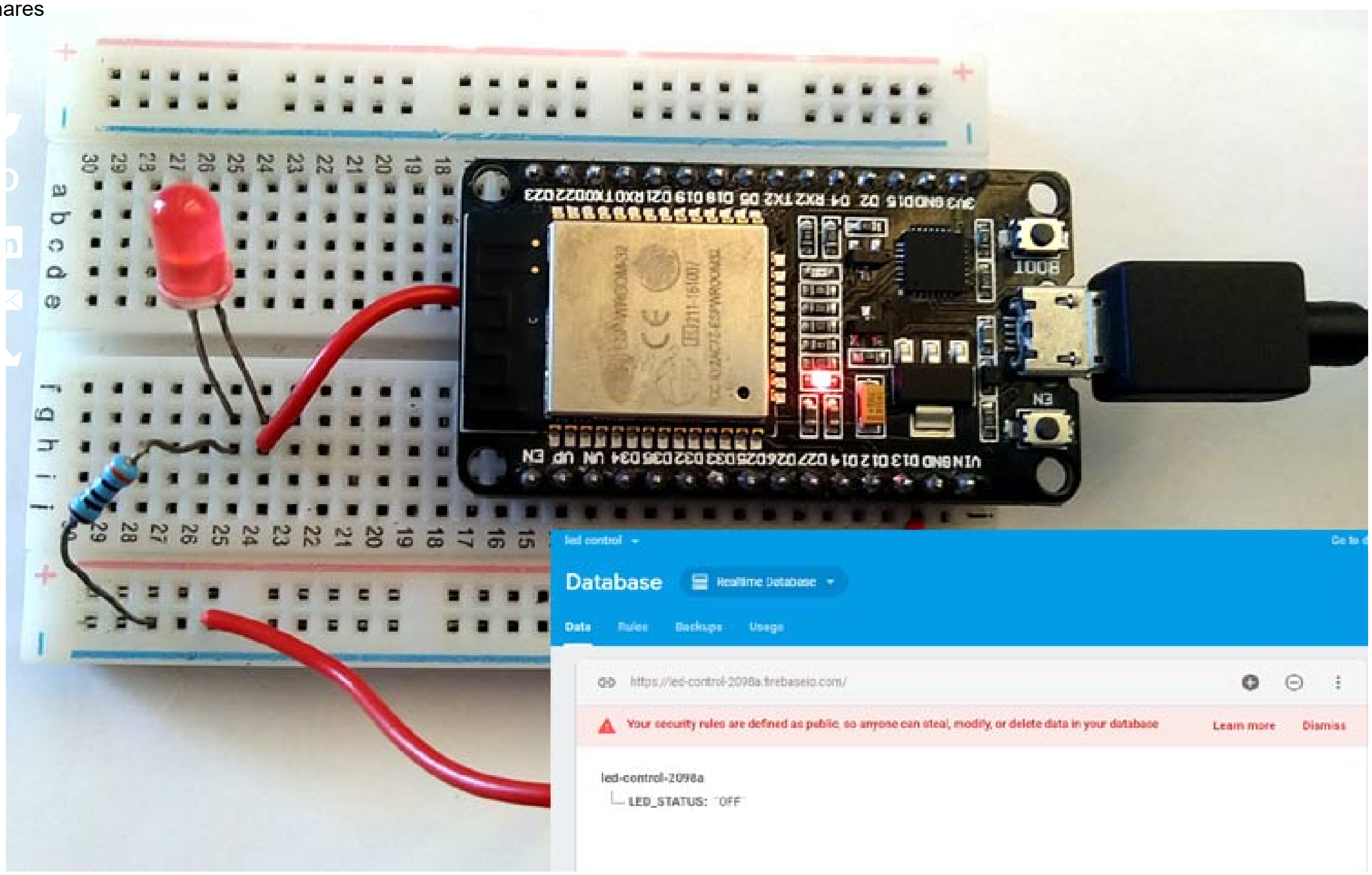
05 Feb, 2019 / 0 Comments

IoT Based LED control using Google Firebase and ESP32 NodeMCU

Shares



Your security rules are defined as public, so anyone can steal, modify, or delete data in your database

led-control-2098a
LED_STATUS: "OFF"

ESP32 is a powerful hardware platform for IoT applications and is widely used for prototyping and development of IoT applications. In some of our previous sessions, we have used ESP32 with many IoT cloud platforms:

- ThingSpeak and ESP32: How to send Data to Cloud using ESP32 (https://iotdesignpro.com/projects/how-to-send-data-to-thingspeak-cloud-using-esp32)
- ESP32 Web Server: Control an LED from Webpage (https://iotdesignpro.com/projects/control-led-using-esp32-based-webserver)
- Google Assistant controlled LED using ESP32 and Adafruit IO (https://iotdesignpro.com/projects/google-assistant-controlled-led-using-ESP32-and-adafruit-io)

- IoT Controlled LED using ESP32 with Blynk App (https://iotdesignpro.com/projects/iot-controlled-led-using-esp32-and-blynk-app)
- How to Trigger LED using IFTTT and ESP32 with Email Notification (https://iotdesignpro.com/projects/iot-triggered-led-using-IFTTT-and-ESP32-with-email-notification)

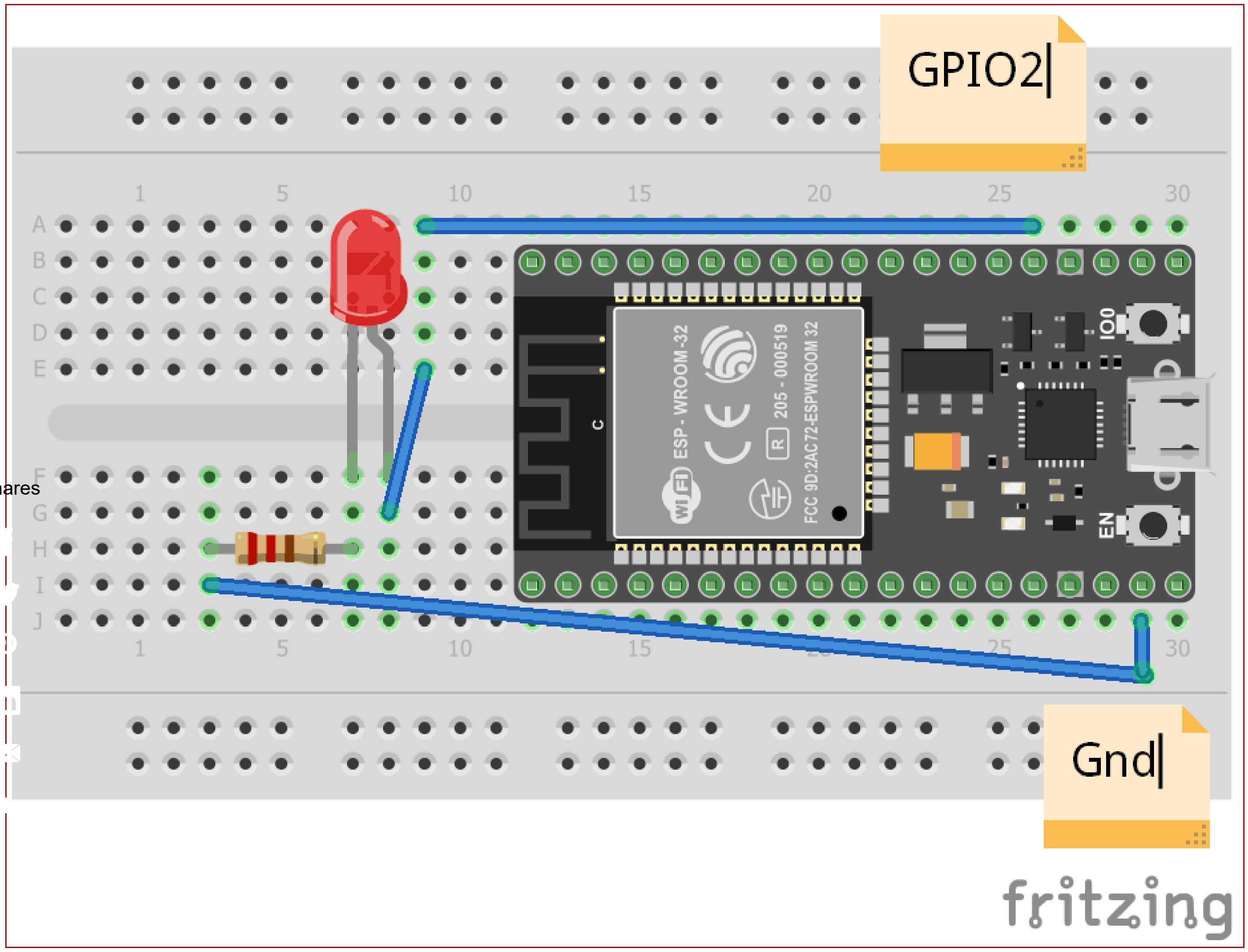
In this article we will see how we can **control LED with Goggle firebase console and ESP32**.
Firebase is Google's database platform which is used to create, manage and modify data generated from any android application, web services, sensors etc. Its basically a mobile and web application development platform which has many services like Firebase cloud messaging, Firebase auth, Realtime database, etc. In realtime database we can see realtime data on firebase cloud and can control any peripheral from anywhere using Internet.
To control LED from Google firebase first we will setup Google firebase console and then ESP32 module.

