

The Things Stack Using Thingspeak

The Things Stack Using Thingspeak

Contents

Objectives

Onboarding the Water Level Sensor using The Things Stack

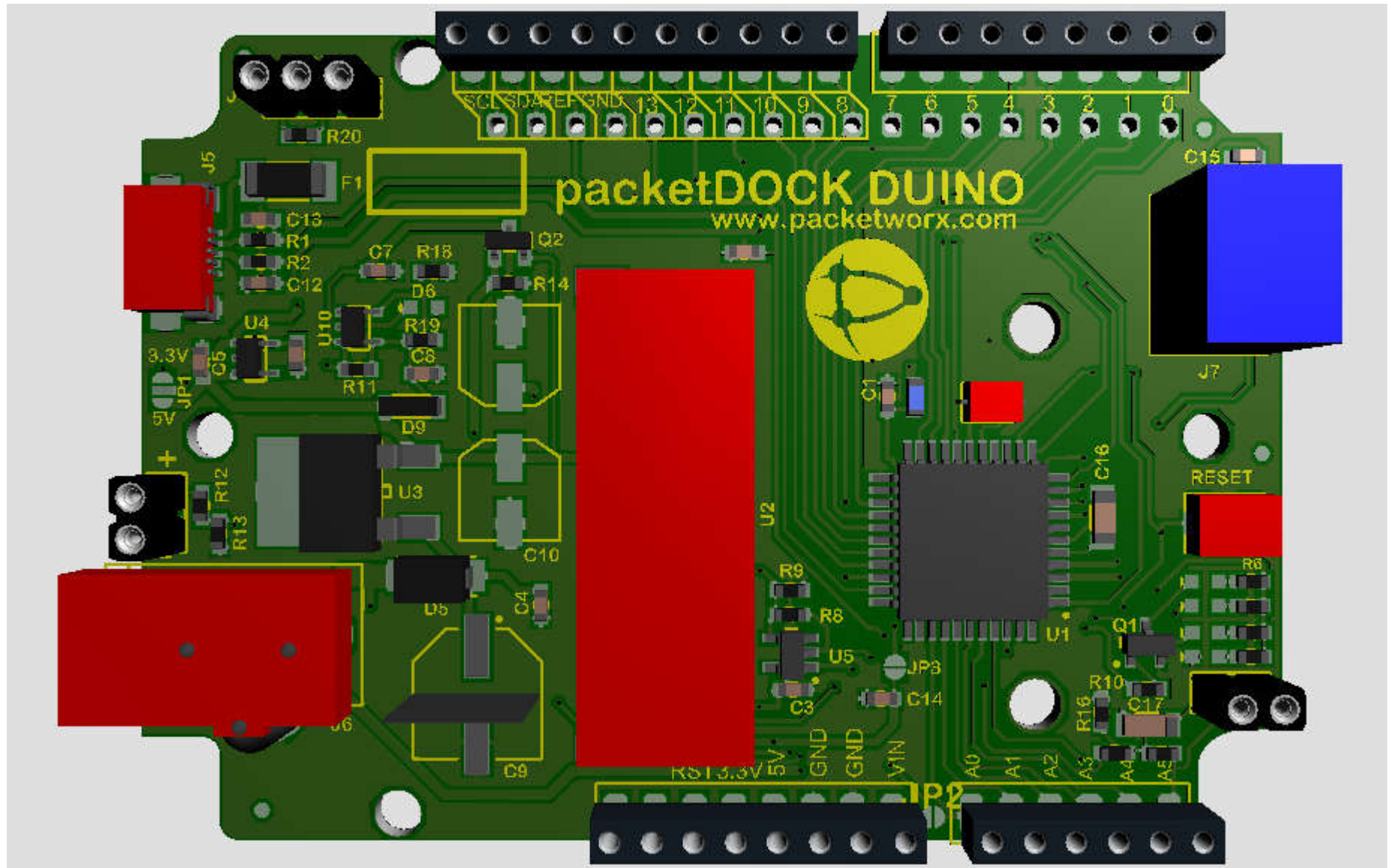
How to make Dashboard using Thingspeak

Objectives

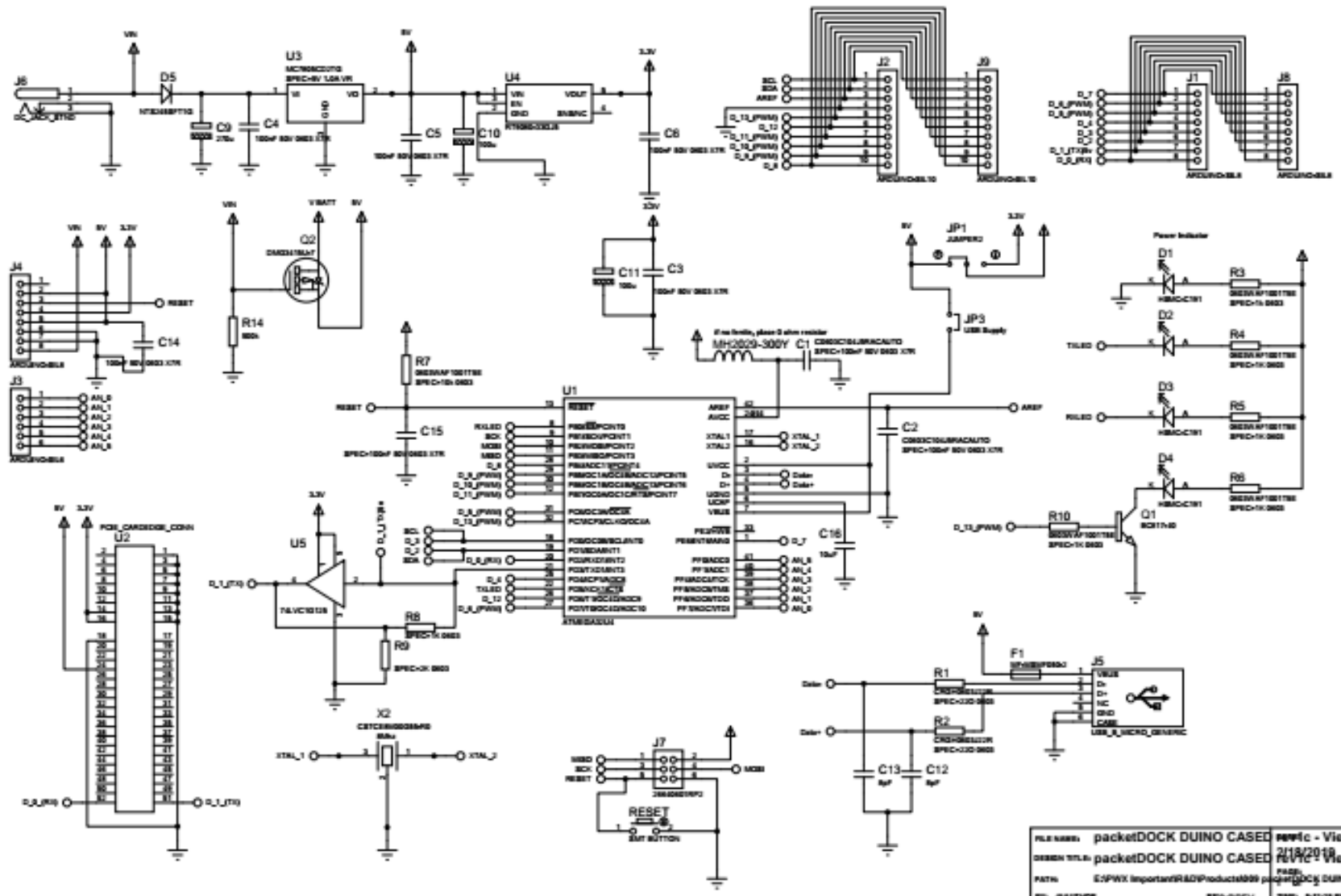
- In this tutorial, you will learn:
 - How to onboard Lora device using The Things Stack.
 - To understand how to make dashboard using Thingspeak



