FIRVENA

LoRaWAN Gateway

LORCOM

User Manual V1.3

English

CONTENTS

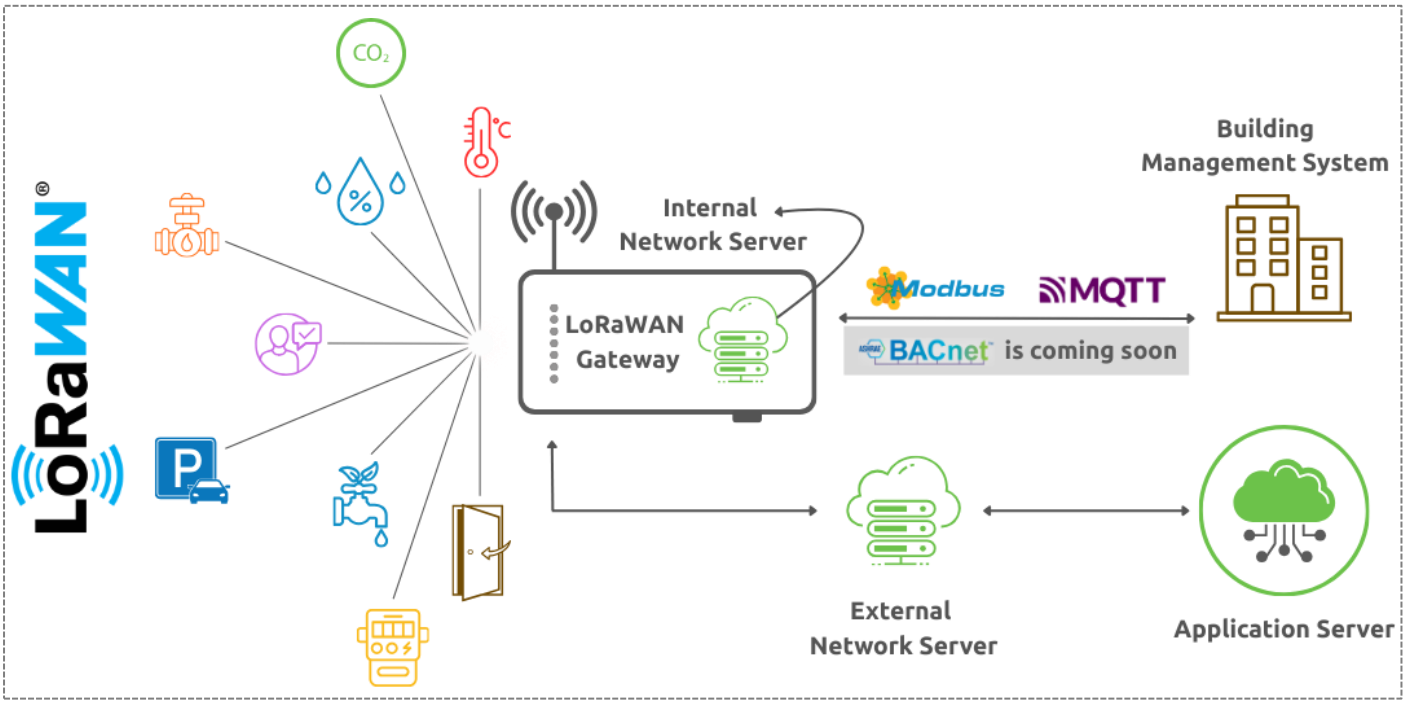
THANK YOU

Thank you for purchasing our product! We believe in your satisfaction with the product that aligns with the company philosophy of the highest care and precision. In case of interesting ideas and concepts, please contact firvena@firvena.cz

www.firvena.com

1 INTRODUCTION

The LORCOM is a LoRaWAN gateway with an integrated *Network server* and built-in integrations to MQTT, Modbus and BACnet. It is designed for indoor use and offers small, compact dimensions. It allows you to extend your building management system with LoRaWAN devices and take advantage of modern, low-power, long-range wireless communication. The gateway also includes Node-RED, which makes it even more flexible. Below is an overview of LORCOM features.



Durable and safe hardware

LORCOM is our own hardware with a powerful and energy-efficient NXP processor with an Arm® Cortex®-A55 core and advanced industrial-grade security. The gateway runs a reliable Linux system based on the Yocto Project.

LoRaWAN chipset Semtech® SX1302

Bidirectional LoRaWAN communication is provided by an 8-channel module based on the new generation Semtech® SX1302 chipset, which offers higher sensitivity and significantly lower power consumption.

Power supply

Power supply can be via PoE (Power over Ethernet) or via an external power supply in the range of 15–30 VDC. Both active and passive PoE are supported.

No licences. No hidden fees.

By purchasing the gateway, you get a full-fledged product with unlimited data points, unlimited LoRaWAN devices and free future upgrades.

Manufactured in EU (CZ)

Each piece is carefully inspected by FIRVENA, this brings exceptional quality.

Webserver

Simple gateway setup using an integrated web user interface, access via a unique name on the local network.

Network server

Thanks to the integrated *Network server (NS)*, the gateway can operate as a standalone unit capable of forwarding decoded data to the BMS system via communication protocols such as MODBUS and MQTT. It also offers easy integration with IoT platforms such as AWS IoT Core, Azure IoT Hub or

Google Cloud Pub/Sub. The gateway can also work as a *Packet forwarder* connected to any external NS (e.g. The Things Stack, ChirpStack or AWS), which ensures efficient device management and seamless data integration into end applications.

Support for new devices

End-device support is provided through flexible configuration files, which in most cases eliminates the need for firmware modification. Users can use pre-built configurations or easily create and customize their own to meet specific application requirements. The solution is fully compatible with the ChirpStack JavaScript codecs that manufacturers commonly provide as part of their LoRaWAN device support.

Modbus

Simple integration of end-devices into BMS via Modbus, configuration via web interface, possibility to export register definitions to CSV file.

Node-RED

Visual programming tool for connecting devices, data and services that allows creating and running custom applications directly inside the gateway.

Time backup

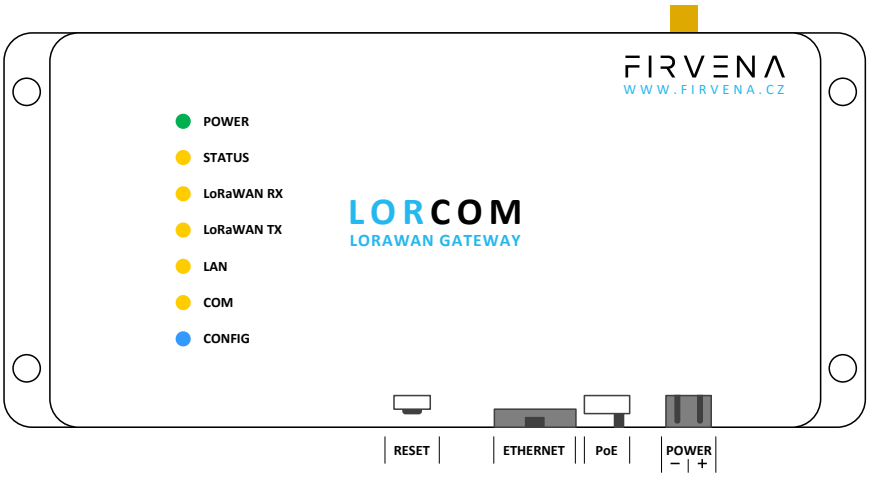
The gateway includes a supercap that ensures accurate time backup for at least 72 hours during power outages.

Support for 1000 LoRaWAN elements

The gateway is designed to serve 1000 elements on the LoRaWAN wireless network.

2 HARDWARE OVERVIEW

On the gateway, there is a power connector, RJ45 Ethernet connector, PoE mode switch, LoRaWAN antenna connector, LED indicators and reset button. The LEDs indicate the system status and activity of communication interfaces; they are very useful for troubleshooting during testing and commissioning of the gateway. The reset button is very useful for troubleshooting network settings.



PoE mode switch:

- **the "A" position (default)** enables the use of ACTIVE power supply
- **the "P" position** enables the use of PASSIVE power supply

LEDs		Meaning
POWER	🟢	Power supply and system status
STATUS	🟡	System status
LoRaWAN RX	🟡	Uplink packet (incoming)
LoRaWAN TX	🟡	Downlink packet (outgoing)
LAN	🟡	Network communication
COM	🟡	reserved
CONFIG	🔵	WebUI connection indicator

RESET button functions:

Hold the button to select a command, then confirm the selection, a brief press cancels the selection.

- **Reboot**:
  ↓hold 1s ▶ STATUS 🟡 on ▶ ↑release ▶ ↓hold 1s ▶ POWER 🟢 off ▶ system is restarting
- **Network reset:**
  ↓hold 3s ▶ STATUS 🟡 blinking slow ▶ ↑release ▶ ↓hold 1s ▶ STATUS 🟡 blinks 2x ▶ default network configuration used until next reboot.
  Use network reset if you are unable to connect to the gateway after changing the network settings.
  The following recovery network settings will be used:
  - IPv4 DHCP and IPv6 SLAAC enabled
  - link-local addresses enabled
  - additional fixed address IPv4 **192.168.1.90/24**, IPv6 **fd00::90/64**
- **Factory reset:**
  ↓hold 10s ▶ STATUS 🟡 blinking fast ▶ ↑release ▶ ↓hold 10s ▶ POWER 🟢 off ▶ system is returning to the factory state, deleting all settings and data
  This is the only way to reset the password.
- **Cancel command selection:** ↓↑ short press

⚠ **Factory reset is irreversible.**
All user settings and data will be lost.

3 TECHNICAL DATA

Category	Parameter	Value
Product	Product name	LORCOM
	Product title	LoRaWAN Gateway
	Vendor name	FIRVENA s.r.o.
Electrical data	Rated power supply	24 VDC, 500 mA
	Supply voltage range	POWER: 15–30 VDC; PoE Active: according to standard 802.3af; PoE Passive: 15–48 VDC
	Power consumption – idle	2.0 W @ 24 VDC (10 % CPU, Ethernet idle)
	Power consumption – full	2.7 W @ 24 VDC (full CPU and Ethernet load)
System	Processor	NXP with Arm® Cortex®-A55 core up to 1.7 GHz
	Memory	1 GB RAM; 8 GB eMMC
	Operating system	Yocto-based Embedded Linux
	Interfaces	1x Gigabit Ethernet; 1x LoRaWAN SX1302 EU868
Ethernet	Speed	1000/100/10 Mbit/s
	Connector	RJ45
	PoE	Yes (according to standard 802.3af)
LoRaWAN	Chipset	Semtech® SX1302
	Region	EU868
	Frequency	863–870 MHz
	Sensitivity	–125 dBm @ 125kHz/SF7 –139 dBm @ 125kHz/SF12
	TX Power	14 dBm
	Connector	SMA (female)
	Number of channels	8 channels
	Network server	Yes
	Number of devices	1000 devices
	LoRaWAN versions	1.0.0, 1.0.1, 1.0.2, 1.0.3, 1.0.4, 1.1.0
	Device classes	CLASS A, CLASS B, CLASS C
Antenna	Band	868 MHz ISM
	Frequency	863–870 MHz
	Peak gain	3.0 dBi
	Connector	SMA (male)
	Dimensions	170 × 18 × 10 mm
Physical	Housing material	Plastic – ABS, white

	IP Code	IP20
	Operating temperature	−20 °C to +70 °C
	Relative humidity	max. 80 %
	Dimensions without antenna	190 × 86 × 35 mm (W × H × D)
	Weight without antenna	260 g

4 SAFETY INFORMATION AND WARNINGS

Please follow the general safety regulations. This device may only be installed by a qualified person (accredited electrician) and after reading these instructions. Improper installation can result in health, property or equipment damage.

The product meets the general safety regulations. The protection Cover IP 20 allows installation only in normal, dry space.

The gateway must be powered from a safe voltage source that meets the requirements for input voltage range and must be installed in accordance with national and general safety standards.

Follow the safety instructions and applicable standards for the country and location of installation. The product may only be used in accordance with this manual.

To avoid the risk of electrical shock or fire, the maximum operating parameters of the gateway must not be exceeded.

Use only unmodified products.

Only cable types with sufficient cross-section and insulation properties may be used for the connection.

STORAGE

The device must be stored in a temperature range 0-40 °C and a relative humidity of up to 80 %, and non-condensing spaces. Products must not be exposed shock, harmful vapors or gases.

REPAIRS

Products are repaired by the manufacturer. Products to be repaired are shipped in a package that ensures shock absorption and protects the products against damage during shipment.

WARRANTY

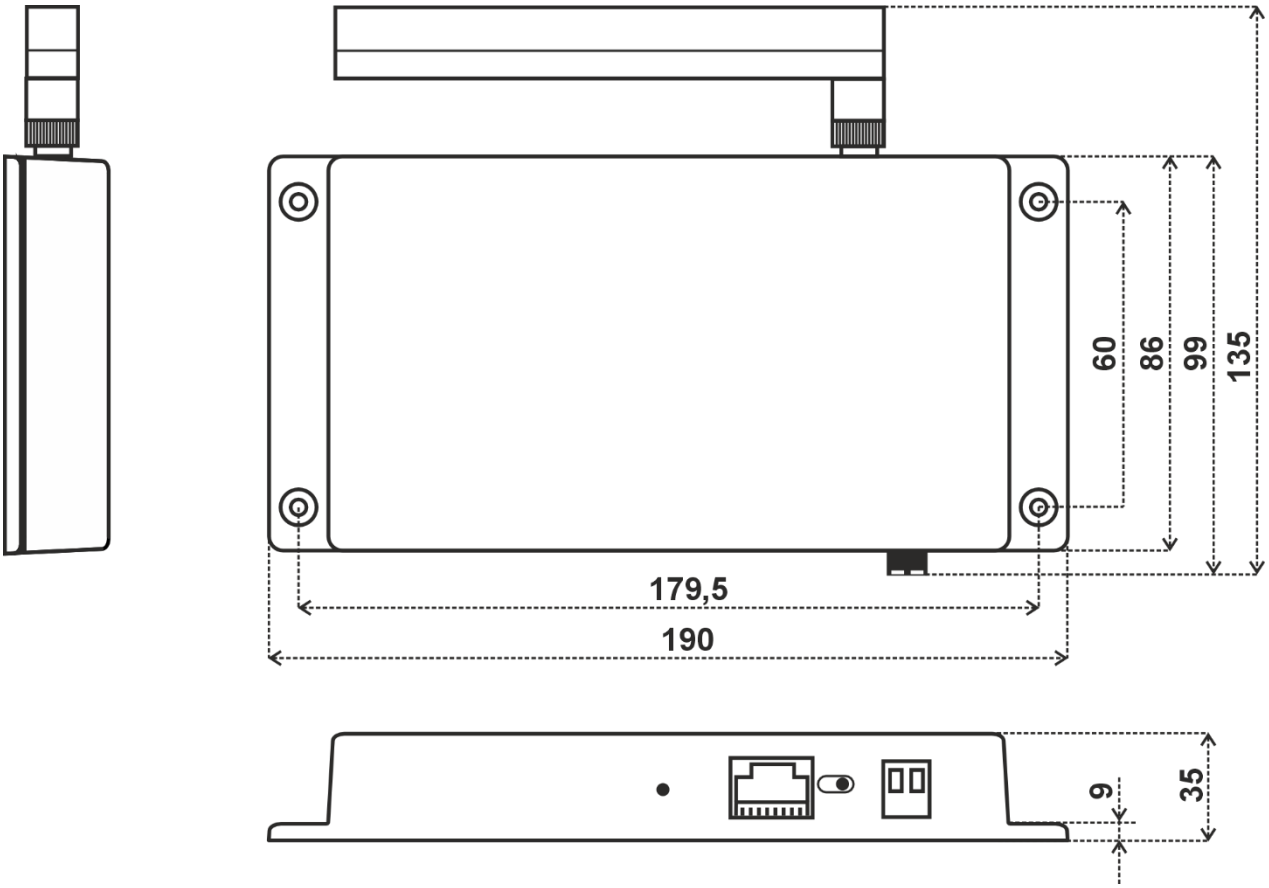
The product is warranted 24 months from the date of delivery that is mentioned on the delivery note. The manufacturer only guarantees properties and parameters that are explicitly described in the technical documentation. Claims, complaints and returns must be directed exclusively to the manufacturer. The complaint must contain the exact product identification, delivery note number and defects description. The manufacturer is not responsible for defects caused by improper storage, improper external connection, damages caused by external influences especially due to unacceptable size, incorrect adjustment, improper installation, incorrect operation or normal wear and tear.