Shares

Requirements

1. ESP32 module
2. USB Cable
3. Breadboard
4. LED
5. Jumper wires
6. Resistor 1K

Circuit Diagram

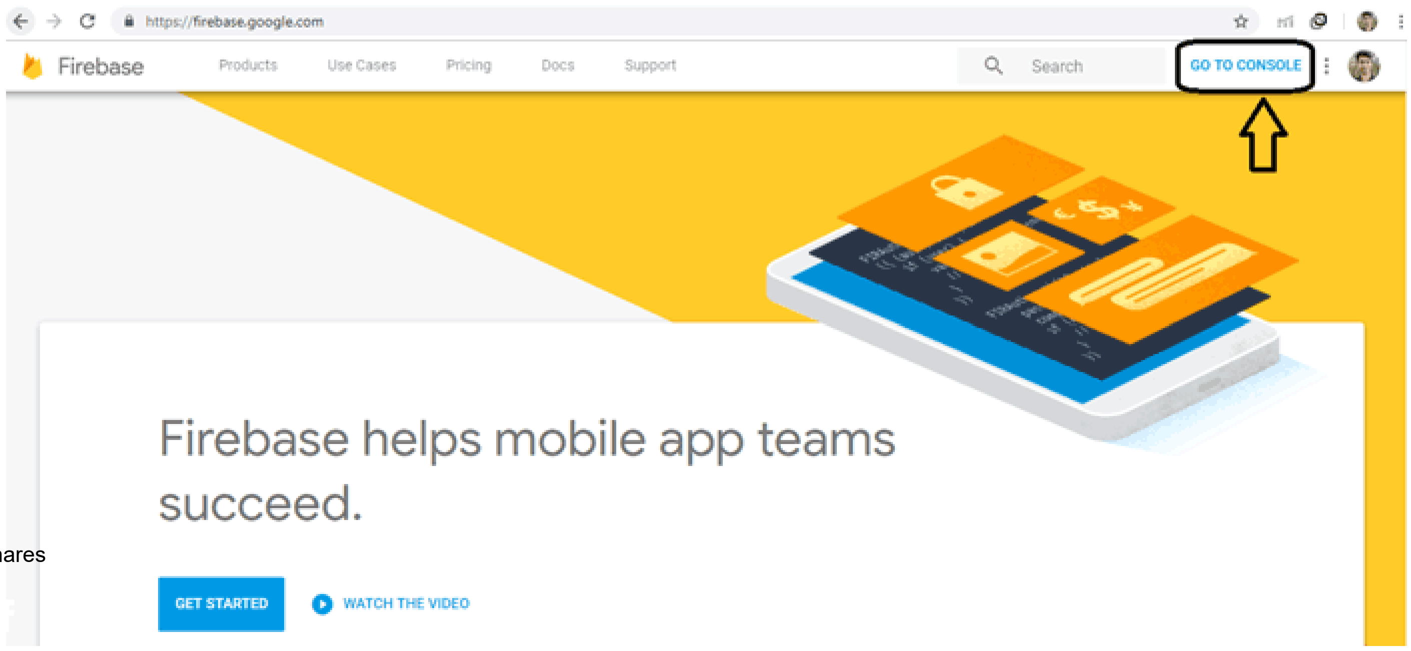


Shares

Setting up Firebase Console for ESP32

If you are using firebase for the first time then you have to create account for firebase or can directly signup using Google Account:

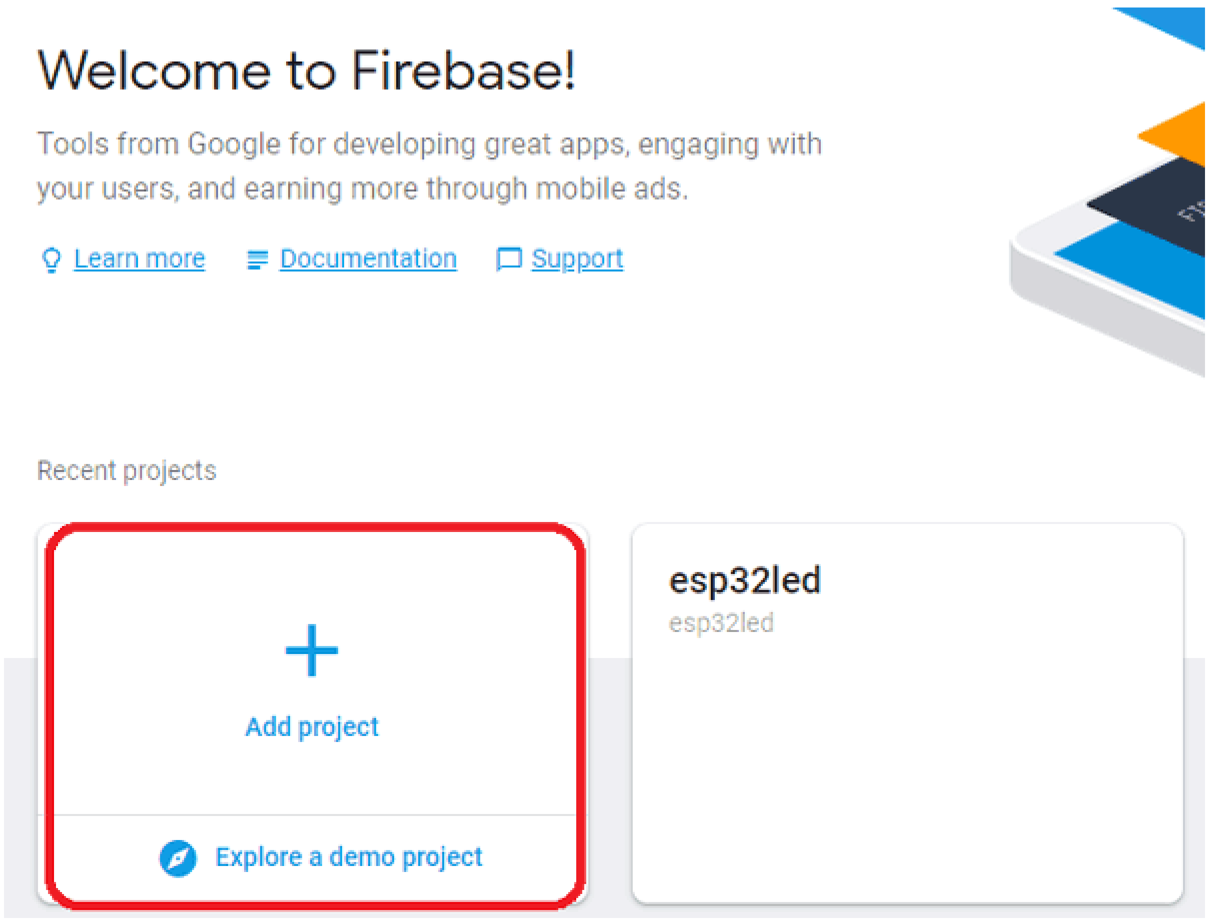
1. Open your browser and go for https://firebase.google.com (https://firebase.google.com)
2. At the right top corner click on **"Go to Console".**



Firebase helps mobile app teams succeed.

