INTRODUCTION TO EMBEDDED SYSTEMS

Instructors

Wayne Okello

wayne.okello@netlabsug.org

David Kateeba

katsdavid72@gmail.com

netLabs!UG

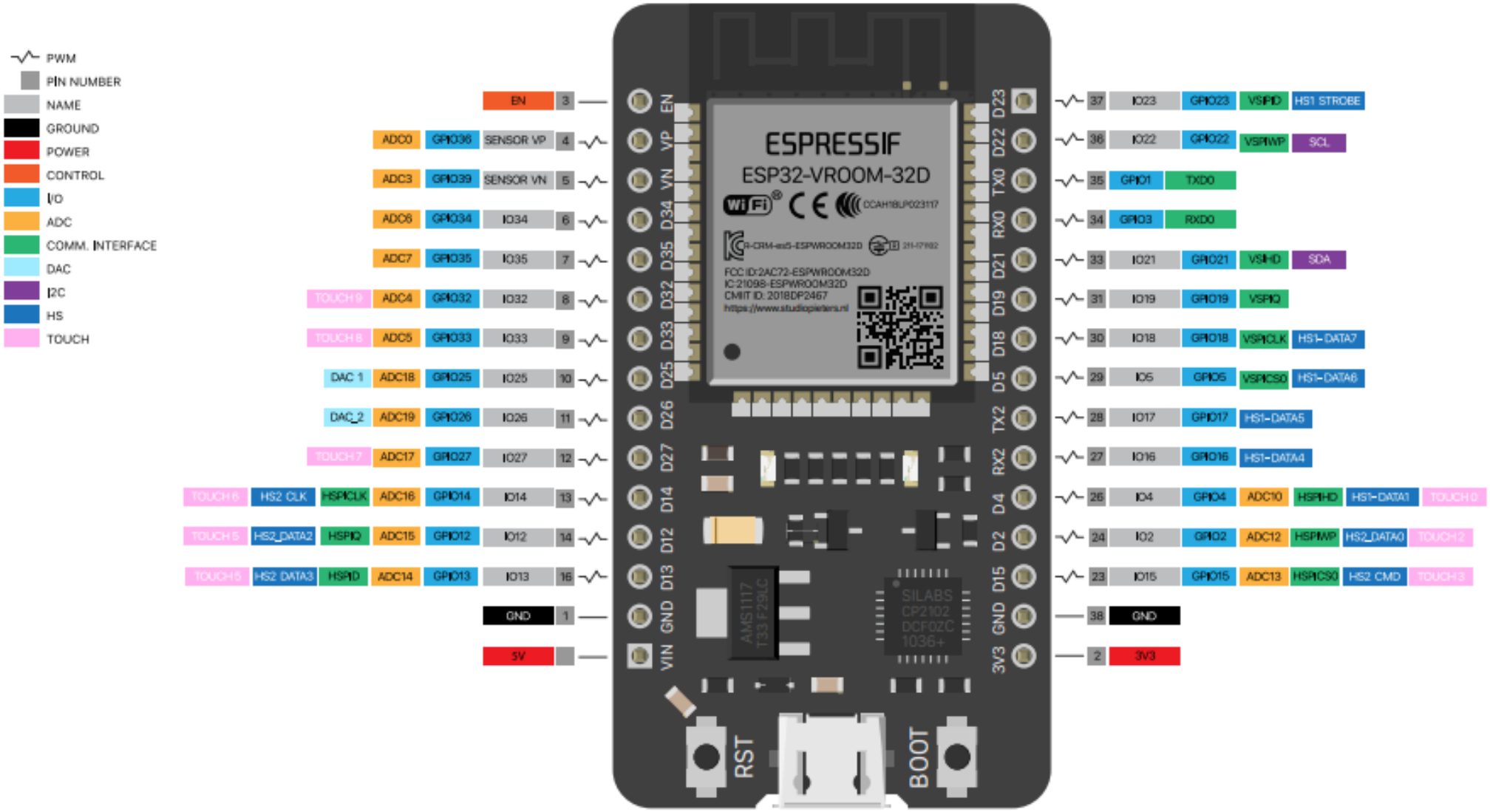
ESP32 Development Board

- It is a feature-rich MCU Board with integrated Wi-Fi and Bluetooth connectivity for a wide-range of applications.
- It is compatible with Arduino IDE.

Why use ESP32?

- It has inbuilt Bluetooth
- It has inbuilt Wi-Fi
- It has many pins
- It consumers less power compared to other MCU boards

Esp32 30-Pin layout



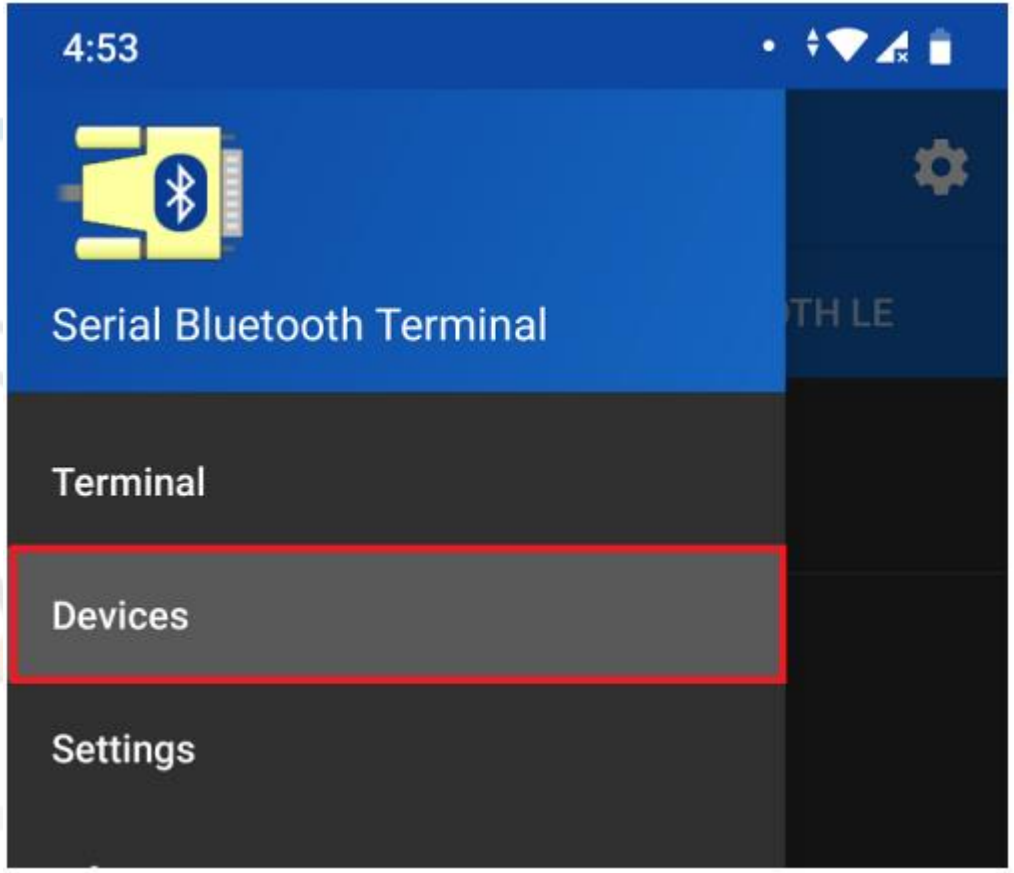
Esp32 Bluetooth

```
#include "BluetoothSerial.h"
#if !defined(CONFIG_BT_ENABLED) || !defined(CONFIG_BLUEDROID_ENABLED)
#error Bluetooth is not enabled! Please run `make menuconfig` to and enable it
#endif
BluetoothSerial SerialBT;

void setup() {
  Serial.begin(115200);
  SerialBT.begin("ESP32test"); //Bluetooth device name
  Serial.println("The device started, now you can pair it with bluetooth!");
}
void loop() {
  if (Serial.available()) {
    SerialBT.write(Serial.read());
  }
  if (SerialBT.available()) {
    Serial.write(SerialBT.read());
  }
  delay(20);
}
```

