



ExpertLoRaWAN

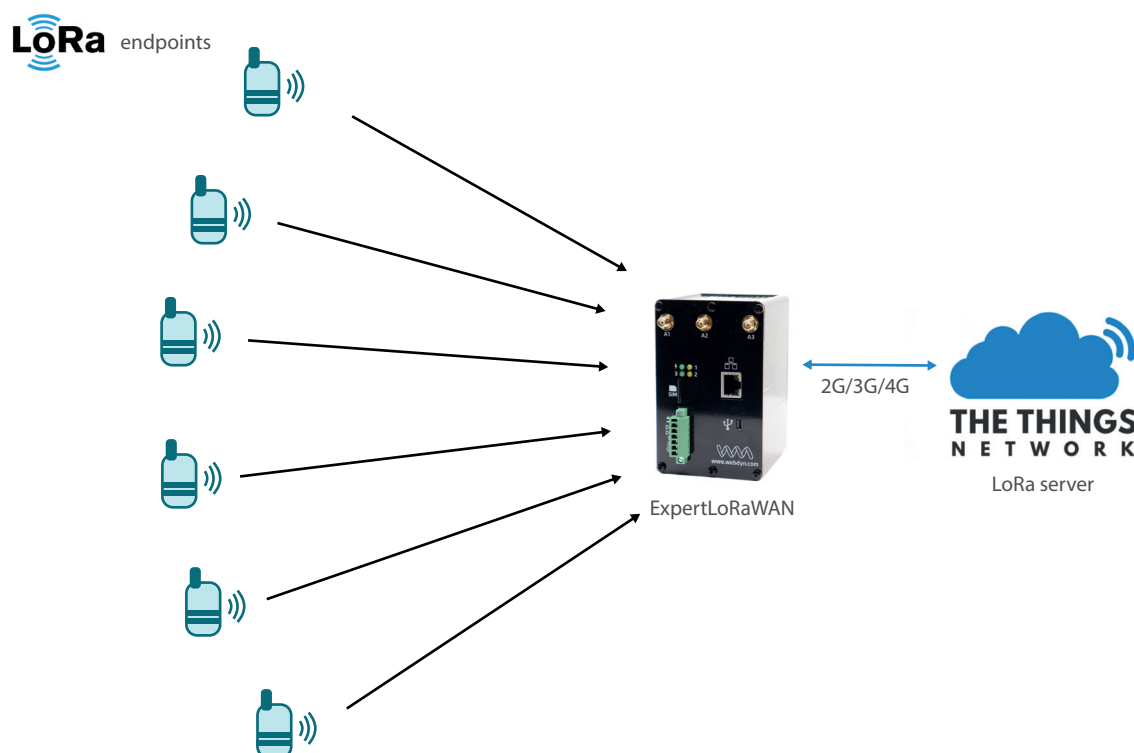
Application Note 1

Create a LoRa network sending data from LoRa sensors to The Things Network LoRa Server

Create a LoRa network sending data from LoRa sensors to The Things Network LoRa Server

1. Scenario Details

The following picture simulates an example scenario with LoRa sensors distributed in a wide area sending data to The Things Network Platform through the Webdyn ExpertLoRaWAN.



This application note shows, step by step, the necessary configurations to be made in the Webdyn ExpertLoRaWAN and The Things Network LoRa Server.

LORA SENSOR

EM500-UDL: distance LoRa sensor

LORA GATEWAY

Webdyn ExpertLoRaWAN: this 4G router has the LoRa-IP Gateway capability (where IP communications can be routed through Ethernet, Wifi or 2G/3G/4G). In the case of the present example, the routing by 2G/3G/4G will be chosen. The Webdyn ExpertLoRaWAN Gateway will collect the LoRa data emitted by the sensors in its range of action and will forward them to the LoRa server in TTN.

