EXPERMINT: 05

- **Title**: Esp32 and ThingSpeak

- **Aim:** To study and implement a program on ESP32/NodeMCU to push and retrieve the data from any cloud like Thingspeak, Thingsboard, AWS, Azure etc.

- **Theory:**

ESP32

ESP32 is a series of feature-rich MCU with integrated Wi-Fi and Bluetooth connectivity for a wide range of applications. Espressif Systems, China, produce it.ESP32 is cheap and nearly ten times faster than Arduino Uno and is a 32-bit versatile device. Developers of ESP32 IC made a small module board with edge castellations. One popular version of such a module board is called ESP-WROOM-32. It is a dual-core, 32-bit microcontroller unit, and all the cores can be controlled individually. It has integrated Wi-Fi, Bluetooth, and Bluetooth Low Energy withmultiple digital and analog I/O pins.

Why ESP32 is better than Raspberry Pi ?

The ESP32 and Raspberry Pi are both microcontrollers that can handle IoT tasks. However, the ESP32 is a better fit for wireless communication tasks, such as building a smart thermostat or weather monitoring system.

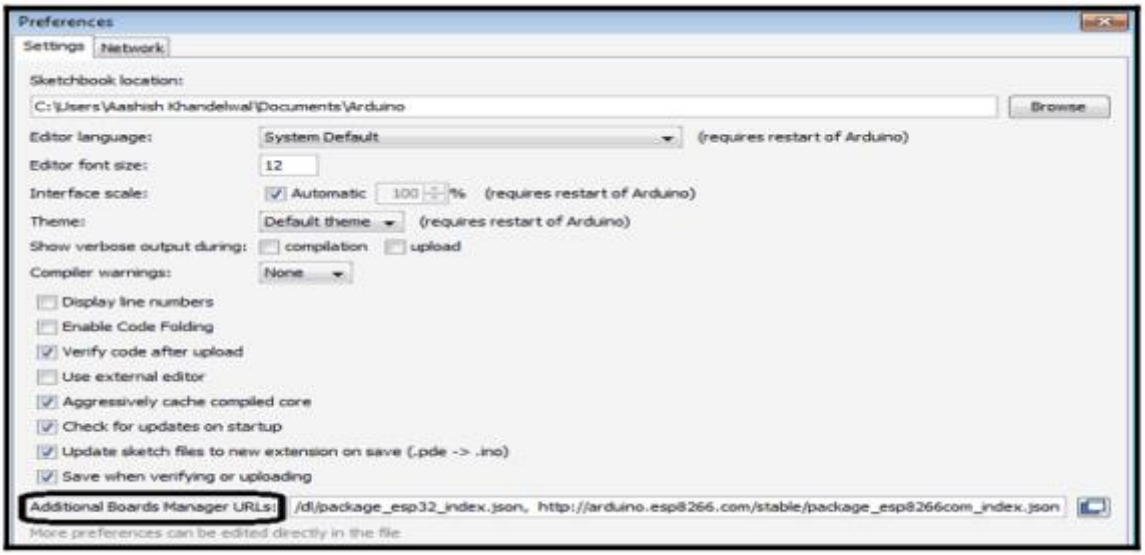
Here are some other advantages of the ESP32:

- Compatible with Arduino code
- Integrated Bluetooth, WiFi, and camera
- MicroPython and Arduino IDE support
- Larger modules than the Pico chip
- Ultra-low consumption functions

Procedure:

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.

