



**AgroSense LoRaWAN® Temperature
& Humidity Sensor Manual
V1.1**

Author: Yuki

Time: 2024.7.30

| Date | Versions | Description | Author |
|-----------|----------|--------------------------------|--------|
| 2024.7.15 | V1.0 | Introduction to Use & Function | Yuki |
| 2024.7.30 | V1.1 | Add use in Datacake | Yuki |

Contents

| | |
|--|----------|
| 1 Product Description | 1 |
| 1.1 Introduction | 1 |
| 1.2 Feature | 1 |
| 1.3 Parameter | 2 |
| 2 Technical route | 3 |
| 2.1 System Framework | 3 |
| 2.2 Regional frequency band | 4 |
| 3 Usage | 5 |
| 3.1 TTN and ThingSpeak | 5 |
| 3.1.1 Network Server configuration | 5 |
| 3.1.2 Decoder | 8 |
| 3.1.3 Application Server configuration | 10 |
| 3.2 Datacake | 12 |

1 Product Description

1.1 Introduction

AgroSense LoRaWAN® Temperature & Humidity Sensor measures temperature and humidity in the atmosphere at the range of -40°C to 85°C and 0 to 100 %RH with accuracy $\pm 0.2^{\circ}\text{C}$ and $\pm 0.2\% \text{RH}$ 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 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.
- **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°C ~ 85°C, 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**.

1.3 Parameter

1. General Parameters

| | |
|----------------------------------|--------------|
| Product Model | AGLWTH01 |
| Temperature Measurement Range | -60°C ~200°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 |

2. Wireless Parameters

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

3. Physical Parameters

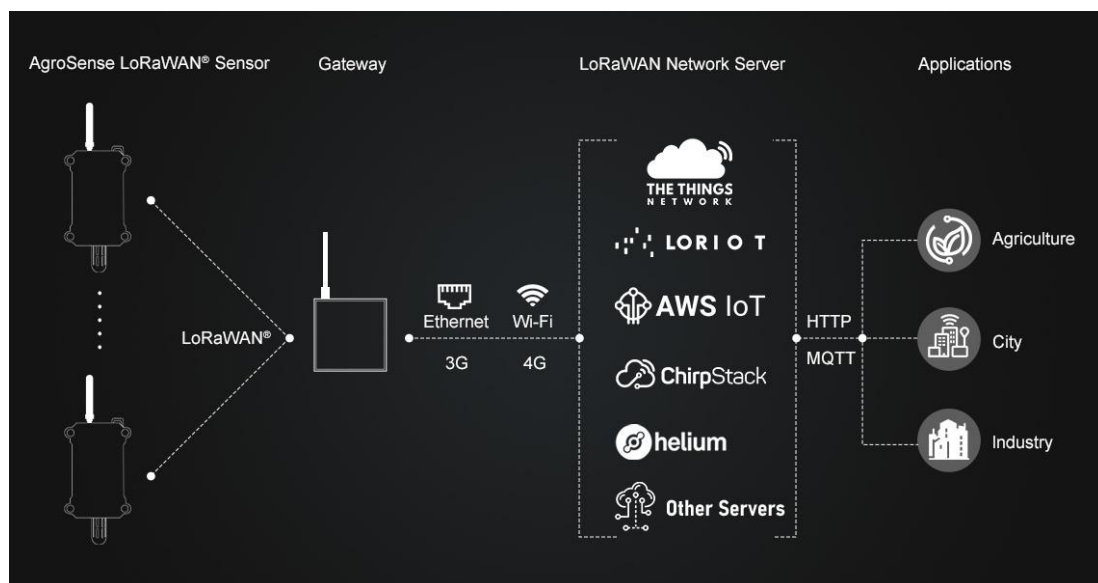
| | |
|-----------------------|------------------------|
| Power Supply | 2 x AAA 1.5V batteries |
| Operating Temperature | -40°C ~85°C |
| Protection Class | IP68 |
| Dimensions | 131 × 62.7 × 27.5 mm |
| Mounting | Wall Mounting |

2 Technical route

2.1 System Framework

AgroSense LoRaWAN® Temperature & Humidity 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 air temperature and humidity is:

1. Collect the air temperature and humidity 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 temperature and humidity 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.

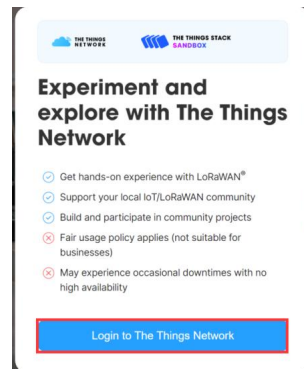
| | |
|---------|---|
| DEV EUI | Unique identification of device, authorized by IEEE |
| APP EUI | Unique identification of application |
| APP Key | One of the join network parameters on OTAA mode, calculated by DE EUI |