Packet Cased Duino

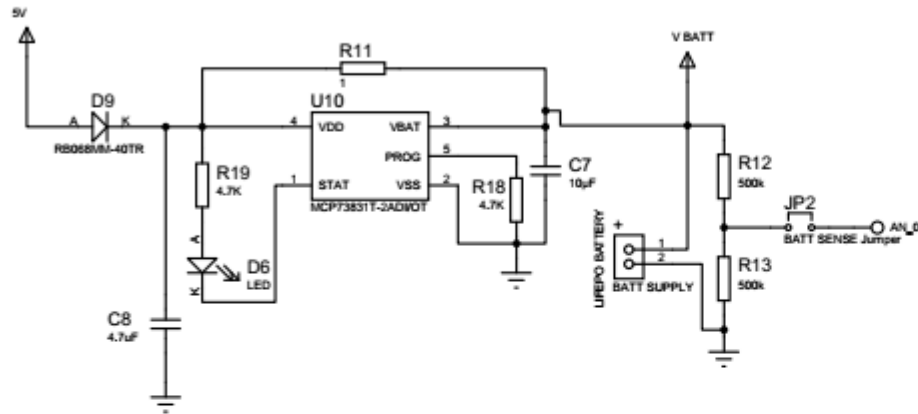


Packet Cased Duino

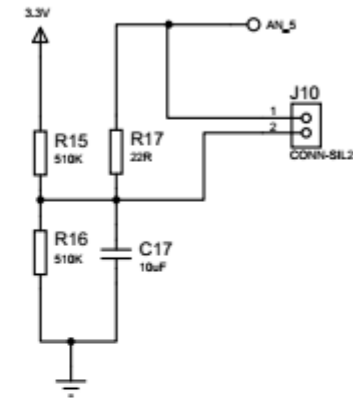


Packet Cased Duino

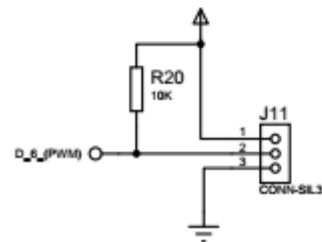
LIPO CHARGER



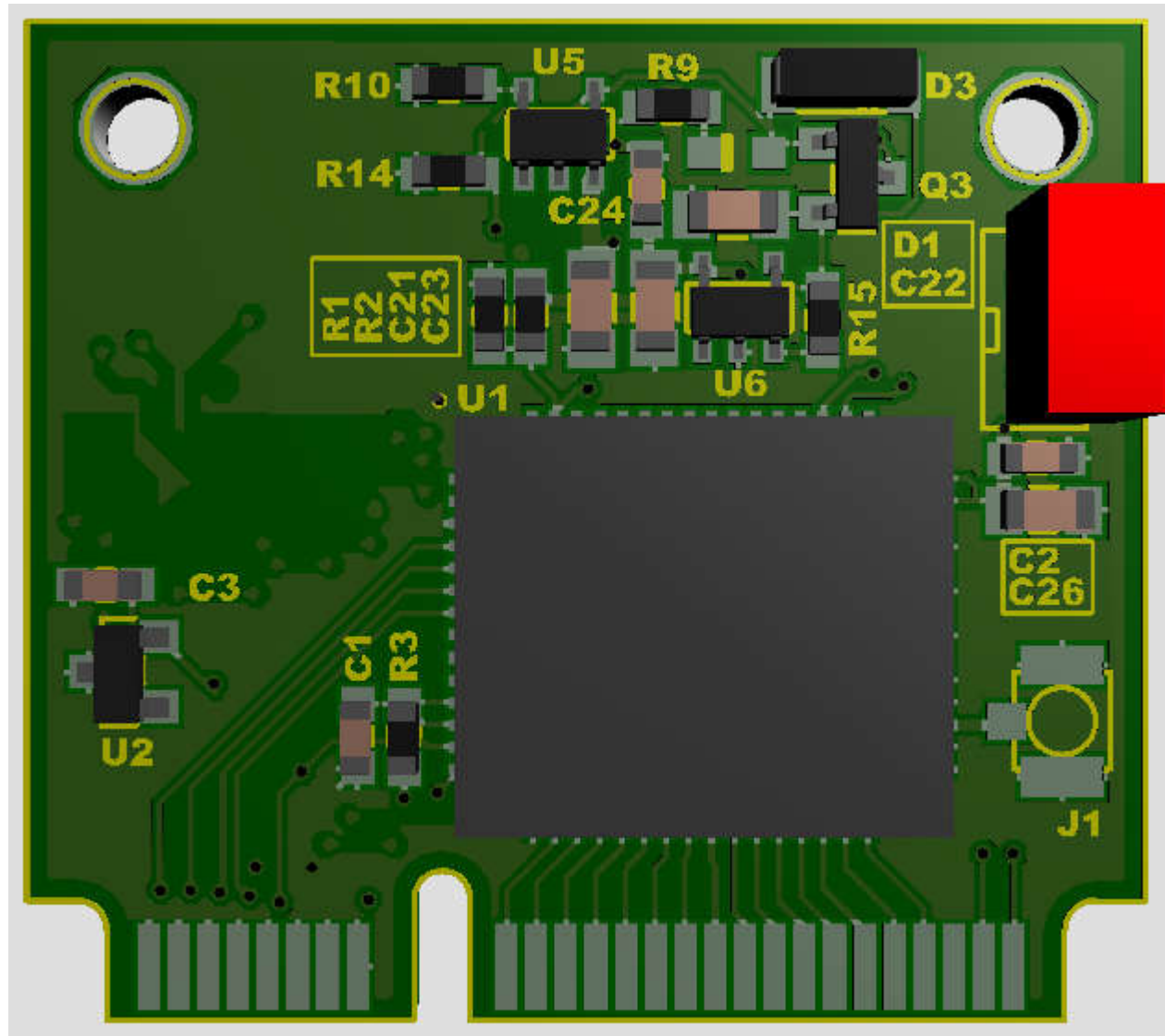
CURRENT CLAMP DRIVER



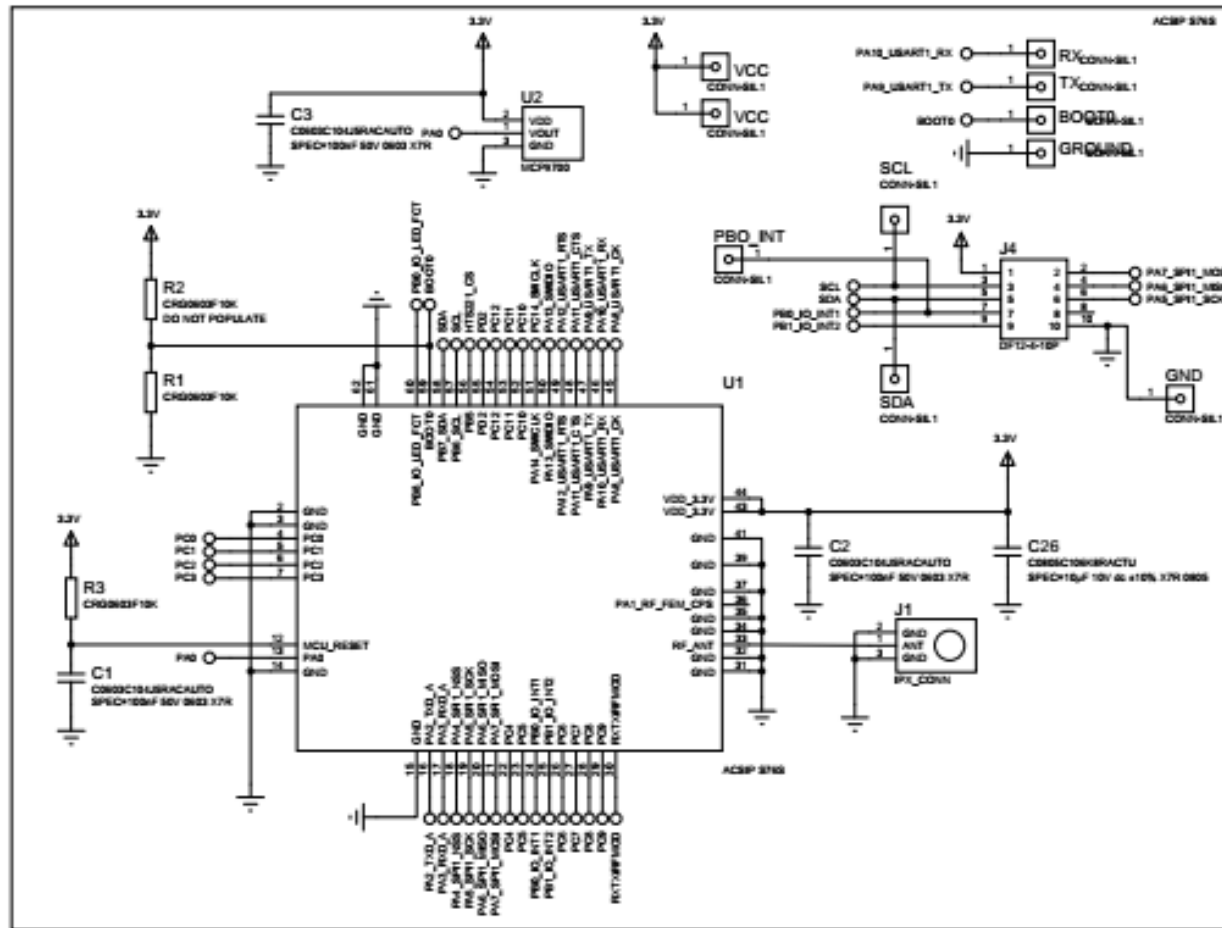
DS18B20 PULL-UP



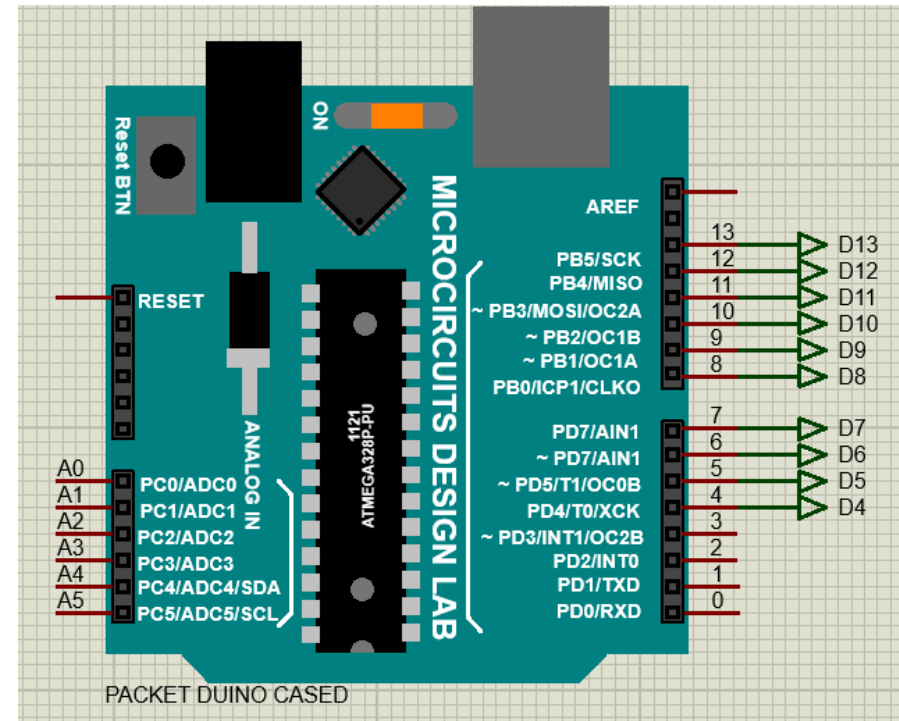
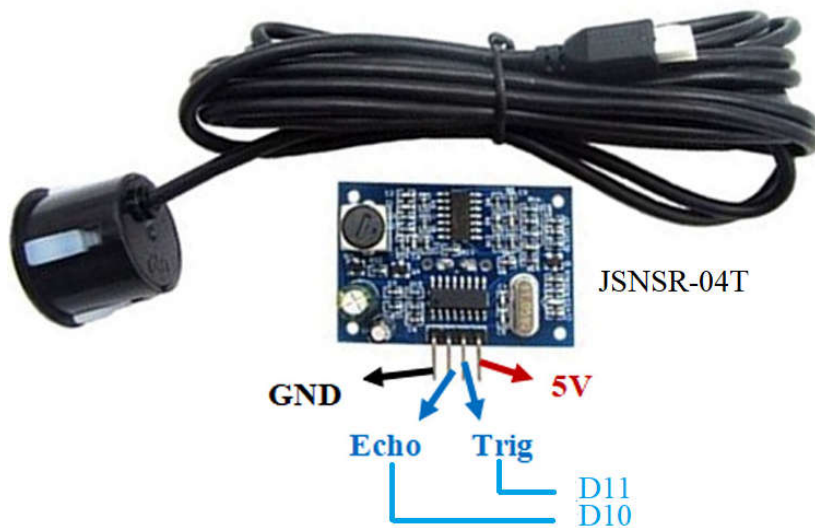
Packet One



Packet One



Water Level Sensor



Water Level Sensor

```
// -----
// Example NewPing library sketch that does a ping about 20 times per second.
// -----
//WaterLevel.ino
#include <NewPing.h>

#define LORASERIAL Serial1

#define TRIGGER_PIN 11 // Arduino pin tied to trigger pin on the ultrasonic sensor.
#define ECHO_PIN 10 // Arduino pin tied to echo pin on the ultrasonic sensor.
#define MAX_DISTANCE 500 // Maximum distance we want to ping for (in centimeters). Maximum sensor distance is rated at
400-500cm.

// Timing Macro Definintions
#define UPLINK_CYCLE 4000

// Change these 3 parameters depending on the application.
//String devAddr = "260B41D1";
//String nwksKey = "5DA12E4DA88255F38F025B0C935517A9";
//String appSKey = "2EC706377E68642A22FD6C547B680B1E";
String deveui = "70B3D57ED004BB9A";
String appeui = "000000000000000001";
String appkey = "9B62273327270C465381F6BFB534E959";
String Buffer;

NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE); // NewPing setup of pins and maximum distance.
```



Water Level Sensor

```
// Timing variables
uint32_t currentMillis = 0;
uint32_t previousMillis = 0;

uint16_t waterLevel = 0;
uint16_t validwaterLevel = 0;

void activate_connection()
{
  //String set_devaddr = "mac set_devaddr ";
  //String set_nwkskey = "mac set_nwkskey ";
  //String set_appskey = "mac set_appskey ";
  //String mac_join = "mac join abp";

  String set_deveui = "mac set_deveui ";
  String set_appeui = "mac set_appeui ";
  String set_appkey = "mac set_appkey ";
  String mac_join = "mac join otaa";

  LORASERIAL.write("sip reset");
  // Serial.write("sip reset");
  delay(5000);
  Buffer = set_deveui;
  Buffer += deveui;
  // Serial.println(Buffer);
  LORASERIAL.print(Buffer);
  delay(5000);
  Buffer = set_appeui;
  Buffer += appeui;
  // Serial.println(Buffer);
}
```



Water Level Sensor

```

LORASERIAL.print(Buffer);
delay(5000);
Buffer = set_appkey;
Buffer += appkey;
// Serial.println(Buffer);
LORASERIAL.print(Buffer);
delay(5000);
LORASERIAL.print(mac_join);
delay(3000);
// LORASERIAL.print("mac tx ucfn 2 1111"); // This is implemented to test connectivity.
// delay(4000); // This delay is needed to ensure that device is active while RX window is
waiting.LORASERIAL.print("sip sleep 1000 uart_on");
// LORASERIAL.print("sip sleep 1000 uart_on");
// delay(1000); // This delay is needed for preparing sleep.
// Serial.print("done");
}

void setup()
{