PRODUCT DISPOSAL

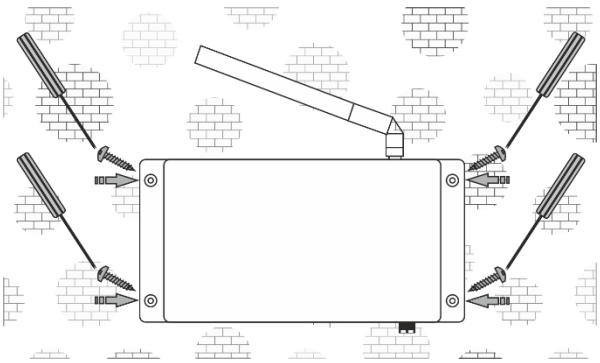
The product does not belong to municipal waste. The product must be disposed to the separate waste collection with the possibility of recycling, according to local regulations and legislation. The product contains electronic components.

5 INSTALLATION

Dimensions (in mm)



Mounting



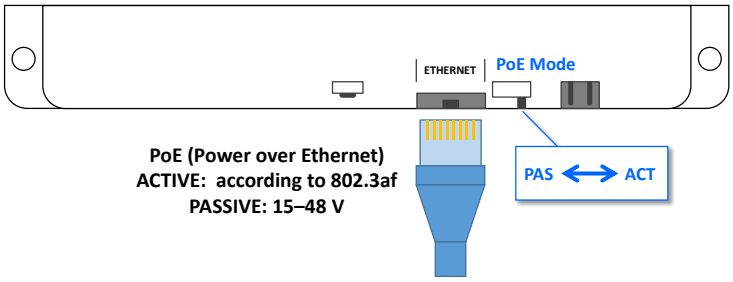
The gateway is fixed by using plastic anchors and Phillips screws to the wall or solid surfaces of suspended ceilings making sure that it is in a good radio position for receiving and transmitting signals. When located in shielded surfaces, the standard supplied antenna can be replaced with another permitted antenna type for the 868 MHz frequency with an SMA (Male) connector.

Power supply

There are two possible power supply connections:

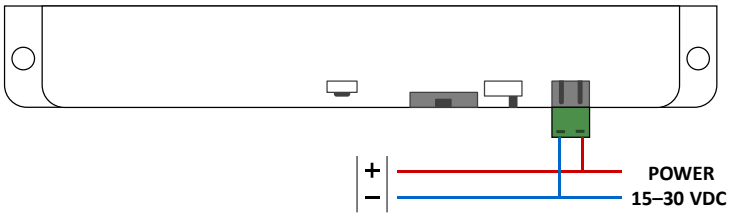
1. Power supply via PoE (Power over Ethernet):

The device supports PoE according to the 802.3af standard, network elements must support this type of power interface. To use passive PoE, set the switch next to the ETHERNET connector to the left position.



2. Power supply from an external source:

The gateway must be powered from a safe voltage source that meets the input voltage range requirements. The electrical installation must be in accordance with national requirements and safety standards.

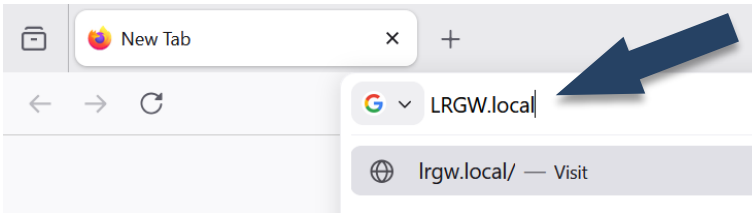


6 CONFIGURATION

6.1 Accessing Web Interface

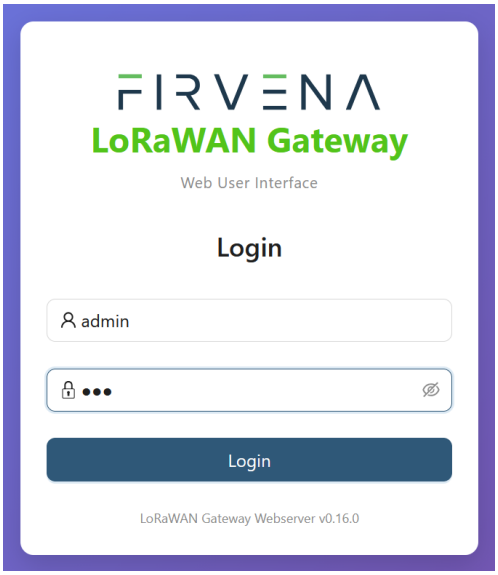
The gateway has an integrated web application (*WebUI*) that allows configuration using a web browser. This section describes how to open *WebUI* for the first time.

Power on the gateway and connect it to the local network via the ETHERNET connector, or connect it directly to a computer with an Ethernet adapter. The gateway should automatically obtain an IP address and other network parameters from a DHCP server, if available, or use automatic address configuration methods.



The *WebUI* is available on **HTTP port 80**, each gateway has a universal name that works in the local network, simply enter "*http://LRGW.local*" into the address bar of your web browser (letter case does not matter, you can write *lrgw.local*).

Login is required. The default password is **"123"**; you must change the password before continuing. The blue CONFIG 🔵 should light up after login.



The connection using LRGW.local may not work for several reasons:

1. Multiple gateways on the same network (name conflict)

If multiple gateways are connected to the same local network, the names are numbered, e.g.: *LRGW.local, LRGW-2.local, LRGW-3.local, …* You can try the numbered versions and watch the CONFIG LED.

The gateway has also a unique name *LRGW-XXXXXX*, which is printed on the product label, the second part are the last six characters from the MAC address, e.g.: *http://LRGW-4BAC34.local*.

2. Blocked mDNS or missing mDNS client; or gateway has other subnet or incorrect network settings

You will need the IP address.

One simple method always works: do the *Network reset* using the RESET button – see Hardware Overview and use the fixed IPv4 or IPv6 address: http://192.168.1.90 or http://[fd00::90].

Configure your computer to have the same subnet and a different IP address, e.g. 192.168.1.95 (subnet mask 255.255.255.0), fd00::95 (prefix length 64), you should be able to connect to the gateway and check/repair its network settings.

- It takes some time for the gateway to boot, the ready state is indicated by STAUS 🟡 off.
- If the password is lost, you need to perform a Factory reset using the RESET button.
- The default addresses *192.168.1.90* and *fd00::90* only work after the Network reset and until the next Reboot.

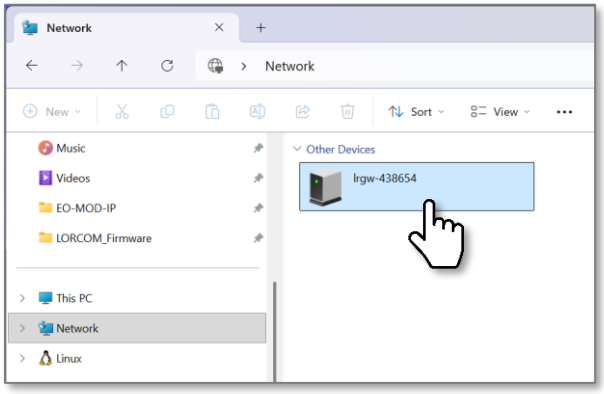
Other access options

DHCP Server: You can check the DHCP clients list if you have access to the local DHCP server (usually through the configuration interface of your router). Look for the client name *"LRGW-XXXXXX"* (same as unique name). To access the *WebUI*, enter the IP address to the address bar of your web browser.

File Explorer in Microsoft Windows (SSDP, UPnP): The gateway is visible as a network device on the Microsoft Windows operating system in *[File Explorer > Network]*. *SSDP* stands for *Simple Service Discovery Protocol*, it is an alternative to *mDNS*.

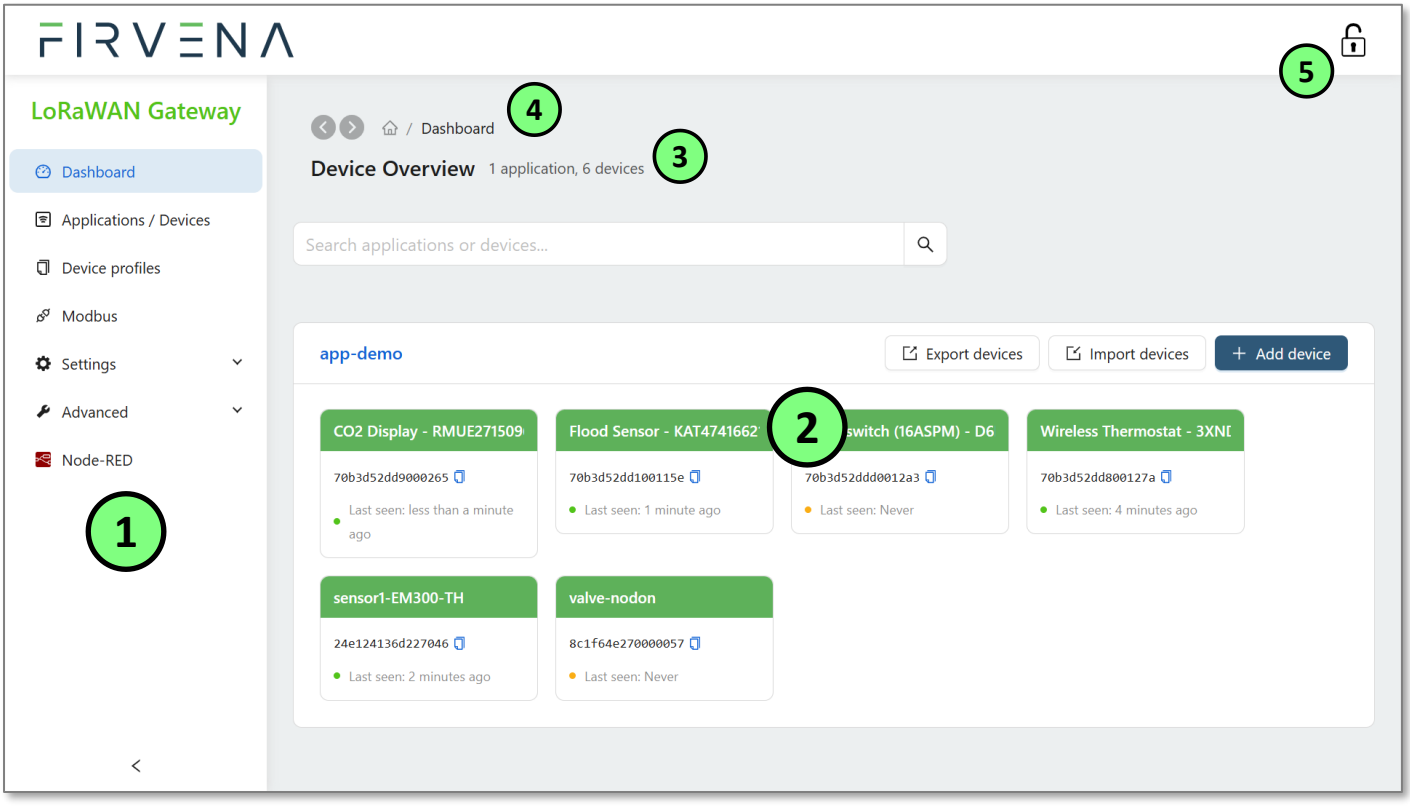
Double click on the device item to open the *WebUI*. *[Right click > Properties]* will display additional information.

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



6.2 WebUI Overview

1. Menu:
   Dashboard: an overview of all applications and devices
   Applications / Devices: view application and device details
   Device profiles: view device profile definition (a configuration shared by devices of the same type)
   Modbus: Modbus overview of all applications and devices
   Settings: system settings (network, firmware update, password, …)
   Advanced: advanced configuration options
   Node-RED: open the Node-RED in a new tab
2. Content
3. Content title
4. Navigation breadcrumb with back and forth buttons
5. Logout



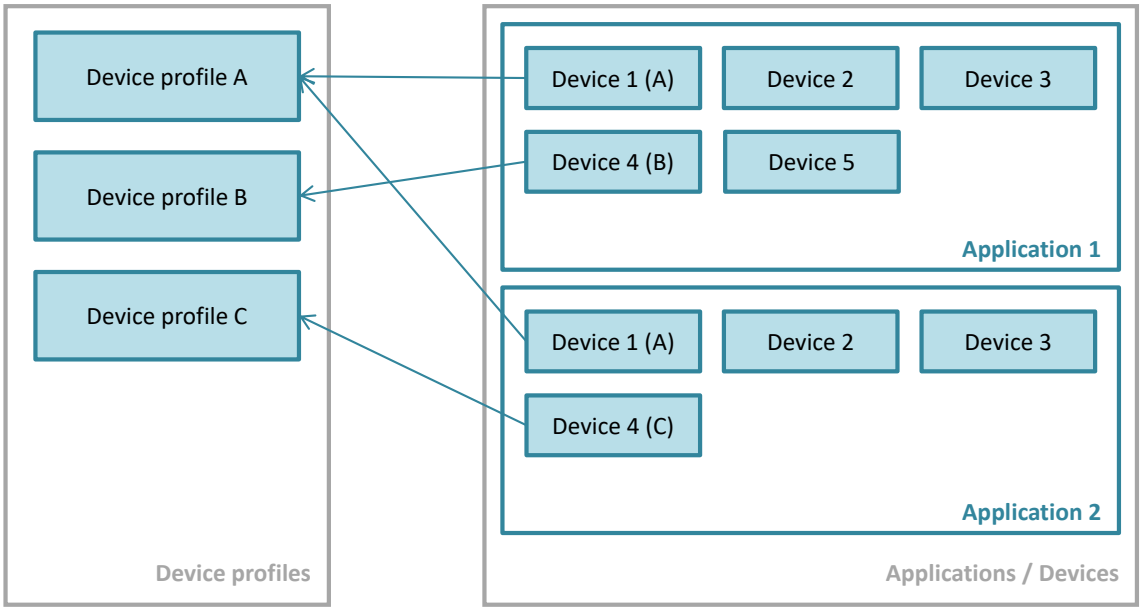
6.3 Adding Devices

6.3.1 Overview

The devices are organized into groups called *"Applications"*. Devices of different type can live under the same application. Each device belongs to one application; it cannot be used in multiple applications.

The division into applications affects other parts of the system in a way that simplifies the subsequent configuration, especially for large projects. You can filter devices by application in MQTT; in Modbus, each application has its own Modbus server. How you use the applications is up to you; for example, each room or floor can be one application, or power measurement and room control can be applications. It is also possible to use one application for everything.

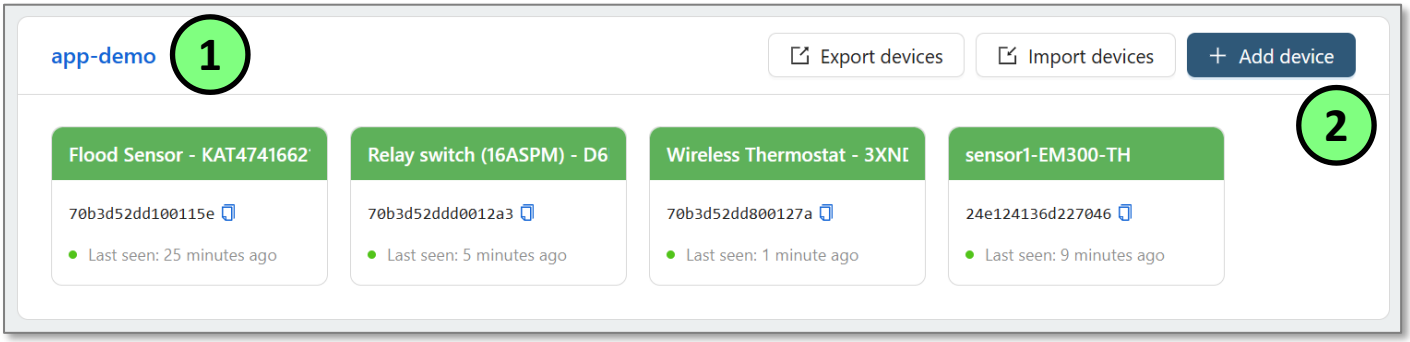
Another important term is *"Device profile"*. This shared configuration applies to all devices of the same type that have the same profile in their settings. The purpose of profiles is to separate generic device settings. Changing the device profile will affect all devices that use it.