LORA SERVER

TTN: the Things Network is a platform that will carry out the LoRa Server mission to manage the LoRa Network. It will be in charge of collecting the data from the sensors relayed by the Webdyn ExpertLoRaWAN Gateway.

2. Configuration of Webdyn ExpertLoRaWAN as LoRa-2G/3G/4G Gateway

The first thing to do on the Webdyn ExpertLoRaWAN Gateway is to configure 2G/3G/4G connectivity. For this, you must enter into the Webdyn ExpertLoRaWAN Gateway configuration web interface through its default IP address 192.168.1.2 and credentials:

User: admin

Password: admin



The image shows the login interface of the Webdyn ExpertLoRaWAN Web Panel Control. At the top, there is a logo for 'webdyn' with a stylized 'w' in grey and teal, and the text 'webdyn' in black. Below the logo is the text 'flexitron group' with a small icon. To the right of the logo, it says 'powered by TITAN' in teal. The main area of the interface is a light grey rectangle containing a login form. The form has two input fields: 'Username:' with the value 'admin' and 'Password:' with the value '*****'. Below these fields is a 'LOGIN' button. At the bottom of the light grey rectangle, the text 'Webdyn ExpertLoRaWan - Web Panel Control' is displayed in teal.

To configure Webdyn ExpertLoRaWAN SIM card network APN information go to WAN -> Basic Setting.

Enable WAN interface and fill the “APN”, “Username” and “Password” fields with the information provided by your Mobile Operator. Please take care about “Sim PIN” (if SIM card is PIN enabled) and keep filled “Call Center” field as showed *99***1#.

webdyn powered by **TITAN**
flexitron group

WAN Basic Settings

☒ **Enabled WAN** Enable GSM WAN interface
 Session Time: Time in minutes (0 = always on)

APN: APN for wireless session
 Username: Username for wireless session
 Password: Password for wireless session

Call center: Call center (normally *99***1#)
 Sim Pin: SIM user pin
 Authentication: Authentication method
 IMSI: If filled, only a valid SIM is allowed
 Network selection: Preferred network selection

DNS selection:
 DNS1: Preferred DNS1
 DNS2: Preferred DNS2

☒ **Remote management** Enable remote management
 Remote TCP Port: TCP Port for remote http connections.

SAVE CONFIG

Then click on “SAVE CONFIG” button and, important, reboot the router using menu

Other->Reboot to allow the router to restart with the new configuration and be able to connect to the Internet.

Once connected to the Internet, go to the configuration menu “Other -> LoRa” and enter the configuration as indicated below, enabling LoRa Gateway option and selecting the “Gateway Lora - Packet Forwarder” mode. You can optionally specify the Latitude/Longitude of where the Gateway will be located.

Then, enter a unique ID for the Gateway with 8 hexadecimal characters (in the example we use the IMEI 3530850900362560). In the gap it indicates “Server address”, enter “eu1.cloud.thethings.network” (if your router will not be in Europe, search TTN for the most appropriate DNS). Finally complete the rest of the parameters as shown in the next screen. Once this screen is configured, save the changes and restart the router.



webdyn

flexitron group

powered by **TITAN**

- Wan
 - Status
 - Basic Settings
 - Keep Online
- LAN
 - Basic Settings
 - DHCP Server
- Firewall
 - NAT
 - Authorized IPs
- Serial Settings
 - Serial Port3-485
 - Serial Port5-USB
 - SSL Certs
- External Devices
 - Logger configuration
 - Temperature Sensor
 - ModBus Devices
 - Distance Sensor
 - Wavenis Concentrator
 - W-MBus Concentrator
 - GPS Receiver
 - Generic Serial Device
 - LoRa**
- VPN
 - OpenVPN Server
 - OpenVPN Client
 - OpenVPN EasyLink
 - IPSec
- Plugins
 - Link
 - Nonat
- Device Manager
 - Cervello
- Other
 - DynDns
 - Private DynDns
 - Digital Input 1

