Introduction To IoT – S25

LAB-5

Introduction:

MQTT (MQ Telemetry Transport) is a Machine-to-Machine connectivity protocol. This is fancy jargon which basically means its a way for devices to communicate. It allows data to be **published** and **subscribed** to by clients through one central server on topic channels. The **server** can be as simple as a local computer or Raspberry Pi (using an IP address) or larger and public like mqtt.thingspeak.com.

In this lab, you will be writing and reading data to/from the Thingspeak channel using MQTT and REST APIs.

Part-1) ESP32 Publish sensor data to Thingspeak using ESP32.

a) Introduction to ThingSpeak:

The platform is primarily aimed towards IoT Projects and data analytics using visuals. To get started with the free services of Thingspeak you will first need to Signup using your email ID, once that is done along with the email verification you will be greeted with a similar-looking page as showed in fig1.

Setting ThingSpeak:

i. First login to the Thingspeak server. https://thingspeak.com/login
ii. If you are a new user, then create the new account.
iii. After login you show this type of webpage, here we have already created few projects, but if you are a new user then click on the New Channel.
iv. After clicking the new channel then give the name of the new channel.
v. If you want to give the description, then write the description. and select the Field 1 and don't forget the click on checkbox, after clicking on the check box give the field name.
vi. Finally scroll down and click on save.

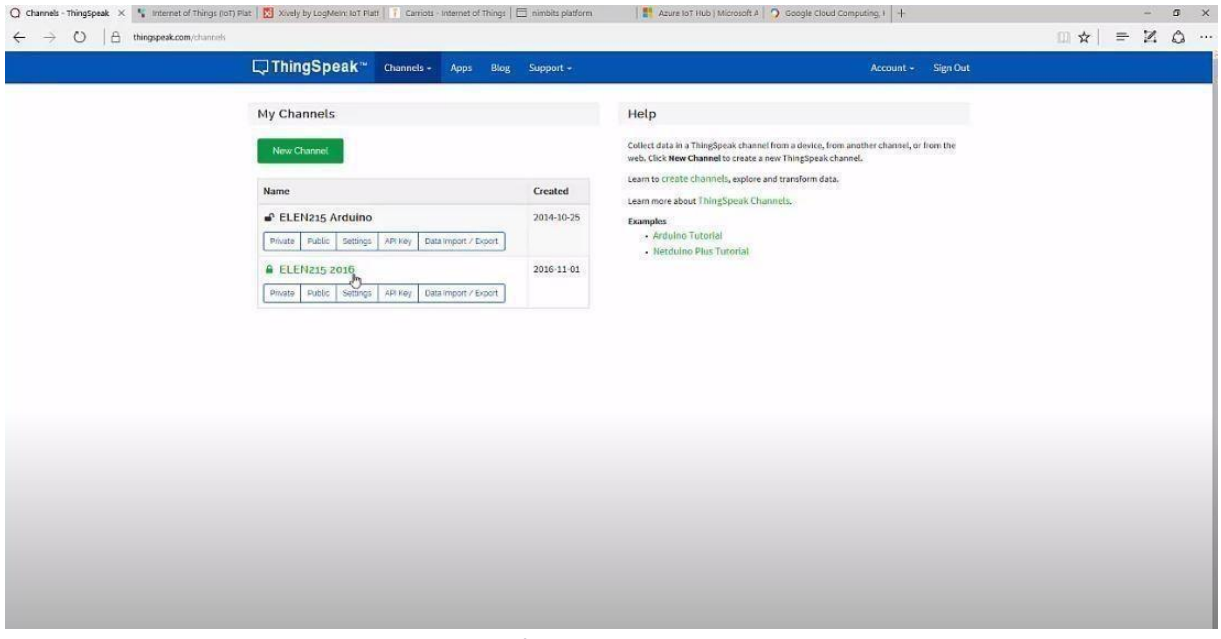

Figure 1

Now we will look at some of the terminologies:

- **Reading/Downloading Data:** Getting data on your ESP32 from the server is a read operation.

- **Writing/Uploading Data**: Sending data from your ESP8266/ESP32 to the server is a write operation.

- **API Key**: To have data security and to prevent anyone randomly from reading/writing data to your server there needs to be some sort of security/password and the API Key is something intended towards this. API Key is a long alphanumeric key which is needed to read/write data to the server. There are separate keys for reading and writing data. (Refer fig:2)

- **Channel**: A channel in ThingSpeak is a software counterpart of an IoT hardware device that you connect to Thingspeak, in our case an ESP32 will utilise one entire channel of our bandwidth. In a free account of ThingSpeak, you can have a maximum of 4 channels.