Shares

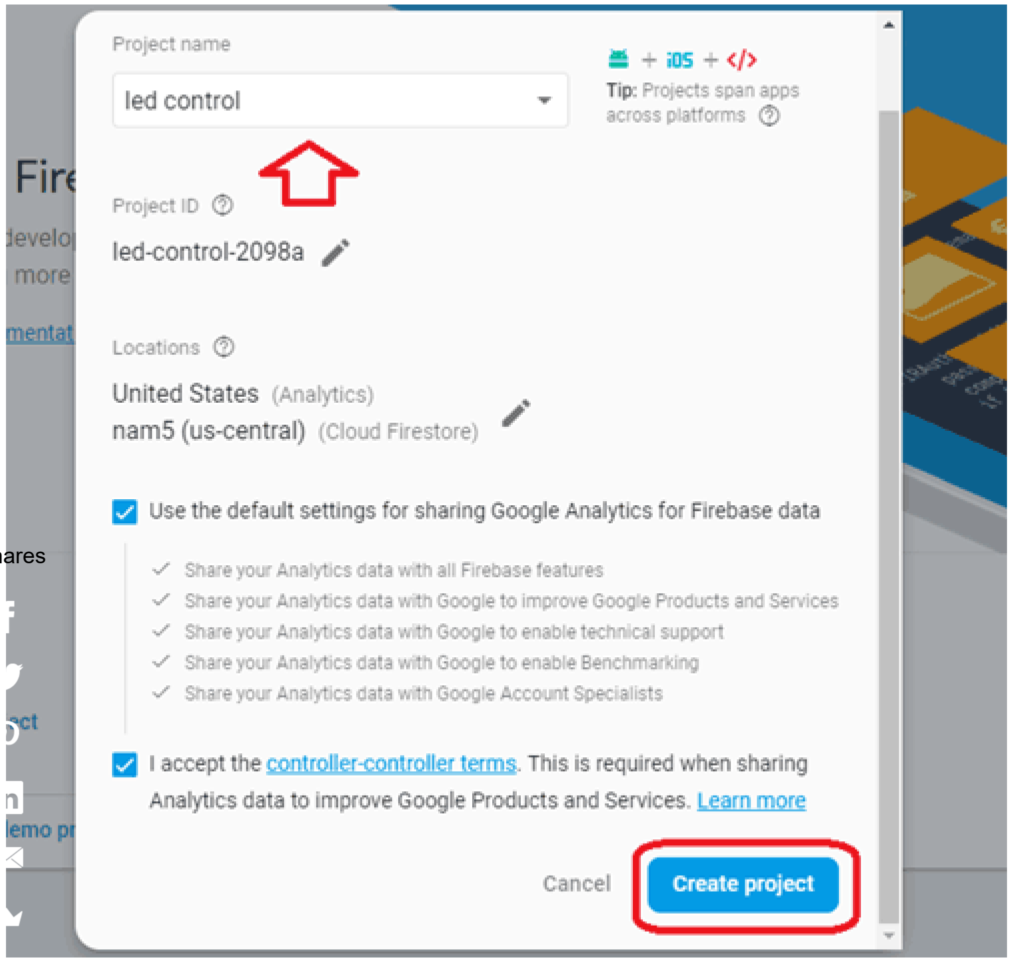
3. Click on "**Add Project**".



Welcome to Firebase!

Tools from Google for developing great apps, engaging with your users, and earning more through mobile ads.

4. Input your project name as you want and click on create project.


Project name

led control

Tip: Projects span apps across platforms

Project ID

led-control-2098a

Locations

United States (Analytics)
nam5 (us-central) (Cloud Firestore)

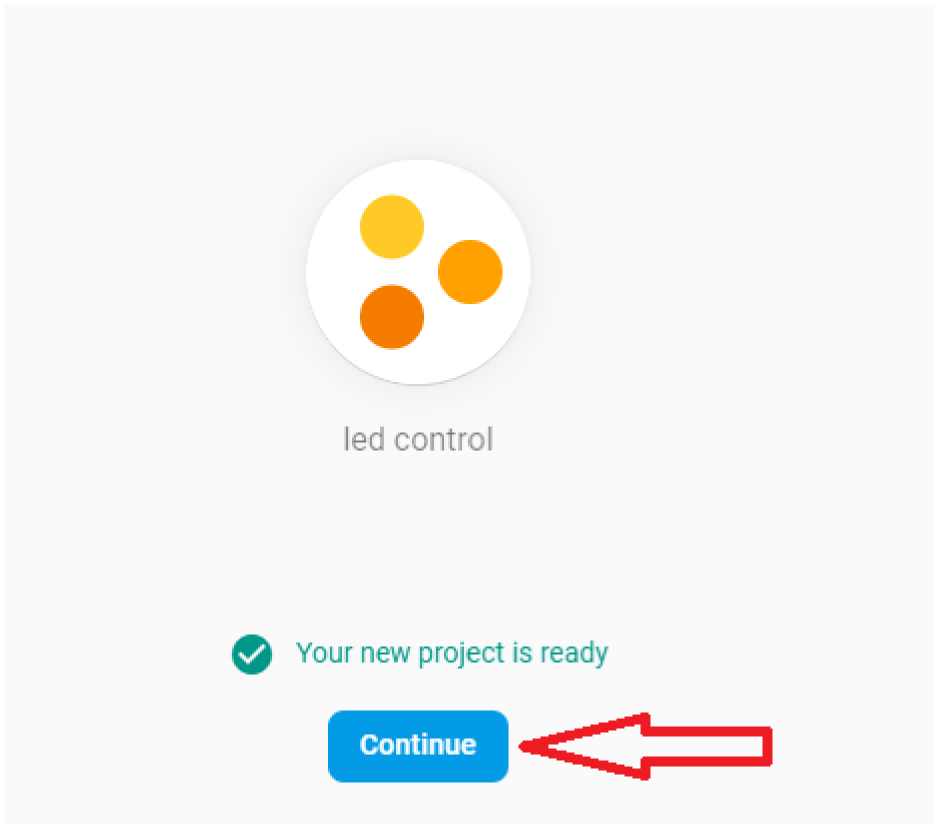
✓ Use the default settings for sharing Google Analytics for Firebase data

- ✓ Share your Analytics data with all Firebase features
- ✓ Share your Analytics data with Google to improve Google Products and Services
- ✓ Share your Analytics data with Google to enable technical support
- ✓ Share your Analytics data with Google to enable Benchmarking
- ✓ Share your Analytics data with Google Account Specialists

