# ESP32 and ThingSpeak: How to Send Data to Cloud using ESP32

### ESP32 is the powerful tool for IoT applications having inbuilt wifi facility. For the sake of simplicity here the temperature sensor and hall sensor data will be sent to ThingSpeak. ThingSpeak is a free web service for displaying the data online and you can access and monitor the data from ThingSpeak from anywhere. You can use any IoT device to send data to ThingSpeak cloud.

### **Prerequisite**

### ESP32

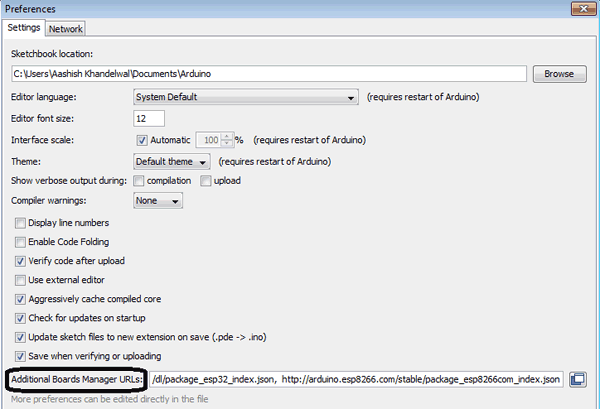
### USB Cable

### Arduino IDE

### **Preparing ESP32 in Arduino IDE**

**Step 1:**Firstly you have to download and install **Arduino IDE** software which you can download from <https://www.arduino.cc/en/Main/Software> for free. If you have already installed in your PC then make sure that it is latest version of IDE as older version doesn’t include ESP32 board.

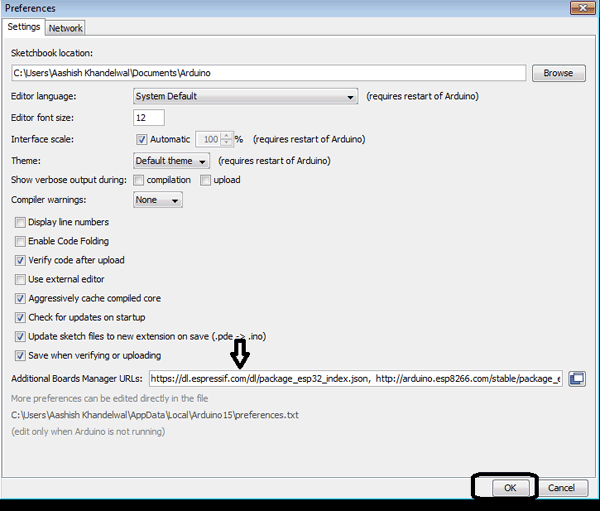
**Step 2:**After installing, open IDE and go to Files -> Preferences and open preference window and see the **“Additional Boards Manager URL’s”** as:



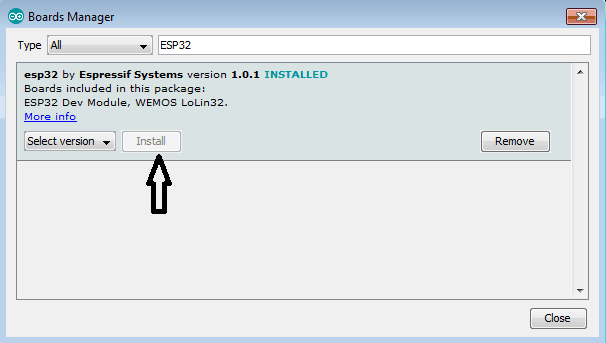
**Step 3:** This box maybe empty or contain some other URL if you have used it previously for ESP8266. You just have to paste below URL into this box if the box contains already another URL then paste it by separating another URL using comma(,).

<https://dl.espressif.com/dl/package_esp32_index.json>

**Step 4:** After pasting the given URL my window looks like this as I already used ESP82666, Now press OK and the window will disappear.