- **Field**: Each channel has 8 fields. A field is a variable and stores/shares a data type, for example when we send temperature and humidity from our device to the server, both the parameters will use one field each of the channel.

That's pretty much it about ThingSpeak!

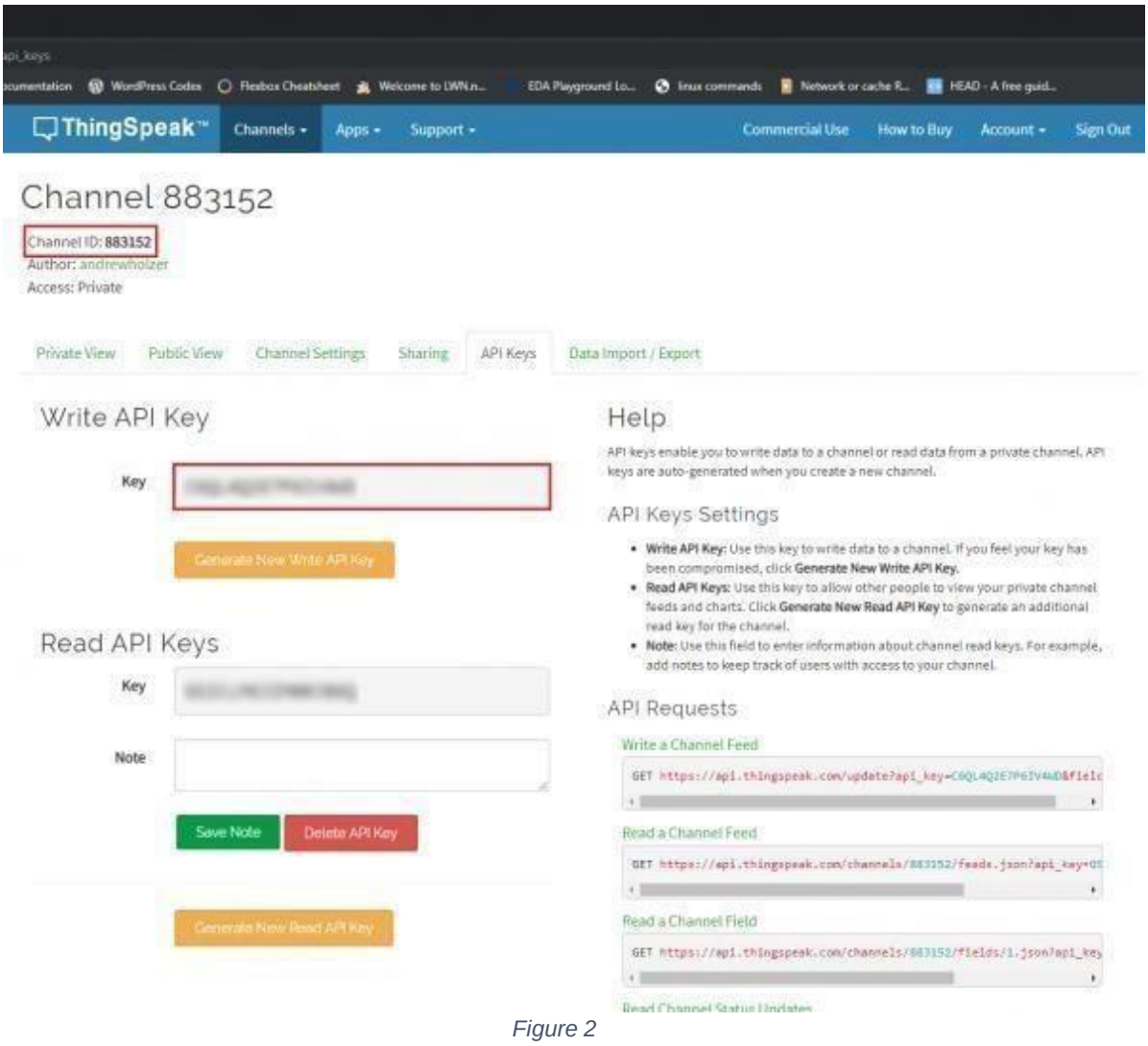

Figure 2

c) Library Files:

Following 3 libraries will be required to run this code. Download the zip file, extract the same and copy this to your Arduino library folder.

