



## AgroSense\_Leaf Moisture SN-3001

### LoRaWAN® Manual

### V1.0

Author: Yuki

Time: 2024.12.07

# Contents

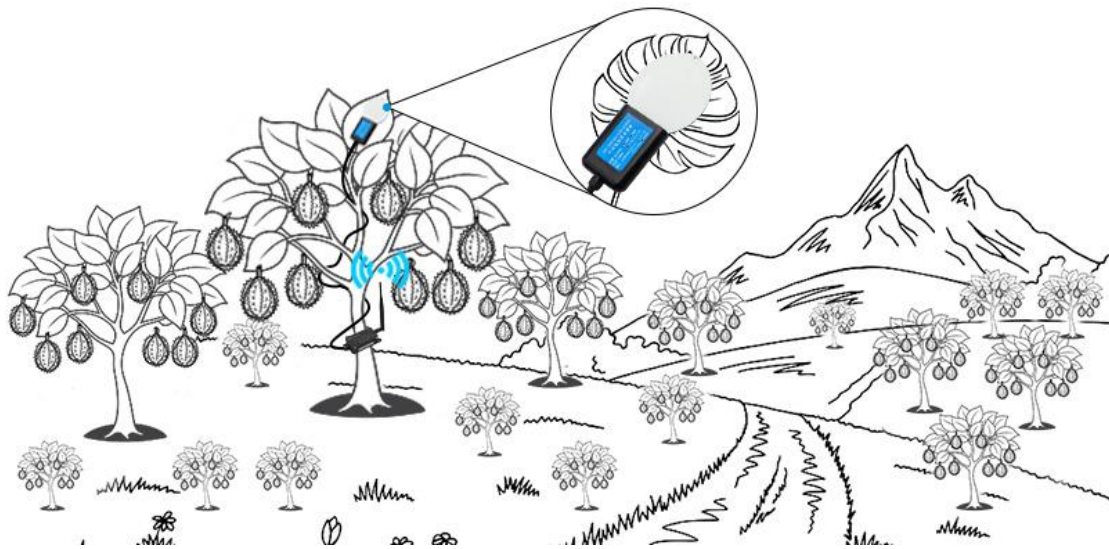
<b>1 Product Description .....</b>	<b>1</b>
1.1 Introduction .....	1
1.2 Feature .....	2
1.3 Parameter .....	3
<b>2 Technical route .....</b>	<b>4</b>
2.1 System Framework .....	4
2.2 Regional frequency band .....	4
<b>3 Usage .....</b>	<b>6</b>
3.1 TTN and ThingSpeak .....	6
3.1.1 Network Server configuration .....	6
3.1.2 Decoder .....	9
3.1.3 Application Server configuration .....	11
3.1.4 Connect the Network Server and Application Server .....	12
3.1.5 Change Time Interval (5-1440min) .....	13
3.2 Datacake .....	15
3.2.1 Change Time Interval (5-1440min) .....	22

# 1 Product Description

## 1.1 Introduction

This AgroSense LoRaWAN® Leaf Moisture Sensor uses the SN-3001 sensor, to measures the leaf moisture and temperature, so to send to the platform to analyze the leaf status such as : watering, moisturizing, dew, frozen, at the range of  $-40^{\circ}\text{C} \sim 60^{\circ}\text{C}$  and  $0\% \sim 100\% \text{ RH}$ , with accuracy  $\pm 0.5^{\circ}\text{C}$  and  $\pm 3\% \text{ RH}$  respectively, also with highly waterproof performance tested to IP68, making it widely applicable in agricultural environmental sensing scenarios to support the smart agricultural production.

The SN-3001 sensor adopts SHT4 high-precision chip, detects leaf moisture and temperature. It employs the FDR method, which enables it to sense the dielectric constant caused by liquid over the leaf surface and then cover the value with respect to leaf moisture. The probe has been designed in a leaf shape to best simulate the real leaf characteristics. Suitable for agricultural production scenarios where precise leaf temperature and humidity are required, such as durian, tomato and cucumber production, etc.



The sensor benefits from LoRaWAN, which ensures stability and reliability. It is capable of covering a long transmission range while maintaining low power consumption. Unlike wireline devices, it is battery-powered, reducing the workload and complexity of deployment, design and development for end-users that can work via powering it, and setting the configuration in the cloud server, for LoRaWAN® remote monitoring. It monitors the air temperature and humidity and report them every 1 hour.



## 1.2 Feature

- Includes a **high precision** sensor.
- Compatible with Worldwide **LoRaWAN® Networks**: Support the universal frequency bands EU868/ US915.
- LoRaWAN version: LoRaWAN Specification **1.0.3**.
- **Long Range**: Up to 2 kilometers in the city, up to 10 kilometers in the wilderness, receive sensitivity -137dBm , transmit power up to 21dBm.
- **Ultra-low power** consumption design, traditional AAA alkaline dry battery can be used for one year.
- **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.
- Suitable for **Harsh Environments**: Can work normally under the temperature of -40℃ ~ 85℃, IP68 waterproof, suitable for outdoor use in harsh conditions, high UV, dusty, heavy rain and other bad weather.
- Monitor data and upload **real-time** data regularly.
- Modify the product parameters through **AT commands**.
- Support **downlink** to modify the time interval, motion status on/off, motion status sensitivity.

