreted as true/false, such as on/off, enabled/disabled, open/closed, etc.

In addition, two directions are distinguished:

- **RX:** data received by the gateway (incoming telegrams)
- **TX:** data transmitted by the gateway (outgoing telegrams)

The data fields are stored internally as *Values*. The *Values* are identified by *Value index*, 15 *Values* is reserved per channel (indexes 0 to 14) – see Fig. 4.2 and Fig. 4.3.

The *Values* from *Telegram data* range represent a model of EnOcean telegram. For simple devices (such as sensors with unidirectional communication) the model of telegram can also be assumed to be a model of EnOcean device. Some more complex EEP definitions consist of several types of telegrams

that represent different commands, so the device cannot be described by a single telegram. Special mapping is created for these devices.

For direction "RX", all used *Values* are mapped to *RX Values* registers (starting at address 0). For direction "TX" all used *Values* are mapped to *TX Values* registers (starting at address 1000). If the channel is not occupied the associated *Value* registers are zeroed.

The mapped data fields from EnOcean telegram are indexed in the same order as they appear in the EEP definition – see [3].

Besides the data values, there are also helper values for each channel that provide status information – see Tab. 4.1.

Tab. 4.1 Helper values

Index	Name	Meaning	Value Range
RX			
15	Telegram counter	Number of received telegrams	0…65535 (overflows to zero)
16	Telegram age	Time elapsed since the last telegram	0…65000 s (65535: no telegram, 65001: range exceeded)
17	Signal	Signal strength (RSSI value measured by the EnOcean transceiver)	0…–255 dBm
18	reserved		
19	Error	Enumeration of channel state and error codes	See Tab. 4.5
TX			
15	Telegram counter	Number of sent telegrams	0…65535 (overflows to zero)
16	Telegram age	Time elapsed since the last telegram	0…65000 s (65535: no telegram, 65001: range exceeded)
17	Send option	Data transmission control	See Tab. 4.7
18	reserved		

Examples of mapping for individual EnOcean devices can be found in ANNEX A.

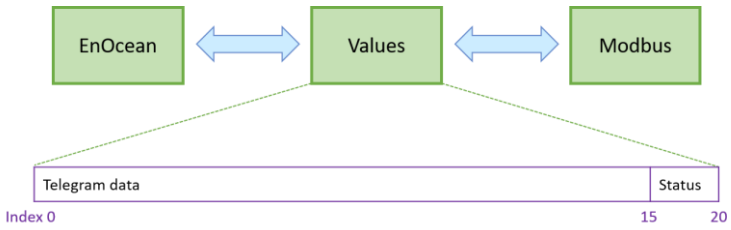


Fig. 4.2 Values

		Register		
CH1	Value 1	0	Temperature	A5-02-05 Temperature Sensors, Temperature Sensor Range 0°C to +40°C 
	Value 2	1		
	Value 3	2		
	Value 4	3		
	Value 5	4		
	...	...	...	
	Value 16	15	T. Counter	
	Value 17	16	T. Age	
	Value 18	17	Signal	
	Value 19	18		
	Value 20	19	Error	
CH2	Value 1	0	Humidity	A5-09-04 Gas Sensor, CO2 Sensor 
	Value 2	1	Concentration	
	Value 3	2	Temperature	
	Value 4	3	H-Sensor	
	Value 5	4	T-Sensor	
	...	...	...	
	Value 16	15	T. Counter	
	Value 17	16	T. Age	
	Value 18	17	Signal	
	Value 19	18		
	Value 20	19	Error	
...				
CH40	Value 1	0	Power Fail	D2-32-02 A.C. Current Clamp, 3 channels 
	Value 2	1	Divisor	
	Value 3	2	Channel 1	
	Value 4	3	Channel 2	
	Value 5	4	Channel 3	
	...	...	...	
	Value 16	15	T. Counter	
	Value 17	16	T. Age	
	Value 18	17	Signal	
	Value 19	18		
	Value 20	19	Error	

Fig. 4.3 Mapping of EnOcean devices

4.2 Connection to Modbus Server

Gateway's Modbus server is on TCP port 502 or UDP port 502. There is a limit of two simultaneous TCP connections.

4.3 Supported Function Codes

Tab. 4.2 Supported Modbus functions

Code	Name	Description
3 (0x03)	Read Holding Registers	Reads a continuous block of registers starting at a given address. Zero values are returned for unused registers within a defined area.
6 (0x06)	Write Single Register	Writes any writable register.
16 (0x10)	Write Multiple Registers	Writes a block of writable registers, behaviour differs depending on the area, see below.

4.4 Modbus Registers

The registers are divided into several areas according to their use. All registers can be read by Modbus function 3 or 4. Readonly registers are marked as “R”, writable registers are marked as “R/W”. “P” marks persistent (non-volatile) registers whose values are retained when the gateway is turned off.

Tab. 4.3 Register map

Area name	Address range	Access	Description	Channel	Base address
RX Values	0…799	R	Last received telegram – values extracted from raw data	CH1	0
				CH2	20
				…	
				CH40	780
Reserved					
TX Values	1000…1799	R/W	Telegram to send – values are built into raw data	CH1	1000
				CH2	1020
				…	
				CH40	1780
Reserved					
RX Raw data	2000…2799	R	Last received telegram – raw data as received in telegram	CH1	2000
				CH2	2020
				…	
				CH40	2780
Reserved					
Channel config	4000…4399	R, P	Channel configuration (i.e. assignment of EnOcean device)	CH1	4000
				CH2	4010
				…	
				CH40	4390
Reserved					
Channel labels	10000…13999	R, P	User defined descriptions for channels	CH1	10000
				CH2	10100
				…	
				CH40	13900
Reserved					
Value descriptors	20000… 32000	R, P	Description of values (e.g. type, unit, multiplier, range)	CH1	20000
				CH2	20300
				…	
				CH40	31700
Reserved					

4.4.1 RX Values

20 registers are reserved for each channel. When the gateway receives a data telegram from a device that is assigned to a channel, it uses the database of supported devices to convert the data contents into *RX Values*. The received data fields are stored in registers from V1 up. The number of data fields depends on the type of EnOcean device, which is specified during the commissioning phase. For devices that use the EEP protocol, the device type is given by RORG, FUNC and TYPE (see [1]).

$$\text{Address} = \text{BaseAddress} + \text{Offset}$$
$$\text{BaseAddress} = 0 + 20 \cdot (\text{ChannelNumber} - 1)$$

Tab. 4.4 RX Values

Group	Offset	Access	Name	Description	Value range
Values	0	R	V1	Value 1	Depends on device type
	1	R	V2	Value 2	
	2	R	V3	Value 3	
	3	R	V4	Value 4	
	…	R			
	14	R	V15	Value 15	
Status	15	R	Telegram counter	Number of received telegrams	0…65535 (overflows to zero)
	16	R	Telegram age	Time elapsed since the last telegram	0…65000 s (65535: no telegram, 65001: range exceeded)
	17	R	Signal	Signal strength (RSSI value measured by the EnOcean transceiver)	0…–255 dBm (decimal value without minus)
	18	R	Reserved		
	19	R	Error	Enumeration of channel state and error codes	See Tab. 4.5

Tab. 4.5 RX Values – Error register

Error register	
Value	**Meaning**
0 – OK	Telegram OK, data has been stored in Values
1 – ASSIGNED	Device assigned, waiting for the first data telegram
3 – NOT SUPPORTED	Device assigned, unsupported device type, data cannot be converted to Values
7 – TIMEOUT	120 minutes without a telegram received
255 – FREE	The channel is not configured, no device assigned

4.4.2 TX Values

20 registers are reserved for each channel. When the send condition is true, the gateway uses the database of supported devices to convert the *TX Values* into the raw data contents and sends a data telegram. The data fields to send are stored in registers from V1 up. The number of data fields depends on the type of EnOcean device, which is specified during the commissioning phase. For devices that use the EEP protocol, the device type is given by RORG, FUNC and TYPE (see [2]).

