A project Report

On

Greenhouse monitoring and control System using IOT

by:

Vanshika Chanderiya

Introduction:

This project is an IOT based solution for your greenhouse, nursery or garden. We will be using some basic sensors to make a project which is feasible and easy to handle. Here we are measuring humidity, Temperature, Soil Moisture and luminosity using integrating the sensors that serve the purpose. This will help in monitoring and maintaining proper condition for the Greenhouse.

We've set some conditions to check the if the temperature is below the suitable condition or light available is less, according to this the lights get switched on automatically.

Components used:

Hardware:

- NodeMCU
- DHT11
- LDR
- Soil Moisture Sensor
- LEDs
- Resistors
- Jumper cables
- Bread board

Software:

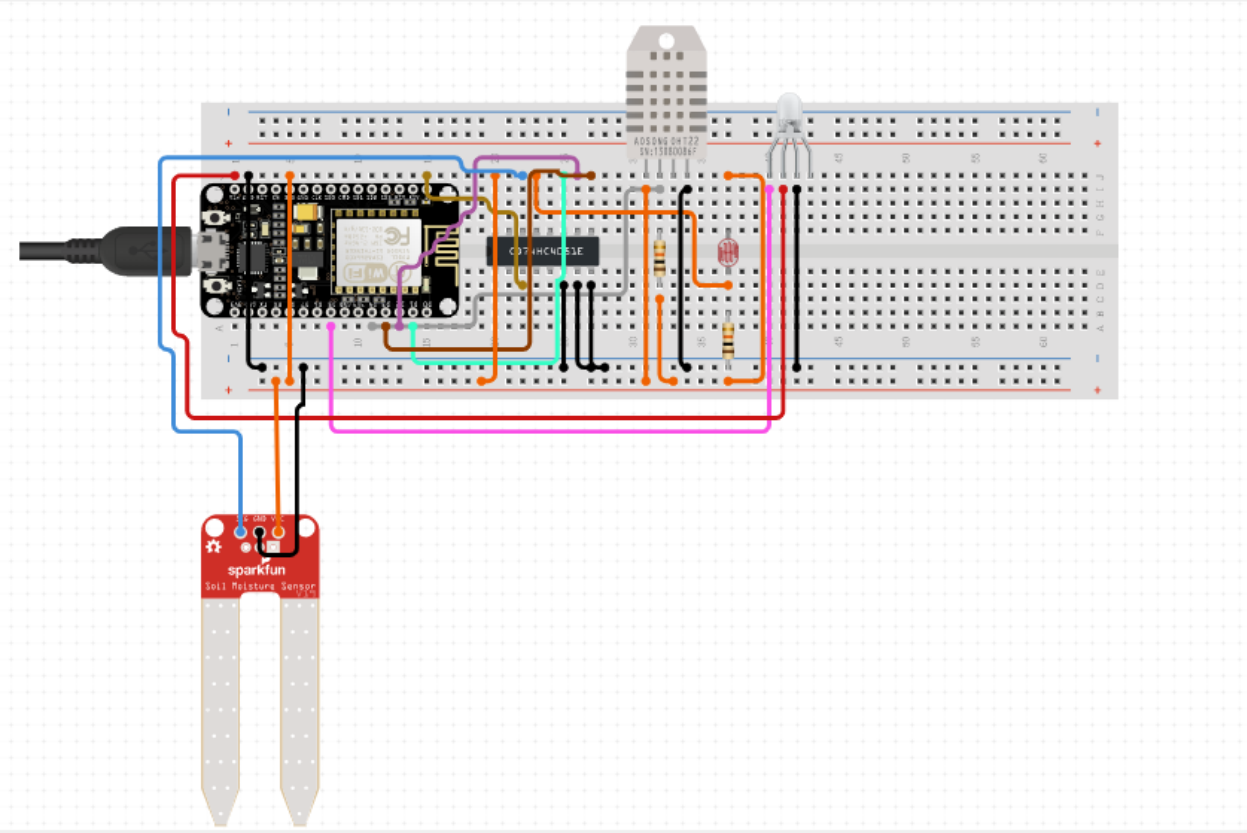
- Arduino IDE
- Blynk app

Other components:

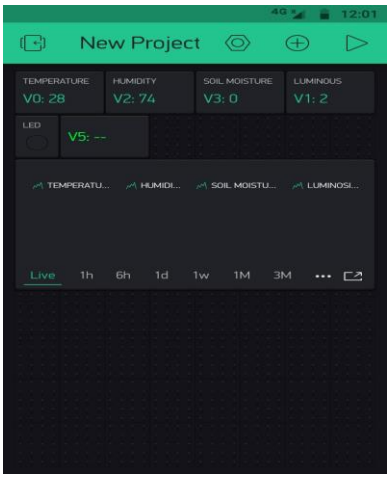
- Android phone
- Raspberry pi setup system for Fog Computation

Circuit Diagram and Working:

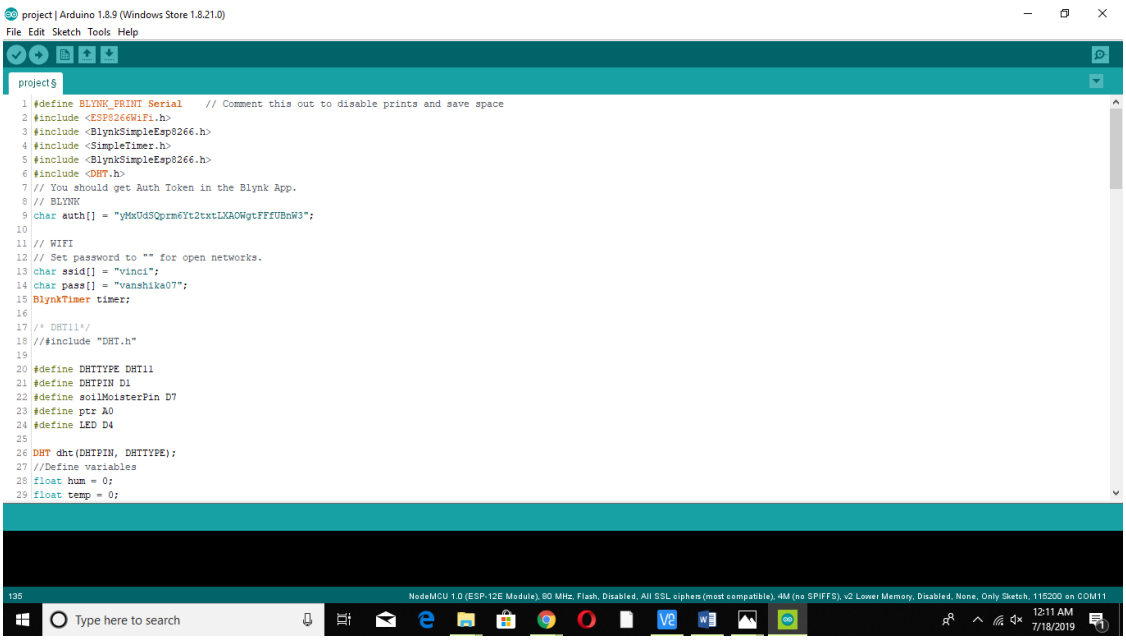
- First step is setting up the sensors accordingly and integrate the whole setup in a proper working environment. The circuit diagram is shown below:

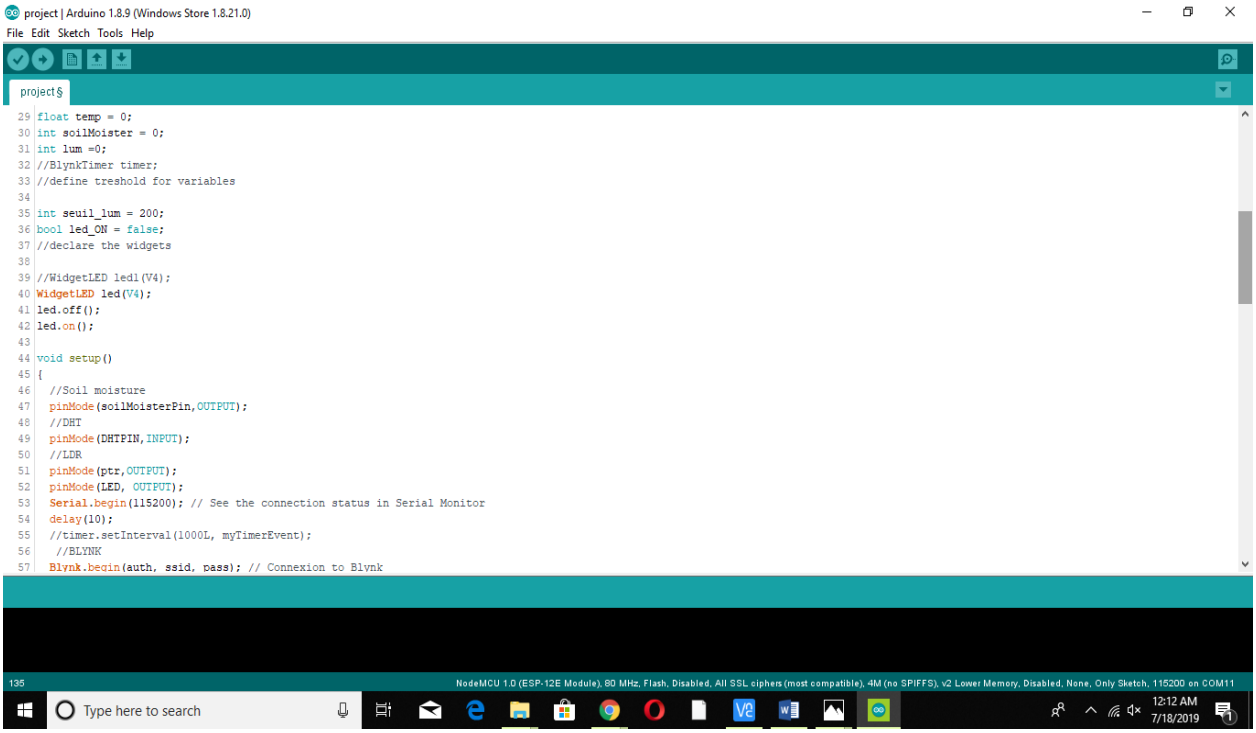


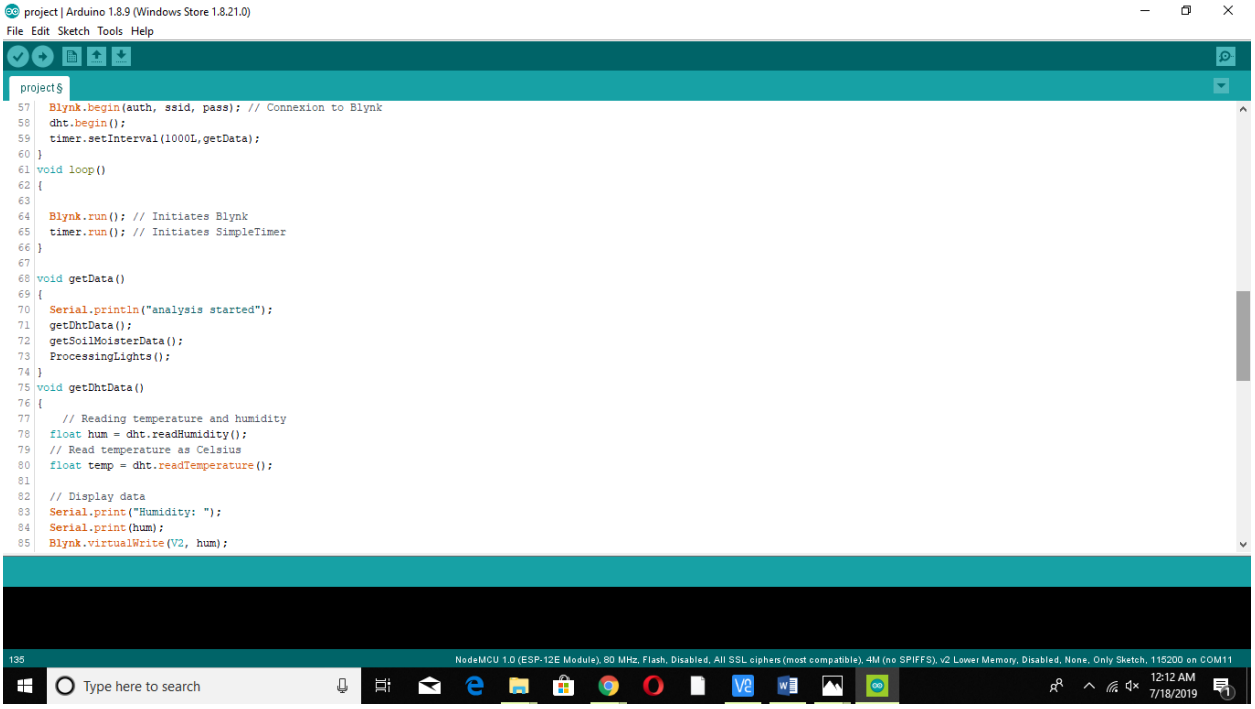
- Once it is done, set up the Blynk app in your android mobile and make 4 visual displays, one LED and one smart chart for the monitoring purpose.

