



# **AgroSense\_Soil & Air EC-TH Pro**

## **LoRaWAN® Manual**

### **V1.0**

Author: Yuki

Time: 2025.10.23

# Contents

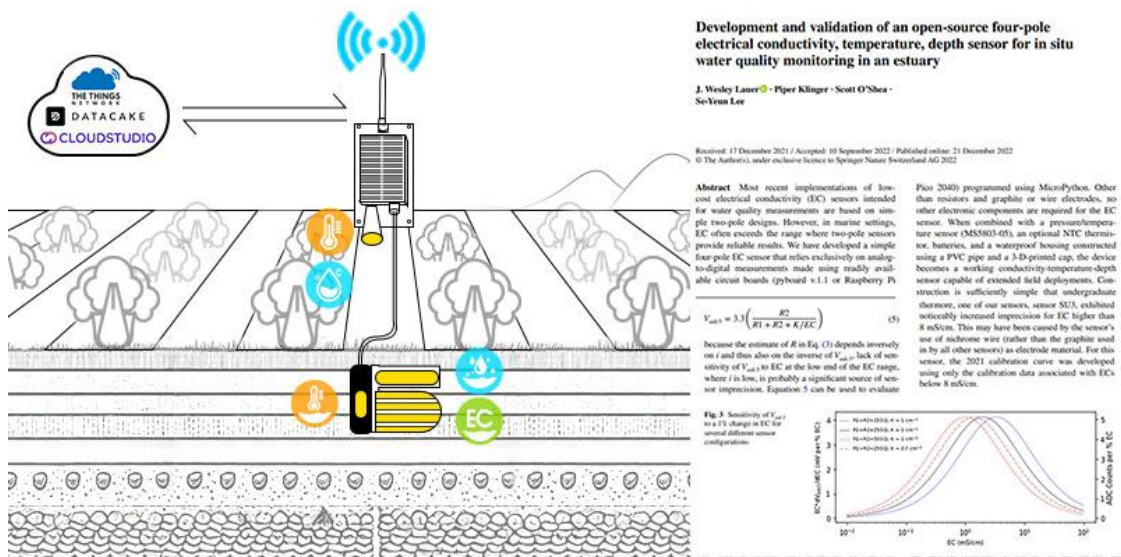
1 Product Description .....	1
1.1 Introduction .....	1
1.2 Feature .....	3
1.3 Parameter .....	4
2 Technical route .....	5
2.1 System Framework .....	5
2.2 Regional frequency band .....	6
3 Usage .....	7
3.1 Usage with TTN &ThingSpeak .....	7
3.2 Usage with Datacake .....	16

# 1 Product Description

## 1.1 Introduction

In modern agricultural production, monitoring soil and air conditions is crucial. Soil moisture directly affects irrigation efficiency, while soil electrical conductivity (EC) reflects salinity and fertility levels that are essential for crop growth. At the same time, air temperature and humidity are key factors influencing plant health and yield. The AgroSense\_Soil & Air EC-TH Pro LoRaWAN® provides an all-in-one solution, enabling farmers and researchers to track these critical parameters accurately and remotely for smarter agricultural management.

The AgroSense Soil & Air EC-TH Pro sensor combines soil Temperature / Humidity / EC Sensor with Air Temperature & Humidity Sensor, measures soil humidity at the range of 12-bit ADC, -40°C to 80°C, temperature and humidity in the atmosphere at the range of -40°C to 85°C and 0 to 100 %RH .



Soil EC Sensor adopts a four-electrode (four-pole) measurement design, measures at the range of 0-20000µS/cm, with 1µS/cm resolution which applies current through two outer electrodes and senses voltage via two inner electrodes. Compared with traditional two-pole probes, this method is far less affected by electrode polarization or fouling, ensuring higher accuracy and stability in long-term field deployments. Reference Environ Monit Assess (2023) [Development and validation of an open-source four-pole electrical conductivity, temperature, depth sensor](#) by J. Wesley Lauer from SEATTLE UNIVERSITY.

# Development and validation of an open-source four-pole electrical conductivity, temperature, depth sensor for in situ water quality monitoring in an estuary

J. Wesley Lauer  · Piper Klinger · Scott O'Shea · Se-Yeun Lee

Received: 17 December 2021 / Accepted: 10 September 2022 / Published online: 21 December 2022  
© The Author(s), under exclusive licence to Springer Nature Switzerland AG 2022

**Abstract** Most recent implementations of low-cost electrical conductivity (EC) sensors intended for water quality measurements are based on simple two-pole designs. However, in marine settings, EC often exceeds the range where two-pole sensors provide reliable results. We have developed a simple four-pole EC sensor that relies exclusively on analog-to-digital measurements made using readily available circuit boards (pyboard v.1.1 or Raspberry Pi

Pico 2040) programmed using MicroPython. Other than resistors and graphite or wire electrodes, no other electronic components are required for the EC sensor. When combined with a pressure/temperature sensor (MS5803-05), an optional NTC thermistor, batteries, and a waterproof housing constructed using a PVC pipe and a 3-D-printed cap, the device becomes a working conductivity-temperature-depth sensor capable of extended field deployments. Construction is sufficiently simple that undergraduate science students can construct one during three 3-h lab periods. Lab calibrations performed on several prototypes at ECs between 0.18 and 45 mS/cm show that confidence limits as good as about  $\pm 3\%$  of EC are possible. Re-calibration of several prototypes 1 year after initial calibration shows that long-term calibration drift is modest. Data collected by the pro-

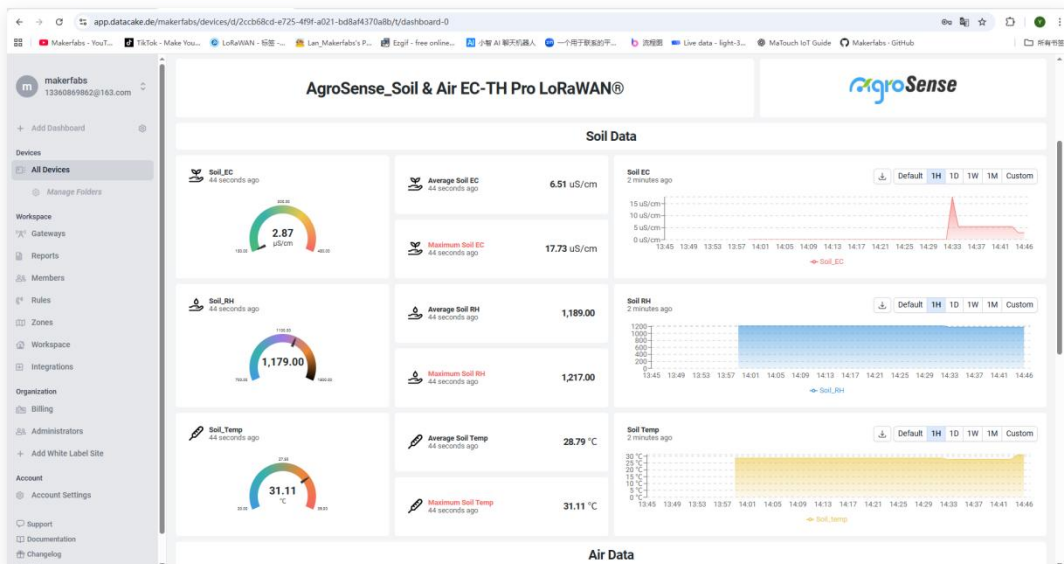
J. W. Lauer (✉)