- End Nodes and Gateway: AgroSense LoRaWAN® Temperature & Humidity Sensor. (The AgroSense series is applicable)
- Network Server: The Things Network. (Lorient, AWS IoT, ChirpStack, ect)
- Application Server: ThingSpeak.(Datacake, Blockbox, akenza, ect)

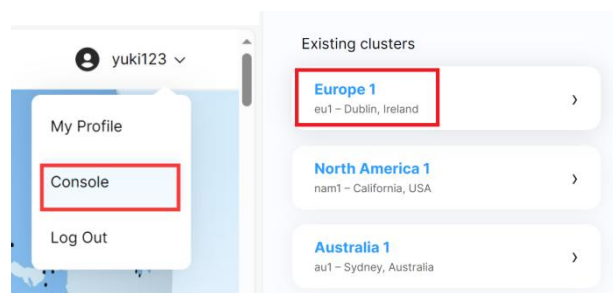
3.1 TTN and ThingSpeak

3.1.1 Network Server configuration

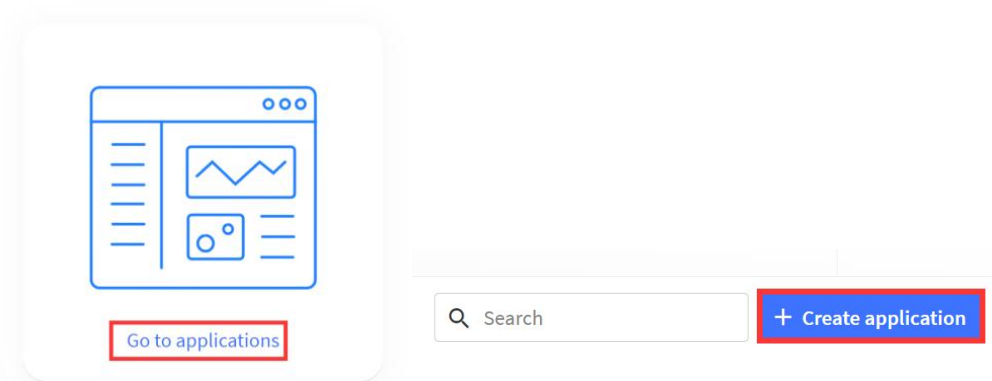
- Open The Things Network in your browser and login it. (Or register an account)