External Devices ▶ LoRa Server

Server Enabled: ☐ Enable LoRa Server

Http Server Port: TCP port for LoRa Webserver

LoRaWAN Band: LoRaWAN regional band configuration

NET ID: Network Identifier (Ex: 010203)

JWT Secret: Password for API

External Devices ▶ LoRa Gateway

Enabled: ☒ Enable LoRa Gateway

Latitude: Optional GPS Latitude. Ex: 40.39924

Longitude: Optional GPS Longitude. Ex: -3.71709

Altitude: Optional GPS Altitude. Ex: 609

LoRa mode: Select the mode of LoRa behaviour

ID: Gateway ID (Ex: 010203040A0B0C0D)

Server address: DNS or IP of server address

UDP server port: (1 ... 66535) Default: 1700

UDP local port: (1 ... 66535) Default: 1700

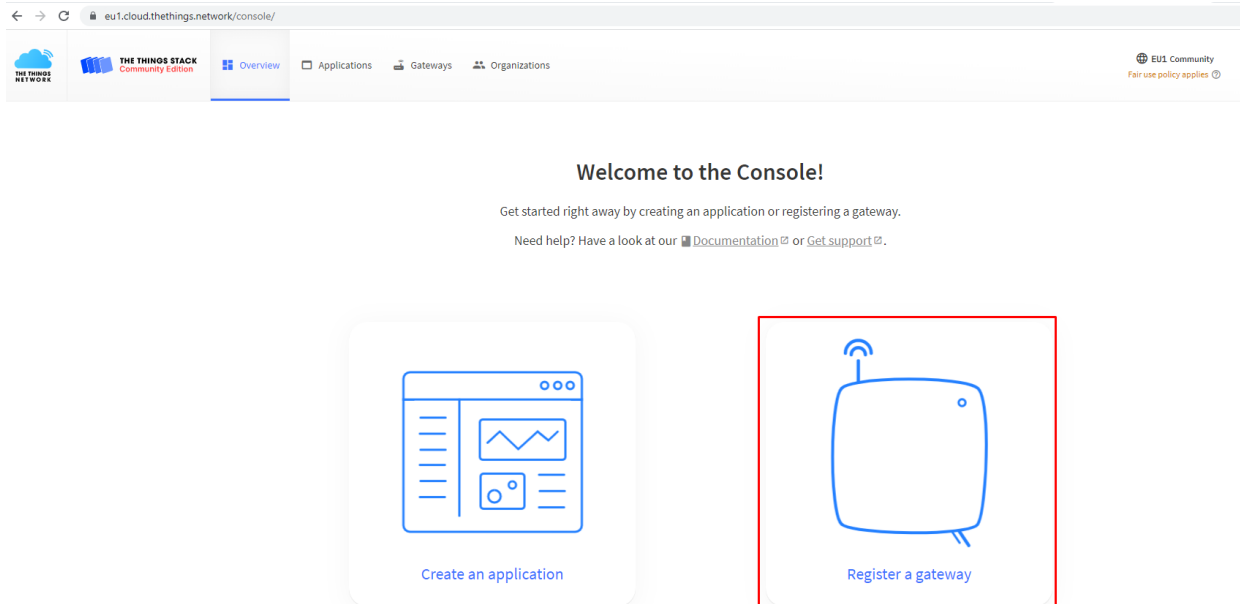
Keepalive: Keepalive interval. Default: 10

Stat interval: Stat interval. Default: 30

Push timeout: Push timeout (ms). Default 100

3. Gateway Registration Configuration in TTN platform

This section assumes that the user has registered at <https://www.thethingsnetwork.org/>. Once registered, you can start building the scenario where the gateways will be added. Choose a cluster in the region you are and then select “Register a gateway” to create one. This will configure the TTN service to accept the connection from our router as a LoRa Gateway.