// Serial.begin(115200); // Open serial monitor at 115200 baud to see ping results.
//
// while (!Serial && millis() < 2000)
// {
//   ; // wait for serial port to connect. Needed for native USB port only
// }

LORASERIAL.begin(115200);

activate_connection();
// Serial.print("started...");

}

```



Water Level Sensor

```

void loop()
{
  currentMillis = millis();

  if(currentMillis - previousMillis > UPLINK_CYCLE)
  {
    previousMillis = currentMillis;

    char cmd[25];
    sprintf(cmd,"mac tx ucnf 1 %02x", validwaterLevel);
    LORASERIAL.print(cmd);
    // Serial.println(cmd);
  }

  delay(500);          // Wait 50ms between pings (about 20 pings/sec). 29ms should be the shortest delay between pings.
  // Serial.print("Ping: ");
  waterLevel = sonar.ping_cm();
  if(waterLevel >= 20)
  {
    validwaterLevel = waterLevel;
  }
  // Serial.print(validwaterLevel); // Send ping, get distance in cm and print result (0 = outside set distance range)
  //
  // Serial.println("cm");
}

```


Water Level Sensor

WaterLevel | Arduino 1.8.13

File Edit Sketch Tools Help

WaterLevel

```

1 // -----
2 // Example sketch showing how to use the NewPing library to
3 // -----
4 // Water level sensor
5 #include <NewPing.h>
6
7 #define TRIGGER_PIN 12
8 #define ECHO_PIN 11
9 #define MAX_DISTANCE 400
10 #define UPLINK_CYCLE 4000
11
12 // Change these 3 parameters depending on the device
13 //String devAddr = "260B41D1";
14 //String nwksKey = "5DA12E4DA88255F38F025B0C938";
15 //String appSKey = "2EC706377E68642A22FD6C547B6";
16 String deveui = "70B3D57ED004BB9A";
17 String appeui = "0000000000000001";
18 String appkey = "9B62273327270C465381F6BFB534E9";
19 String Buffer;
20
21 NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE); // NewPing setup of pins and
22
23 // Timing variables
  
```

Auto Format Ctrl+T

Archive Sketch

Fix Encoding & Reload

Manage Libraries... Ctrl+Shift+I

Serial Monitor Ctrl+Shift+M

Serial Plotter Ctrl+Shift+L

WiFi101 / WiFININA Firmware Updater

Board: "Adafruit Feather 32u4"

Port

Get Board Info

Programmer: "AVRISP mkII"

Burn Bootloader

Boards Manager...

Adafruit Boards

Arduino ARM (32-bits) Boards

Arduino AVR Boards

ESP32 Arduino

ESP8266 Boards (3.0.2)

RAKwireless ESP32 Modules

RAKwireless nRF Modules

Rakwireless Raspberry Modules

STM32 Boards (selected from submenu)

STM32F1 Boards (Arduino_STM32)

STM32F4 Boards (STM32duino.com)

STM8 Boards

STM8S Boards

Adafruit Flora

• Adafruit Feather 32u4

Adafruit Feather 328P

Adafruit Gemma (ATtiny85 @ 8MHz)

Adafruit Trinket (ATtiny85 @ 8MHz)

Adafruit Trinket (ATtiny85 @ 16MHz)

Adafruit Metro

Pro Trinket 5V/16MHz (USB)

Pro Trinket 3V/12MHz (USB)

Pro Trinket 5V/16MHz (FTDI)

Pro Trinket 3V/12MHz (FTDI)

Adafruit Circuit Playground Classic

Adafruit ItsyBitsy 32u4 5V 16MHz

Adafruit ItsyBitsy 32u4 3V 8MHz

Adafruit Bluefruit Micro

Adafruit 32u4 Breakout

Done Saving.

