

# **AgroSense\_2 Channel CT Monitor**

## **LoRaWAN® Manual**

### **V1.0**

Author: Yuki

Time: 2025.05.20

## 目录

|                                      |    |
|--------------------------------------|----|
| 1 Product Description .....          | 1  |
| 1.1 Introduction .....               | 1  |
| 1.2 Feature .....                    | 1  |
| 1.3 Parameter .....                  | 2  |
| 2 Technical route .....              | 4  |
| 2.1 System Framework .....           | 4  |
| 2.2 Regional frequency band .....    | 5  |
| 3 Usage .....                        | 6  |
| 3.1 Interface Specification .....    | 6  |
| 3.2 Usage with TTN &ThingSpeak ..... | 7  |
| 3.3 Usage with Datacake .....        | 17 |

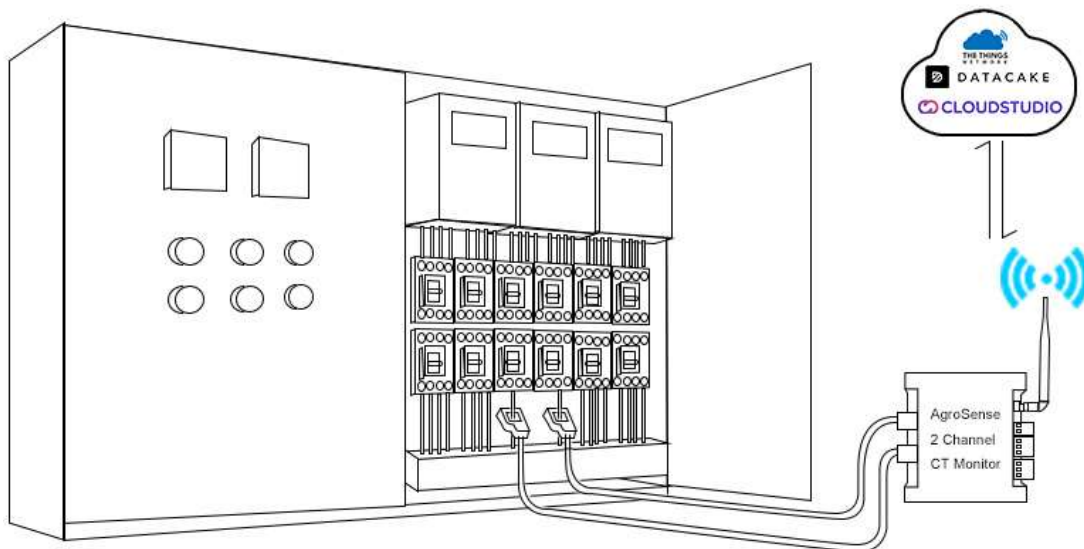
# 1 Product Description

## 1.1 Introduction

AgroSense 2-Channel LoRaWAN® CT Monitor integrates two non-intrusive clamp-on current transformers, two 0–5V analog inputs, and a high-precision temperature and humidity sensor into a single compact IoT device. Simply snap the CTs onto conductors—no power interruption or wiring modifications required—and begin monitoring energy consumption (default range: 100A /1V; customizable to other ranges with compatible interfaces) alongside environmental conditions. Reporting data once per minute over LoRaWAN® (up to 5 km line-of-sight), it delivers near-real-time insights for smart agriculture, industrial automation, and intelligent buildings, driving targeted efficiency improvements, cost savings, and optimized energy management.

The device complies with the LoRaWAN® V1.0.3 OTAA Class C standard and offers high sensitivity (-137 dBm) and strong anti-interference capability. It supports global frequency bands (EU868/US915) and works seamlessly with platforms such as TTN, Datacake, and CloudStudio to enable data upload and remote control.

Users can send downlink commands to adjust data reporting intervals. Local storage for 3,300+ records ensures critical data is retained during connectivity gaps.



## 1.2 Feature

- Includes 2-channel independently **CT Monitor**.
- Includes 2-channel 0-5V analog **signal input**.
- Integrates a high-accuracy **temperature and humidity sensor**.

- LoRaWAN version: LoRaWAN Specification 1.0.3. OTAA **Class C**.
- Monitor data and upload **real-time** data regularly.
- Modify the product parameters through **AT commands**.
- Support **downlink** to modify the time interval.
- Integrated data logging capability with a storage capacity of up to 3300 records.
- Compatible with Worldwide **LoRaWAN® Networks: Support** the universal frequency bands EU868/US915.
- **Long Range:** Up to 2 kilometers in the city, up to 10 kilometers in the wilderness, receive sensitivity -137dBm , transmit power up to 22dBm.
- **Data encryption:** Provide end-to-end secure communication, including device authentication and network data encryption, to ensure the security of data transmission and prevent data theft and malicious attacks.
- **High stability and reliability:** good stability in noisy environments, able to penetrate buildings and obstacles, so it can maintain good communication quality in urban and suburban environments.

## 1.3 Parameter

### 1. General Parameters

|                                  |             |
|----------------------------------|-------------|
| Product Model                    | AGLW2CT01   |
| Temperature Measurement Range    | -40°C ~80°C |
| Temperature Measurement Accuracy | ±0.5°C      |
| Temperature Resolution           | 0.1°C       |
| Humidity Measurement Range       | 0%-100% RH  |
| Humidity Measurement Accuracy    | ±2%         |
| Humidity Resolution              | 0.024% RH   |
| ADC Measurement Range            | 12-bit ADC  |
| ADC Measurement Accuracy         | 1/4096      |
| ADC channel count                | 2           |
| CT Supply                        | 100A/1V     |
| CT channel count                 | 2           |

### 2. Wireless Parameters

|                               |                                   |
|-------------------------------|-----------------------------------|
| Communication Protocol        | Standard LoRaWAN® protocol V1.0.3 |
| Network Access/Operating Mode | OTAA <b>Class C</b>               |
| MAX Transmit Power            | 22dBm                             |
| Receiver Sensitivity          | -137dBm/125kHz SF=12              |

|                |             |
|----------------|-------------|
| Frequency Band | EU868/US915 |
|----------------|-------------|

### 3.Physical Parameters

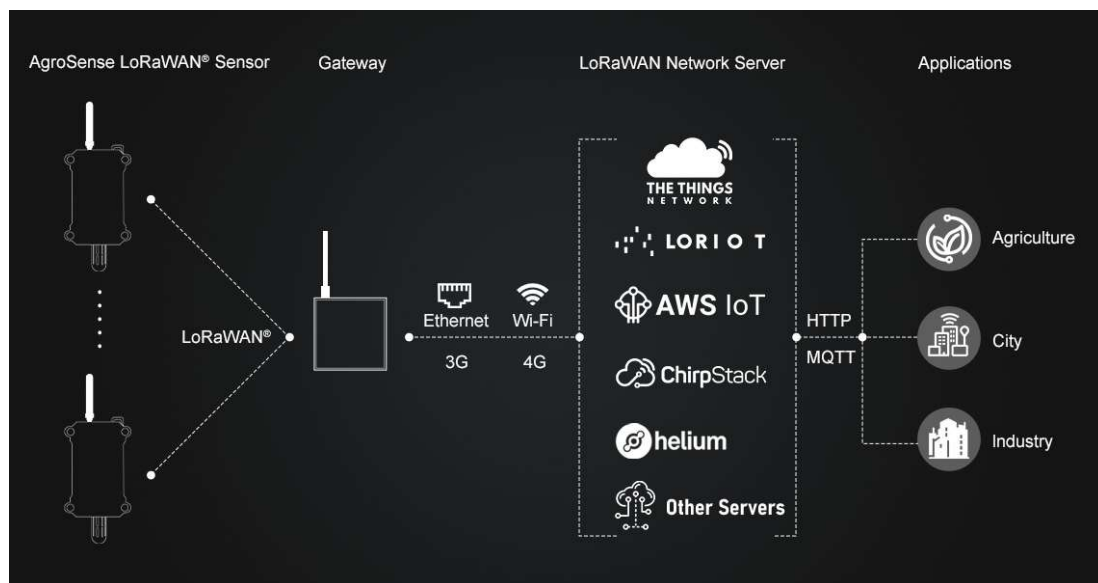
|                       |                 |
|-----------------------|-----------------|
| Power Supply          | DC 12-24V       |
| Operating Temperature | -40°C ~85°C     |
| Dimensions            | 95 × 90 × 40 mm |
| Mounting              | Wall Mounting   |