Now we proceed to enter the Webdyn ExpertLoRaWAN Gateway parameters.

In the “Gateway EUI” box, enter the Gateway ID that was defined during the router configuration process (in the example, 3530850900362560). Choose the working frequency of your application (Europe = 868MHz).

eu1.cloud.thethings.network/console/gateways/add

Add gateway

General settings

Gateway ID ⓘ *

webdyn-expertlora-gw

Gateway EUI ⓘ

35 30 85 09 00 36 25 60

Gateway name ⓘ

Webdyn ExpertLoRa Gateway

Gateway description ⓘ

Webdyn ExpertLoRa Gateway

Optional gateway description; can also be used to save notes about the gateway

Gateway Server address

eu1.cloud.thethings.network

The address of the Gateway Server to connect to

Require authenticated connection ⓘ

☐ Enabled

Controls whether this gateway may only connect if it uses an authenticated Basic Station or MQTT connection

Gateway status ⓘ

☒ Make status public

The status of this gateway may be visible to other users

Gateway location ⓘ

☒ Make location public

When set to public, the gateway location may be visible to other users of the network

Attributes ⓘ

+ Add attributes

Attributes can be used to set arbitrary information about the entity, to be used by scripts, or simply for your own organization

LoRaWAN options

Frequency plan ⓘ *

Europe 863-870 MHz (SF9 for RX2 - recommended)

Once completed the Gateway registration process, after a few minutes the Gateway should appear as “Connected” as shown in the following figure and you can see its live data. At this point, the Webdyn ExpertLoRaWAN Gateway is already correctly connected with the TTN platform.

Overview Applications **Gateways** Organizations

Gateways (1)


Search Claim gateway + Add gateway

ID ⓘ	Name ⓘ	Gateway EUI ⓘ	Status	Created at ⓘ
webdyn-expertlora-gw	Webdyn ExpertLoRa Gateway	35 30 85 09 00 36 25 60	Connected	4 minutes ago

Gateways

Organizations

Gateways > Webdyn ExpertLoRa Gateway



Webdyn ExpertLoRa Gateway
 ID: webdyn-expertlora-gw

↑ 7 ↓ 0

Last activity 59 seconds ago

1 Collaborator

0 API keys

General information

Gateway ID

webdyn-expertlora-gw

Gateway EUI

35 30 85 09 00 36 25 60

Gateway description

Webdyn ExpertLoRa Gateway

Created at

Jun 28, 2022 14:02:42

Last updated at

Jun 28, 2022 14:10:40

Gateway Server address

eu1.cloud.thethings.network

LoRaWAN information

Frequency plan

EU_863_870_TTN

Global configuration

Download global_conf.json

Live data

See all activity →

14:10:40

Update gateway ["antennas", "location_public", "update_locat

14:10:25

Receive gateway status Metrics: { ackr: 0, rxfw: 0, rxin: 0,

14:09:55

Receive gateway status Metrics: { ackr: 0, rxfw: 2, rxin: 2,

14:09:54

Receive uplink message DevAddr: 00 00 04 B8 <> FCnt:

14:09:53

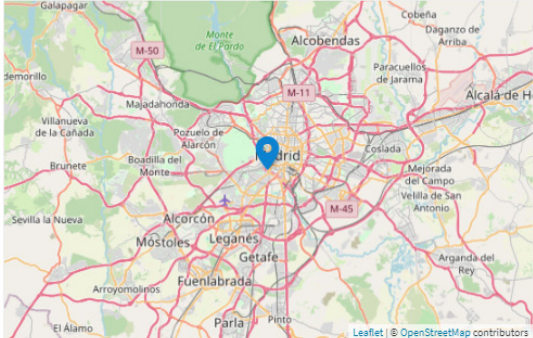
Receive uplink message DevAddr: 00 00 04 B8 <> FCnt:

14:09:25

Receive gateway status Metrics: { ackr: 0, rxfw: 0, rxin: 0,

Location

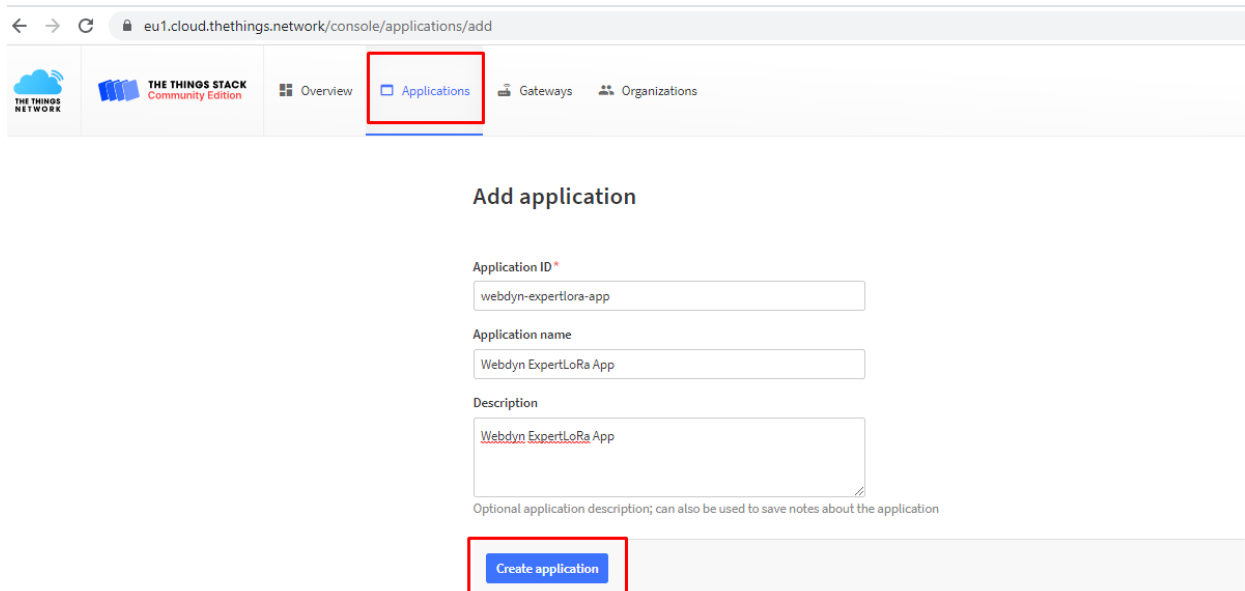
Change location settings →



4. Setting Up an Application in TTN

To register the different LoRa sensors, an application must be created in TTN. To do so, in the console, you must go to the “Applications” menu. Once there, click on the button “Add application”. Complete the Application ID fields (write a unique identifier of your choice).

The next screen shows the data to be completed.