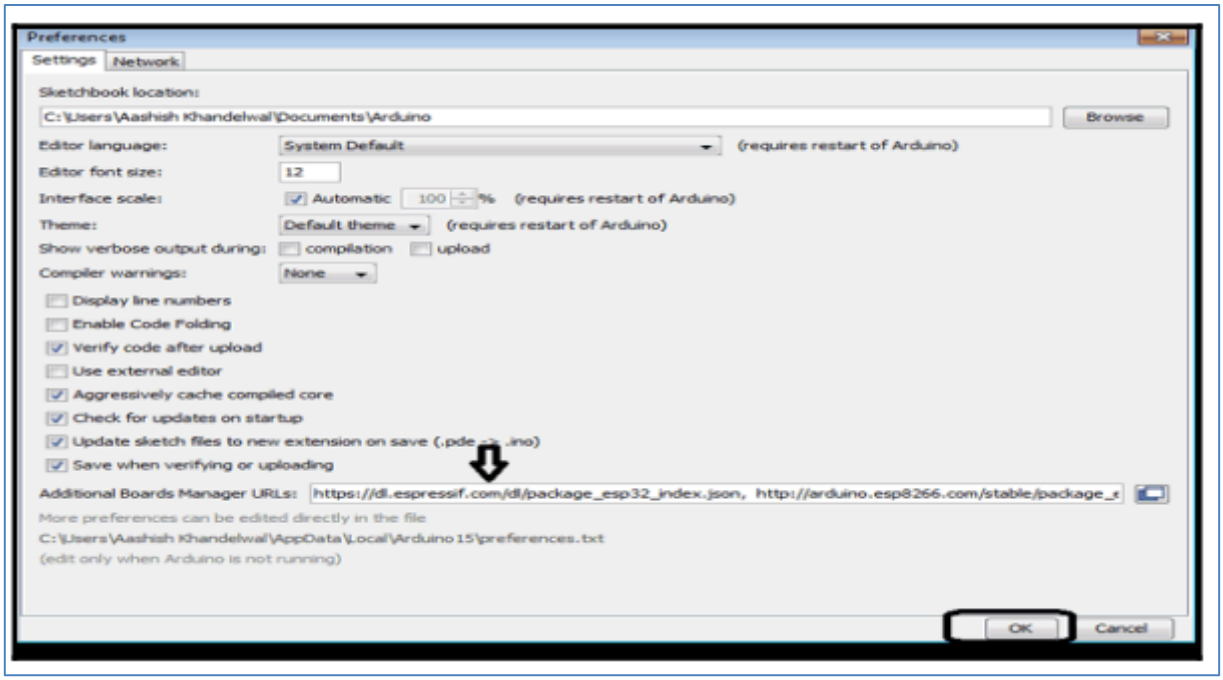
Step 2: After installing, open IDE and go to Files -> Preferences and open preference window and see the "**Additional Boards Manager URL's**" as:



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.



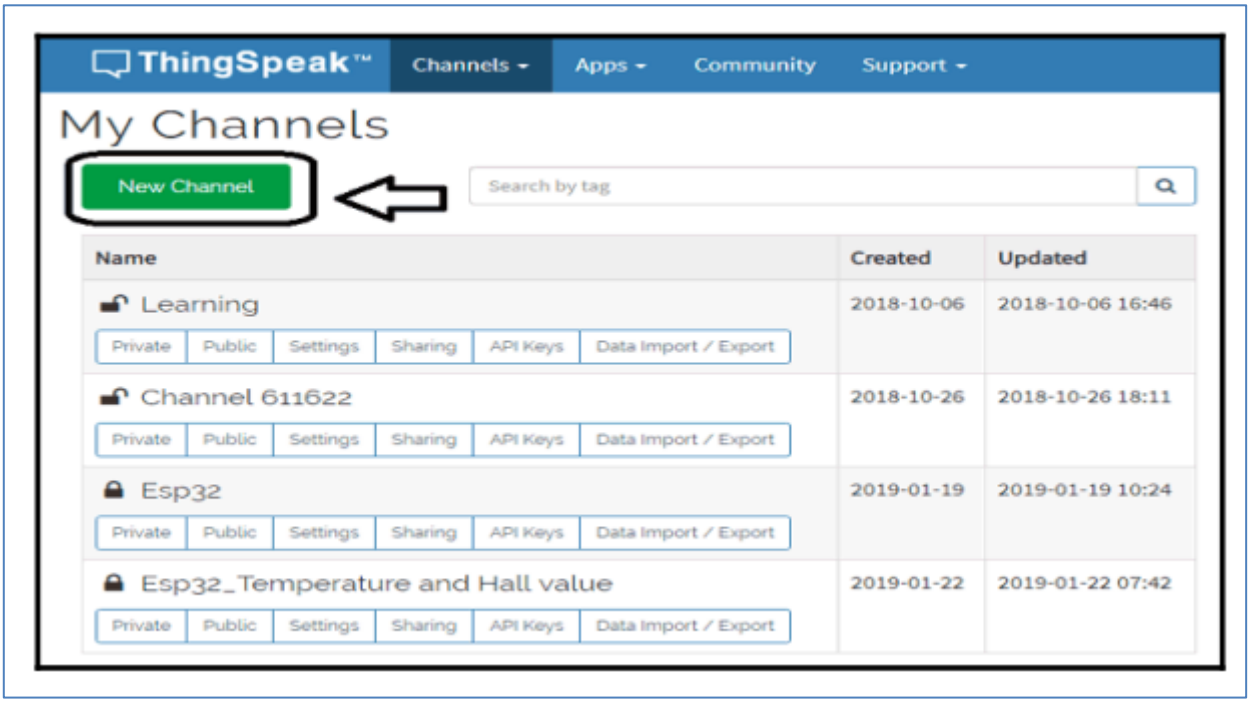
Step 5: Now go to Tools-> Board-> Board Manager and search for ESP32 and press install, it will take some time to install make sure that you have internet connection while installing after installing your window looks like this:

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

ThingSpeak Setup for ESP32

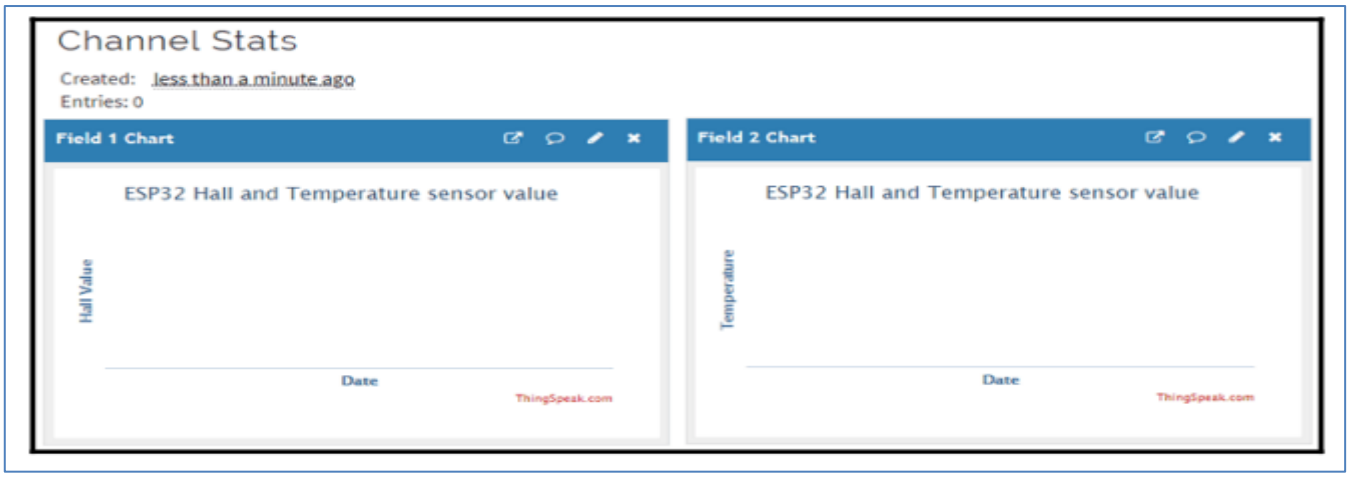
ThingSpeak is a free web service which helps us in IoT based projects. By using ThingSpeak server, we can monitor our data over the internet using the API and channels provided by ThingSpeak. In this section I am explaining about how to send sensor data of ESP32 to ThingSpeak server. For this you have to follow following steps:

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.