## 2 Technical route

### 2.1 System Framework

AgroSense\_2 Channel CT Monitor uses LoRaWAN technology, and its network architecture includes four parts: End Nodes, Concentrator/Gateway, Network Server and Application Server.

|                      |   |
|----------------------|---|
| End Nodes            | It is responsible for collecting sensing data and then transmitting it to Gateway via the LoRaMAC protocol. |
| Concentrator/Gateway | It is mainly responsible for transmitting node data to the server.  |
| Network Server       | Organize the data into JSON packets and decode them.  |
| Application Server   | Display the data.   |



#### Uplink:

##### 1.Data Collection & Transmission

Sensor data and transmits it to the Gateway via LoRaWAN® protocol.

##### 2.Gateway Forwarding

The Gateway packages the raw data and forwards to the Network Server.

##### 3.Data Decoding & Routing

The Network Server decodes the payload and forwards it to the designated Application Server.

##### 4.User Monitoring

The Application Server processes the data and updates the user interface (APP), allowing real-time monitoring of data.

### Downlink:

#### 1.Command Generation

A downlink commands generated in the Network Server or Application Server through a predefined API/interface. (Example: Set sampling interval to 10 minute.)

#### 2.Gateway Transmission

The command is encapsulated into a downlink packet and sent to the Gateway via the network.

#### 3.End Node Execution

The Gateway transmits the downlink command to the target End Node using the wireless protocol. The End Node parses the command and performs the corresponding action (e.g., modify configuration).

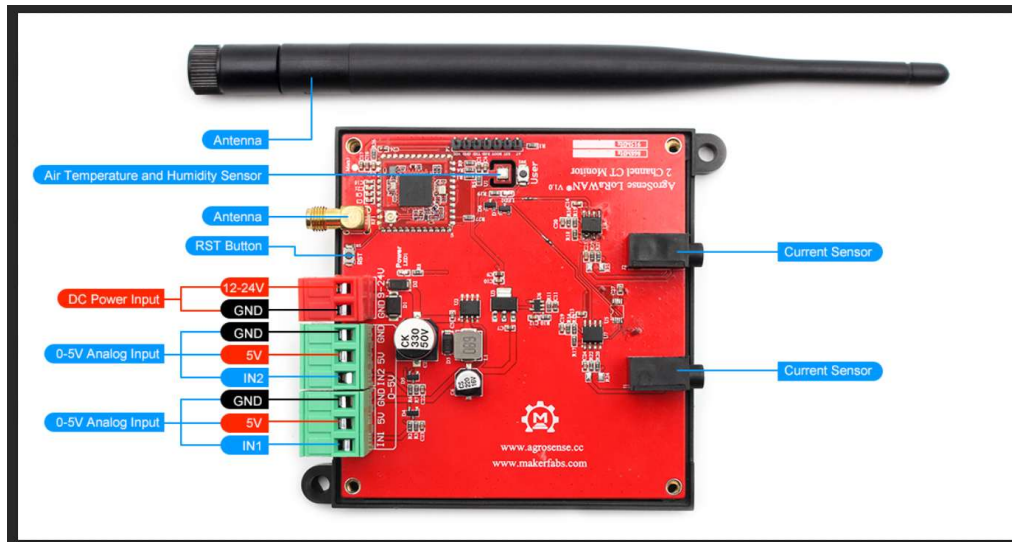
## 2.2 Regional frequency band

At the present moment, our product solely accommodates compatibility with the US915 and EU868.

| area        | frequency band | center frequency |
|-------------|----------------|------------------|
| China       | 470-510MHz     | CN486MHz         |
| America     | 902-928MHz     | US915MHz         |
| Europe      | 863-870MHz     | EU868MHz         |
| Korea       | 920-923MHz     | KR922MHz         |
| Australia   | 915-928MHz     | AU923MHz         |
| New Zealand | 921-928MHz     | NZ922MHz         |
| Asia        | 920-923MHz     | AS923MHz         |

## 3 Usage

### 3.1 Interface Specification



#### 3.1.1 DC Power Input

AgroSense\_2 Channel CT Monitor operates from an external DC power source.

|                      |   |
|----------------------|---|
| Power Specifications | DC 12-24V $\pm 10\%$                    |
| Terminal Definitions | 12-24V: Positive terminal (power input) |
|                      | GND: Negative terminal (ground)         |

#### 3.1.2 Dual-Channel 0-5V Analog Input

AgroSense\_2 Channel CT Monitor provides 2-channel analog sensor interfaces (0-5V), enabling flexible deployment of analog signal-based sensors.(e.g.pressure, light intensity)

|                      |  |
|----------------------|--|
| Sensor Power Output  | 5V $\pm 0.2V$<br>Max load current: 500mA |
| Terminal Definitions | GND: Negative terminal (ground)          |
|                      | 5V: Positive terminal (power output)     |
|                      | IN1/IN2: (Signal Input)                  |



### 3.1.3 Air Temperature & Humidity Detection

**AgroSense\_2 Channel CT Monitor** integrates a high-precision temperature and humidity sensor AHT20, capable of monitoring and feedback of environmental parameters to provide data support for system regulation.

- Temperature Sensor: -40°C to +85°C
- Relative Humidity Sensor: 0% to 100% RH

### 3.1.4 TWO-Channel CT Monitor

**AgroSense\_2 Channel CT Monitor** supports two independent, plug-and-play current detection channels, enabling users to begin monitoring energy consumption instantly without any wiring modifications.

## 3.2 Usage with TTN & ThingSpeak

In the phase, We use The Things Network(TTN) as data server, and Thingspeak as console to display data& control the relays.

