CSE423: Embedded System Summer-2020

ThingSpeak Arduino Weather Station

Todays Lecture

- *Get familiar with ThningSpeak*
- *Creating account in ThingSpeak*
- *API Keys*
- *Integrating Arduino + ESP-01*
- *Collecting Data*

What is ThingSpeak

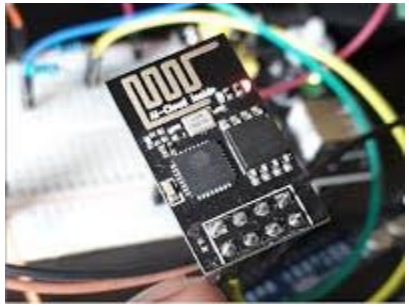
- **ThingSpeak** is a Web Service that lets you collect and store sensor data in the cloud and develop Internet of Things (IOT) applications.

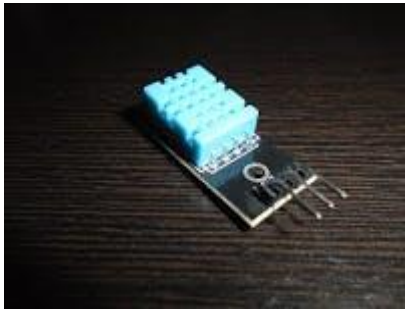
- It works with Arduino, Raspberry Pi and MATLAB (premade libraries and APIs exists).