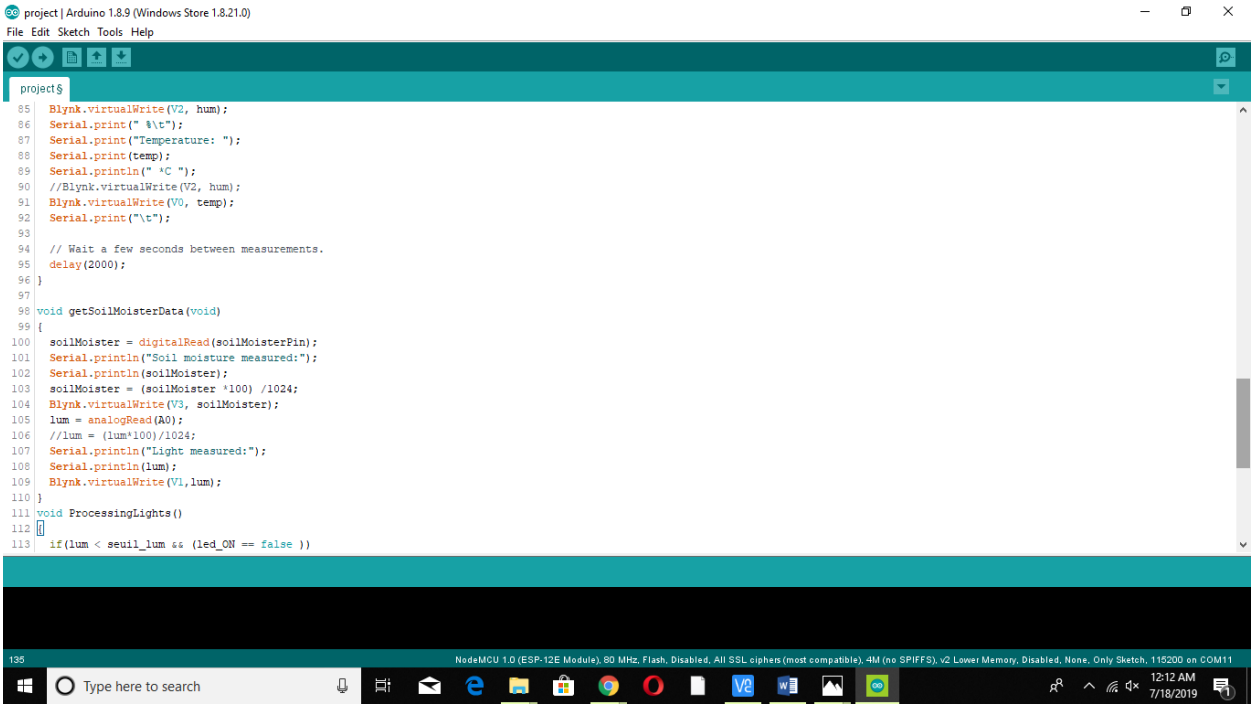


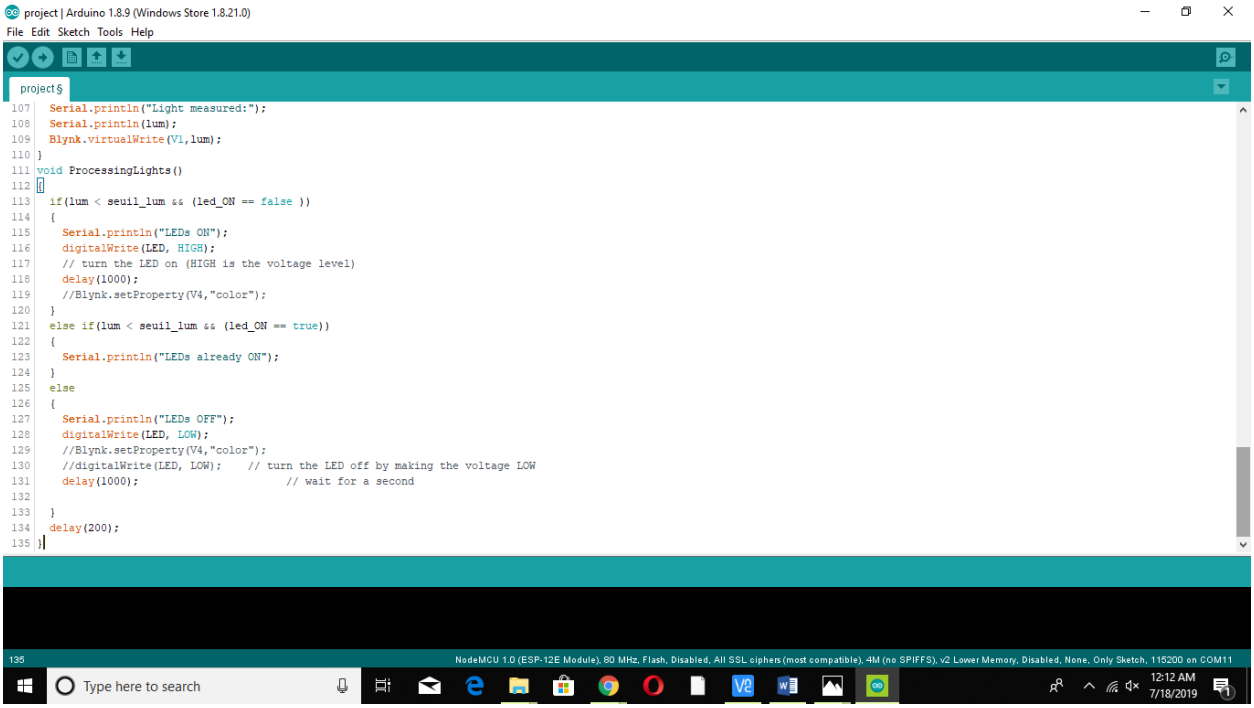
- After this, coding in Arduino IDE and uploading it in NodeMCU. The code for the following setup is :