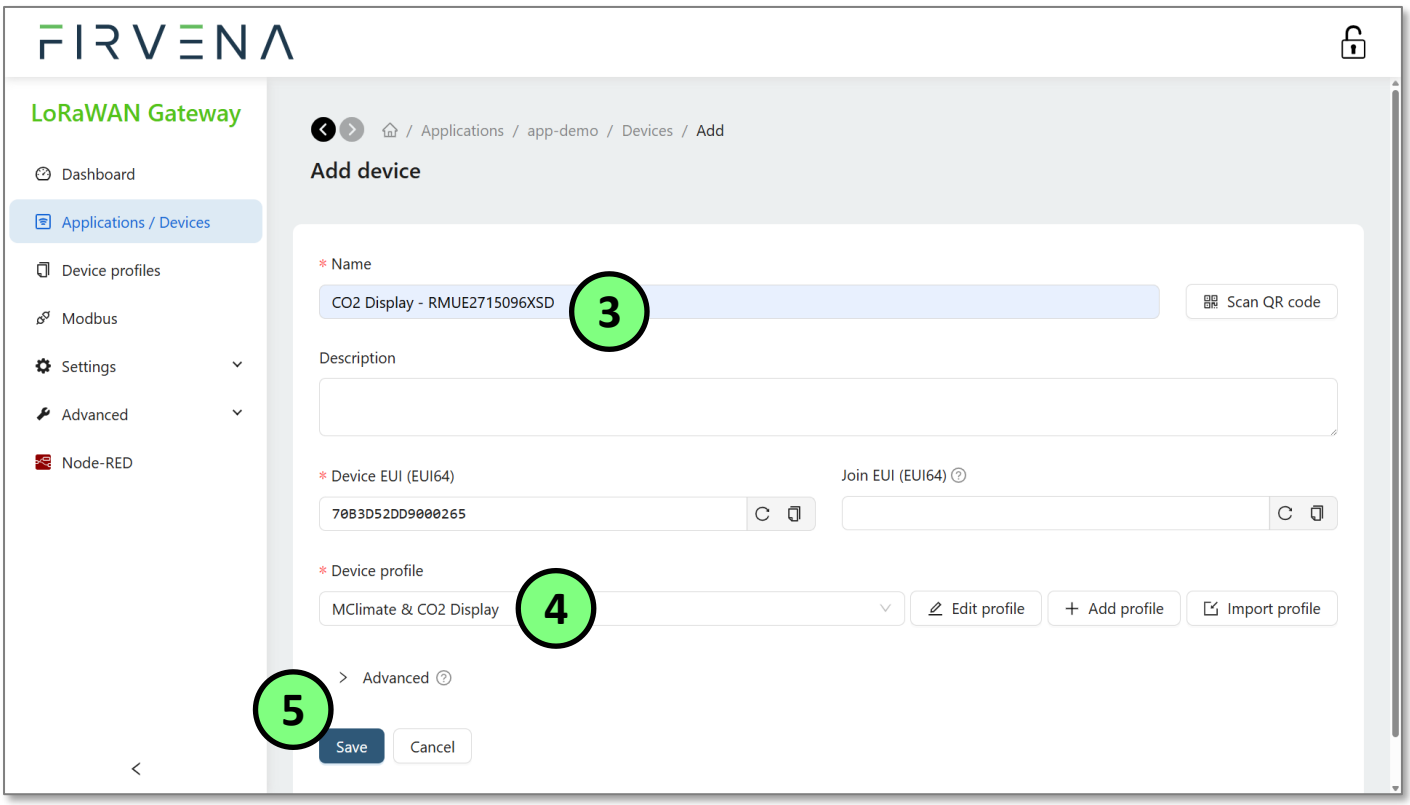


6.3.2 Add device

We will use MClimate's CO2 sensor as an example.

1. Choose/create an application (app-demo)
2. Use [Dashboard > Add device]
   or [Applications / Devices > Application details > Devices > Add device]
3. Fill in the **Name** and **Device EUI**
4. Choose/create a **Device profile** – see Device Profiles
5. Save

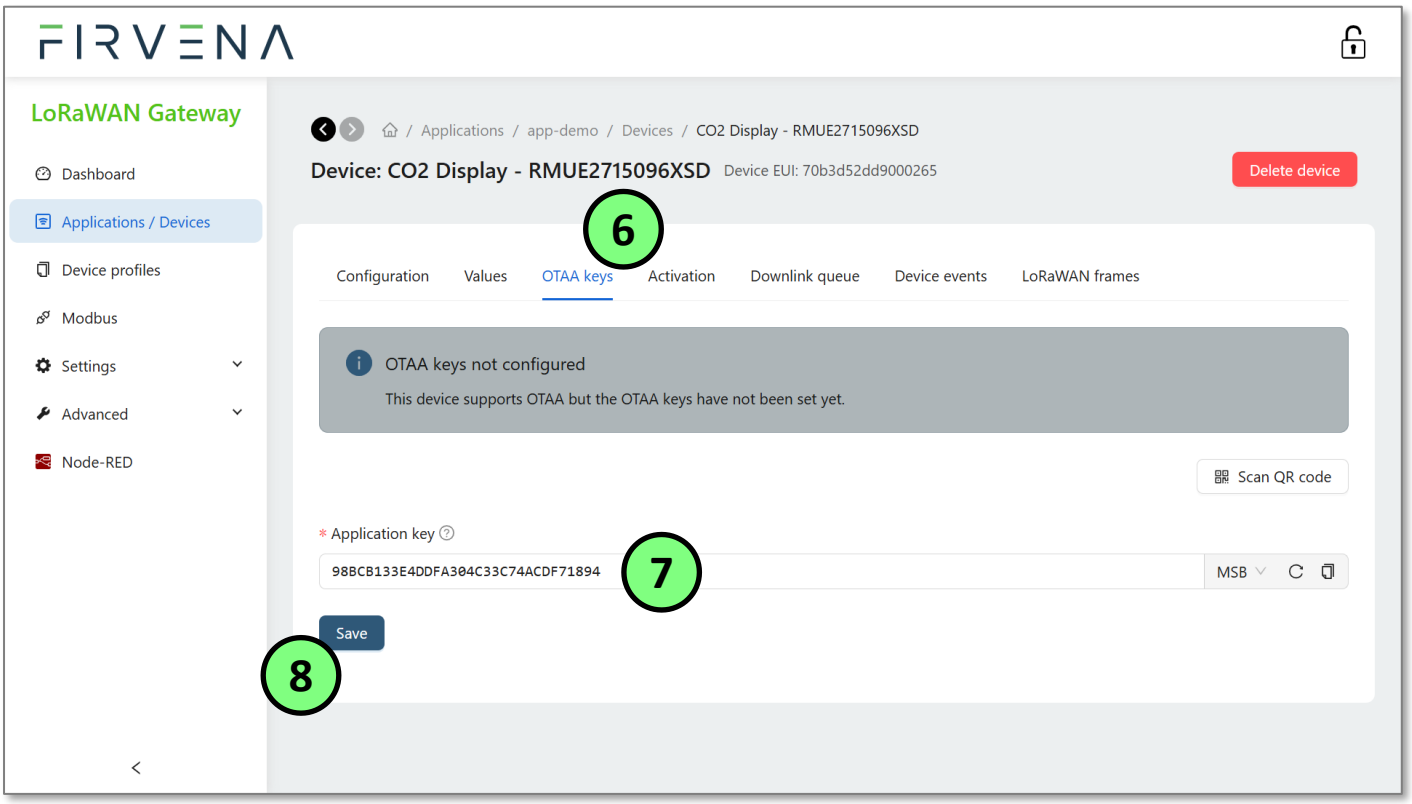




> ➢ You can omit Join EUI (formerly App EUI), it will be set automatically from device join request. In fact, it is not used at all, because no external *Join server* is used.

The next step is to configure security key(s):

6. Use [Device details > OTAA keys]
7. Enter a unique **Application key**, which must match the key stored in the device
8. Save

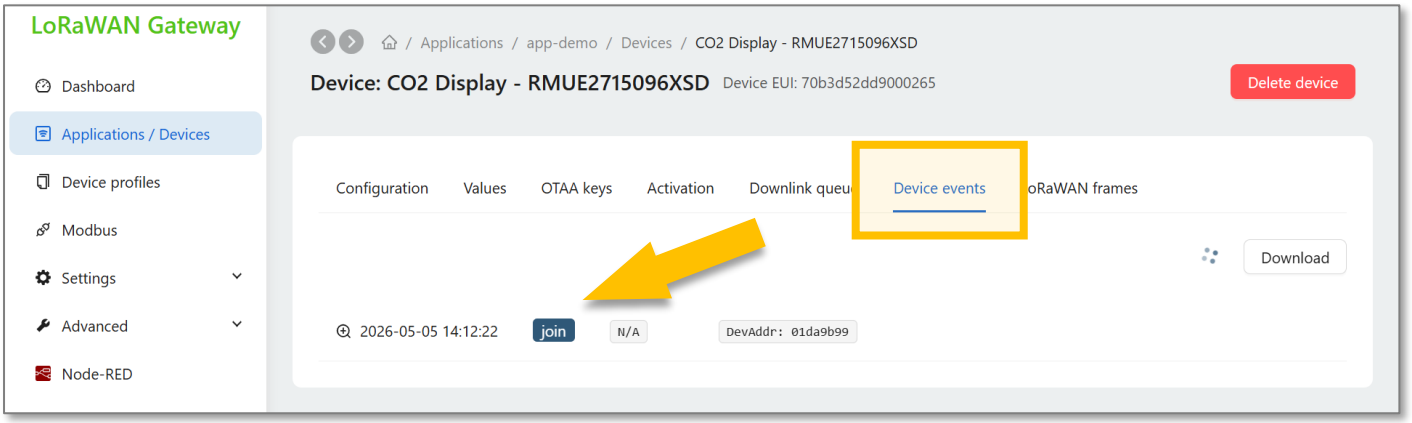


➢ The Application key is fixed (provided by the manufacturer) or sometimes configurable on the device. Contact the manufacturer or seller; for example, MClimate sends the keys in a CSV file.

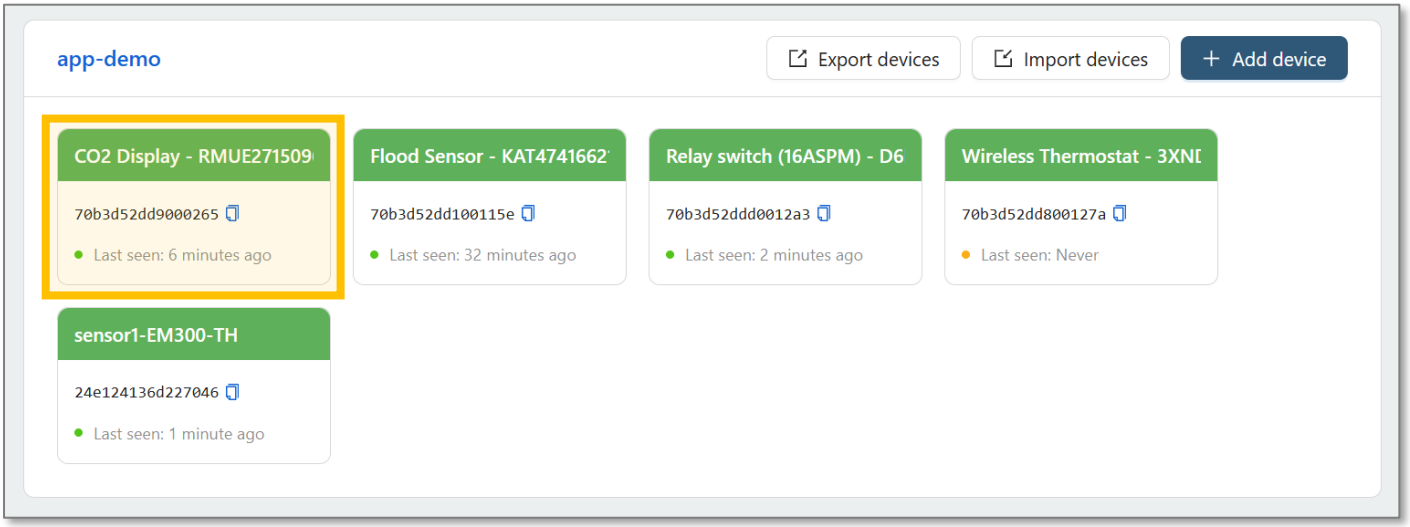
Keep the OTAA keys secret and do not print them on the device
These keys are used to secure the communication between the device and application.

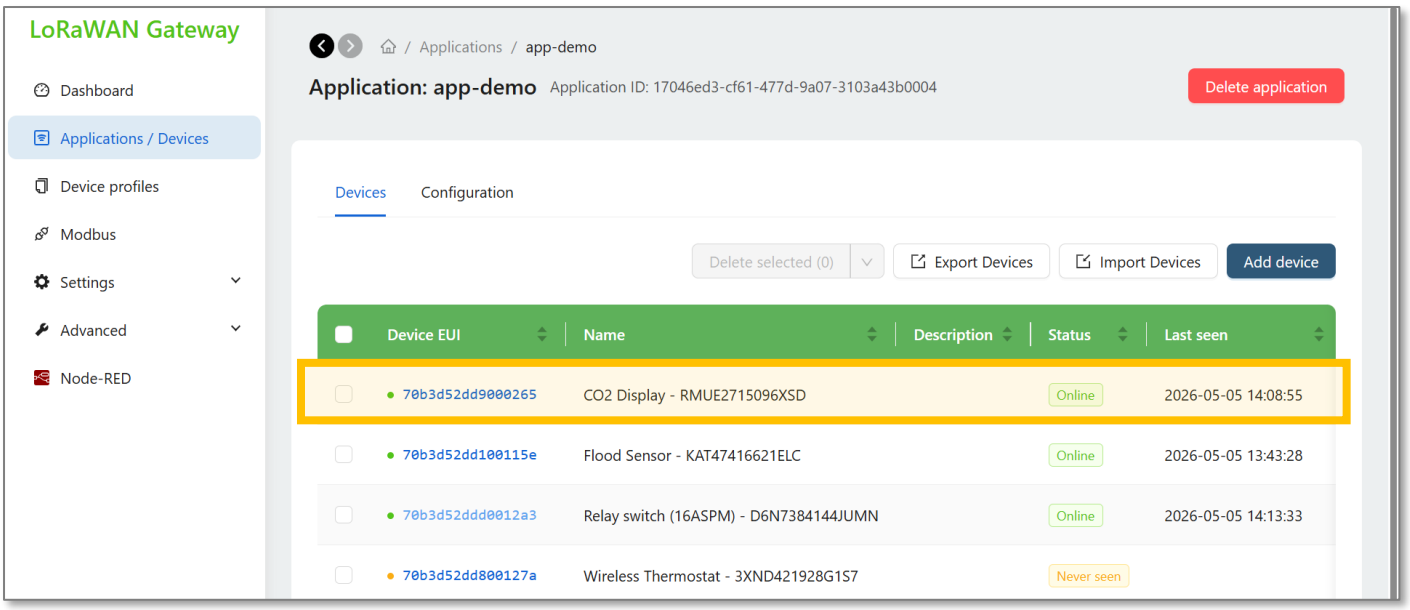
Now you can activate the device to start the *OTAA Join* procedure. *OTAA Join* is usually triggered by inserting the battery, using a reset button or using another control interface on the device – follow the device documentation.