The registers can be set individually by the function 6 or all at once by function 16. It is possible to set more channels using a single request, the starting address must be the base address of the first channel being written, unused and readonly registers can be set to any value in the request.

$$\text{Address} = \text{BaseAddress} + \text{Offset}$$
$$\text{BaseAddress} = 0 + 20 \cdot (\text{ChannelNumber} - 1)$$

Tab. 4.6 TX Values

Group	Offset	Access	Name	Description	Value range
Values	0	R/W	V1	Value 1	Depends on device type
	1	R/W	V2	Value 2	
	2	R/W	V3	Value 3	
	3	R/W	V4	Value 4	
	…	R/W			
	14	R/W	V15	Value 15	
Status	15	R	Telegram counter	Number of sent telegrams	0…65535 (overflows to zero)
	16	R	Telegram age	Time elapsed since the last telegram	0…65000 s (65535: no telegram, 65001: range exceeded)
	17	R/W	Send option	Data transmission control	See Tab. 4.7
	18	R	Reserved		
	19	R	Reserved		

Tab. 4.7 TX Values – Send option register

Send option register	
Value	**Meaning**
0 – NotChange	Writing this value has no effect, the send option will not change
1 – None (default)	Transmitting disabled
2 – SendNow	Transmit once immediately, the send option will not change
3 – OnReceived	Automatic response when telegram received from the assigned device (default for A5-20-01, etc.)
4…10	Reserved
11 – OnWriteV1	Transmit when register V1 written[1)]
…	
25 – OnWriteV15	Transmit when register V15 written
26 – OnWriteAny	Transmit when any register V1…V15 written

1) When using function 16, the telegram is sent after all registers updated.

4.4.3 RX Raw Data

20 registers are reserved for each channel. These registers contain the payload bytes of the last received telegram. The length of the data varies depending on the telegram type, which is identified by the RORG byte (see [1]>Ch.3). In most cases, it is not necessary to use these registers and *Value* registers should be used instead.

$$\text{Address} = \text{BaseAddress} + \text{Offset}$$
$$\text{BaseAddress} = 2000 + 20 \cdot (\text{ChannelNumber} - 1)$$

Tab. 4.8 Raw data – RPS, 1BS

Offset	Access	Name	Description	Value range
0	R	RORG	RORG byte	0xF6: RPS, 0xD5: 1BS
1	R	DB_0	Data byte 0	0…0xFF
2	R	Status	ERP1 Status byte	Bits 3…0: Repeater count
3…19		Reserved		

Tab. 4.9 Raw data – 4BS

Offset	Access	Name	Description	Value range
0	R	RORG	RORG byte	0xA5: 4BS
1	R	DB_3	Data byte 3	0…0xFF
2	R	DB_2	Data byte 2	0…0xFF
3	R	DB_1	Data byte 1	0…0xFF
4	R	DB_0	Data byte 0	0…0xFF
5	R	Status	ERP1 Status byte	Bits 3…0: Repeater count
6…19		Reserved		

Tab. 4.10 Raw data – VLD, MSC

Offset	Type	Access	Name	Description	Value range
0	UINT8	R	RORG	RORG byte	0xD2: VLD, 0xD1: MSC
1	UINT8	R	Data length (N)	Number of bytes in the Data array	1…14
2…8	UINT8[14]	R	Data	Data bytes DB_(N–1)…DB_0	HI: DB_(N–1); LO: DB_(N–2)[1)]
					…
					HI: DB_3; LO: DB_2
					HI: DB_1; LO: DB_0
9	UINT8	R	Status	ERP1 Status byte	Bits 3…0: Repeater count
10…19			Reserved		

1) HI is the high byte of the register (bits 15…8), LO is the low byte of the register (bits 7…0).

4.4.4 Channel Config

10 registers are reserved for each channel. Sender ID identifies the EnOcean device that the channel is listening to. The EEP identifies the type of the device and is needed for conversion of raw data to *Values*.

The registers can be set individually by the function 6 or all at once by function 16. It is possible to set more channels using a single request, the starting address must be the base address of the first channel being written, unused and readonly registers can be set to any value in the request.

$$\text{Address} = \text{BaseAddress} + \text{Offset}$$
$$\text{BaseAddress} = 4000 + 10 \cdot (\text{ChannelNumber} - 1)$$

Tab. 4.11 Channel config

Group	Offset		Access	Name	Description	Value range	Default
ID	0	Bits 15…8	R, P	ID3	Sender ID_3 (MSB)	0…0xFF	0xFF
		Bits 7…0	R, P	ID2	Sender ID_2	0…0xFF	0xFF
	1	Bits 15…8	R, P	ID1	Sender ID_1	0…0xFF	0xFF
		Bits 7…0	R, P	ID0	Sender ID_0 (LSB)	0…0xFF	0xFF
EEP	2		R, P	RORG		0…0xFF	0x00
	3		R, P	FUNC		0…0xFF	0x00
	4		R, P	TYPE		0…0xFF	0x00
	5…9			Reserved			

4.4.5 Channel Labels

100 registers are reserved for each channel. The channel label is designed to store text information, it can be a user friendly name that helps to identify the data or EnOcean device. The maximum size of Label is 126 bytes. The UTF-8 encoding is used because it is compatible with ASCII encoding, ASCII characters (Unicode 0x0000 to 0x007F) are encoded into one byte in UTF-8 (0x00 to 0x7F).

$$\text{Address} = \text{BaseAddress} + \text{Offset}$$
$$\text{BaseAddress} = 10000 + 100 \cdot (\text{ChannelNumber} - 1)$$

Tab. 4.12 Channel labels

Offset	Type	Access	Name	Description	Value range	Default
0	UINT8	R, P	Encoding	Character encoding of Label	0: UTF-8	0: UTF-8
1	UINT8	R, P	Label length (N)	Number of bytes in the Label array	0…126	
2…64	UINT8[126]	R, P	Label	Array of bytes b0…b(N−1)	HI: b0; LO: b1 HI: b2; LO: b3 … HI: b(N−2); LO: b(N−1)	1)
65…99			Reserved			

1) Default label is "{EEP} {Title} (SenderID)", e.g. "A5-02-05 Temperature Sensors, Temperature Sensor Range 0°C to +40°C (05-0C-54-74)"

4.4.6 Value Descriptors

$$\text{Address} = \text{BaseAddress} + \text{Offset}$$
$$\text{BaseAddress} = 20000 + 300 \cdot (\text{ChannelNumber} - 1)$$

Tab. 4.13 Value descriptors

Group		Offset	Access	Name	Description	Value range
RX	Value 1	0	R	Value ID	e.g. 3704 -> Channel 37, Value 4	100…4000
		1	R	Type	Type of the value	0:none; 1:UINT16; 2:INT16; 3:UINT32_MSB 4:UINT32_LSB
		2	R	Min	Minimum valid value	
		3	R	Max	Maximum valid value	
		4	R	Step	Conversion of register value: Value = Register · Step	1:1; 2:0.1; 3:0.01; 4:0.001;
		5	R	Unit	Unit of measurement	See below
		6…9			Reserved	
	Value 2	10…19				
	…					
	Value 15	140…149				
TX	Value 1	150	R	Value ID	e.g. 13704 -> TX, Channel 37, Value 4	10100…14000
		…				
	Value 2	160…169				
	…					
	Value 15	290…299				

Type 0:none means that the corresponding *Value* register is unused.

