Item needed:

Smartphone that already installed Telegram

1x NodeMCU ESP32

1x DHT22 sensor

1x 10k Ohm resistor

8X jumper wire

1x power bank/power plug

2x bread board.

Component Assemble step:

1. Assemble the component as shown in the picture.

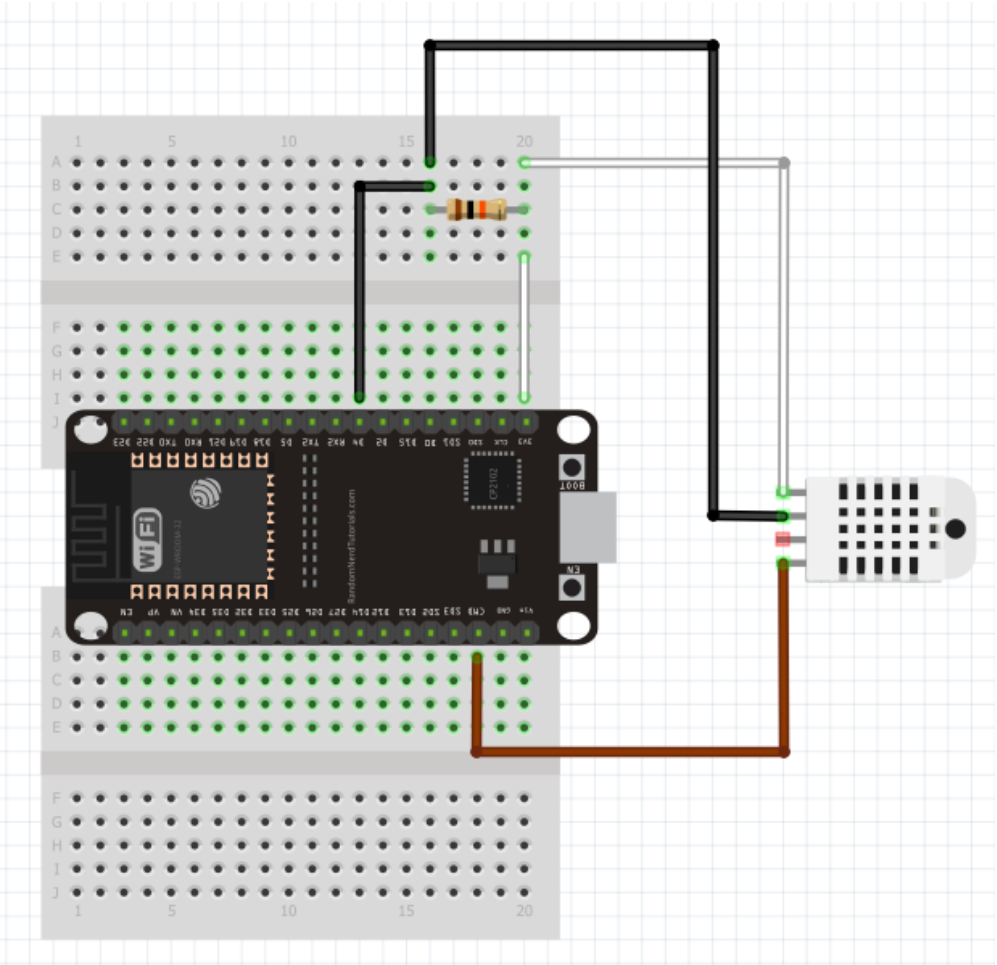

Figure 1: Component layout in Fritzing.