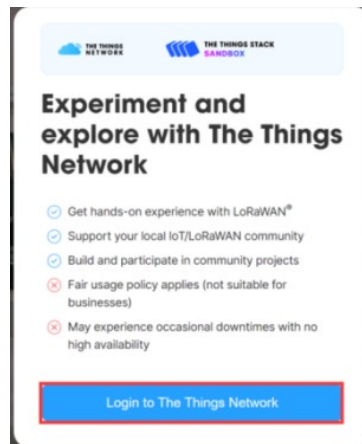
we need to configuration the country/area frequency, inputting DEV EUI/ APP EUI/ APP Key, decodes, and connect to ThingSpeak.

|         |   |
|---------|---|
| DEV EUI | Unique identification of device, authorized by IEEE                   |
| APP EUI | Unique identification of application                                  |
| APP Key | One of the join network parameters on OTAA mode, calculated by DE EUI |

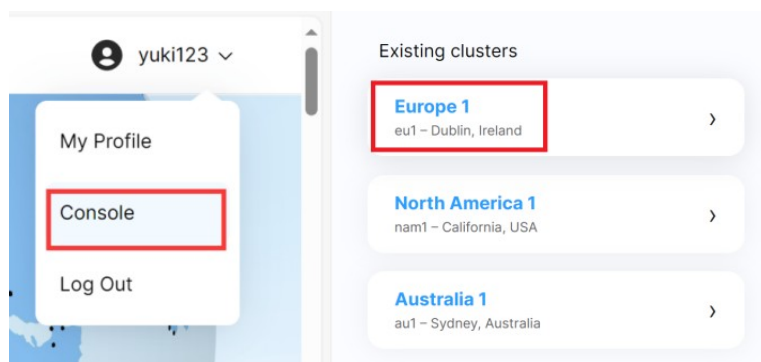
- End Nodes and Gateway: AgroSense\_4 Channel Relay.(The AgroSense series is applicable)
- Network Server: The Things Network. ( Datacake, Lorient, AWS IoT, ChirpStack, ect)
- Application Server: ThingSpeak.(Datacake, Blockbax, akenza, ect)

### 3.2.1 Network Server configuration

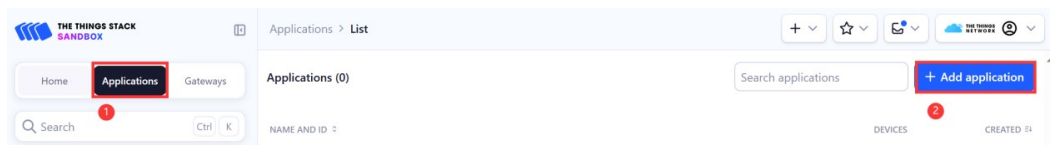
- Open The Things Network in your browser and login it. (Or register an account)



- Click “Console” and select clusters. (we take the European region for example.)



- Click “Go to applications” --> “+ Create application”.



- Write the Application ID and click “Create application”.

Application ID \*

agrosense-sensor

Application name

My new application

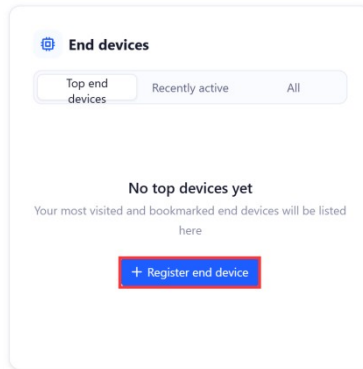
Description

Description for my new application

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

Create application

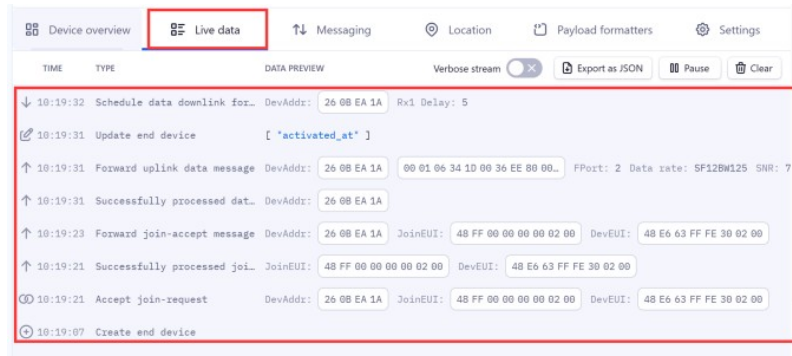
- Click “+ Register and device”.



- Following the steps, **select class C** and input the DEV EUI/ APP EUI/ APP Key (notice: JoinEUI=APP EUI) and subsequently click on "Register end device" to complete the registration process.

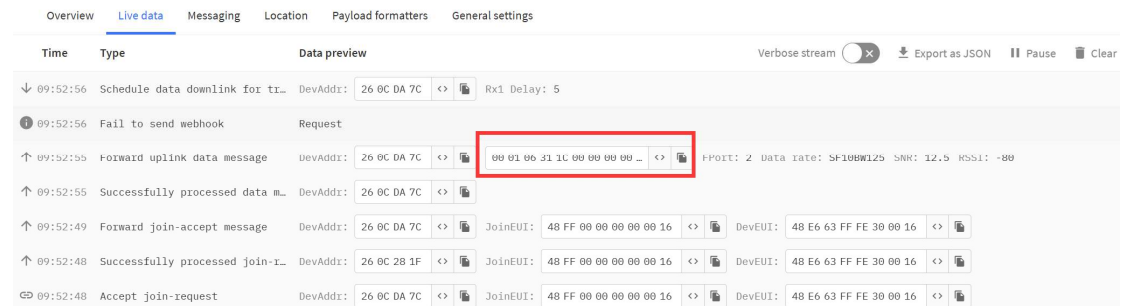


- Connects to 12-24VDC power supply and press RES button, you can see the device is connected successfully in the TTN.



## 3.2.2 Decoder

- Now, we need to decode the data.



| Data length | Data description                        | Value range | Explanation   |
|-------------|---|-------------|---|
| byte 0      | Data packet sequence number high 8 bits | 0-0xFFFF    | Counting starts from 0 and increments, resetting back to 0 after reaching 65535   |
| byte 1      | Data packet sequence number low 8 bits  |             |   |
| byte 2      | ADC_1 bits 8 to 15                      |             | The value is amplified by a factor of 10. To get the actual value, divide it by 10. For example, if the value is 0x21 (33), the actual voltage is 3.3 V |
| byte 3      | ADC_1 bits 0 to 7                       |             |   |
| byte 4      | ADC_2 bits 8 to 15                      |             | The value is amplified by a factor of 10. To get the actual value, divide it by 10. For example, if the value is 0x21 (33), the actual voltage is 3.3 V |
| byte 5      | ADC_2 bits 0 to 7                       |             |   |
| byte 6      | Current_1 bits 8 to 15                  |             | This value is obtained after magnifying the data by 1000 times.   |
| byte 7      | Current_1 bits 0 to 7                   |             |   |

### AgroSense\_4 Channel Relay LoRaWAN®

|          |   |  |   |
|----------|---|--|---|
| byte 8   | Current_2 bits 8 to 15                              |  | This value is obtained after magnifying the data by 1000 times.   |
| byte 9   | Current_2 bits 0 to 7                               |  |   |
| byte 10  | Humidity sensor bits 8 to 15                        |  | The value is amplified by a factor of 10. For example, if the value from the 8th to the 15th bit is 0x02, and the lower 8 bits value is 0x85, then the relative humidity value obtained is 0x00000285= 645. After converting and dividing by 10, the actual relative humidity is 6.45%RH. |
| byte 11  | Humidity sensor bits 0 to 7                         |  |   |
| byte 12  | Temperature sensor bits 8 to 15                     |  | The value is amplified by a factor of 10. For example, if the value from the 8th to the 15th bit is 0x02, and the lower 8 bits value is 0x85, then the Temperature value obtained is 0x00000285= 645. After converting and dividing by 10, the actual Temperature is 6.45℃.               |
| byte 13  | Temperature sensor bits 0 to 7                      |  |   |
| byte 14  | data transmission interval bits 24 to 31            |  | The time interval for data transmission has been increased by a factor of 1000.<br>The unit is seconds.   |
| byte 15  | data transmission interval bits 16 to 23            |  |   |
| byte 16  | data transmission interval bits 8 to 15             |  |   |
| byte 17  | data transmission interval bits 0 to 7              |  |   |
| Downlink |   |  |   |
| Fport 1  | Change the data sending interval                    |  |   |
| Fport 5  | Upload the quantity of the latest local logged data |  |   |

