



**AgroSense LoRaWAN[®] Temperature
& Humidity Sensor Manual
V1.0**

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 Network Server configuration	5
3.2 Decoder	8
3.3 Application Server configuration	10
3.4 Connect the Network Server and Application Server	11

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

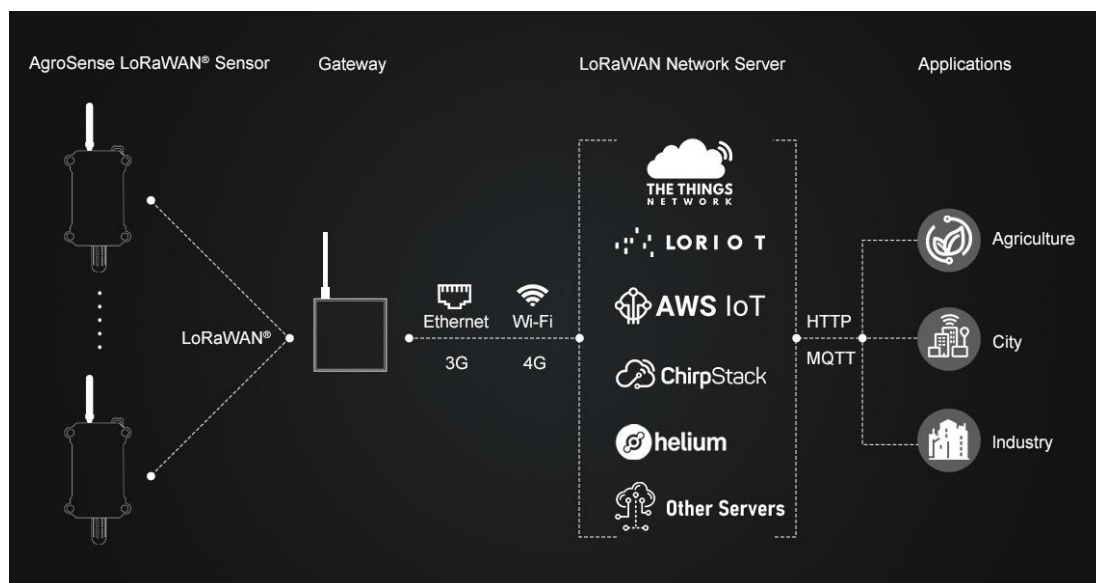
Power Supply	2 x AAA 1.5V batteries
Operating Temperature	-40°C ~85°C
Protection Class	IP68
Dimensions	131 × 62.7 × 27.5 mm
Mounting	Wall Mounting

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

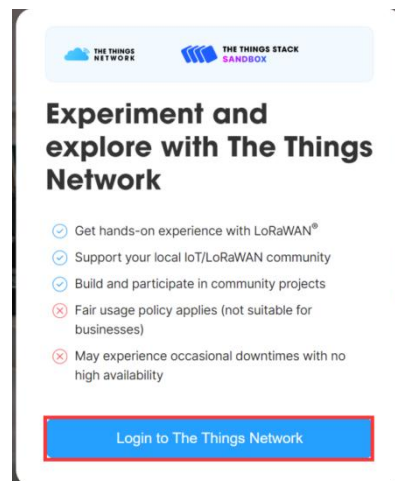
We use The Things Network as our Network Server, we need to configuration the country/ area frequency, inputting DEV EUI/ APP EUI/ APP Key, decodes, and connect to ThingSpeak.

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

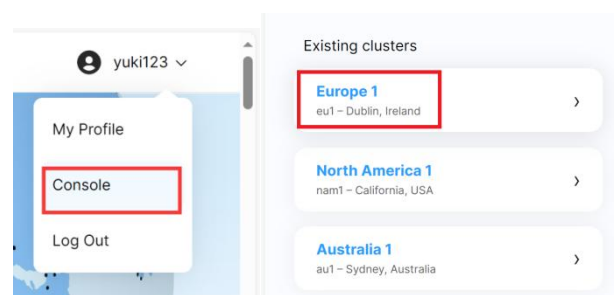
- End Nodes and Gateway: AgroSense LoRaWAN® Temperature & Humidity Sensor. (The AgroSense series is applicable)
- Network Server: The Things Network. (Loriot, AWS IoT, ChirpStack, ect)
- Application Server: ThingSpeak.(Datacake, Blockbox, akenza, ect)