## 1.3 Parameter

### 1. General Parameters

Product Model	AGLWLM01
Temperature Measurement Range	-40°C ~60°C
Temperature Measurement Accuracy	±0.5°C
Temperature Resolution	0.1°C
Humidity Measurement Range	0%-100% RH
Humidity Measurement Accuracy	±3%
Humidity Resolution	0.1% RH

### 2. Wireless Parameters

Communication Protocol	Standard LoRaWAN® protocol V1.0.3
Network Access/Operating Mode	OTAA Class A
MAX Transmit Power	21dBm
Receiver Sensitivity	-137dBm/125kHz SF=12
Frequency Band	EU868/US915

### 3. Physical Parameters

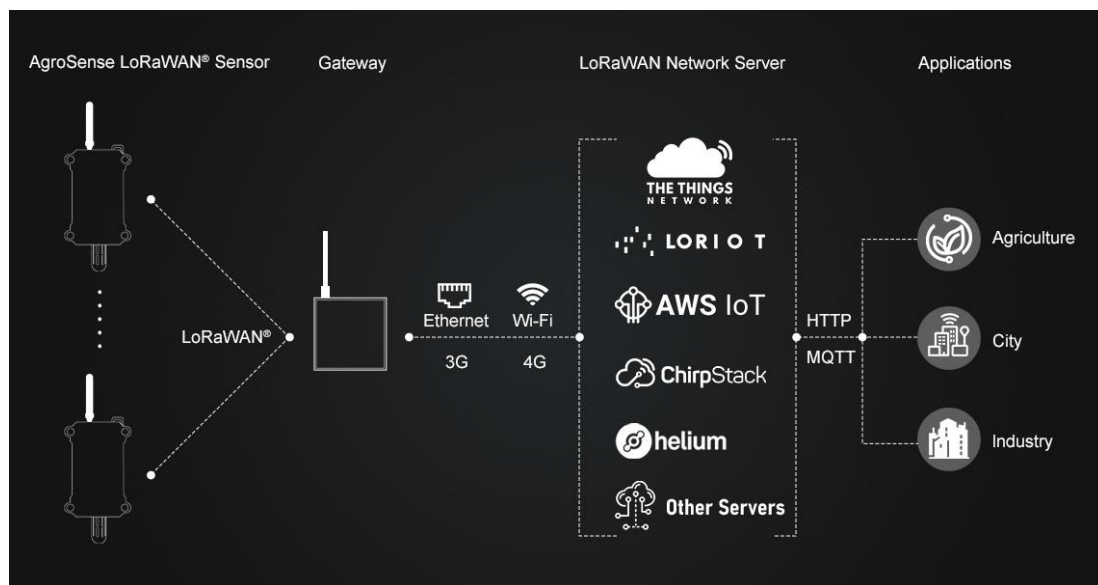
Lead Length	2 meter (custom length available)
Power Supply	1 x 18650 3.7V Lion batteries
Operating Temperature	-40°C ~60°C
Protection Class	IP67
Dimensions	131 × 62.7 × 27.5 mm
Mounting	Wall Mounting

## 2 Technical route

### 2.1 System Framework

AgroSense\_Leaf Moisture Sensor 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.



The steps to achieve the detection of leaf moisture is:

1. Collect the leaf moisture data by sensor, and send the data from End Node to Gateway.
2. The Gateway packages node data and transmits it to the Network Server.
3. The Network Server decodes the data and sends it to the Applications.
4. Finally, user can monitor the leaf moisture in the APP.

### 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

We use The Things Network as our Network Server, we need to configuration the country/ area frequency, inputting DEV EUI/ APP EUI/ APP Key, decodes, and connect to ThingSpeak or Datacake.

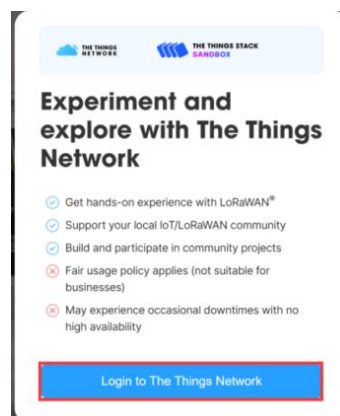
DEV EUI	Unique identification of device, authorized by IEEE
APP EUI	Unique identification of application
APP Key	One of the join network parameters on OTAA mode, calculated by DE EUI

- End Nodes and Gateway: AgroSense\_Leaf Moisture Sensor LoRaWAN®. (The AgroSense series is applicable)
- Network Server: The Things Network. ( Loriot, AWS IoT, ChirpStack, ect)
- Application Server: ThingSpeak.(Datacake, Blockbax, akenza, ect)

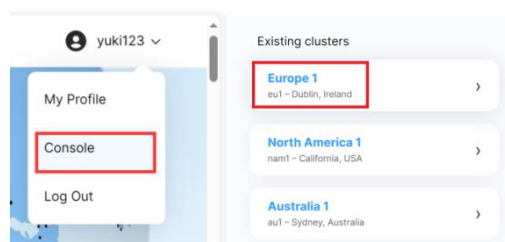
### 3.1 TTN and ThingSpeak

#### 3.1.1 Network Server configuration

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

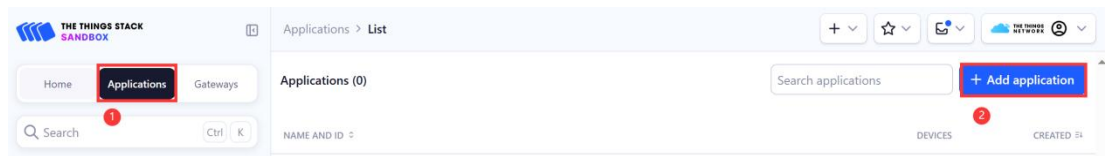


- Click “Console” and select clusters. (we take the European region for example.)





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



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

**Application ID \***

agrosense-sensor

**Application name**

My new application

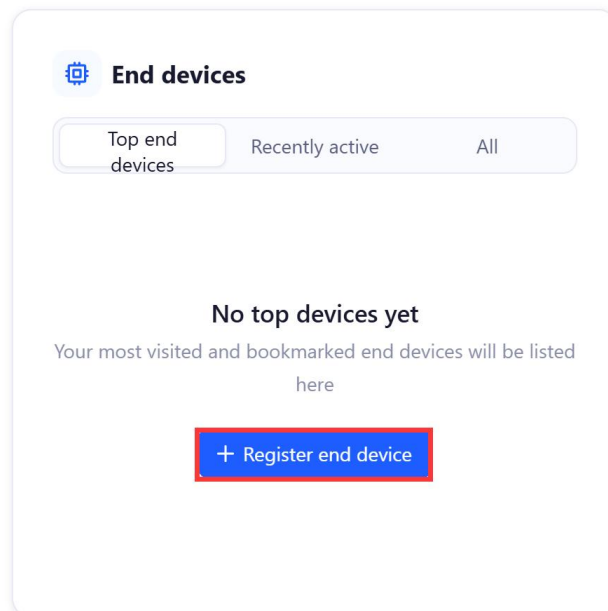
**Description**

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, 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.