i) PubSubClient.h
ii) WiFi.h
iii) Thingspeak.h

e) Procedure:

i. Create account on the ThingSpeak website and add a new device to publish data.
ii. Take your ESP32 and start coding.
iii. Include the above-mentioned libraries.

iv. Publish and Subscribe to a ThingSpeak Channel Using Secure MQTT - MATLAB & Simulink - MathWorks India
v. Use the above link to get pseudo code for publishing using MQTT to ThingSpeak server. The example in the link if for ESP8266 which is very similar to ESP32.
vi. Also, in the example they have used secure MQTT but there could be issues with that hence you can use normal MQTT as well.
vii. Use appropriate logical reasoning and debugging to edit the to satisfy the requirements of the lab.

f) Expected output:

A random data should be plotted on the Thingspeak server in the Private View tab. You can use random number generator to create this data.

Part 2: Displaying Ultrasonic Sensor Data on ThingSpeak

a) Components Required

1. Ultrasonic Sensor (HC-SR04)
2. ESP32
3. Breadboard
4. Few Jumper Wires

b) Circuit Diagram

1. VCC of the Ultrasonic Sensor → 3.3V/5V of ESP32
2. GND of the Ultrasonic Sensor → GND of ESP32
3. Trigger (TRIG) Pin → GPIO 5 (or any other digital pin)
4. Echo Pin → GPIO 18 (or any other digital pin)

c) Library Files Required

1. WiFi.h → Connect ESP32 to WiFi
2. ThingSpeak.h → Send data to ThingSpeak
3. PubSubClient.h → MQTT communication

d) Pseudocode for Ultrasonic Sensor Experiment

1. Include necessary libraries for WiFi, ThingSpeak, and MQTT.
2. Connect to WiFi using credentials.
3. Initialize the ultrasonic sensor pins (TRIG and ECHO).
4. Measure distance by sending a pulse from TRIG and measuring the response time on ECHO.
5. Send the distance data to ThingSpeak via MQTT.
6. In the thinkspeak channel go to Private/Public View , then select **Add Visualisation** and add field chart for the particulare field data. (Properties of the graph can be changed later).

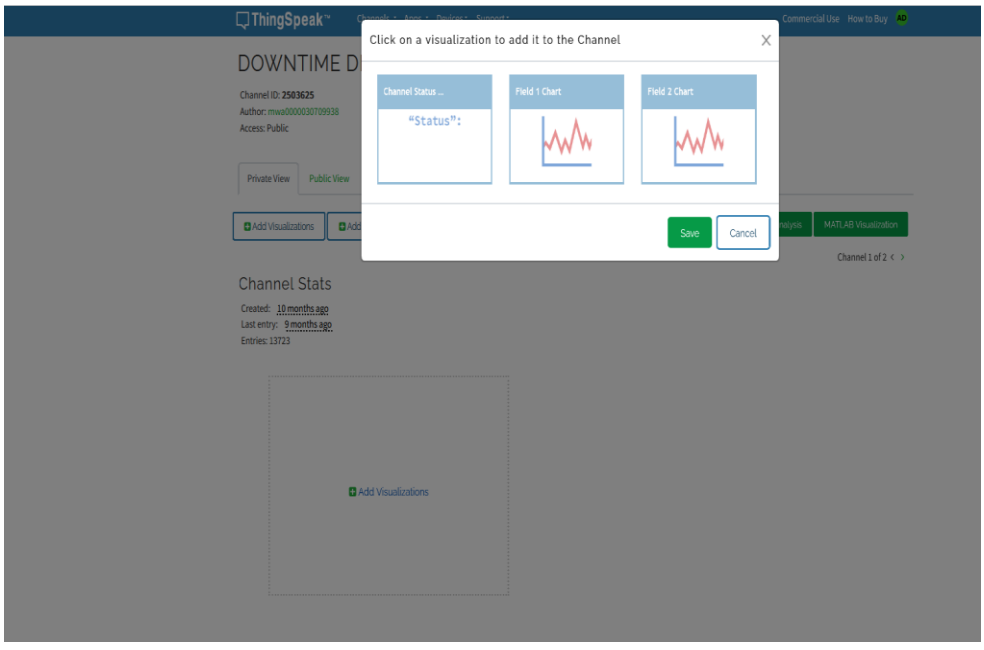

Figure 3

e) Expected Output

- The ultrasonic sensor measures distance and updates the ThingSpeak graph in real time.
- The ThingSpeak dashboard displays the distance trend in a live plot.