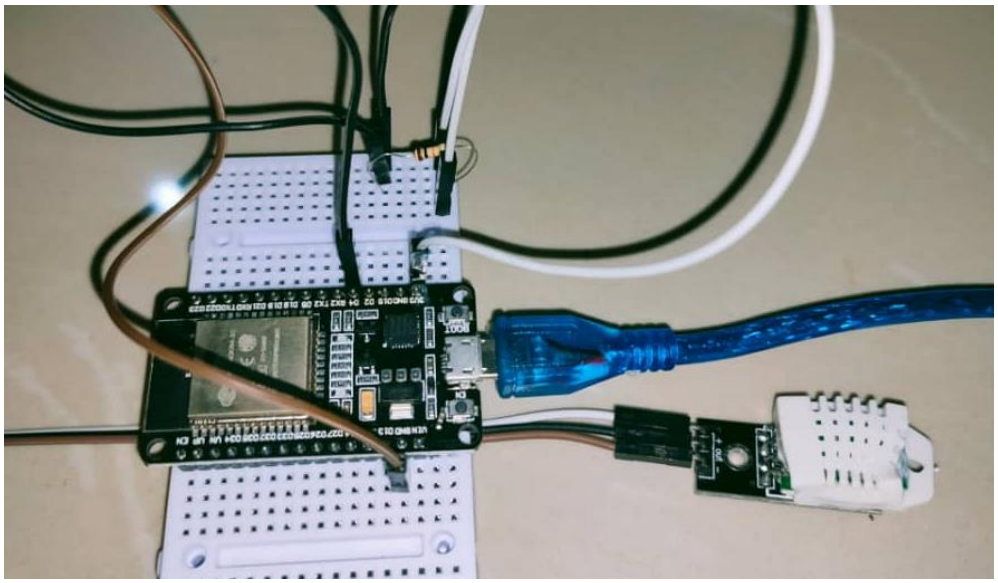

Figure 2: Actual component built.

Create Telegram bot step:

1. Download and install Telegram to your smartphone.
2. Search for BotFather and start creating your own Telegram bot

Figure 3: Description of BotFather.

3. Start the Botfather and choose the command /newbot to create a new bot.

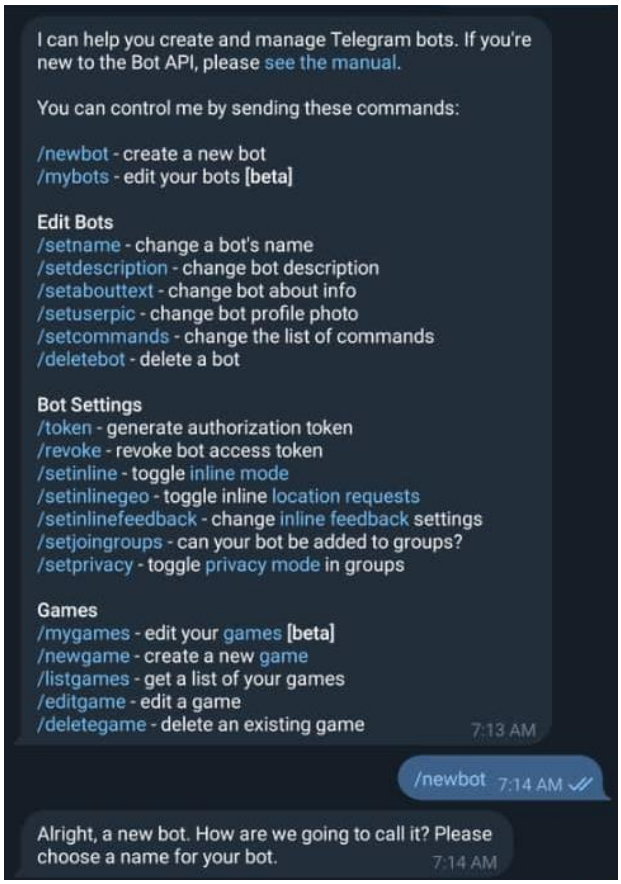

Figure 4: List of available Commands

4. Follow the guide by giving your bot a name and a username that must be end with"bot"
5. Once your done, remember to get your bot API Key and put change it inside the code later.
6. You are able to find the bot by searching using the username that has been set.

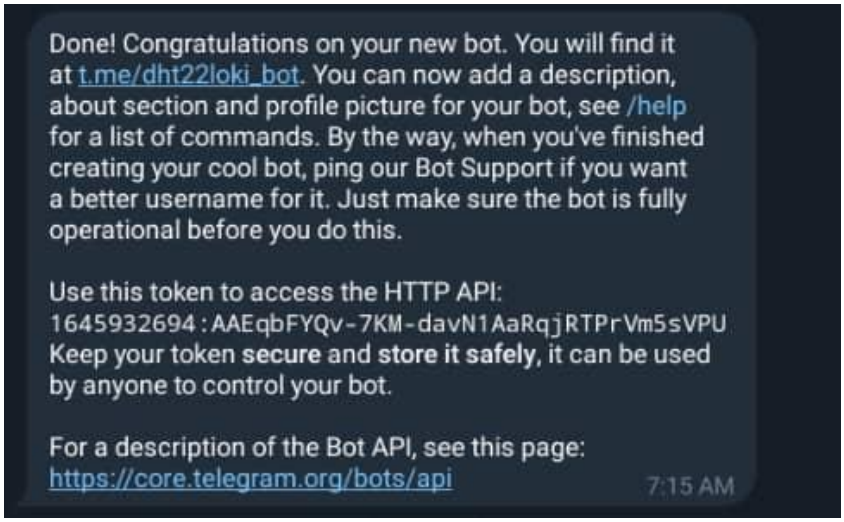

Figure 5: finish creating bot and get the API key.

