



Connecting OpenSmartMonitor over LoRaWAN

Devtank Ltd.

Marcus Holder

27.09.2022

A short instruction guide to configure an OSM sensor to the three major LoRaWAN networks: Chirpstack, TheThingsNetwork and Helium.



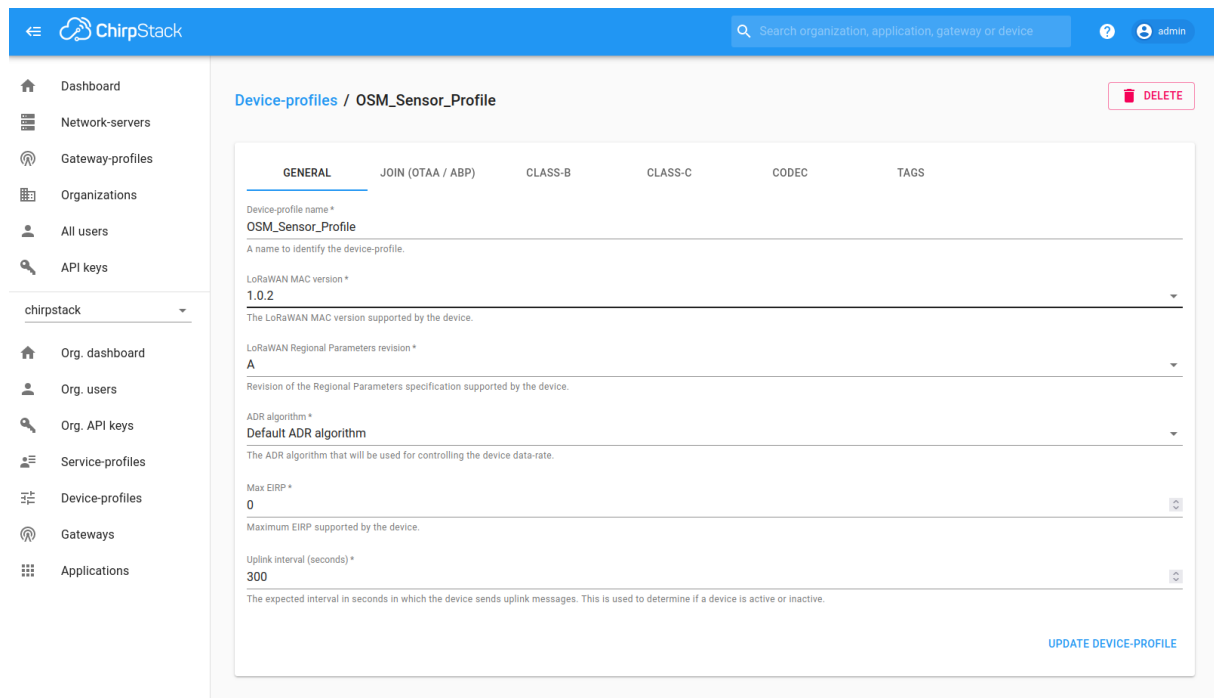
Contents

1. ChirpStack	3
1.1 Set up	3
1.2 Add devices	4
2. TheThingsNetwork	6
2.1 Set up	6
2.2 Add devices	7
3. Helium	10
3.1 Set up	10
3.2 Add devices	11

1. ChirpStack

1.1 Set up

- Before accessing the ChirpStack console, you must install it on your machine. See instructions on this at <https://www.chirpstack.io/docs/>
- Once you have this set up, access the console.
- Default login should be admin:admin but should be changed upon login.
- To add a gateway, select 'Gateways' and press 'Create'.
- You will need a device profile, navigate to 'Device-profiles' on the left panel of the console. Fill in all of the fields under the 'General' tab.