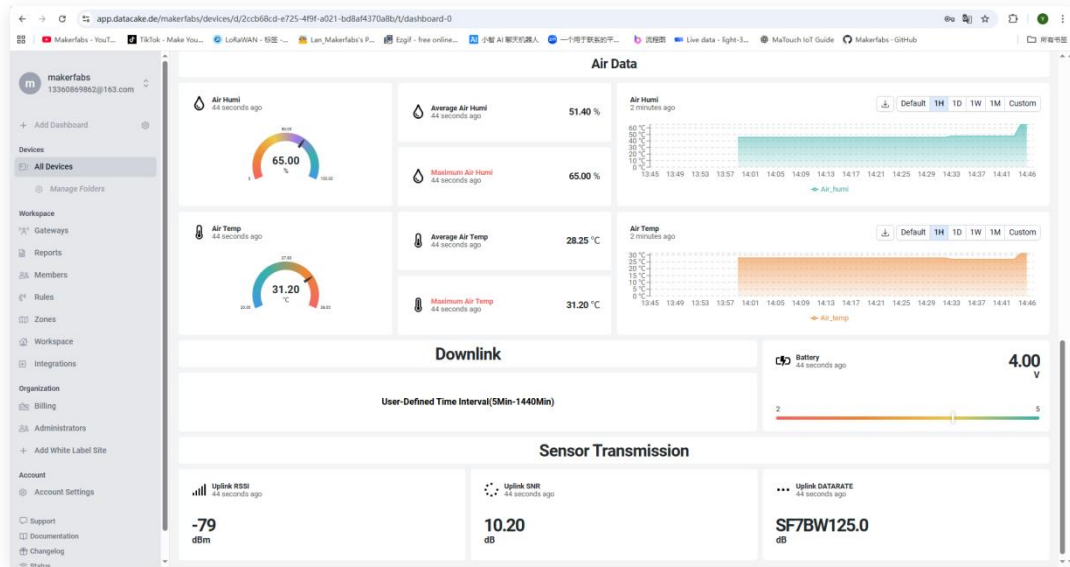
Department of Civil & Environmental Engineering, Seattle University, 901 12th Avenue, Seattle, WA 98122, USA  
e-mail: lauerj@seattleu.edu

P. Klinger · S.-Y. Lee

Department of Civil & Environmental Engineering, Seattle University, Seattle, WA 98122, USA

This product is based on LoRaWAN. We have cooperated with cloud service vendors such as ThingSpeak and Datacake, allowing users to remotely monitor and control the device via the cloud.





Also, this product cased with IP67 case, solar panel powered, can be used long-term in filed application.

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 by dual lithium batteries and a solar panel, the sensor achieves ultra-low power consumption, enabling more than 10 years of continuous operation, 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.

## 1.2 Feature

- Includes a **high precision** sensor.
- LoRaWAN version: LoRaWAN Specification 1.0.3. OTAA **Class A**.
- Monitor data and upload **real-time** data regularly.
- Modify the product parameters through **AT commands**.
- Support **downlink** to modify the time interval.
- 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	AGLWSAEC
Air Temperature Measurement Range	-40°C ~85°C
Air Temperature Measurement Accuracy	±0.3°C
Air Temperature Resolution	0.01°C
Air Humidity Measurement Range	0%-100% RH
Air Humidity Measurement Accuracy	±2%
Air Humidity Resolution	0.024% RH
Soil Temperature Measurement Range	-40°C ~80°C
Soil Temperature Measurement Accuracy	±0.5°C
Soil Temperature Resolution	0.1°C
Soil Humidity Measurement Range	0%-100% RH
Soil Humidity Measurement Accuracy	±2%
Soil Humidity Resolution	0.1% RH
EC Measurement Range	0-20000µS/cm
EC Resolution	1µS/cm

### 2.Wireless Parameters

Communication Protocol	Standard LoRaWAN® protocol V1.0.3
Network Access/Operating Mode	OTAA <b>Class A</b>
MAX Transmit Power	22dBm
Receiver Sensitivity	-137dBm/125kHz SF=12
Frequency Band	EU868/US915

### 3.Physical Parameters

Batteries Power Supply	2 x 18650 3.7V Lion batteries
Solar Power Supply	5V2W
Operating Temperature	-40°C ~85°C
Protection Class	IP67
Dimensions	115 × 85 × 35 mm
Mounting	Wall Mounting

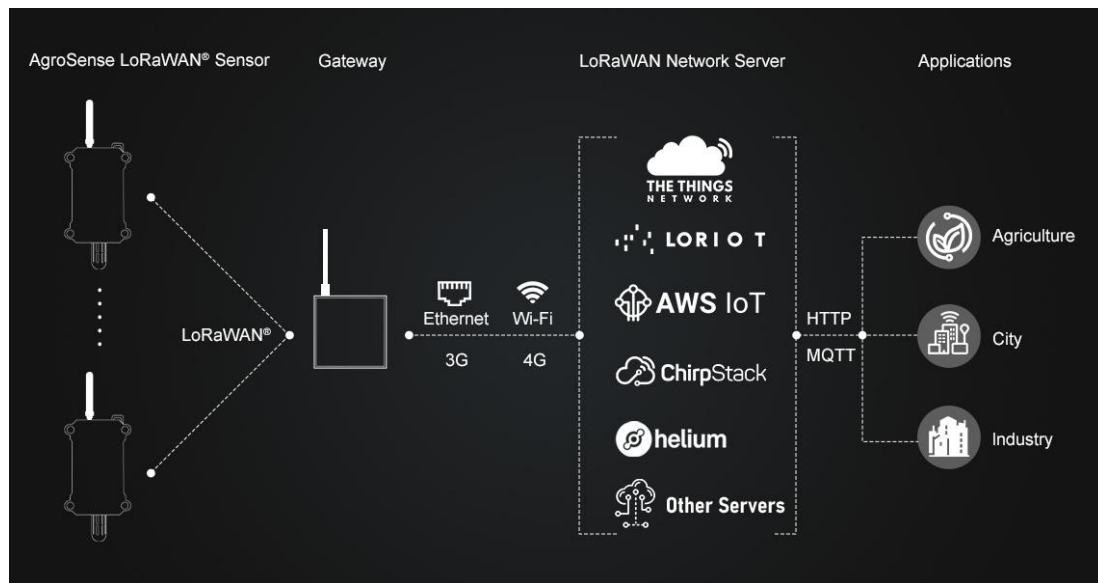


## 2 Technical route

### 2.1 System Framework

AgroSense\_Soil & Air EC-TH Pro 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 minutes; Control Valve ON/OFF.)