**End device type**

Input method ⓘ

☐ Select the end device in the LoRaWAN Device Repository

☒ Enter end device specifics manually 1

**Frequency plan** ⓘ \*

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

**LoRaWAN version** ⓘ \*

LoRaWAN Specification 1.0.3 | v

**Regional Parameters version** ⓘ \*

RP001 Regional Parameters 1.0.3 revision A | v 2

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

**Provisioning information**

**JoinEUI** ⓘ \*

48 FF 00 00 00 00 01 65 3 4 Confirm

Continue, please enter the JoinEUI and device so we can determine onboarding options

**Provisioning information**

**JoinEUI** ⓘ \*

48 FF 00 00 00 00 01 65 Reset

This end device can be registered on the network

**DevEUI** ⓘ \*

48 E6 63 FF FE 30 01 65 Generate 0/50 used

**AppKey** ⓘ \*

4A 35 62 6B 95 AB 5B 4D 3F 3B DE 12 71 B1 6F 2A Generate

**End device ID** ⓘ \*

eui-48e663ffe300165

This value is automatically prefilled using the DevEUI

**After registration**

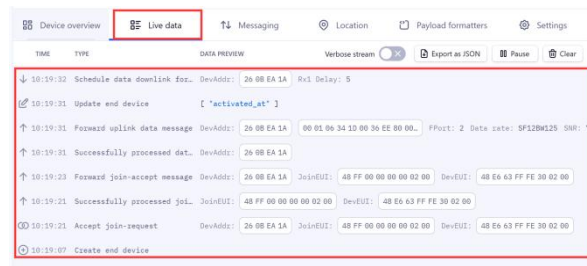
☒ View registered end device

☐ Register another end device of this type

Register end device

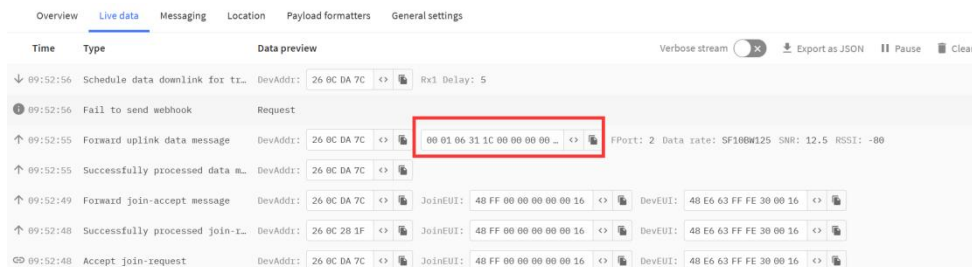


- Plug the battery and press RES button, you can see the device is connected successfully in the TTN.



### 3.1.2 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	Battery voltage		The value is obtained by amplifying the data by 10 times, and the actual value needs to be divided by 10 to convert to the actual battery voltage. The purpose of multiplying by 10 is to retain one decimal place of the voltage value. For example, if the value is 0x21 = 33, then the battery voltage is 3.3V.
byte 3	The data validity flag		0 is invalid, 1 is valid.
byte 4	Humidity sensor bits 8 to 15		This value is obtained after magnifying the data by 10 times. To obtain the actual relative humidity value, the real value needs to be calculated by dividing

### AgroSense\_Leaf Moisture Sensor LoRaWAN®

<b>byte 5</b>	Humidity sensor bits 0 to 7		it by 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.
<b>byte 6</b>	Temperature sensor bits 8 to 15		This value is obtained after magnifying the data by 10 times. To obtain the actual Temperature value, the real value needs to be calculated by dividing it by 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 Temperature is 6.45°C.
<b>byte 7</b>	Temperature sensor bits 0 to 7		
<b>byte 8</b>	data transmission interval bits 24 to 31		The time interval for data transmission has been increased by a factor of 100.
<b>byte 9</b>	data transmission interval bits 16 to 23		
<b>byte 10</b>	data transmission interval bits 8 to 15		
<b>byte 11</b>	data transmission interval bits 0 to 7		

Example: 0x00, 0x01, 0x28, 0x01, 0x00, 0x33, 0x01, 0x3F, 0x00, 0x36, 0xEE, 0x80

Data parsing:

Battery voltage is 4V.

Humidity value is 5.1%.

Temperature value is 31.9°C.

data transmission interval value is 3600.

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

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

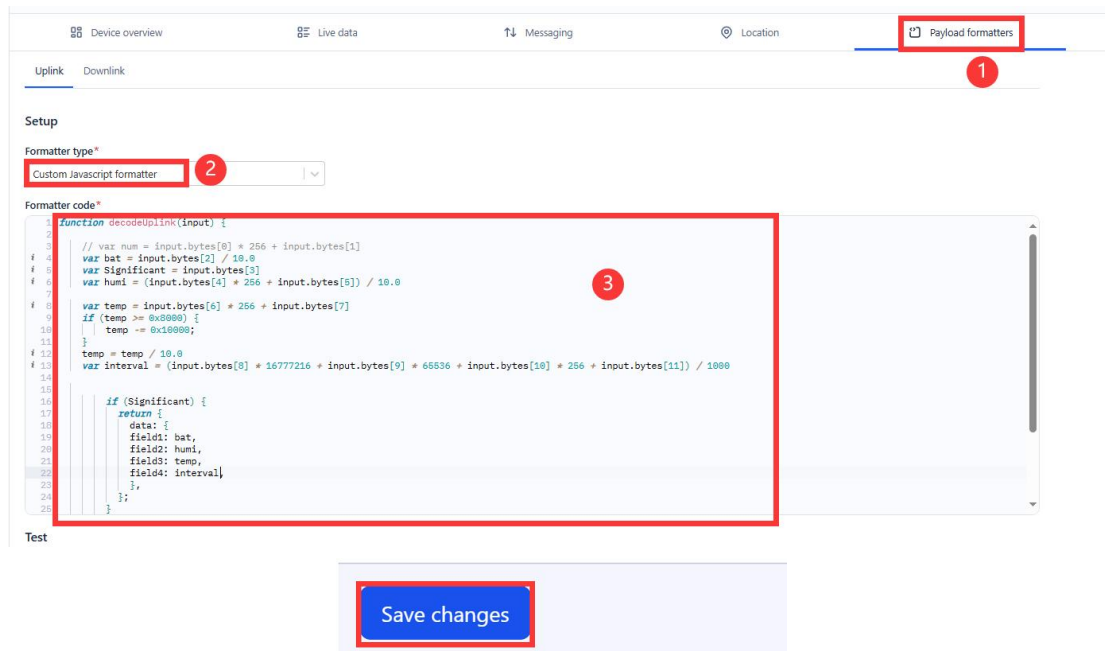
```
    // var num = input.bytes[0] * 256 + input.bytes[1]
    var bat = input.bytes[2] / 10.0
    var Significant = input.bytes[3]
    var humi = (input.bytes[4] * 256 + input.bytes[5]) / 10.0

    var temp = input.bytes[6] * 256 + input.bytes[7]
    if (temp >= 0x8000) {
        temp -= 0x10000;
    }
    temp = temp / 10.0
```

```
var interval = (input.bytes[8] * 16777216 + input.bytes[9] * 65536 + input.bytes[10] * 256 + input.bytes[11])
/ 1000
```

```
if (Significant) {
  return {
    data: {
      field1: bat,
      field2: humi,
      field3: temp,
      field4: interval,
    },
  };
}
else {
  return {
    data: {
      Significant: "data invalid",
    },
  };
}
}
```