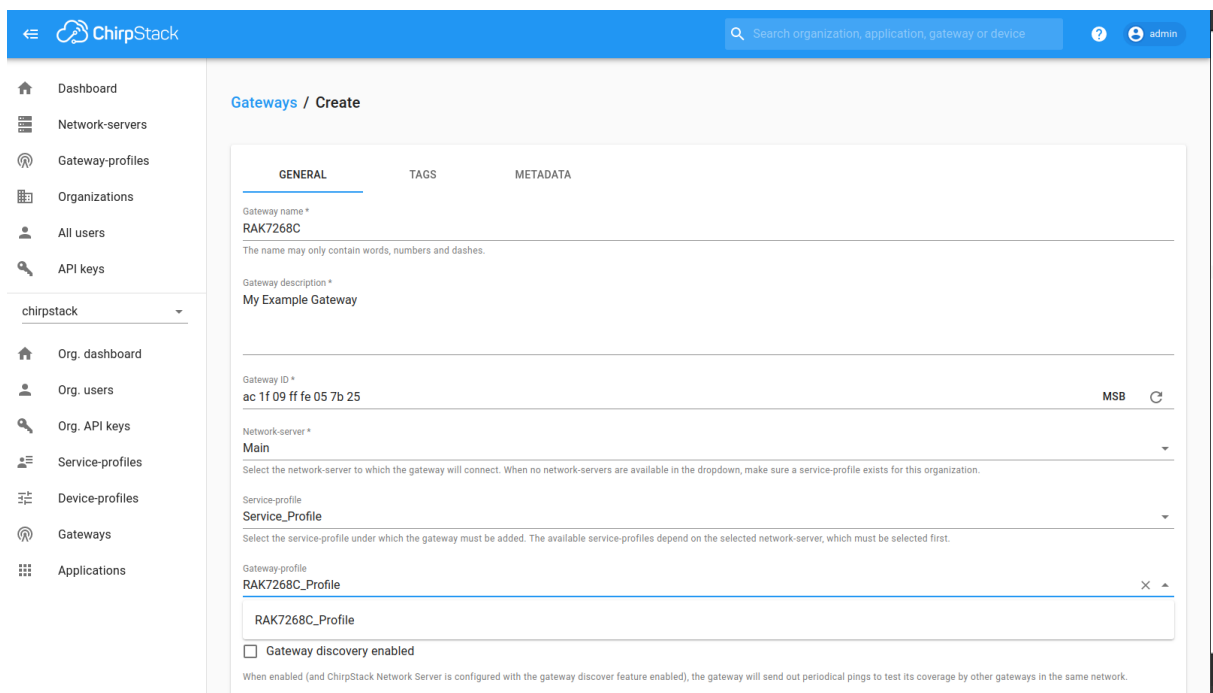


The screenshot shows the ChirpStack web interface. The left sidebar contains navigation options: Dashboard, Network-servers, Gateway-profiles, Organizations, All users, API keys, and a dropdown menu for 'chirpstack' with sub-items: Org. dashboard, Org. users, Org. API keys, Service-profiles, Device-profiles, Gateways, and Applications. The main content area is titled 'Device-profiles / OSM_Sensor_Profile' and features a 'DELETE' button. Below the title are tabs for 'GENERAL', 'JOIN (OTAA / ABP)', 'CLASS-B', 'CLASS-C', 'CODEC', and 'TAGS'. The 'GENERAL' tab is selected and contains the following fields:

- Device-profile name * **OSM_Sensor_Profile**
- A name to identify the device-profile.
- LoRaWAN MAC version * **1.0.2**
- The LoRaWAN MAC version supported by the device.
- LoRaWAN Regional Parameters revision * **A**
- Revision of the Regional Parameters specification supported by the device.
- ADR algorithm * **Default ADR algorithm**
- The ADR algorithm that will be used for controlling the device data-rate.
- Max EIRP * **0**
- Maximum EIRP supported by the device.
- Uplink interval (seconds) * **300**
- The expected interval in seconds in which the device sends uplink messages. This is used to determine if a device is active or inactive.

An 'UPDATE DEVICE-PROFILE' button is located at the bottom right of the form.

- To submit a custom JavaScript decoder, navigate to the Codec tab in Device-Profiles.
- Choose Custom JavaScript codec functions and enter the script, finally select 'Update Device-Profile'.
- Give the gateway a name and description depending on the model you are using.
- Provide the gateway ID, network server, service profile and gateway profile.