You can watch the *Device events* log:



The device status should go green:

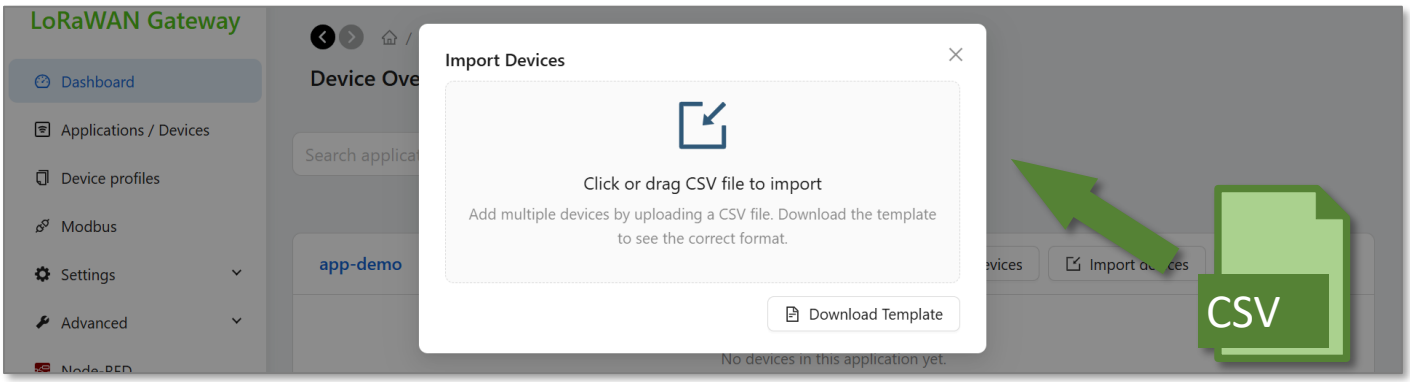


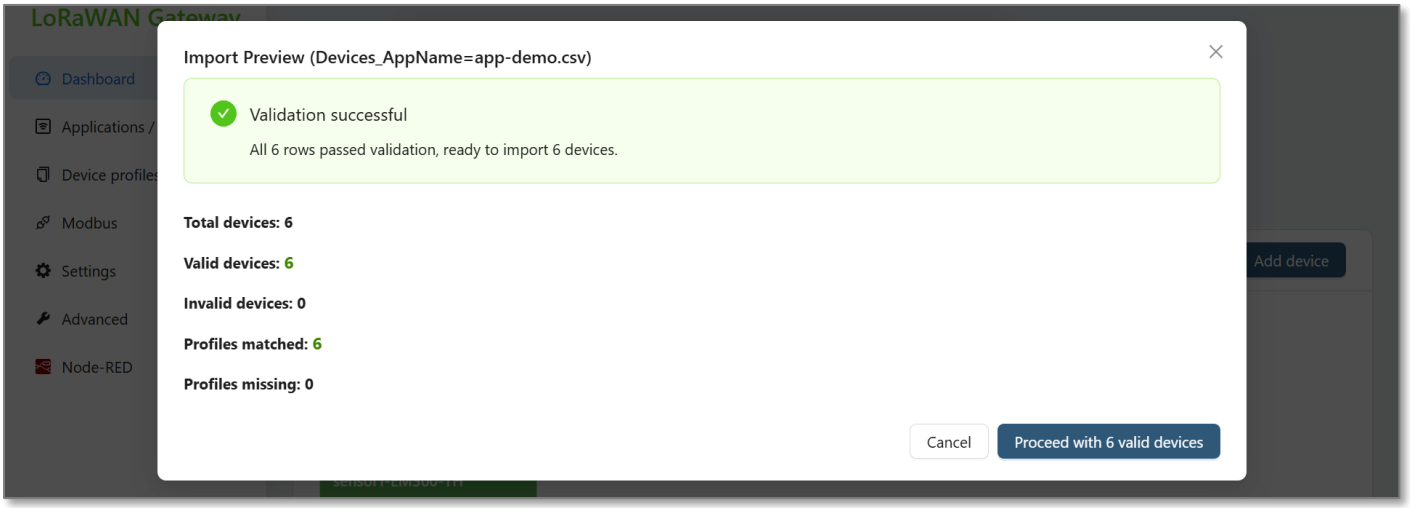


6.3.3 Bulk Add Device

You can import multiple devices from a CSV file. This saves a lot of time.

1. Choose/create an application (app-demo)
2. Use [Dashboard > Import devices]
   or [Applications / Devices > Application details > Devices > Import devices]
3. Upload the file
4. You will go through a validation logic
5. Proceed



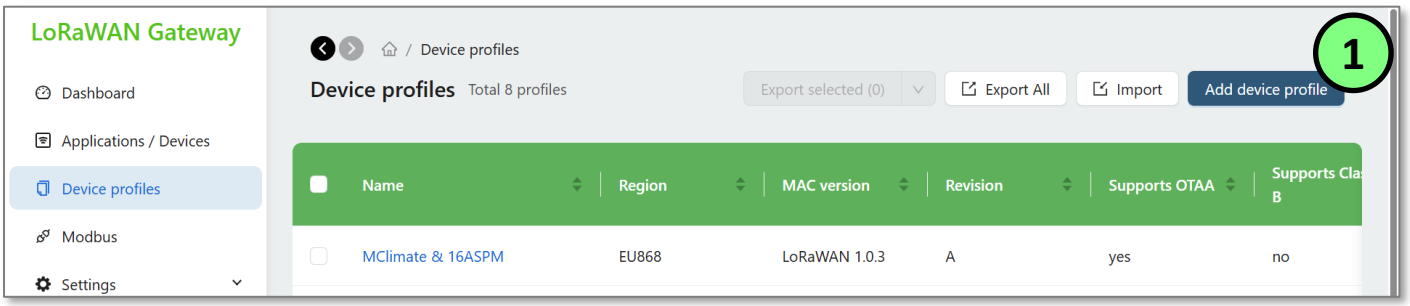


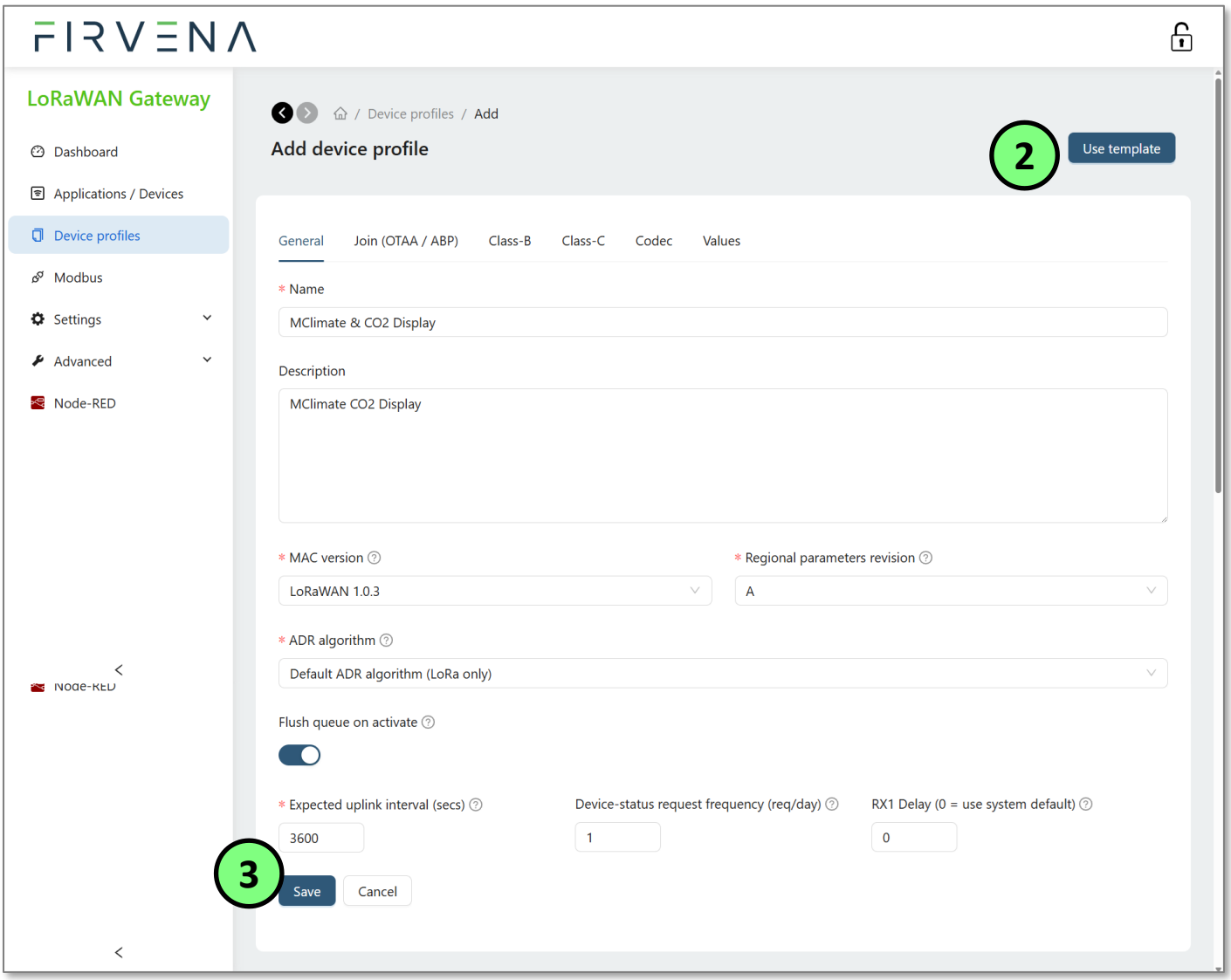
➢ Device profiles are matched based on their name, so a device profile with the same name as in the *"deviceProfileName"* column must exist prior to the import.

6.4 Device Profiles

To add a device profile:

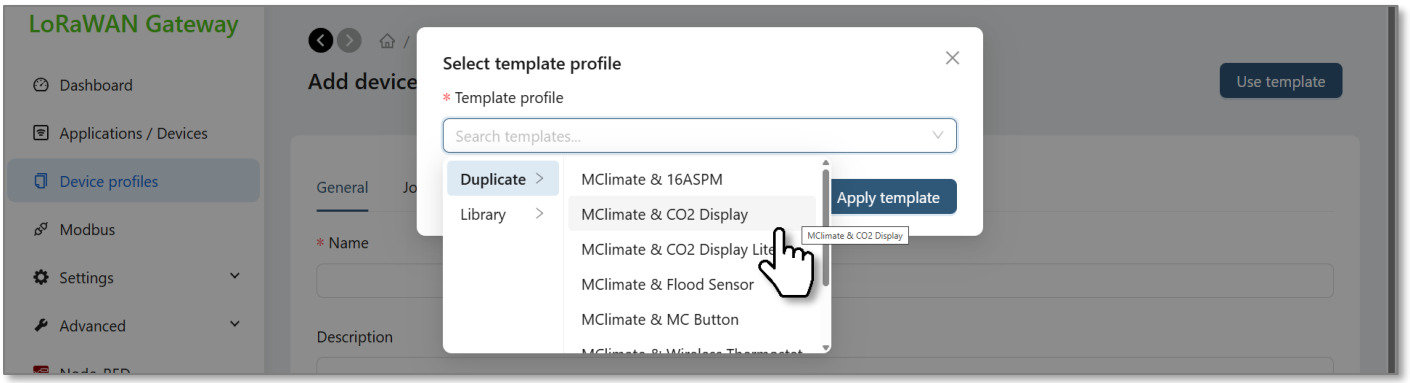
1. Use [Device profiles > Add device profile]
2. Fill in the parameters manually or choose a template
   You should find the parameters in the datasheet or device support from the manufacturer. Leave the default values if you are not sure what values to set, you can refine it later.
3. Save

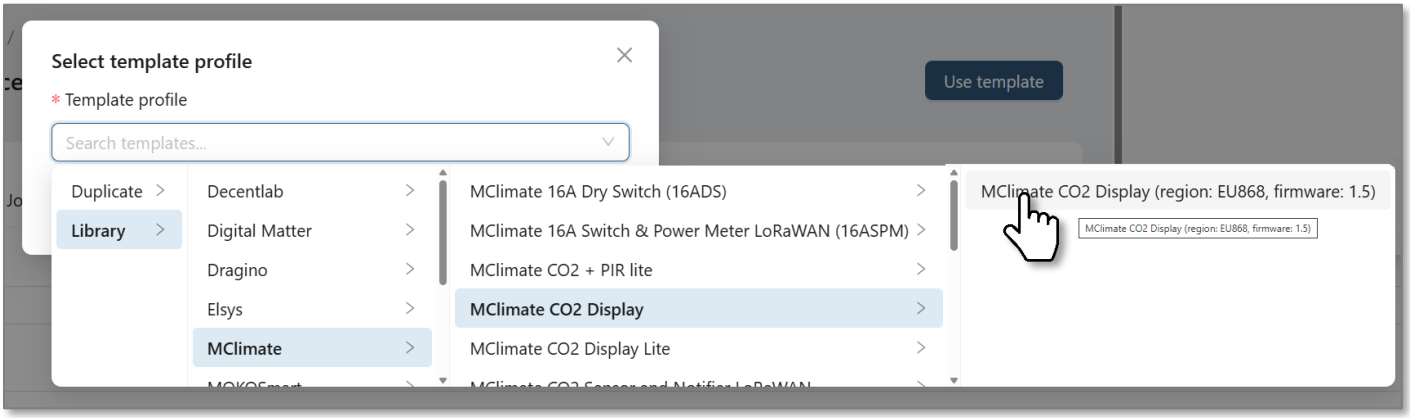




Using a template

You can *"Duplicate"* an existing profile or search for the device in the *"Library"*:



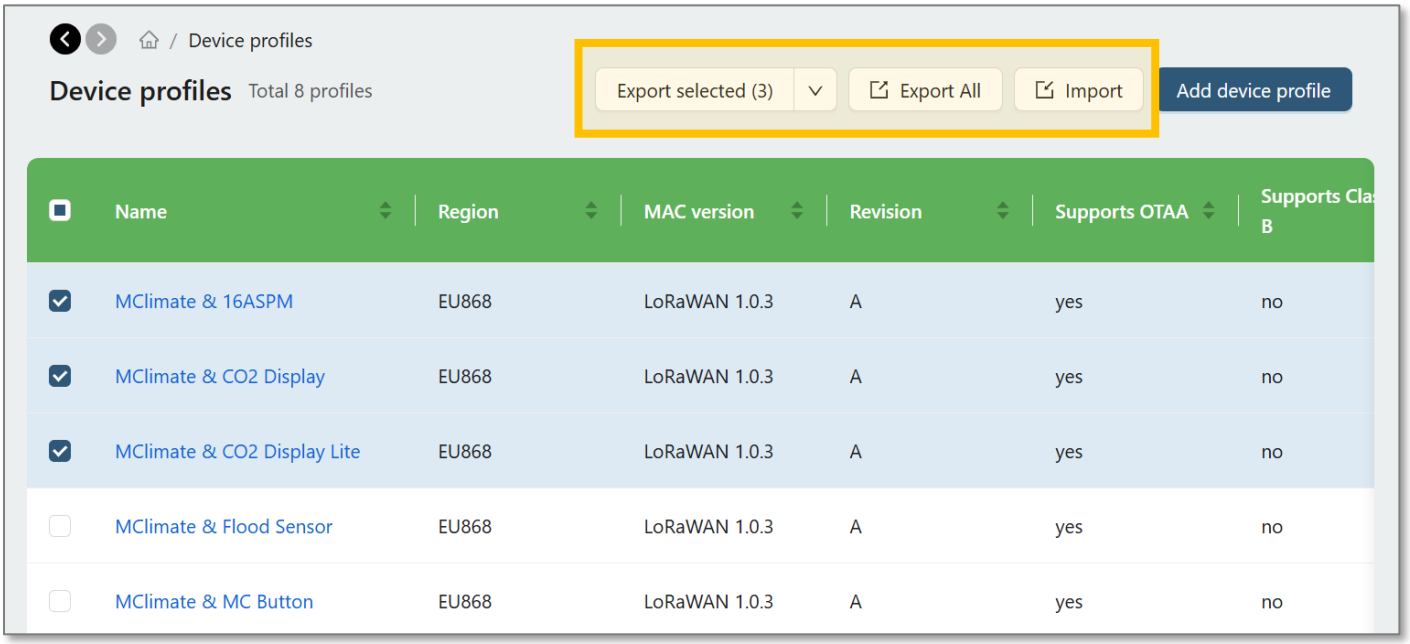


General recommendation for library firmware version selection: *Choose the highest available firmware version that is lower than your device firmware version and has the same major version number, for example:*

- *your device V1.8; available templates V1.3, V1.4, V1.5 → choose V1.5*
- *your device V1.8; available templates V1.3, V1.4, V1.5, V1.8, V2.0 → choose V1.8*

Export/Import device profiles

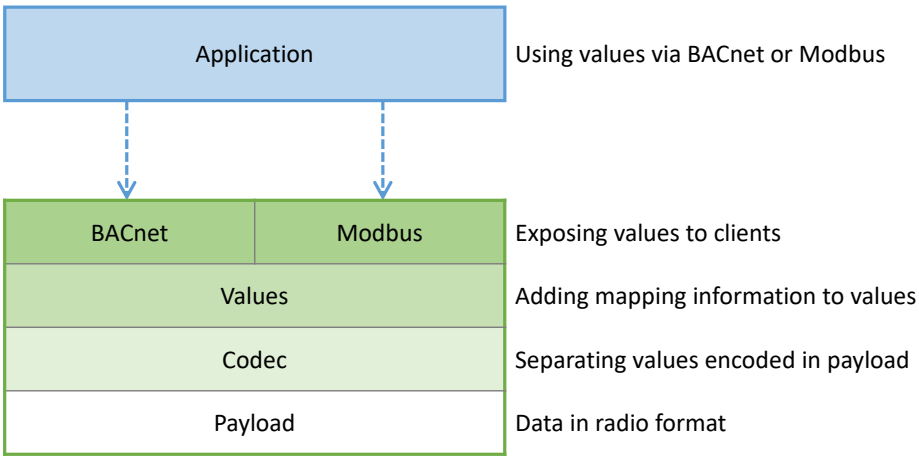
You can also import/export the profiles to use them in another gateway:



6.5 Values

The “Values” configuration in device profile is a recipe for creating Modbus registers and BACnet objects; it is also used to create the table in [Device details > Values].