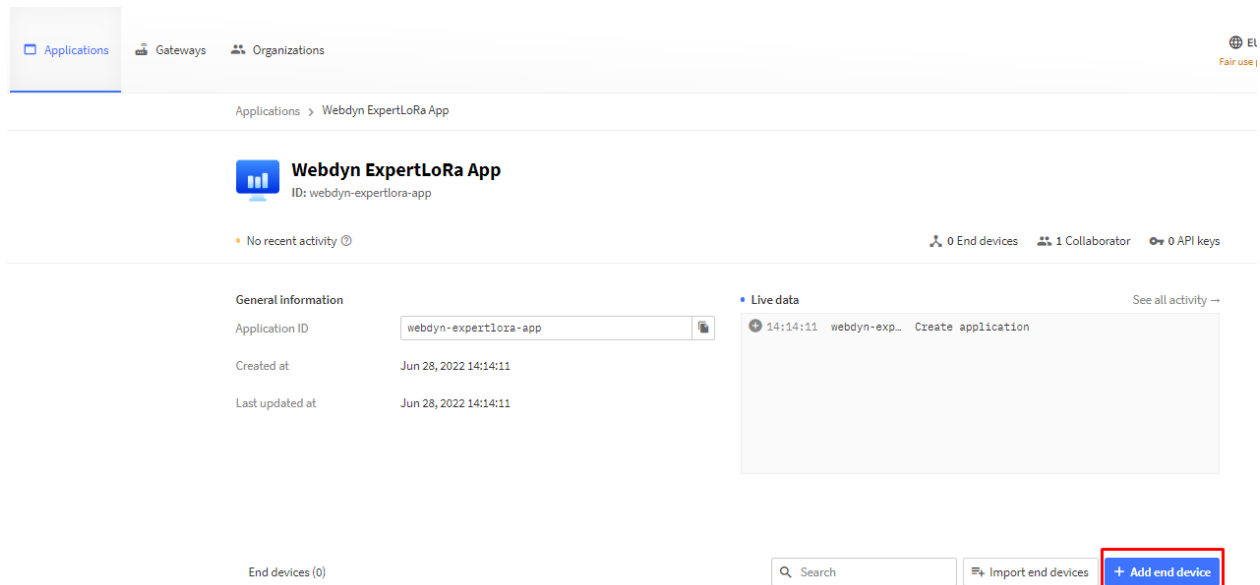


The screenshot shows the TTN console interface at the URL `eu1.cloud.thethings.network/console/applications/add`. The top navigation bar includes 'Overview', 'Applications' (highlighted with a red box), 'Gateways', and 'Organizations'. The main form is titled 'Add application' and contains the following fields:

- Application ID ***: A text input field containing 'webdyn-expertlora-app'.
- Application name**: A text input field containing 'Webdyn ExpertLoRa App'.
- Description**: A text area containing 'Webdyn ExpertLoRa App'.

Below the description field, there is a note: 'Optional application description; can also be used to save notes about the application'. At the bottom of the form, there is a blue button labeled 'Create application', which is highlighted with a red box.

Once the Application is created, we proceed to add a new device.



The screenshot shows the TTN console interface for the 'Webdyn ExpertLoRa App'. The top navigation bar includes 'Applications' (highlighted with a red box), 'Gateways', and 'Organizations'. The main content area displays the application details:

- Application ID**: webdyn-expertlora-app
- Created at**: Jun 28, 2022 14:14:11
- Last updated at**: Jun 28, 2022 14:14:11

Below the details, there is a section for 'End devices (0)'. At the bottom right, there is a blue button labeled '+ Add end device', which is highlighted with a red box.

5. LoRa Sensors Registration in TTN Application

In the same screen above, click on “Add end device” button to register the different LoRa sensors in the newly created application.

You can register the end device from the LoRaWAN device repository or manually if it does not exist there. In this case, for the example, the EM500-UDL sensor is in the repository, so we just need to fill the frequency plan and the DevEUI (identifying number printed on the attached label of each sensor), AppEUI and AppKey parameters. This data is available in the sensor’s user manual and can be configured in the mobile app.

Register end device

[From The LoRaWAN Device Repository](#) [Manually](#)

1. Select the end device

Brand ⓘ *

Model ⓘ *

Hardware Ver. ⓘ *

Firmware Ver. ⓘ *

Profile (Region) *


Milesight IoT Co., Ltd | ▾

EM500-UDL | ▾

V1.3 | ▾

2.28 | ▾

EU_863_870 | ▾



EM500-UDL
LoRaWAN Specification 1.0.3, RP001 Regional Parameters 1.0.3 revision A, Over the air activation (OTAA), Class A
The Milesight EM500-UDL consists of an ultrasonic distance/level sensor that provides highly accurate distance measurement for a variety of applications in harsh environments like snow level monitoring, flood monitoring, wastewater management, fill level monitoring in grains or fertilizer. The measurement is based on ultrasonic waves transmitted and reflected back from nearby objects.
[Product website](#) | [Data sheet](#)