#### 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., activate valve, 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

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

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

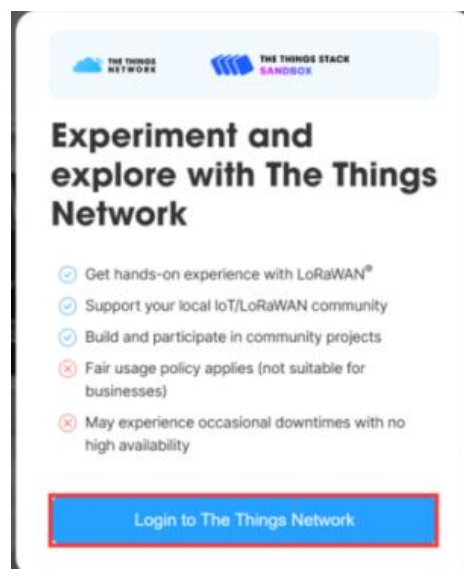
DEV EUI	Unique identification of device, authorized by IEEE
APP EUI	Unique identification of application
APP Key	One of the join network parameters on OTAA mode, calculated by DE EUI

- End Nodes and Gateway:AgroSense\_Soil & Air EC-TH Pro(The AgroSense series is applicable)
- Network Server: The Things Network. ( Datacake, Loriot, AWS IoT, ChirpStack, ect)
- Application Server: ThingSpeak.(Datacake, Blockbax, akenza, ect)

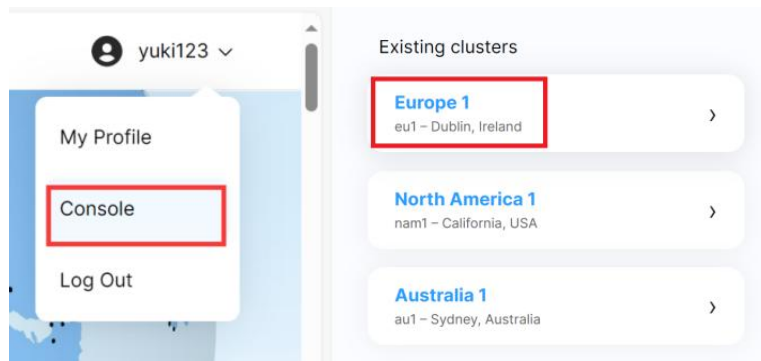
#### 3.1 Usage with TTN &ThingSpeak

##### 3.1.1 Network Server configuration

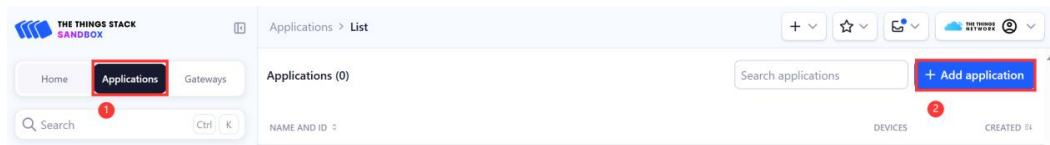
- Open The Things Network in your browser and login it. (Or register an account)



- 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

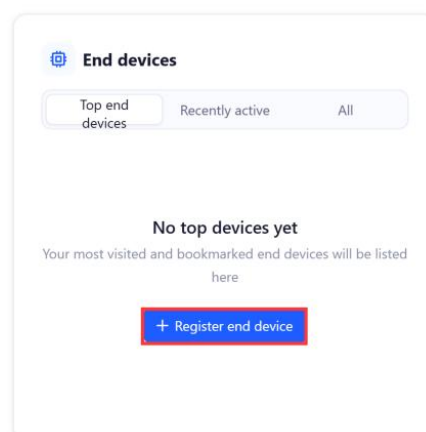
Description

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

Frequency plan

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

LoRaWAN version

LoRaWAN Specification 1.0.3

Regional Parameters version

RP001 Regional Parameters 1.0.3 revision A

Show advanced activation, LoRaWAN class and cluster settings

**Provisioning information**

JoinEUI

48 FF 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

**Provisioning information**

JoinEUI

48 FF 00 00 00 01 65

Confirm

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

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

Device overview

Live data

Messaging

Location

Payload formatters

Settings

TIME TYPE DATA PREVIEW

Verbose stream

Export as JSON

Pause

Clear

10:19:32 Schedule data download for... DevAddr: 26 0B EA 1A Rx1 Delay: 5

10:19:31 Update end device [ 'activated\_at' ]

10:19:31 Forward uplink data message DevAddr: 26 0B EA 1A 00 01 06 34 1D 00 36 EE 00 00 FPort: 2 Data rate: SF12BW125 SNR: 7

10:19:31 Successfully processed dat... DevAddr: 26 0B EA 1A

10:19:23 Forward join-accept message DevAddr: 26 0B EA 1A JoinEUI: 48 FF 00 00 00 02 00 DevEUI: 48 E6 63 FF FE 30 02 00

10:19:21 Successfully processed joi... JoinEUI: 48 FF 00 00 00 02 00 DevEUI: 48 E6 63 FF FE 30 02 00

10:19:21 Accept join-request DevAddr: 26 0B EA 1A JoinEUI: 48 FF 00 00 00 02 00 DevEUI: 48 E6 63 FF FE 30 02 00

10:19:07 Create end device

### 3.1.2 Decoder

- Now, we need to decode the data.

Overview

Live data

Messaging

Location

Payload formatters

General settings

Time Type Data preview

Verbose stream

Export as JSON

Pause

Clear

09:52:56 Schedule data download for tr... DevAddr: 26 0C DA 7C Rx1 Delay: 5

09:52:56 Fail to send webhook Request

09:52:55 Forward uplink data message DevAddr: 26 0C DA 7C 00 01 06 31 1C 00 00 00 00 FPort: 2 Data rate: SF10BW125 SNR: 12.5 RSSI: -80

09:52:55 Successfully processed data m... DevAddr: 26 0C DA 7C

09:52:49 Forward join-accept message DevAddr: 26 0C DA 7C JoinEUI: 48 FF 00 00 00 00 16 DevEUI: 48 E6 63 FF FE 30 00 16

09:52:48 Successfully processed join-r... DevAddr: 26 0C 28 1F JoinEUI: 48 FF 00 00 00 00 16 DevEUI: 48 E6 63 FF FE 30 00 16

09:52:48 Accept join-request DevAddr: 26 0C DA 7C JoinEUI: 48 FF 00 00 00 00 16 DevEUI: 48 E6 63 FF FE 30 00 16

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	F	

### AgroSense\_Soil & Air EC-TH Pro LoRaWAN®

	sequence number low 8 bits		
byte 2	Battery voltage		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	Soil temp bits 8 to 15		The value is amplified by a factor of 100. To get the actual value, divide it by 100. For example, if the value is 0x08FC = 2300, then the value is 23.
byte 4	Soil temp bits 0 to 7		
byte 5	Soil RH bits 8 to 15		Soil RH ADC value, For example, if the value is 0x0379 = 889, then the value is 889.
byte 6	Soil RH bits 0 to 7		
byte 7	Soil EC bits 24 to 31		The value is amplified by a factor of 100. To get the actual value, divide it by 100. For example, if the value is 0x000008FC = 2300, then the value is 23.
byte 8	Soil EC bits 16 to 23		
byte 9	Soil EC bits 8 to 15		
byte 10	Soil EC bits 0 to 7		
byte 11	Air temp 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 0x0113 (275), the actual value is 27.5
byte 12	Air temp bits 0 to 7		
byte 13	Air Humi 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 0x01D2 (466), the actual value is 46.6
byte 14	Air Humi bits 0 to 7		
byte 15	data transmission interval bits 24 to 31		The time interval for data transmission has been increased by a factor of 1000.  The unit is seconds.
byte 16	data transmission interval bits 16 to 23		
byte 17	data transmission interval bits 8 to 15		
byte 18	data transmission interval bits 0 to 7		
Downlink			
Fport 1	Change the data sending interval		5min-1440min

Example: 0x00 0x01 0x24 0x08 0xFC 0x03 0x79 0x00 0x00 0x08 0xFC 0x01 0x13 0x01 0xD2 0x00 0x36 0xEE 0x80

Data parsing:

Battery voltage is 3.6V.