Arduino IDE step:

1. Open Arduino IDE.
2. Go to files and click preference in the Arduino IDE.

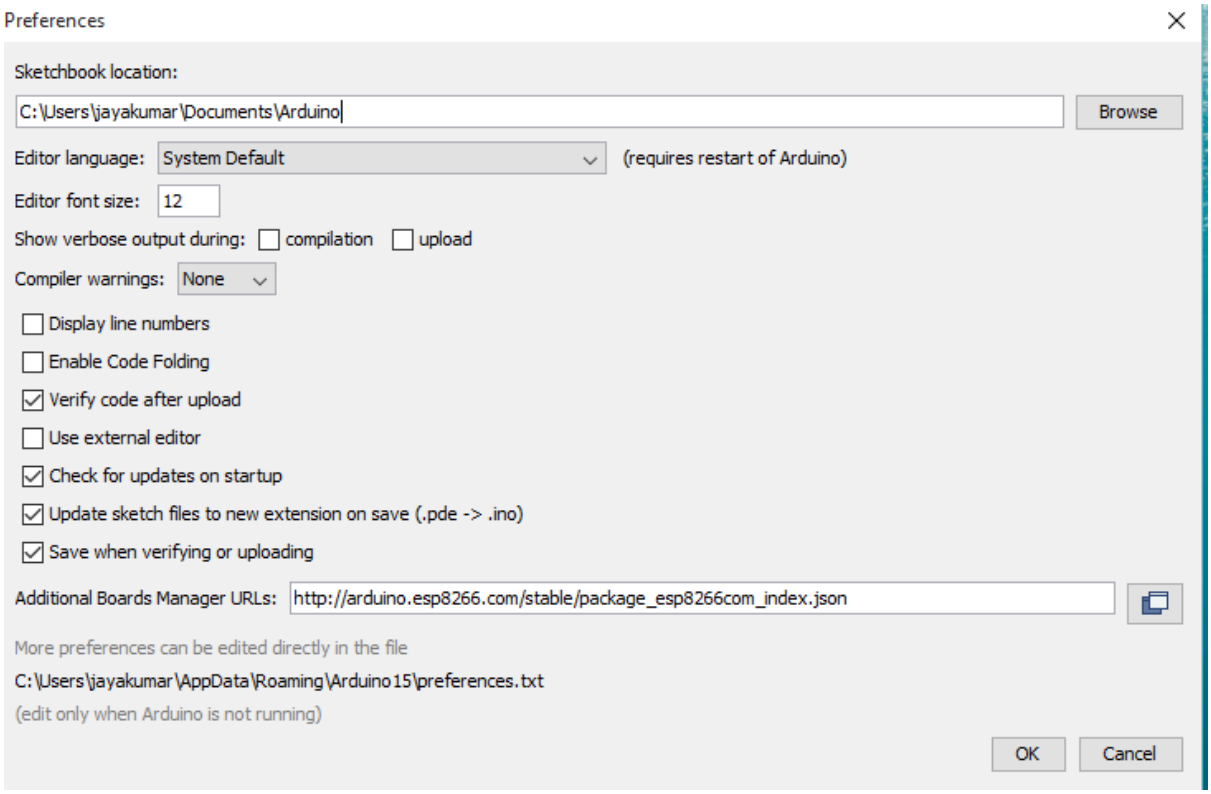

Figure 6: preference in Arduino IDE.

1. copy the below code in the Additional boards Manager

   https://dl.espressif.com/dl/package_esp32_index.json,
   http://arduino.esp8266.com/stable/package_esp8266com_index.json

2. click OK to close the preference Tab.