94

Adafruit Feather 32u4 on CC

Water Level Sensor

WaterLevel | Arduino 1.8.13

File Edit Sketch Tools Help

WaterLevel

```

1 // -----
2 // Example sketch showing how to use the NewPing library to
3 // -----
4 // Water level sensor
5 #include <NewPing.h>
6
7 #define TRIGGER_PIN 12
8 #define ECHO_PIN 11
9 #define MAX_DISTANCE 400 // cm
10 #define NEWPING_MAX 30 // cm, maximum range of sensor (distance
11 #define NEWPING_DELAY 10 // ms, NewPing delay time
12 #define NEWPING_WAITING 100 // ms, NewPing waiting time
13 // Timing
14 #define UPLINK_CYCLE 4000
15
16 // Change these 3 parameters depending on the sensor
17 //String devAddr = "260B41D1";
18 //String nwksKey = "5DA12E4DA88255F38F025B0C938";
19 //String appSKey = "2EC706377E68642A22FD6C547B6";
20 String deveui = "70B3D57ED004BB9A";
21 String appeui = "0000000000000001";
22 String appkey = "9B62273327270C465381F6BFB534E9";
23 String Buffer;
24
25 NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE); // NewPing setup of pins and
26
27 // Timing variables
  
```

Auto Format Ctrl+T

Archive Sketch

Fix Encoding & Reload

Manage Libraries... Ctrl+Shift+I

Serial Monitor Ctrl+Shift+M

Serial Plotter Ctrl+Shift+L

WiFi101 / WiFININA Firmware Updater

Board: "Adafruit Feather 32u4"

Port

Get Board Info

Programmer: "AVRISP mkII"

Burn Bootloader

Boards Manager...

Adafruit Boards

Arduino ARM (32-bits) Boards

Arduino AVR Boards

ESP32 Arduino

ESP8266 Boards (3.0.2)

RAKwireless ESP32 Modules

RAKwireless nRF Modules

Rakwireless Raspberry Modules

STM32 Boards (selected from submenu)

STM32F1 Boards (Arduino_STM32)

STM32F4 Boards (STM32duino.com)

STM8 Boards

STM8S Boards

Adafruit Flora

• Adafruit Feather 32u4

Adafruit Feather 328P

Adafruit Gemma (ATtiny85 @ 8MHz)

Adafruit Trinket (ATtiny85 @ 8MHz)

Adafruit Trinket (ATtiny85 @ 16MHz)

Adafruit Metro

Pro Trinket 5V/16MHz (USB)

Pro Trinket 3V/12MHz (USB)

Pro Trinket 5V/16MHz (FTDI)

Pro Trinket 3V/12MHz (FTDI)

Adafruit Circuit Playground Classic

Adafruit ItsyBitsy 32u4 5V 16MHz

Adafruit ItsyBitsy 32u4 3V 8MHz

Adafruit Bluefruit Micro

Adafruit 32u4 Breakout

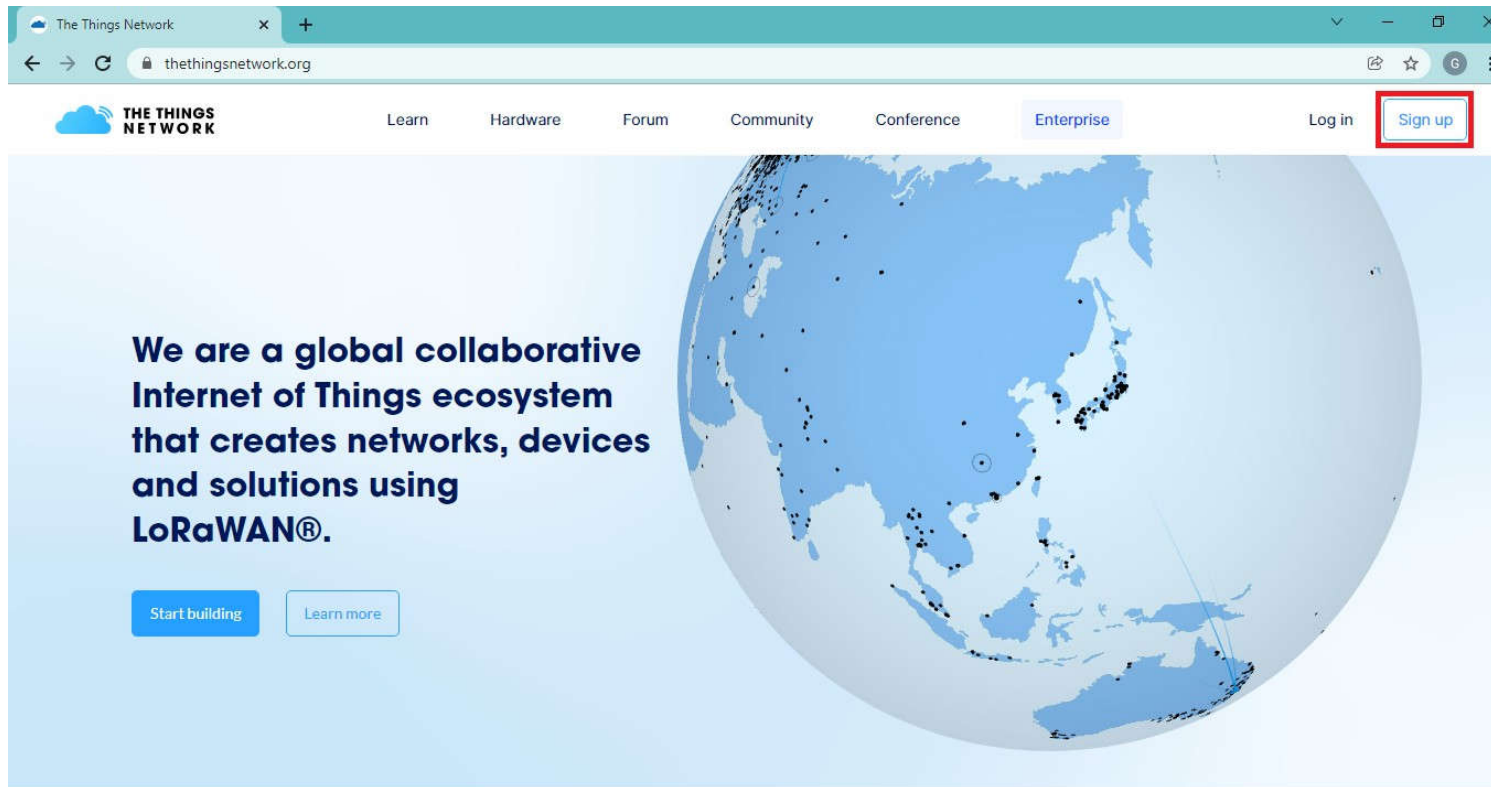
rated at 400-500cm.

Done Saving.

94

Adafruit Feather 32u4 on CC

Water Level Sensor




Water Level Sensor


CREATE AN ACCOUNT

Welcome aboard! Fill in your details to create an account on The Things Network and start exploring the world of LoRaWAN.


USERNAME
Your public name.

 rakwireless_apps

EMAIL ADDRESS
Your email address stays private. An activation email will be sent to you shortly (please check your spam folder).

 rakwireless_apps@rakwireless.com

PASSWORD
Use at least 6 characters.



NEWSLETTER
Subscribe to the newsletter.