- 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-t-h-sensor

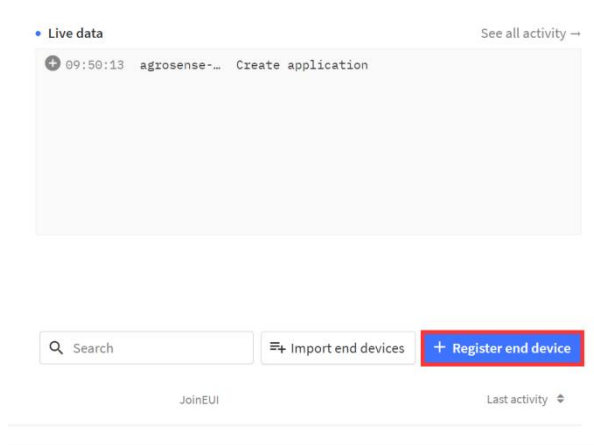
Application name
My new application

Description
Description for my new application

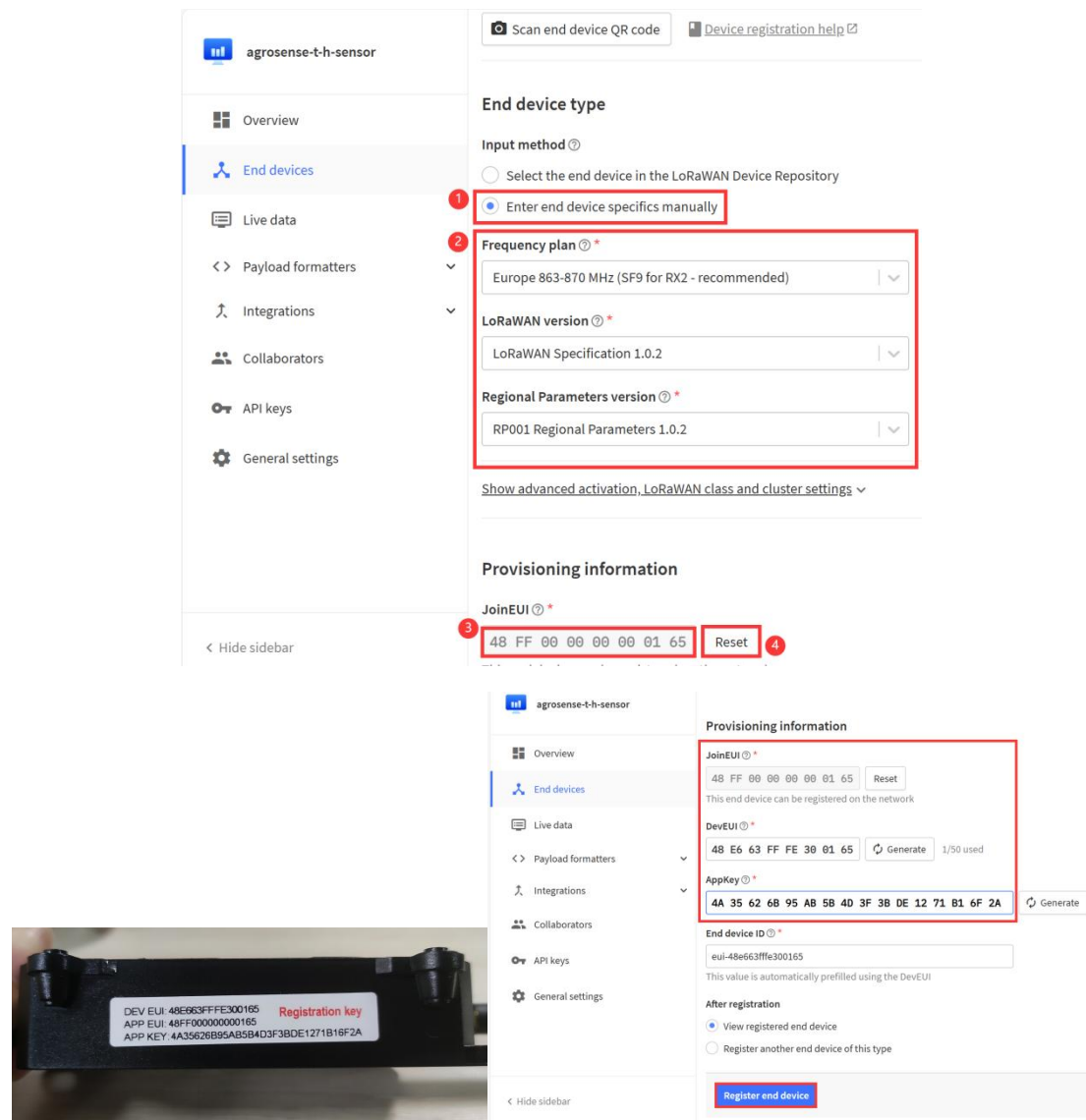
Optional application description; can also be used to save notes about the app

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.



agrosense-t-h-sensor

Scan end device QR code Device registration help

Overview

End devices

Live data

Payload formatters

Integrations

Collaborators

API keys

General settings

End device type

Input method

Select the end device in the LoRaWAN Device Repository

Enter end device specifics manually

Frequency plan

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

LoRaWAN version

LoRaWAN Specification 1.0.2

Regional Parameters version

RP001 Regional Parameters 1.0.2

Show advanced activation, LoRaWAN class and cluster settings

Provisioning information

JoinEUI

48 FF 00 00 00 01 65 Reset

DevEUI

48 E6 63 FF FE 30 01 65 Generate 1/50 used

AppKey

4A 35 62 6B 95 AB 5B 40 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

DEV EUI: 48E663FFE300165 Registration key
APP EUI: 48FF00000000165
APP KEY: 4A35626B95AB5B403F3BDE1271B16F2A

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



Live data

See all activity

10:08:21 Schedule data downlink for transmission on Gateway Server De

10:08:20 Forward uplink data message DevAddr: 26 0B 2E 6F <> €

10:08:20 Successfully processed data message DevAddr: 26 0B 2E 6F <>

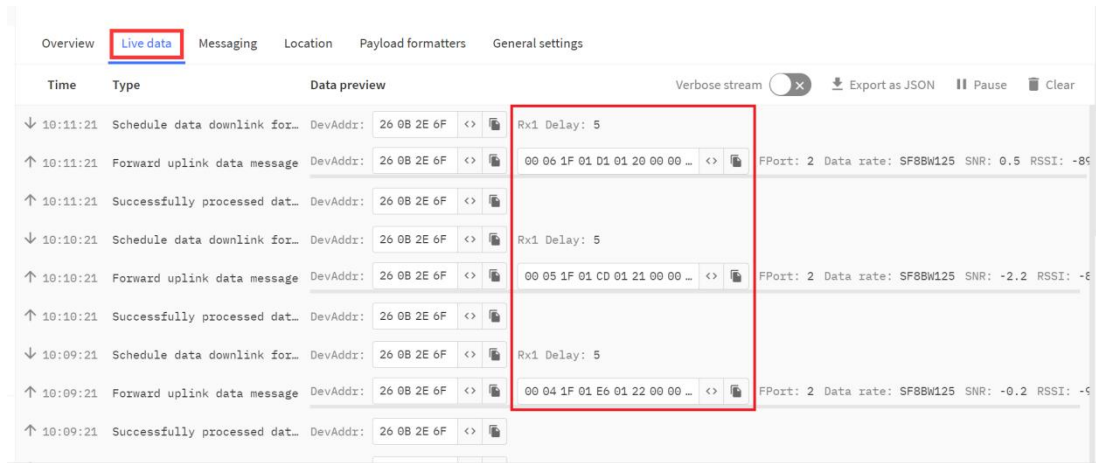
10:07:21 Schedule data downlink for transmission on Gateway Server De