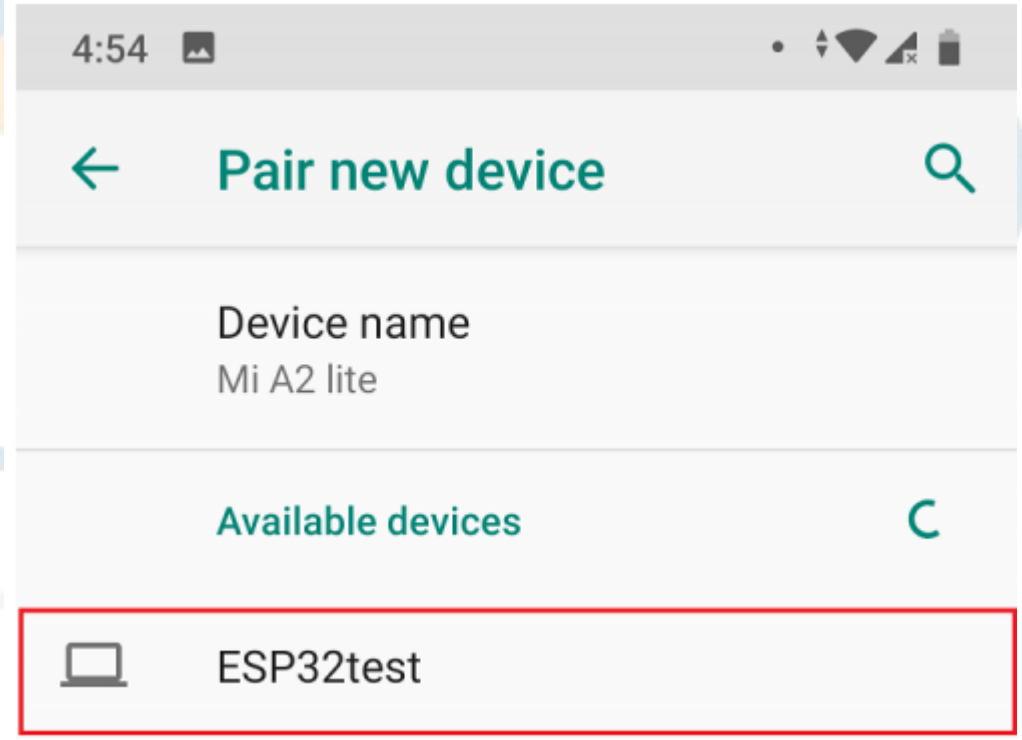
Connecting ESP 32 to a Smartphone via Bluetooth

- Download any Bluetooth Terminal application on your smartphone.
- For example “Serial Bluetooth Terminal” available in the Play Store for Android devices.
- For example “Bluetooth Terminal” available in the Apple Store for Apple devices.

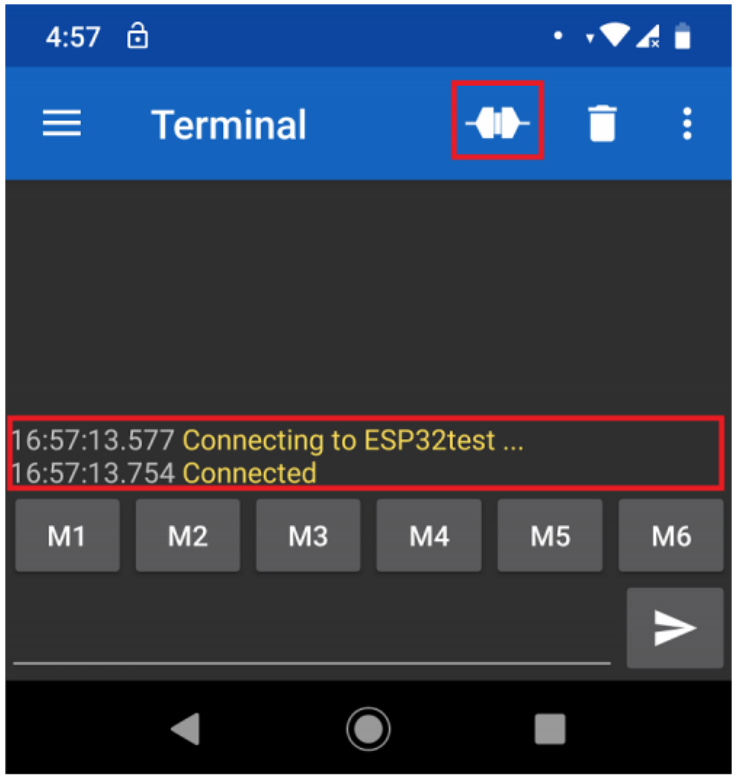
- To connect to the ESP32 for the first time, you need to pair a new device.
- Go to Devices.



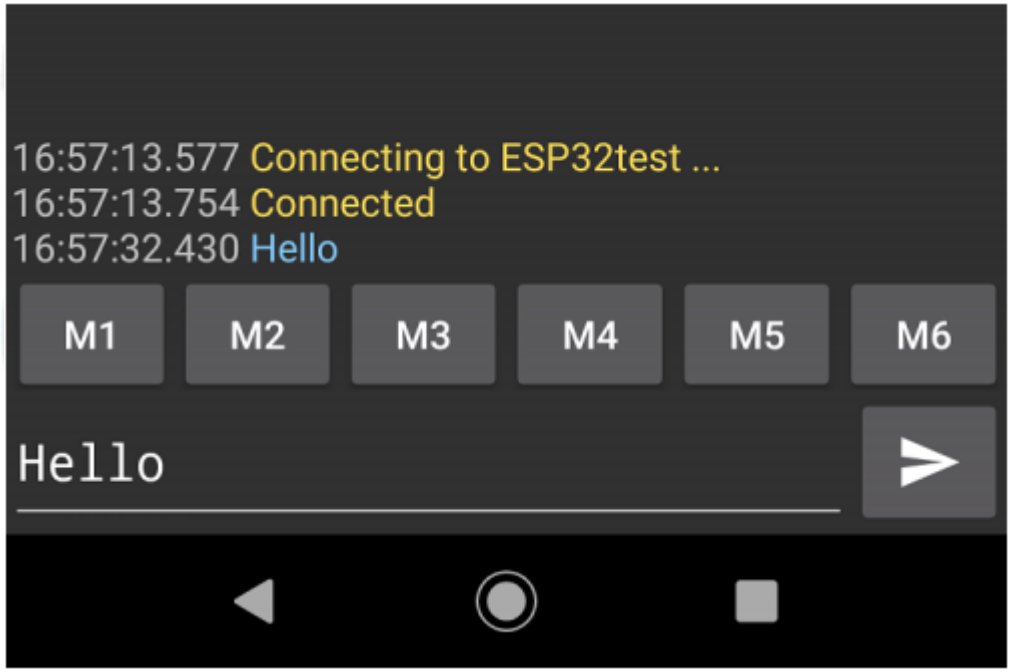
- Click the settings icon, and select Pair new device.
- You should get a list with the available Bluetooth devices, including the ESP32test. Pair with the ESP32test.