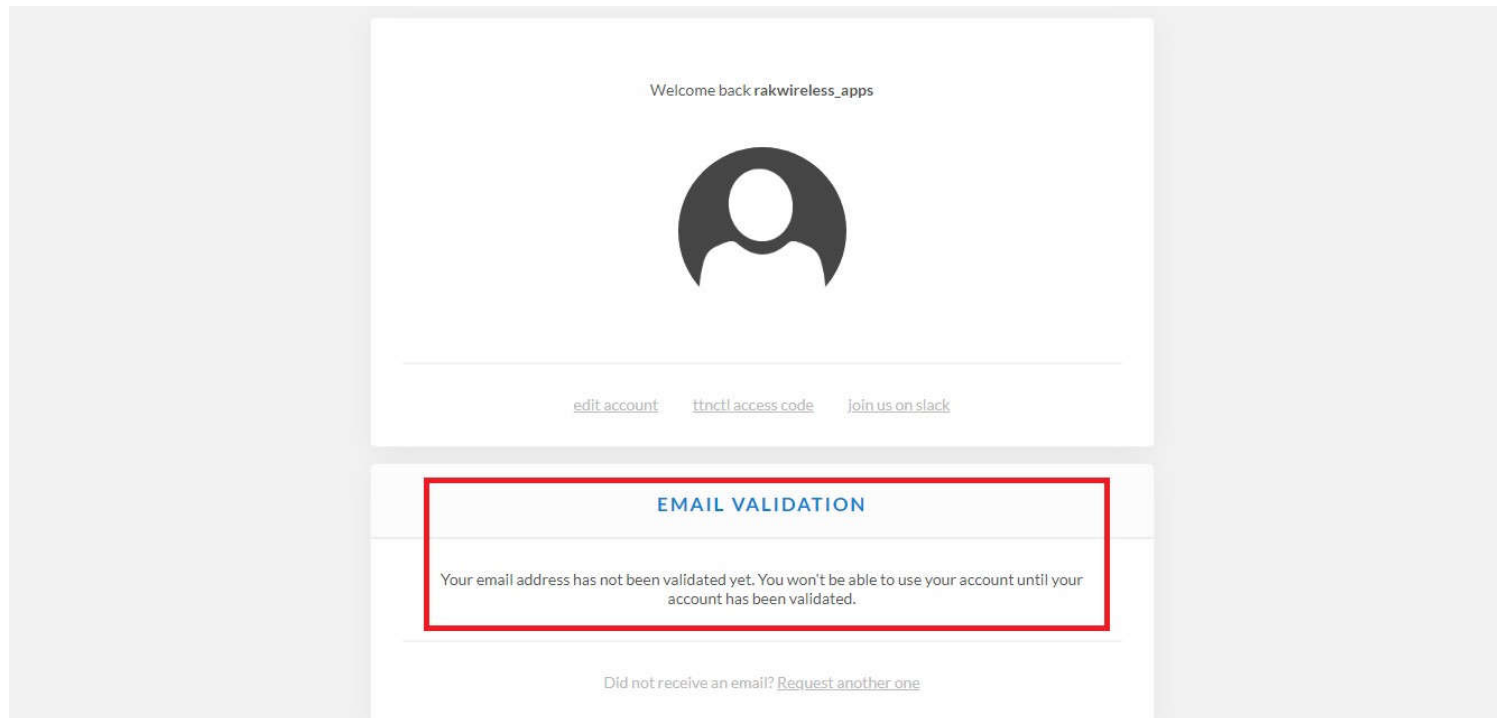
☒

Create account

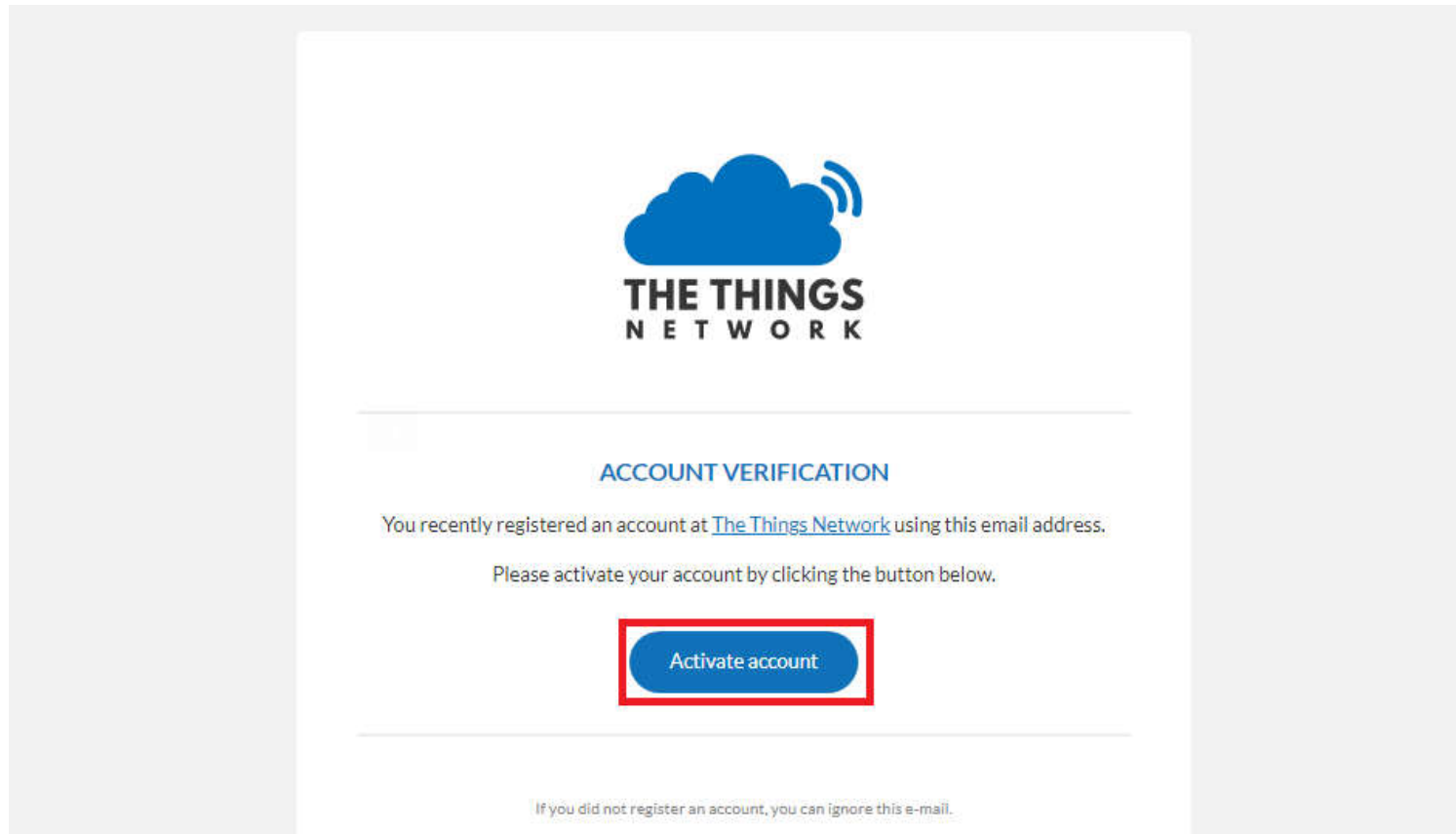
By registering an account you agree to our [Terms and Conditions](#) and [Privacy Policy](#).



Water Level Sensor



Water Level Sensor



Water Level Sensor



Login to The Things Stack with **The Things ID**

Username or email •

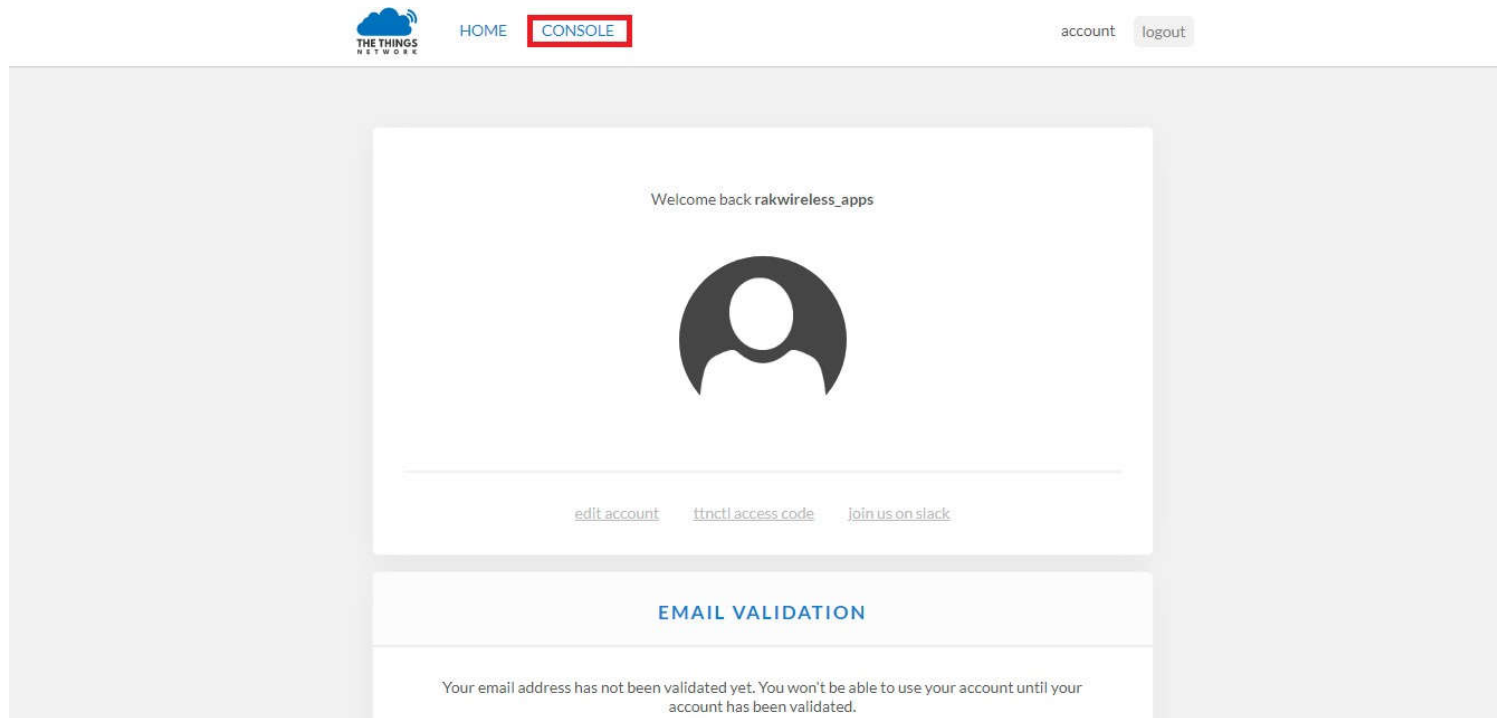
Password •

[Register](#) [Forgot password?](#)