Soil temp is 23°C.

Soil RH is 889.

Soil EC is 23 μ S/cm.

Air temp is 27.5°C.

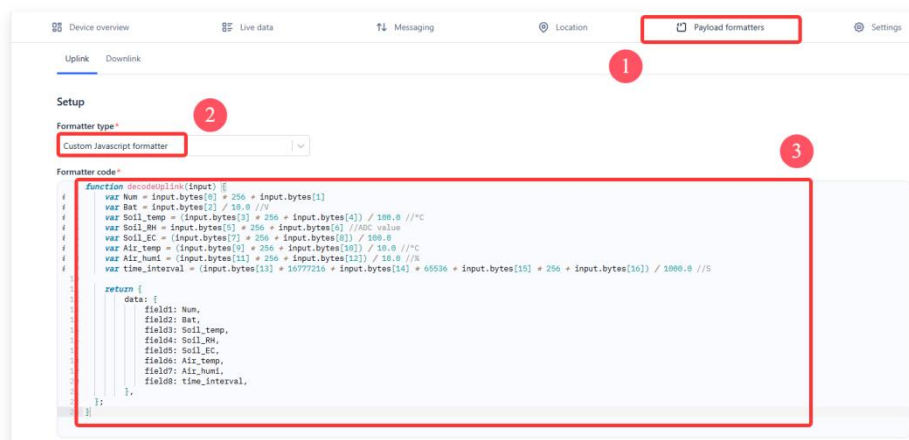
Air Humi is 46.6%.

Data transmission interval value is 3600s.

- 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 //V
    var Soil_temp = (input.bytes[3] * 256 + input.bytes[4]) / 100.0 //°C
    var Soil_RH = input.bytes[5] * 256 + input.bytes[6] //ADC value
    // 1270 corresponds to the ADC value in air, and 815 corresponds to the ADC value in water.
    // Based on this, the ADC can be converted into a percentage.
    // Since water quality varies from place to place, customers need to modify these values themselves.
    var Soil_RH_Percentage=(1270-Soil_RH)*100/(1270-815) // %
    var Soil_EC = (input.bytes[7] * 16777216 + input.bytes[8] * 65536 + input.bytes[9] * 256 + input.bytes[10]) /
    100.0 //µS/cm
    var Air_temp = (input.bytes[11] * 256 + input.bytes[12]) / 10.0 //°C
    var Air_humi = (input.bytes[13] * 256 + input.bytes[14]) / 10.0 // %
    var interval = (input.bytes[15] * 16777216 + input.bytes[16] * 65536 + input.bytes[17] * 256 +
    input.bytes[18]) / 1000.0 //S
    return {
        data: {
            //field1: Num,
            field1: Bat,
            field2: Soil_temp,
            //field3: Soil_RH,
            field3: Soil_RH_Percentage,
            field4: Soil_EC,
            field5: Air_temp,
            field6: Air_humi,
            field7: time_interval,
        },
    },
};
}
```

- Select “Payload formatters” and follow the steps.



Save changes

### 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)



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”.



My Channels

New Channel

Name	AgroSense_Soil Monitor (Humidity/Temperature/EC)	
Description		
Field 1	Num	<input checked="" type="checkbox"/>
Field 2	Bat	<input checked="" type="checkbox"/>
Field 3	Soil temp	<input checked="" type="checkbox"/>
Field 4	Soil_RH	<input checked="" type="checkbox"/>
Field 5	Soil_EC	<input checked="" type="checkbox"/>
Field 6	Air_temp	<input checked="" type="checkbox"/>
Field 7	Air_humi	<input checked="" type="checkbox"/>
Field 8	time_interval	<input checked="" type="checkbox"/>

Save Channel

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

Channel ID: 2599652

Author: mwa000034232775

Access: Private

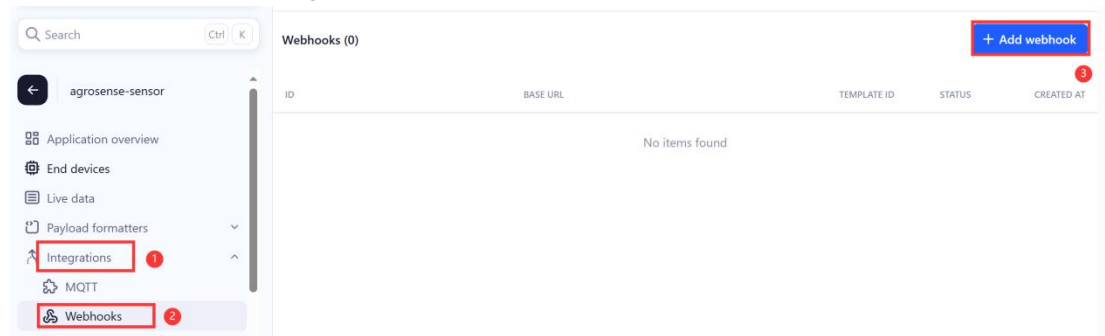
Private View Public View Channel Settings Sharing API Keys

Write API Key

Key N9IBFTBI3J36T779

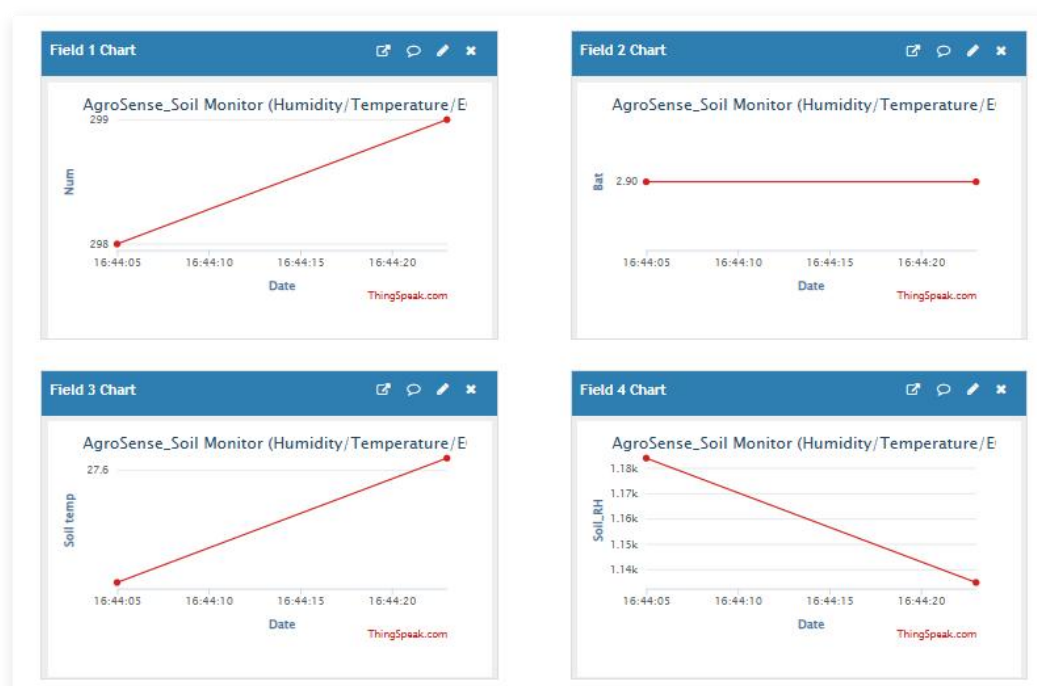
### 3.1.4 Connect the Network Server and Application Server

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

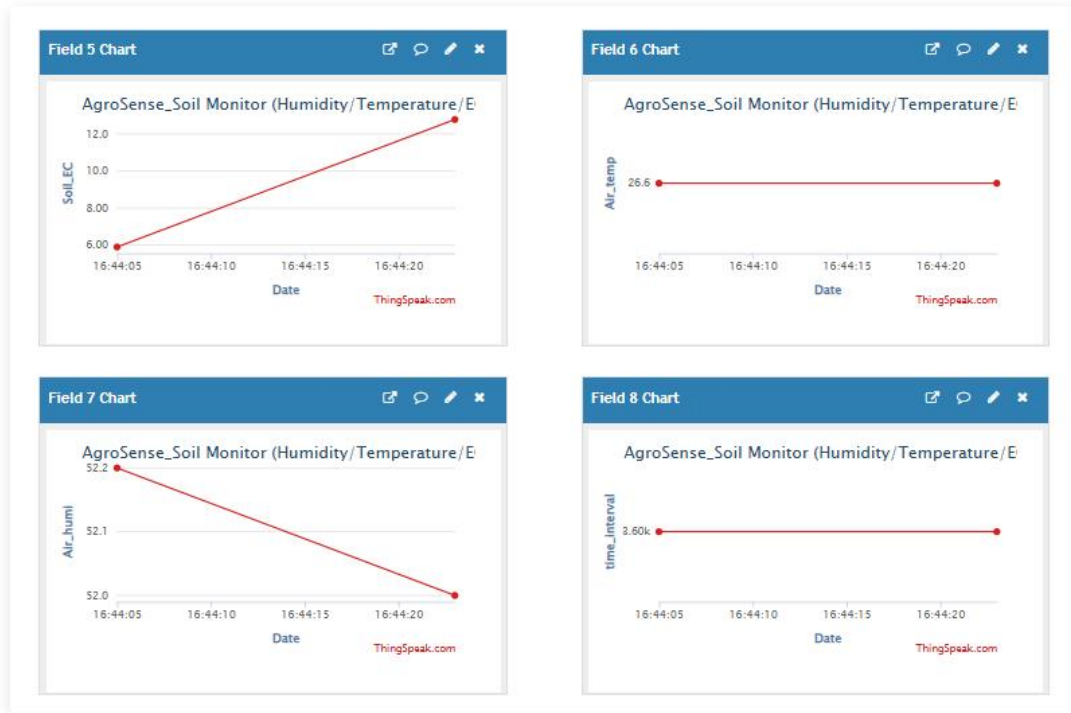


- Select “ThingSpeak”, Fill in the Webhook ID and paste the Channel ID and API Key, click “Create ThingSpeak Webhook”.

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







### 3.1.5 Downlink

- Modification time interval (Fport1)

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

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](#).

Device overview Live data Messaging Location Payload formatters

Uplink: Downlink

Setup

Formatter type\*: Custom Javascript formatter

Formatter code\*