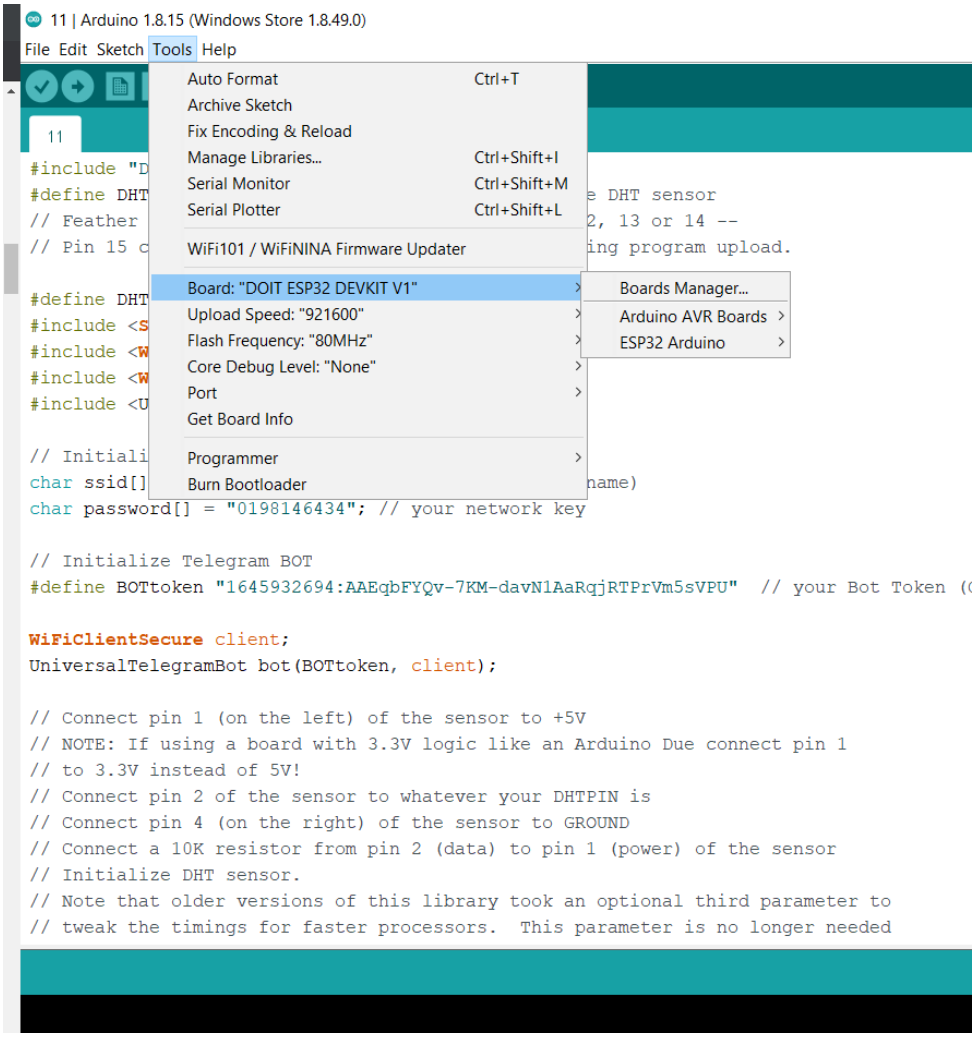

Figure 7: search for board manager.