The Things ID by The Things Industries



Water Level Sensor



Water Level Sensor

**THE THINGS NETWORK**

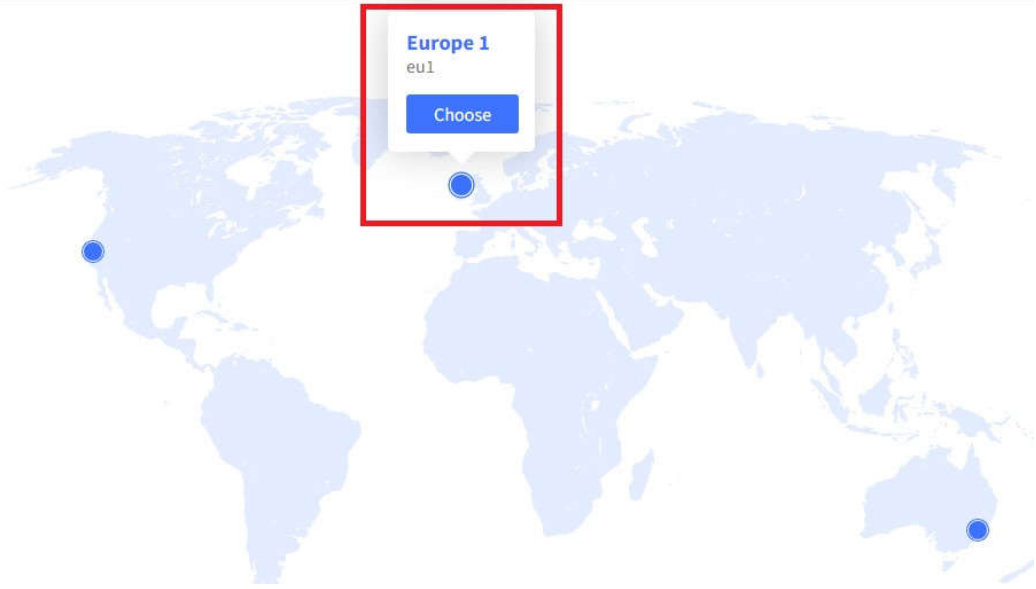
The Things Network Cluster Picker

Select a cluster to start adding devices and gateways.

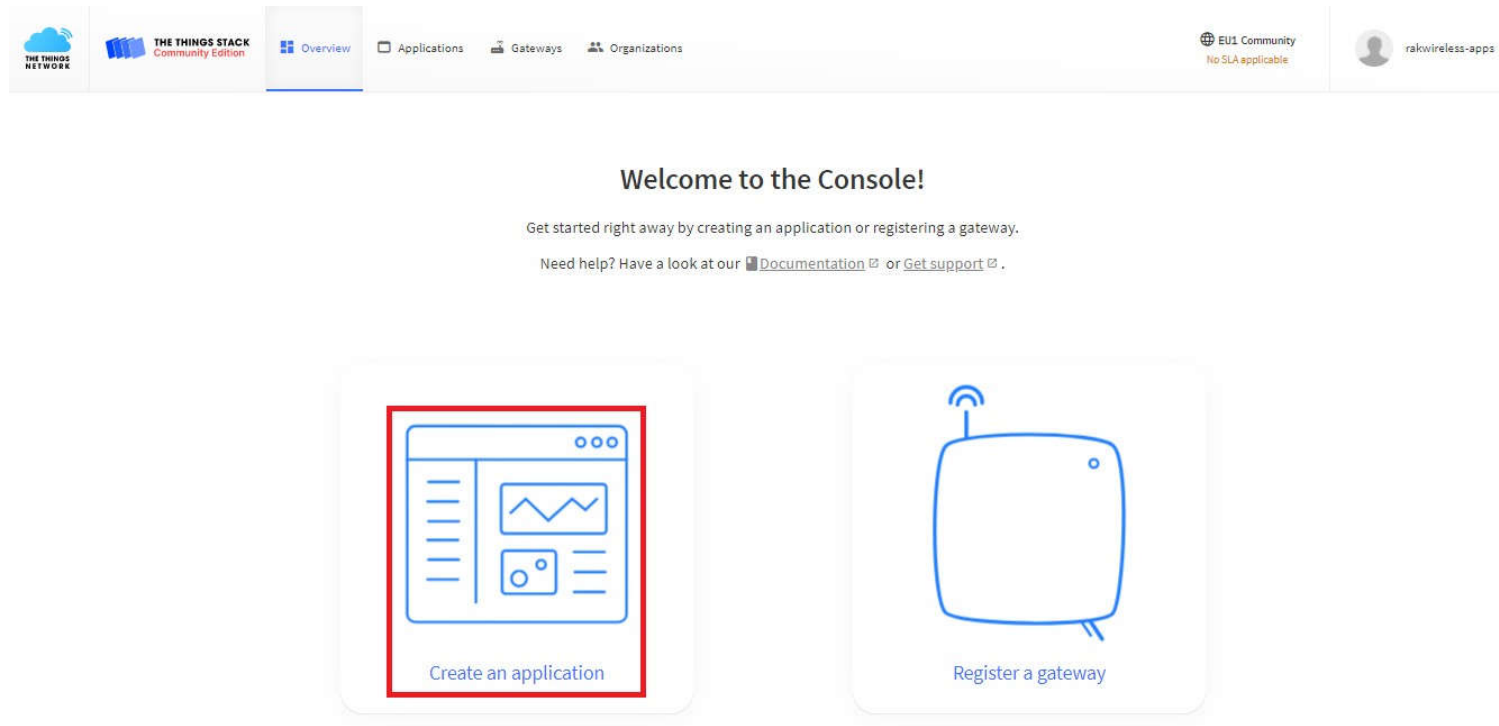
Europe 1
eu1 - Dublin, Ireland

North America 1
nam1 - California, United States

Australia 1
au1 - Sydney, Australia



Water Level Sensor



Water Level Sensor

Add application

Application ID *

water-level-iecep

Application name

water level iecep



Description


water level iecep using thingspeak

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

Create application



Water Level Sensor

**water level iecep**
ID: water-level-iecep

No recent activity ?

0 End devices 1 Collaborator 0 API keys

General information

Application IDwater-level-iecep

Created atMar 26, 2022 24:12:36

Last updated atMar 26, 2022 24:12:36

• Live data

See all activity →

+ 24:12:36 water-leve... Create application

End devices (0)

Search

Import end devices

+ Add end device



Water Level Sensor

Register end device

From The LoRaWAN Device Repository **Manually**

Frequency plan ⓘ *

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

LoRaWAN version ⓘ *

LoRaWAN Specification 1.0.2 | v

Regional Parameters version ⓘ *

RP001 Regional Parameters 1.0.2 revision B | v

Show advanced activation, LoRaWAN class and cluster settings v

DevEUI ⓘ *

70 B3 D5 7E D0 04 E7 7D ☐ Generate 1/50 used

AppEUI ⓘ *

00 00 00 00 00 00 00 01

AppKey ⓘ *

66 9F FB 02 E9 CF 00 FC 0A D8 8F A3 E9 35 D0 E6

End device ID ⓘ *

water-level-iecep-rizal

This value is automatically prefilled using the DevEUI

After registration

☒ View registered end device

☐ Register another end device of this type

Register end device

Water Level Sensor

Overview Live data Messaging Location Payload formatters Claiming General settings

General information

End device ID: water-level-iecep-rizal

Description: This end device has no description

Created at: Mar 26, 2022 24:16:47

Activation information

AppEUI: 00 00 00 00 00 00 00 01

DevEUI: 70 B3 D5 7E D0 04 E7 7D

Root key ID: n/a

AppKey: 66 9F FB 02 E9 CF 00 FC 0A D8 8F A3 E9 3...