2. Enter registration data

Frequency plan ⓘ *

Europe 863-870 MHz (SF9 for RX2 - recommended) | ▾

AppEUI ⓘ *

24 E1 24 C0 00 2A 00 01 | Fill with zeros

DevEUI ⓘ *

24 E1 24 12 6A 21 74 74 | [Generate](#) 0/50 used

AppKey ⓘ *

55 72 40 4C 69 6E 6B 4C 6F 52 61 32 30 31 38 23 | [Generate](#)

End device ID ⓘ *

eui-24e124126a217474

This value is automatically prefilled using the DevEUI

After registration

☒ View registered end device

☐ Register another end device of this type

[Register end device](#)

In this case, by creating the Milesight end device from the repository, it already extracts the script to decode the payload data of the LoRa frame, otherwise this can be added in the Payload formatters section. The next picture shows the live data from the sensor. Check the sensor user manual to configure additional settings.

The screenshot shows the Webdyn ExpertLoRa App interface. The sidebar on the left contains navigation options: Overview, End devices, Live data, Payload formatters, Integrations, Collaborators, API keys, and General settings. The main area displays the 'Live data' tab for the device 'eui-24e124126a217474'. The 'Data preview' column shows a list of messages with their corresponding DevAddr, Rx Delay, and decoded payload information. The payload information includes distance, battery level, and RSSI values.

Time	Type	Data preview
09:04:13	Schedule data downlink for transmissi...	DevAddr: 26 08 29 5C <> Rx Delay: 5
09:04:13	Forward uplink data message	DevAddr: 26 08 29 5C <> Payload: { distance: 0.3 ; 03 82 2C 01 <> FPort: 85 Data rate: SF7BW125 SNR: 7.2 RSSI: -27
09:04:13	Successfully processed data message	DevAddr: 26 08 29 5C <>
09:02:13	Schedule data downlink for transmissi...	DevAddr: 26 08 29 5C <> Rx Delay: 5
09:02:13	Forward uplink data message	DevAddr: 26 08 29 5C <> Payload: { distance: 0.3 ; 03 82 2C 01 <> FPort: 85 Data rate: SF7BW125 SNR: 9.5 RSSI: -19
09:02:13	Successfully processed data message	DevAddr: 26 08 29 5C <>
09:00:13	Schedule data downlink for transmissi...	DevAddr: 26 08 29 5C <> Rx Delay: 5
09:00:13	Forward uplink data message	DevAddr: 26 08 29 5C <> Payload: { distance: 0.3 ; 03 82 2C 01 <> FPort: 85 Data rate: SF7BW125 SNR: 6.8 RSSI: -19
09:00:13	Successfully processed data message	DevAddr: 26 08 29 5C <>
08:58:43	Schedule data downlink for transmissi...	DevAddr: 26 08 29 5C <> Rx Delay: 5
08:58:42	Forward uplink data message	DevAddr: 26 08 29 5C <> Payload: { battery: 100, distance: 0.3 ; 01 75 64 03 82 2C 01 03 <> FPort: 85 Data rate: SF7BW125 SNR: 9.8 RSSI: -17
08:58:42	Successfully processed data message	DevAddr: 26 08 29 5C <>
08:58:26	Schedule data downlink for transmissi...	DevAddr: 26 08 29 5C <> Rx Delay: 5
08:58:26	Update end device	["activated_at"]
08:58:26	Forward uplink data message	DevAddr: 26 08 29 5C <> Payload: { ; FF 08 FF FF 01 01 FF 16 <> FPort: 85 Data rate: SF9BW125 SNR: 12.8 RSSI: -19
08:58:26	Successfully processed data message	DevAddr: 26 08 29 5C <>
08:58:22	Forward join-accept message	DevAddr: 26 08 29 5C <>