Conversion example for temperature -10…30 °C stored in register RX.CH37.V1. Descriptors are as follows:

Group		Offset	Access	Name	Dec (s16)	Hex (u16)	Interpreted value
RX	Value 1	0	R	Value ID	3701	0x0E75	
		1	R	Type	1	0x0001	numeric
		2	R	Min	-100	0xFF9C	-10°C
		3	R	Max	300	0x012C	30°C
		4	R	Step	2	0x0002	0.1
		5	R	Unit	62	0x005F	°C
		6…9					

Value of register is interpreted as follows:

Dec (s16)	Hex (u16)	Interpreted value
227	0x00E3	Register · Step = 22.7°C
-51	0xFFCD	Register · Step = -5.1°C

Codes for units are listed in ANNEX B (compatible with BACnetEngineeringUnits enumeration).

5 Configuration

The gateway has a built-in web application (*Web UI*) that serves to configure it using a web browser. Alternatively, you can use the *EO-BAC Tool* application, in that case, a USB connection is required.

5.1 Device Discovery

This chapter describes different ways to access the *Web UI*.

Connect the gateway to your network (Ethernet connector).

The DHCP client is enabled by default, so the IP address and other network parameters should be obtained from the DHCP server automatically.

If DHCP fails, e.g. there is no DHCP server, the last saved network configuration is used. The default network configuration is:

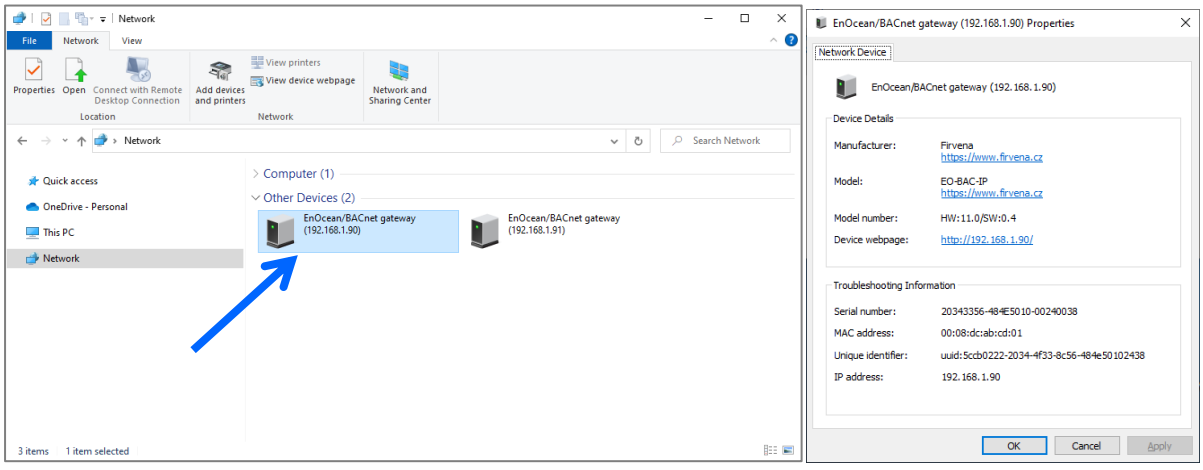
Parameter	Value
IP address	192.168.1.90
Subnet mask	255.255.255.0
Default gateway	192.168.1.1
DNS server	8.8.8.8

The methods of discovering the gateway are as follows:

Discovery using UPnP

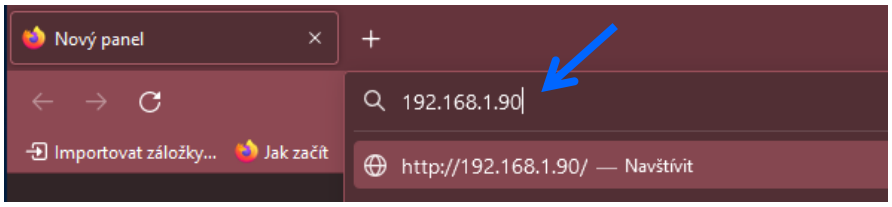
This method is recommended when using the Windows operating system and UPnP messages are allowed in the network to which the gateway is connected.

Open the *Network* item in the *File Explorer*:



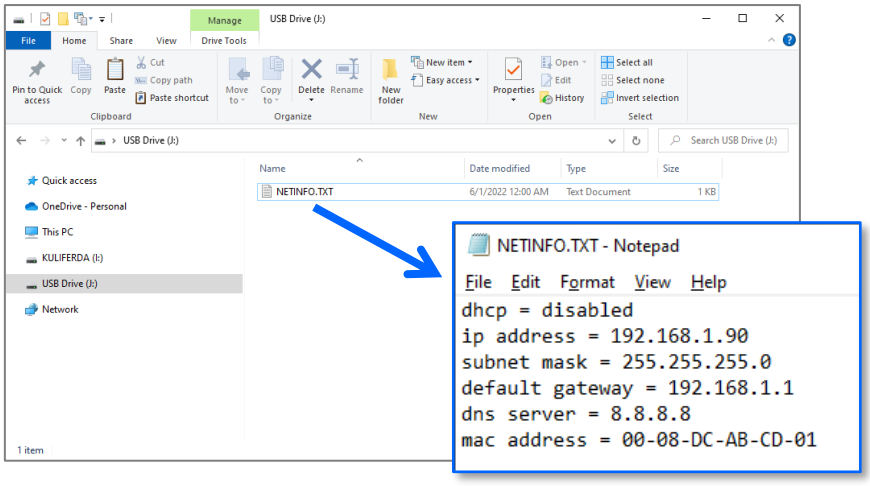
The gateway is under the group *"Other Devices"*. The description and IP address of the gateway are shown. Double click on the device item to open the *Web UI*. [Right click > Properties] will display additional information.

If the gateway is not shown, try to refresh the list: [Click on the list > press F5 key] or [Right click > Refresh].



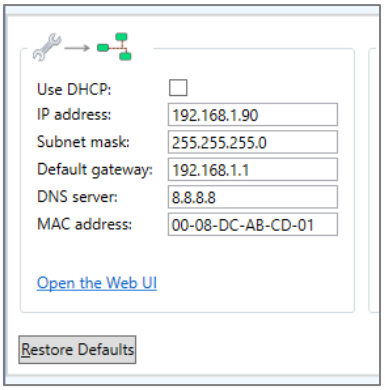
USB MSC

In the *USB MSC* mode (see Ch. 2), the gateway presents itself as an external USB drive. Connect the gateway to your computer using USB. Find a file named NETINFO.TXT in the root directory and open it. The file contains the current IP address. To access the *Web UI*, enter the IP address to the address bar of your web browser.



Using EO-BAC Tool

In the *USB HID* mode (see Ch. 2), the gateway presents itself as a Custom HID USB device. Connect the gateway to your computer using USB and use the *EO-BAC Tool*.



DHCP server

If you have access to the local DHCP server (usually through the configuration interface of your router), the IP address should be in DHCP client list, look for the host name “EO-MOD-IP”. To access the *Web UI*, enter the IP address to the address bar of your web browser.

5.2 Web UI

This chapter is a brief guide to the use of the *Web UI* application.

The *Web UI* is used to configure the gateway. The main purpose of the application is to manage EnOcean devices connected to the gateway. The application is also a useful verification tool whereby you can evaluate whether your system works well. It allows to see the states, measured quantities, communication intervals or signal strength of the connected EnOcean devices.