Example:0x00 0x8C 0x00 0x08 0x00 0x20 0x01 0x40 0x00 0x40 0x01 0xCC 0x01 0x15 0x00 0x00 0xEA 0x60

Data parsing:

ADC1 value is 1.0v

ADC2 value is 3.2v

Current\_1 is 0.5v

Current\_2 is 0.1v

Humidity value is 46.1%

Temperature value is 27.8°C

Data transmission interval value is 60s

- Know how to decode it after, we need to write it in code. (You can check it out on [Github](#))

```
// This file contains the uplink and downlink for ttn
```

```
// Uplink
```

```
function decodeUplink(input) {
```

```
    var bytes = input.bytes;
```

```
    var num = bytes[0] * 256 + bytes[1];
```

```
    var ADC_1 = (bytes[2] * 256 + bytes[3]) / 10.0;
```

```
    var ADC_2 = (bytes[4] * 256 + bytes[5]) / 10.0;
```

```

var CT1_ADC = (bytes[6] * 256 + bytes[7]) / 10.0; //This voltage needs to be combined with the sensor's ratio;
var CT2_ADC = (bytes[8] * 256 + bytes[9]) / 10.0; //This voltage needs to be combined with the sensor's ratio;

var CT1 = CT1_ADC*60 //if the sensor is 60A/1V, multiply by 60; if it's 100A/1V, multiply by 100; if it's 300A/1V,
multiply by 300.
var CT2 = CT2_ADC*60 //if the sensor is 60A/1V, multiply by 60; if it's 100A/1V, multiply by 100; if it's 300A/1V,
multiply by 300.

var humidity = (bytes[10] * 256 + bytes[11]) / 10.0;
var temperature = bytes[12] * 256 + bytes[13];
if (temperature >= 0x8000) {
    temperature -= 0x10000;
}
temperature = temperature / 10.0;

var interval = (bytes[14] * 16777216 + bytes[15] * 65536 + bytes[16] * 256 + bytes[17]) / 1000;
return {
    data: {
        ADC_1: ADC_1, //0-5V ADC
        ADC_2: ADC_2, //0-5V ADC
        CT1: CT1, //A
        CT2: CT2, //A
        temperature: temperature,
        humidity: humidity,
        interval: interval
    },
    warnings: [],
    errors: []
};
}

```

- Select “Payload formatters” and follow the steps.

Device overview Live data Messaging Location Payload formatters 1 Settings

Uplink Downlink

Setup

Formatter type\*  
Custom Javascript formatter 2

Formatter code\*  

```

1 // This file contains the uplink and downlink for ttn
2 // Uplink
3 function decodeUplink(input) {
4     var bytes = input.bytes;
5
6     var num = bytes[0] * 256 + bytes[1];
7     var ADC_1 = (bytes[2] * 256 + bytes[3]) / 10.0;
8     var ADC_2 = (bytes[4] * 256 + bytes[5]) / 10.0;
9     var CT1_ADC = (bytes[6] * 256 + bytes[7]) / 10.0; //This voltage needs to be combined with the sensor's ratio;
10    var CT2_ADC = (bytes[8] * 256 + bytes[9]) / 10.0; //This voltage needs to be combined with the sensor's ratio;
11
12    var CT1 = CT1_ADC*60 //if the sensor is 60A/1V, multiply by 60; if it's 100A/1V, multiply by 100; if it's 300A/1V, multiply by 300.
13    var CT2 = CT2_ADC*60 //if the sensor is 60A/1V, multiply by 60; if it's 100A/1V, multiply by 100; if it's 300A/1V, multiply by 300.
14
15    var humidity = (bytes[10] * 256 + bytes[11]) / 10.0;
16    var temperature = bytes[12] * 256 + bytes[13];
17    if (temperature >= 0x8000) {
18        temperature -= 0x10000;
19    }
20    temperature = temperature / 10.0;
21
22    var interval = (bytes[14] * 16777216 + bytes[15] * 65536 + bytes[16] * 256 + bytes[17]) / 1000;
23    return {
24        data: {
25            ADC_1: ADC_1, //0-5V ADC

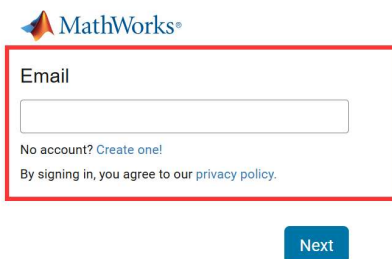
```

Save changes

### 3.2.3 Application Server configuration

In the Application Server configuration, we need to create ThingSpeak channel and get Channel ID and API Key, this is the key to our connection to TTN.

- Login to the ThingSpeak. (Or register an account)



MathWorks®

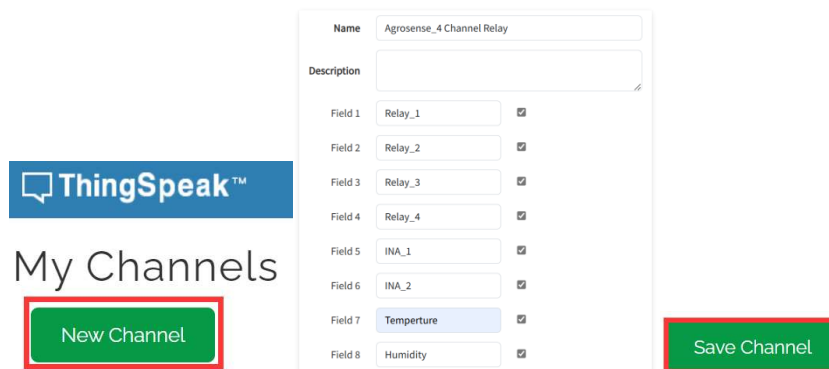
Email

No account? [Create one!](#)

By signing in, you agree to our [privacy policy](#).

Next

- Click “New Channel”, fill in the Channel name and field names and click “Save Channel”.



ThingSpeak™

My Channels

New Channel

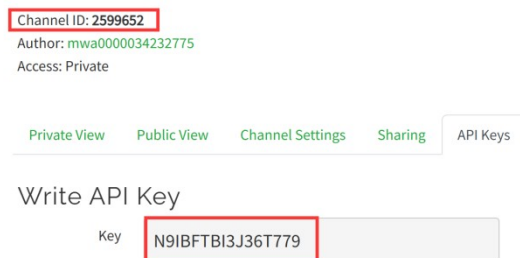
Name: Agrosense\_4 Channel Relay

Description:

|         |             |                                     |
|---------|-------------|-------------------------------------|
| Field 1 | Relay_1     | <input checked="" type="checkbox"/> |
| Field 2 | Relay_2     | <input checked="" type="checkbox"/> |
| Field 3 | Relay_3     | <input checked="" type="checkbox"/> |
| Field 4 | Relay_4     | <input checked="" type="checkbox"/> |
| Field 5 | INA_1       | <input checked="" type="checkbox"/> |
| Field 6 | INA_2       | <input checked="" type="checkbox"/> |
| Field 7 | Temperature | <input checked="" type="checkbox"/> |
| Field 8 | Humidity    | <input checked="" type="checkbox"/> |

Save Channel

- After successful creation, copy the Channel ID and API Key.