You can think of it as a layered structure:



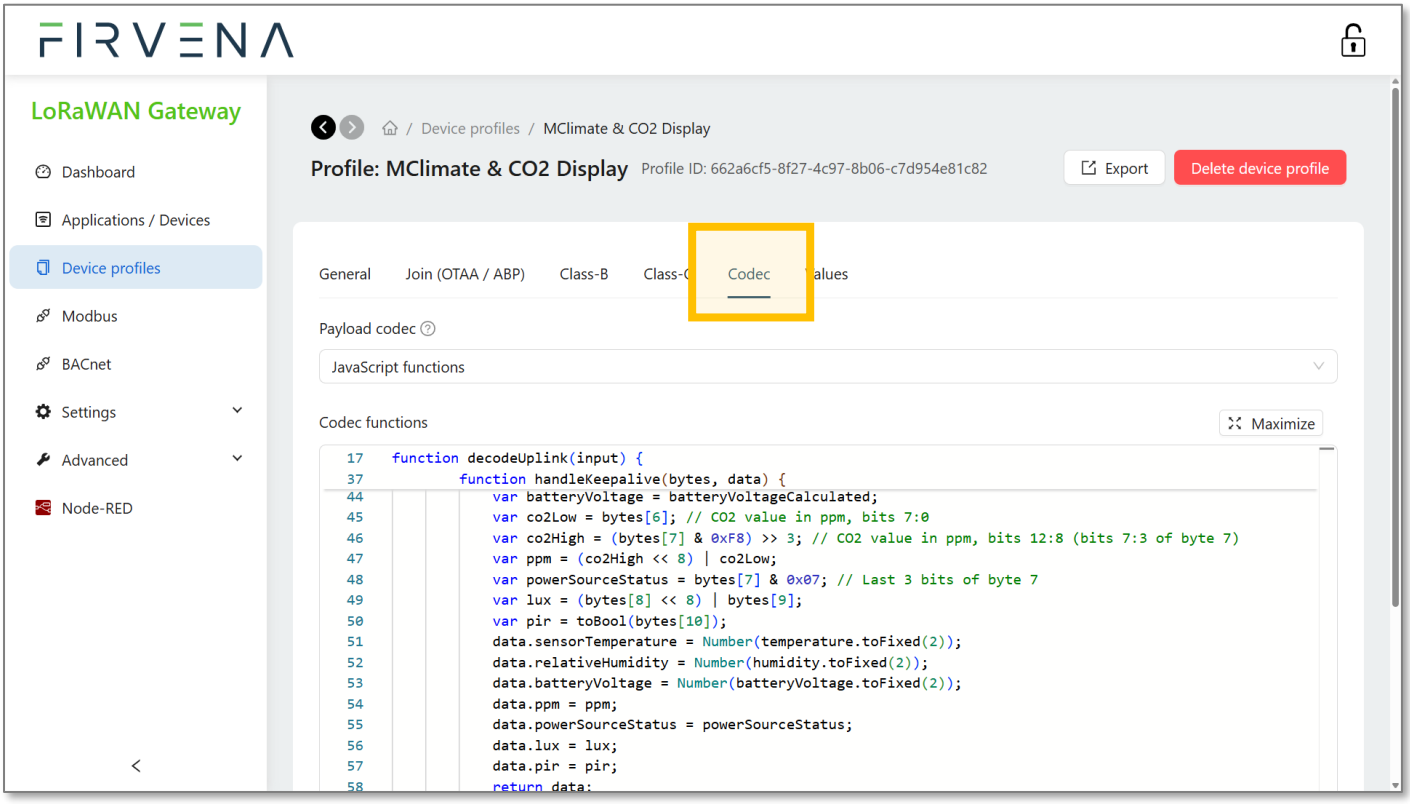
The table in [Device details > Values] is very useful for testing the device profile:

- It emulates the BACnet and Modbus interface behaviour.
- It shows all values and metavalues defined in the device profile.
- It shows instance numbers of BACnet objects and addresses of Modbus registers, if the BACnet/Modbus server is enabled for the parent application.
- It allows to watch uplink data and send downlinks.

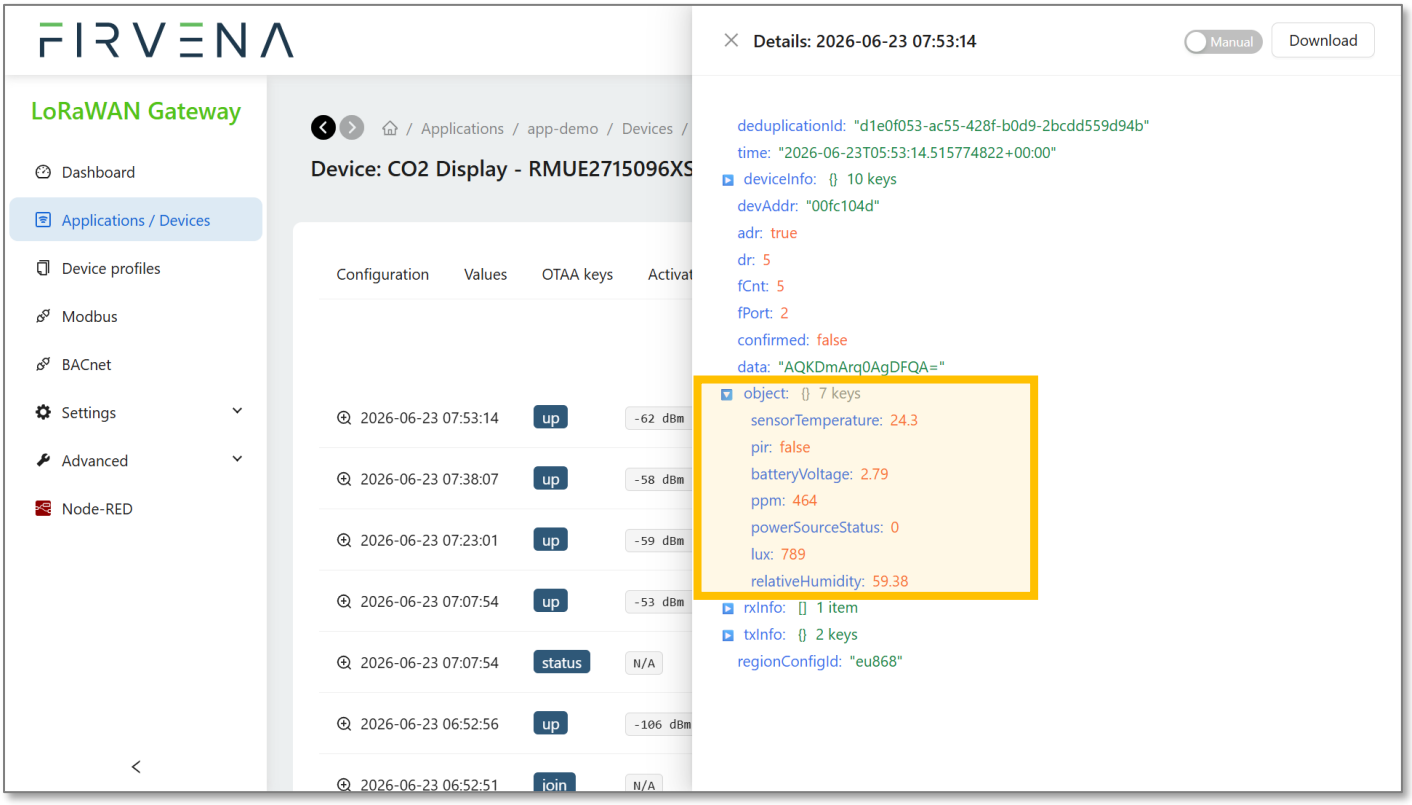
The workflow for creating a new device profile may be as follows:

1. Definition of codec in [Device profile > Codec]
2. Testing of codec output in [Device details > Device events] → adjustments in codec
3. Definition of values in [Device profile > Values]
4. Testing in [Device details > Values] → adjustments in device profile
5. Testing via a BACnet or Modbus client → adjustments in device profile

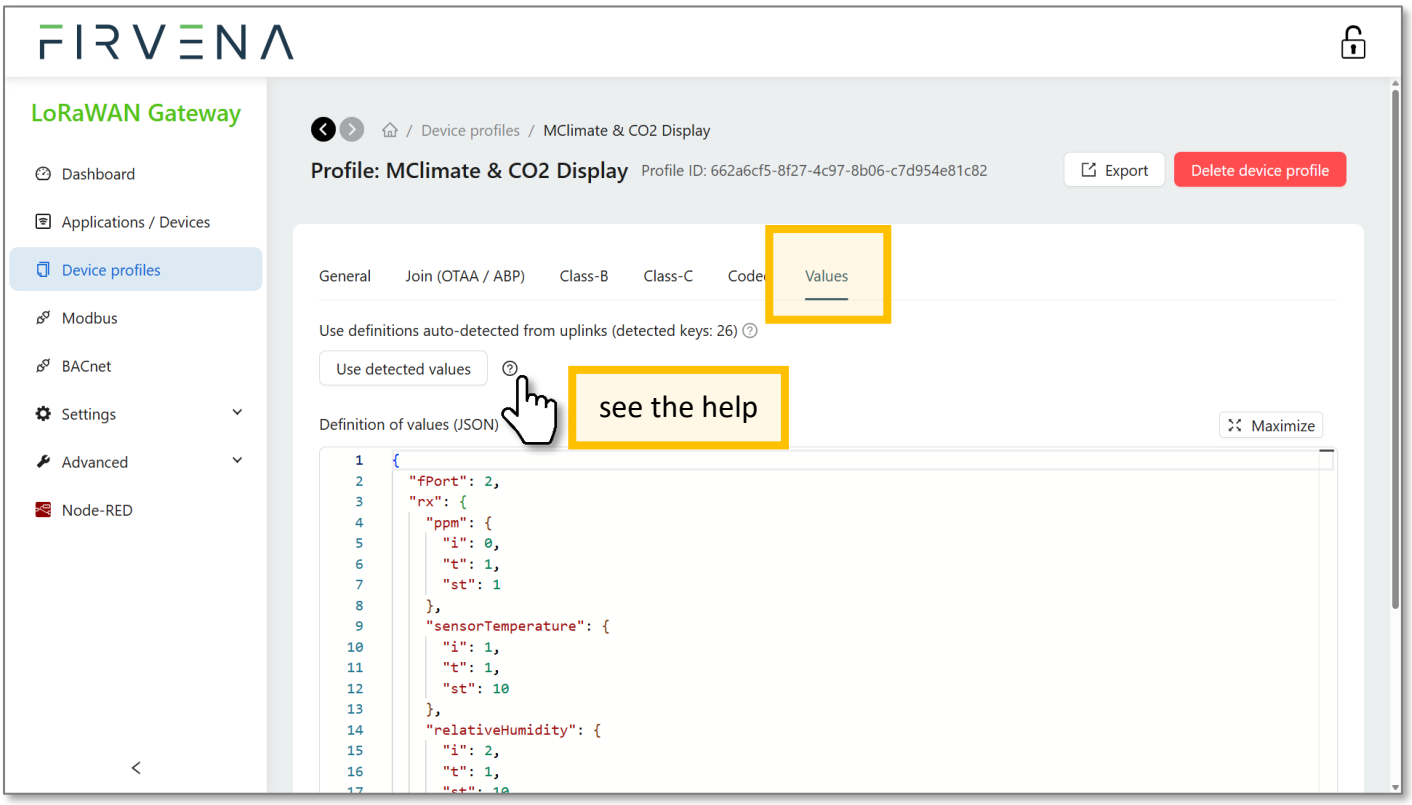
Definition of codec:



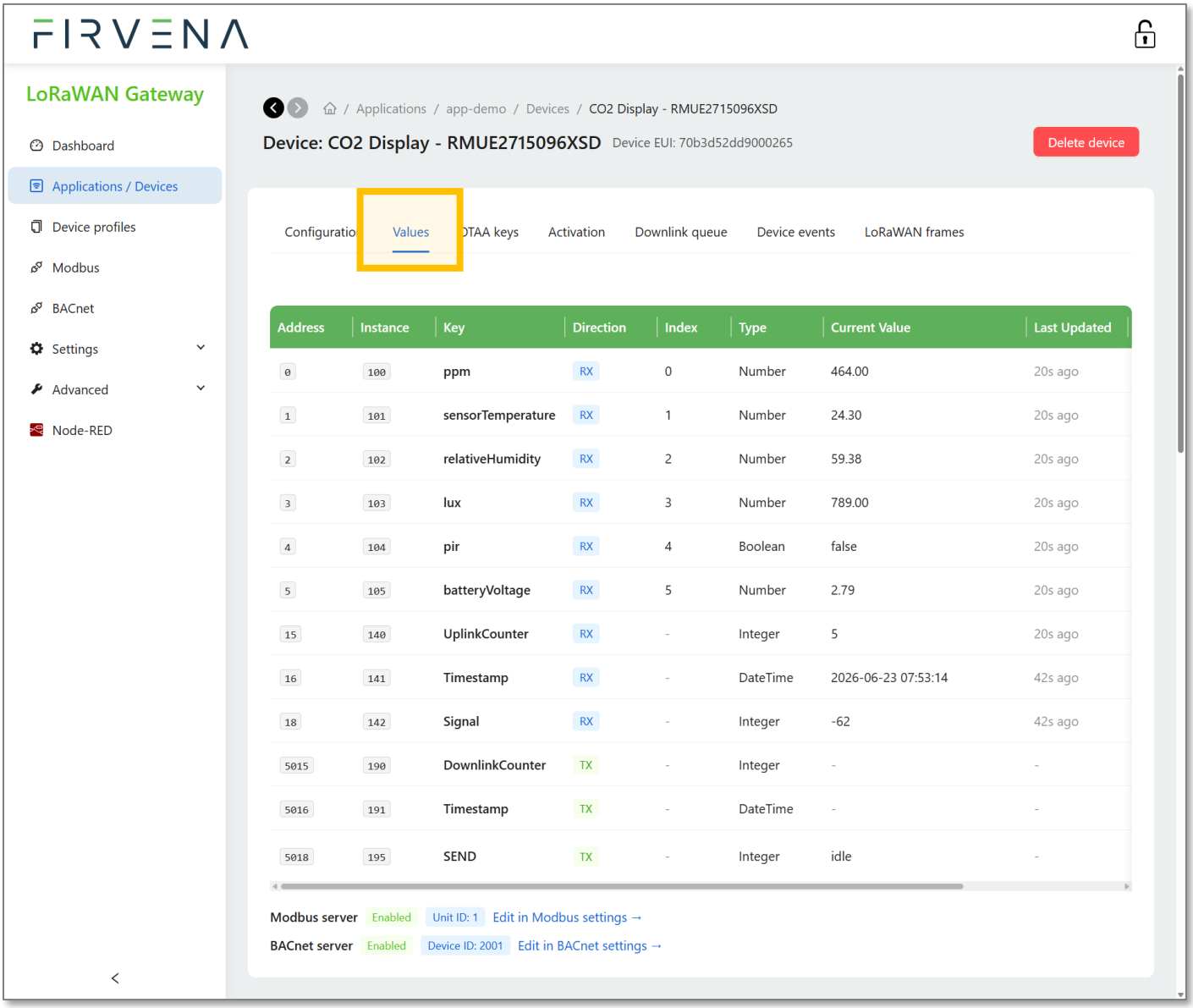
Testing of codec output:



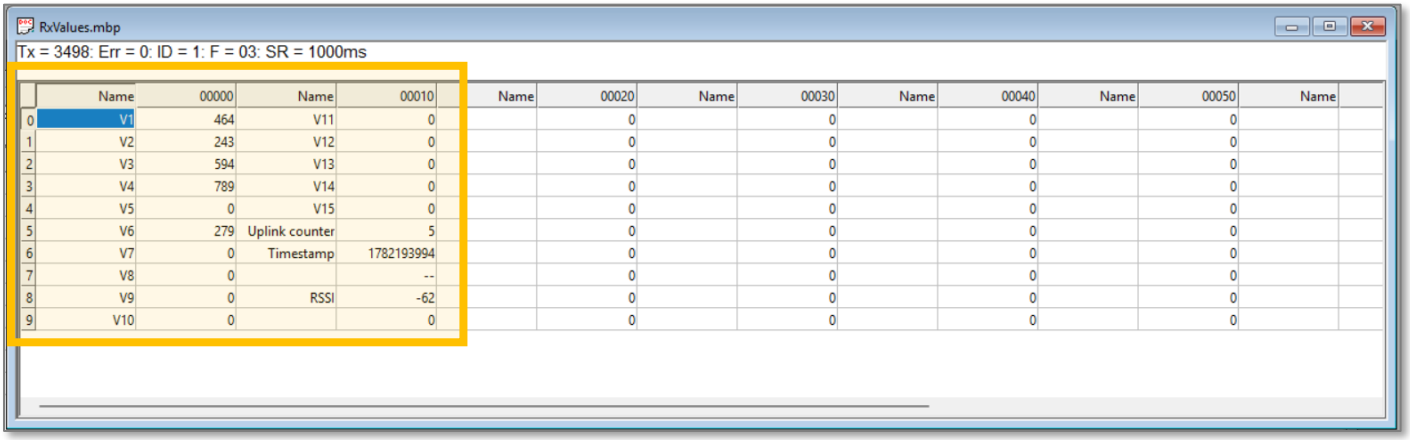
Definition of values:



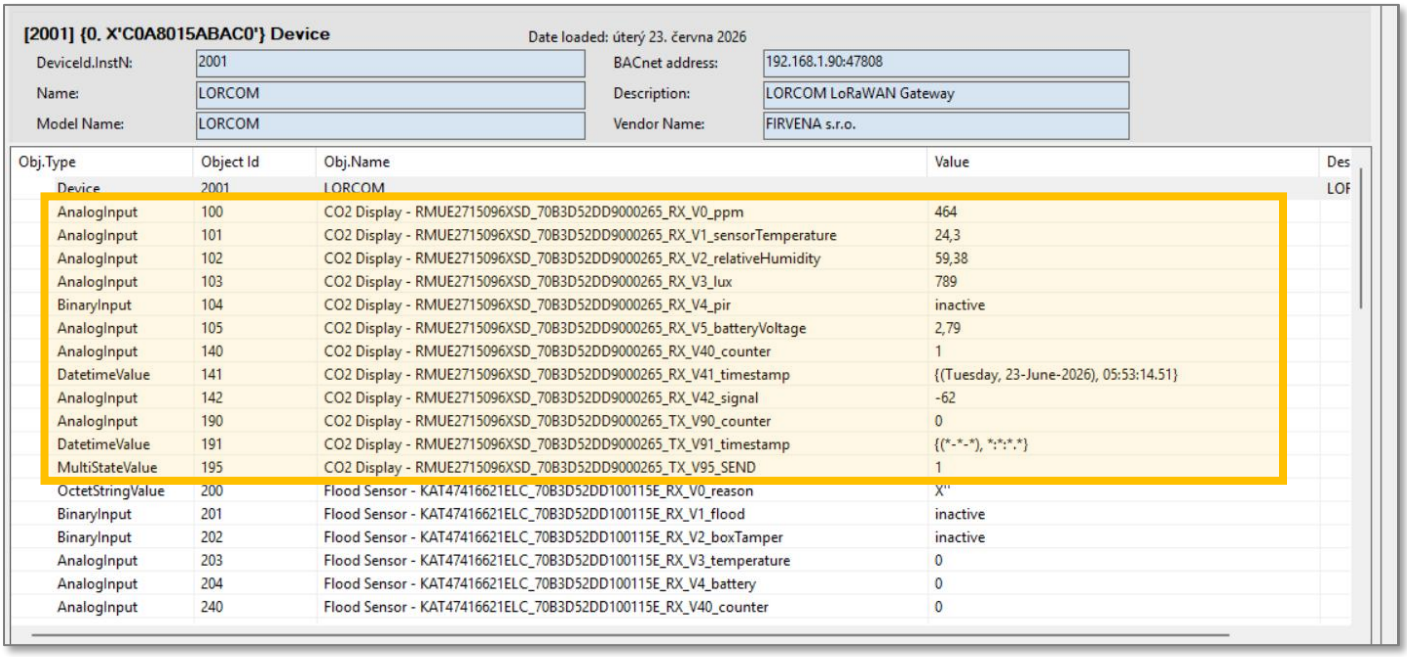
Testing in [Device details > Values]:



Modbus registers via a client:



BACnet objects via a client:



[2001] {0. X'C0A8015ABAC0'} Device — Date loaded: úterý 23. června 2026

DeviceId.InstN:	2001	BACnet address:	192.168.1.90:47808
Name:	LORCOM	Description:	LORCOM LoRaWAN Gateway
Model Name:	LORCOM	Vendor Name:	FIRVENA s.r.o.

Obj.Type	Object Id	Obj.Name	Value	Des
Device	2001	LORCOM		LOF
AnalogInput	100	CO2 Display - RMUE2715096XSD_70B3D52DD9000265_RX_V0_ppm	464	
AnalogInput	101	CO2 Display - RMUE2715096XSD_70B3D52DD9000265_RX_V1_sensorTemperature	24,3	
AnalogInput	102	CO2 Display - RMUE2715096XSD_70B3D52DD9000265_RX_V2_relativeHumidity	59,38	
AnalogInput	103	CO2 Display - RMUE2715096XSD_70B3D52DD9000265_RX_V3_lux	789	
BinaryInput	104	CO2 Display - RMUE2715096XSD_70B3D52DD9000265_RX_V4_pir	inactive	
AnalogInput	105	CO2 Display - RMUE2715096XSD_70B3D52DD9000265_RX_V5_batteryVoltage	2,79	
AnalogInput	140	CO2 Display - RMUE2715096XSD_70B3D52DD9000265_RX_V40_counter	1	
DatetimeValue	141	CO2 Display - RMUE2715096XSD_70B3D52DD9000265_RX_V41_timestamp	{(Tuesday, 23-June-2026), 05:53:14.51}	
AnalogInput	142	CO2 Display - RMUE2715096XSD_70B3D52DD9000265_RX_V42_signal	-62	
AnalogInput	190	CO2 Display - RMUE2715096XSD_70B3D52DD9000265_TX_V90_counter	0	
DatetimeValue	191	CO2 Display - RMUE2715096XSD_70B3D52DD9000265_TX_V91_timestamp	{(*-*-*), *:*:*.*}	
MultiStateValue	195	CO2 Display - RMUE2715096XSD_70B3D52DD9000265_TX_V95_SEND	1	
OctetStringValue	200	Flood Sensor - KAT47416621ELC_70B3D52DD100115E_RX_V0_reason	X''	
BinaryInput	201	Flood Sensor - KAT47416621ELC_70B3D52DD100115E_RX_V1_flood	inactive	
BinaryInput	202	Flood Sensor - KAT47416621ELC_70B3D52DD100115E_RX_V2_boxTamper	inactive	
AnalogInput	203	Flood Sensor - KAT47416621ELC_70B3D52DD100115E_RX_V3_temperature	0	
AnalogInput	204	Flood Sensor - KAT47416621ELC_70B3D52DD100115E_RX_V4_battery	0	
AnalogInput	240	Flood Sensor - KAT47416621ELC_70B3D52DD100115E_RX_V40_counter	0	

Note on the architecture

[WebUI > Device details > Values], BACnet interface and Modbus interface do not share state. For example, they have their own uplink counter, downlink counter and memory of uplink/downlink values. What they share is the memory of device events and downlink queue.

Note on the downlink implementation:

- fPort is taken from valueMap.fPort, unless overridden by codec.
- Unconfirmed downlinks are used by default. No expiration set.
- Downlink enqueue is controlled by the SEND option.

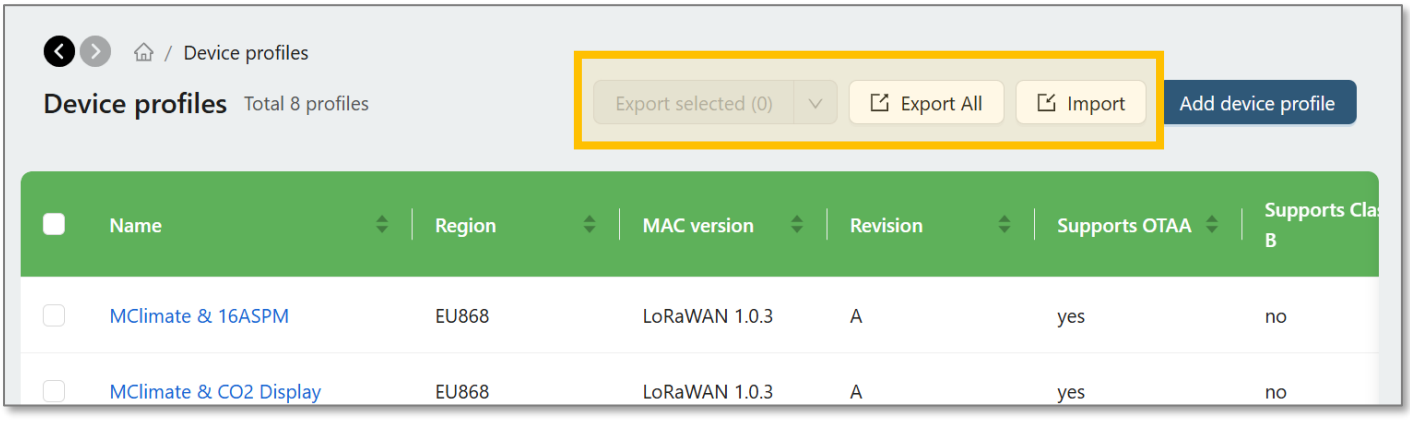
6.6 Device Library

The gateway contains a library of LoRaWAN end-devices with predefined configurations. This library is being continuously extended in new firmware releases. The device library is used as a source for the device profile templates – see Using a template

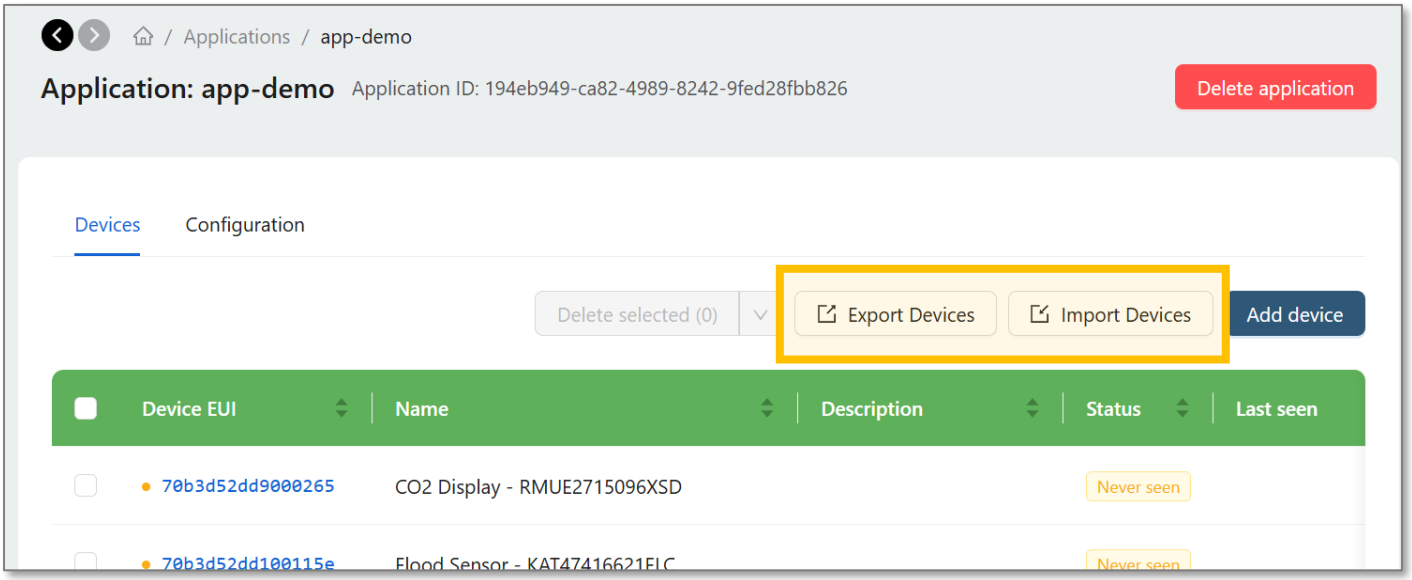
6.7 Backup/Restore

The overall system backup is not supported. You can backup device profiles and devices by exporting them.

Device profiles export/import:



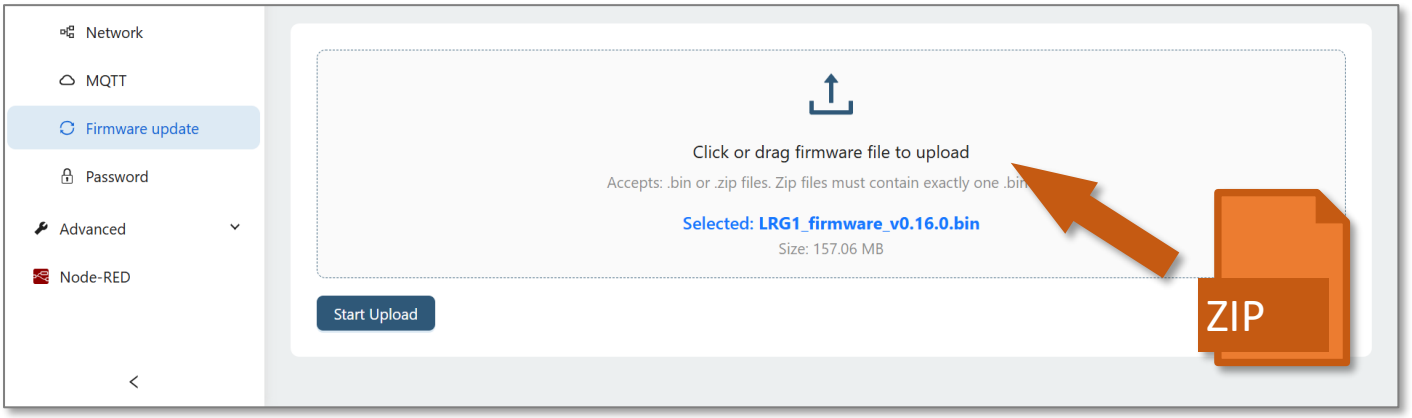
Per-application device export/import:



The exported CSV contains the OTAA keys, keep it secret
These keys are used to secure the communication between the device and application.