5.2.1 Main Page Overview

1. Dashboard – overview of configured channels
2. Toggles dashboard view:
    - Detailed – more information, only configured channels are displayed
    - Condensed – less information, both configured and empty channels are displayed
3. Menu items:
    - Add new device
    - Settings
    - Telegram log
4. Login button:
    - Locked – configuration locked and cannot be changed, only dashboard is accessible
    - Unlocked – configuration of the gateway can be changed
5. Gateway identification

The main page is accessible without a password. The default password is “123”, you can change it in [Settings > Change Password].

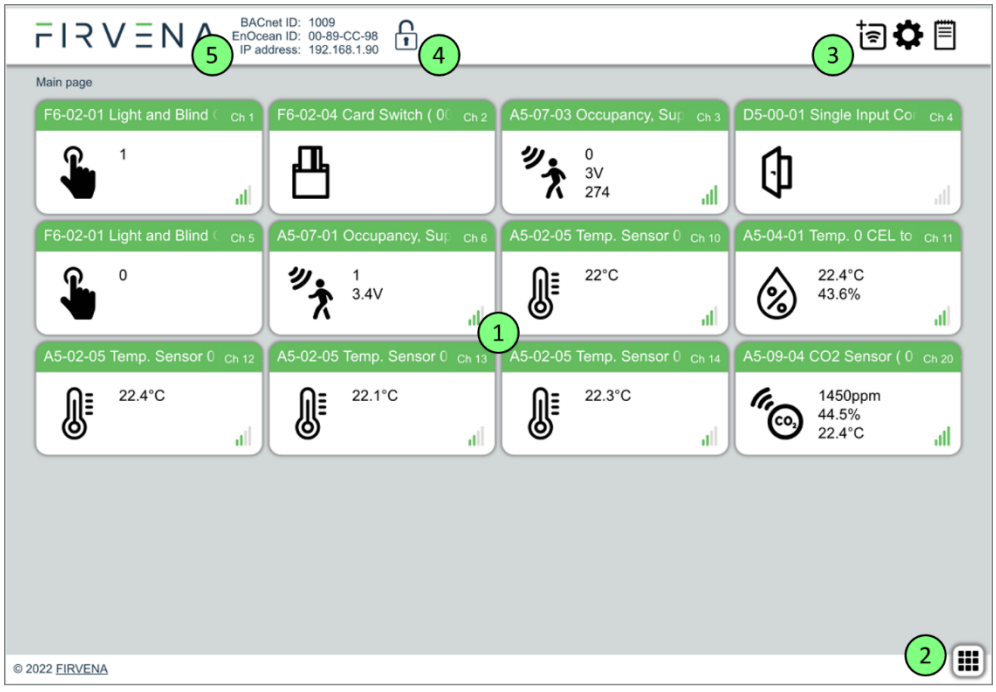


Fig. 5.1 Main page overview

5.2.2 Assigning EnOcean elements – unidirectional

This teach-in procedure is only applicable for unidirectional communication when the gateway only receives data.

Let's have a humidity sensor (A5-04-01) we want to assign to the channel 1. The procedure is as follows (see Fig. 5.2):

1. Click the "Add new" icon, a dialog box appears.
2. Push the pairing button to transmit a teach-in telegram.
3. The received telegram is displayed in the dialog box.
4. Select the channel number 1.
5. Click "Save" to confirm changes
6. Now the sensor is assigned to channel 1 and its data is available through the Modbus interface.

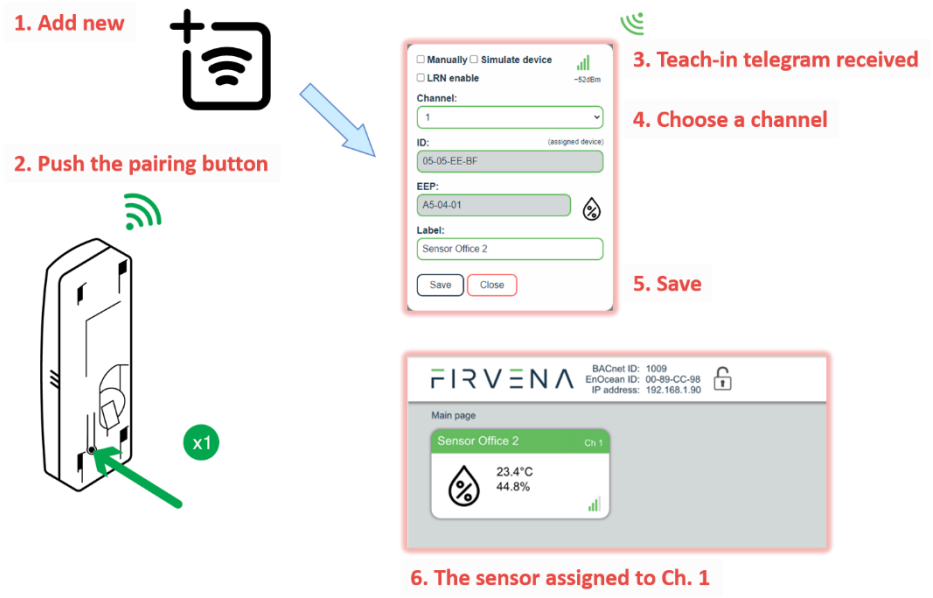


Fig. 5.2 Adding elements – teach-in procedure

Optionally, the *Label* can be set for the device. The *Label* text is used in the *Channel labels* registers (see Ch. 4.4.5). If the *Label* field is left empty, the default text will be used.

The knowledge of EEP allows the gateway to interpret the received data correctly. Some types of EnOcean devices do not provide the EEP information in their teach-in telegram or even not have a special telegram for teach-in (e.g. buttons and switches). In that case, the EEP must be set manually, it is usually given by a label on the device or a datasheet

Devices can also by assigned manually by entering the ID and EEP – see Fig. 5.3.

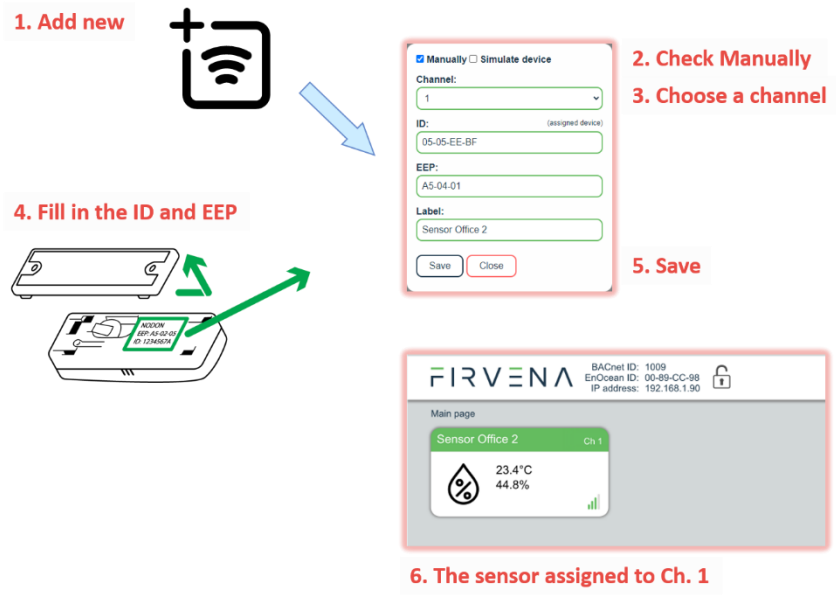
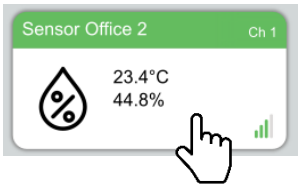


Fig. 5.3 Adding elements – manually

Click on the channel box to view more settings and status information:



5.2.3 Assigning EnOcean elements – bidirectional

This teach-in procedure is applicable for EnOcean devices with bidirectional communication profiles when the gateway is supposed to receive data from the device and also transmit data to the assigned device.