3. After completing the above steps, go to Tools and board, and then select board Manager.

Figure 8: Board manager

Search for ESP32 and press install button for the “ESP32 by Espressif Systems“

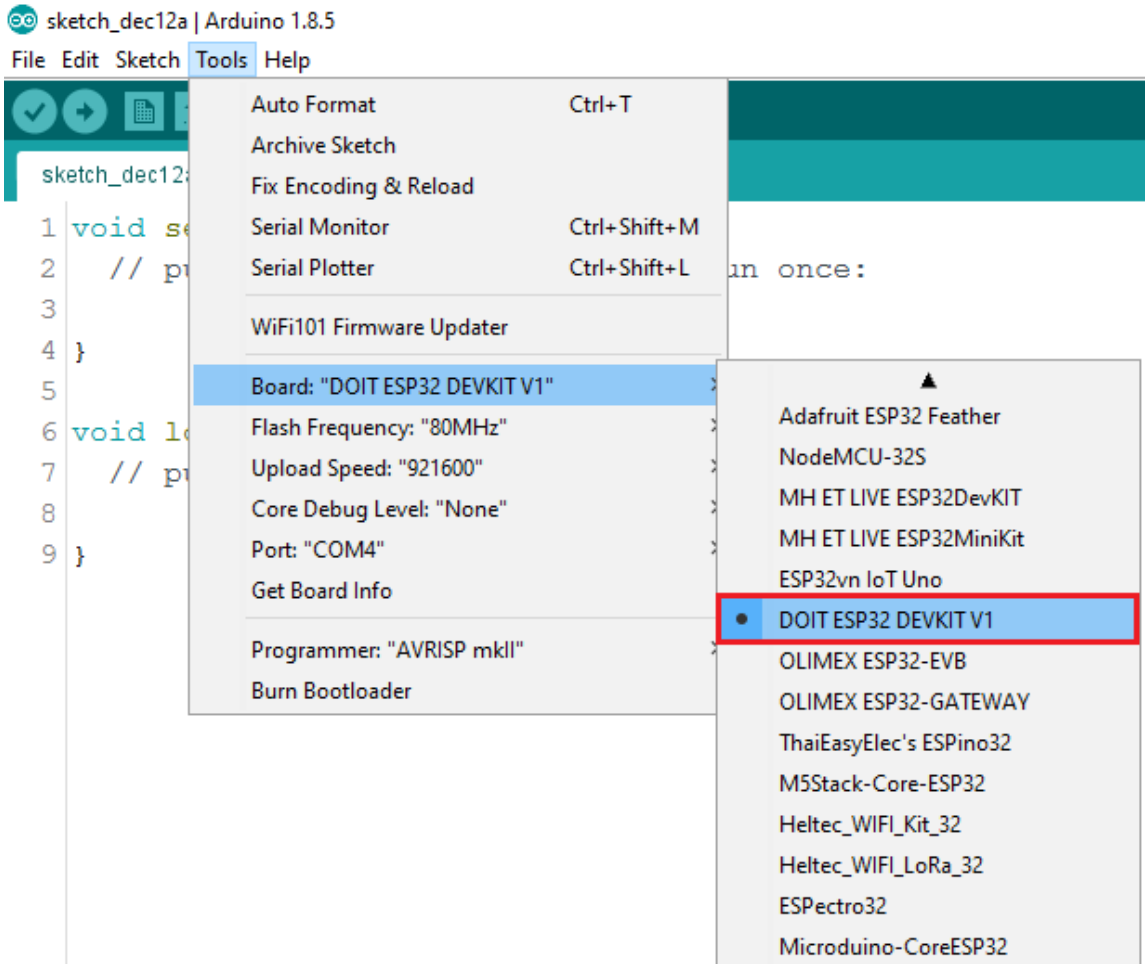

Figure 9: verified the board is same.

Installing important libaries

Go to Tools and click Manage Libaries in Ardiuno IDE

4. Search for “DHT” on the Search box and install the DHT library from Adafruit.

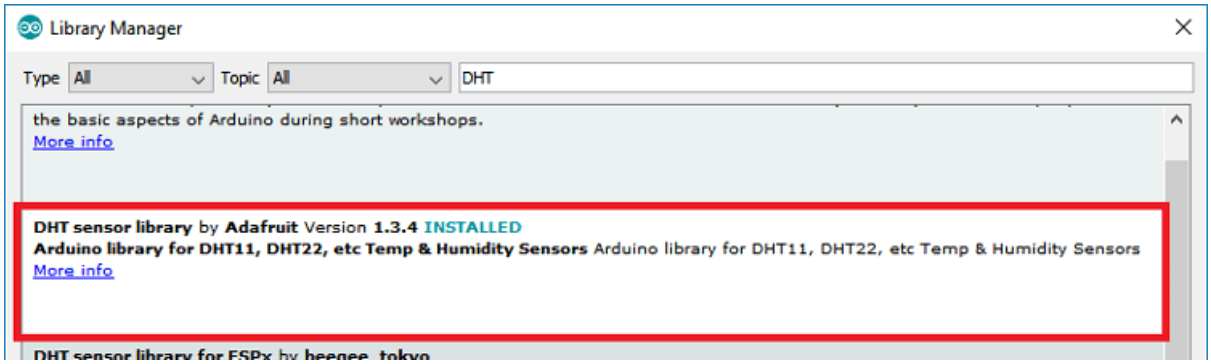

Figure 10: Install DHT sensor Library by Adafruit.