6.8 Firmware Update

The firmware is constantly being improved and extended to support new features. The latest version is available for download on FIRVENA website in the Downloads section. Download the file and upload it to the gateway using *WebUI* in [Settings > Firmware update]:



7 Data

7.1 MQTT

The gateway has an internal MQTT broker with two interfaces: **public and private**. The public interface is **0.0.0.0:1883**, it is only used for packet forwarder topics `eu868/gateway/<gateway_id>/#`. The gateway messages correspond to log items recorded in [Advanced > LoRa Gateways > Gateway details > LoRaWAN frames] (http://LRGW.local/#/advanced/lora/gateways/<gateway_id>/frames).

The private interface is **127.0.0.1:1884** and it is not exposed outside the system. This interface exposes application topics with decrypted and decoded data of end-devices. The device topic is `application/<application_id>/device/<device_id>/#`, following the events/commands. The device

messages correspond to log items shown in [Device details > Device events] (http://LRGW.local/#/applications/<application_id>/devices/<device_id>/events). Per-device gateway messages are in [Device details > LoRaWAN frames].

Messages from the local packet forwarder are in the *Protobuf* binary format. The application messages are in JSON format.

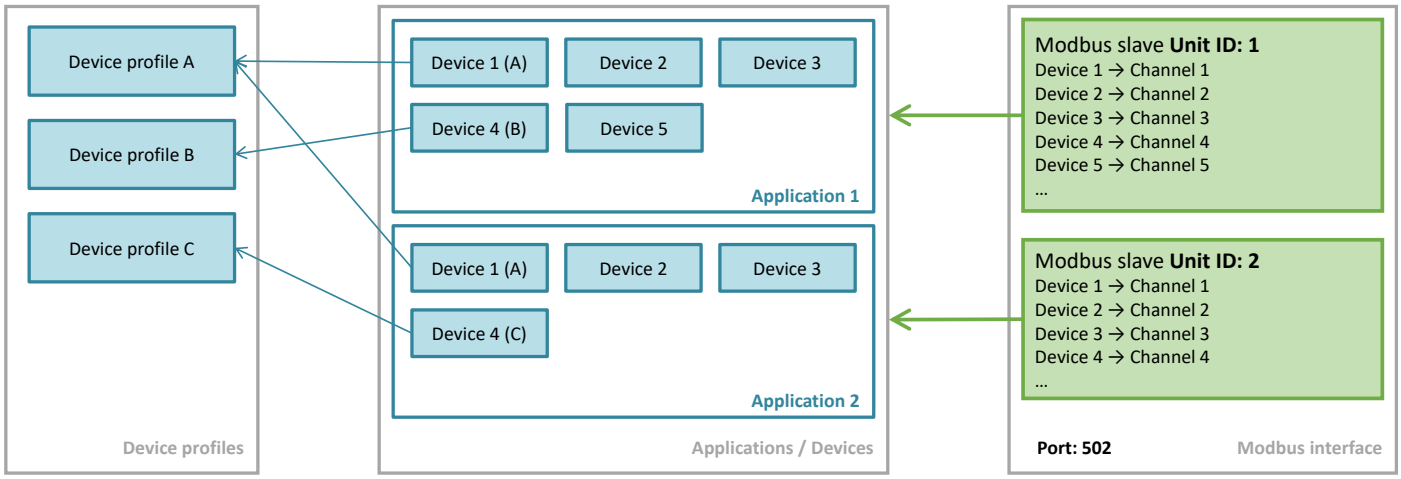
The local MQTT communication is forwarded to an external MQTT broker using a bridge. It is configured in [Settings > MQTT]

7.2 Modbus

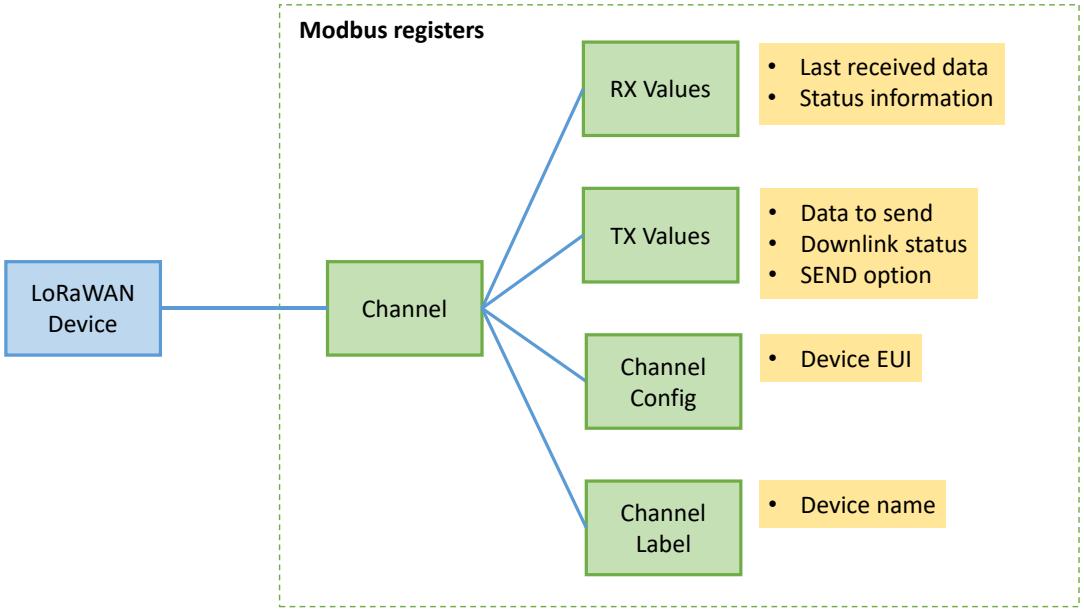
The Modbus configuration is in [WebUI > Modbus].

7.2.1 Mapping of Devices

Applications are mapped to Modbus slaves with different Unit ID, the registers inside a slave are organized into channels, devices are mapped to channels in alphabetical order by device name:

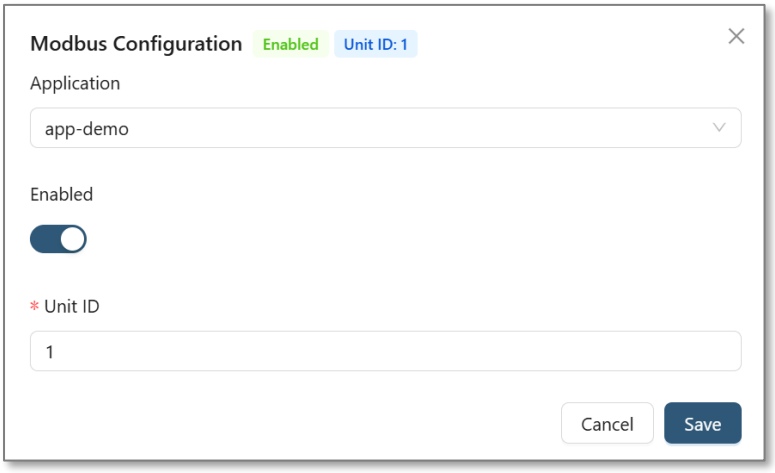


Channel overview:



- RX Values and TX Values are defined in Device profile – see Configuration > Values
- Device name is used as a channel label.

Assignment of Unit ID can be changed in the application configuration in [WebUI > Modbus]:



Each Modbus slave has 200 channels, which means it can handle up to 200 devices – see Modbus Registers

Vocabulary:

- Gateway: the product this manual is about – LORCOM LoRaWAN Gateway
- Device: a LoRaWAN end-device connected to the gateway (temperature sensor, …, whatever needs to be visible through the gateway)
- Application: a group of devices within a gateway
- Channel: a path to the device
- Direction RX/TX: a division within a channel; there are two directions RX (from device, uplink, incoming, receive) and TX (to device, downlink, outgoing, send)
- Value: a data point containing a part of transmitted payload data (temperature, humidity, …)
- Metavalue: a data point containing information that is not part of the payload (counter, timestamp, signal)

7.2.2 Modbus Registers

The table below provides an overview of registers for a single Modbus slave. The contents of RX Values and TX Values registers depend on the type of LoRaWAN device and Device profile configuration. You can export the current list of registers to a CSV file in [WebUI > Modbus]

Area name	Address range	Description	Channel	Base address
RX Values	0…3999	Last received data and channel status	CH1	0
			CH2	20
			…	
			CH200	3980
Reserved				
TX Values	5000…8999	Data to send	CH1	5000
			CH2	5020
			…	
			CH200	8980
Reserved				
Channel Config	10000…13999	Channel configuration (assignment of LoRaWAN device)	CH1	10000
			CH2	10020
			…	
			CH200	13980
Reserved				
Channel Labels	20000…29999	User-defined descriptions of channels	CH1	20000
			CH2	20050
			…	
			CH200	29950
Reserved				

RX Values

There are 20 registers reserved for each channel. When the gateway receives an uplink from the assigned device, it decodes the payload and stores the new values in registers V1, V2, … V15. The number of RX Values and their meaning depends on the type of LoRaWAN device and on the Device profile configuration.

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

Offset	Type	Access	Name	Description
0		R	V1	The contents depend on device profile
1		R	V2	
2		R	V3	
3		R	V4	
…		R		

14		R	V15	
15	U16	R	Uplink counter	Incremented on uplink event, overflows to 1 0: no uplink yet 1…65535: number of uplinks received (65534, 65535 → 1, 2, 3, …)
16	U32	R	Timestamp HI	Unix timestamp of the last uplink (seconds since 1 Jan 1970 00:00:00 UTC)
17			Timestamp LO	
18	S16	R	Signal	Signal strength of the last uplink (RSSI value in dBm)
19				Reserved

TX Values

There are 20 registers reserved for each channel. When the send condition is true, the gateway takes the current values stored in registers V1, V2, … V15, encodes the payload and appends a new downlink to the downlink queue. The number of TX Values and their meaning depends on the type of LoRaWAN device and on the Device profile configuration.

$$\text{Address} = \text{BaseAddress} + \text{Offset}$$
$$\text{BaseAddress} = 5000 + 20 \cdot (\text{ChannelNumber} - 1)$$

Offset	Type	Access	Name	Description
0		RW	V1	The contents depend on device profile
1		RW	V2	
2		RW	V3	
3		RW	V4	
…		RW		
14		RW	V15	
15	U16	R	Downlink counter	Incremented on downlink event, overflows to 1 0: no downlink yet 1…65535: number of downlinks sent (65534, 65535 → 1, 2, 3, …)
16	U32	R	Timestamp HI	Unix timestamp of the last downlink (seconds since 1 Jan 1970 00:00:00 UTC)
17			Timestamp LO	
18	S16	RW	SEND	Send option for the channel – see SEND option
19				Reserved

SEND option:

Value	Description
1: idle (initial state)	Downlink is disabled
2: all_send_now	Send all values The gateway will take all current values from registers V1, V2, … V15 and send them in a single downlink (or as few as possible).
4: single_on_write_any	Send a single value on write event The gateway will be watching for write events on registers V1, V2, … V15, if a register is written, the gateway will send just the written value.

Permanent states: 1, 4; temporary states: 2

Writing 2 executes and returns to the previous permanent state: 1→2→1; 4→2→4

Channel Config

There are 20 registers reserved for each channel. The Device EUI uniquely identifies a LoRaWAN device.

$$\text{Address} = \text{BaseAddress} + \text{Offset}$$
$$\text{BaseAddress} = 10000 + 20 \cdot (\text{ChannelNumber} - 1)$$

Offset	Type	Access	Name	Description	Example*
0	U8[8]	R	Device EUI	Unique identifier of the assigned device, array of bytes b0…b7	0x70B3
1					0xD52D
2					0xD900
3					0x0265
4…9			Reserved		

**Example: Device EUI = 70b3d52dd9000265.*

Channel Labels

There are 50 registers reserved for each channel. These registers are designed to store a friendly name of the assigned LoRaWAN device that helps to identify it. The maximum size of a label is 98 bytes.

$$\text{Address} = \text{BaseAddress} + \text{Offset}$$
$$\text{BaseAddress} = 20000 + 50 \cdot (\text{ChannelNumber} - 1)$$

Offset	Type	Access	Name	Description	Example*
0	U8	R	Label length (N)	Number of bytes in the Label array	28
1…49	U8[98]	R	Label	Array of bytes b0…b(N−1)	0x434F (“CO”)
					0x3220 (“2 ”)
					0x4469 (“Di”)
					0x7370 (“sp”)
					0x6C61 (“la”)
					0x7920 (“y ”)
					0x2D20 (“- ”)
					0x524D (“RM”)
					0x5545 (“UE”)
					0x3237 (“27”)
					0x3135 (“15”)
					0x3039 (“09”)
					0x3658 (“6X”)
					0x5344 (“SD”)
					0x0000
					…
					0x0000

**Example: Label = “CO2 Display - RMUE2715096XSD”.*

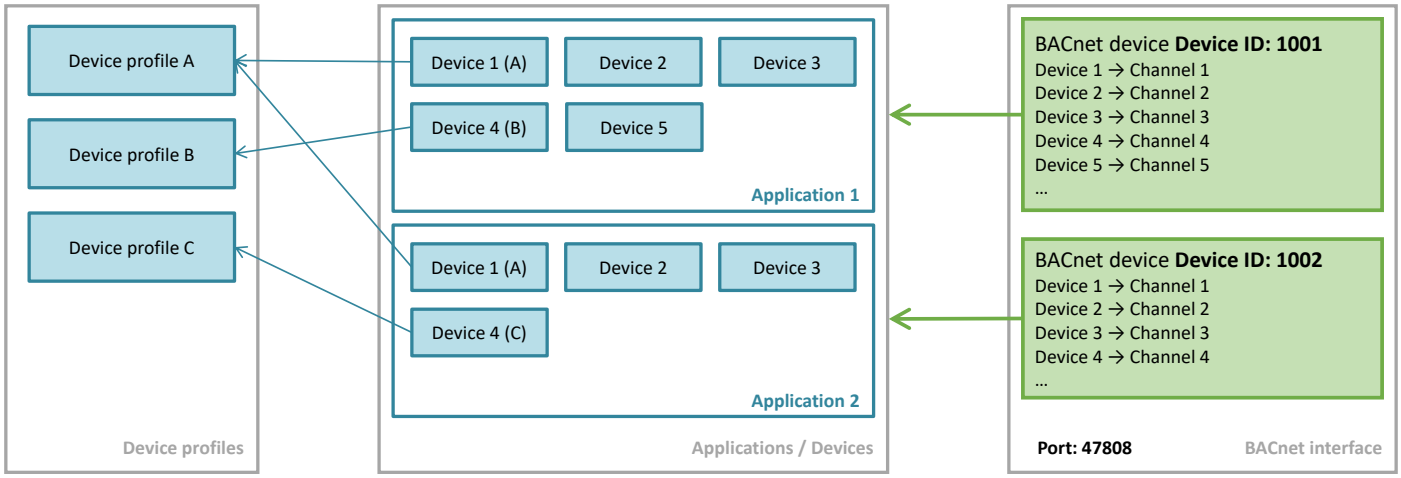
7.3 BACnet

The BACnet configuration is in [WebUI > BACnet].

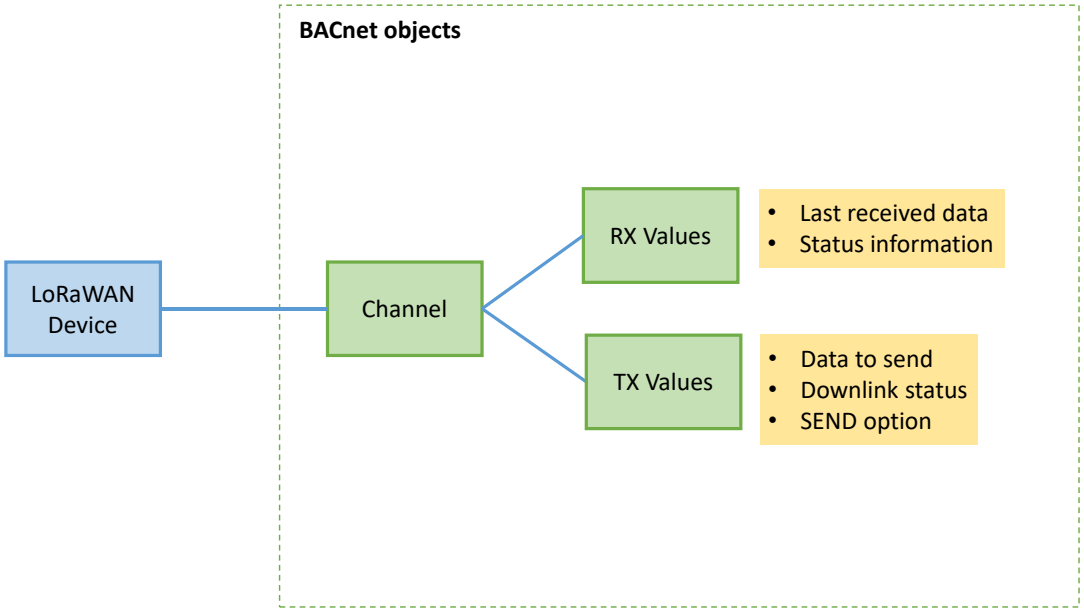
Current implementation allows only one BACnet-enabled application.