```

1 // Encoder function to be used in the TTN console for downlink payload
2 function Encoder(input) {
3   var minutes = input.minutes;
4   // Converting minutes to seconds
5   var seconds = minutes * 60;
6   // If the number of seconds is less than 300 seconds, set it to 300 seconds
7   if (seconds < 300) {
8     seconds = 300;
9   }
10   var payload = [
11     (seconds >> 24) & 0xFF,
12     (seconds >> 16) & 0xFF,
13     (seconds >> 8) & 0xFF,
14     seconds & 0xFF
15   ];
16   return payload;
17 }

```

2、Click “Save changes”.

Save changes

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

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

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

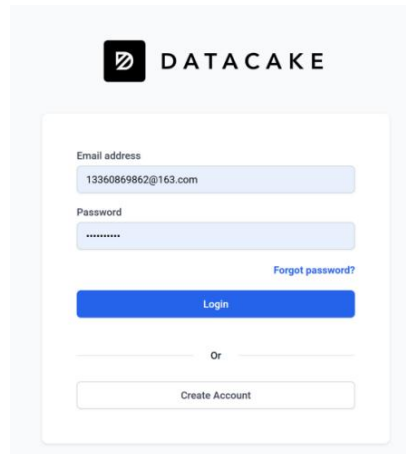
The screenshot shows the 'Messaging' tab in the LoRaWAN interface. Under the 'Schedule downlink' section, the 'Insert Mode' is set to 'Replace downlink queue'. The 'FPort' is set to 1. The 'Payload type' is set to 'JSON'. The 'Payload' field contains the JSON object: {"minutes": 5}. At the bottom, there is a 'Schedule downlink' button.

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

## 3.2 Usage with Datacake

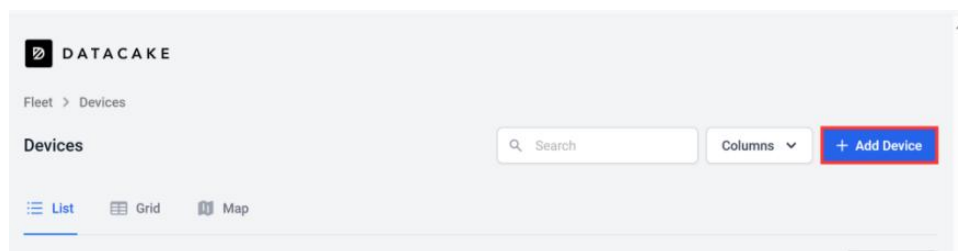
In this phase, we use DataCake(<https://datacake.co/>) as the data server & console.

### 1、Login datacake or Create Account

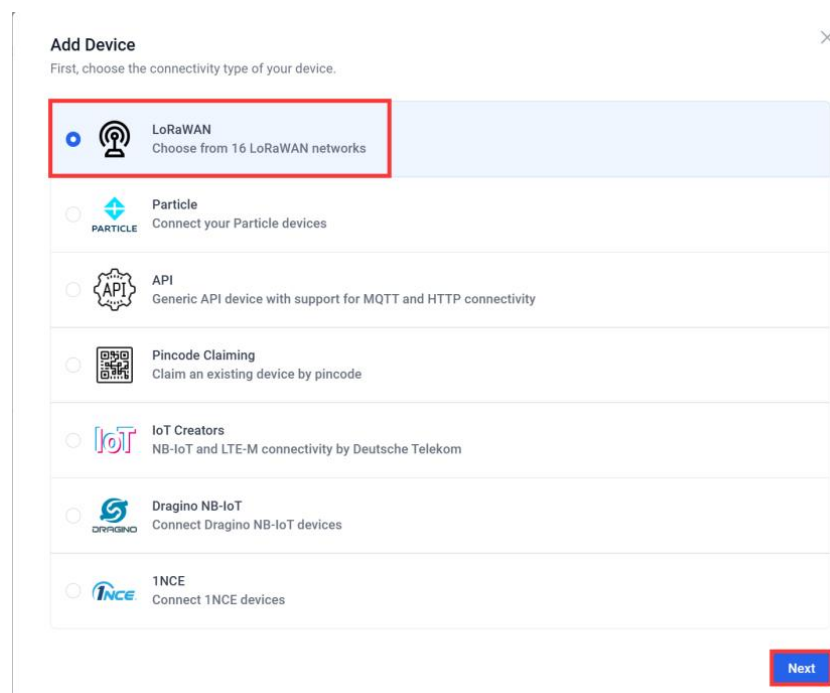


The image shows the DataCake login and registration interface. At the top is the DataCake logo. Below it is a form with two input fields: 'Email address' (containing '13360869862@163.com') and 'Password' (containing '\*\*\*\*\*'). There is a 'Forgot password?' link next to the password field. Below the password field is a blue 'Login' button. Underneath the login button 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'. At the bottom right of the dialog is a red button labeled 'Next'.

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

Downlinks

☐

**The Things Stack V3**

TTN V3 / Things Industries

Uplinks

Downlinks

☐

**Helium**

Use your own console

Uplinks

Downlinks

☐

**LORIoT**

Uplinks

Downlinks

☐

**ChirpStack**

Uplinks

Downlinks

☐

**Activity**

Uplinks

Downlinks

☐

**KPN**

Uplinks

Downlinks

Showing 1 to 6 of 15 results

Previous

Next

Back

Next

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