The screenshot shows the ChirpStack web interface for creating a new gateway. The left sidebar contains navigation options like Dashboard, Network-servers, Gateway-profiles, Organizations, All users, API keys, and a dropdown for 'chirpstack'. The main content area is titled 'Gateways / Create' and has three tabs: GENERAL, TAGS, and METADATA. The GENERAL tab is selected and contains the following fields:

- Gateway name ***: RAK7268C (Note: The name may only contain words, numbers and dashes.)
- Gateway description ***: My Example Gateway
- Gateway ID ***: ac 1f 09 ff fe 05 7b 25 (with a refresh icon and 'MSB' label)
- Network-server ***: Main (with a dropdown arrow and a note: 'Select the network-server to which the gateway will connect. When no network-servers are available in the dropdown, make sure a service-profile exists for this organization.')
 - Service-profile**: Service_Profile (with a dropdown arrow and a note: 'Select the service-profile under which the gateway must be added. The available service-profiles depend on the selected network-server, which must be selected first.')
 - Gateway-profile**: RAK7268C_Profile (with a dropdown arrow and a note: 'Select the service-profile under which the gateway must be added. The available service-profiles depend on the selected network-server, which must be selected first.')
 - Gateway discovery enabled (with a note: 'When enabled (and ChirpStack Network Server is configured with the gateway discover feature enabled), the gateway will send out periodical pings to test its coverage by other gateways in the same network.')

- 'Gateway discovery enabled' should be unchecked.
- The location can be set manually here if not set on the gateway.
- Click Save and Apply.
- Refresh the page until you see that there has been communication with the gateway.

1.2 Add devices

- Select Applications in the left hand panel.
- Select 'Create' and provide a name, description and provide a service profile.
- Within your application, you can add integration's such as a database to store your device data.
- To add a device, select 'Create' within your application.

Applications / SmartFactory / Devices / Create

GENERAL VARIABLES TAGS

Device name *
Example Sensor

The name may only contain words, numbers and dashes.

Device description *
My First Sensor

Device EUI *
4d b1 fe 43 7e b5 50 72 MSB ↻

Device-profile *
OSM_Sensor_Profile

Disable frame-counter validation
Note that disabling the frame-counter validation will compromise security as it enables people to perform replay-attacks.

Device is disabled
ChirpStack Network Server will ignore received uplink frames and join-requests from disabled devices.

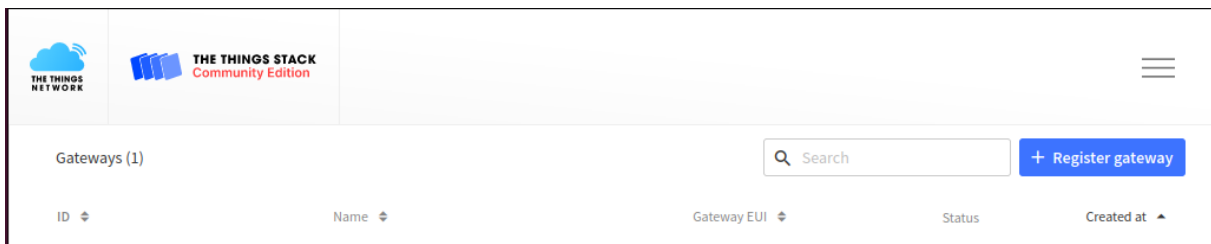
CREATE DEVICE

- Fill in the required fields and select 'create device'.
- In the next window, generate an application key and select 'set device-keys'.