3.1 Network Server configuration

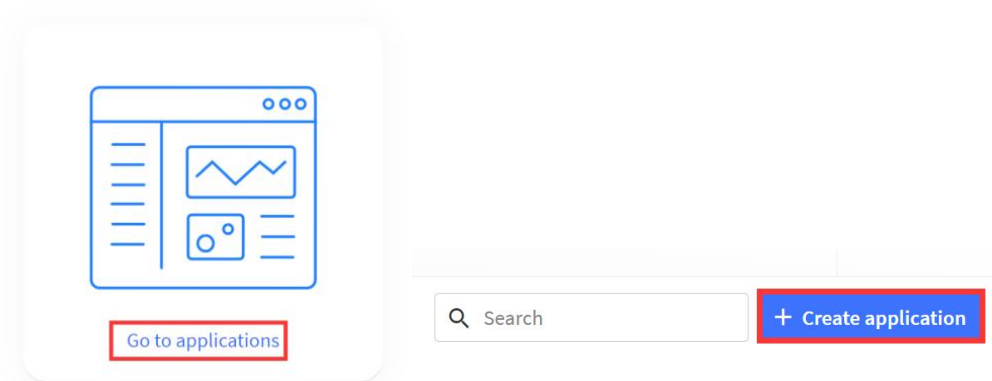
- Open The Things Network in your browser and login it. (Or register an account)



- 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

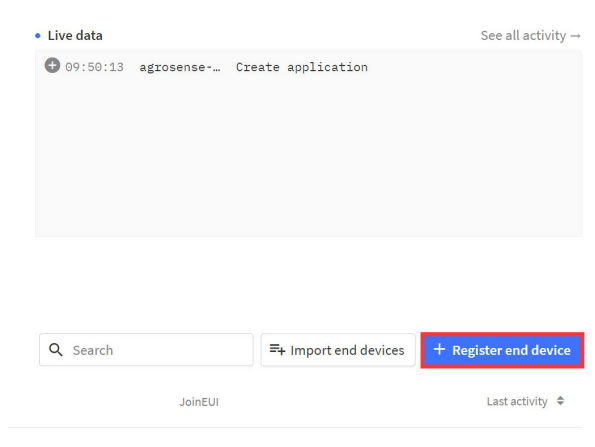
Application name
My new application

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.

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

Registration key

- Plug the battery and press 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 <>

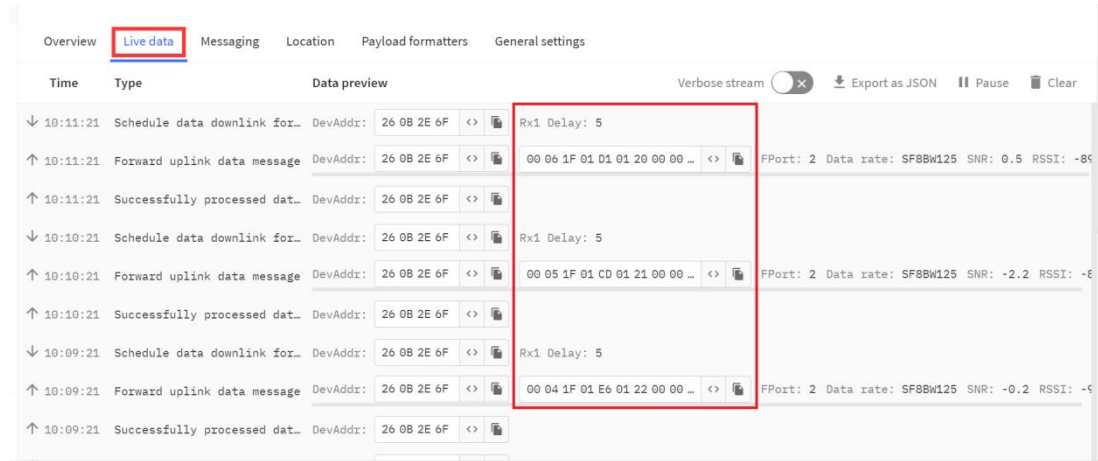
10:07:21 Schedule data downlink for transmission on Gateway Server De

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

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

3.2 Decoder

- Now, we need to decoder 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](#))

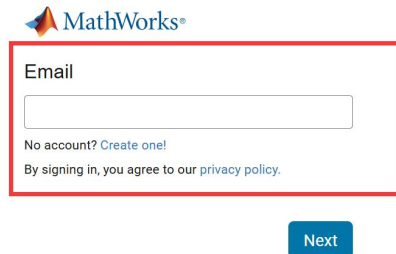
```
function decodeUplink(input) {
  var num = input.bytes[0] * 256 + input.bytes[1]
  var bat = input.bytes[2] / 10.0
  var humi = (input.bytes[3] * 256 + input.bytes[4]) / 10.0
  // var temp = (input.bytes[5] * 256 + input.bytes[6]) / 10.0
  var temp = input.bytes[5] * 256 + input.bytes[6]
  if (temp >= 0x8000) {
    temp -= 0x10000;
  }
  temp = temp / 10.0
  return {
    data: {
      field1: num,
      field2: bat,
      field3: humi,
      field4: temp,
    },
  };
}
```

- Select “Payload formatters” and follow the steps.