5. After installing the DHT library from Adafruit, type “Adafruit Unified Sensor” in the search box. Scroll all the way down to find the library and install it.

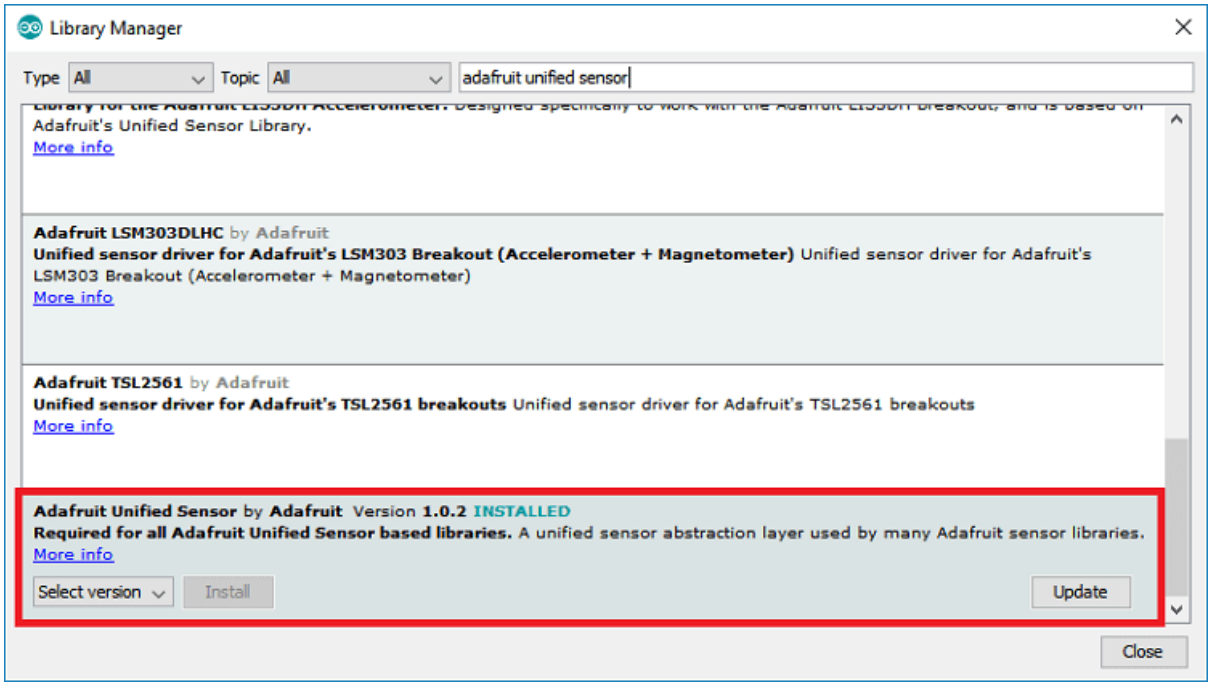

Figure 11: Install Adafruit Unified Sensor

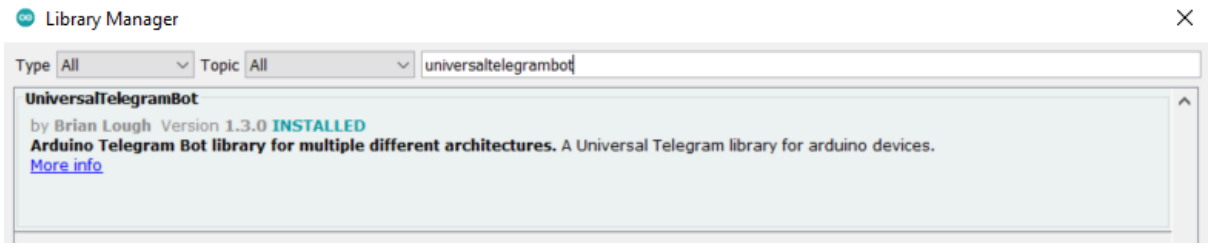

Figure 12: Install UniversalTelegramBot

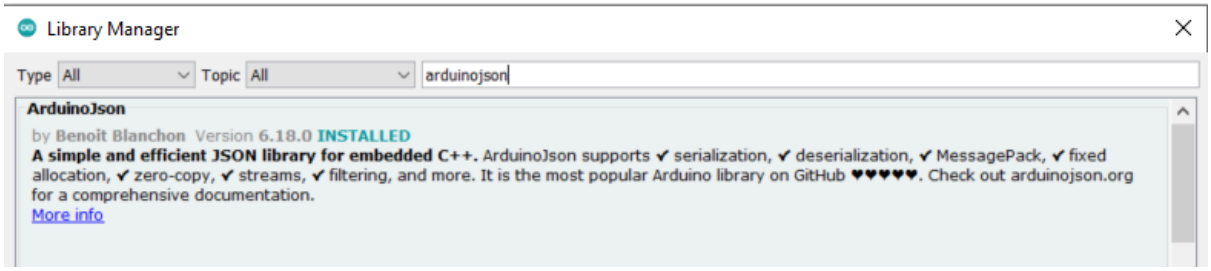

Figure 13: Install ArduinoJson

Figure 14: Install ThingSpeak

6. After installing the libraries, restart your Arduino IDE.

7. Connect the component to the correct port.
8. Copy the following code to the Arduino IDE

```
#include "DHT.h"
#define DHTPIN 4     // Digital pin connected to the DHT sensor
#define DHTTYPE DHT22
#include <SPI.h>
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>

const char ssid[] = "*******"; // your network SSID (name)
const char password[] = "************"; // your network key

WiFiClient client;

// Initialize Telegram BOT
#define BOTtoken "****************************************"  // your Bot
Token (Get from Botfather)


WiFiClientSecure net_ssl;
UniversalTelegramBot bot(BOTtoken, net_ssl);

#include <ThingSpeak.h>
const long CHANNEL = 1420049;
const char *WRITE_API = "9LE30HTX2O0WWX9H";

DHT dht(DHTPIN, DHTTYPE);

                    //parameter
                    // Read humidity as percentage
                      float humidity = dht.readHumidity();
                    // Read temperature as Celsius (the default)
                      float temperature = dht.readTemperature();
                              // Compute heat index in Celsius (isFahreheit = false)
                      float hic = dht.computeHeatIndex(temperature, humidity, false);
//for sensor interval
long prevMillisSensor = 0;
int intervalSensor = 2000;
//for ThingSpeak interval
long prevMillisThingSpeak = 0;
int intervalThingSpeak = 1.8e+6; // 20000 Minimum ThingSpeak write interval is
15 seconds

//telegram bot interval
int botRequestDelay = 500;
unsigned long lastTimeBotRan;
```

```
//message function
void handleNewMessages(int numNewMessages) {
Serial.println("handleNewMessages");
Serial.println(String(numNewMessages));

for ( int i=0; i<numNewMessages; i++) {
 String chat_id = String(bot.messages[i].chat_id);
  String text = bot.messages[i].text;
  String statusText = "";
                    // Read humidity as percentage
                      float humidity = dht.readHumidity();
                    // Read temperature as Celsius (the default)
                      float temperature = dht.readTemperature();
                    // Compute heat index in Celsius (isFahreheit = false)
                      float hic = dht.computeHeatIndex(temperature, humidity, false);