**Step 5**: Now go to Tools-> Board-> Board Manager and search for ESP32 and press **install**, it will take some time to install make sure that you have internet connection while installing after installing your window looks like this:

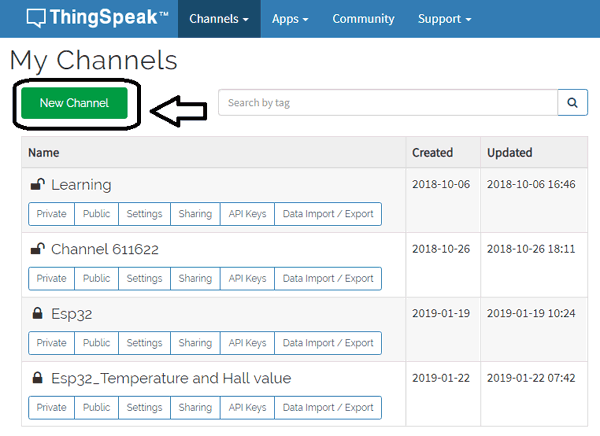


 After this close the window of board manager and your IDE is ready to work with ESP32.

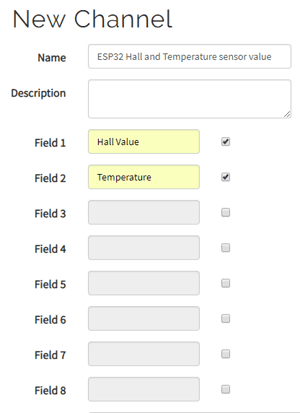
### **ThingSpeak Setup for ESP32**

ThingSpeak is a free web service which helps us in IoT based projects. By using ThingSpeak server, we can monitor our data over the internet using the API and channels provided by ThingSpeak. The details regarding **how to send sensor data of ESP32 to ThingSpeak server** is provided below.

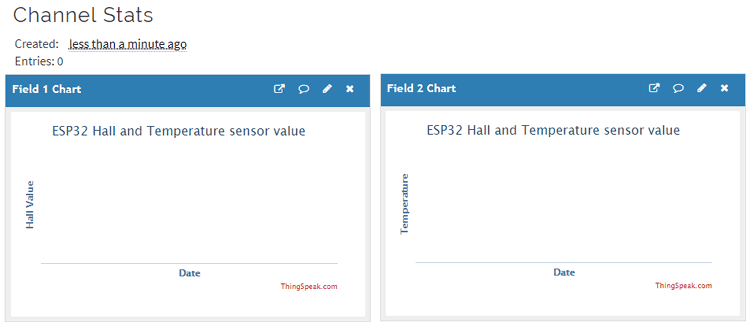
1. Firstly go to <https://thingspeak.com/> and create an account and sign in to this server.
2. After signing in you will find below window in which number of channels are listed in this go to *New channel*.



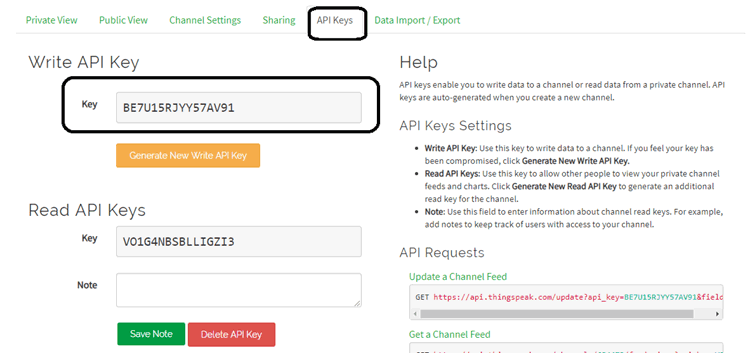
1. After clicking on *New Channel* you will find a window in which you have to enter some details about the channel, in this project we want to analyze temperature and Hall sensor value of ESP32 so you will require 2 fields. So enter the details as shown and save the channel.