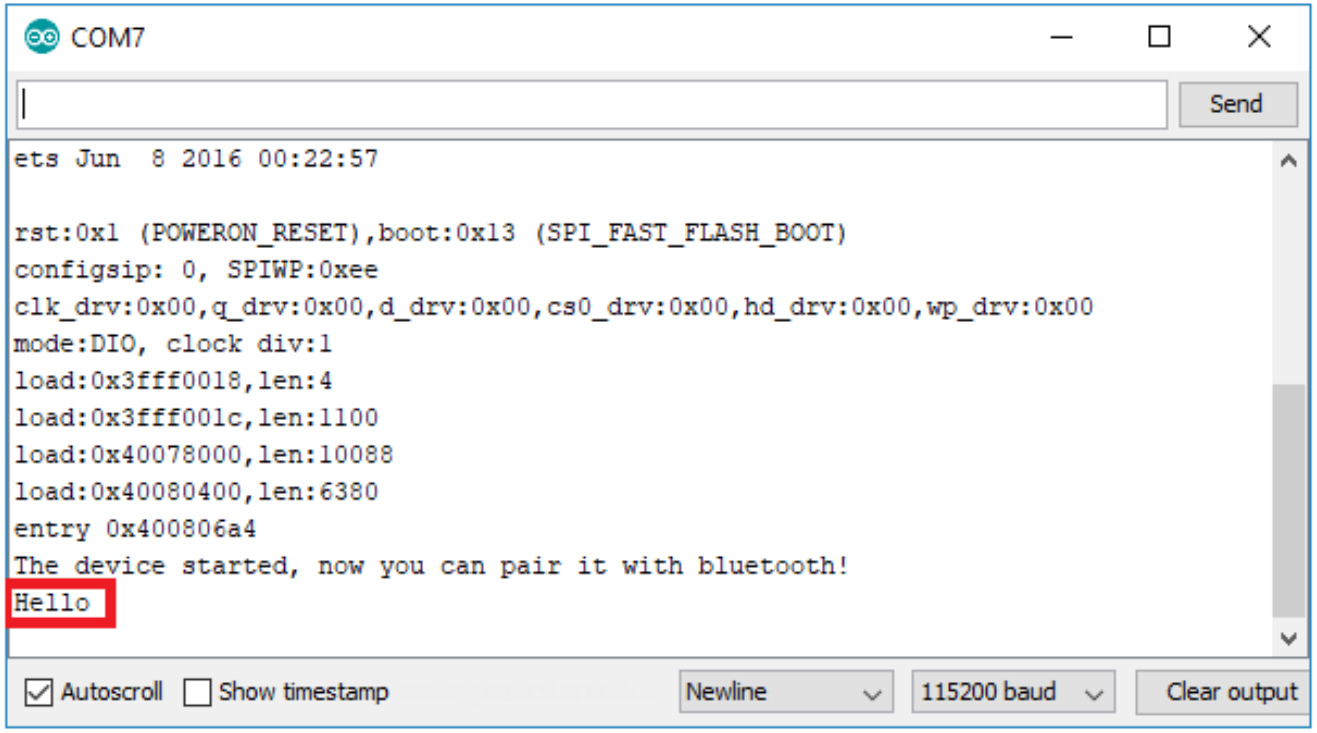
- Then, go back to the Serial Bluetooth Terminal. Click the icon at the top to connect to the ESP32.
- You should get a “Connected” message.



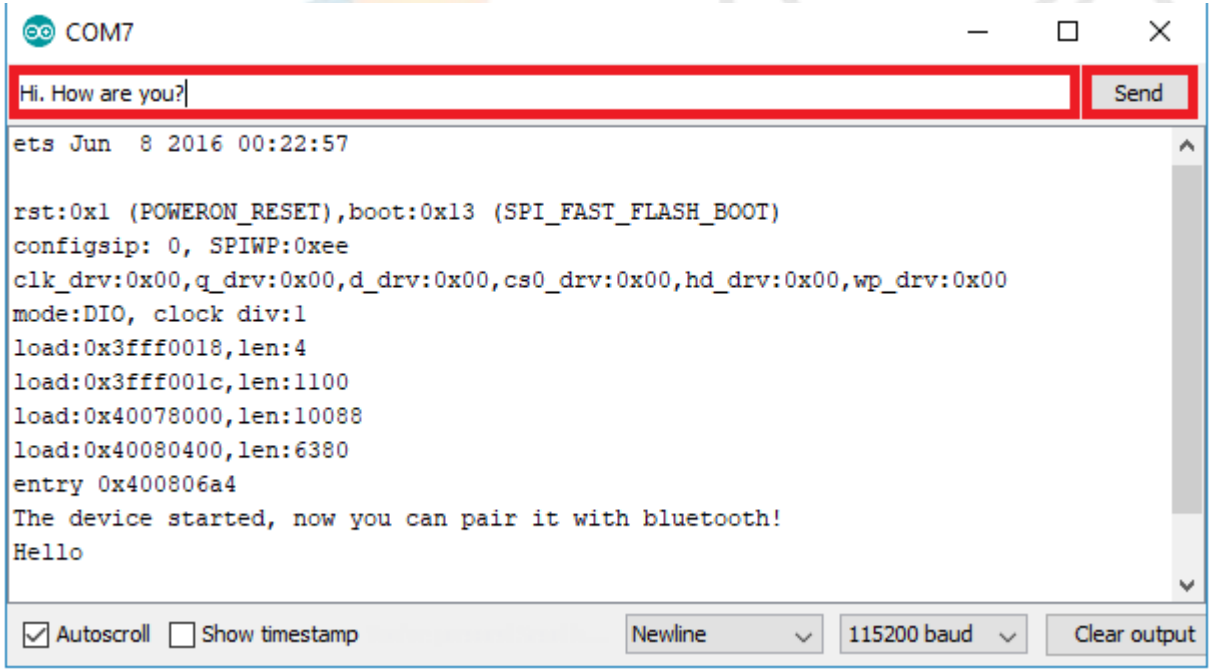
- After that, type something in the Serial Bluetooth Terminal app. For example, “Hello”.



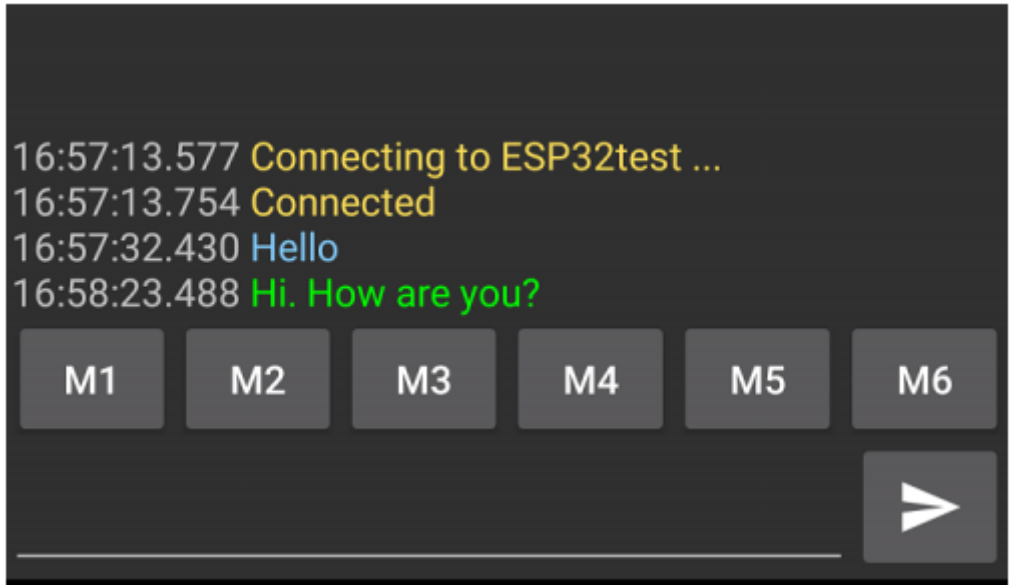
- You should instantly receive that message in the Arduino IDE Serial Monitor.