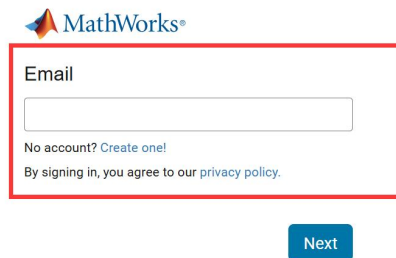
- Select “Payload formatters” and follow the steps.



### 3.1.3 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®

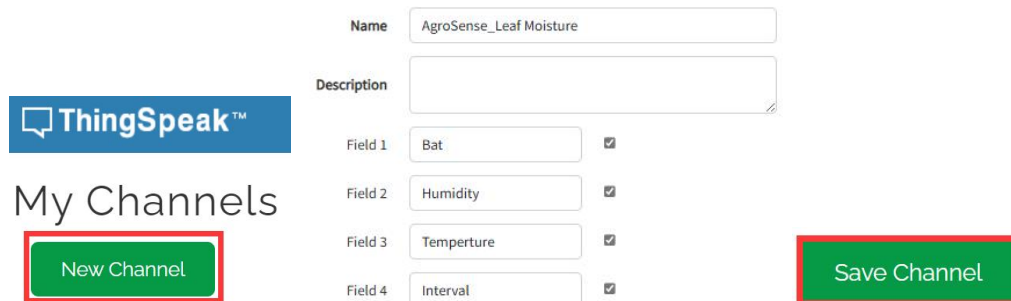
Email

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\_Leaf Moisture

Description:

Field 1: Bat ☒

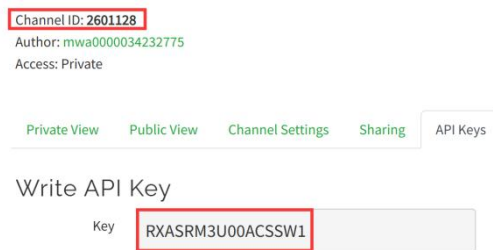
Field 2: Humidity ☒

Field 3: Temperture ☒

Field 4: Interval ☒

Save Channel

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



Channel ID: 2601128

Author: mwa0000034232775

Access: Private

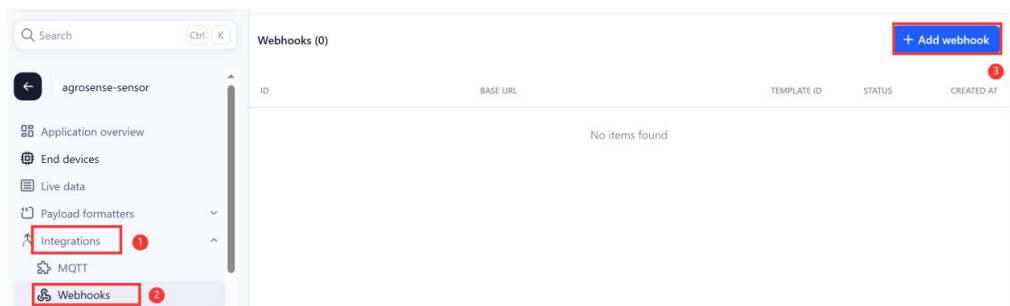
Private View Public View Channel Settings Sharing API Keys

Write API Key

Key: RXASRM3U00ACSSW1

### 3.1.4 Connect the Network Server and Application Server

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



Search  Ctrl K

agrosense-sensor

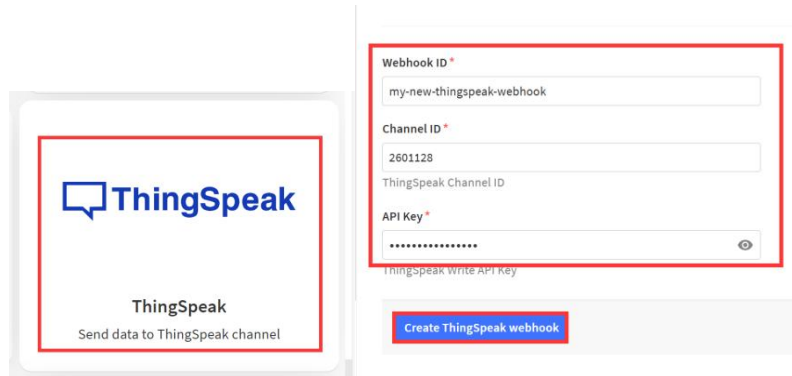
- Application overview
- End devices
- Live data
- Payload formatters
- Integrations 1
- MQTT
- Webhooks 2

Webhooks (0)