Let's have a valve actuator (A5-20-01) we want to assign to the channel 1. The procedure is as follows (see Fig. 5.4):

1. Click the "Add new" icon, a dialog box appears.
2. Check "LRN enable"
3. Select the channel number 1.

Channel 1 is now in the teach-in mode.

4. Push the pairing button to transmit a teach-in telegram from the actuator.
5. Gateway receives the telegram and sends a teach-in response.
6. The device is automatically saved to the selected channel and its data is available through the Modbus interface.

To change the Label, enter the new text and click "Save".

Notes:

- *The device being assigned usually signals successful teach-in, e.g. by flashing LED. If it signals an error, the teach-in must be repeated.*

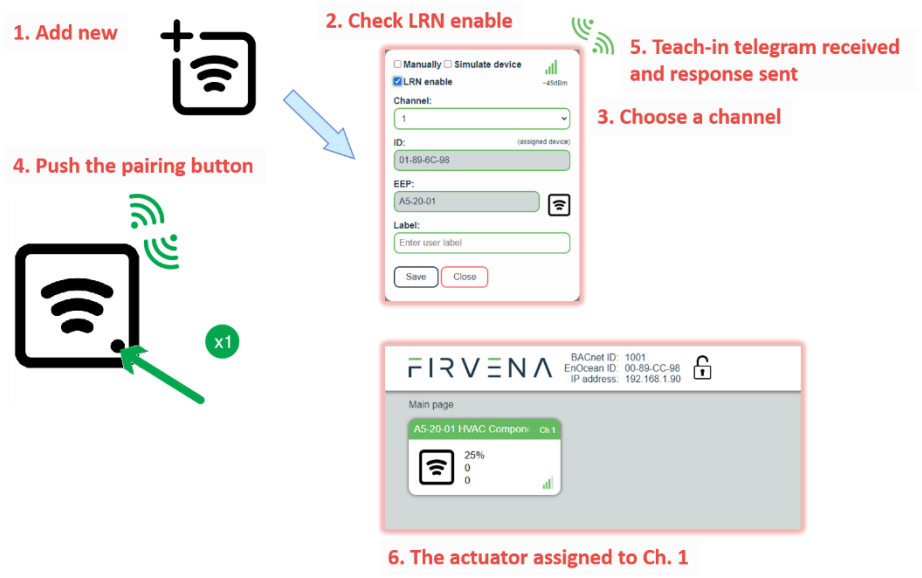


Fig. 5.4 Adding elements – bidirectional teach-in procedure

5.2.4 Assigning Smart ACK devices

To assign a Smart ACK device to the gateway, the procedure is the same as for bidirectional profiles (5.2.3). Repeaters are not supported, there must be a direct connection between the gateway and the Smart ACK device.

5.2.5 Definition of Virtual Device

The gateway can be used to simulate a real EnOcean device. This function is necessary when controlling actuators that do not implement any bidirectional profile, typically some types of relay switches.

For example, we want to control a relay switch that supports reception of a Door/Window Contact D5-00-01. The procedure is as follows.

First, define a virtual device of type D5-00-01, for example at channel 1 (see Fig. 5.5):

1. Click the “Add new” icon, a dialog box appears.
2. Check “Simulate device”
3. Select a channel number (channel 1)
4. Select the type of device (D5-00-01)
5. Click “Save” to confirm changes
6. The device is saved to the selected channel and its data is available through the Modbus interface.

Second, pair the virtual device with the relay switch:

1. Click on the channel 1 box to see channel details (Fig. 5.6)
2. Put the relay switch to the pairing mode (follow the procedure given by the manufacturer of the switch)
3. In channel details click on the “Send LRN” button
4. The gateway will transmit a teach-in telegram of the virtual device
5. The switch will receive the teach-in telegram and save the virtual device

Third, test the connection:

1. Navigate to [Edit channel > Values]
2. Set the data to be transmitted (0:closed or 1:open)
3. Use “Send Now” to transmit a data telegram

Notes:

- *Each virtual device must have a unique ID, this is given by the MyID setting.*
- *To enable receiving the actual state of the switch, assign it to a different channel.*

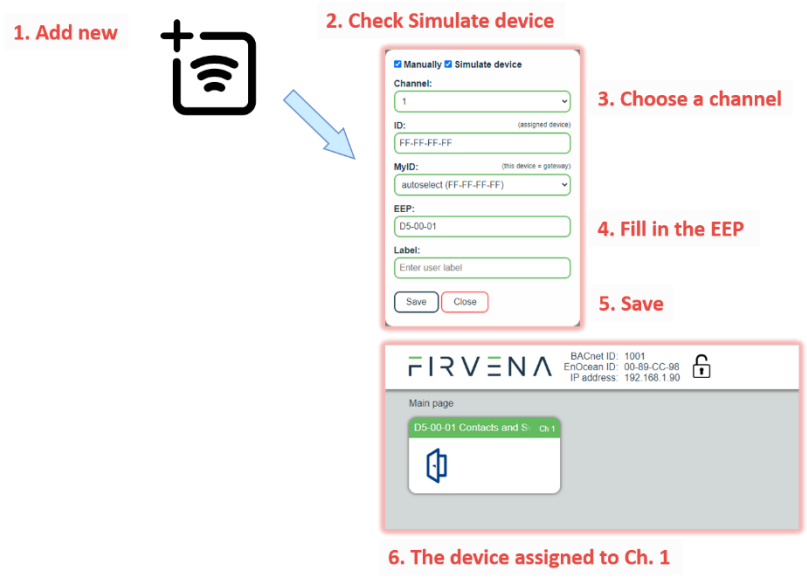


Fig. 5.5 Adding elements – virtual device

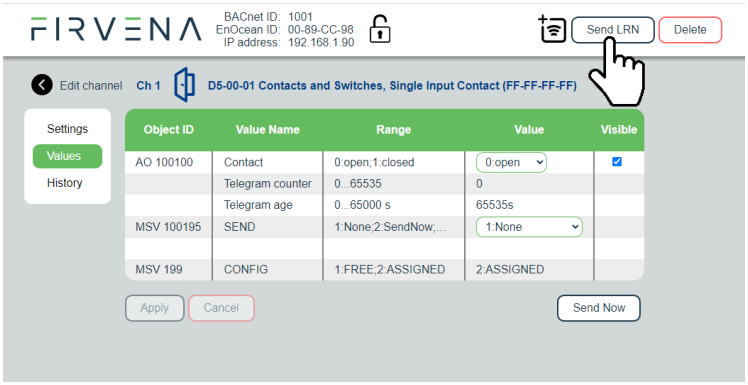
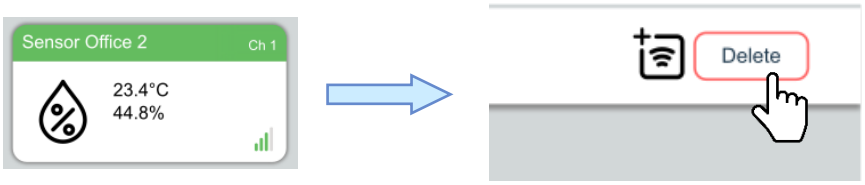


Fig. 5.6 Virtual device – send teach-in telegram

5.2.6 Removing EnOcean elements

1. Click on a channel box.
2. Click on the “Delete” button.



To remove all elements, use [Settings > Factory Reset].