- Link: https://thingspeak.com/

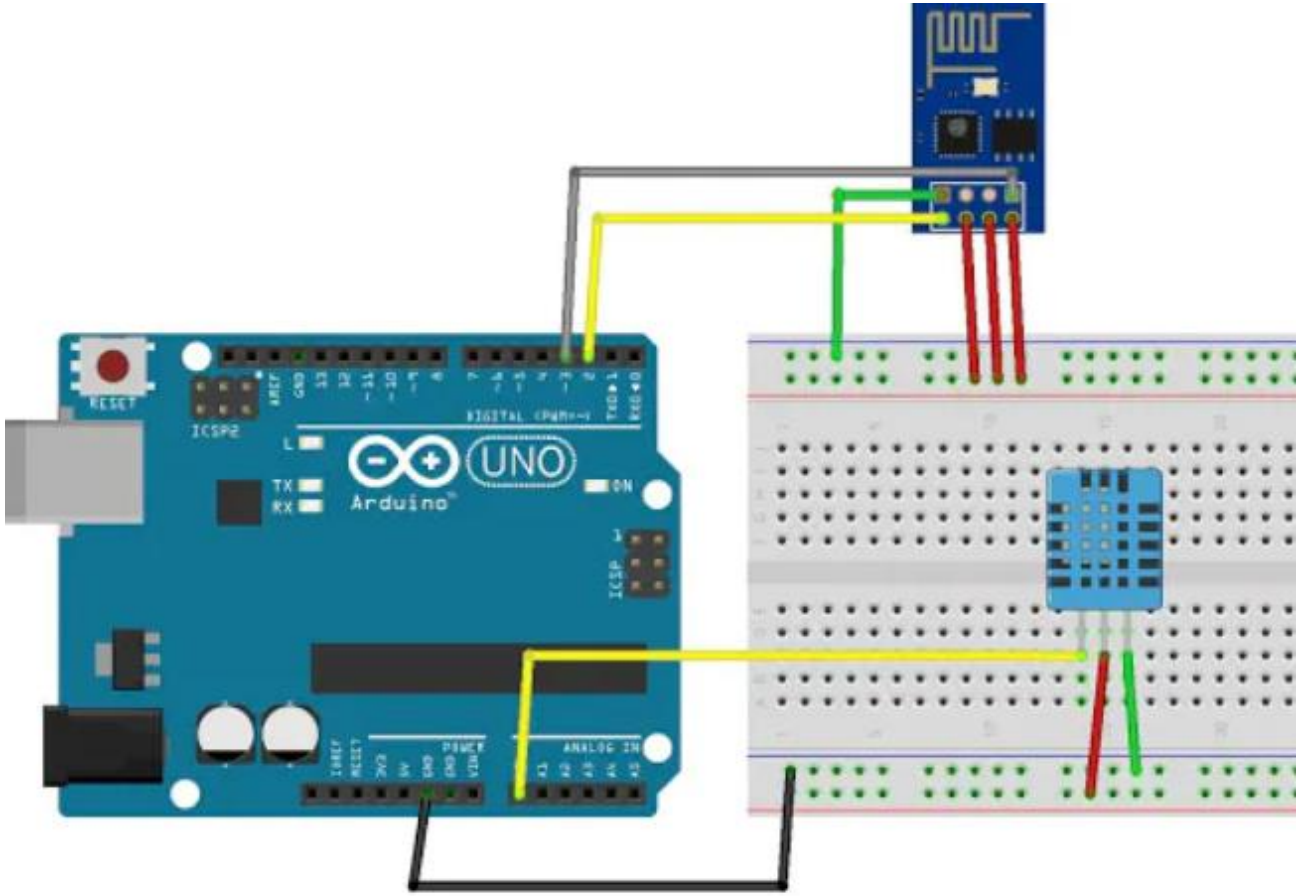
Hardware Requirements

- Arduino UNO
- ESP – 01
- DHT-11 sensor (Temperature and Humidity Sensor)
- Breadboard
- External Power Supply
- Connecting wires





Connection Diagram



CONNECTIONS

ESP -01

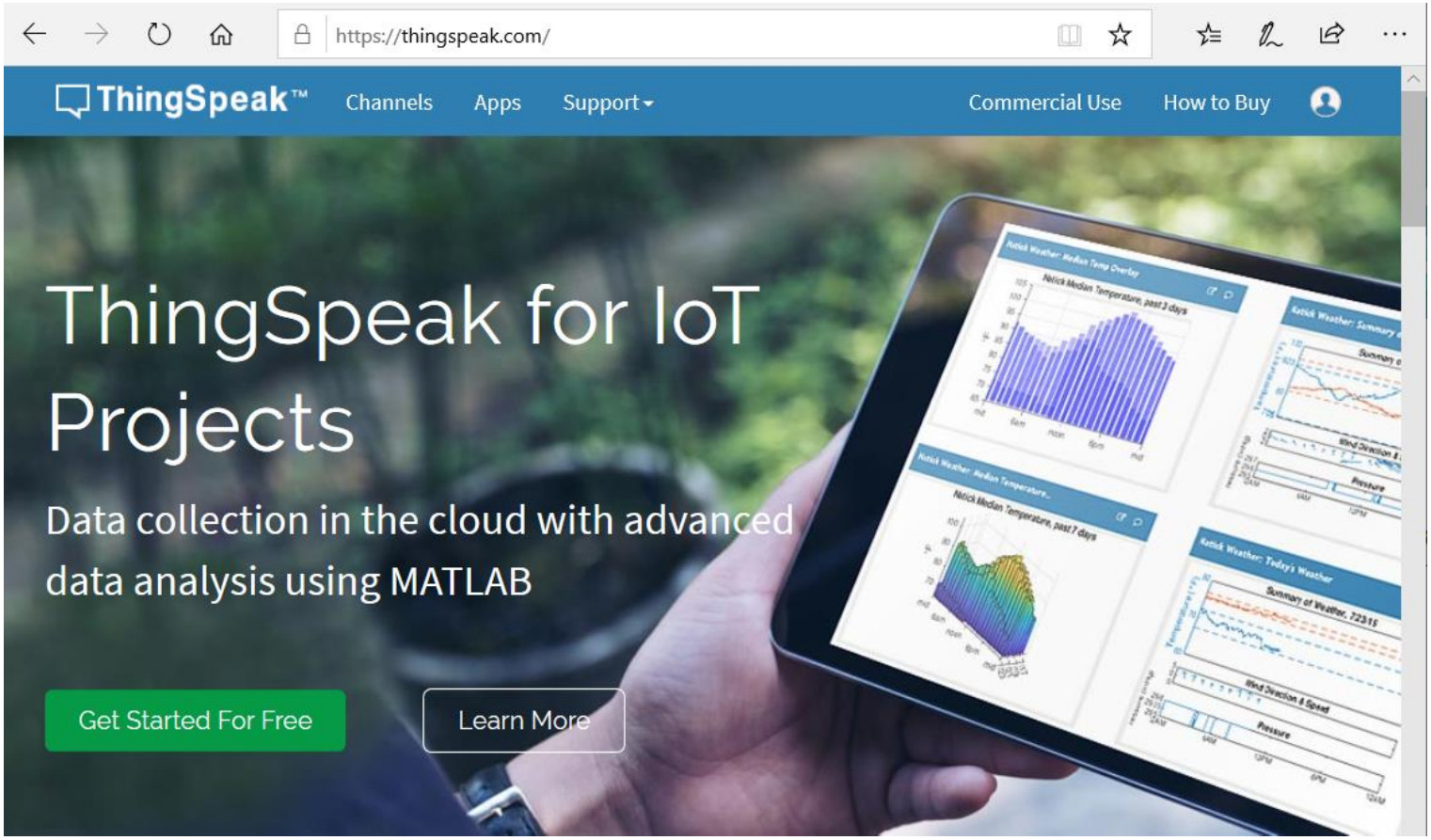
1. VCC - 3.3V
2. GND - GND
3. CH_PD - 3.3V
4. RESET - 3.3V
5. TX - 2 (Arduino)
6. RX - 3 (Arduino)

