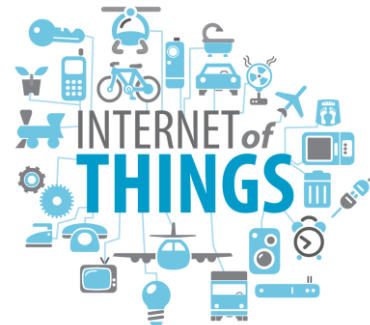


# INTERNET OF THINGS

## 4.4

### ESP32 STORE DATA IN CLOUD

---



THINGSPEAK



# New Channel

Name

Description

Field 1



Field 2



---

**Step 1:** Sign up new account in **mathworks.com**

**Step 2:** Sign in with mathworks account in **thingspeak.com**

**Step 3:** Go to **Channels > New Channel > Input Fields > Save Channel**

Private View

Public View

Channel Settings

Sharing

API Keys

## Write API Key

Key

WKG80B1HSGVC9P4F



Generate New Write API Key

**Step 4:** Select **API Keys** Tab.

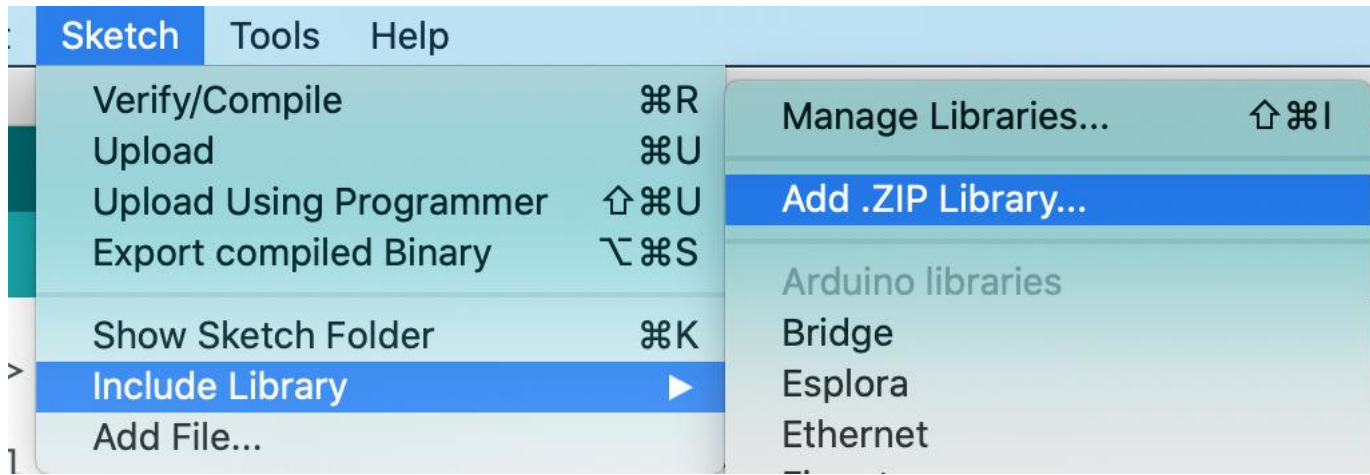
Remember **Write API Key**

## Read API Keys

Key

NPJHMUBEI7ARA0KN





---

**Step 5:** Download Zip file to your computer

<https://github.com/mathworks/thingspeak-arduino>

**Step 6:** In Arduino IDE, choose *Sketch > Include Library > Add Zip Library > Select Zip file*

## weather

Channel ID: 1080479

Author: mwa0000018758431

Access: Private

```
#include "ThingSpeak.h"