NwkKey: n/a

Session information

No data available


Live data

+ 24:16:47 Create end device

Location

No location information

Water Level Sensor

**water-level-iecep-rizal**
ID: water-level-iecep-rizal

↑ n/a ↓ n/a • No activity yet ⓘ

Overview Live data Messaging Location **Payload formatters** Claiming General settings

Uplink Downlink

Setup

Formatter type *
Custom Javascript formatter ▼

Formatter code *

```
1 function decodeUplink(input) {  
2   return {  
3     data: {  
4       bytes: input.bytes  
5     },  
6     warnings: [],  
7     errors: []  
8   };  
9 }
```

Test

Byte payload FPort
 1

Decoded test payload

Complete uplink data

Water Level Sensor

```
function Decoder(bytes, port)
{
  // Decode an uplink message from a buffer
  // (array) of bytes to an object of fields.
  var level;

  var decoded = {};

  if (port == 1)
  {
    level = (bytes[0]);
    level |= (bytes[1]);
    decoded.field1 = level;

    return decoded;
  }
}
```



Water Level Sensor

Uplink

Downlink

Setup

Formatter type *

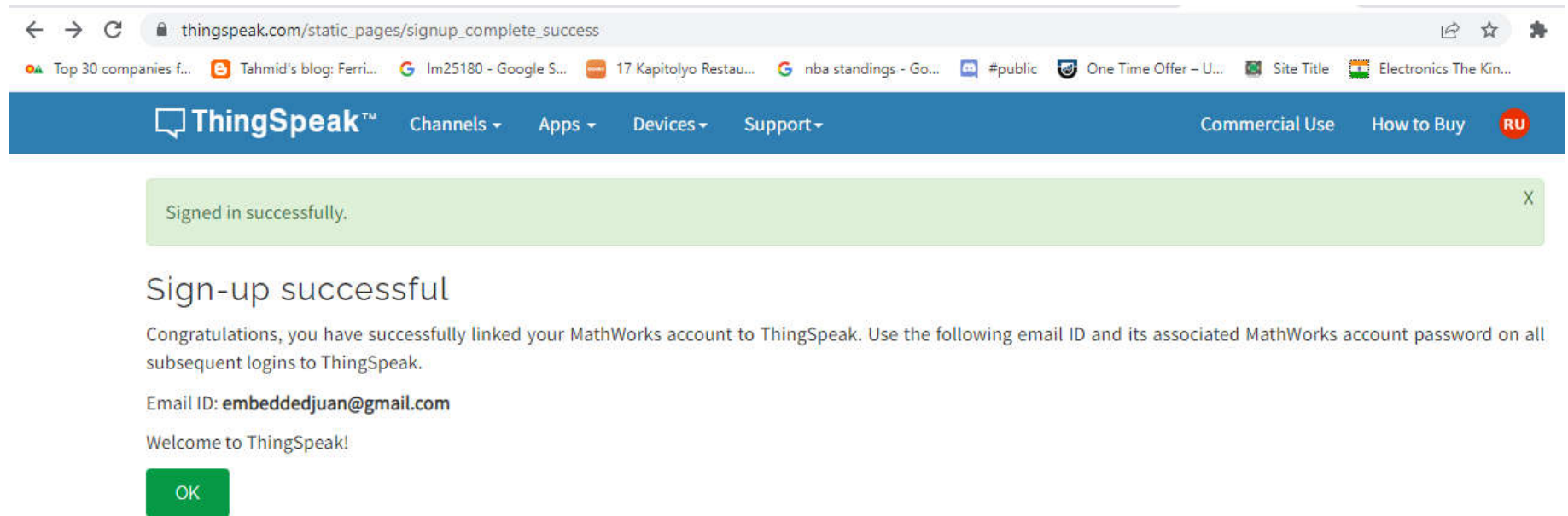
Custom Javascript formatter

Formatter code *

```
1 function Decoder(bytes, port)
2 {
3   // Decode an uplink message from a buffer
4   // (array) of bytes to an object of fields.
5   var level;
6
7   var decoded = {};
8
9   if (port == 1)
10  {
11    level = (bytes[0]);
12    level |= (bytes[1]);
13    decoded.field1 = level;
14
15    return decoded;
16  }
17 }
```

Save changes

Water Level Sensor



The screenshot shows a web browser window with the URL `thingspeak.com/static_pages/signup_complete_success`. The browser's address bar and tabs are visible at the top. The ThingSpeak website header is blue, featuring the logo and navigation links: Channels, Apps, Devices, and Support. On the right side of the header, there are links for Commercial Use, How to Buy, and a red button labeled RU. A green notification banner at the top of the page content area states "Signed in successfully." with a close button (X) on the right. Below the banner, the heading "Sign-up successful" is displayed in a large, dark font. Underneath the heading, a paragraph of text reads: "Congratulations, you have successfully linked your MathWorks account to ThingSpeak. Use the following email ID and its associated MathWorks account password on all subsequent logins to ThingSpeak." This is followed by the text "Email ID: **embeddedjuan@gmail.com**" and "Welcome to ThingSpeak!". At the bottom of this section, there is a green button with the text "OK".

Signed in successfully.

Sign-up successful

Congratulations, you have successfully linked your MathWorks account to ThingSpeak. Use the following email ID and its associated MathWorks account password on all subsequent logins to ThingSpeak.


Email ID: **embeddedjuan@gmail.com**

Welcome to ThingSpeak!

OK





Water Level Sensor

 Channels ▾ Apps ▾ Devices ▾ Support ▾

My Channels

New Channel

Search by tag 

Name ▴ ▾	Created ▴ ▾	Updated ▴ ▾
 Water Level <div>Private Public Settings Sharing API Keys Data Import / Export</div>	2022-03-25	2022-03-25 08:36

Water Level Sensor



New Channel

Name	<input type="text" value="Water Level IECEP Rizal"/>	
Description	<input type="text" value="Water Level IECEP Rizal using <u>Thingspeak</u>"/>	
Field 1	<input type="text" value="Field Label 1"/>	<input checked="" type="checkbox"/>
Field 2	<input type="text"/>	<input type="checkbox"/>
Field 3	<input type="text"/>	<input type="checkbox"/>
Field 4	<input type="text"/>	<input type="checkbox"/>
Field 5	<input type="text"/>	<input type="checkbox"/>
Field 6	<input type="text"/>	<input type="checkbox"/>
Field 7	<input type="text"/>	<input type="checkbox"/>



Water Level Sensor

Link to GitHub	<input type="text" value="https://github.com/"/>
Elevation	<input type="text"/>
Show Channel Location	<input type="checkbox"/>
Latitude	<input type="text" value="0.0"/>
Longitude	<input type="text" value="0.0"/>
Show Video	<input type="checkbox"/>
	<input checked="" type="radio"/> YouTube
	<input type="radio"/> Vimeo
Video URL	<input type="text" value="http://"/>
Show Status	<input type="checkbox"/>
	<div>Save Channel</div>

Water Level Sensor



Water Level Sensor

ThingSpeak™ Channels ▾ Apps ▾ Devices ▾ Support ▾

Private View Public View Channel Settings Sharing API Keys Data

Write API Key

Key

2A3G1MTU5HXG1V82

Generate New Write API Key

Read API Keys

Key

G8KYSBE9QSQ00TSQ


Note

Save Note

Delete API Key



Water Level Sensor


[Channels ▾](#)
[Apps ▾](#)
[Devices ▾](#)
[Support ▾](#)

Water Level IECEP Rizal

Channel ID: **1685520**

Water Level IECEP Rizal using Thingspeak

Author: mwa0000026056247

Access: Private

[Private View](#)
[Public View](#)
[Channel Settings](#)
[Sharing](#)
[API Keys](#)
[Data Import / Export](#)

Channel Sharing Settings

☒ Keep channel view private
☐ Share channel view with everyone
☐ Share channel view only with the following users:

Email Address

Enter email here

Add User

Help

ThingSpeak a the settings o requires the a

Channel

- Keep c you wi
- Share of your
- Share



Water Level Sensor

Live data

< > Payload formatters

Integrations

MQTT

Webhooks

Storage Integration

AWS IoT

Azure IoT Hub

LoRa Cloud

Collaborators

API keys

General settings

Hide sidebar

Formatter code*

```

1 function Decoder(bytes, port)
2 {
3   // Decode an uplink message from a buffer
4   // (array) of bytes to an object of fields.
5   var level;
6   //var battInt;
7   var decoded = {};
8
9   if (port == 1)
10  {
11    level = (bytes[0]);
12    level |= (bytes[1]);
13    decoded.field1 = level;
14    //battInt = (bytes[2] & 0x80 ? 0xFFFF<<16 : 0) | bytes[2]<<8 | by
15    //decoded.battery = (battInt+6.61/1023)/1.00;
16    return decoded;
17  }
18 }

```

Paste application formatter

Save changes

Decoded test payload

Complete uplink data

[Learn more about payload formatters](#)