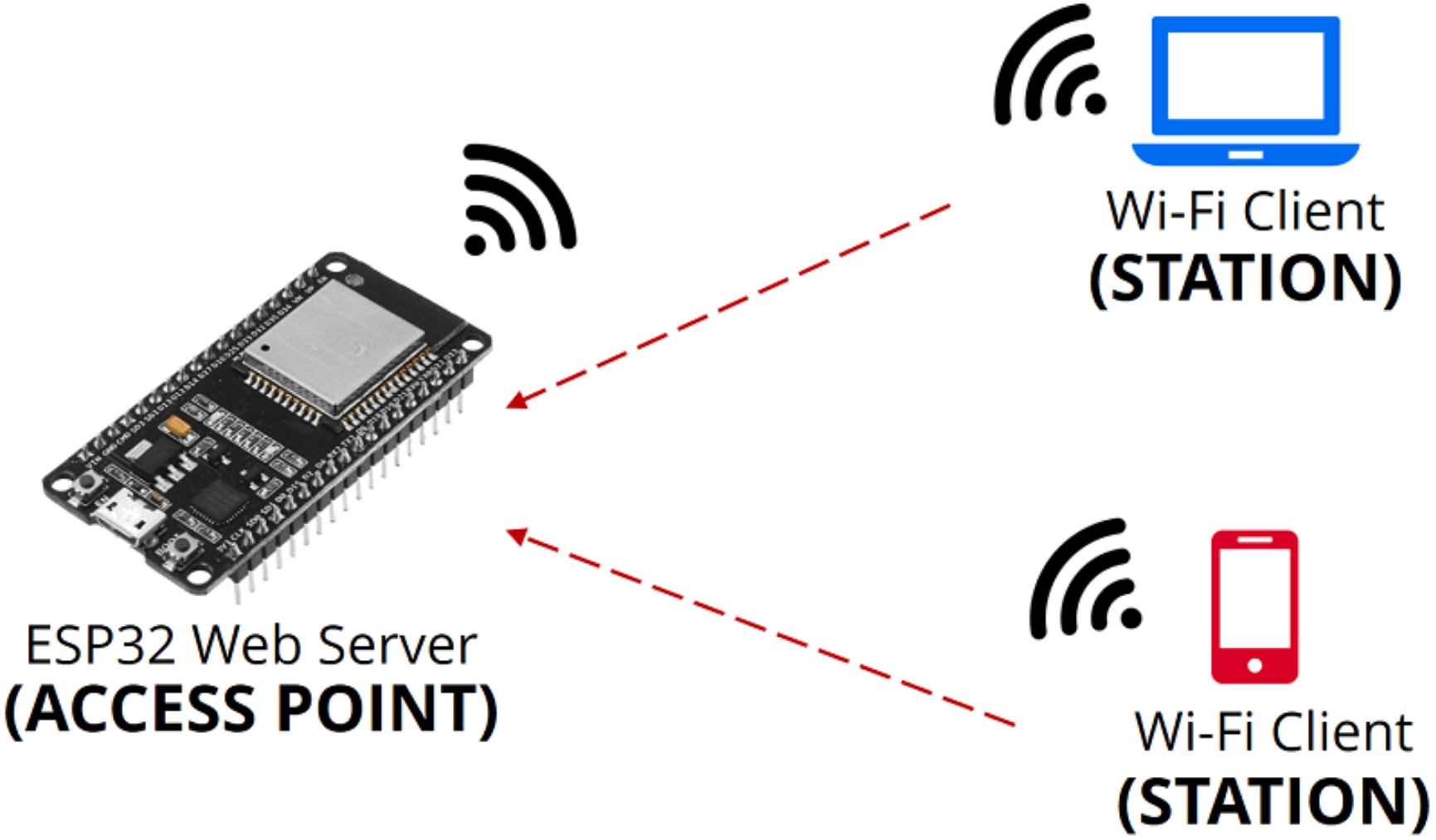
- You can also exchange data between your Serial Monitor and your smartphone.
- Type something in the Serial Monitor top bar and press the “Send” button.



- You should instantly receive that message in the Serial Bluetooth Terminal App.



Esp32 Wi-Fi Access point



Esp32 Wi-Fi Connections Access point

```
#include <WiFi.h>

const char* ssid  = “ESP32          // SSID Name
const char* password = “jrs123";   // SSID Password - Set to NULL to have an open AP

void setup()
{
    Serial.begin(115200);
    Serial.println("Creating AP");
    WiFi.mode(WIFI_AP);
    WiFi.softAP(ssid, password);
    Serial.print("AP Created with IP Gateway ");
    Serial.println(WiFi.softAPIP());
}

void loop(){}
```