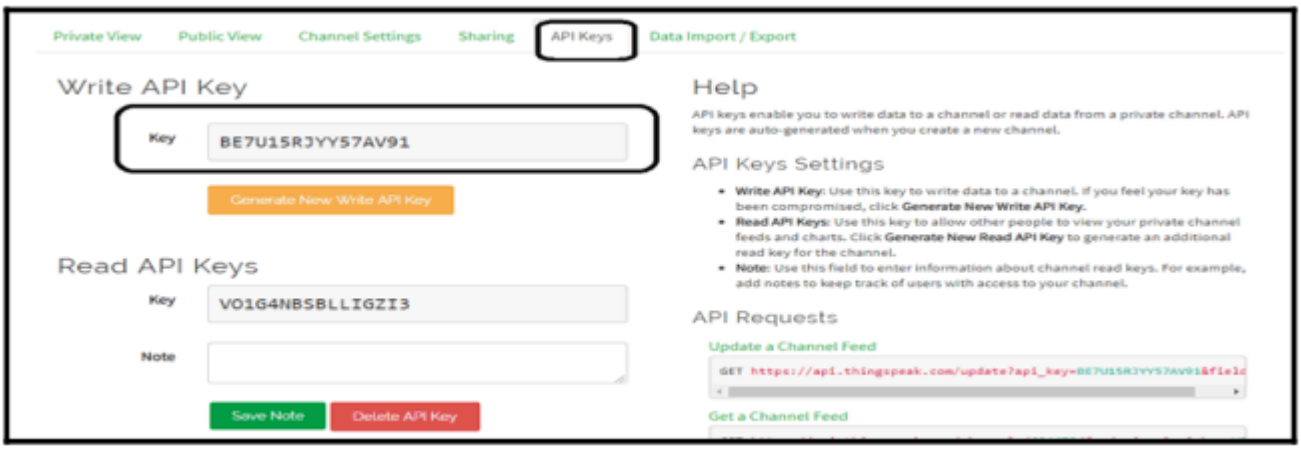


3. After clicking on New Channel you will find a window in which you have to enter some details about the channel, in this project we want to analyze temperature and Hall sensor value of ESP32 so you will require 2 fields. So enter the details as shown and save the channel.

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



5. 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.



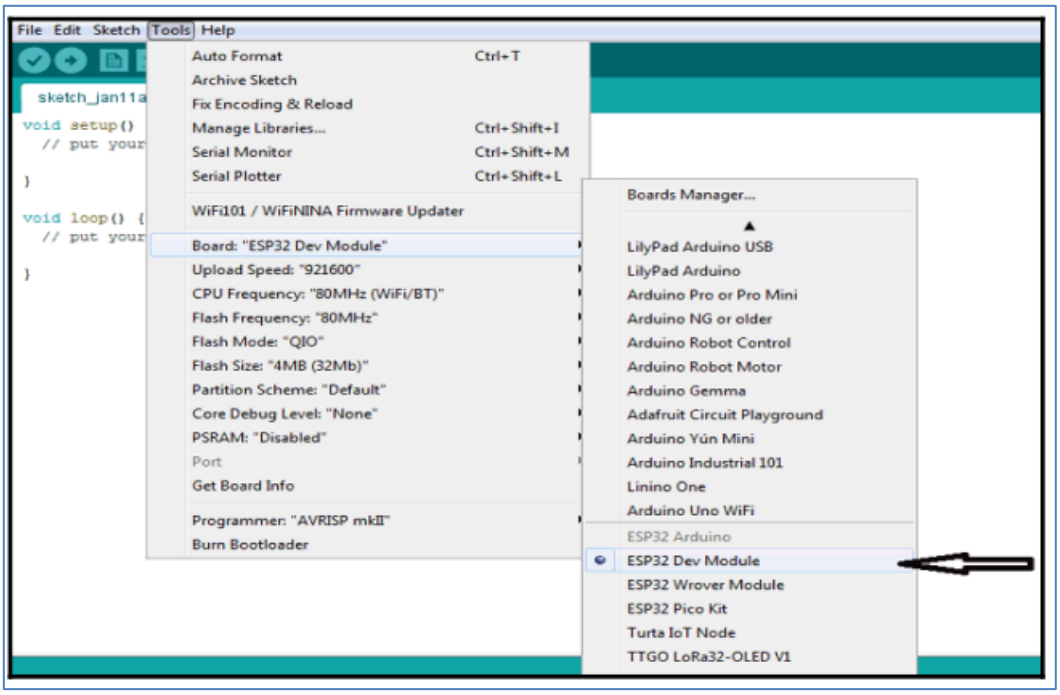
Program code:

```
//ESP32 with Thingspeak
#include <WiFi.h>
#include <HTTPClient.h>
#include "DHT.h"
#define DHTPIN 5
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
const char* ssid = "Test"; // replace with your wifi ssid and password
const char* password = "12345678";
const char* serverName = "http://api.thingspeak.com/update";
String apiKey = "DKB0DG7AEAPIQKHC"; // enter your api key
void setup() {
Serial.begin(115200);
WiFi.begin(ssid, password);
dht.begin();
Serial.println("Connecting");
while(WiFi.status() != WL_CONNECTED)
{
delay(500);
Serial.print(".");
}
Serial.println("");
Serial.print("Connected to WiFi network with IP Address: ");
Serial.println(WiFi.localIP());
}
void loop() {
if(WiFi.status()== WL_CONNECTED){
```

```
WiFiClient client;

HTTPClient http;

delay(10000);

float h = dht.readHumidity();

float t = dht.readTemperature();

if (isnan(t) || isnan(h)) {

Serial.println(F("Failed to read from DHT sensor!"));

return;

}

http.begin(client, serverName);

http.addHeader("Content-Type", "application/x-www-form-urlencoded");

String httpRequestData = "api_key=" + apiKey + "&field1=" + String(t) + "&field2=" +

String(h);

int httpResponseCode = http.POST(httpRequestData);

Serial.print("HTTP Response code: ");

Serial.println(httpResponseCode);

http.end();

} else {

Serial.println("WiFi Disconnected");

}}
```