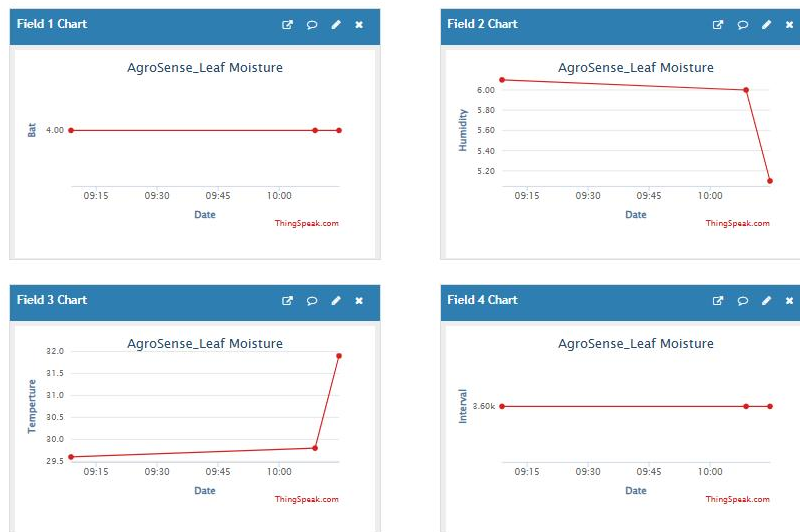
+ Add webhook 3

ID	BASE URL	TEMPLATE ID	STATUS	CREATED AT
No items found				

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



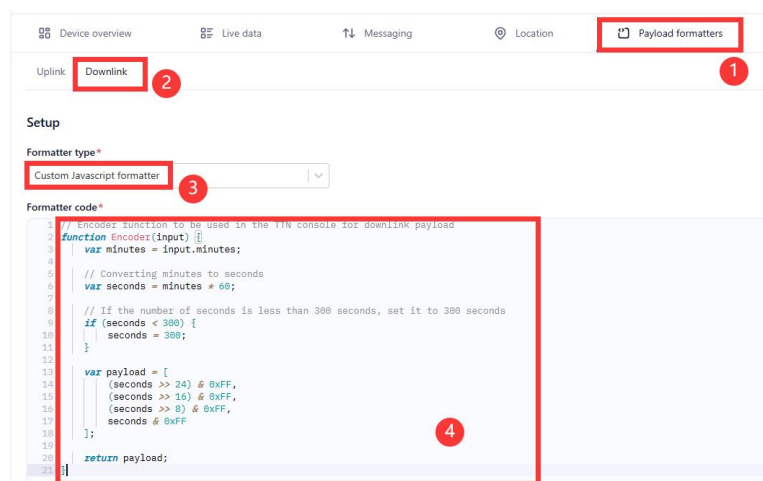
- Press RES button, wait about a minute, you will successfully see the data in ThingSpeak.(You will receive the data every hour.)



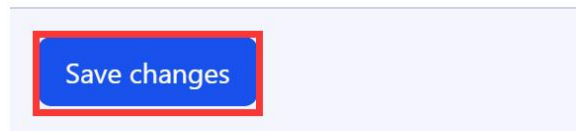
### 3.1.5 Change Time Interval (5-1440min)

1、If you need to change time Interval (Default 60 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

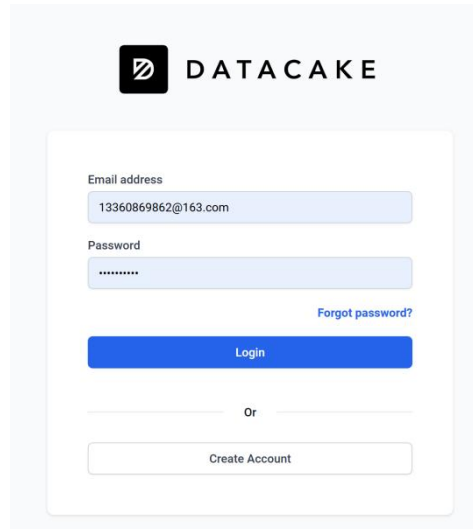
**Schedule downlink**

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



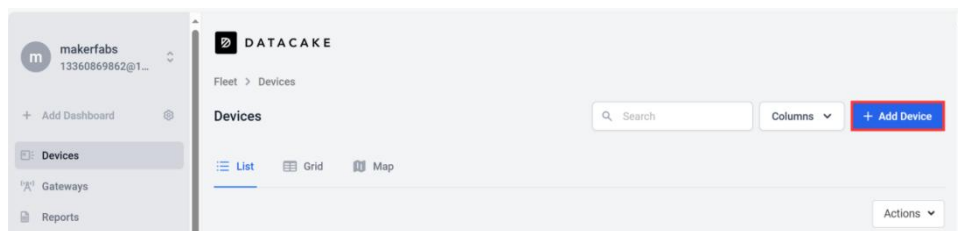
## 3.2 Datacake

### 1、Login datacake or Create Account

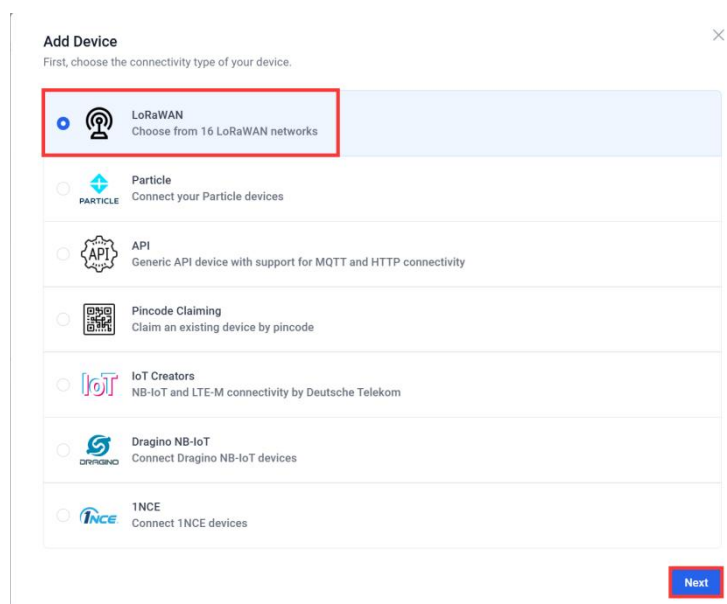


The image shows the Datacake login and registration interface. At the top is the Datacake logo. Below it is a form with two input fields: 'Email address' containing '13360869862@163.com' and 'Password' with masked characters. A 'Forgot password?' link is next to the password field. Below the fields is 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. It has a title bar 'Add Device' and a subtitle 'First, choose the connectivity type of your device.' Below is a list of connectivity options, each with a radio button and an icon:
 