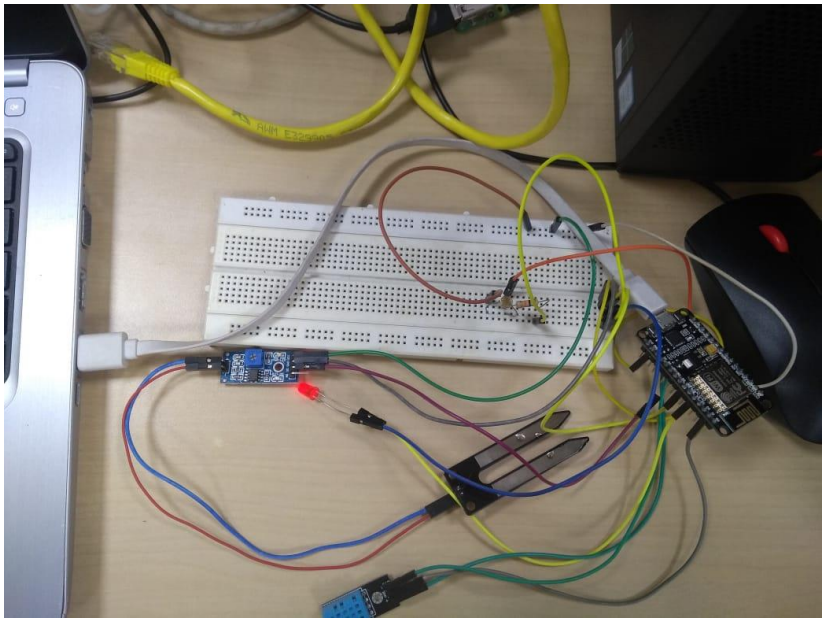

project | Arduino 1.8.9 (Windows Store 1.8.21.0)
File Edit Sketch Tools Help
project §
29 float temp = 0;
30 int soilMoister = 0;
31 int lum =0;
32 //BlynkTimer timer;
33 //define treshold for variables
34
35 int seuil_lum = 200;
36 bool led_ON = false;
37 //declare the widgets
38
39 //WidgetLED led1(V4);
40 WidgetLED led(V4);
41 led.off();
42 led.on();
43
44 void setup()
45 {
46 //Soil moisture
47 pinMode(soilMoisterPin,OUTPUT);
48 //DHT
49 pinMode(DHTPIN,INPUT);
50 //LDR
51 pinMode(ptr,OUTPUT);
52 pinMode(LED, OUTPUT);
53 Serial.begin(115200); // See the connection status in Serial Monitor
54 delay(10);
55 //timer.setInterval(1000L, myTimerEvent);
56 //BLYNK
57 Blynk.begin(auth, ssid, pass); // Connexion to Blynk
135
NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, Disabled, All SSL ciphers (most compatible), 4M (no SPIFFS), v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM11
Type here to search
12:12 AM
7/18/2019