Channel ID: 2599652

Author: mwa000034232775

Access: Private

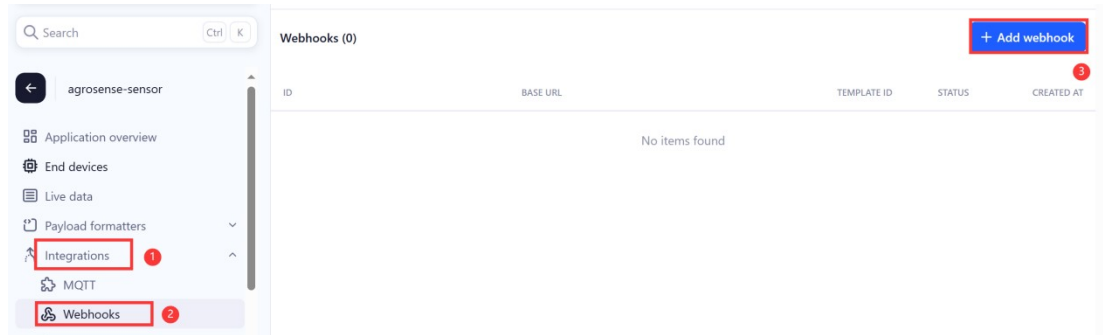
Private View Public View Channel Settings Sharing API Keys

Write API Key

Key: N9IBFTBI3J36T779

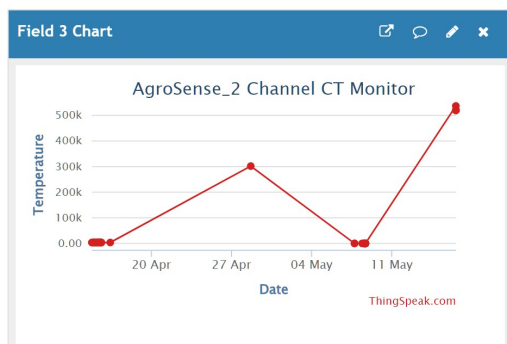
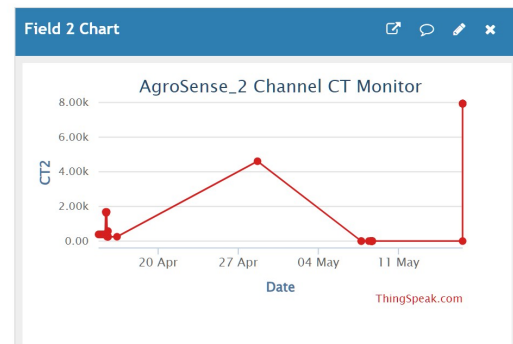
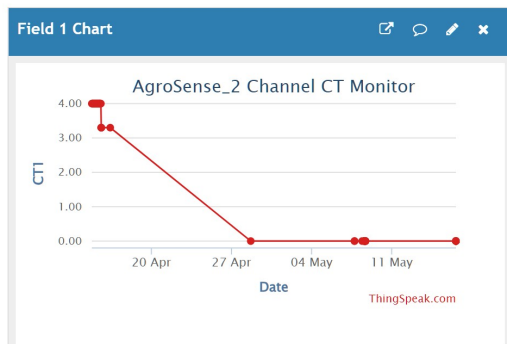
### 3.2.4 Connect the Network Server and Application Server

- In the TTN, click “integrations” --> “Webhooks” --> “+ Add webhook”.



- Select “ThingSpeak”, Fill in the Webhook ID and paste the Channel ID and API Key, click “Create ThingSpeak Webhook”.

- Press RST button, wait about a minute, you will successfully see the data in ThingSpeak.(You will receive the data every hour.)



### 3.2.5 Downlink

The downlink has two functions:



- Modification time interval (Fport1)

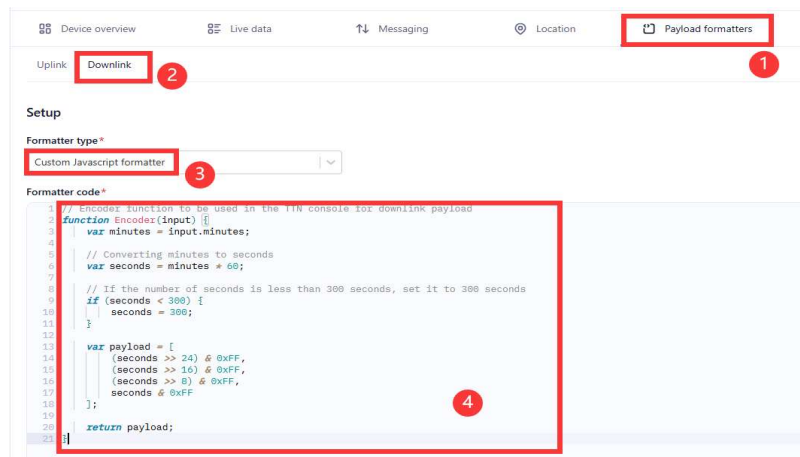
Modify the time interval for uploading data, the default is one hour.

- Upload the quantity of the latest local logged data (Fport5)

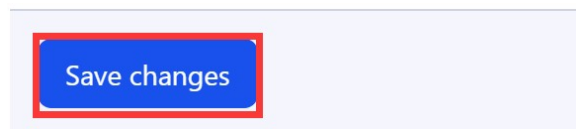
### Modify the time interval :

1、 If you need to change time Interval (Default 1 minutes), you can click “Payload formatters-->Downlink” and follow the steps.

Formatter code you can find in [Github](#).



2、 Click “Save changes”.



3、 Click “Messaging-->Schedule downlink”.

**Note:** you must use this format:

```
{
  "minutes": 5
}
```

Device overview

Live data

Messaging

Schedule downlink

Simulate uplink

Schedule downlink

Insert Mode

☒ Replace downlink queue

☐ Push to downlink queue (append)

FPort\*

1

Payload type

☐ Bytes

☒ JSON

Payload

```
{
  "minutes": 5
}
```

The decoded payload of the downlink message

☐ Confirmed downlink

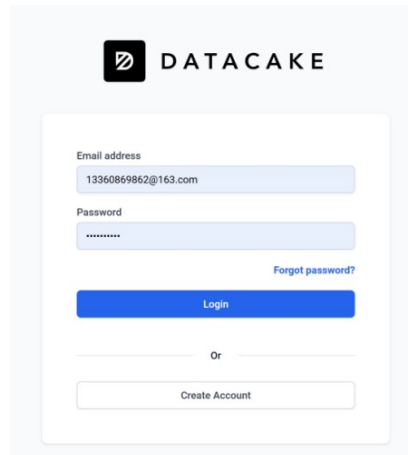
Schedule downlink

4、The modified interval will be updated after the next data upload.