DHT - 11

1. VCC - 5V
2. OUT - A0
3. GND - GND

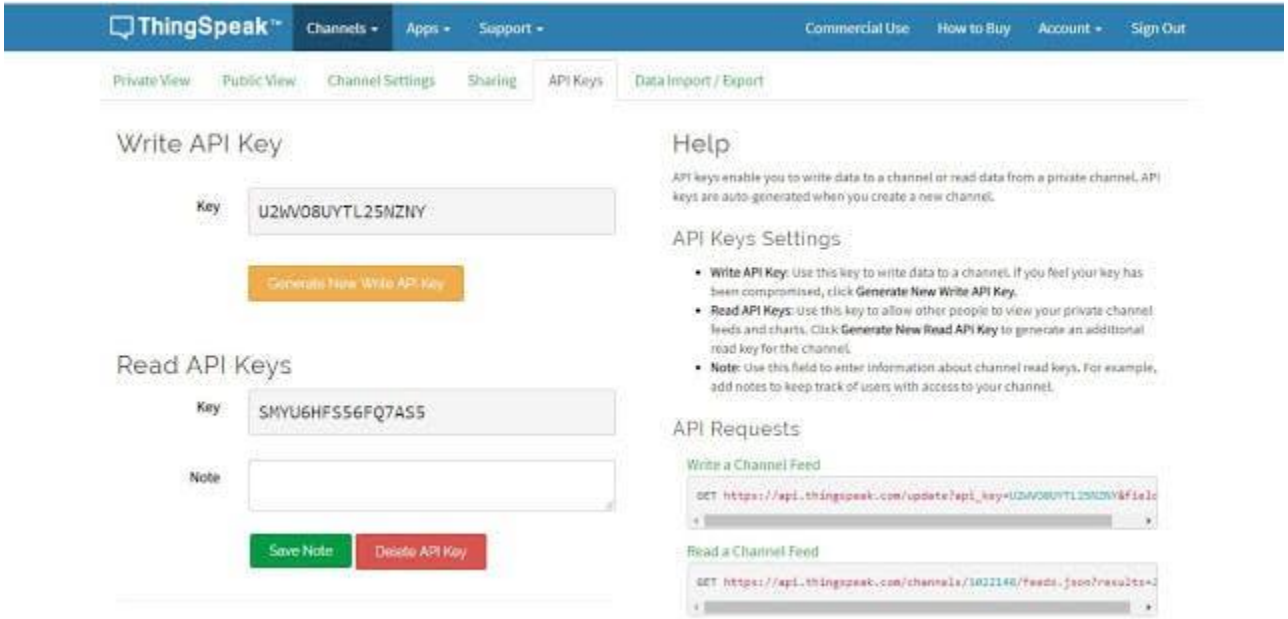
SETTING UP THINGSPEAK

Go to the thingspeak home page- https://thingspeak.com/



SETTING UP THINGSPEAK

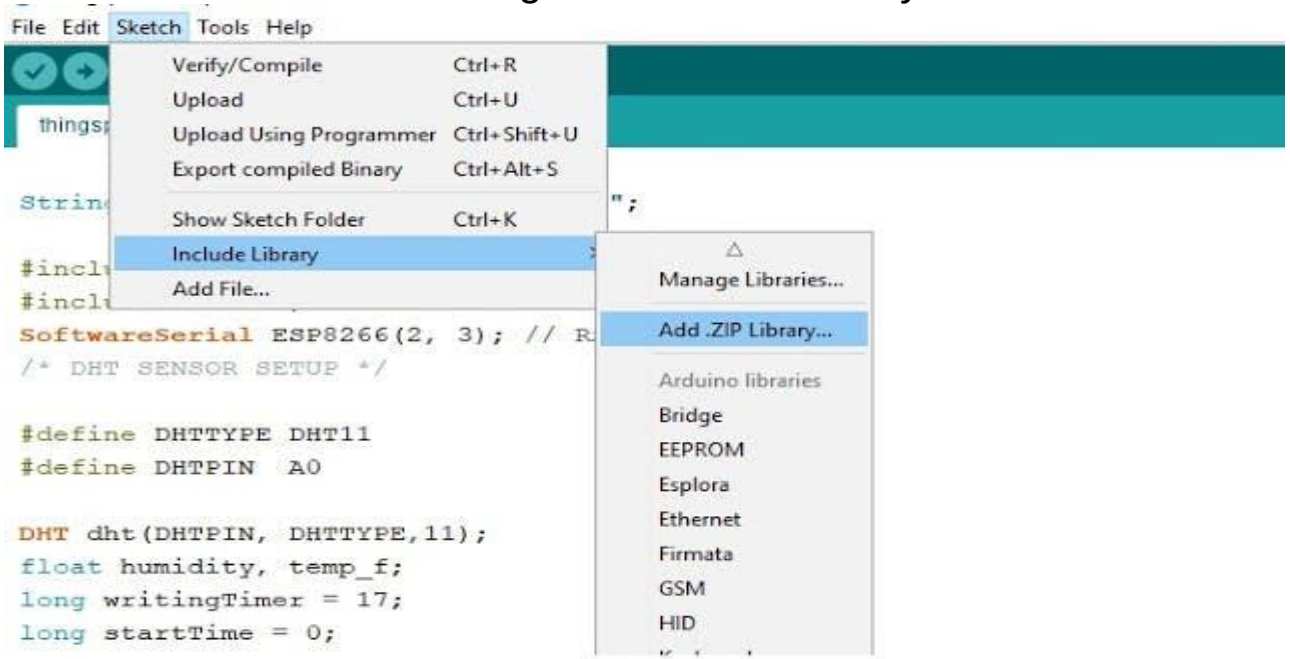
Sign up and create a New Channel with two fields Temperature and Humidity. On creating a new channel, you'll get something called the API Keys. We will use "Write API Keys" for the purpose.



SETTING UP THE ARDUINO IDE

1. The libraries you need to install on your own are the Adafruit Unified Sensor Library and the DHT sensor library. Download links DHT Sensor by adafruit- https://github.com/adafruit/DHT-sensor-libraryAdafruit Unified Sensor Library- https://github.com/adafruit/Adafruit_Sensor

2. Download both and add them using the Add.ZIP Library feature in the IDE



Code-1

```
String myAPIkey = "Your Write API key here";  // API from ThingSpeak

#include <SoftwareSerial.h>
#include <DHT.h>;
SoftwareSerial ESP8266(2, 3); // Rx,  Tx

#define DHTTYPE DHT11
#define DHTPIN  A0

DHT dht(DHTPIN, DHTTYPE,11);
float humidity, temp_f;
long writingTimer = 17;
long startTime = 0;
long waitTime = 0;


boolean relay1_st = false;
boolean relay2_st = false;
unsigned char check_connection=0;