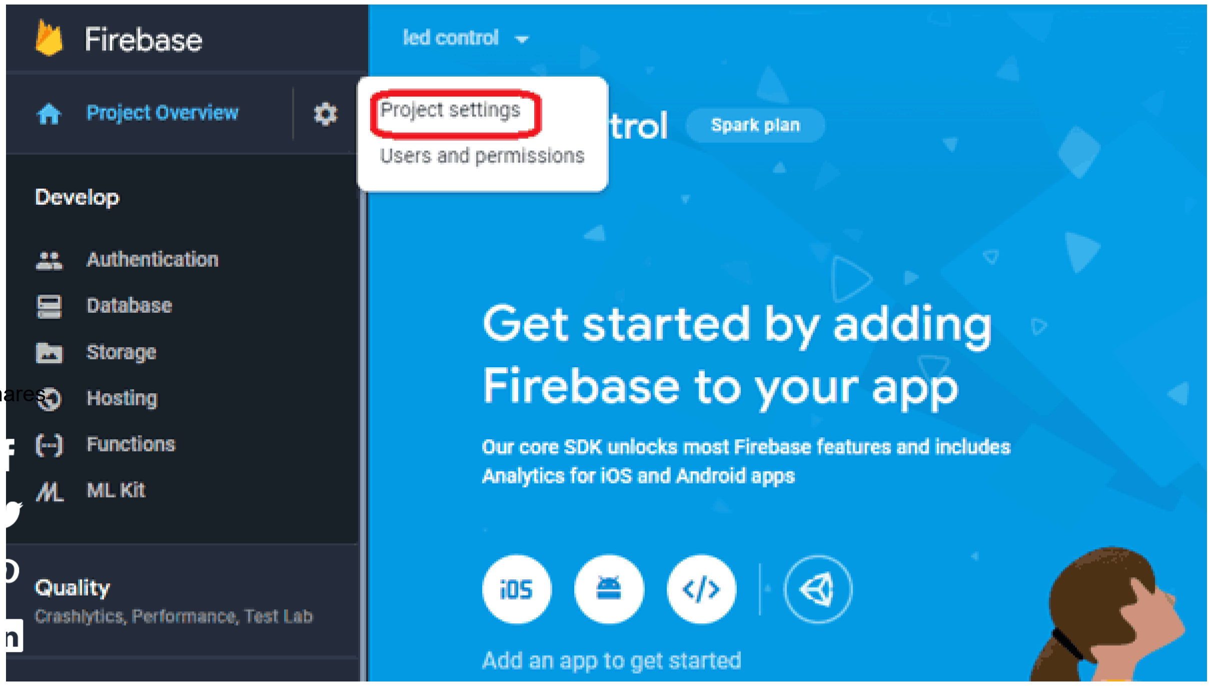
✓ I accept the controller-controller terms. This is required when sharing Analytics data to improve Google Products and Services. Learn more

Shares

5. Now your project is created and click on “**Continue**”.

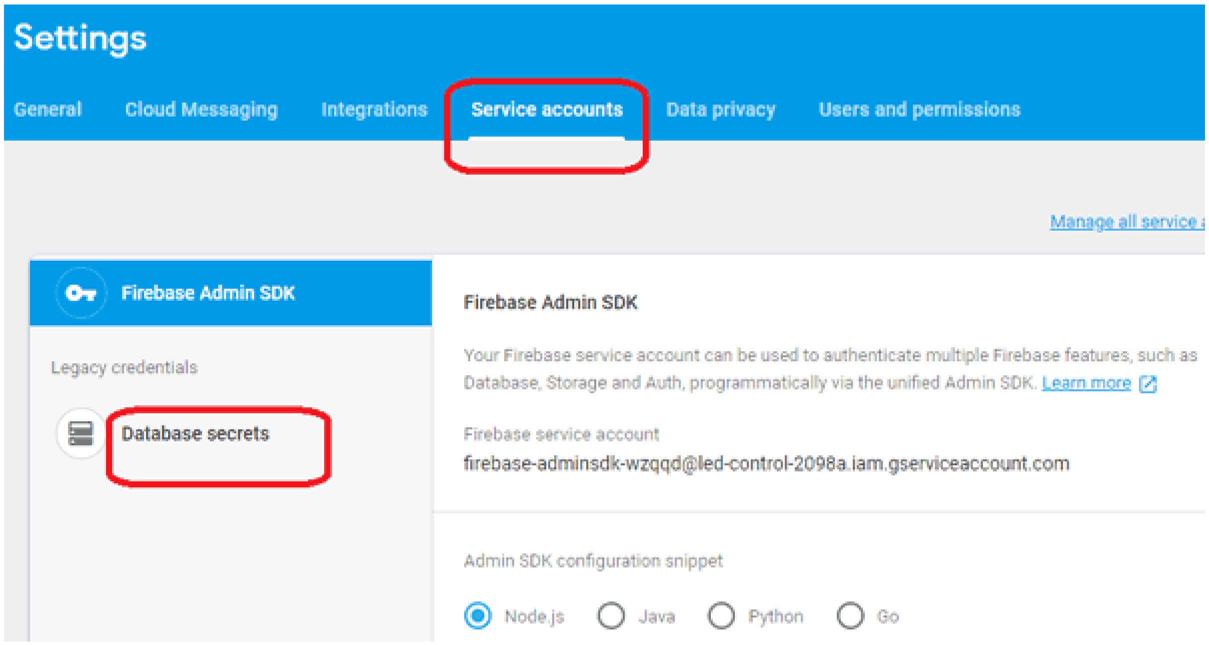


6. Now you will need host name and authorization key/secret key for this project while programming your ESP32; so now we will see how these parameters can be taken from this.
7. Go to setting icon and click on "**Project Setting**".


Firebase
led control
Project Overview
Project settings
Users and permissions
trol
Spark plan
Develop
Authentication
Database
Storage
Hosting
Functions
ML Kit
Quality
Crashlytics, Performance, Test Lab
Get started by adding Firebase to your app
Our core SDK unlocks most Firebase features and includes Analytics for iOS and Android apps
iOS
Add an app to get started

Shares

8. Now click on **Service accounts** and then **Database secrets**.

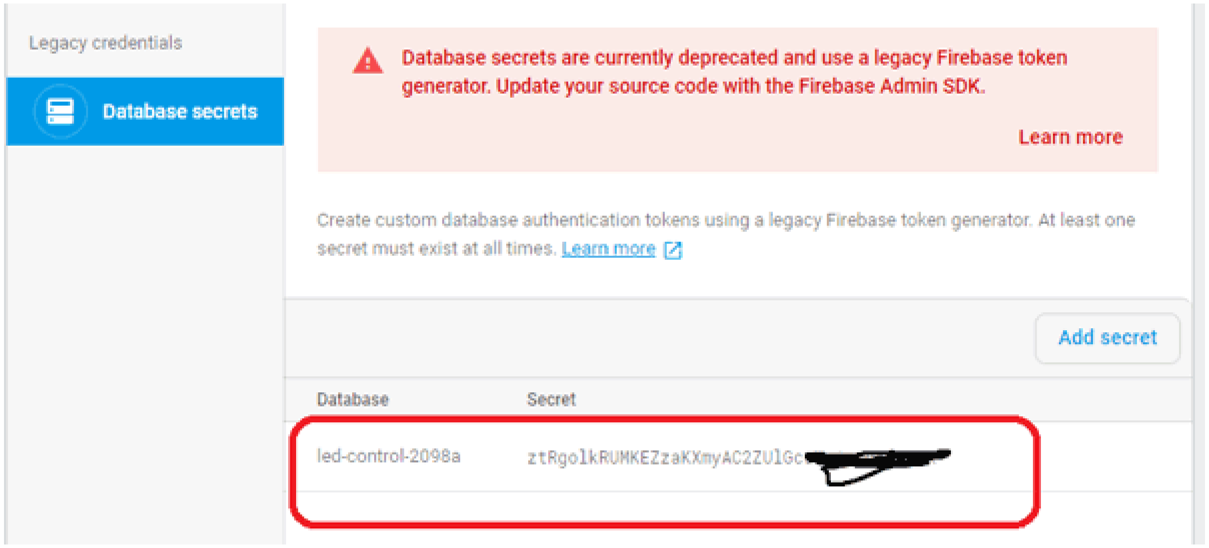


Your Firebase service account can be used to authenticate multiple Firebase features, such as Database, Storage and Auth, programmatically via the unified Admin SDK. Learn more

Firebase service account
firebase-adminsdk-wzqqd@led-control-2098a.iam.gserviceaccount.com

Admin SDK configuration snippet
[✓ Node.js] Java Python Go

9. On clicking on *Database Secrets* you will find a secret key, copy this key and save it in notepad, this is your firebase authorization key which you will need later.