### 3.3 Usage with Datacake

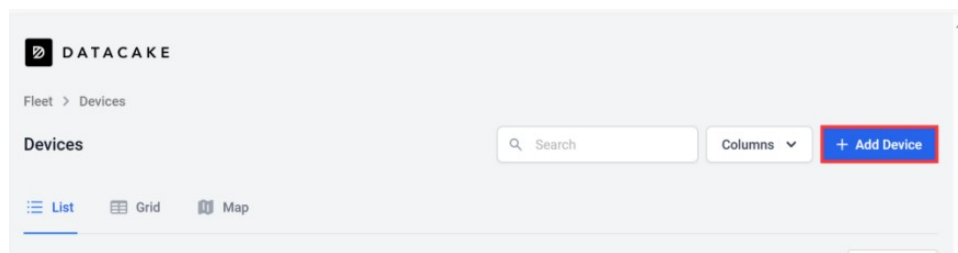
In this phase, we use DataCake(<https://datacake.co/>) as the data server & console.

#### 1、Login datacake or Create Account

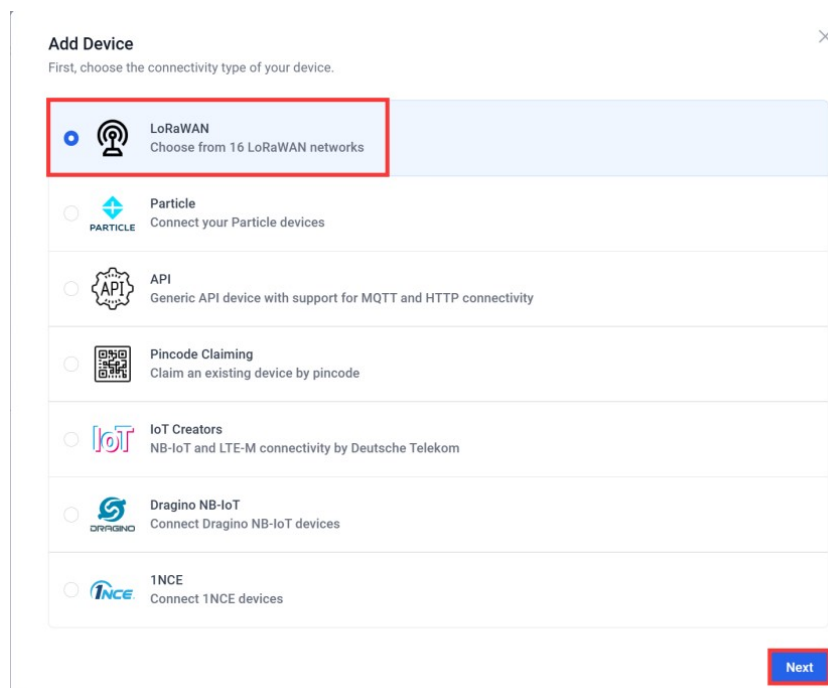


The image shows the DataCake login and registration interface. At the top is the DataCake logo. Below it is a form with two input fields: 'Email address' (containing '13360869862@163.com') and 'Password' (masked with dots). A 'Forgot password?' link is next to the password field. Below the fields is a blue 'Login' button. Underneath is an 'Or' separator, followed by a 'Create Account' button.

#### 2、Click “Add Device”



#### 3、Select LoRaWAN and click “Next”



The image shows the 'Add Device' dialog box. The title is 'Add Device' with a close button. Below the title is the instruction 'First, choose the connectivity type of your device.' There is a list of connectivity options, each with a radio button and an icon. The first option, 'LoRaWAN', is selected and highlighted with a red box. It includes the text 'Choose from 16 LoRaWAN networks'. Other options include 'Particle', 'API', 'Pincode Claiming', 'IoT Creators', 'Dragino NB-IoT', and '1NCE'. A red 'Next' button is at the bottom right.

4、Select a Product based on your needs, take "Create new empty product" as an example.

Add LoRaWAN Device

You can add individually billed devices.

STEP 1

STEP 2

STEP 3

STEP 4

Product

Network Server

Devices

Plan

### Datacake Product

You can add devices to an existing product on Datacake, create a new empty product or start with one of the templates. Products allow you to share the same configuration (fields, dashboard and more) between devices.

**New Product from template**  
Create new product from a template

**Existing Product**  
Add devices to an existing product

**New Product**  
Create new empty product

### New Product

If your device is not available as a template, you can start with an empty device. You will have to create the device definition (fields, dashboard) and provide the payload decoder in the device's configuration.

Product Name

Agrosense sensor

Back
Next

5、Select "Datacake LNS"

Add LoRaWAN Device

STEP 1

STEP 2

STEP 3

STEP 4

Product

Network Server

Devices

Plan

### Network Server

Please choose the LoRaWAN Network Server that your devices are connected to.

☒

Datacake LNS
AUTOMATIC SETUP

Start and scale easily with a managed LNS

Uplinks
Downloads

☐

The Things Stack V3
TTN V3 / Things Industries

Uplinks
Downloads

☐

Helium
Use your own console

Uplinks
Downloads

☐

LORIoT

Uplinks
Downloads

☐

ChirpStack

Uplinks
Downloads

☐

Activity

Uplinks
Downloads

☐

KPN

Uplinks
Downloads

Showing 1 to 6 of 15 results

Previous
Next

Back
Next

18

6、Enter DEVEUI、APPEUI、APPKEY、FREQUENCY(take 915 for example) and DEVICE CLASS.

7、Choose the type according to your needs, and click “Add 1 device”.

8、Click to go to the device you just added.

**DATA CAKE**

Fleet > Devices

Devices

Search Columns + Add Device

List Grid Map

| DEVICE  | PRIMARY | SECONDARY | DEVICE SIGNAL | DEVICE BATTERY | Actions |
|---|---------|-----------|---------------|----------------|---------|
| AgroSense_Air Temperature and Humidity Sensor | 40.2    | 25        | -48           | 2.5            | 👁️ ⋮ ⌵  |
| AgroSense-carbon dioxide (CO2) sensor         | 0       | N/A       | -86           | 3.3            | 👁️ ⋮ ⌵  |
| agrosense sensor                              | N/A     | N/A       | N/A           | N/A            | 👁️ ⋮ ⌵  |

Showing 1 to 3 of 3 results

50 per page Previous Next



```

    temperature -= 0x10000;

}

temperature = temperature / 10.0;

var interval = (input.bytes[14]* 16777216 + input.bytes[15]* 65536 + input.bytes[16] * 256 + input.bytes[17])
/ 1000

var decoded =
{
    ADC_1:ADC_1,
    ADC_2:ADC_2,
    CT1:CT1,
    CT2:CT2,
    temperature:temperature,
    humidity:humidity,
    interval:interval,
};

// Test for LoRa properties in normalizedPayload
try {
    if (normalizedPayload.gateways && normalizedPayload.gateways.length > 0) {
        decoded.lora_rssi = normalizedPayload.gateways[0].rssi || 0;
        decoded.lora_snr = normalizedPayload.gateways[0].snr || 0;
    } else {
        decoded.lora_rssi = 0;
        decoded.lora_snr = 0;
    }
}

decoded.lora_datarate = normalizedPayload.spreading_factor
                        || normalizedPayload.data_rate
                        || (normalizedPayload.networks && normalizedPayload.networks.lora &&
normalizedPayload.networks.lora.dr)
                        || "unknown";