#include <ESP8266WiFi.h>

char ssid[] = "your wifi name";
char pass[] = "your wifi password";

unsigned long myChannelNumber = 0; // Replace the 0 with your channel number
const char * myWriteAPIKey = "your Write API"; // Paste your ThingSpeak Write API Key

WiFiClient client;

void setup() {
  //Connect to WiFi Network. DIY

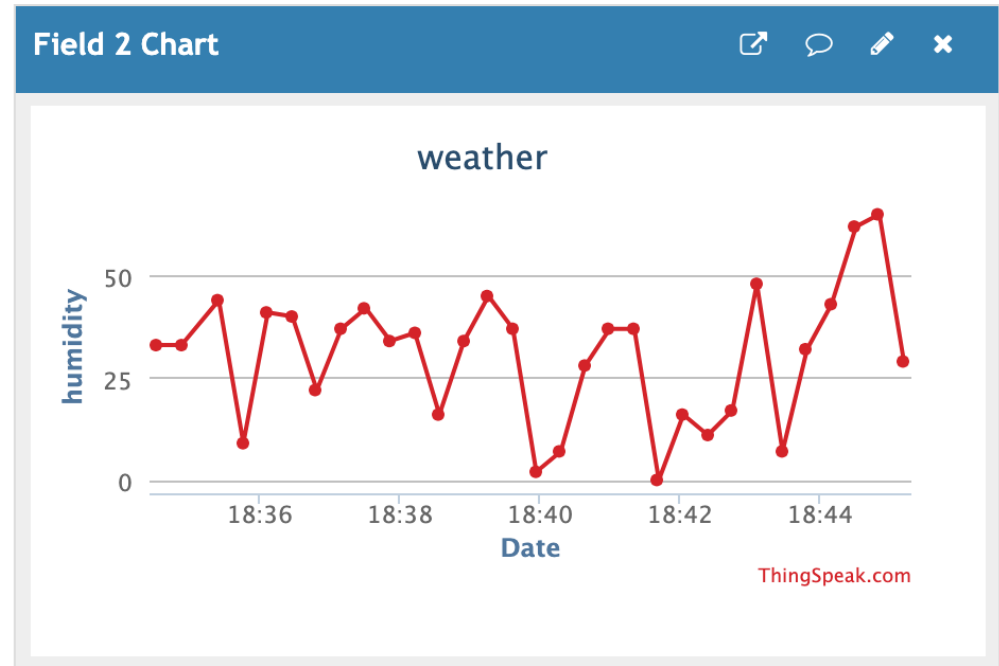
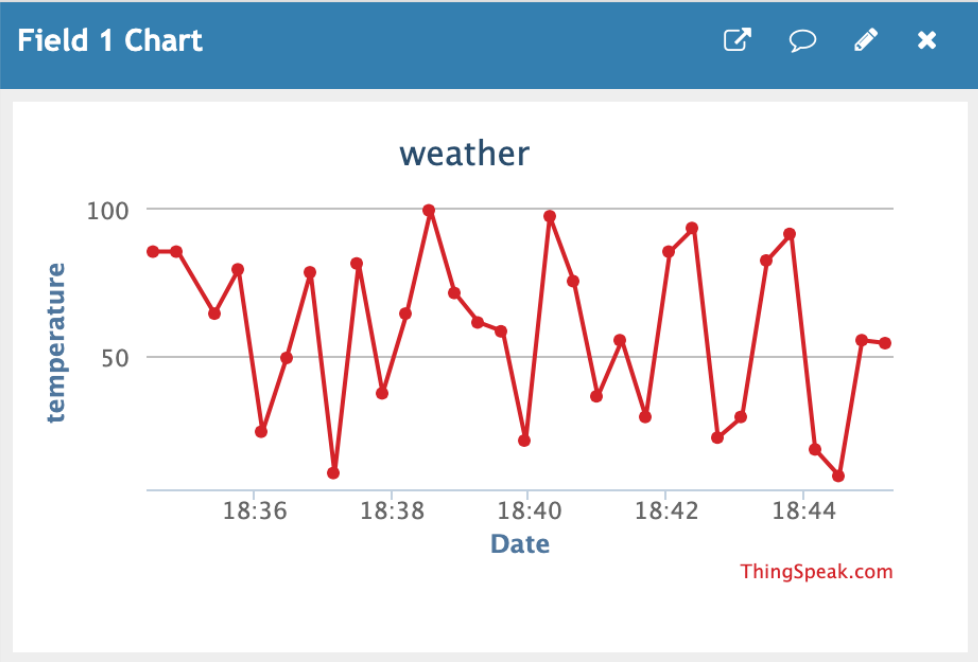
  ThingSpeak.begin(client);
}
```



```
void loop() {  
  int h = random(100);  
  int t = random(70);  
  // Write data to fields  
  ThingSpeak.setField(1, h); //setField(field, value)  
  ThingSpeak.setField(2, t); //setField(field, value)  
  int returncode = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);  
  
  // Check return code  
  if (returncode == 200) {  
    Serial.println("Channel update successful.");  
  }  
  else {  
    Serial.println("Problem updating channel. HTTP error code " + String(z));  
  }  
  
  delay(20000);  
}
```