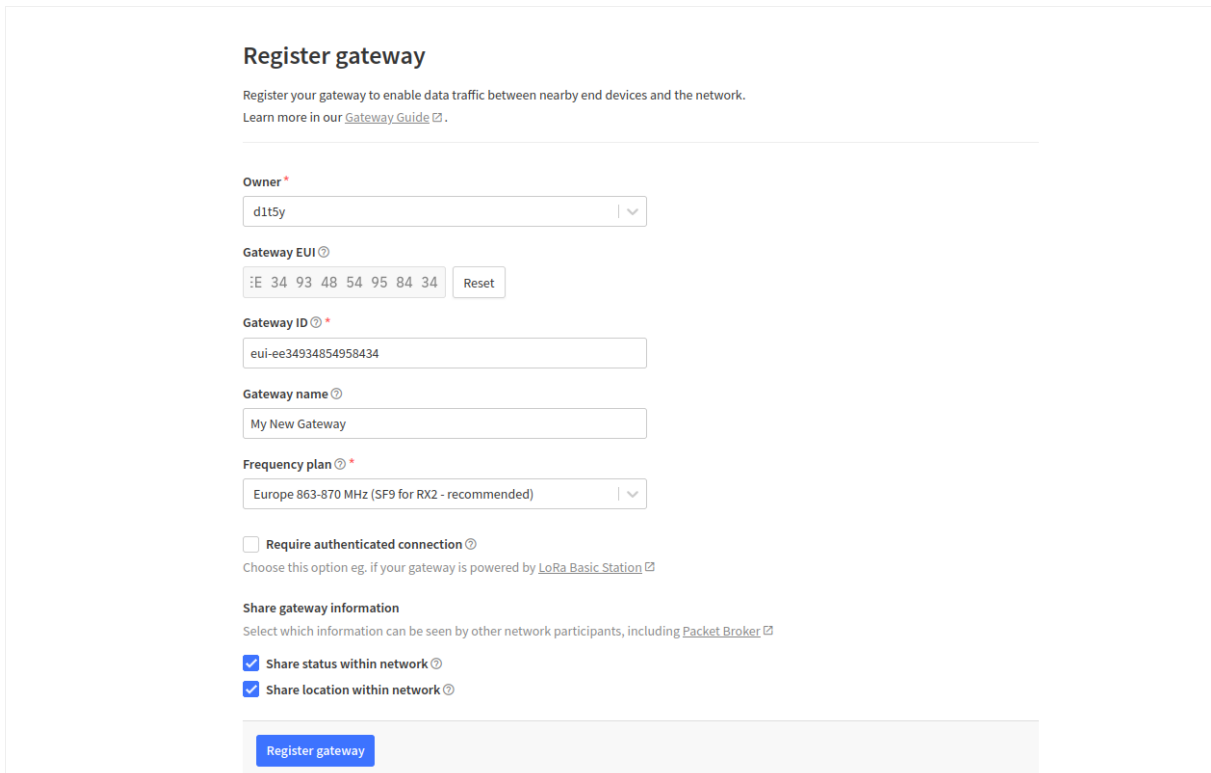
2. The Things Network

2.1 Set up

- Access The Things Stack console at <https://console.cloud.thethings.network>.
- Select a cluster depending on your location e.g Europe 1.
- You will need to register your gateway, you can do this by navigating to 'Go to gateways'.
- Select 'Register Gateway'



- Provide an Owner and the Gateway EUI.
- Provide a gateway name and select a frequency plan.