10:07:21 Forward uplink data message DevAddr: 26 0B 2E 6F <> €

10:07:21 Successfully processed data message DevAddr: 26 0B 2E 6F <>

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 | Humidity sensor bits 8 to 15 | | This value is obtained by amplifying the data by 10 times. To obtain the actual relative Humidity value, the relative value needs to be converted by dividing by 10. For example, if the value of bits 8 to 15 is 0x02, and the low 8 bits value is 0x85, then the obtained temperature value is 0x0285 = 645. it is divided by 10 to get 64.5%RH. |
| byte 4 | Humidity sensor bits 0 to 7 | | |
| byte 5 | Temperature sensor bits 8 to 15 | | This value is obtained by amplifying the data by 10 times. To obtain the actual relative Temperature value, the relative value needs to be converted by dividing by 10. For example, if the value of bits 8 to 15 is 0x02, and the low 8 bits value is 0x85, then the obtained temperature value is 0x0285 = 645. it is divided by 10 to get 64.5°C. |
| byte 6 | Temperature sensor bits 0 to 7 | | |
| byte 7 | NC | | |
| byte 8 | NC | | |

Example: 0x00, 0x04, 0x1D, 0x01, 0x89, 0x00, 0xF8, 0x00, 0x00

Data parsing:

Battery voltage is 2.9V.

Humidity is 39.3%

Temperature is 24.8 °C.