```

```

} catch (error) {

    console.log('LoRa property parsing error:', error);

    decoded.lora_rssi = 0;

    decoded.lora_snr = 0;

    decoded.lora_datarate = "unknown";

}

return [

    { field: "ADC_1", value: decoded.ADC_1 },

    { field: "ADC_2", value: decoded.ADC_2 },

    { field: "CT1", value: decoded.CT1 },

    { field: "CT2", value: decoded.CT2 },

    { field: "humidity", value: decoded.humidity },

    { field: "temperature", value: decoded.temperature },

    { field: "interval", value: decoded.interval },

    { field: "lora_rssi", value: decoded.lora_rssi },

    { field: "lora_snr", value: decoded.lora_snr },

    { field: "lora_datarate", value: decoded.lora_datarate },

];

}

```

10、 Follow the steps to add a field. (Every fields is the same way)

**Fields**  
Fields describe the data the device will store.

+ Add Field



**Fields**  
Fields describe the data the device will store.

[+ Add Field](#)

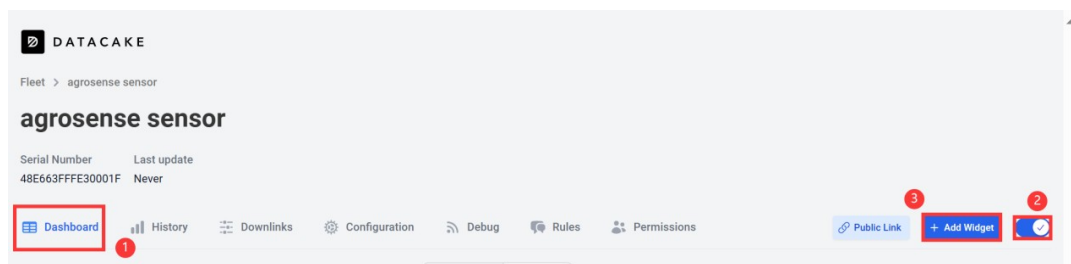
● Live data

| NAME          | IDENTIFIER    | TYPE    | ROLE | CURRENT VALUE | LAST UPDATE    |
|---------------|---------------|---------|------|---------------|----------------|
| ADC_1         | ADC_1         | Float   | N/A  | 0             | 2 minutes ago  |
| ADC_2         | ADC_2         | Float   | N/A  | 0             | 2 minutes ago  |
| CT1           | CT1           | Float   | N/A  | 0             | 2 minutes ago  |
| CT2           | CT2           | Float   | N/A  | 0             | 2 minutes ago  |
| humidity      | HUMIDITY      | Float   | N/A  | 0             | 1 minute ago   |
| temperature   | TEMPERATURE   | Float   | N/A  | 0             | 1 minute ago   |
| interval      | INTERVAL      | Integer | N/A  | 0             | 1 minute ago   |
| lora_rssi     | LORA_RSSI     | Integer | N/A  | 0             | 22 seconds ago |
| lora_snr      | LORA_SNR      | Integer | N/A  | 0             | 4 seconds ago  |
| lora_datarate | LORA_DATARATE | String  | N/A  |               | in 6 seconds   |

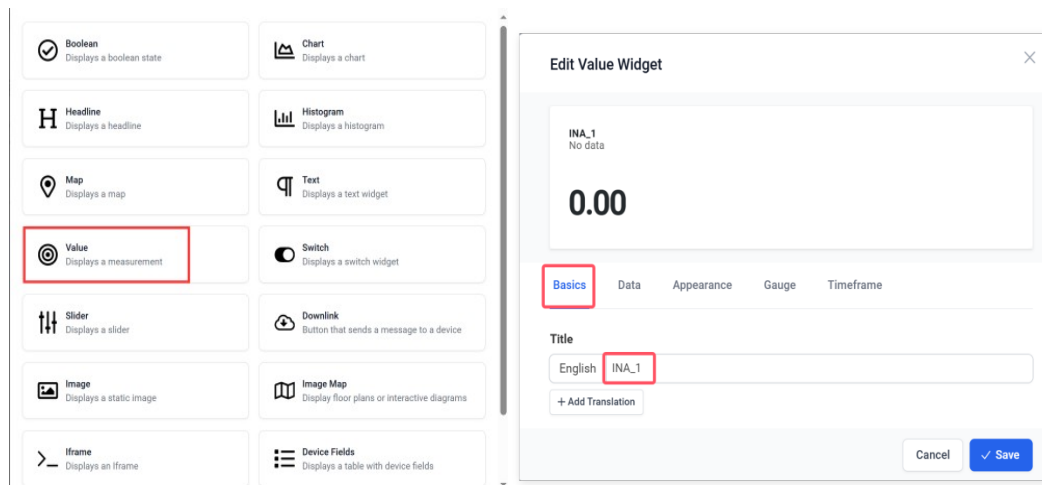
11、 Press RST button, wait until the sensor connects to the gateway successfully, you will see the data the sensor is currently reading.

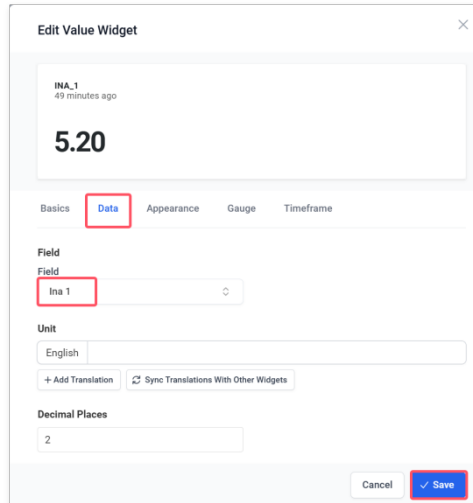
12、 To get a better look at the data, we can add widget.

Click “Dashboard-->switch-->+ Add Widget”.

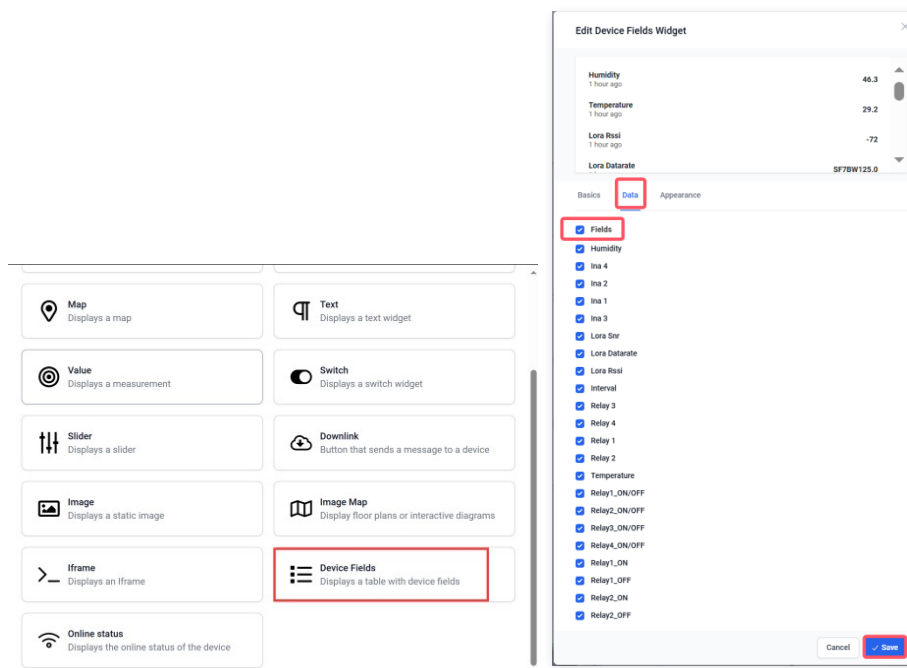


13、 Select “Value” and set Title, Field.ADC\_1, ADC\_2, CT1,CT2,Temperature, Humidityas the same way.