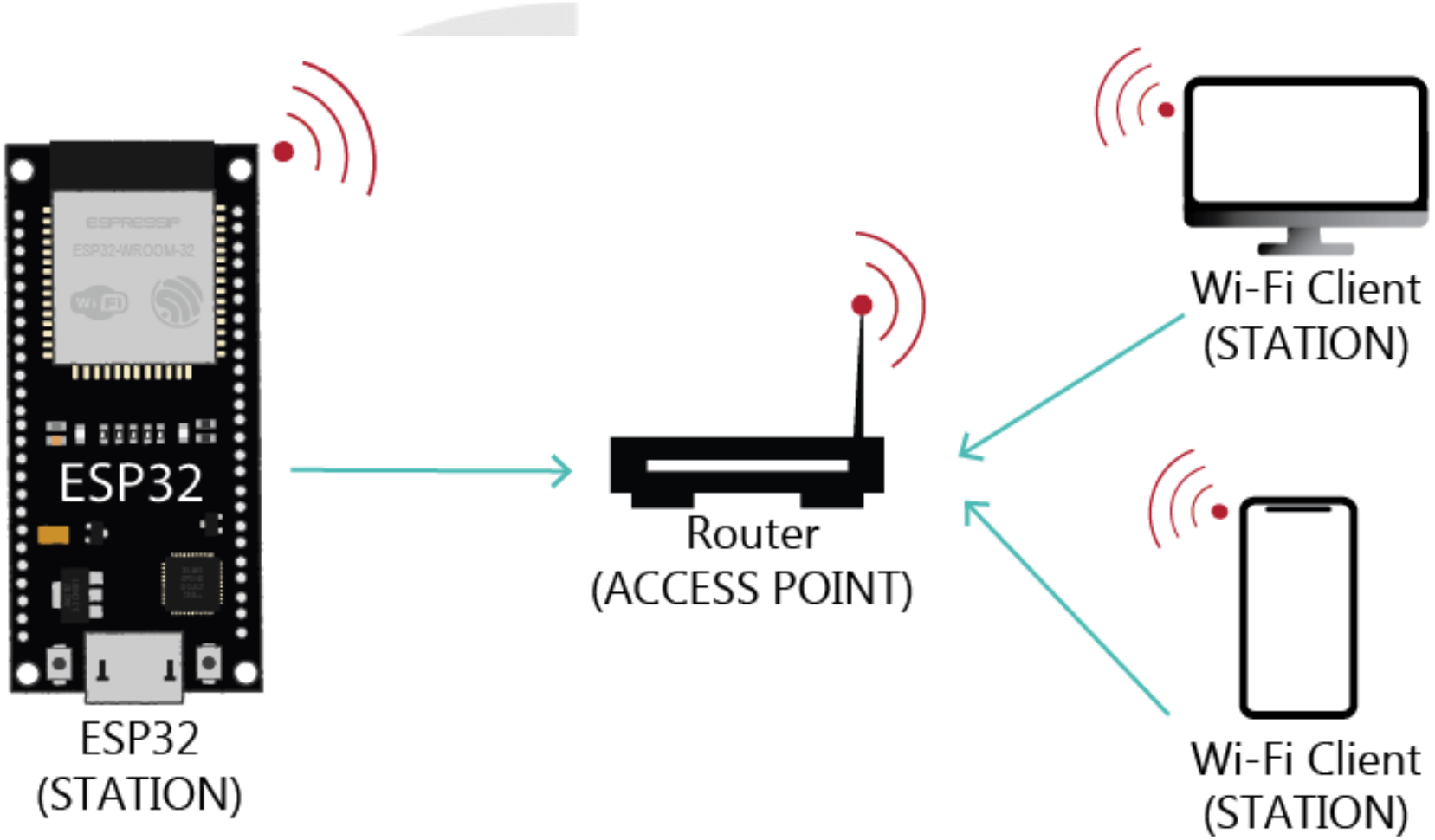
Creating an Open Access Point

```
#include <WiFi.h>

const char* ssid  = “ESP32
const char* password = NULL;

void setup()
{
    Serial.begin(115200);
    Serial.println("Creating AP");
    WiFi.mode(WIFI_AP);
    WiFi.softAP(ssid, password);
    Serial.print("AP Created with IP Gateway ");
    Serial.println(WiFi.softAPIP());
}

void loop(){}
```

Wi-Fi Access Point Customization

```
#include <WiFi.h>

const char* ssid          = “ESP32";              // SSID Name
const char* password      = “jrs123";   // SSID Password - Set to NULL to have an open AP
const int   channel       = 10;                  // WiFi Channel number between 1 and 13
const bool  hide_SSID     = false;                  // To disable SSID broadcast -> SSID will not appear in a basic WiFi scan
const int   max_connection = 2;                  // Maximum simultaneous connected clients on the AP

void setup()
{
    Serial.begin(115200);
    Serial.println("Creating AP");
    WiFi.mode(WIFI_AP);
    WiFi.softAP(ssid, password, channel, hide_SSID, max_connection);
    Serial.print("AP Created with IP Gateway ");
    Serial.println(WiFi.softAPIP());
}

void loop(){}
```

Esp32 Wi-Fi Station



Esp32 Wi-Fi Station

```
#include <WiFi.h>

const char* ssid = "yourNetworkName";
const char* password = "yourNetworkPassword";

void setup(){
    Serial.begin(115200);
    delay(1000);

    WiFi.mode(WIFI_STA); //Optional
    WiFi.begin(ssid, password);
    Serial.println("Connecting");

    while(WiFi.status() != WL_CONNECTED){
        Serial.print(".");
        delay(100);
    }

    Serial.println("Connected to the WiFi network");
    Serial.print("Local ESP32 IP: ");
    Serial.println(WiFi.localIP());
}
```

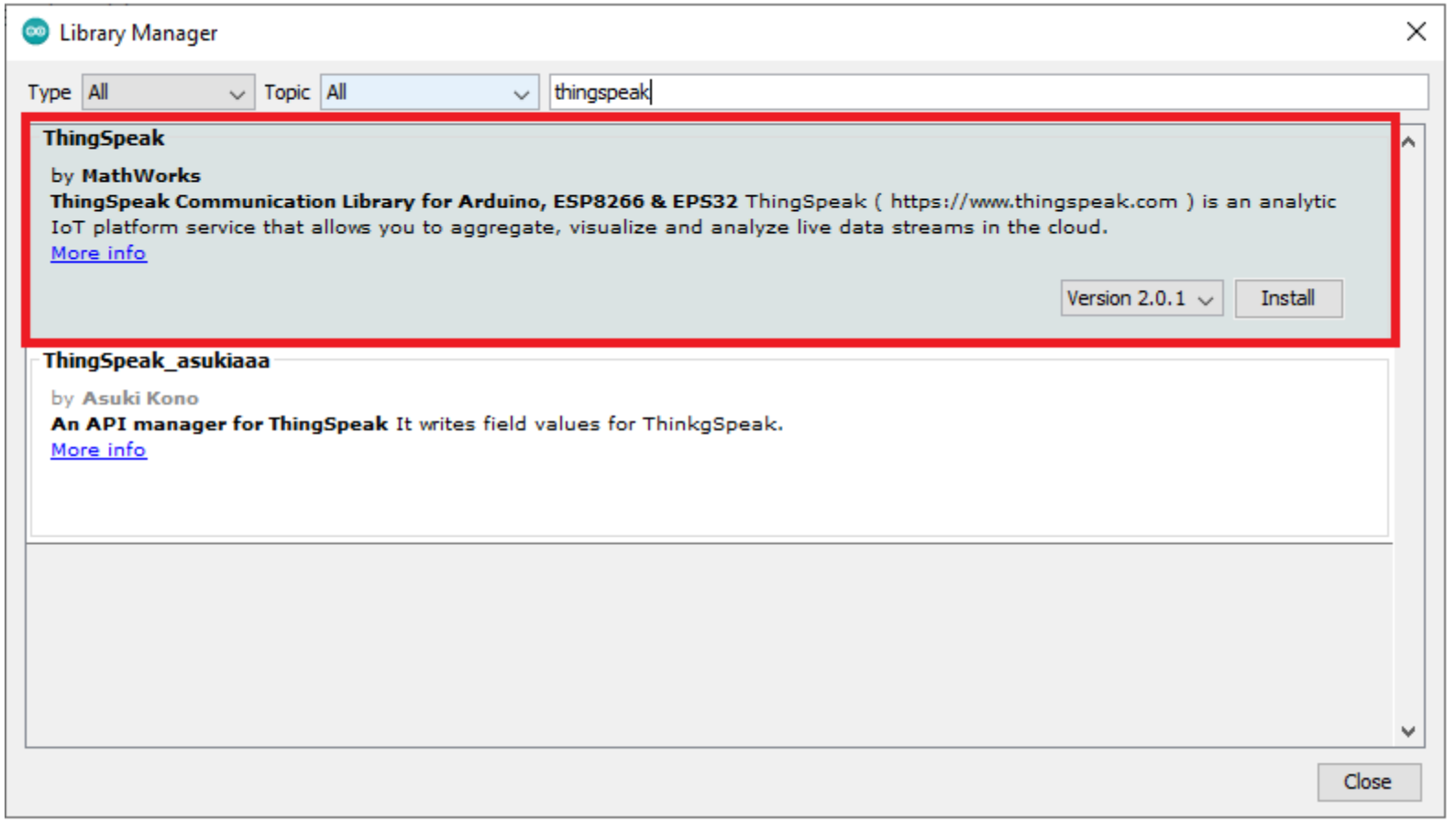
Setting ESP32 both as Wi-Fi AP and Station

```
#include <WiFi.h>
const char* wifi_network_ssid     = “bruno";
const char* wifi_network_password =  “bruno123";
const char *soft_ap_ssid          = “JRS";
const char *soft_ap_password      = NULL;
```

```
void setup()
{
    Serial.begin(115200);
    WiFi.mode(WIFI_AP_STA);

    Serial.println("Creating ESP32 AP");
    WiFi.softAP(soft_ap_ssid, soft_ap_password);
    Serial.print("AP Created with IP Gateway ");
    Serial.println(WiFi.softAPIP());

    WiFi.begin(wifi_network_ssid, wifi_network_password);
    Serial.println("Connecting to WiFi Network");

    while(WiFi.status() != WL_CONNECTED)
    {
        Serial.print(".");
        delay(100);
    }

    Serial.print("Connected to the WiFi network with local IP : ");
    Serial.println(WiFi.localIP());
}

void loop() {}
```