  String from_name = bot.messages[i].from_name;
  if (from_name == "") from_name = "Guest";

   if (text == "/start") {
    String welcome = "ESP32 controller, " + from_name + ".\n";
    welcome += "Welcome to Akmal's Switly Bot\n";
    welcome += "/status : view current all parameter\n";
    welcome += "/temperature : view current Temperature\n";
    welcome += "/humidity : view current Humidity\n";
    welcome += "/onFan : view current Humidity\n";
    welcome += "/offFan : Off the mist fan\n\n";
    welcome += "view the graph at\n";
    welcome += "https://thingspeak.com/channels/1420049";
    bot.sendMessage(chat_id, welcome, "Markdown");
   }

   if (text == "/status") {
      statusText = "Temperature: " + String(temperature) + " °C" +
                   "\nHumidity: " + String(humidity) + " %"
                   "\nHeat Index: " + String(hic) + " °C";
      bot.sendMessage(chat_id, statusText, "");
   }
    if (text == "/temperature") {
      statusText = "Temperature: " + String(temperature) + " °C";
      bot.sendMessage(chat_id, statusText, "");
   }

   if (text == "/humidity") {
      statusText ="Humidity: " + String(humidity) + " %";
      bot.sendMessage(chat_id, statusText, "");
   }
  }
}
```

```
void setup()
{
  Serial.begin(115200);
  Serial.println();
  Serial.println("Akmal Aiman FYP' Swiftly Bot");
  Serial.println();

//begin telegram
  net_ssl.setInsecure();

  WiFi.mode(WIFI_STA);


  ThingSpeak.begin(client);  // begin ThingSpeak
   dht.begin();// begin dht22 sensor
}

void loop()
{
  // Connect or reconnect to WiFi
  if (WiFi.status() != WL_CONNECTED) {
    Serial.print("Attempting to connect to SSID: ");
    Serial.println(ssid);
    while (WiFi.status() != WL_CONNECTED) {
      WiFi.begin(ssid, password);
      Serial.print(".");
      delay(5000);
    }
    Serial.println("\nConnected.");
  }

  //sensor reading
  if (millis() - prevMillisSensor > intervalSensor) {

    temperature = dht.readTemperature();
    Serial.print("Temperature: ");
    Serial.print(temperature);
    Serial.print(" °C  ");

    humidity = dht.readHumidity();
    Serial.print("Humidity: ");
    Serial.print(humidity);
    Serial.print("%  ");

    hic = dht.computeHeatIndex(temperature, humidity, false);
    Serial.print("° Heat index: ");
    Serial.print(hic);
    Serial.println("");
```

```
  prevMillisSensor = millis();
 }


                //commands checking for telegram bot
                if (millis() > lastTimeBotRan + botRequestDelay)  {
                 int numNewMessages = bot.getUpdates(bot.last_message_received
+ 1);

                 while(numNewMessages) {
                  Serial.println("got response");
                  handleNewMessages(numNewMessages);