- LoRaWAN** (selected, highlighted with a red box): Choose from 16 LoRaWAN networks
- Particle: Connect your Particle devices
- API: Generic API device with support for MQTT and HTTP connectivity
- Pincode Claiming: Claim an existing device by pincode
- IoT Creators: NB-IoT and LTE-M connectivity by Deutsche Telekom
- Dragino NB-IoT: Connect Dragino NB-IoT devices
- 1NCE: Connect 1NCE devices

 At the bottom right is a red-bordered 'Next' button.

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

5、Select "Datacake LNS"

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.

DEVICE	PRIMARY	SECONDARY	DEVICE SIGNAL	DEVICE BATTERY
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

9、Click “Configuration”, enter Decoder and click “Save”.(You can check it out on [Guihub](#))



```
function Decoder(payload, port) {
    var input = {
        bytes: payload
    };

    // var num = input.bytes[0] * 256 + input.bytes[1];
    var bat = input.bytes[2] / 10.0
    var Significant = input.bytes[3]
    var humi = (input.bytes[4] * 256 + input.bytes[5]) / 10.0

    var temp = input.bytes[6] * 256 + input.bytes[7]
    if (temp >= 0x8000) {
        temp -= 0x10000;
    }
    temp = temp / 10.0
    var interval = (input.bytes[8] * 16777216 + input.bytes[9] * 65536 + input.bytes[10] * 256 + input.bytes[11])
    / 1000

    var decoded = {
        bat: bat,
        Significant: Significant,
        humi: humi,
        temp: temp,
        interval: interval,
    };

    // Test for LoRa properties in normalizedPayload
    try {
        console.log('normalizedPayload:', normalizedPayload); // Log to check normalizedPayload structure

        decoded.lora_rssi =
            (normalizedPayload.gateways    &&    Array.isArray(normalizedPayload.gateways)    &&
normalizedPayload.gateways.length > 0 && normalizedPayload.gateways[0].rssi) || 0;
        decoded.lora_snr =
            (normalizedPayload.gateways    &&    Array.isArray(normalizedPayload.gateways)    &&
normalizedPayload.gateways.length > 0 && normalizedPayload.gateways[0].snr) || 0;
        decoded.lora_datarate = normalizedPayload.data_rate || 'not retrievable';
    } catch (error) {
        console.log('Error occurred while decoding LoRa properties: ' + error);
    }

    if (Significant) {
        return [
            { field: "bat", value: decoded.bat },

```

```

    { field: "humi", value: decoded.humi },
    { field: "temp", value: decoded.temp },
    { 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 }
  ];
}
else {
  return [
    { field: "Significant", value: "data invalid" },
  ];
}
}

```

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

#### Add Field

Fields define the schema of the data the device stores.

Type: Float

Name: bat

Identifier: BAT

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

#### Fields

Fields describe the data the device will store.

NAME	IDENTIFIER	TYPE
bat	BAT	Float
humi	HUMI	Float
temp	TEMP	Float
interval	INTERVAL	Float

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

### Fields

Fields describe the data the device will store.

+ Add Field

Live data

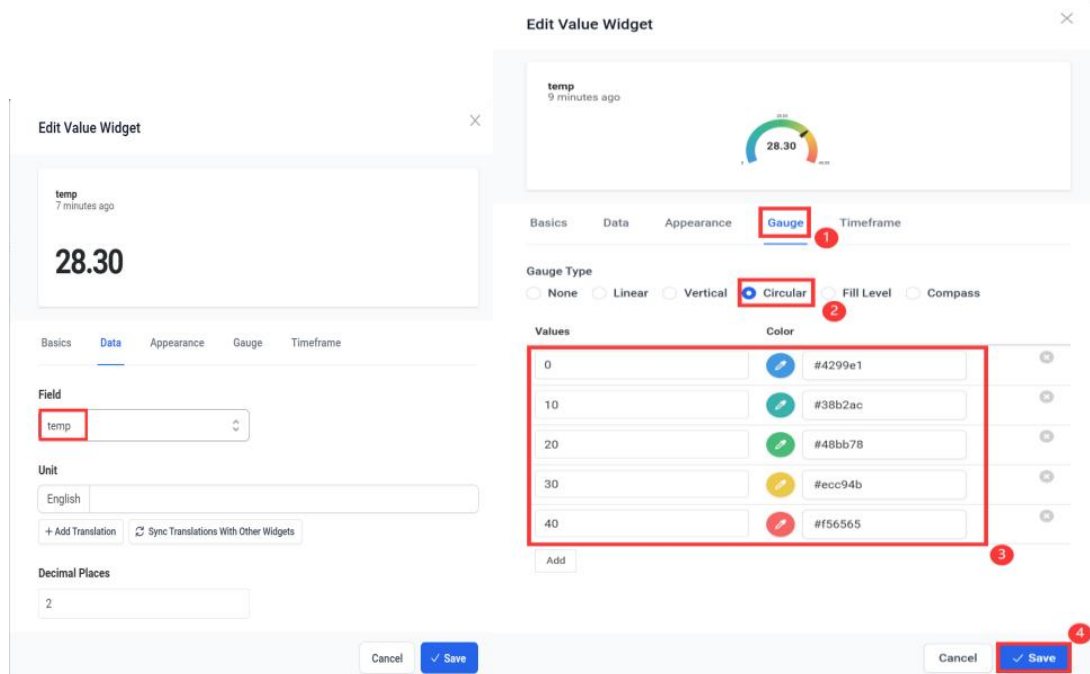
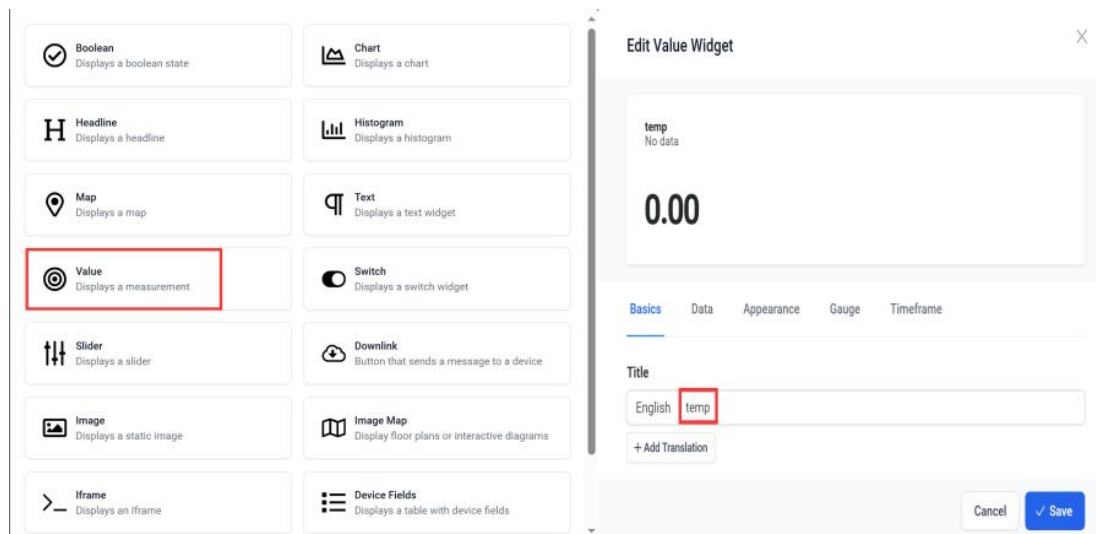
NAME	IDENTIFIER	TYPE	ROLE	CURRENT VALUE	LAST UPDATE
bat	BAT	Float	N/A	4.1	3 minutes ago
humi	HUMI	Float	N/A	7.3	3 minutes ago
temp	TEMP	Float	N/A	28.1	3 minutes ago
interval	INTERVAL	Float	N/A	3,600	3 minutes ago