5.2.7 Backup and Restore

The configuration of the gateway can be exported to a file for later recovery or reuse. The backup file is compatible with the file used by *EO-BAC Tool* application.

To back up the configuration:

1. Navigate to [Settings > Backup and Restore].
2. Click on Download.
3. A file named config.json is downloaded.

To restore the configuration:

1. Navigate to [Settings > Backup and Restore].
2. Select a file or drag it to the dashed rectangle.
3. Click on Upload.

Notes:

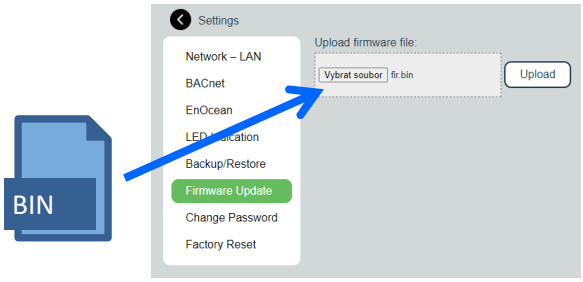
- *Only channels are restored, the "Gateway Settings" are preserved.*

6 Firmware Update

The firmware is constantly being improved and extended to support new features and EnOcean devices. The actual version of the firmware is available for download on the FIRVENA website.

6.1 Over Network

Open the *Web UI* and navigate to [Settings > Firmware Update]:



Select a file or drag it to the dashed rectangle and click on “Upload”.

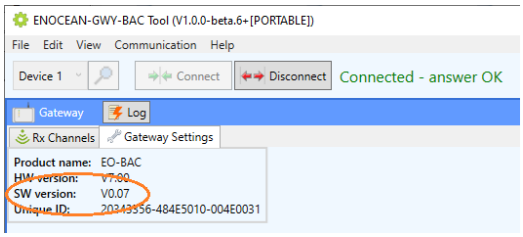
When the uploading is done, the gateway reboots and the connection is lost:

Gateway checks the file and overwrites the current firmware with the new one.

The connection should be resumed within several minutes.

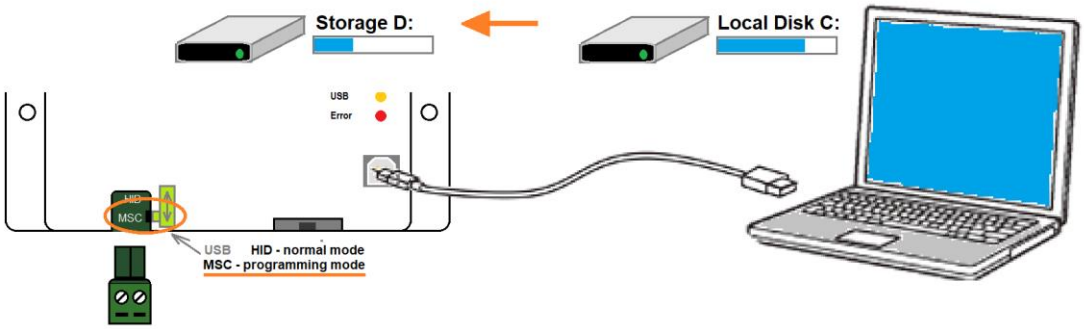
6.2 Over USB

The number of firmware version can be determined using the *EO-BAC Tool*:

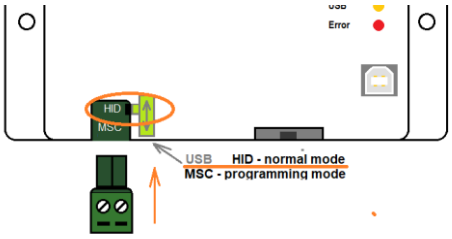


To update firmware in the gateway:

- Unplug the POWER connector
- Set the left switch to the “MSC” position
- Connect the gateway to a computer using a USB cable with type B connector
- The device appears as an external disk, copy the new firmware file to the disk



- Set the left switch back to the “HID” position, the gateway reboots
- Now, the gateway checks the file and overwrites the current firmware with the new one
- The result is indicated by LEDs
- Disconnect the USB cable and plug the POWER connector back



References

[1] EnOcean Technical Specifications (https://www.enocean-alliance.org/specifications/)

[2] EnOcean Equipment Profiles (https://www.enocean-alliance.org/wp-content/uploads/2020/07/EnOcean-Equipment-Profiles-3-1.pdf)

[3] Communication telegrams defined in EnOcean equipment profiles (http://tools.enocean-alliance.org/EEPViewer/)

[4] Smart Acknowledge – Bidirectional communication with energy harvesting devices (https://www.enocean-alliance.org/wp-content/uploads/2020/04/SmartAcknowledge_Specification_v1.7.pdf)

[5] EnOcean Unique Radio Identifier – EURID Specification (https://www.enocean-alliance.org/wp-content/uploads/2021/03/EURID-v1.2.pdf)

[6] Modbus Protocol Specifications and Implementation Guides: https://modbus.org/specs.php

[7] Modbus Application Protocol Specification: https://modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf

[8] Modbus TCP/IP Implementation Guide: https://modbus.org/docs/Modbus_Messaging_Implementation_Guide_V1_0b.pdf

Revision History

Date	Version	Author	Description
2022-04-01	V1.0	J.BAC	Initial release
2022-08-05	V1.1	J.BAC	Added bidirectional communication
2022-11-01	V1.2	J.BAC	Added Ch. 5.2.5 Updated figures Updated ANNEX A

ANNEX A Mapping examples of EnOcean devices

A.1 Basic Examples

RX Values:

Ch.	Device	Register	Value Name	Range	Value	Dec	Hex
1	A5-02-05 	0	Temperature	0…40 °C	22.4°C	224	0x00E0
		15	Telegram counter	0…65535	12	12	0x000C
		16	Telegram age	0…65000 s	252s	252	0x00FC
		17	Signal	0…–255dBm	–49dBm	49	0x0031
		19	Error		0 (OK)		
2	00-00-00	20	None		0		
		…	…		…		
		34	None		0		
		35	Telegram counter	0…65535	0		
		36	Telegram age	0…65000 s	65535		
		37	Signal	0…–255dBm	0		
		39	Error		255 (FREE)		
…							
32	D5-00-01 	620	Contact	0:open, 1:closed	1		
		635	Telegram counter	0…65535	50		
		636	Telegram age	0…65000 s	10s		
		637	Signal	0…–255dBm	–58dBm		
		639	Error		0 (OK)		
…							
39	00-00-00						
40	A5-09-04 	780	Humidity	0…100 %	45%	450	0x01C2
		781	Concentration	0…2550 ppm	1451ppm	1451	0x05AB
		782	Temperature	0…51 °C	23.2°C	232	0x00E8
		783	H-Sensor	0:Not available, 1:Available	1		
		784	T-Sensor	0:Not available, 1:Available	1		
		795	Telegram counter	0…65535	50		
		796	Telegram age	0…65000 s	10s		
		797	Signal	0…–255dBm	–52dBm		
		7999	Error		0 (OK)		

A.2 RPS Buttons and Switches

The following example applies to F6-02-01, F6-02-02, F6-02-03 and F6-02-04

Ch.	Device	Register	Value Name	Range	Value	Dec	Hex
1	F6-02-02 	0	BI	0:released, 1:pressed	1		
		1	B0	0:released, 1:pressed	0		
		2	AI	0:released, 1:pressed	0		
		3	A0	0:released, 1:pressed	0		
		4	Rocker B	-1:null, 0:off, 1:on	1		
		5	Rocker A	-1:null, 0:off, 1:on	0		
2	00-00-00						
…							

V4 and V5 remember the rocker state for channel A and B, this is out of the EEP definition. Rocker B goes 1:on when BI was pressed, Rocker B goes 0:off when B0 was pressed. When no telegram has been received yet, rocker has the initial value -1:null.