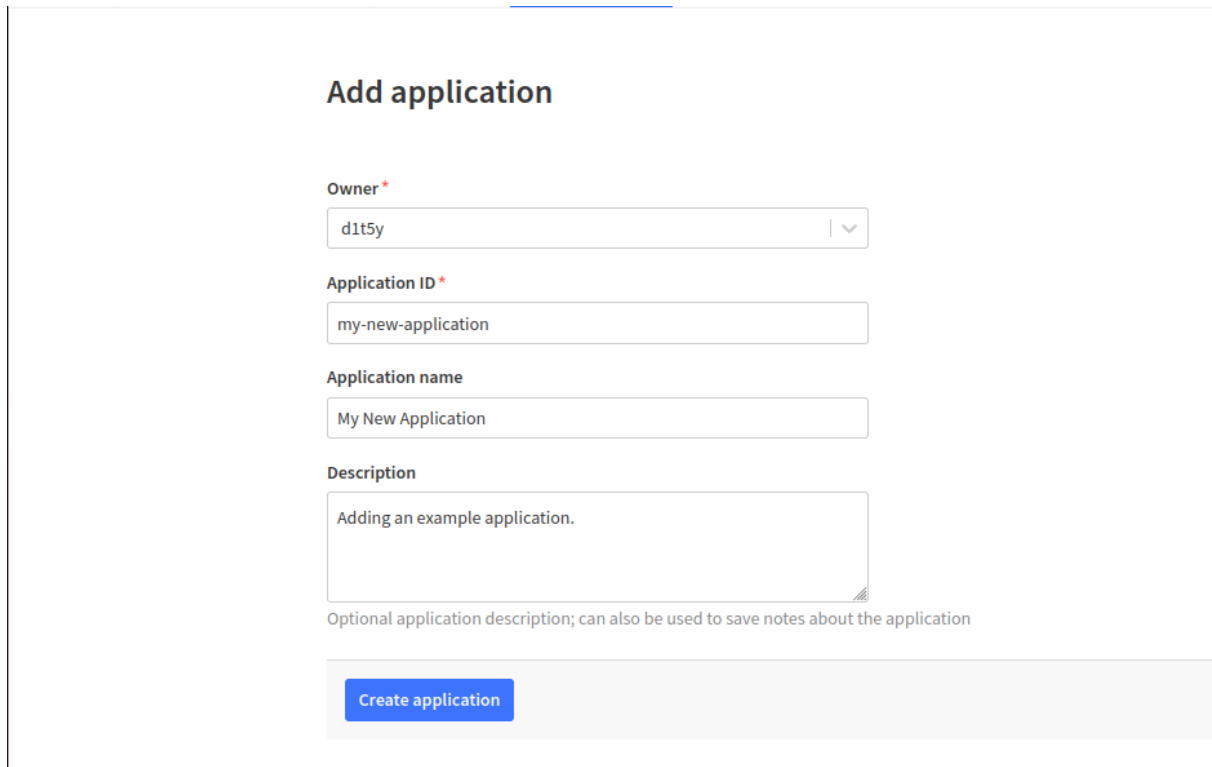


The screenshot shows the 'Register gateway' form. It includes fields for Owner (dropdown), Gateway EUI (text input with a Reset button), Gateway ID (text input), Gateway name (text input), and Frequency plan (dropdown). There is also a checkbox for 'Require authenticated connection' and a section for 'Share gateway information' with checkboxes for 'Share status within network' and 'Share location within network'. A 'Register gateway' button is at the bottom.

- Check the documentation of your gateway provider for any additional configuration you may need to connect your gateway.

2.2 Add devices

- Select the 'Applications' header followed by 'Add application'.
- Provide an Owner, Application ID, Application name and Description.



The screenshot shows a web form titled "Add application". It contains the following fields:

- Owner ***: A dropdown menu with "d1t5y" selected.
- Application ID ***: A text input field containing "my-new-application".
- Application name**: A text input field containing "My New Application".
- Description**: A text area containing "Adding an example application." Below the text area is a note: "Optional application description; can also be used to save notes about the application".

At the bottom of the form is a blue button labeled "Create application".

- Your application should have been created. Inside it select 'Add end device'.
- To add the devices configuration, select the heading 'Manually'.
- Provide a frequency plan, select LoRaWAN version 1.0.0.
- Generate or enter the DevEUI and AppEUI of the sensor. Note this down.
- The AppEUI should be the same as the DevEUI.
- Generate or enter the AppKey of the device. Note this down.
- Select 'Register end device' to finalise.

From The LoRaWAN Device Repository [Manually](#)

Frequency plan ⓘ *

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

LoRaWAN version ⓘ *

LoRaWAN Specification 1.0.0 | ▾

Regional Parameters version ⓘ *

TS001 Technical Specification 1.0.0 | ▾

[Show advanced activation, LoRaWAN class and cluster settings](#) ▾

DevEUI ⓘ *

70 B3 D5 7E D0 05 5C 5B 1/50 used

AppEUI ⓘ *

70 B3 D5 7E D7 05 5C 5B

AppKey ⓘ *

1F 96 05 7C 1A 27 61 88 73 D9 88 FA BC 5D A4 65

End device ID ⓘ *

eui-70b3d57ed0055c5b

This value is automatically prefilled using the DevEUI

After registration

View registered end device

Register another end device of this type

- You should see activity on the overview page if the gateway and device are alive.

Overview Applications Gateways Organizations

Applications > My New Application > End devices > eui-fdcbfa0b3f005ea8

eui-fdcbfa0b3f005ea8
ID: eui-fdcbfa0b3f005ea8

↑4 ↓4 • Last activity 1 minute ago

Overview Live data Messaging Location Payload formatters Claiming General settings

General information

End device ID: eui-fdcbfa0b3f005ea8

Frequency plan: Europe 863-870 MHz, 6 channels for roaming...