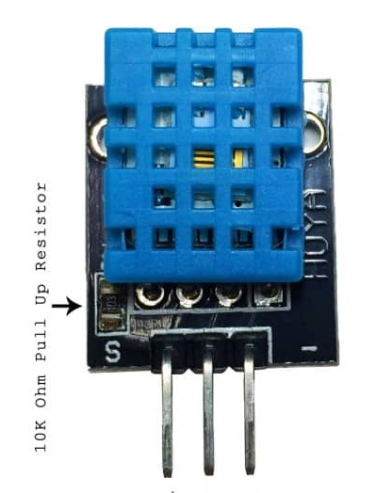
1. After saving of channel you will find channel stats window showing details about your channels.

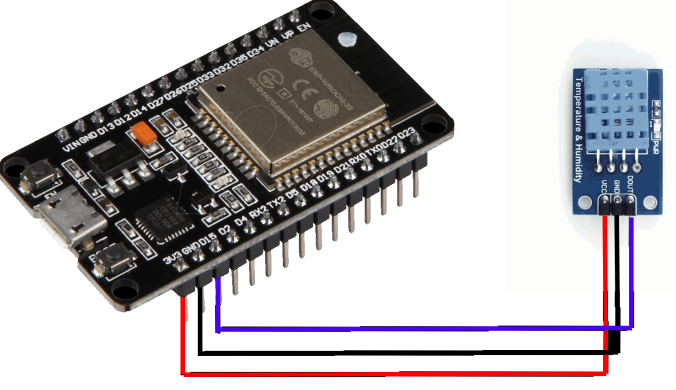


1. Now go to API key menu which shows you **Write API keys and Read API key**, Copy Write API key as you will required this API during programming of ESP32.



 Circuit design:





The electric connection to the ESP32 DevKit is very simple, as the DHT series can be powered direct with 3.3V. Only 3 wires are needed: VCC, GND and the data line. Data Line is connected to GPIO5 (D5).

**Code:**

#include <WiFi.h>

#include <WebServer.h>

#include "DHT.h"

#include <ThingSpeak.h>

#define DHTTYPE DHT11

const char\* ssid = "CPH1701";

const char\* password = "12345678";

WiFiClient client;

long myChannelNumber = 1223990;

const char myWriteAPIKey[] = "81M00UIMQS9FCOA8";;

uint8\_t analogpin = 4;

int light;

uint8\_t DHTPin = 5;

DHT dht(DHTPin, DHTTYPE);

void setup() {

Serial.begin(115200);

delay(100);

pinMode(DHTPin, INPUT);

pinMode(analogpin, INPUT);

dht.begin();

ThingSpeak.begin(client);

Serial.println("Connecting to ");

Serial.println(ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(1000);

Serial.print(".");

}

Serial.println("");

Serial.println("WiFi connected..!");

Serial.print("Got IP: "); Serial.println(WiFi.localIP());

}

void loop() {

light = analogRead(analogpin);

Serial.print("Light - ");

Serial.println(light,DEC);

delay(4000);

float h = dht.readHumidity();

float t = dht.readTemperature();

Serial.println("Temperature: " + (String) t);

Serial.println("Humidity: " + (String) h);

ThingSpeak.writeField(myChannelNumber, 1, t, myWriteAPIKey);