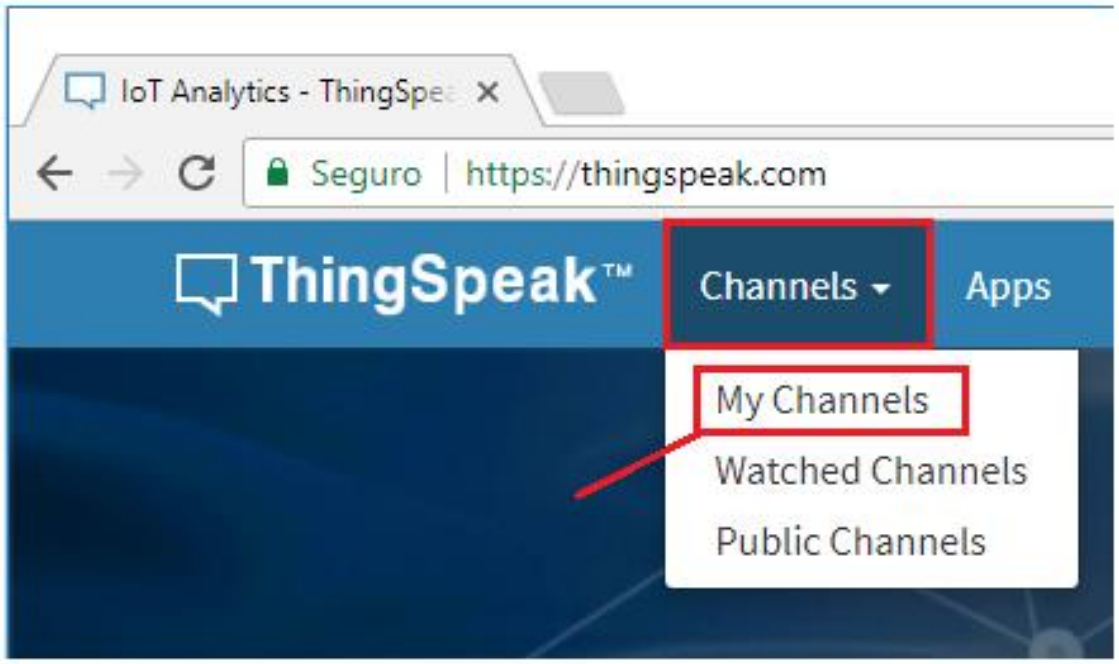
Sending Sensor Readings to Cloud (ThingSpeak)

- To send sensor readings to ThingSpeak, we'll use the thingspeak-arduino library.
- You can install this library through the Arduino Library Manager. Go to Sketch > Include Library > Manage Libraries… and search for "ThingSpeak" in the Library Manager.
- Install the ThingSpeak library by MathWorks.

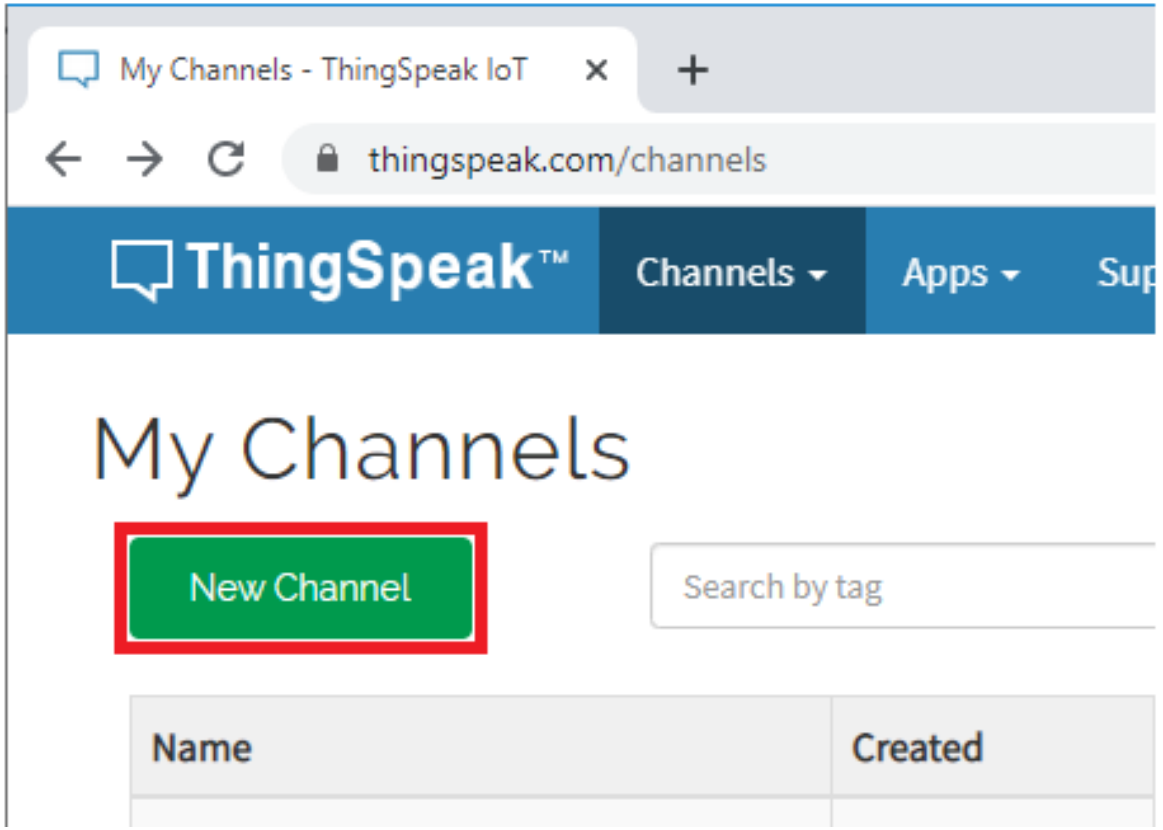

Library Manager
Type All
Topic All
thingspeak
ThingSpeak
by MathWorks
ThingSpeak Communication Library for Arduino, ESP8266 & EPS32 ThingSpeak (https://www.thingspeak.com) is an analytic IoT platform service that allows you to aggregate, visualize and analyze live data streams in the cloud.
More info
Version 2.0.1
Install
ThingSpeak_asukiaaa
by Asuki Kono
An API manager for ThingSpeak It writes field values for ThinkgSpeak.
More info
Close