A.3 A5-20-01 HVAC Components, Battery Powered Actuator

The actuator wakes up periodically, transmits the actual value and waits for a response with a new setpoint, which must be sent within 1 second. The response is built from *TX Values* registers. The response also contains other settings, e.g. Set point type selection, Set point inverse, Summer mode, Service mode.

Direction RX (from actuator):

Ch.	Device	Register	Value Name	Range	Value	Dec	Hex
1	A5-20-01 	0	Current Value	0...100 %	25%	25	0x0019
		1	Service On	1:on	0	0	0x0000
		2	Energy input enabled	1:true	0	0	0x0000
		3	Energy Storage	1:true	1	1	0x0001
		4	Battery capacity	0:true	1	1	0x0001
		5	Contact, cover open	1:true	0	0	0x0000
		6	Failure temperature sensor, out off range	1:true	0	0	0x0000
		7	Detection, window open	1:true	0	0	0x0000
		8	Actuator obstructed	1:true	0	0	0x0000
		9	Temperature	0...40 °C	22.59°C	226	0x00E2
2	00-00-00						
…							

Direction TX (to actuator):

Ch.	Device	Register	Value Name	Range	Value	Dec	Hex
1	A5-20-01	1000	Valve position or Temperature Setpoint	0...100 %	25%	25	0x0019
		1001	Temperature from RCU	0...40 °C	0°C	0	0x0000
		1002	Summer mode	0...1	0	0	0x0000
		1003	Setpoint selection	0...1	0	0	0x0000
		1004	Set point inverse	0...1	0	0	0x0000
			SEND		3 (OnReceive)		
2	00-00-00						
...							

A.4 D2-01-12 Electronic switches and dimmers with Energy Measurement and Local Control, Type 0x12

The group of devices D2-01-XX uses several telegrams (commands), each type supports only certain commands and functions, e.g. type 0x02 has one dimmable output, type 0x12 has two relay outputs without dimming function, this is given by the EEP specification.

This actuator has two output channels, the example shows switching on of the second channel. TX data are first prepared by writing into the TX data objects, then the control telegram (CMD1) is sent by writing the Send option 2 (SendNow). Actuator returns status message (CMD4).

The Send option can also be configured so that the gateway sends when Output value is written (14:OnWrite_V3).

Direction RX (from actuator):

Ch.	Device	Register	Value Name	Range	Value	Dec	Hex
1	D2-01-12 	0	Command ID	4...4	4	4	0x0004
		1	I/O channel	0...31	1	1	0x0001
		2	Output value	0...100 %	100%	100	0x0064
2	00-00-00						
...							

Direction TX (to actuator):

Ch.	Device	Register	Value Name	Range	Value	Dec	Hex
1	D2-01-12	1000	Command ID	1...1	1	1	0x0001
		1001	Dim value	0...4	0	0	0x0000
		1002	I/O channel	0...31	1	1	0x0001
		1003	Output value	0...100 %	100%	100	0x0064
		1017	SEND		2 (SendNow)	2	0x0002
2	00-00-00						
...							

A.5 D2-11-07 Bidirectional Room Operating Panel (Smart ACK)

What is Smart ACK?

EnOcean sensors are in sleep mode most of the time to reduce power consumption, so they cannot receive any telegram. The Smart ACK protocol enables bidirectional communication with energy self-sufficient devices. For example, Room Operating Panels D2-11-XX utilize the Smart ACK communication to receive data, which is used to show symbols on the display or override some parameters.

The Smart ACK protocol is described in [4]. When a message is sent to a Smart ACK Sensor, a device called “Post Master” stores it in a “Mailbox” until the sensor is ready to receive telegrams. When the sensor wakes up, it checks the Mailbox. The Post Master sends the message buffered in the Mailbox or Mailbox Empty message if the Mailbox is empty. The sensor receives the response from Post Master and returns to sleep mode. The Mailbox is established in Post Master during teach-in process.

The gateway does not support repeaters, there must be a direct connection between the gateway and the Smart ACK device, i.e. Post Master and Mailbox are located in the gateway.

Example with SR06 LCD Thermocon

Direction RX (from sensor):

The sensor sends two types of messages, ID 0 or ID 2. When Message ID is 0, only Set Point Type is valid, other values should be ignored.

Communication is initiated by the sensor on heartbeat (default 1000 s), change of measured value or button press (parameter change), which is indicated by TelegramType.

Ch.	Device	Register	Value Name	Range	Value	Dec	Hex
1	D2-11-07 	0	Set Setpoint type	0:Temperature correction;1:Temperature setpoint	1	1	0x0001
		1	Telegram Type	0:Heartbeat;1:Change of temperature or humidity value;2:User caused parameter change	2	2	0x0002
		2	Message ID	0:ID-0;2:ID-2	2	2	0x0002
		3	Temperature	0...40 °C	23.84°C	238	0x00EE
		4	Humidity	0...100 %	0%	0	0x0000
		5	Setpoint offset	0...255	170	170	0x00AA
		6	Basesetpoint	15...30 °C	21°C	21	0x0015

		7	Valid temperature correction	1:-1...1K;2:-2...2K;3:-3...3K;4:-4...4K;5:-5...5K;6:-6...6K;7:-7...7K;8:-8...8K;9:-9...9K;10:-10...10K	3	3	0x0003
		8	Fan speed	0:Auto;1:Speed 0;2:Speed 1;3:Speed 2;4:Speed 3;7:Not available	7	7	0x0007
		9	Occupancy state	0:State Unoccupied;1:State Occupied	0	0	0x0000
2	00-00-00						
…							

Direction TX (to sensor):

The gateway responds with message ID 1. Settings are changed by writing these values and SendOption = 2:SendNow, changes will apply next time the sensor wakes up. Without a response, the sensor uses the last settings.

SetPointType, TemperatureCorrection, BaseSetpoint and ValidTemperatureCorrection must be mirrored from RX data if no change is required.

Ch.	Device	Register	Value Name	Range	Value	Dec	Hex
1	D2-11-07	1000	Set Setpoint type	0:Temperature correction;1:Temperature setpoint	1	1	0x0001
		1001	Display heating symbol	0:Heating symbol off;1:Heating symbol on	0	0	0x0000
		1002	Display cooling symbol	0:Cooling symbol off;1:Cooling symbol on	0	0	0x0000
		1003	Display window open symbol	0:Window open symbol off;1:Window open symbol on	1	1	0x0001
		1004	Message ID	1:ID-1	1	1	0x0001
		1005	Temperature correction	0...255	128	128	0x0080
		1006	Basesetpoint	15...30 °C	21°C	21	0x0015
		1007	Valid temperature correction	1:-1...1K;2:-2...2K;3:-3...3K;4:-4...4K;5:-5...5K;6:-6...6K;7:-7...7K;8:-8...8K;9:-9...9K;10:-10...10K	3	3	0x0003
		1008	Fan speed	0:Auto;1:Speed 0;2:Speed 1;3:Speed 2;4:Speed 3;7:Not available	0	0	0x0000
		1009	Occupancy state	0:State Unoccupied;1:State Occupied	0	0	0x0000
		1017	SEND		2 (SendNow)	2	0x0002
2	00-00-00						
…							

A.6 D2-15-00 People Activity Sensor

Ch.	Device	Register	Value Name	Range	Value	Dec	Hex
1	D2-15-00 	0	Presence	0:Present;1:Not Present;2:Not detectable;3:Presence Detector error	0	0	0x0000
		1	Energy Storage Status	0:High;1:Medium;2:Low;3:Critical	0	0	0x0000
		2	Pir Update Rate	1...16 s	1s	1	0x0001
		3	Pir Counter	0...65535	7568	7568	0x1D90
		4	Activity	0...100 %	52%	52	0x0034
2	00-00-00						
...							