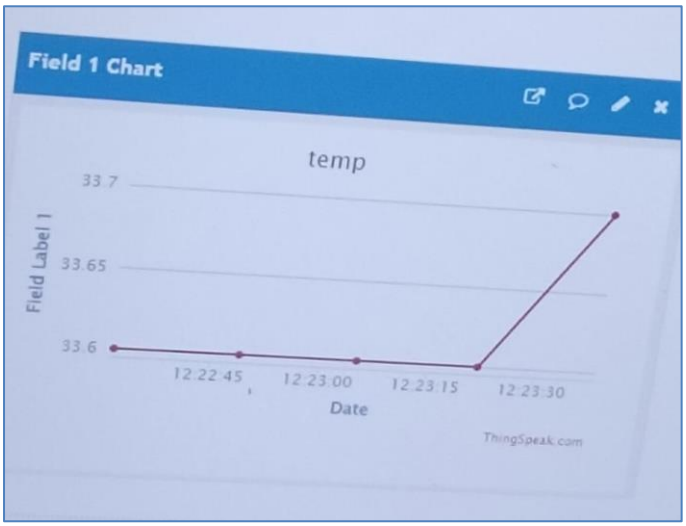
Now to upload the program in ESP32 we have to follow some steps as follows:

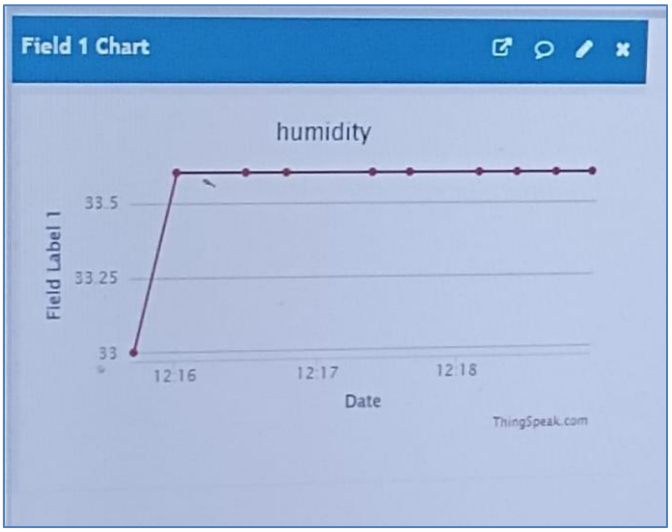
1. Open your Arduino IDE and create a new file and save it where you want.
2. Copy the given code.
3. Now go to Tools--> Board--> ESP32 Dev Module.



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

Result:





Conclusion:

The study and implementation of a program on ESP32/NodeMCU to push and retrieve data from cloud platforms represent a significant step towards building scalable, efficient, and intelligent IoT solutions. By leveraging the capabilities of cloud computing, we can unlock new opportunities for innovation, collaboration, and value creation in the rapidly evolving IoT landscape.