The image shows two screenshots of the 'Add LoRaWAN Device' form. The left screenshot shows the 'DEVEUI' field with the value '48 15 63 FF FE 30 00 1E' and the 'NAME' field with 'agrosense sensor'. The right screenshot shows the 'APPKEY' field with 'F3 F0 45 8F 9F 21 C7 C8 54 89 7A A8 C5 34 CC A9', the 'FREQUENCY' field with 'United States 902-928 MHz, FSB 2', and the 'DEVICE CLASS' field with 'Class C'.

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

The image shows the 'Add LoRaWAN Device' form, specifically the 'STEP 3: Devices' section. The 'Free' plan is selected, which costs €0.00 / month and offers 7 days data retention, 500 datapoints / day, and max 5 per workspace. The 'Add 1 device' button is highlighted.

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

**DATA CAKE**

Fleet > Devices

Devices

Search Columns + Add Device

List Grid Map

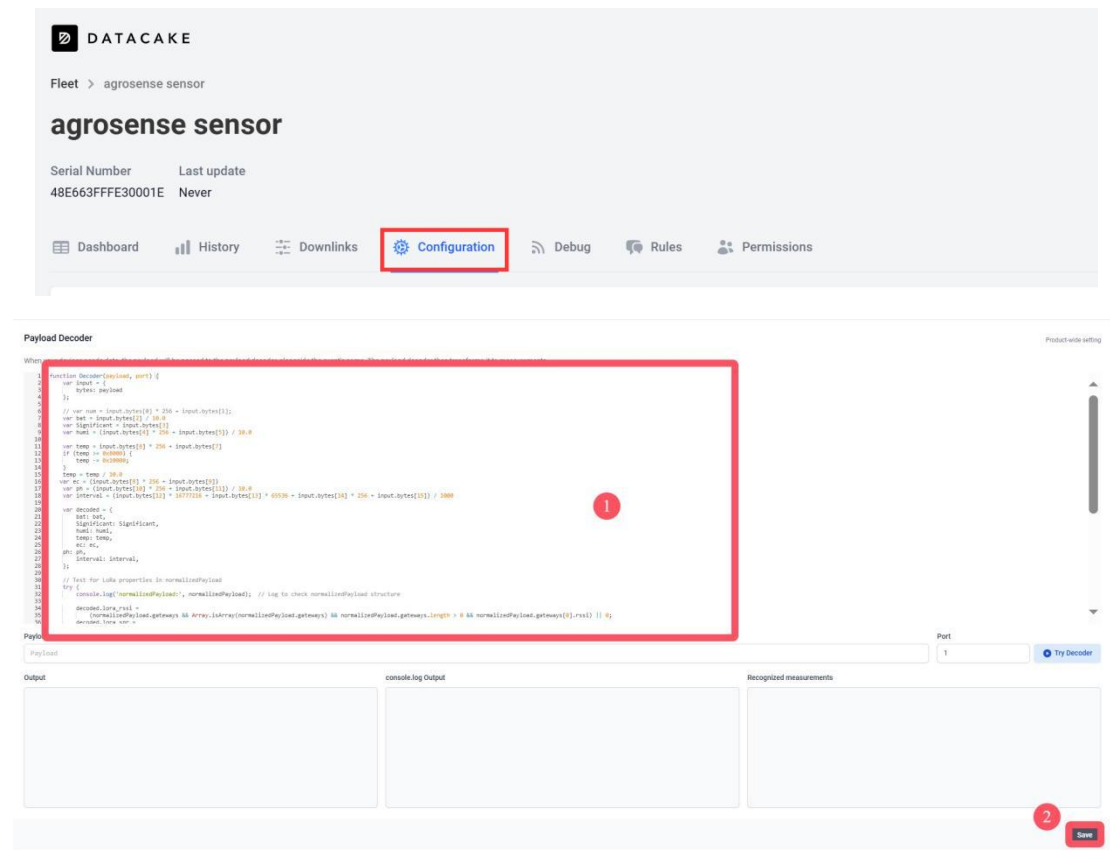
Actions

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	

Showing 1 to 3 of 3 results

50 per page Previous Next

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 //V
    var Soil_temp = (input.bytes[3] * 256 + input.bytes[4]) / 100.0 //° C
    var Soil_RH = input.bytes[5] * 256 + input.bytes[6] //ADC value
    // 1270 corresponds to the ADC value in air, and 815 corresponds to the ADC value in water.
    // Based on this, the ADC can be converted into a percentage.
    // Since water quality varies from place to place, customers need to modify these values themselves.
    var Soil_RH_Percentage=(1270-Soil_RH)*100/(1270-815) //%
    var Soil_EC = (input.bytes[7] * 16777216 + input.bytes[8] * 65536 + input.bytes[9] * 256 + input.bytes[10]) /
    100.0 //μS/cm
    var Air_temp = (input.bytes[11] * 256 + input.bytes[12]) / 10.0 //° C
    var Air_humi = (input.bytes[13] * 256 + input.bytes[14]) / 10.0 //%
    var interval = (input.bytes[15] * 16777216 + input.bytes[16] * 65536 + input.bytes[17] * 256 +
    input.bytes[18]) / 1000.0 //S

    var decoded =
    {
        //NUM:Num
    }
}
```

```

    BAT:Bat
    SOIL_TEMP:Soil_temp
    //SOIL_RH:Soil_RH
    SOIL_RH_PERCENTAGE:Soil_RH_Percentage
    SOIL_EC:Soil_EC
    AIR_TEMP:Air_temp
    AIR_HUMI:Air_humi
    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 decoded;
}

```

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

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

Assign roles to highlight key measurement fields across the platform; each role is limited to one field per product.

Semantic

None

Assign semantic fields to standardize data across devices.