Random Data





```
// Paste your ThingSpeak Read API Key  
const char * myReadAPIKey = "NPJHMUBEI7ARA0KN";
```

```
int t = ThingSpeak.readIntField(myChannelNumber, 1, myReadAPIKey);  
int h = ThingSpeak.readIntField(myChannelNumber, 2, myReadAPIKey);  
Serial.print("Temperature:");  
Serial.println(t);  
Serial.print("Humidity:");  
Serial.println(h);
```

Read data from ThingSpeak




**Cloud Storage**  
for Firebase


FIREBASE

# Let's start with a name for your project<sup>?</sup>

Project name

demo

 fir-a87d8

 fit.hcmus.edu.vn

Continue

---

**Step 1:** Sign up/Sign in  
**firebase.google.com** using Google  
account


**Step 2:** Add new project

**Step 3:** Named for your project

# Configure Google Analytics

Choose or create a Google Analytics account 

NamCaoXuan

Analytics location 

Vietnam

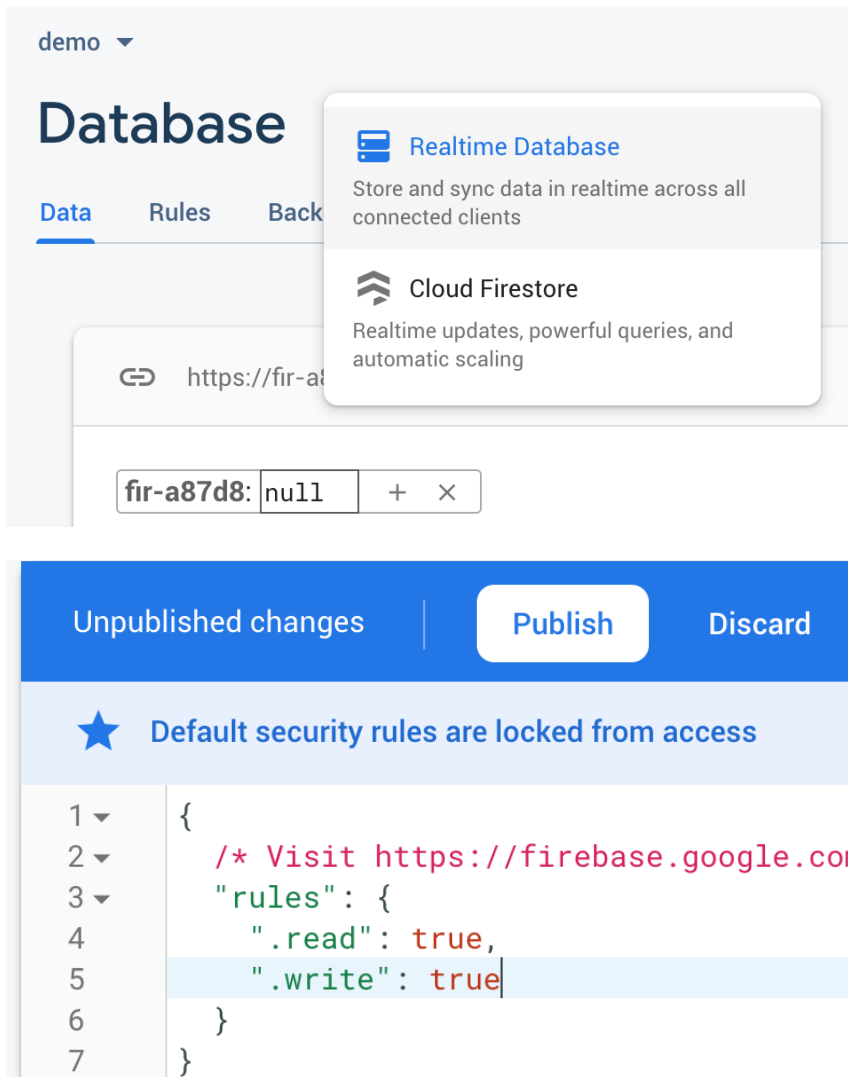
Data sharing settings and Google Analytics terms