Legacy credentials

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

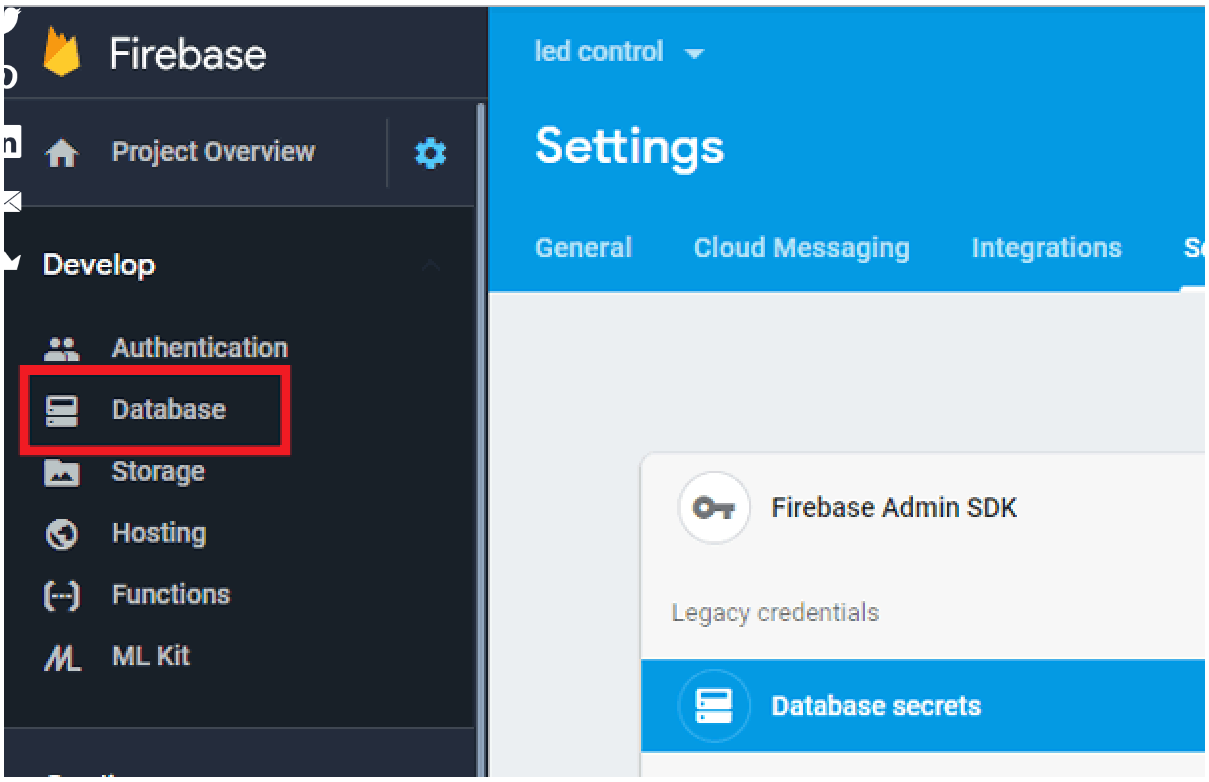
Create custom database authentication tokens using a legacy Firebase token generator. At least one secret must exist at all times. Learn more

Add secret

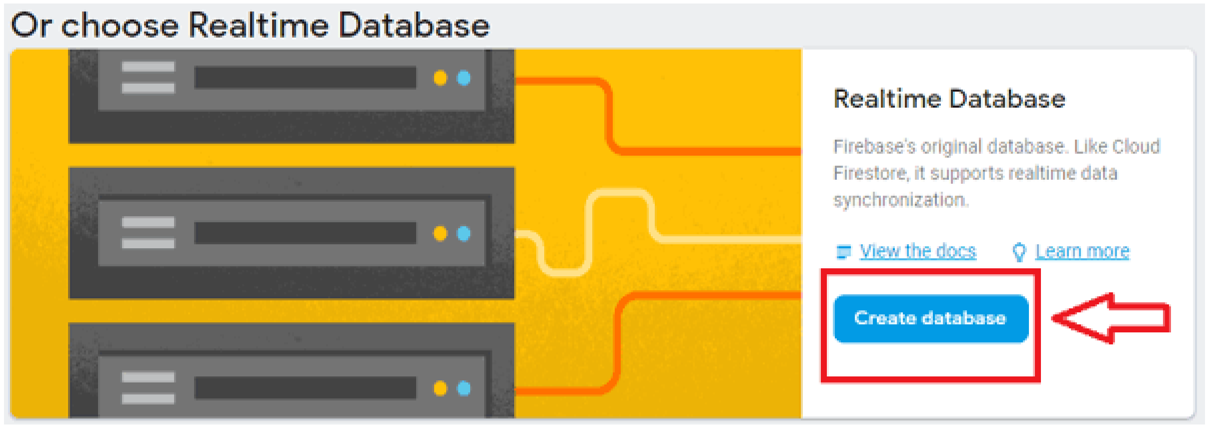
Database	Secret
led-control-2098a	ztRgolkRUMKEZzaKXmyAC2ZUlGc[redacted]

Shares

10. Now click on "**Database**" at left control bar.



11. Now scroll down and click on **"Create database".**

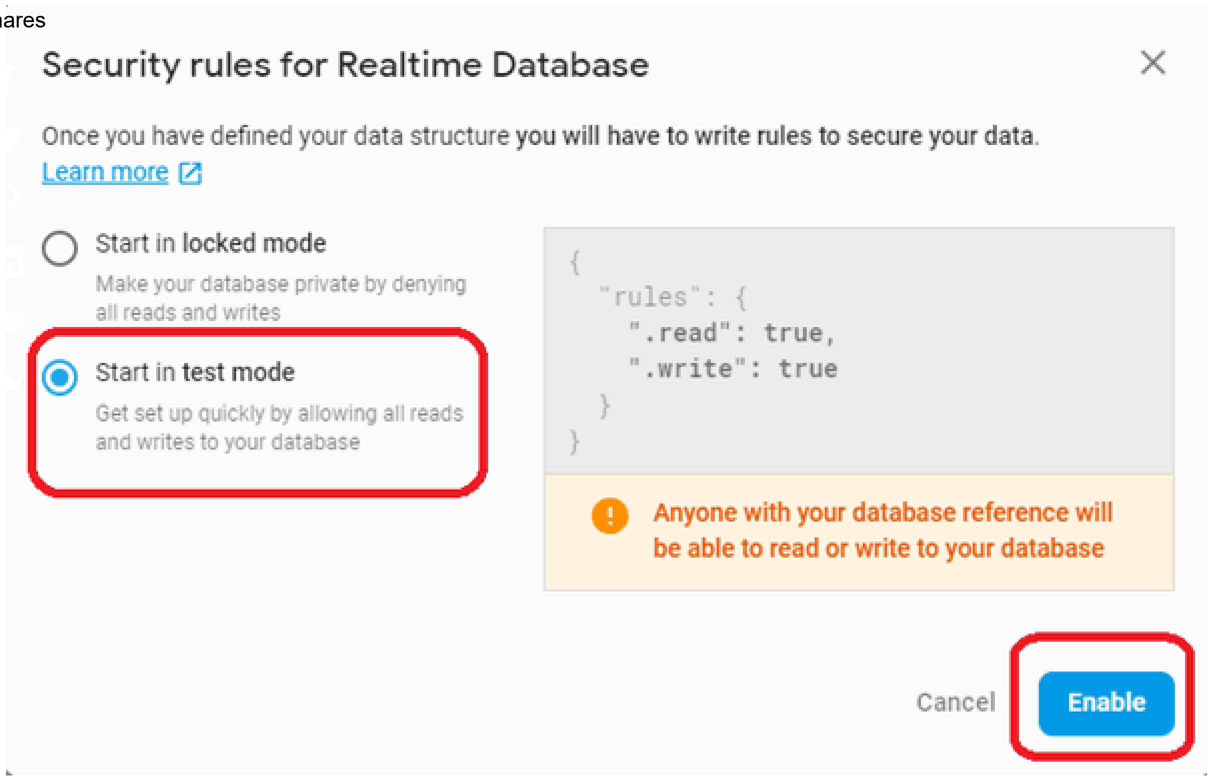

Or choose Realtime Database

Realtime Database

Firebase's original database. Like Cloud Firestore, it supports realtime data synchronization.