- 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 humi = (input.bytes[3] * 256 + input.bytes[4]) / 10.0
  // var temp = (input.bytes[5] * 256 + input.bytes[6]) / 10.0
  var temp = input.bytes[5] * 256 + input.bytes[6]
  if (temp >= 0x8000) {
    temp -= 0x10000;
  }
  temp = temp / 10.0
  return {
    data: {
      field1: num,
      field2: bat,
      field3: humi,
      field4: temp,
    },
  };
}
```

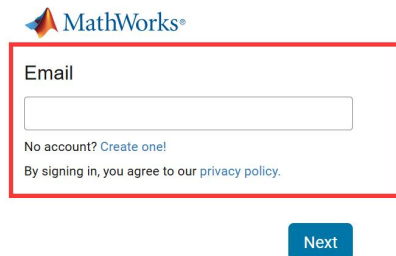
- Select “Payload formatters” and follow the steps.

The screenshot shows the 'Payload formatters' configuration page for a device with EUI ID: eui-48e663ffe300012. The page is divided into a sidebar and a main content area. The sidebar on the left contains navigation links: Overview, End devices, Live data, Payload formatters (selected), Integrations, Collaborators, API keys, and General settings. The main content area has tabs for Overview, Live data, Messaging, Location, Payload formatters (selected), and General settings. The 'Payload formatters' tab is active, showing a 'Setup' section. A red box highlights the 'Custom Javascript formatter' option in the 'Formatter type' dropdown. Another red box highlights the 'Formatter code' text area, which contains the JavaScript code for decoding the payload. A third red box highlights the 'Save changes' button at the bottom right. The code in the text area is the same as the one provided in the previous block.

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®

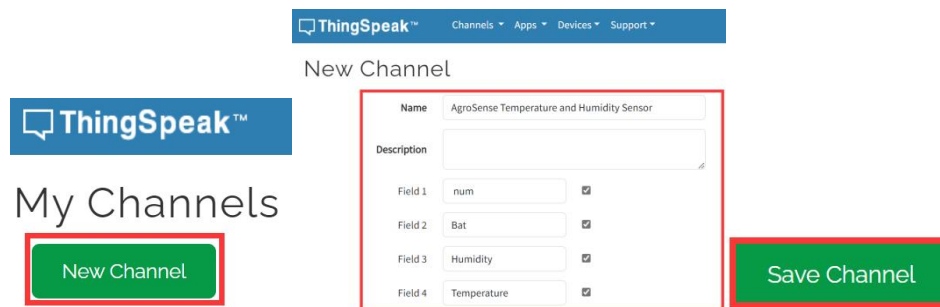
Email

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™ Channels Apps Devices Support

New Channel

ThingSpeak™

My Channels

New Channel

Name: AgroSense Temperature and Humidity Sensor

Description:

Field 1: num ☒

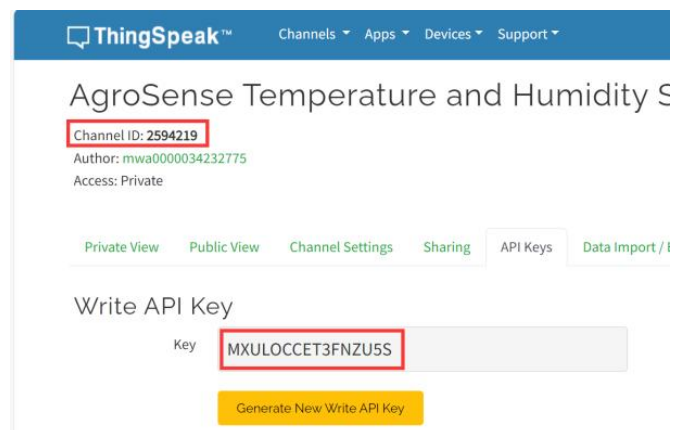
Field 2: Bat ☒

Field 3: Humidity ☒

Field 4: Temperature ☒

Save Channel

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



ThingSpeak™ Channels Apps Devices Support

AgroSense Temperature and Humidity S

Channel ID: 2594219

Author: mwa0000034232775

Access: Private

Private View Public View Channel Settings Sharing API Keys Data Import / i