Creating New Channel

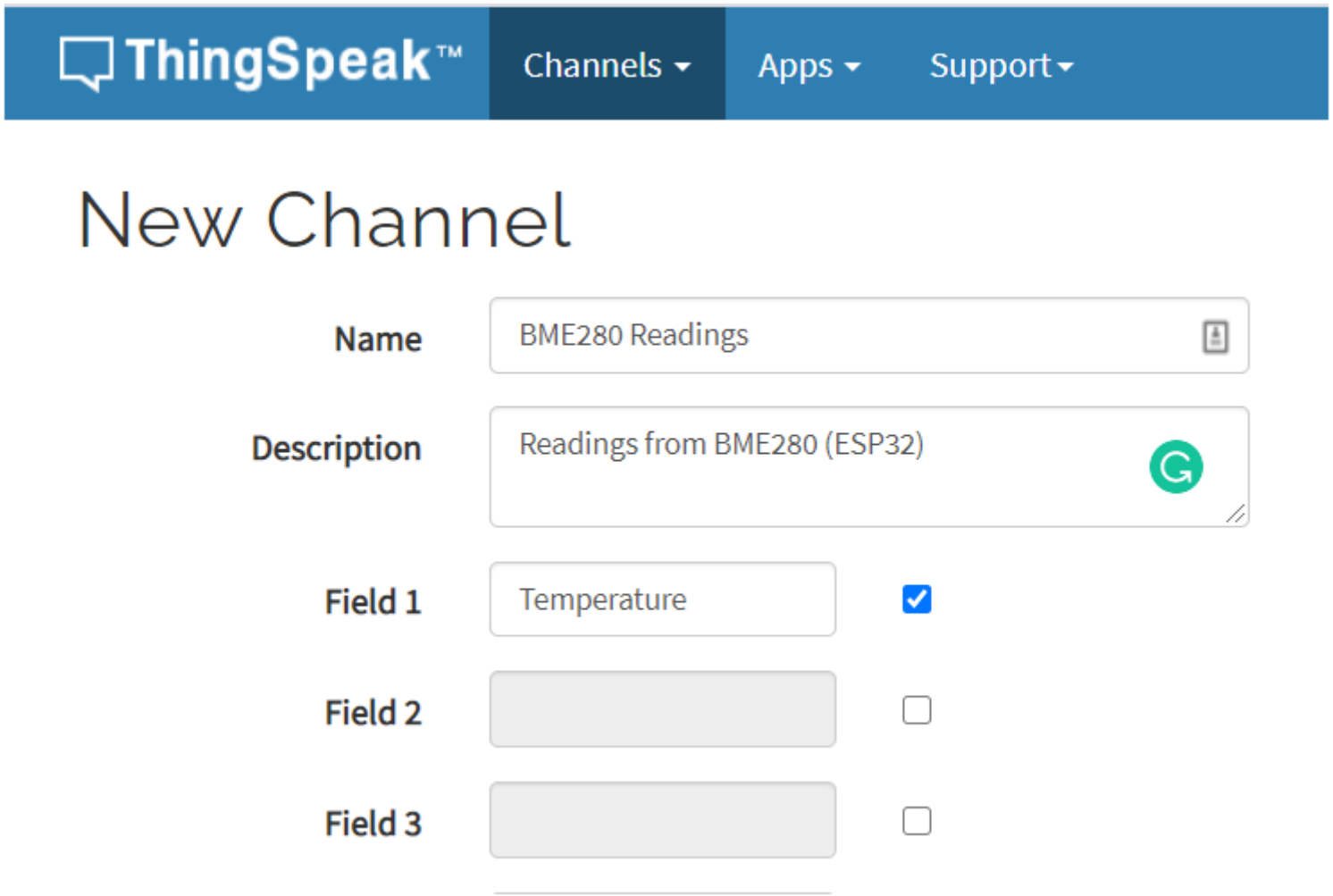
- After your account is ready, sign in, open the “Channels” tab and select “My Channels“.



- Press the “New Channel” button to create a new channel.



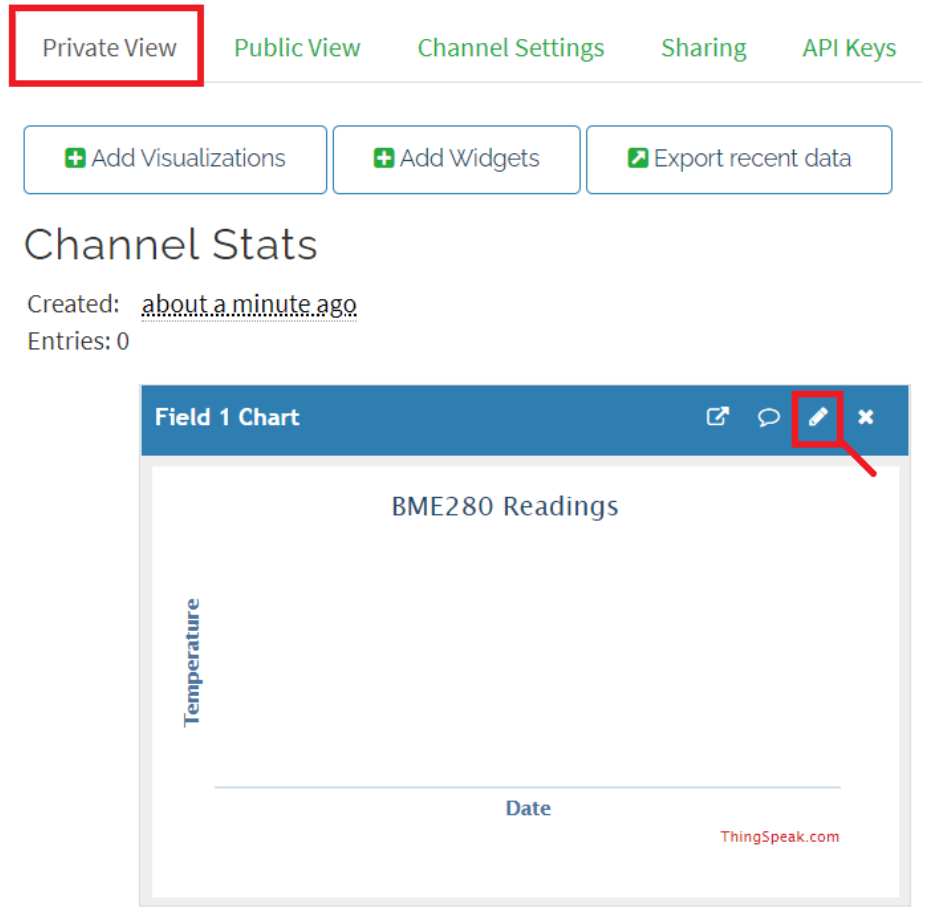
- Type a name for your channel and add a description. In this example, we'll just publish temperature.
- If you want to publish multiple readings (like humidity and pressure), you can enable more fields.