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

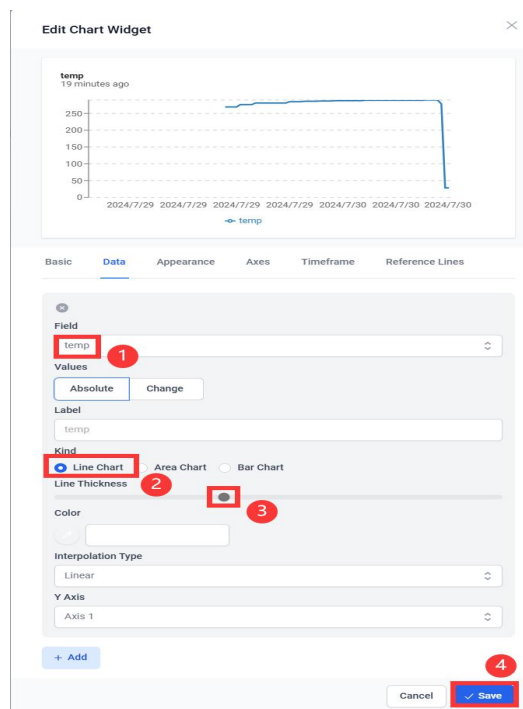
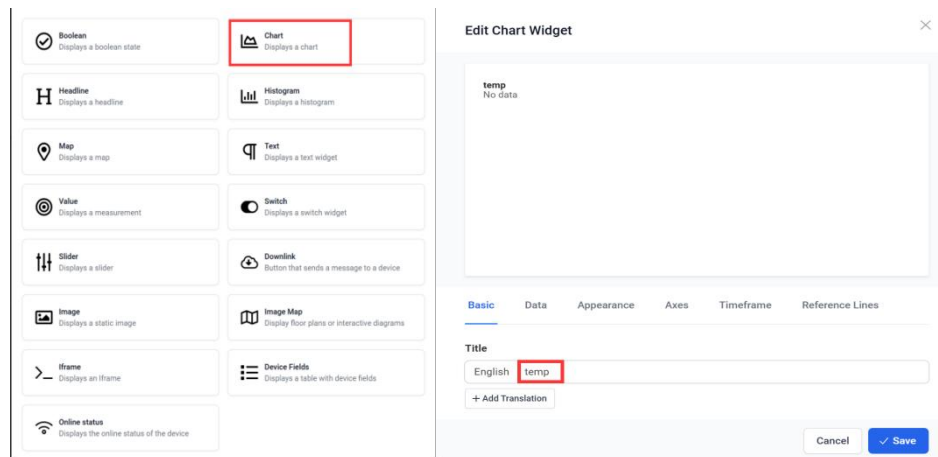
Click “Dashboard-->switch-->+ Add Widget”.



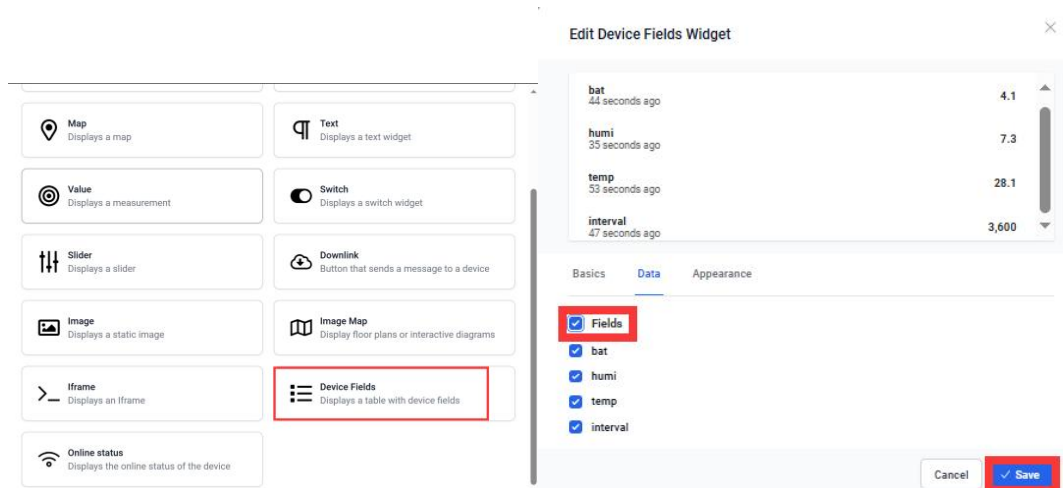
13、Select “Value” and set Title, Field and presentation form as well as the interval color.