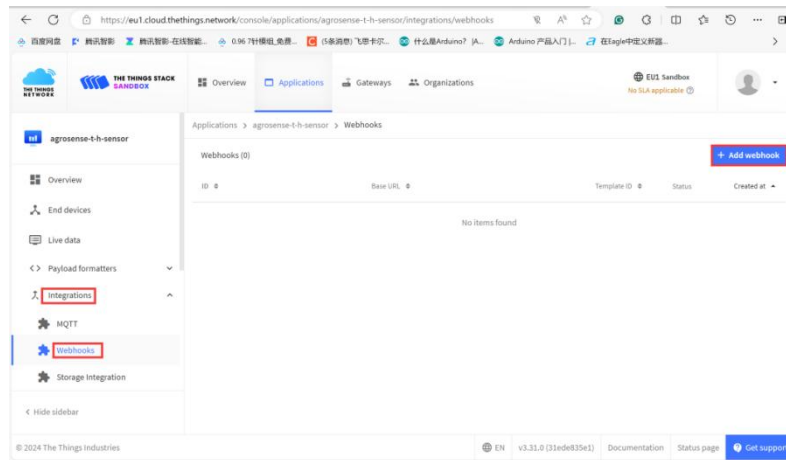
Write API Key

Key: MXULOCCE3FNZU5S

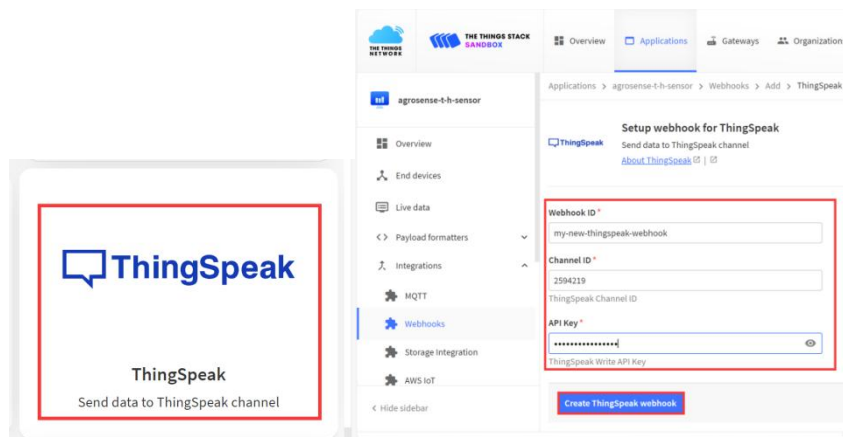
Generate New Write API Key

3.1.4 Connect the Network Server and Application Server

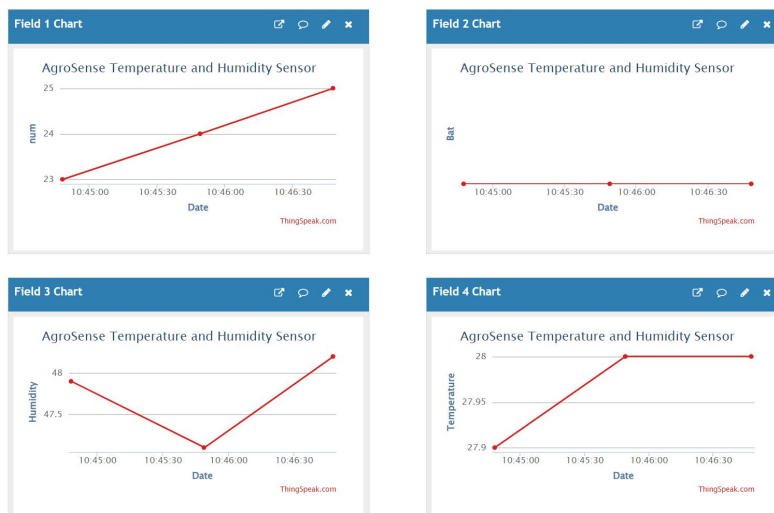
- In the TTN, click “integrations” --> “Webhooks” --> “+ Add webhook”.



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

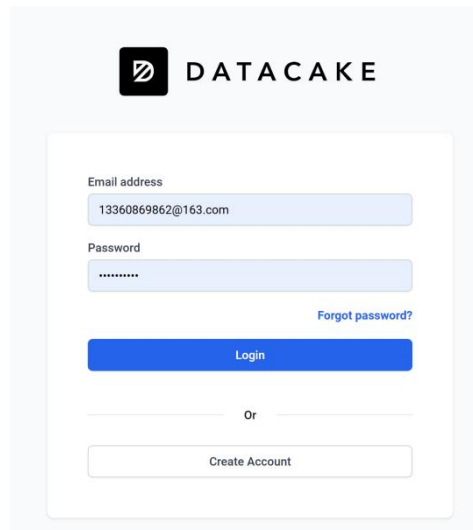


- Press RES button, wait about a minute, you will successfully see the data in ThingSpeak.(You will receive the data every hour.)



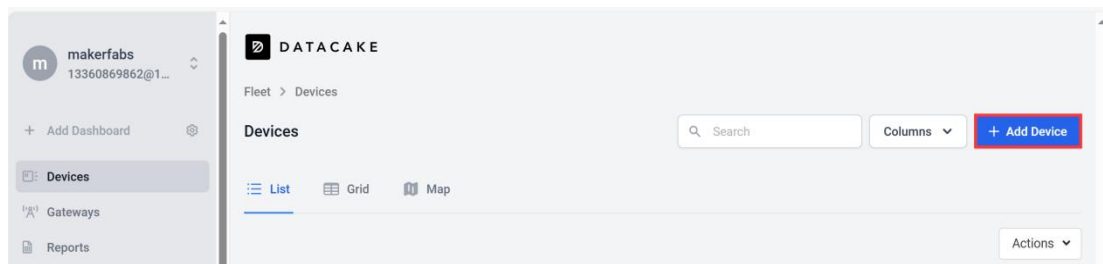
3.2 Datacake

1、Login datacake or Create Account

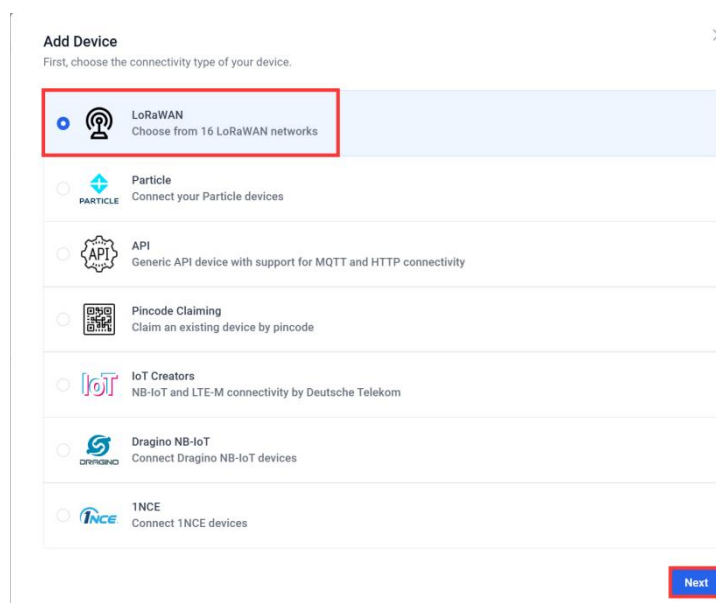


The image shows the Datacake login and registration interface. At the top is the Datacake logo. Below it is a form with two input fields: 'Email address' containing '13360869862@163.com' and 'Password' with masked characters. A 'Forgot password?' link is next to the password field. Below the fields is 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 prompts the user to 'First, choose the connectivity type of your device.' There is a list of options: 'LoRaWAN' (selected with a radio button and highlighted by a red box, with subtext 'Choose from 16 LoRaWAN networks'), 'Particle' (with subtext 'Connect your Particle devices'), 'API' (with subtext 'Generic API device with support for MQTT and HTTP connectivity'), 'Pincode Claiming' (with subtext 'Claim an existing device by pincode'), 'IoT Creators' (with subtext 'NB-IoT and LTE-M connectivity by Deutsche Telekom'), 'Dragino NB-IoT' (with subtext 'Connect Dragino NB-IoT devices'), and '1NCE' (with subtext 'Connect 1NCE devices'). A red-bordered 'Next' button is at the bottom right.

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 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 | Actions |
|--------|----------|-----------|---------------|----------------|---------|
| light | 97.49 lx | 41 | N/A | N/A | [Icons] |
| soil | 1,584 | 2.9 | N/A | N/A | [Icons] |
| T&H | N/A | N/A | N/A | N/A | [Icons] |

9、Click “Configuration”, enter Decoder and click “Save”.