☒ Use the default settings for sharing Google Analytics data. [Learn more](#)

- ✓ Share your Analytics data with Google to improve Google Products and Services
- ✓ Share your Analytics data with Google to enable Benchmarking
- ✓ Share your Analytics data with Google to enable Technical Support

---

## Step 4: Configure Google Analytics (optional)



**Step 5:** Developer Menu > Database > Create database

**Step 6:** Change Database mode is **Realtime Database**

**Step 7:** Choose Rules tab > Edit rules > set “**true**” for read and write rule > Publish

demo ▾

# Database



Realtime Database ▾

 **Data**

Rules

Backups

Usage



<https://fir-a87d8.firebaseio.com/>



**fir-a87d8:** null

+


×

**Step 8:** Copy your firebase  
host


---

General Cloud Messaging Integrations **Service accounts** Data privacy Users and permissions


[Manage service account permissions](#)

 **Firebase Admin SDK**


Legacy credentials

 **Database secrets**

Other service accounts

 **5 service accounts from Google Cloud Platform**

**Database Secrets**

 Database secrets are currently deprecated and use a legacy Firebase token generator. Update your source code with the Firebase Admin SDK. [Learn more](#)

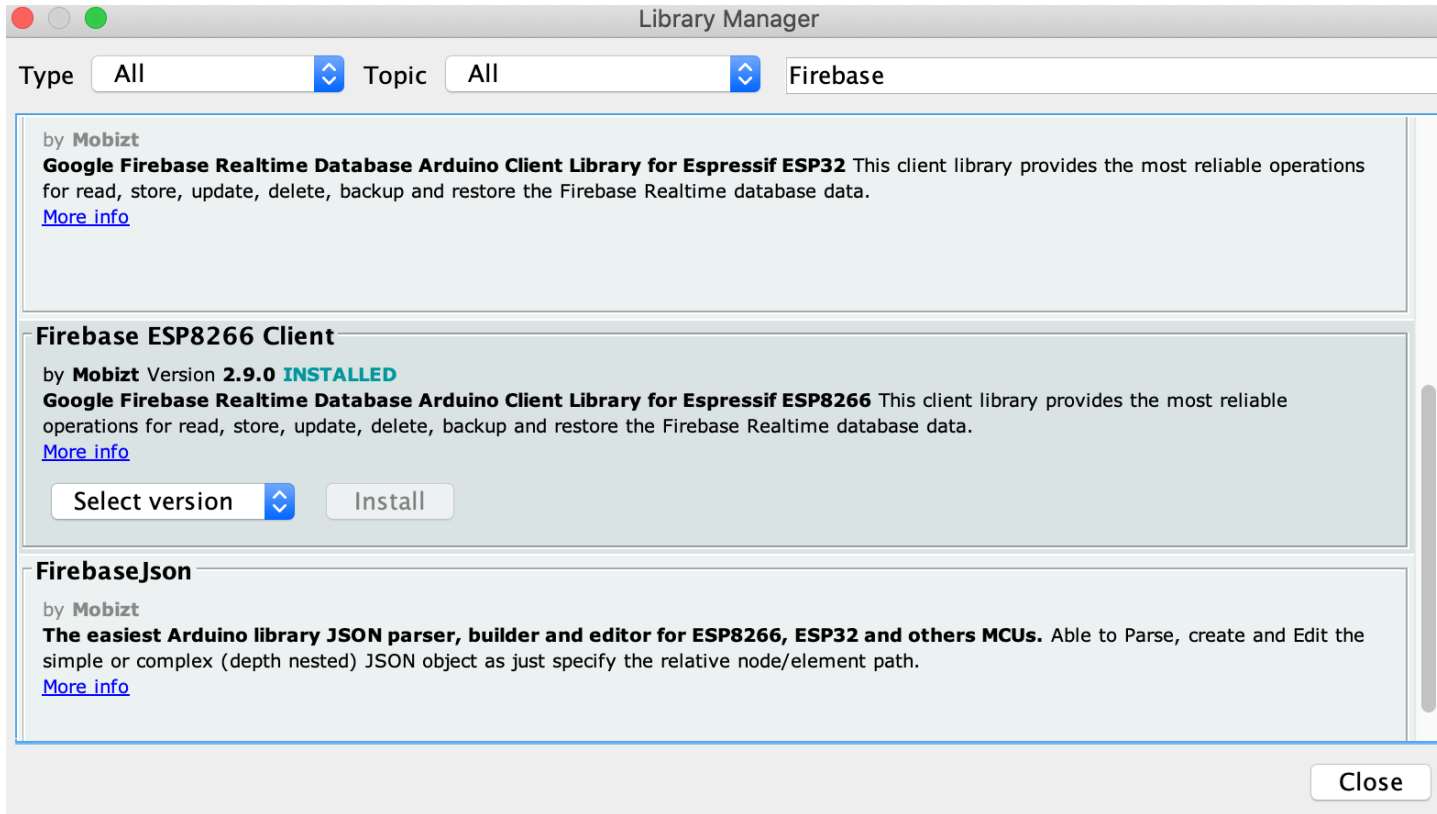
Create custom database authentication tokens using a legacy Firebase token generator. At least one secret must exist at all times. [Learn more](#)

[Add secret](#)

| Database  | Secret |