unsigned char times_check=0;
boolean error;
```

Code-2

```
void setup()
{
  Serial.begin(9600);
  ESP8266.begin(9600);
   dht.begin();
  startTime = millis();
  ESP8266.println("AT+RST");
  delay(2000);
  Serial.println("Connecting to Wifi");
   while(check_connection==0)
  {
    Serial.print(".");
  ESP8266.print("AT+CWJAP=\"Wifi Network's name\",\"PassWord\"\r\n");
  ESP8266.setTimeout(5000);
 if(ESP8266.find("WIFI CONNECTED\r\n")==1)
 {
 Serial.println("WIFI CONNECTED");
 break;
 }
```

Code-3

```
times_check++;
 if(times_check>3)
 {
  times_check=0;
   Serial.println("Trying to Reconnect..");
  }
  }
}
void loop()
{
  waitTime = millis()-startTime;
  if (waitTime > (writingTimer*1000))
  {
    readSensors();
    writeThingSpeak();
    startTime = millis();
  }
}
```

Code-4

```
void readSensors(void)
{
  temp_f = dht.readTemperature();
  humidity = dht.readHumidity();
}


void writeThingSpeak(void)
{
  startThingSpeakCmd();
  // preparacao da string GET
  String getStr = "GET /update?api_key=";
  getStr += myAPIkey;
  getStr +="&field1=";
  getStr += String(temp_f);
  getStr +="&field2=";
  getStr += String(humidity);
  getStr += "\r\n\r\n";
  GetThingspeakcmd(getStr);
}
```