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

NAME	IDENTIFIER	TYPE
Bat	BAT	Float
Soil_temp	SOIL_TEMP	Float
Soil_RH	SOIL_RH	Float
Soil_EC	SOIL_EC	Float
Air_temp	AIR_TEMP	Float
Air_humi	AIR_HUMI	Float
Interval	INTERVAL	Float
Lora Rssi	LORA_RSSI	Integer
Lora Datarate	LORA_DATARATE	String
Lora Snr	LORA_SNR	Float

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

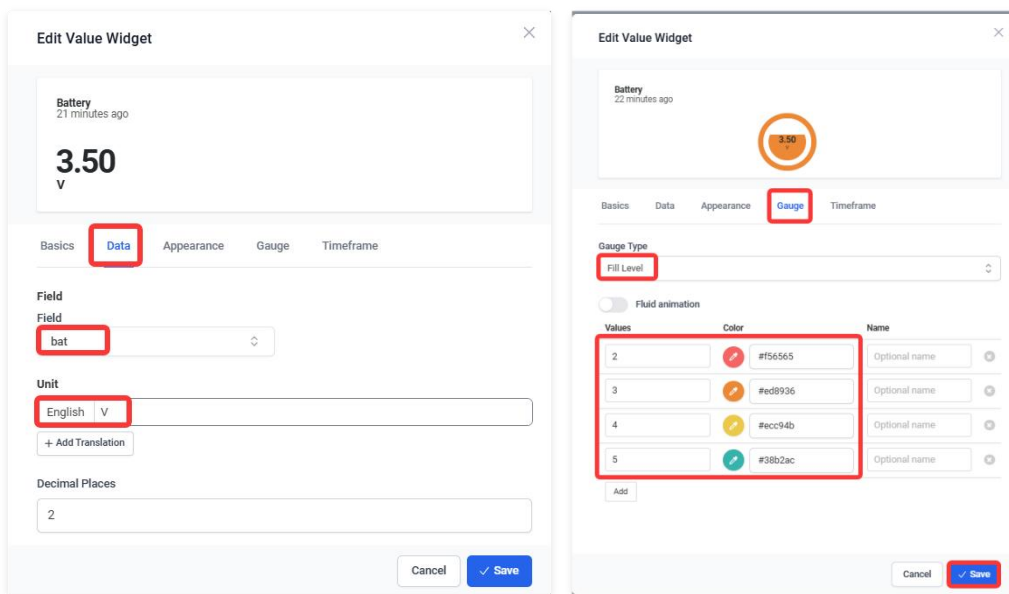
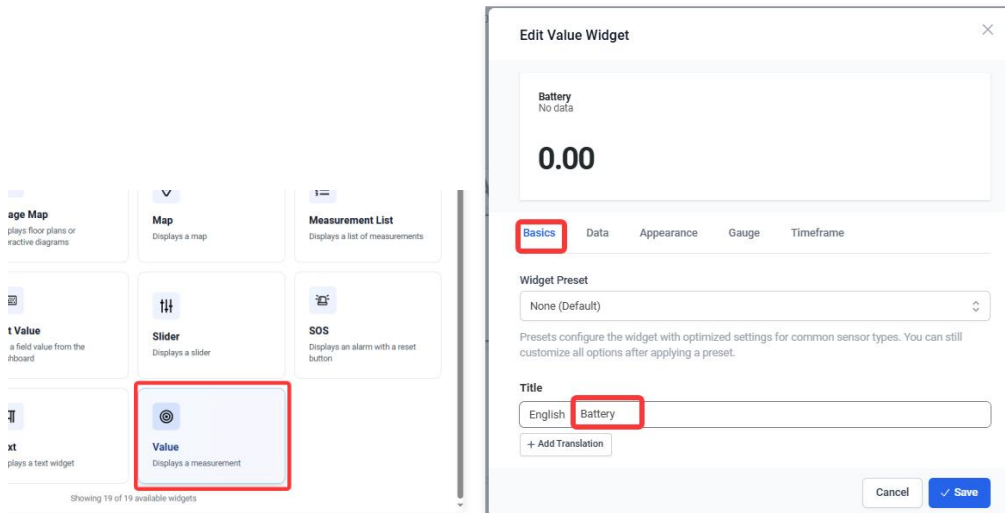
NAME	IDENTIFIER	TYPE	ROLE / SEMANTIC	CURRENT VALUE	LAST UPDATE
Bat	BAT	Float	Battery	2.9	0 seconds ago
Soil_temp	SOIL_TEMP	Float		27.83	0 seconds ago
Soil_RH	SOIL_RH	Float		1,051	0 seconds ago
Soil_EC	SOIL_EC	Float		185.55	0 seconds ago
Air_temp	AIR_TEMP	Float		26	0 seconds ago
Air_humi	AIR_HUMI	Float		63	0 seconds ago
Interval	INTERVAL	Float		3,600	0 seconds ago
Lora Rssi	LORA_RSSI	Integer		-87	0 seconds ago
Lora Datarate	LORA_DATARATE	String		SF7BW125.0	0 seconds ago
Lora Snr	LORA_SNR	Float		11.2	0 seconds ago

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

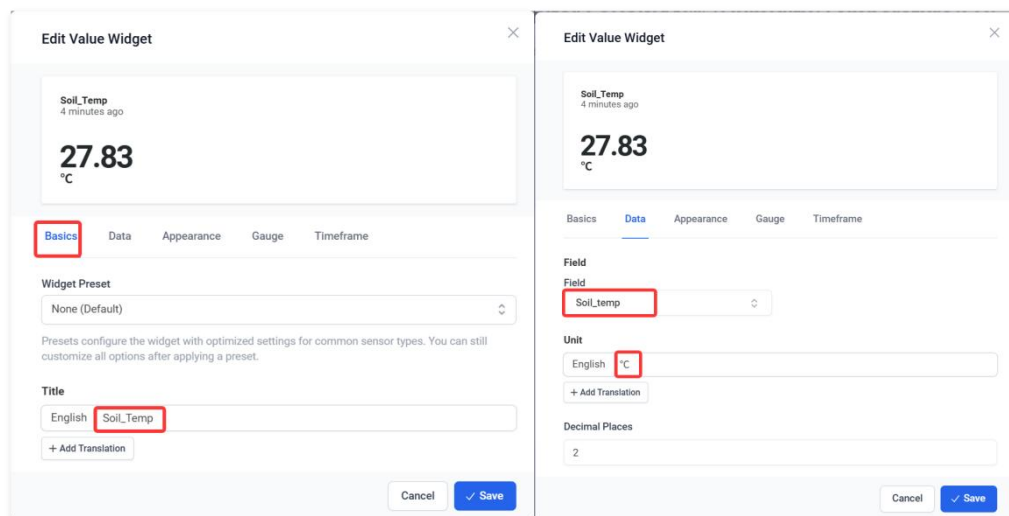
Click “Dashboard-->switch-->+ Add Widget”.



13、 Add “Battery” field, Select “Value” and follow the steps below to complete the setup.



14、 Add “Soil\_Temp” field, Select “Value” and follow the steps below to complete the setup.



**Edit Value Widget**

Soil\_Temp  
10 minutes ago

27.83  
°C

Basics Data Appearance **Gauge** Timeframe

Gauge Type  
Vertical

Values	Color	Name
20	#4299e1	Optional name
25	#38b2ac	Optional name
30	#ed8936	Optional name
35	#f56565	Optional name

Direction  
☒ Ascending ☐ Descending

+ Add Reference Line

Cancel Save

15、Add “Soil\_RH” field, Select “Value” and follow the steps below to complete the setup.

**Edit Value Widget**

Soil\_RH  
12 minutes ago

1,051.00

Basics Data Appearance Gauge Timeframe

Widget Preset  
None (Default)

Presets configure the widget with optimized settings for common sensor types. You can still customize all options after applying a preset.