                  numNewMessages = bot.getUpdates(bot.last_message_received +
1);
                 }
                 lastTimeBotRan = millis();
                }



 //sending the parameter to ThingSpeak with the interval of 20second(ThingSpeak
min is 15 second)
 if (millis() - prevMillisThingSpeak > intervalThingSpeak) {

  // Set the fields with the values
  ThingSpeak.setField(1, temperature);
  ThingSpeak.setField(2, humidity);

  // Write to the ThingSpeak channel
  int x = ThingSpeak.writeFields(CHANNEL, WRITE_API);
  if (x == 200) {
   Serial.println("Channel update successful.");
  }
  else {
   Serial.println("Problem updating channel. HTTP error code " + String(x));
  }

  prevMillisThingSpeak = millis();
 }

          //alert notification checking
          int alert;{
           int a=0;
           String chat_id = String(bot.messages[a].chat_id);
           String text = bot.messages[a].text;
           String statusText = "";

         //temperature check ideal  is 27-32
```

```
                if(temperature > 32){
                statusText ="the farm is hot!! Turn the Mist Fan on to keep it cool.
current temperature: " + String(temperature) + "°C";
                Serial.println("the farm is hot, Turn the Fan on to keep it cool. in loop");
                bot.sendMessage(chat_id, statusText, "");
                }
                else if(temperature < 27){
                statusText ="the farm is cold!! Turn the Mist Fan off to keep it warm.
current temperature: " + String(temperature) + "°C";
                Serial.println("the farm is cold, Turn the Fan on to keep it warm. in
loop");
                bot.sendMessage(chat_id, statusText, "");
                }

              //Humidity check ideal is 80-95
                if(humidity > 95){
                statusText ="the farm is too humid!! Turn the Mist Fan off to keep it dry.
current humidity : " + String(humidity) + "% ";
                Serial.println("the farm is too humid!! Turn the Mist Fan off to keep it
dry.  in loop");
                bot.sendMessage(chat_id, statusText, "");
                }
                else if(humidity  < 80){
                statusText ="the farm is too dry!! Turn the Mist Fan on to keep it humid.
current humidity : " + String(humidity) + "% ";
                Serial.println("the farm is too dry!! Turn the Mist Fan on to keep it
humid. in loop");
                bot.sendMessage(chat_id, statusText, "");
                }
              }

}
```

9. Change the Nework SSID and Key to your WiFi name and password to allow it connect to the internet.
10. Change the telegram API token key to your own API token key.
11. Upload the code to ESP32 and the system is on.