12. Now choose "**Start in test mode**" and click on **Enable**.

Shares


Security rules for Realtime Database

Once you have defined your data structure you will have to write rules to secure your data. Learn more

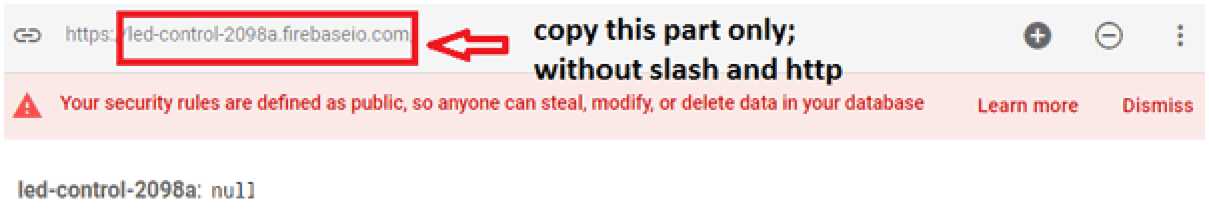
Start in locked mode
Make your database private by denying all reads and writes

Start in test mode
Get set up quickly by allowing all reads and writes to your database

```
{
  "rules": {
    ".read": true,
    ".write": true
  }
}
```

Anyone with your database reference will be able to read or write to your database

13. Now your database is created and you will have to come here again to control your LED, for now just copy the given URL without slash and http in notepad this is your **firebase host** which you will be required later.


https://led-control-2098a.firebaseio.com/

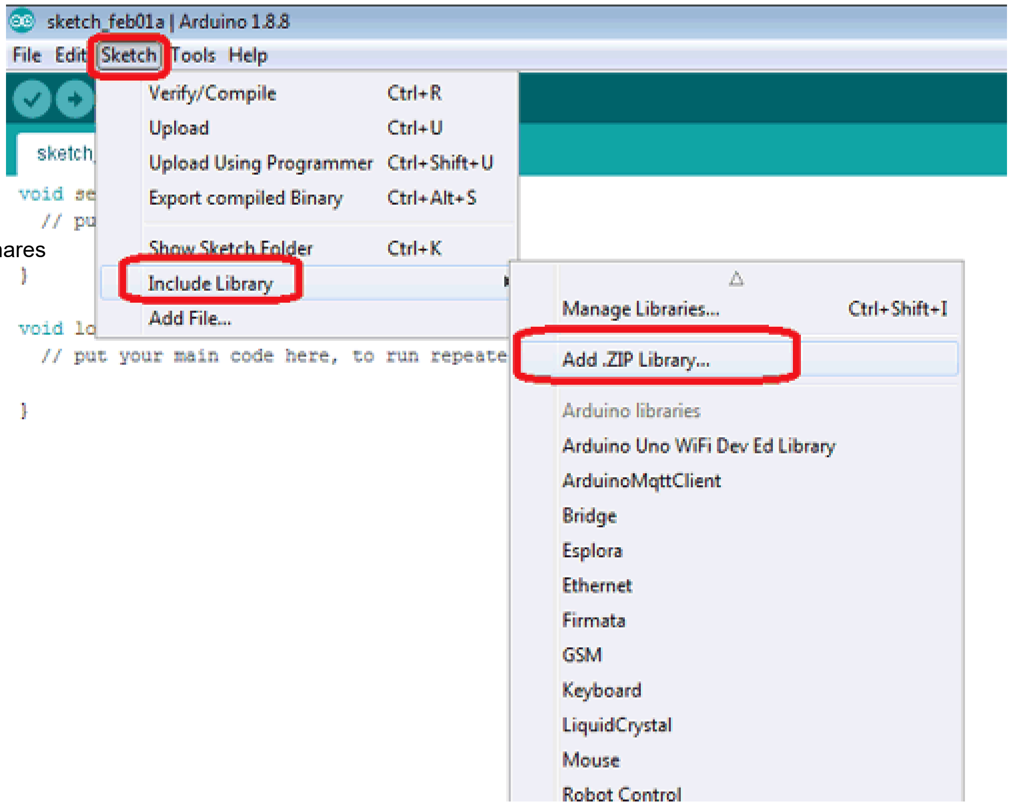
copy this part only; without slash and http

Your security rules are defined as public, so anyone can steal, modify, or delete data in your database

led-control-2098a: null

Setting up ESP32 module with Firebase

To work with **ESP32 using Google firebase** you will need a firebase library so firstly download that library using below link and save it in Arduino library files.
https://github.com/ioxhop/IOXhop_FirebaseESP32 (https://github.com/ioxhop/IOXhop_FirebaseESP32)
Now open your Arduino IDE and go to *Sketch---> include Library--> Add .ZIP library* and add the file you downloaded from the above link.


sketch_feb01a | Arduino 1.8.8
File Edit Sketch Tools Help
sketch
Verify/Compile Ctrl+R
Upload Ctrl+U
Upload Using Programmer Ctrl+Shift+U
Export compiled Binary Ctrl+Alt+S
Show Sketch Folder Ctrl+K
Include Library
Add File...
void se
// pu
}
void lo
// put your main code here, to run repeate
}
Manage Libraries... Ctrl+Shift+I
Add .ZIP Library...
Arduino libraries
Arduino Uno WiFi Dev Ed Library
ArduinoMqttClient
Bridge
Esplora
Ethernet
Firmata
GSM
Keyboard
LiquidCrystal
Mouse
Robot Control

Shares

After installing the library you are ready to work with Google firebase using ESP32.
Now go to Tools and select ESP32 Dev board and appropriate COM port and copy the code given below and edit it for network credentials, firebase secret key and firebase host. After editing the upload the code into ESP32 using Arduino IDE.

Programming ESP32 for Google Firebase

Complete code for controlling LED using ESP32 is given at the end of the tutorial
Firstly include the libraries for using ESP32 and firebase.

```
#include <WiFi.h>
#include <IOXhop_FirebaseESP32.h>
```

Now enter your firebase host, secret key and network credentials.

```
#define FIREBASE_HOST "esp32led.firebaseio.com"
#define FIREBASE_AUTH "4QHdeFZquTh4fdXZkum2EPt2A50gXXXXXXXXX"
#define WIFI_SSID "XXXXXX"
#define WIFI_PASSWORD "XXXXXXXXXX"
```

In the *setup* function, define output pin, delay, baud rate and connect to your Wi-Fi.