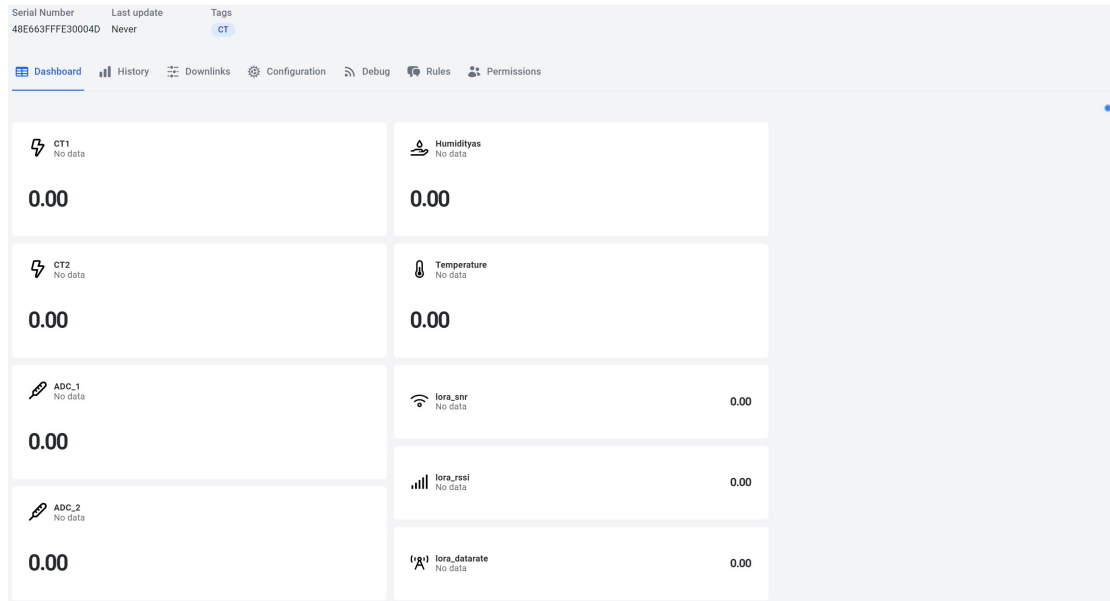




14、Select Device Fields, check “Fields” and click “Save”.



15、Click the switch to save, and you can see the data visually.



### 3.3.1 Downlink

The downlink has the following functions:

- Modification time interval (Fport1)

Modify the time interval for uploading data, the default is one hour.

- Upload the quantity of the latest local logged data (Fport5)

Users can view previous data based on this feature.

#### Modify the time interval :

1 、 If you need to change time Interval (Default 1 minutes), you can click “Configuration-->Fields-->+Add Field”

Add Field

Fields define the schema of the data the device stores.

Type

Integer

Name

Sending Time Interval

Identifier

SENDING\_TIME\_INTERVAL

The field identifier is a unique string that can consist of uppercase letters, numbers and underscores. Once a field has been created, the identifier can not be changed.

Unit

Optional

Role

None

You can define the role of a field, which are unique per product and can be used to add context to global visualisations and reports.

Formula

Optional

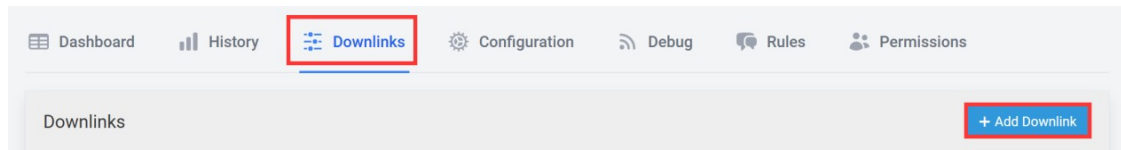
Formulas can be used to perform calculations on values from other fields. Fields that have a formula can not be written to from a decoder or via the API.

☐ Use Formula

Cancel

Add Field

2、Click "Downlink-->Add Downlink".

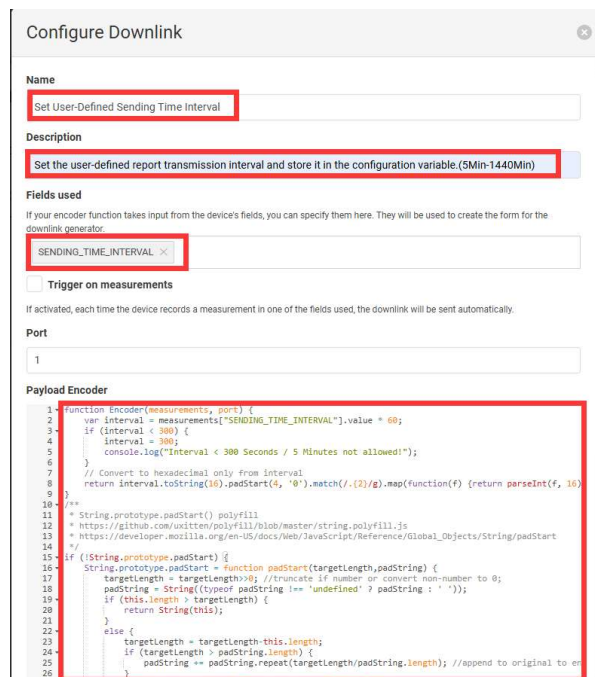


Enter name、description、 fields used and payload encoder respectively.

Name: Set User-Defined Sending Time Interval

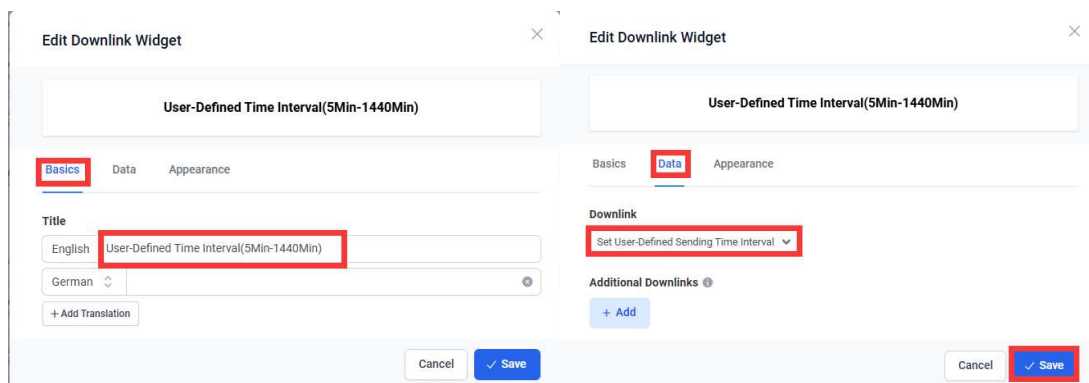
Description: Set the user-defined report transmission interval and store it in the configuration variable.(10S -1440Min)

Payload Encoder: copy in [Github](#).



3、Click "Dashboard-->switch-->+ Add Widget".

Select "Downlink" and setting as follow image.



- 4、Click the switch to save, and you can click to change your time Interval.

User-Defined Time Interval(5Min-1440Min)

Sending Time Interval

1

Cancel

Save measurements and send downlink