7.3.1 Mapping of Devices

Applications are mapped to BACnet devices with different Device ID (Device Instance), the objects inside a BACnet device are organized into channels, LoRaWAN devices are mapped to channels in alphabetical order by device name:



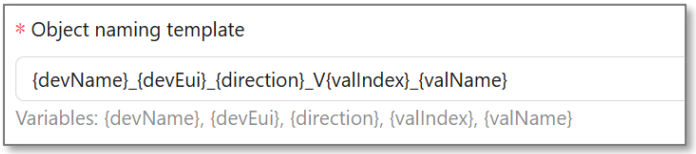
Channel overview:



- RX Values and TX Values are defined in Device profile – see Configuration > Values
- Device name and EUI are used in object names, this ensures the uniqueness of object names.

Object naming template

Object names are created using a template. You can change the template in [WebUI > BACnet > BACnet Settings].



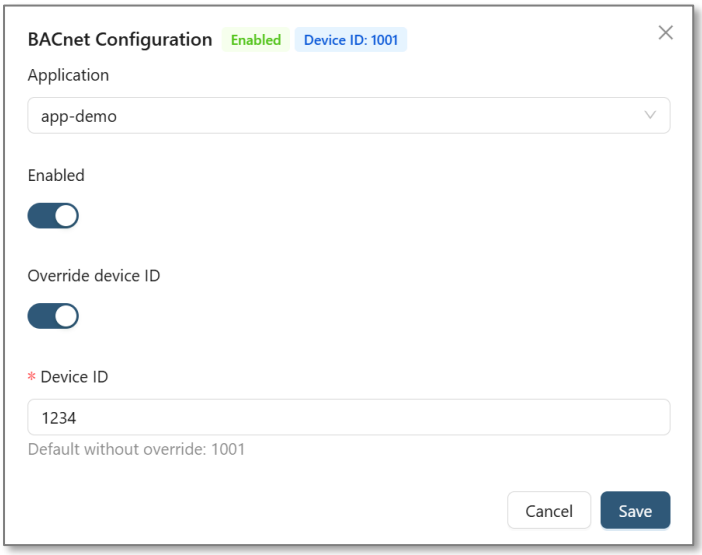
The default object names are designed to have a logical structure that keeps the objects well organized and simplifies searching among a large amount of objects. The uniqueness of object names in the BACnet network is also taken into account to eliminate conflicts.

Example (device name is FL1.105.co2 and FL1.105.light):
FL1.105.co2_70B3D52DD9000265_RX_V0_ppm
FL1.105.co2_70B3D52DD9000265_RX_V1_temperature
FL1.105.co2_70B3D52DD9000265_RX_V2_humidity
…
FL1.105.light_70B3D52DDD0012A3 _RX_V0_relayStateON
FL1.105.light_70B3D52DDD0012A3 _TX_V0_setRelayStateON
FL1.105.light_70B3D52DDD0012A3 _TX_V95_SEND

Device ID

The Device Instance must be unique within a BACnet network. By default, the Device IDs are assigned using the Device ID base, see [WebUI > BACnet > BACnet Settings].

The default assignment of Device ID can be overridden in the application configuration in [WebUI > BACnet]:



Vocabulary:

- Gateway: the product this manual is about – LORCOM LoRaWAN Gateway
- Device: a LoRaWAN end-device connected to the gateway (temperature sensor, …, whatever needs to be visible through the gateway). Do not confuse with BACnet device.
- Application: a group of devices within a gateway
- Channel: a path to the device
- Direction RX/TX: a division within a channel; there are two directions RX (from device, uplink, incoming, receive) and TX (to device, downlink, outgoing, send)
- Value: a data point containing a part of transmitted payload data (temperature, humidity, …)
- Metavalue: a data point containing information that is not part of the payload (counter, timestamp, signal)
- BACnet Device object: an object representing a BACnet device containing other BACnet objects

7.3.2 BACnet objects

The table below provides an overview of BACnet objects for a single BACnet device. The number and contents of RX Values and TX Values objects depend on the type of LoRaWAN device and Device profile configuration.

The BACnet objects are organized into channels. Default channel size is 100 objects. Channel is divided into two directions RX and TX (direction uplink and downlink). Direction is divided into values (payload data) and metavalues (status information like signal strength or message counters). For direction, 50 objects are reserved, 40 for values and 10 for metavalues.

Channel	Base	Direction		Object ID	Description
Reserved	0			0…99	
CH1	100	RX	Values	100…139	Last received data
			Metavalues	140…149	Uplink and channel status information
		TX	Values	150…189	Data to send
			Metavalues	190…199	Downlink status and SEND option
CH2	200	RX	Values	200…239	
			Metavalues	240…249	
		TX	Values	250…289	
			Metavalues	290…299	
CH3	300	RX	Values	300…339	
			Metavalues	340…349	
		TX	Values	350…389	
			Metavalues	390…399	
…	…			…	
CH33	3300	RX	Values	3300…3339	
			Metavalues	3340…3349	
		TX	Values	3350…3389	
			Metavalues	3390…3399	
…	…			…	
CH{N}	{N}00	RX	Values	{N}00…{N}39	Last received data
			Metavalues	{N}40…{N}49	Uplink and channel status information
		TX	Values	{N}50…{N}89	Data to send
			Metavalues	{N}90…{N}99	Downlink status and SEND option
…	…			…	

RX Direction

There are 50 objects reserved for each channel. When the gateway receives an uplink from the assigned device, it decodes the payload and stores the new values in objects V0, V1, … V39. The number of RX Values and their meaning depends on the type of LoRaWAN device and on the Device profile configuration.

$$\text{ObjectID} = \text{Base} + \text{Offset}$$
$$\text{Base} = 100 \cdot \text{ChannelNumber}$$

Offset	Object Type	Value	Description
0		V0	The number, type and contents of these objects depend on device profile
1		V1	
2		V2	
3		V3	
…			
39		V39	
40	AnalogInput	Uplink counter	Incremented on uplink event
41	DateTimeValue	Timestamp	Timestamp of the last uplink
42	AnalogInput	Signal	Signal strength of the last uplink (RSSI value in dBm)
42…49			Reserved

TX Direction

There are 50 objects reserved for each channel. When the send condition is true, the gateway takes the current values stored in objects V0, V1, … V39, encodes the payload and appends a new downlink to the downlink queue. The number of TX Values and their meaning depends on the type of LoRaWAN device and on the Device profile configuration.

$$\text{ObjectID} = \text{Base} + \text{Offset}$$
$$\text{Base} = 100 \cdot \text{ChannelNumber}$$

Offset	Object Type	Value	Description
50		V0	The number, type and contents of these objects depend on device profile
51		V1	
52		V2	
53		V3	
…			
89		V39	
90	AnalogInput	Downlink counter	Incremented on downlink event
91	DateTimeValue	Timestamp	Timestamp of the last downlink
92…94			Reserved
95	MultiStateValue	SEND	Send option for the channel – see SEND option
96…99			Reserved

SEND option:

Value	Description
1: idle (initial state)	Downlink is disabled
2: all_send_now	Send all values The gateway will take all current values from objects V0, V1, … V39 and send them in a single downlink (or as few as possible).
4: single_on_write_any	Send a single value on write event The gateway will be watching for write events on objects V0, V1, … V39, if an object is written, the gateway will send just the written value.

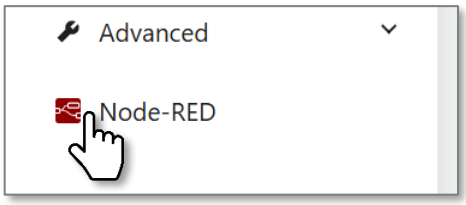
Permanent states: 1, 4; temporary states: 2

Writing 2 executes and returns to the previous permanent state: 1→2→1; 4→2→4

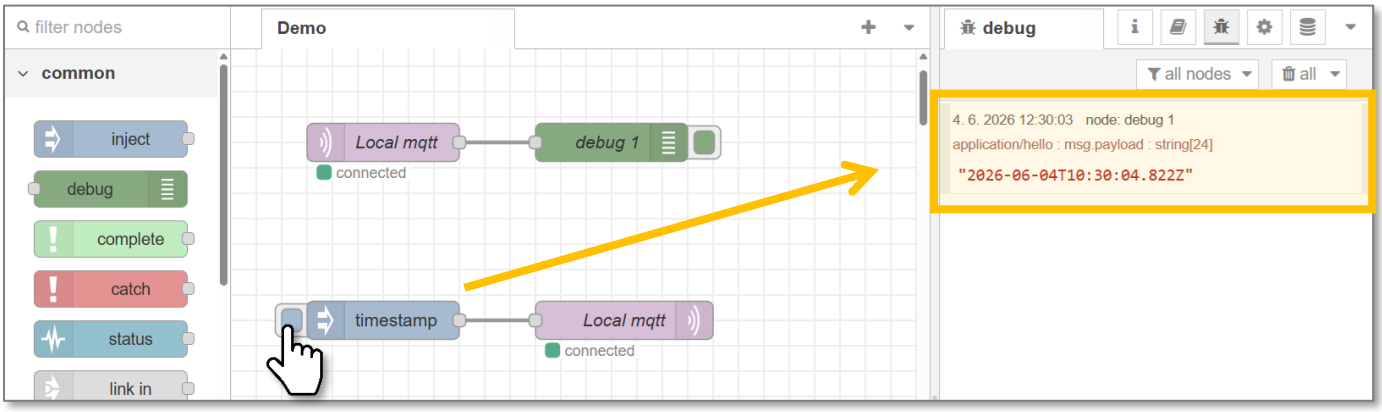
8 Other Features

8.1 Node-RED

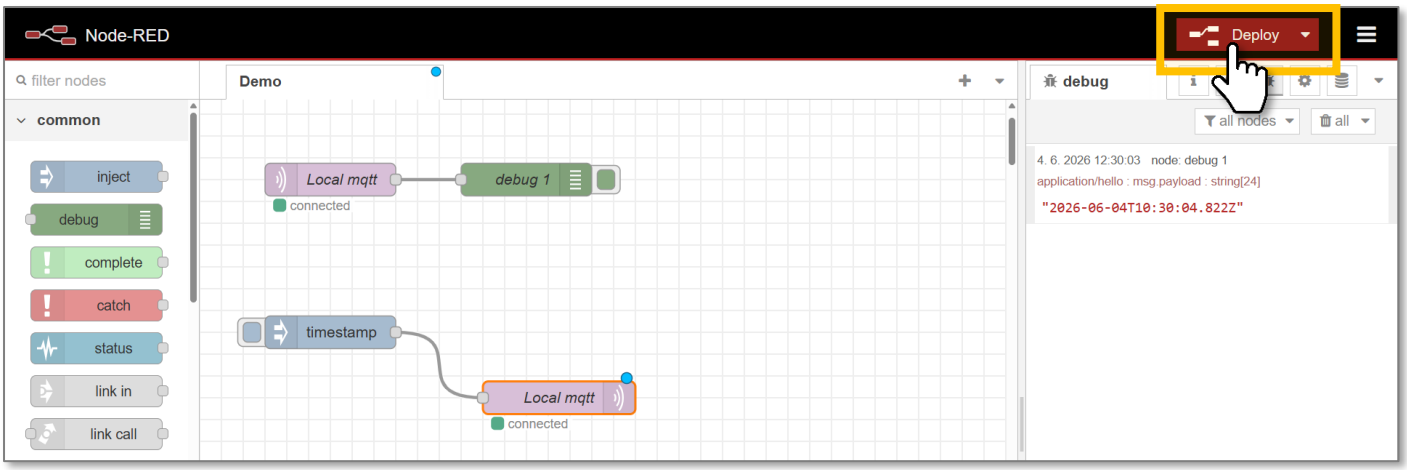
To open the Node-RED development environment, use [WebUI > Node-RED]. It is on the same port as *WebUI* under the root path */nodered*, for example: http://LRGW.local/nodered. The same login as for *WebUI* is required.



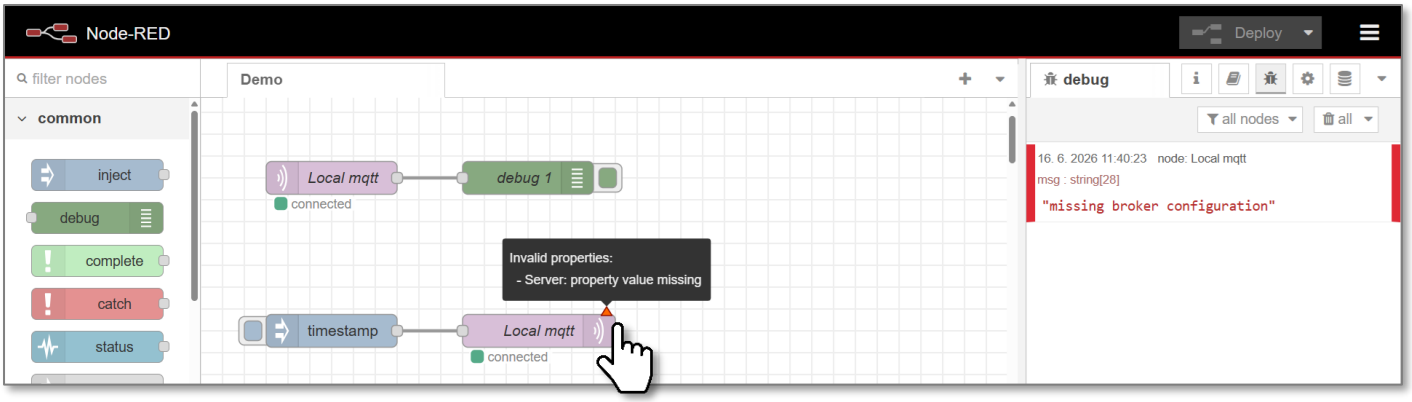
There is a demo showing how the local MQTT broker works:



Any changes must be *"deployed"* before use (changed nodes and flows are marked with a blue dot):



Nodes having an error are marked with a red triangle:



➢ See https://nodered.org for more information about Node-RED.

8.2 Packet Forwarder

[WebUI > Advanced > Packet Forwarder]

The gateway can work as a *Packet forwarder* to an external *Network server*.

Available forwarder types:

Name	Description
UDP Forwarder	UDP Forwarder compatible with the Semtech UDP forwarder protocol. This forwarder is not recommended for production environment. Simple, but weak security and scalability. Software used: https://github.com/chirpstack/chirpstack-udp-forwarder
ChirpStack MQTT Forwarder	Packet forwarder used in ChirpStack ecosystem, it requires an MQTT broker between the gateway and Network server. Simple, security depends on MQTT setup, scalable. Software used: https://github.com/chirpstack/chirpstack-mqtt-forwarder

8.3 Connecting other gateways

[WebUI > Advanced > LoRa Gateways]

Other LoRaWAN gateways can connect to the internal *Network server*. There is always one “local” gateway, which is the packet forwarder running inside the system.

This can be used to extend the range by simple packet forwarders, with this device as a central point collecting the data from edge gateways. The number of gateways must not exceed 10 gateways. The number of end-devices is limited by system resources, which may vary depending on the specific application, the gateway should be able to handle at least 1000 end-devices.

Supported packet forwarders: ChirpStack MQTT Forwarder.

REVISIONS

Date	Version	Description
2026-05-07	V1.0	Initial release
2026-05-19	V1.1	Technical data addition
2026-06-17	V1.2	Style: bold emphasis bolder; uppercase lrgw->LRGW; Update&expand: Accessing Web Interface; Node-RED;
2026-06-23	V1.3	Update&expand: Configuration > Values; Data > Modbus Add: Data > BACnet