ThingSpeak™
Channels
Apps
Support
New Channel
Name
BME280 Readings
Description
Readings from BME280 (ESP32)
Field 1
Temperature
Field 2
Field 3

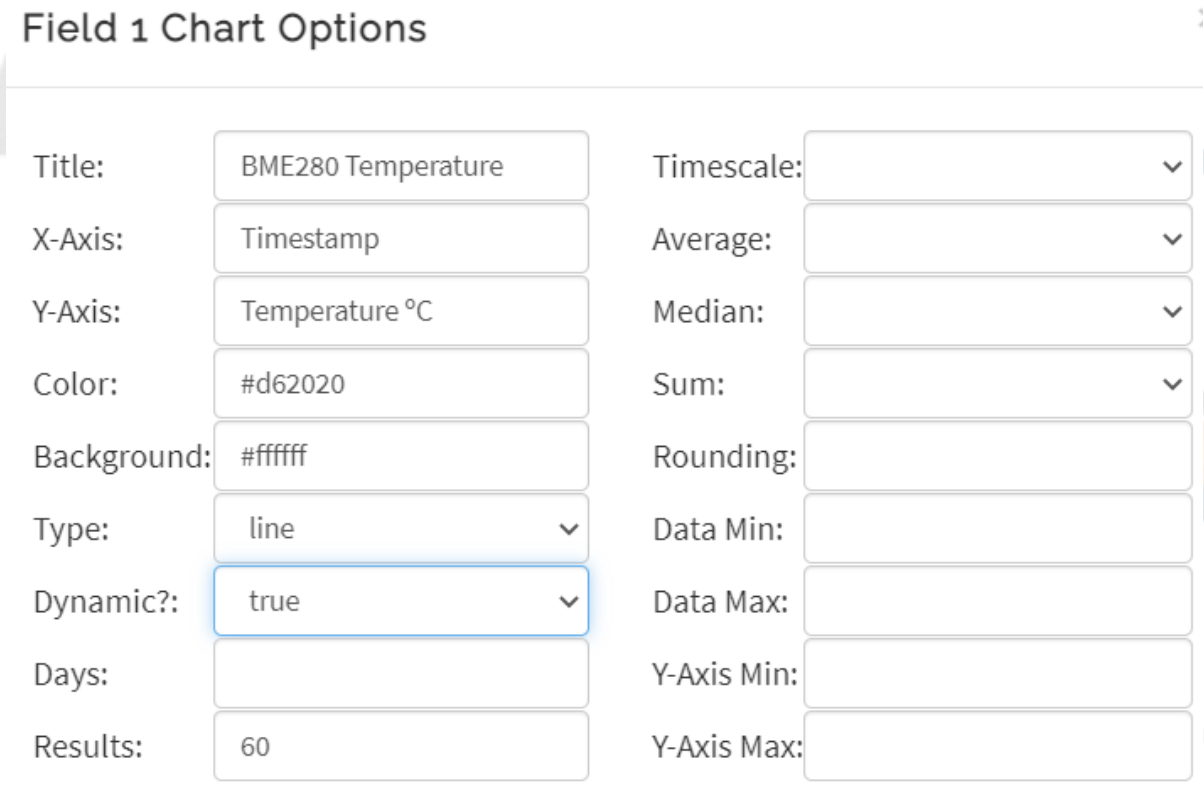
- Click the Save Channel button to create and save your channel.

Customizing Chart

- The chart can be customized, go to your Private View tab and click on the edit icon.

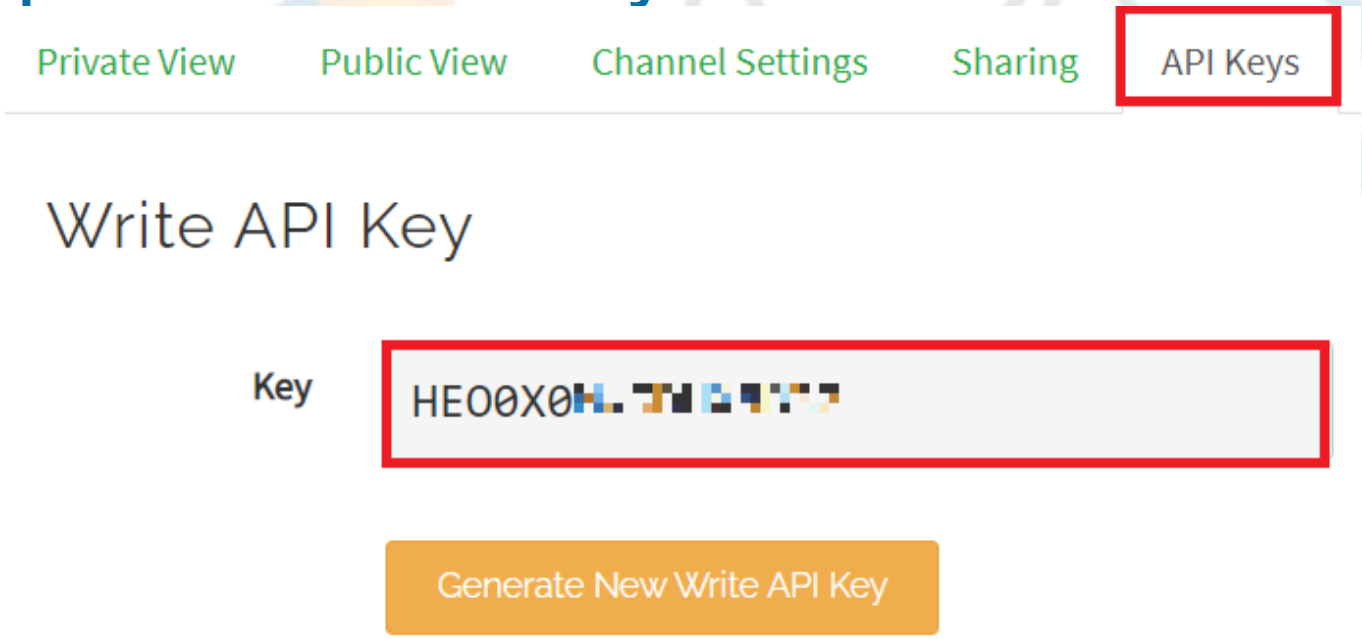


- You can give a title to your chart, customize the background color, x and y axis, and much more.



API Key

- To send values from the ESP32 to ThingSpeak, you need the Write API Key.
- Open the “API Keys” tab and copy the Write API Key to a safe place because you’ll need it in a moment.



Sending DHT Sensor Readings to ThingSpeak

```
#include <WiFi.h>
#include "ThingSpeak.h"
#include <DHT.h>
#define DHTPIN 2
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);
const char* ssid = "Cheap Hardware @ 2K";
const char* password = "passwordyoo";
WiFiClient client;
unsigned long myChannelNumber = 1924259;
const char * myWriteAPIKey = "V7YB32F9PMLB9JXU";
```

```
void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  dht.begin();
  // Connect to WiFi network
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED)
  {
   delay(500);
   Serial.print(".");
  }
  Serial.println("WiFi connected");
  Serial.println(WiFi.localIP());
  ThingSpeak.begin(client);
}
```

```
void loop() {
  // put your main code here, to run repeatedly:
  float tempC = dht.readTemperature();
  float humi = dht.readHumidity();
  Serial.print("Temperature Value is :");
  Serial.print(tempC);
  Serial.print("C ");
  Serial.print("Humidity Value is :");
  Serial.print(humi);
  Serial.println("% ");
  // set the fields with the values
  ThingSpeak.setField(1, tempC);
  ThingSpeak.setField(2, humi);
  // Write to ThingSpeak. There are up to 8 fields in a channel, allowing you to store up to 8 different
  int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);

  if(x == 200){
    Serial.println("Channel update successful.");
  }
  else{
    Serial.println("Problem updating channel. HTTP error code " + String(x));
  }
  delay(2000);
}
```