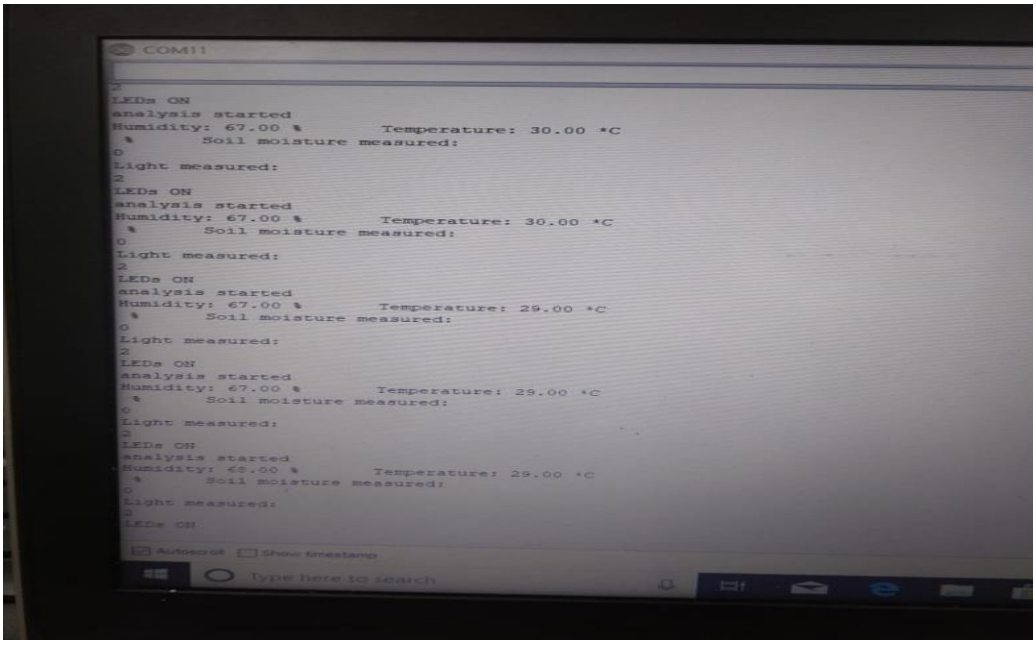

project | Arduino 1.8.9 (Windows Store 1.8.21.0)
File Edit Sketch Tools Help
project §
57 Blynk.begin(auth, ssid, pass); // Connexion to Blynk
58 dht.begin();
59 timer.setInterval(1000L,getData);
60 }
61 void loop()
62 {
63
64 Blynk.run(); // Initiates Blynk
65 timer.run(); // Initiates SimpleTimer
66 }
67
68 void getData()
69 {
70 Serial.println("analysis started");
71 getDhtData();
72 getSoilMoisterData();
73 ProcessingLights();
74 }
75 void getDhtData()
76 {
77 // Reading temperature and humidity
78 float hum = dht.readHumidity();
79 // Read temperature as Celsius
80 float temp = dht.readTemperature();
81
82 // Display data
83 Serial.print("Humidity: ");
84 Serial.print(hum);
85 Blynk.virtualWrite(V2, hum);
135
NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, Disabled, All SSL ciphers (most compatible), 4M (no SPIFFS), v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM11
Type here to search
12:12 AM
7/18/2019


project | Arduino 1.8.9 (Windows Store 1.8.21.0)
File Edit Sketch Tools Help
project §
Blynk.virtualWrite(V2, hum);
Serial.print(" %\t");
Serial.print("Temperature: ");
Serial.print(temp);
Serial.println(" *C ");
//Blynk.virtualWrite(V2, hum);
Blynk.virtualWrite(V0, temp);
Serial.print("\t");
// Wait a few seconds between measurements.
delay(2000);
}
void getSoilMoisterData(void)
{
soilMoister = digitalRead(soilMoisterPin);
Serial.println("Soil moisture measured:");
Serial.println(soilMoister);
soilMoister = (soilMoister *100) /1024;
Blynk.virtualWrite(V3, soilMoister);
lum = analogRead(A0);
//lum = (lum*100)/1024;
Serial.println("Light measured:");
Serial.println(lum);
Blynk.virtualWrite(V1,lum);
}
void ProcessingLights()
{
if(lum < seuil_lum && (led_ON == false))
135
NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, Disabled, All SSL ciphers (most compatible), 4M (no SPIFFS), v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM11
Type here to search
12:12 AM
7/18/2019