|-----------|--------|
| fir-a87d8 | .....  |

**Step 9:** Select *Gear icon* >  
*Project Settings* > *Service accounts* > *Database secrets* > *Copy Secret key*





---

**Step 10:** In Arduino IDE,  
Sketch > Include Library >  
Manage Libraries...

**Step 11:** Search “**Firebase**”  
and install the latest version  
of “**Firebase ESP8266 Client**”



```
#include <ESP8266WiFi.h>
#include <FirebaseESP8266.h>

#define FIREBASE_HOST "your firebase host"
#define FIREBASE_AUTH "your database secret key"

#define ssid "your wifi network"
#define password "your wifi password"

FirebaseData firebaseData;
```

```
void setup() {  
    //Connect to wifi network. DIY  
  
    Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);  
    Firebase.reconnectWiFi(true);  
  
    if (!Firebase.beginStream(firebaseData, "/Nodes/led"))  
    {  
        Serial.println("Could not begin stream");  
        Serial.println("REASON: " + firebaseData.errorReason());  
        Serial.println();  
    }  
}
```

```
void loop() {  
  for (int i= 0; i<10; i++) {  
    Firebase.setInt(firebaseData, "/Nodes/led", i);  
    delay(1000);  
  }  
}
```

Write data to Firebase

```
void loop() {  
    if (Firebase.getInt(firebaseData, "/Nodes/led")) {  
        if (firebaseData.dataType() == "int") {  
            Serial.println(firebaseData.intData());  
        }  
    }  
    delay(1000);  
}
```

Read data from Firebase



AWS IoT

AWS IoT