LoRaWAN version: LoRaWAN Specification 1.0.0

Regional Parameters version: TS001 Technical Specification 1.0.0

Created at: Sep 21, 2022 14:45:26

Live data

- 11:11:14 Schedule data downlink f
- 11:11:14 Forward uplink data mess
- 11:11:14 Successfully processed d
- 11:08:15 Schedule data downlink f
- 11:08:14 Forward uplink data mess
- 11:08:14 Successfully processed d

- To submit the protocol to decode data from your device, head to 'Payload Formatters' on the left panel and select 'Uplink'.
- Select Custom JavaScript formatter under 'Formatter Type'.
- To apply this protocol to each device, go to 'End Devices' - 'Payload Formatters' - 'Uplink' - 'Formatter type' (Custom Javascript formatter) - 'Paste Application formatter' - 'Save changes'.

My New Application

Overview Live data Messaging Location Payload formatters Claiming General settings

Uplink Downlink

Setup

Formatter type*: Custom Javascript formatter

Formatter code*

```

267 function decodeUplink(bytes)
268 {
269   var pos = 0;
270   var data = {};
271
272   var protocol_version = bytes.bytes[pos++];
273
274   if (protocol_version != 1)
275   {
276     return data;
277   }
278
279   var name;
280   while(pos < bytes.bytes.length)
281   {
282     name = "";
283     for (var i = 0; i < 4; i++)
284     {
285       if (bytes.bytes[pos] != 0)
286       {
287         name += String.fromCharCode(bytes.bytes[pos]);
288       }
289       pos++;

```

Test

Byte payload: 02 BD 01 02 5F 01 02 17 02 00 FPort: 1 Test decoder

Decoded test payload

```

{
  "BAT": 10000,
  "BAT_max": 10000,
  "BAT_min": 10000,
  "CCI": 291,
  "CCI_max": 298,
  "CCI_min": 285,
  "CNT2": 0,
  "HUMI": 4259,
  "HUMI_max": 4273,
  "HUMI_min": 4245,
  "LGHT": 0,

```

Complete uplink data

```

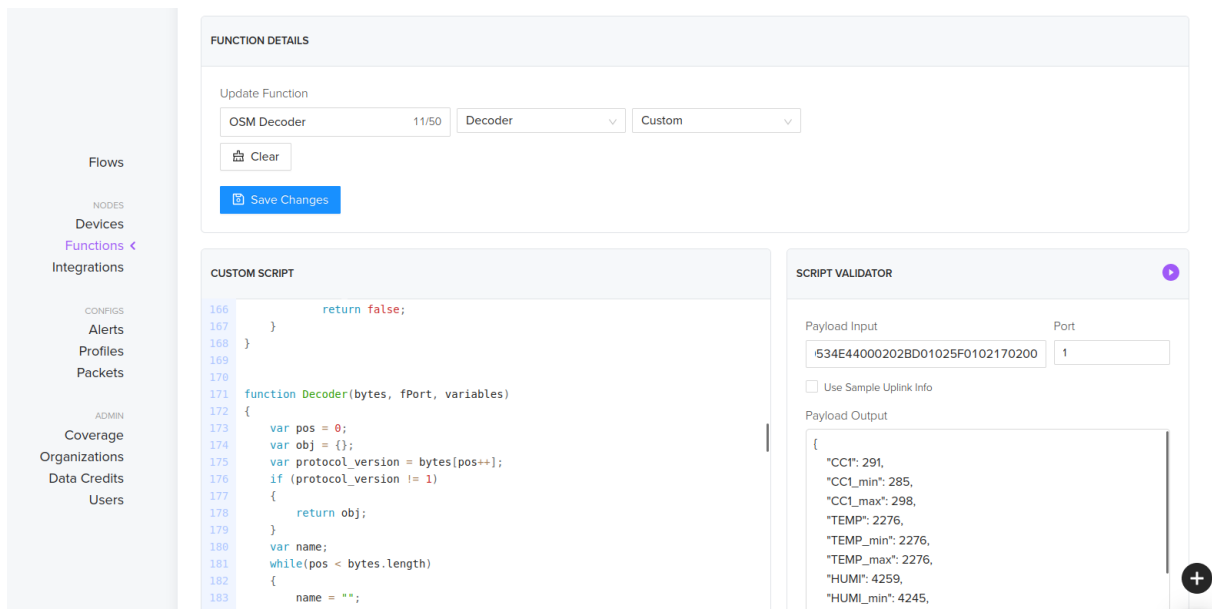
{
  "f_port": 1,
  "frm_payload": "AUNDMQACA1MBAH0BA1oBVEVNUAIC5AgCSA9IVU1JAgKJEAKVEAKXEEJ8V",
  "decoded_payload": {
    "BAT": 10000,
    "BAT_max": 10000,
    "BAT_min": 10000,
    "CCI": 291,
    "CCI_max": 298,
    "CCI_min": 285,