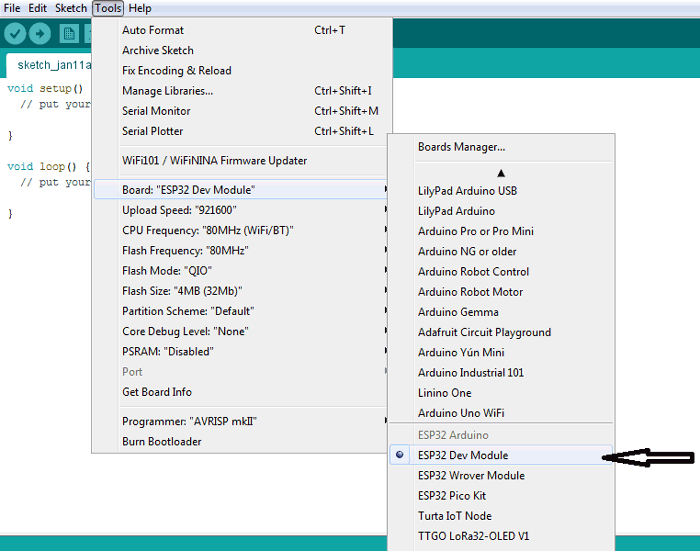
ThingSpeak.writeField(myChannelNumber, 2, h, myWriteAPIKey);

delay(4000);

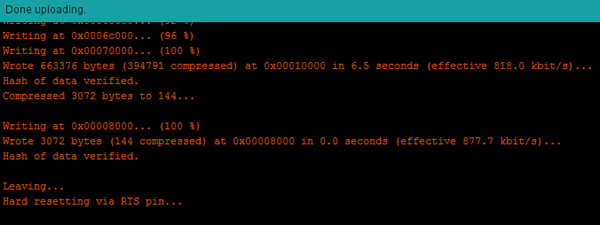
}

Now to upload the program in ESP32 we have to follow some steps as follows:

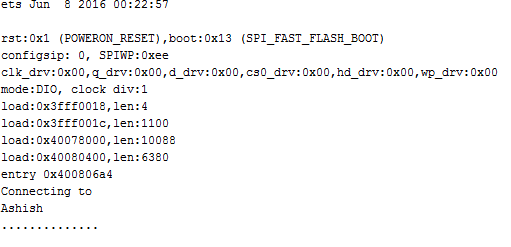
1. Open your Arduino IDE and create a new file and save it where you want.
2. Copy the given code.
3. Now go to Tools--> Board--> ESP32 Dev Module.



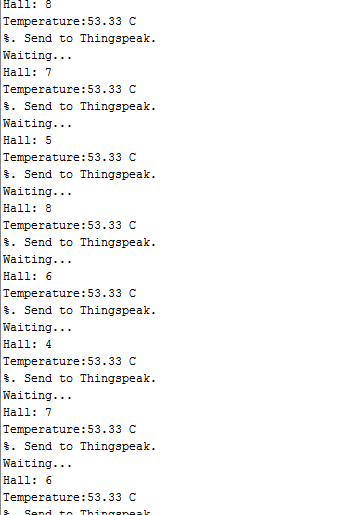
1. Now go to *Tools-->Port* and select port to which your ESP32 is connected.
2. Now click on upload to upload the code.
3. After complete uploading you will find message like this in your output console.



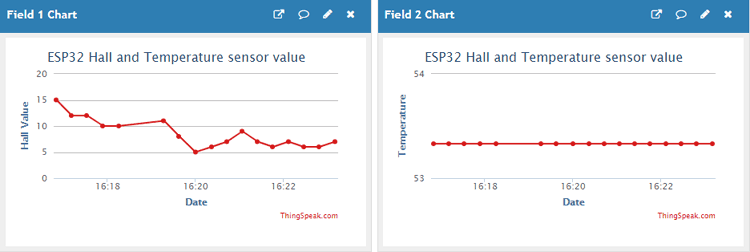
1. Now open your serial monitor and press reset button of your ESP32, it starts connecting to your hotspot.



1. After connecting successfully it **starts displaying temperature and hall sensor value on your serial monitor window** and sending these values to ThingSpeak.



1. Now open your ThingSpeak account in your browser and you will find graphs displaying temperature and hall sensor values.



This is how you are successfully able to send temperature and hall sensor values to ThingSpeak cloud using ESP32. Now you can simply add any sensor to ESP32 and can send the sensor value to ThingSpeak IoT cloud to be monitored from anywhere.