The Activity is computed by gateway based on two subsequent values of the Pir Counter. When the Pir Update Rate is 1s and the sensor transmits data every 2 minutes, 100% corresponds to the Pir Counter increment of 120.

$$Activity[\%] = \frac{(PIRCounterCurrent - PIRCounterLast) \cdot PIRUpdateRate[s]}{TelegramAgeLast[s] - TelegramAgeCurrent[s]}$$

ANNEX B Enumeration of Units

```
/*000*/"m2",
/*001*/"ft2",
/*002*/"mA",
/*003*/"A",
/*004*/"Ohm",
/*005*/"V",
/*006*/"kV",
/*007*/"MV",
/*008*/"VA",
/*009*/"kVA",
/*010*/"MVA",
/*011*/"var",
/*012*/"kvar",
/*013*/"Mvar",
/*014*/"°",
/*015*/"[?]",
/*016*/"J",
/*017*/"kJ",
/*018*/"Wh",
/*019*/"kWh",
/*020*/"Btu",
/*021*/"UK",
/*022*/"th",
/*023*/"J/kg",
/*024*/"[?]",
/*025*/"cph",
/*026*/"cpm",
/*027*/"Hz",
/*028*/"g/kg",
/*029*/"% rh",
/*030*/"mm",
/*031*/"mm",
/*032*/"Inch",
/*033*/"[?]",
/*034*/"w/f2",
/*035*/"[?]",
/*036*/"lm",
/*037*/"lx",
/*038*/"fc",
/*039*/"kg",
/*040*/"[?]",
/*041*/"t",
/*042*/"kg/s",
/*043*/"kg/min",
/*044*/"kg/h",
/*045*/"[?]",
/*046*/"[?]",
/*047*/"W",
/*048*/"kW",
/*049*/"MW",
/*050*/"Btu/h",
/*051*/"hp",
/*052*/"[?]",
/*053*/"Pa",
/*054*/"kPa",
/*055*/"Bar",
/*056*/"[?]",
/*057*/"[?]",
/*058*/"[?]",
/*059*/"[?]",
/*060*/"[?]",
/*061*/"[?]",
/*062*/"°C",
/*063*/"K",
/*064*/"°F",
/*065*/"[?]",
/*066*/"[?]",
/*067*/"Year",
/*068*/"Month",
/*069*/"week",
/*070*/"Day",
/*071*/"h",
/*072*/"m",
/*073*/"s",
/*074*/"m/s",
/*075*/"km/h",
/*076*/"ft/s",
/*077*/"ft/min",
/*078*/"[?]",
/*079*/"[?]",
/*080*/"[?]",
/*081*/"[?]",
/*082*/"l",
/*083*/"[?]",
/*084*/"[?]",
/*085*/"m3/s",
/*086*/"[?]",
/*087*/"l/s",
/*088*/"l/min",
/*089*/"[?]",
/*090*/"°",
/*091*/"°C/h",
/*092*/"°C/min",
/*093*/"[?]",
/*094*/"[?]",
/*095*/"",
/*096*/"ppm",
/*097*/"ppb",
/*98*/"%",
/*99*/"%/s",
/*100*/"[?]",
/*101*/"[?]",
/*102*/"[?]",
/*103*/"rad",
/*104*/"[?]",
/*105*/"[?]",
/*106*/"[?]",
/*107*/"[?]",
/*108*/"[?]",
/*109*/"[?]",
/*110*/"[?]",
/*111*/"[?]",
/*112*/"[?]",
/*113*/"[?]",
/*114*/"[?]",
/*115*/"inch2",
/*116*/"cm2",
/*117*/"[?]",
/*118*/"cm",
/*119*/"[?]",
/*120*/"[?]",
/*121*/"°K",
/*122*/"kOhm",
/*123*/"MOhm",
/*124*/"mV",
/*125*/"kJ/kg",
/*126*/"MJ",
/*127*/"[?]",
/*128*/"[?]",
/*129*/"kHz",
/*130*/"MHz",
/*131*/"[?]",
/*132*/"mW",
/*133*/"hPa",
/*134*/"mBar",
/*135*/"m3/h",
/*136*/"l/h",
/*137*/"[?]",
/*138*/"[?]",
/*139*/"[?]",
/*140*/"[?]",
/*141*/"[?]",
/*142*/"[?]",
/*143*/"[?]",
/*144*/"[?]",
/*145*/"mOhm",
/*146*/"MWh",
/*147*/"kBtu",
/*148*/"[?]",
/*149*/"[?]",
/*150*/"[?]",
/*151*/"[?]",
/*152*/"[?]",
/*153*/"N",
/*154*/"g/s",
/*155*/"g/min",
/*156*/"t/h",
/*157*/"[?]",
/*158*/"[?]",
/*159*/"ms",
/*160*/"Nm",
/*161*/"mm/s",
/*162*/"[?]",
/*163*/"[?]",
/*164*/"[?]",
/*165*/"[?]",
/*166*/"[?]",
/*167*/"[?]",
/*168*/"[?]",
/*169*/"A/m2",
/*170*/"F",
/*171*/"H",
/*172*/"[?]",
/*173*/"S",
/*174*/"S/m",
/*175*/"T",
/*176*/"[?]",
/*177*/"[?]",
/*178*/"Wb",
/*179*/"cd",
/*180*/"cd/m2",
/*181*/"K/h",
/*182*/"K/min",
/*183*/"J/s",
/*184*/"rad/s",
/*185*/"m2/N",
/*186*/"kg/m3",
/*187*/"Ns",
/*188*/"N/m",
/*189*/"W/m/K",
/*190*/"uS",
/*191*/"[?]",
/*192*/"[?]",
/*193*/"km",
/*194*/"um",
/*195*/"g",
/*196*/"mg",
/*197*/"mm",
/*198*/"mm/s",
/*199*/"dB",
/*200*/"dBmV",
/*201*/"dBV",
/*202*/"mS",
/*203*/"W/hr",
/*204*/"kWhr",
/*205*/"[?]",
/*206*/"[?]",
/*207*/"[?]",
/*208*/"g/g",
/*209*/"kg/kg",
/*210*/"g/kg",
/*211*/"mg/g",
/*212*/"mg/kg",
/*213*/"g/ml",
/*214*/"g/l",
/*215*/"mg/l",
/*216*/"ug/l",
/*217*/"g/m3",
/*218*/"mg/m3",
/*219*/"ug/m3",
/*220*/"[?]",
/*221*/"g/cm3",
/*222*/"Bq",
/*223*/"[?]",
/*224*/"[?]",
/*225*/"Gy",
/*226*/"mGy",
/*227*/"[?]",
/*228*/"Sv",
/*229*/"[?]",
/*230*/"[?]",
/*231*/"[?]",
/*232*/"[?]",
/*233*/"[?]",
/*234*/"pH",
/*235*/"[?]",
/*236*/"[?]
```