Shares

```
void setup() {
  Serial.begin(9600);
  delay(1000);
  pinMode(2, OUTPUT);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to ");
  Serial.print(WIFI_SSID);
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
```

The below statement tries to connect with the firebase server. If the host address and authorization key are correct then it will connect successfully.

```
Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
```

Now this is the class provided by firebase library to send string to firebase server. With the help of this we can change the status of LED.

```
Firebase.setString("LED_STATUS", "OFF");
```

After sending one status string to firebase path, write this statement to get the status of LED from same path and save it to variable.

```
fireStatus = Firebase.getString("LED_STATUS");
```

If received string is “ON” or “on” then just turn on the output LED.

```
if (fireStatus == "ON")
 {
Serial.println("Led Turned ON");
digitalWrite(2, HIGH);
 }
```

If received string is “OFF” or “off” then just turn off the output LED.

```
else if (fireStatus == "OFF") {
 Serial.println("Led Turned OFF");
 digitalWrite(2, LOW);
}
```

Shares

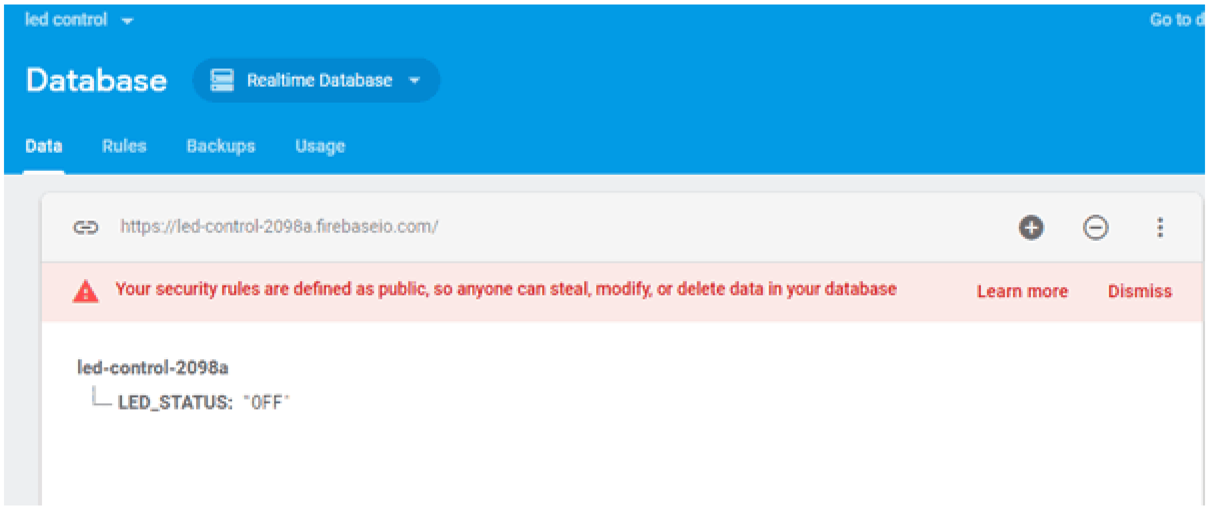
If received string is not any of these then just ignore and print some error message.

```
else {
Serial.println("Wrong Credential! Please send ON/OFF");
  }
```

The complete code is given at the end of this article, you can check from there, edit it and then upload it.

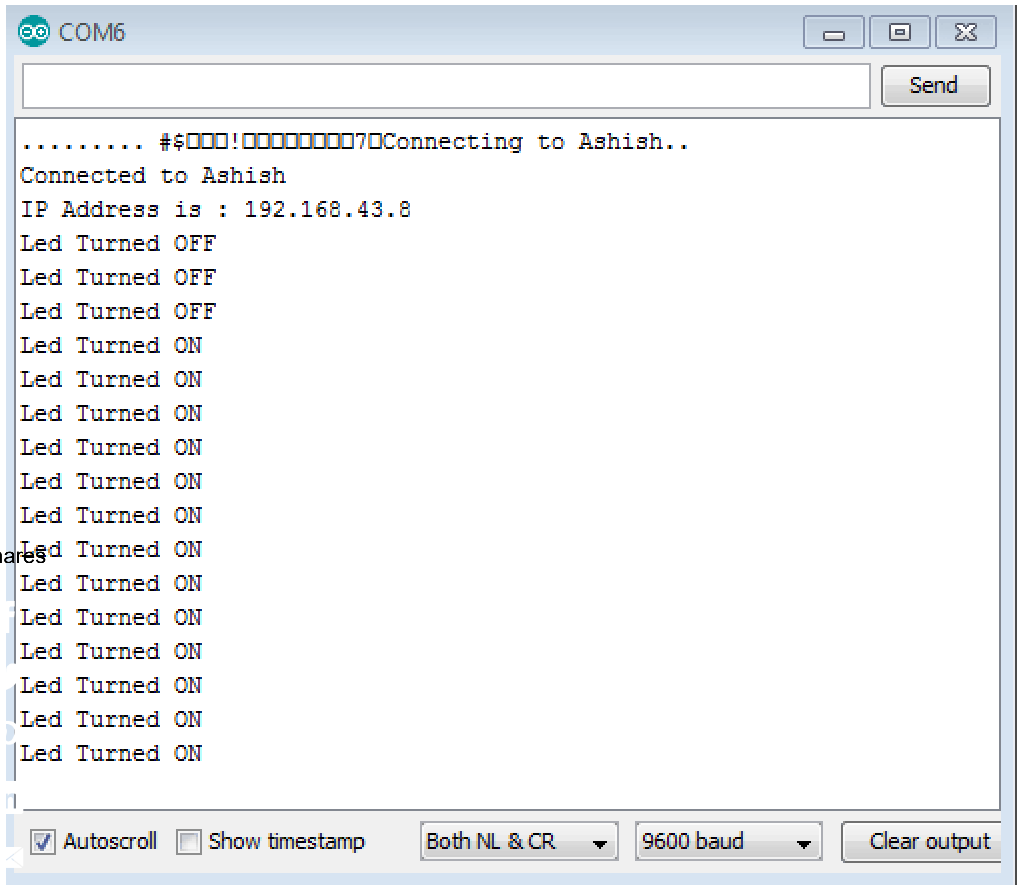
Output

After uploading the code open your serial monitor and in your browser open firebase and go to console and then select your project that you created earlier. In this go to database and it will show initially LED STATUS is OFF, from here you can change the LED status by writing ON here, your LED will change to ON state and you can see your LED state in your serial monitor also.



Your security rules are defined as public, so anyone can steal, modify, or delete data in your database

```
led-control-2098a
└ LED_STATUS: "OFF"
```



```
......... #$□□□!□□□□□□□□7□Connecting to Ashish..
Connected to Ashish
IP Address is : 192.168.43.8
Led Turned OFF
Led Turned OFF
Led Turned OFF
Led Turned ON
Led Turned ON
Led Turned ON
Led Turned ON
Led Turned ON
Led Turned ON
Led Turned ON
Led Turned ON
Led Turned ON
Led Turned ON
Led Turned ON
Led Turned ON
Led Turned ON
```