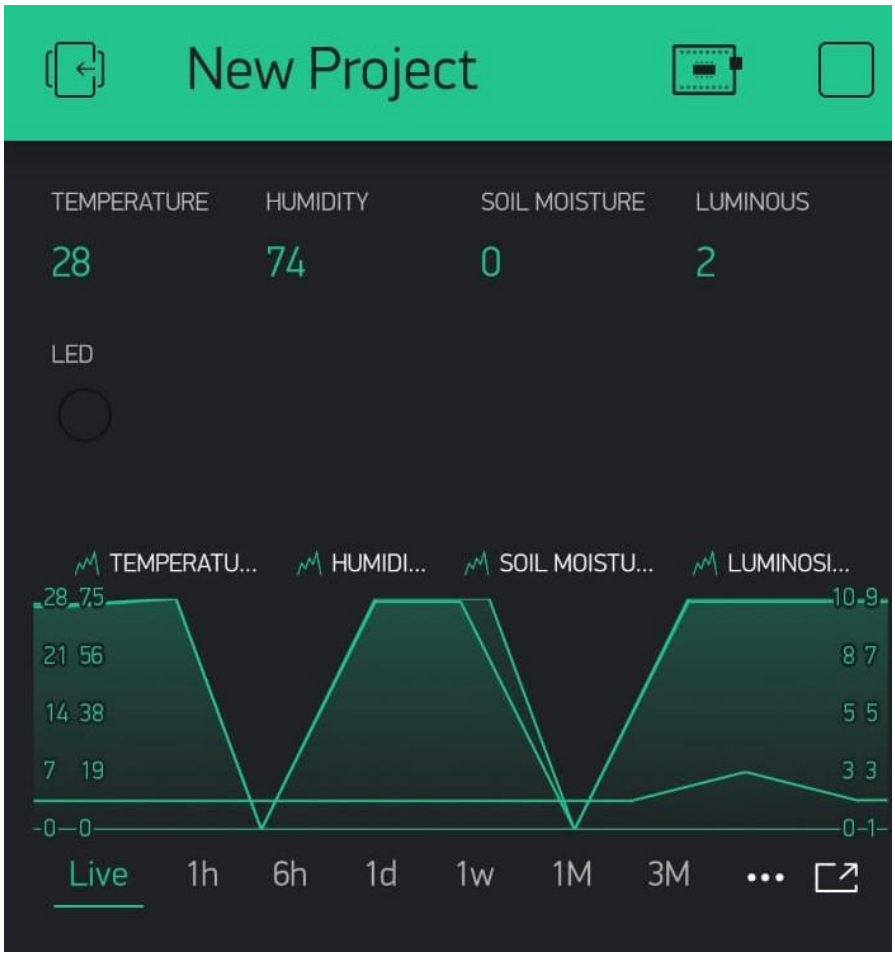

project | Arduino 1.8.9 (Windows Store 1.8.21.0)
File Edit Sketch Tools Help
project §
Serial.println("Light measured:");
Serial.println(lum);
Blynk.virtualWrite(V1,lum);
}
void ProcessingLights()
{
if(lum < seuil_lum && (led_ON == false))
{
Serial.println("LEDs ON");
digitalWrite(LED, HIGH);
// turn the LED on (HIGH is the voltage level)
delay(1000);
//Blynk.setProperty(V4,"color");
}
else if(lum < seuil_lum && (led_ON == true))
{
Serial.println("LEDs already ON");
}
else
{
Serial.println("LEDs OFF");
digitalWrite(LED, LOW);
//Blynk.setProperty(V4,"color");
//digitalWrite(LED, LOW); // turn the LED off by making the voltage LOW
delay(1000); // wait for a second
}
delay(200);
}
135
NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, Disabled, All SSL ciphers (most compatible), 4M (no SPIFFS), v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM11
Type here to search
12:12 AM
7/18/2019

- Now that coding part is done, let's see the result. The pictures below are of the setup I had and the results shown by the sensors :





The following picture is of the live data that was getting stored on the cloud with it's graph.



Result:

The following setup serves the purpose of general monitoring of your greenhouse along with warnings and auto lumination when required. This can help you save a lot of time and can contribute in a better growth of plants with minimal effort and cost.

Future Scope:

The project can further be integrated with acctuators like water pump that can be turned on automatically whenever the Soil Moisture content goes below the average level and fans can also be implemented according to the situation.

GitHub link for the same : https://github.com/vanshikachanderiya/IOTProject