```

function Decoder(payload, serial) {
  var input = 0;
  var bytes = payload;
  var num = input.bytes[0] * 256 + input.bytes[1];
  var hum = (input.bytes[2] * 256 + input.bytes[3]) / 100;
  var temp = (input.bytes[4] * 256 + input.bytes[5]) / 100;

  // Convert to Celsius
  if (input.temp) num = 0;
  if (input.hum) hum = 0;
  if (input.temp) temp = 0;

  return {
    "temp": num,
    "hum": hum,
    "temp": temp,
    "hum": hum
  };
}
    
```

```
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 humi = ( input.bytes[3] * 256 + input.bytes[4] ) / 10.0;  
    var temp = ( input.bytes[5] * 256 + input.bytes[6] ) / 10.0;  
  
    // 检查并处理 NaN 值  
    if (isNaN(num)) num = 0;  
    if (isNaN(bat)) bat = 0;  
    if (isNaN(humi)) humi = 0;  
    if (isNaN(temp)) temp = 0;  
  
    return [  
        {  
            field: "num",  
            value: num  
        },  
        {  
            field: "bat",  
            value: bat  
        },  
        {  
            field: "humi",  
            value: humi  
        },  
        {  
            field: "temp",  
            value: temp  
        }  
    ];  
}
```

10、Follow the steps to add a field.

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: num

Identifier: NUM

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

Add Field

Fields define the schema of the data the device stores.

Type: Float

Name: bat

Identifier: BAT

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

Add Field

Fields define the schema of the data the device stores.

Type: Float

Name: hum

Identifier: HUMI

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