Title  
English **Soil\_RH**

+ Add Translation

Cancel Save

**Edit Value Widget**

Soil\_RH  
12 minutes ago

1,051.00

Basics **Data** Appearance Gauge Timeframe

Field  
Field  
**Soil\_RH**

Unit  
English

+ Add Translation Sync Translations With Other Widgets

Decimal Places  
2

Cancel Save

**Edit Value Widget**

Soil\_RH  
14 minutes ago

1,051.00

Basics Data Appearance **Gauge** Timeframe

Gauge Type  
Circular

Values	Color	Name
1500	#000	Optional name
1300	#ed8936	Optional name
1100	#9f7aea	Optional name
900	#38b2ac	Optional name
700	#4299e1	Optional name

Add

Cancel Save

16、Add “Soil\_EC” field, Select “Value” and follow the steps below to complete the setup.

**Edit Value Widget**

Soil\_EC  
No data

0.00

Basics **Data** Appearance Gauge Timeframe

Widget Preset  
None (Default)

Presets configure the widget with optimized settings for common sensor types. You can still customize all options after applying a preset.

Title  
English Soil\_EC

+ Add Translation

Cancel Save

**Edit Value Widget**

Soil\_EC  
25 minutes ago

185.55  
µS/cm

Basics **Data** Appearance Gauge Timeframe

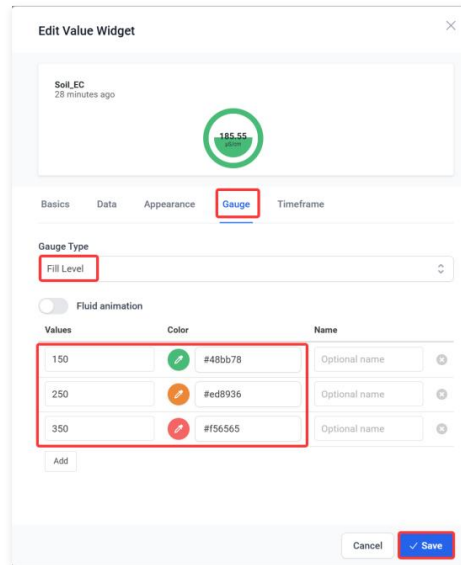
Field  
Field  
Soil\_EC

Unit  
English µS/cm

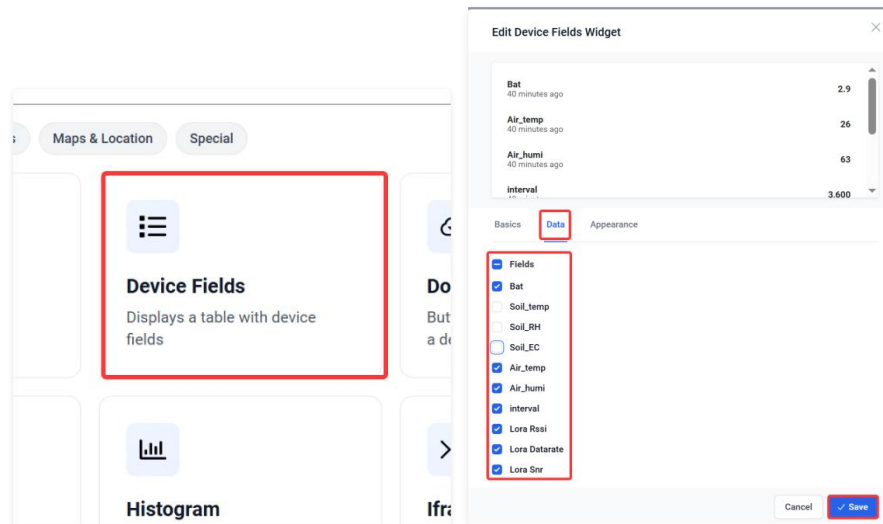
+ Add Translation

Decimal Places  
2

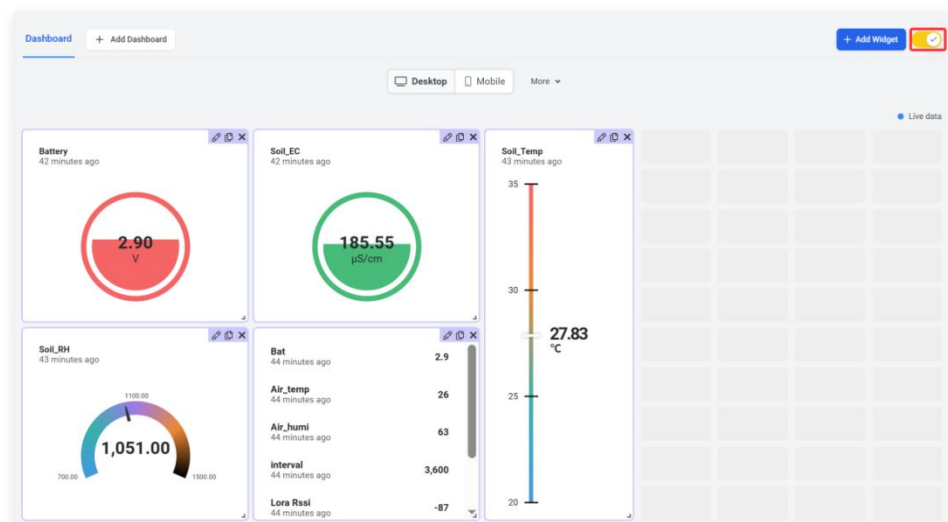
Cancel Save



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



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

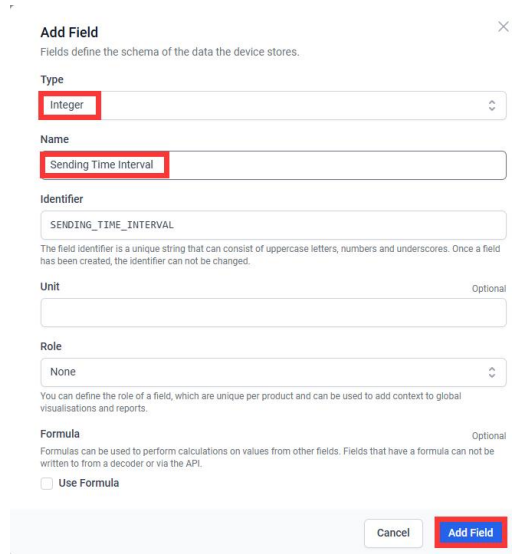


### 3.2.1 Downlink

- Modification time interval (Fport1)

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

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

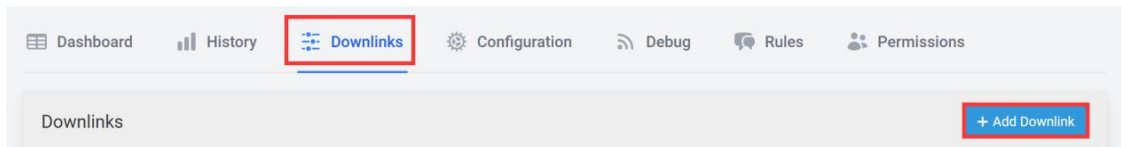
Formula: Optional

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”.



Dashboard History **Downlinks** Configuration Debug Rules Permissions

Downlinks + 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     }
22     else {
23       targetLength = targetLength-this.length;
24       if (targetLength > padString.length) {
25         padString += padString.repeat(targetLength/padString.length); //append to original to en
26       }
27     }
28   }
29 }

```

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: [dropdown]  
+ 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.

**Sending Time Interval**

1

Cancel | Save measurements and send downlink