```

3. Helium

3.1 Set up

- Open up the helium console at <https://console.helium.com>.
- Sign up. You will be asked to verify your email address, the link from this email will redirect you to the console.
- Create an organisation when prompted to.
- To add a custom JavaScript protocol, navigate to 'Functions' under 'Nodes' and enter the custom script.



The screenshot displays the Helium console interface. On the left is a navigation sidebar with categories: Flows, NODES (Devices, Functions, Integrations), CONFIGS (Alerts, Profiles, Packets), and ADMIN (Coverage, Organizations, Data Credits, Users). The main area is titled 'FUNCTION DETAILS' and includes an 'Update Function' section with dropdowns for 'OSM Decoder' (11/50), 'Decoder', and 'Custom', along with 'Clear' and 'Save Changes' buttons. Below this is a 'CUSTOM SCRIPT' editor with the following code:

```

166         return false;
167     }
168 }
169
170
171 function Decoder(bytes, fPort, variables)
172 {
173     var pos = 0;
174     var obj = {};
175     var protocol_version = bytes[pos++];
176     if (protocol_version != 1)
177     {
178         return obj;
179     }
180     var name;
181     while(pos < bytes.length)
182     {
183         name = "";
184         ...

```

To the right of the script editor is a 'SCRIPT VALIDATOR' panel. It features a 'Payload Input' field containing the hex string '534E44000202BD01025F0102170200' and a 'Port' field set to '1'. There is an unchecked checkbox for 'Use Sample Uplink Info'. The 'Payload Output' section shows a JSON object:

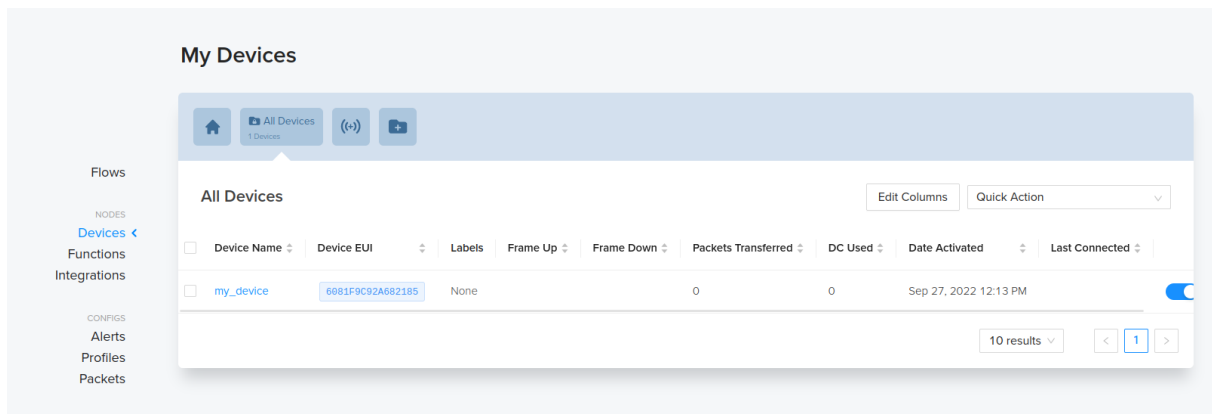
```

{
  "CC1": 291,
  "CC1_min": 285,
  "CC1_max": 298,
  "TEMP": 2276,
  "TEMP_min": 2276,
  "TEMP_max": 2276,
  "HUMI": 4259,
  "HUMI_min": 4245,

```

3.2 Add devices

- To add a device - select 'Devices' under 'Nodes' and choose 'Add New Device'.



- Give the device a name, Dev EUI, App Key and fill the App EUI with zeros.
- Press Save Device.
- In order for the device to connect, the sensor must be within range of Heliums Hotspot coverage.
- Explore hotspots at <https://explorer.helium.com/>.