14、Select Chart and set Title, Field, Kind, Line Thickness and click “save”.

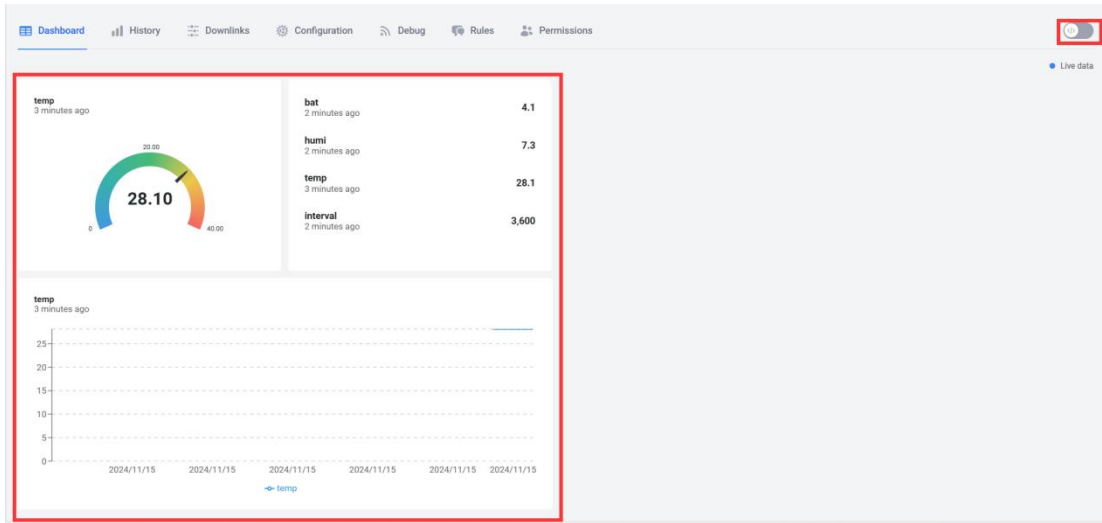


15、Select Device Fields, check “Fields” and click “Save”.



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

17、The steps for humidity are the same as above, and you can add your own.

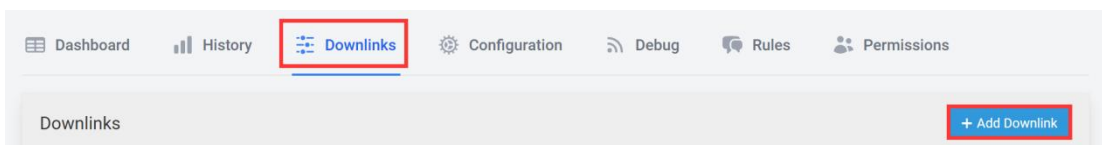


### 3.2.1 Change Time Interval (5-1440min)

1、If you need to change time Interval (Default 60 minutes), you can click "Configuration-->Fields-->+Add Field"

The 'Add Field' dialog box is shown with the following fields: 'Type' (Integer), 'Name' (Sending Time Interval), 'Identifier' (SENDING\_TIME\_INTERVAL), 'Unit' (Optional), 'Role' (None), and 'Formula' (Optional). The 'Add Field' button is highlighted in red.

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.(5Min-1440Min)

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

**Configure Downlink**

**Name**  
Set User-Defined Sending Time Interval

**Description**  
Set the user-defined report transmission interval and store it in the configuration variable.(5Min-1440Min)

**Fields used**  
If your encoder function takes input from the device's fields, you can specify them here. They will be used to create the form for the downlink generator.  
SENDING\_TIME\_INTERVAL

☐ Trigger on measurements  
If activated, each time the device records a measurement in one of the fields used, the downlink will be sent automatically.

**Port**  
1

**Payload Encoder**

```

1 function Encoder(measurements, port) {
2   var interval = measurements["SENDING_TIME_INTERVAL"].value * 60;
3   if (interval < 300) {
4     interval = 300;
5     console.log("Interval < 300 Seconds / 5 Minutes not allowed!");
6   }
7   // Convert to hexadecimal only from interval
8   return interval.toString(16).padStart(4, '0').match(/.{2}/g).map(function(f) {return parseInt(f, 16)
9 }
10 /**
11  * String.prototype.padStart() polyfill
12  * https://github.com/uxitten/polyfill/blob/master/string.polyfill.js
13  * https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/padStart
14  */
15 if (!String.prototype.padStart) {
16   String.prototype.padStart = function padStart(targetLength,padString) {
17     targetLength = targetLength>>0; //truncate if number or convert non-number to 0;
18     padString = String((typeof padString !== 'undefined' ? padString : ' '));
19     if (this.length > targetLength) {
20       return String(this);
21     } else {
22       targetLength = targetLength-this.length;
23       if (targetLength > padString.length) {
24         padString += padString.repeat(targetLength/padString.length); //append to original to en
25       }
26     }
27   }
28 }

```

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

Select “Downlink” and setting as follow image.

**Edit Downlink Widget**

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

**Basics** Data Appearance

**Title**  
English User-Defined Time Interval(5Min-1440Min)  
German + Add Translation

Cancel Save

**Edit Downlink Widget**

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

Basics **Data** Appearance

**Downlink**  
Set User-Defined Sending Time Interval

**Additional Downlinks**  
+ Add

Cancel Save

- 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