Water Level Sensor

Water level test

Overview

End devices

Live data

Payload formatters

Integrations

MQTT

Webhooks

Storage Integration

AWS IoT

Azure IoT Hub

LoRa Cloud

Collaborators

Applications > Water level test > Webhooks

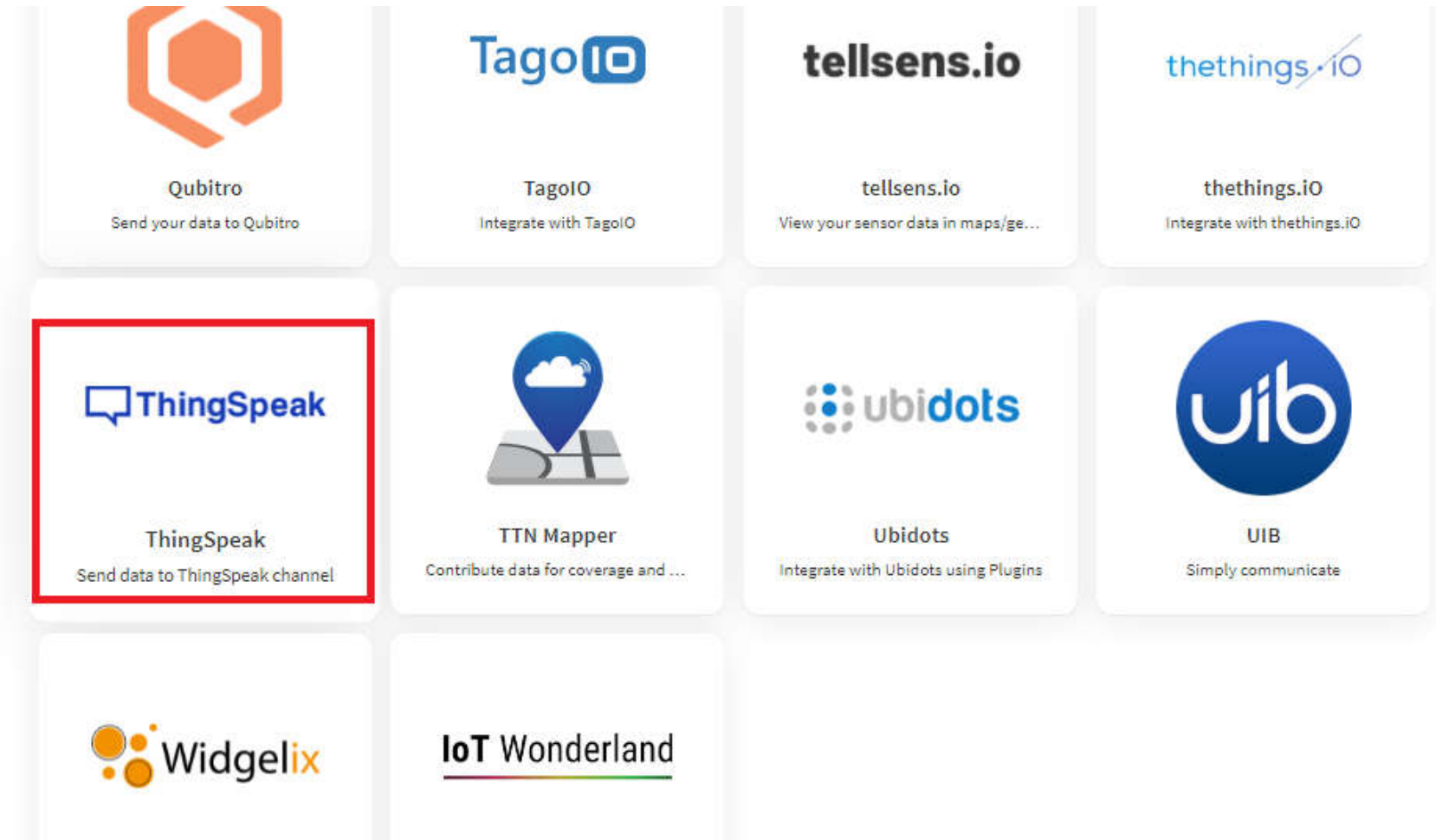
Webhooks (0)

+ Add webhook

ID	Base URL	Template ID	Format
No items found			



Water Level Sensor



Water Level Sensor

Add custom webhook

Template information

 **ThingSpeak** Send data to ThingSpeak channel

[About ThingSpeak](#) | [Documentation](#)


Template settings

Webhook ID *

Channel ID *

ThingSpeak Channel ID

API Key *

ThingSpeak Write API Key

Create thingspeak webhook



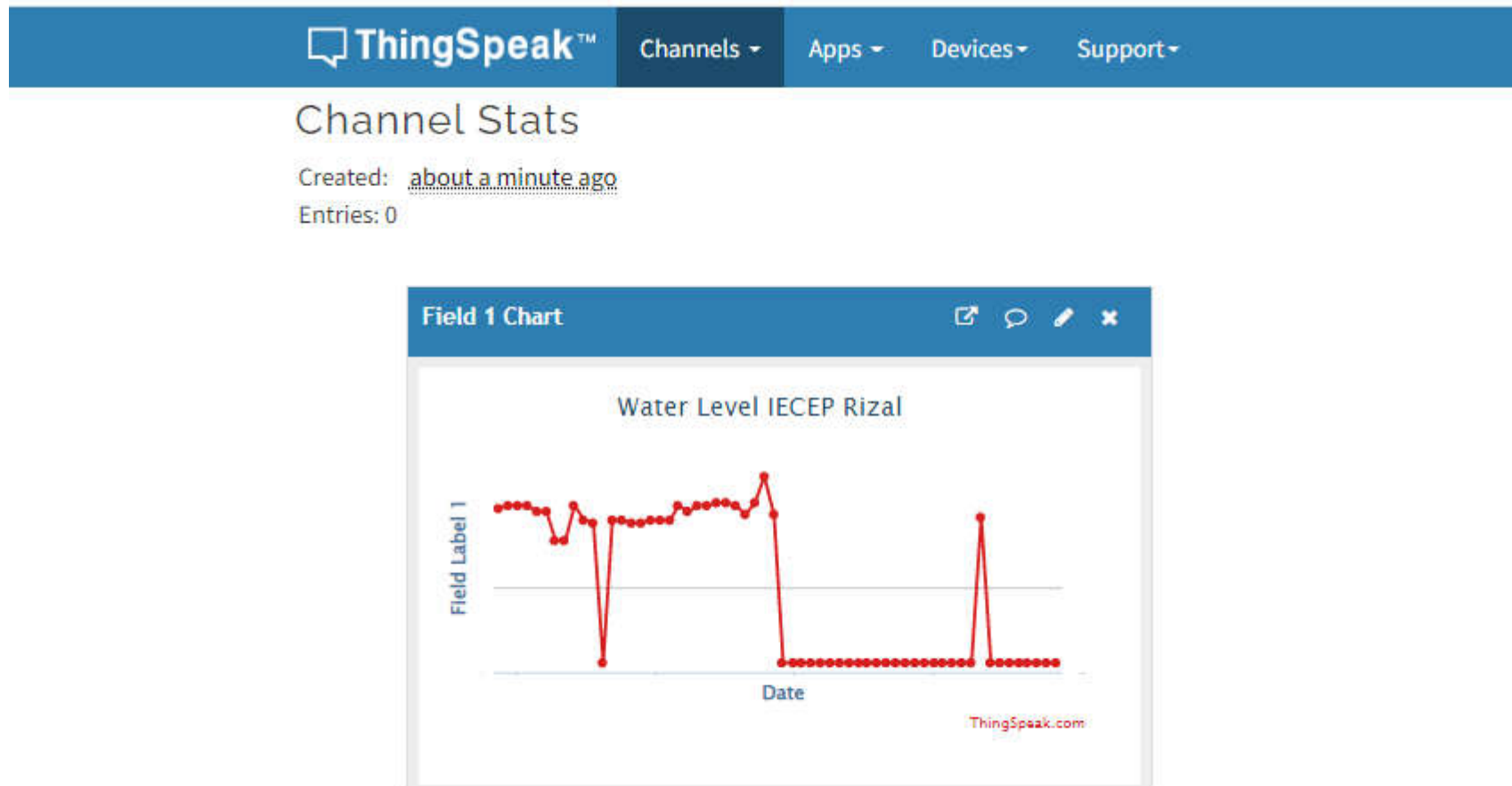
Water Level Sensor

Applications > Water level > **Live data**

Time	Entity ID	Type	Data preview	Verbose stream	Export as JSON	Pause	Clear
24:52:47	water-level	Fail to send webhook	Error:undefined:undefined				
↑ 24:52:47	water-level	Forward uplink data message	Payload: { field1: 72 } 48 <>	FPort: 1 Data rate: SF7BW125 SNR: 11.2 RSSI: -60			
↑ 24:52:37	water-level	Forward uplink data message	Payload: { field1: 73 } 49 <>	FPort: 1 Data rate: SF7BW125 SNR: 13.5 RSSI: -57			
24:52:27	water-level	Fail to send webhook	Error:undefined:undefined				
↑ 24:52:27	water-level	Forward uplink data message	Payload: { field1: 86 } 56 <>	FPort: 1 Data rate: SF7BW125 SNR: 13.8 RSSI: -59			
↑ 24:52:17	water-level	Forward uplink data message	Payload: { field1: 74 } 4A <>	FPort: 1 Data rate: SF7BW125 SNR: 14 RSSI: -58			
24:52:06	water-level	Fail to send webhook	Error:undefined:undefined				



Water Level Sensor



THANK YOU !

Q&A