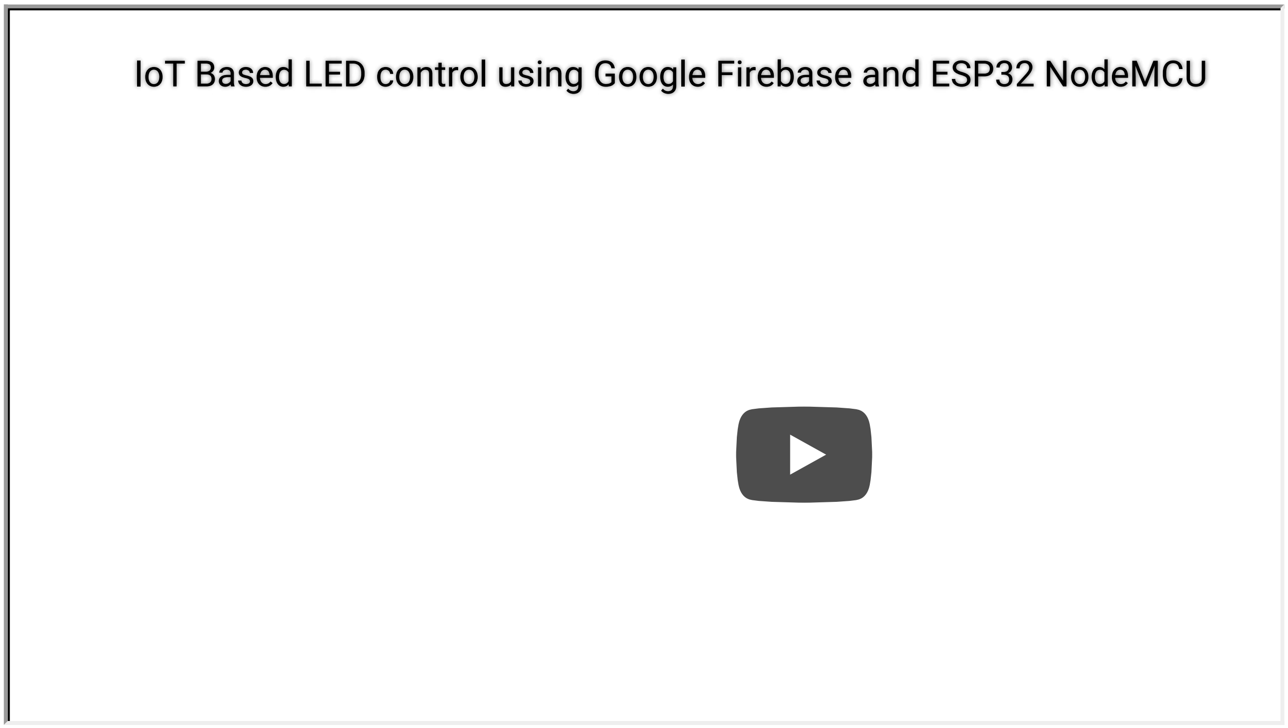
Shares

Code

```
#include <WiFi.h>                                    // esp32 library
#include <IOXhop_FirebaseESP32.h>                                        // firebase library
#define FIREBASE_HOST "XXXXXX "                    // the project name address from firebase id
#define FIREBASE_AUTH "ztRgolkRUMKEZzaKXmyAC2ZUlGcuWojXXXXXXXXX"                  // the secret key
generated from firebase
#define WIFI_SSID "Ashish"                         // input your home or public wifi name
#define WIFI_PASSWORD "XXXXXXXX"                               //password of wifi ssid
String fireStatus = "";                            // led status received from firebase
int led = 2;
void setup() {
  Serial.begin(9600);
  delay(1000);
  pinMode(2, OUTPUT);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);                                 //try to connect with wifi
  Serial.print("Connecting to ");
  Serial.print(WIFI_SSID);
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.print("Connected to ");
  Serial.println(WIFI_SSID);
  Serial.print("IP Address is : ");
  Serial.println(WiFi.localIP());                               //print local IP address
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);                          // connect to firebase
  Firebase.setString("LED_STATUS", "OFF");                          //send initial string of led status
}
void loop() {
  fireStatus = Firebase.getString("LED_STATUS");                  // get led status input from firebase
  if (fireStatus == "ON") {                      // compare the input of led status received from firebase
    Serial.println("Led Turned ON");
    digitalWrite(2, HIGH);                                      // make output led ON
  }
  else if (fireStatus == "OFF") {            // compare the input of led status received from firebase
    Serial.println("Led Turned OFF");
    digitalWrite(2, LOW);                                       // make output led OFF
  }
  else {
    Serial.println("Wrong Credential! Please send ON/OFF");
  }
}
```

Shares

Video



Tags

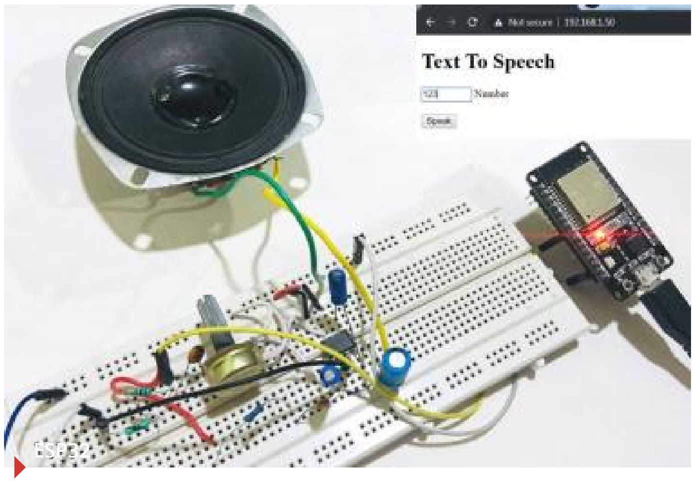
ESP32 (/TAGS/ESP32) **ESP32 IOT (/TAGS/ESP32-IOT)** **IOT LED (/TAGS/IOT-LED)** **GOOGLE FIREBASE (/TAGS/GOOGLE-FIREBASE)**

RELATED ARTICLES

Shares

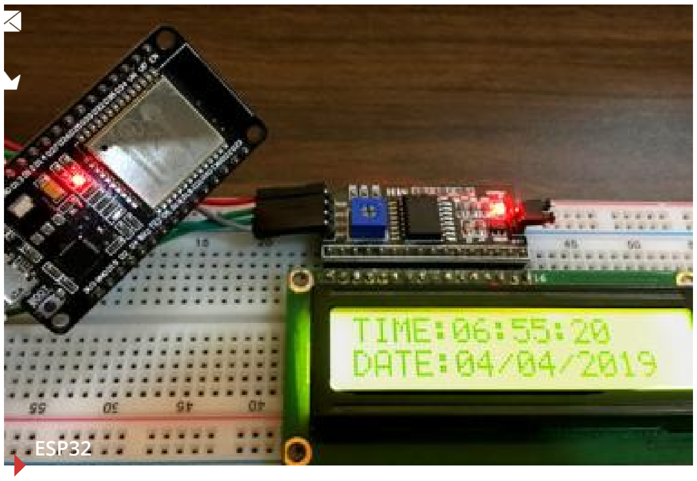
• 22 Jul, 2019

How to connect ESP32 with IBM Watson Cloud Platform (/projects/how-to-connect-esp32-with-ibm-watson-cloud-platform)



• 09 Jul, 2019

ESP32 based Webserver for Text to Speech (TTS) Conversion (/projects/esp32-based-text-to-speech-converter-webserver)



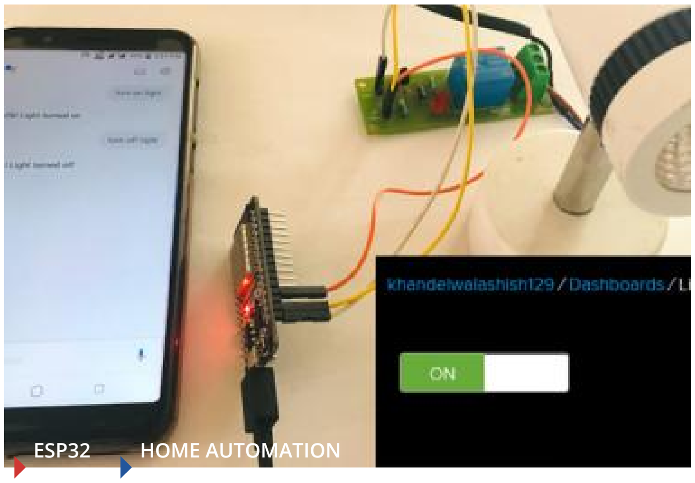
• 14 May, 2019

Internet Clock using 16x2 LCD and ESP32 (/projects/internet-clock-using-16x2-lcd-and-esp32)

```
--------new message from broker-----
channel:/icircuit/ESP32/serialdata/rx
data:Hello world
```

• 09 May, 2019

How to Connect ESP32 to MQTT Broker (/projects/how-to-connect-esp32-mqtt-broker)



• 07 May, 2019

Shares

IoT Controlled LED using Cayenne and ESP32 (/projects/iot-controlled-led-using-cayenne-and-esp32)

• 11 Feb, 2019

Google Assistant Based Home Appliance Control using ESP32 and Adafruit IO (/projects/google-assistant-based-home-appliance-control-using-esp32-and-adafruit-io)

Shares

INFORMATION

- Contact Us (https://iotdesignpro.com/contact)
- Cookie Policy (https://iotdesignpro.com/cookie-policy)
- Privacy Policy (https://iotdesignpro.com/privacy-policy)

NEWSLETTER SUBSCRIPTION

Stay Informed - Subscribe to our Newsletter.

SUBSCRIBE