Add Field

Fields define the schema of the data the device stores.

Type: Float

Name: temp

Identifier: TEMP

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

Fields

Fields describe the data the device will store.

[+ Add Field](#)

[Live data](#)

| NAME | IDENTIFIER | TYPE | ROLE | CURRENT VALUE | LAST UPDATE |
|-------------|-------------|-------|------|---------------|---------------|
| num | NUM | Float | N/A | 0 | 3 minutes ago |
| bat | BAT | Float | N/A | 0 | 2 minutes ago |
| humidity | HUMIDITY | Float | N/A | 0 | 2 minutes ago |
| temperature | TEMPERATURE | Float | N/A | 0 | in 0 seconds |

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

Fields + Add Field

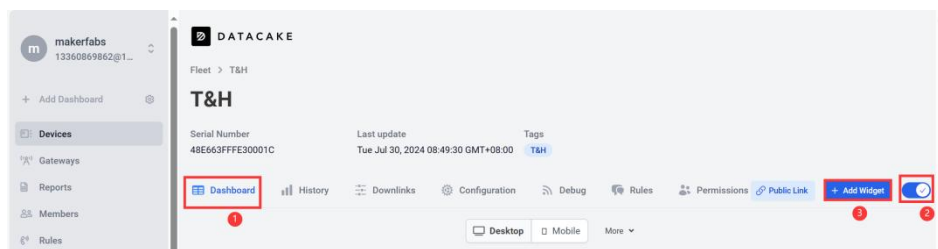
Fields describe the data the device will store.

Live data

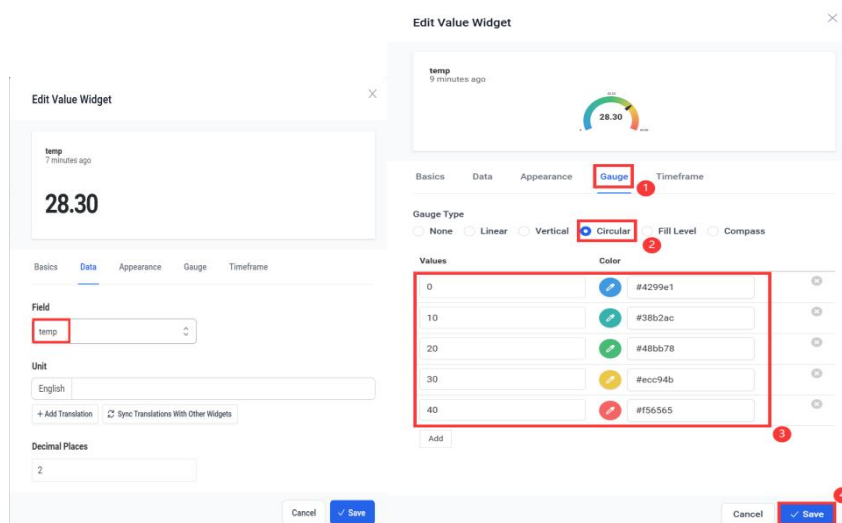
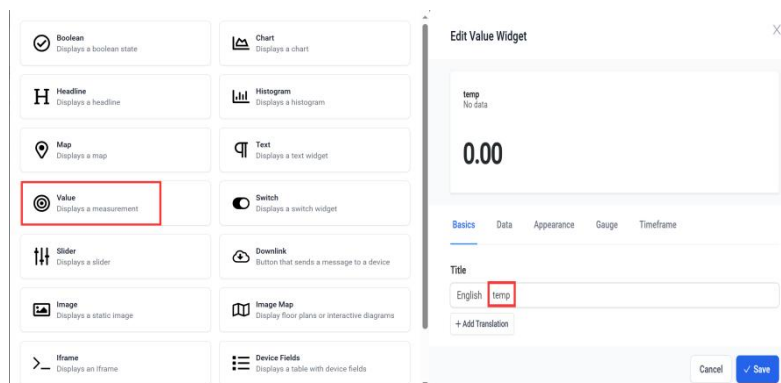
| NAME | IDENTIFIER | TYPE | ROLE | CURRENT VALUE | LAST UPDATE |
|-------|------------|-------|------|---------------|---------------|
| num | NUM | Float | N/A | 20 | 7 seconds ago |
| bat | BAT | Float | N/A | 2.9 | 7 seconds ago |
| humid | HUMI | Float | N/A | 62.8 | 7 seconds ago |
| temp | TEMP | Float | N/A | 28.3 | 7 seconds ago |

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

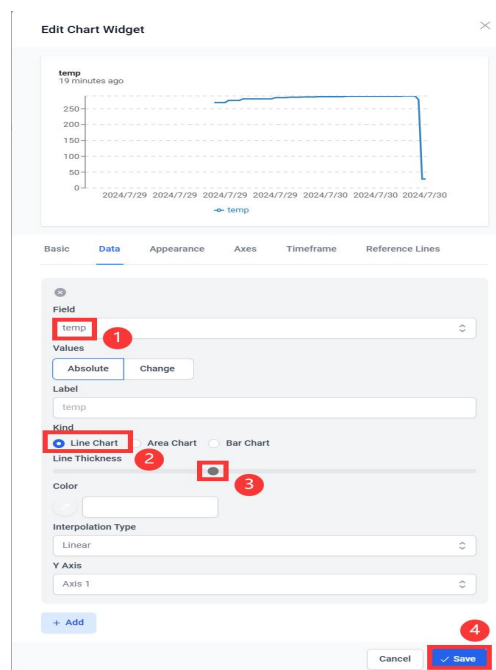
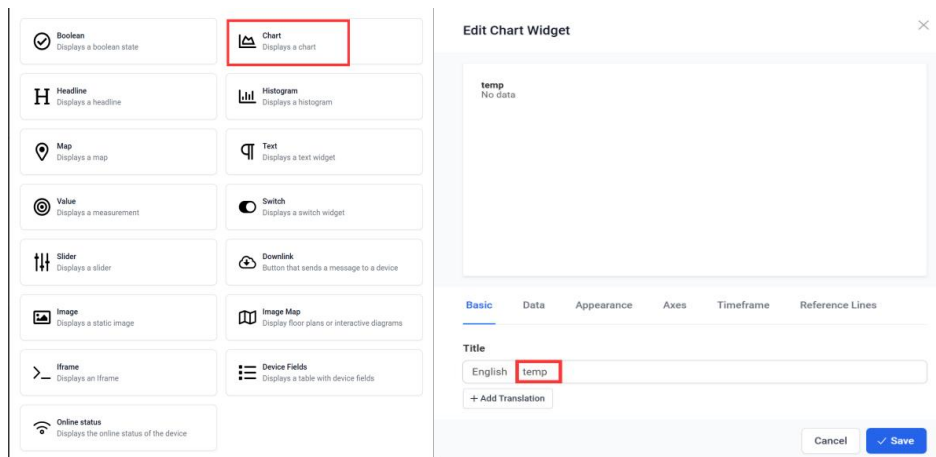
Click “Dashboard-->switch-->+ Add Widget”.



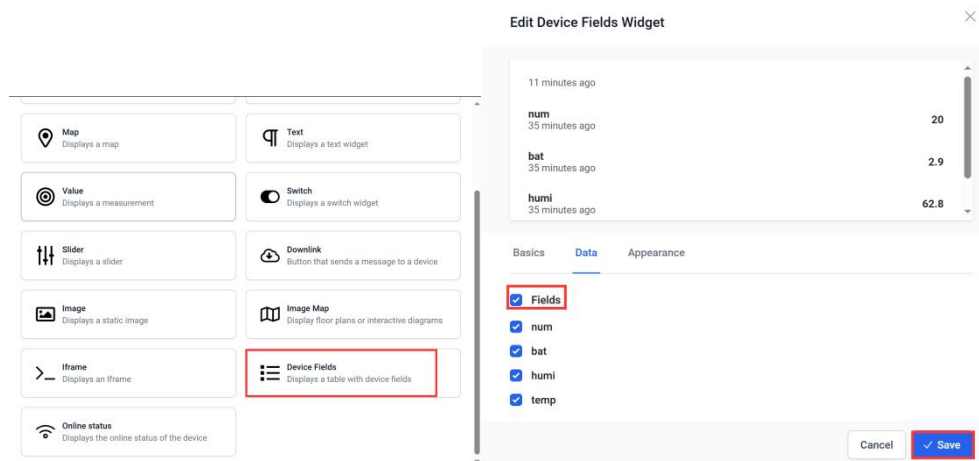
13、Select “Value” and set Title, Field and presentation form as well as the interval color.



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.