Code-5

```
void startThingSpeakCmd(void)
{
  ESP8266.flush();
  String cmd = "AT+CIPSTART=\"TCP\",\"";
  cmd += "184.106.153.149"; // api.thingspeak.com IP address
  cmd += "\",80";
  ESP8266.println(cmd);
  Serial.print("Start Commands: ");
  Serial.println(cmd);

  if(ESP8266.find("Error"))
  {
    Serial.println("AT+CIPSTART error");
    return;
  }
}
```

Code-6

```
String GetThingspeakcmd(String getStr)
{
  String cmd = "AT+CIPSEND=";
  cmd += String(getStr.length());
  ESP8266.println(cmd);
  Serial.println(cmd);
if(ESP8266.find(">"))
  {
    ESP8266.print(getStr);
    Serial.println(getStr);
    delay(500);
    String messageBody = "";
    while (ESP8266.available())
    {
      String line = ESP8266.readStringUntil('\n');
      if (line.length() == 1)
      {
        messageBody= ESP8266.readStringUntil('\n');
      }
    }
    Serial.print("MessageBody received: ");
    Serial.println(messageBody);
    return messageBody;
  }
  else
  {
    ESP8266.println("AT+CIPCLOSE");
    Serial.println("AT+CIPCLOSE");
  }
}
```

Before you upload the code

There are a few things to do in the code before you upload it.

- Paste your **Write API key from Thingspeak** here:

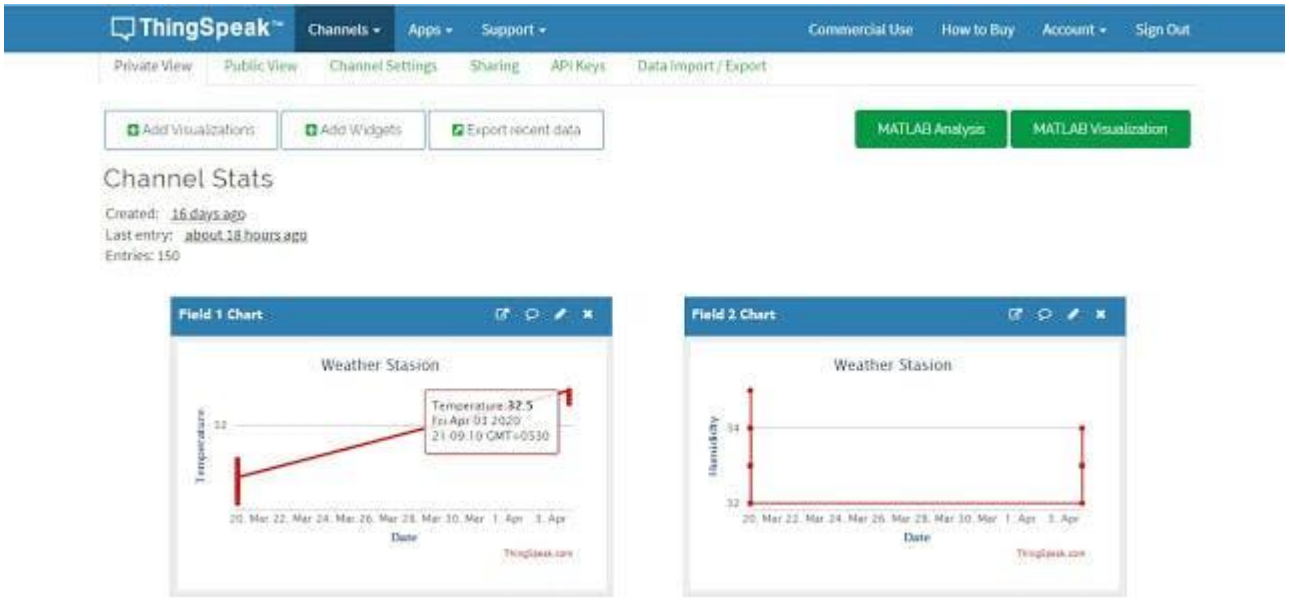
```
String myAPIkey = "Your Write API key here";
```

- Enter your **Wi-Fi SSID** and **Password** here:

```
ESP8266.print("AT+CWJAP=\"Wifi Network's name\",\"PassWord\"\r\n");
```

Final step

Upload the code. If all was done correctly, your Thingspeak Channel should look like this.



Congratulations, you just completed an IOT based task.