3.3 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®

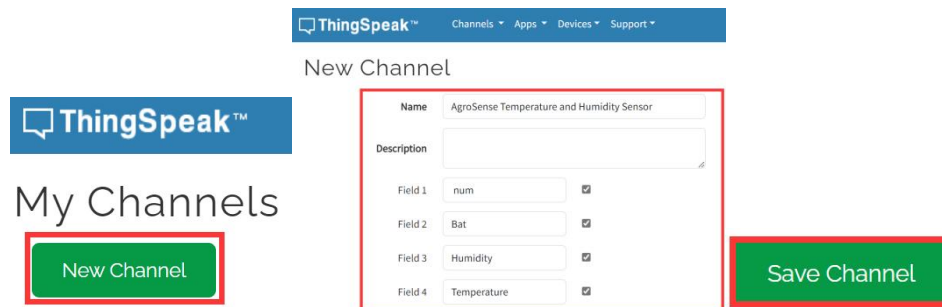
Email

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 ☒

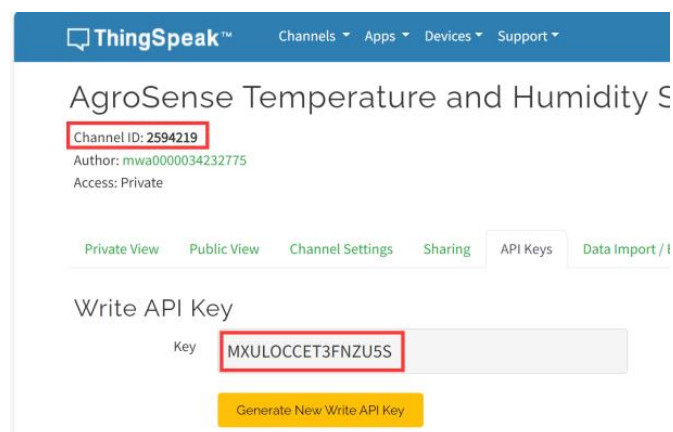
Field 2: Bat ☒

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

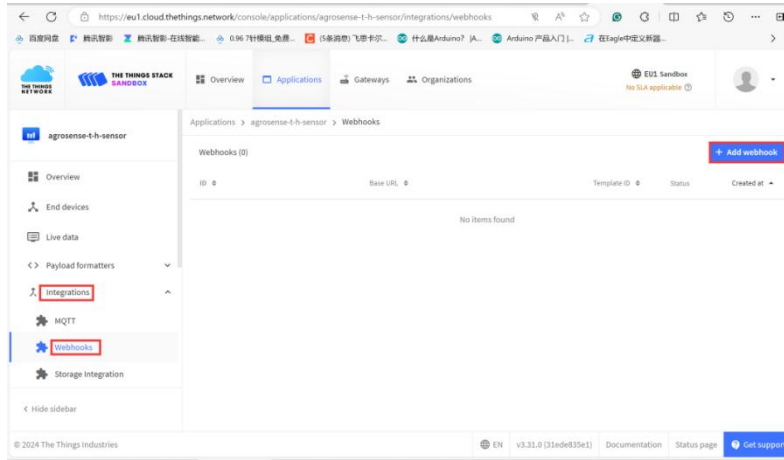
Write API Key

Key: MXULOCCE3FNZU5S

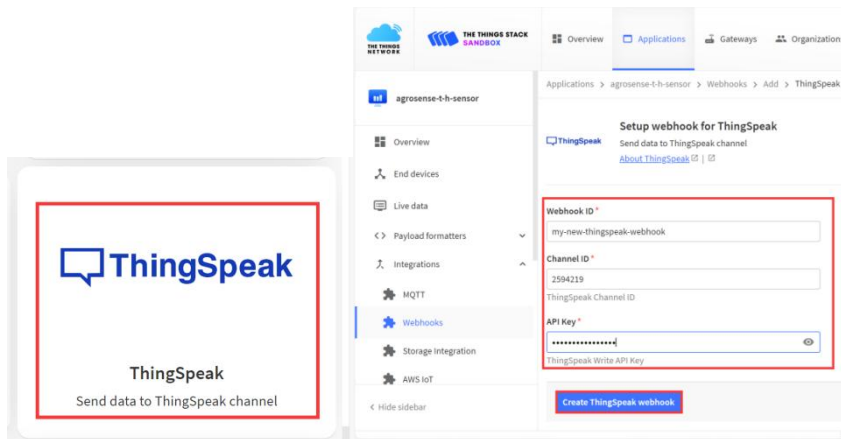
Generate New Write API Key

3.4 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”.



- Wait about a minute, you will successfully see the data in ThingSpeak.

