

IOT PROJECT – GREENHOUSE AUTOMATION USING IOT

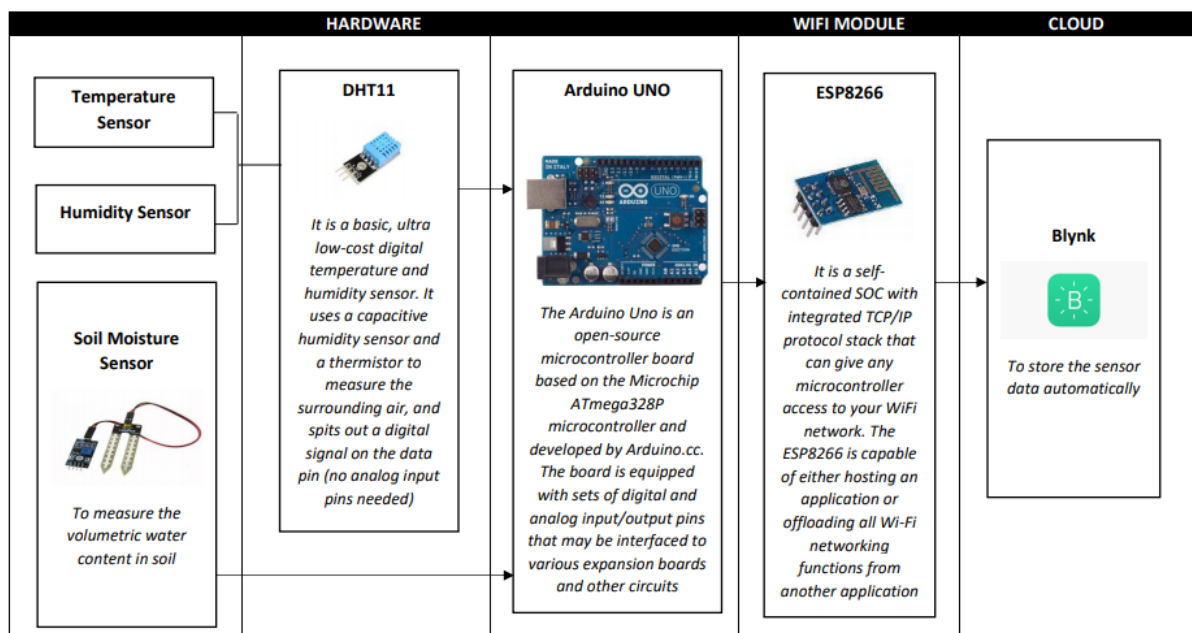
Prepared by Nurul Addina Binti Khairuddin

Objective of the project is to develop a Remote Monitoring System, which can detect Soil moisture levels from Ground and Temperature, Humidity from atmosphere. The system will send data in regular intervals of time to Cloud. Data should be stored on the cloud and reports should be generated based on sensor data coming in real time.

List of Deliverables

1.1 - Architecture Diagram & 3.1 – Connection and Programs

Architecture Diagram



CONNECTION DETAILS

Arduino UNO	ESP8266	DHT11	Soil Moisture
3V3	VCC, CH_PD, RST	VCC	-
RESET	GND	-	-
GND	GND	GND	GND
TXD	TXD	-	-
RXD	RXD	-	-
-	GPIO_2	DATA	-
5V	-	-	VCC
A1	-	-	A0

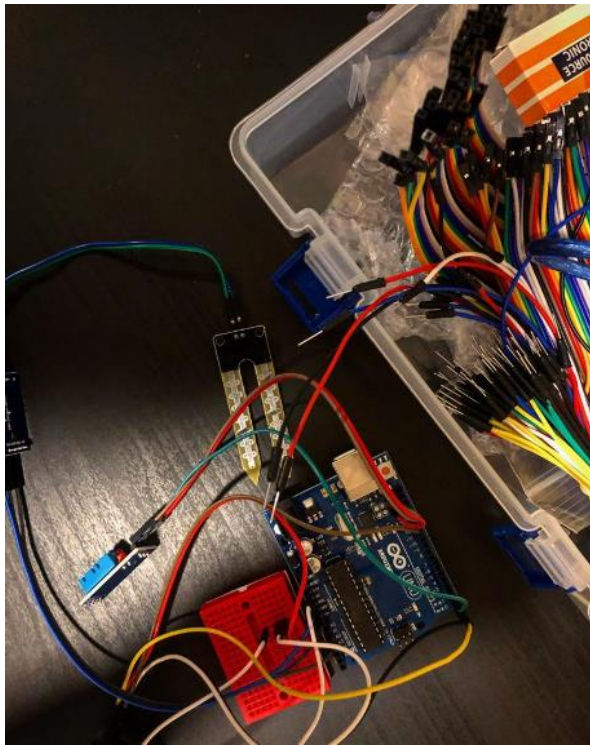
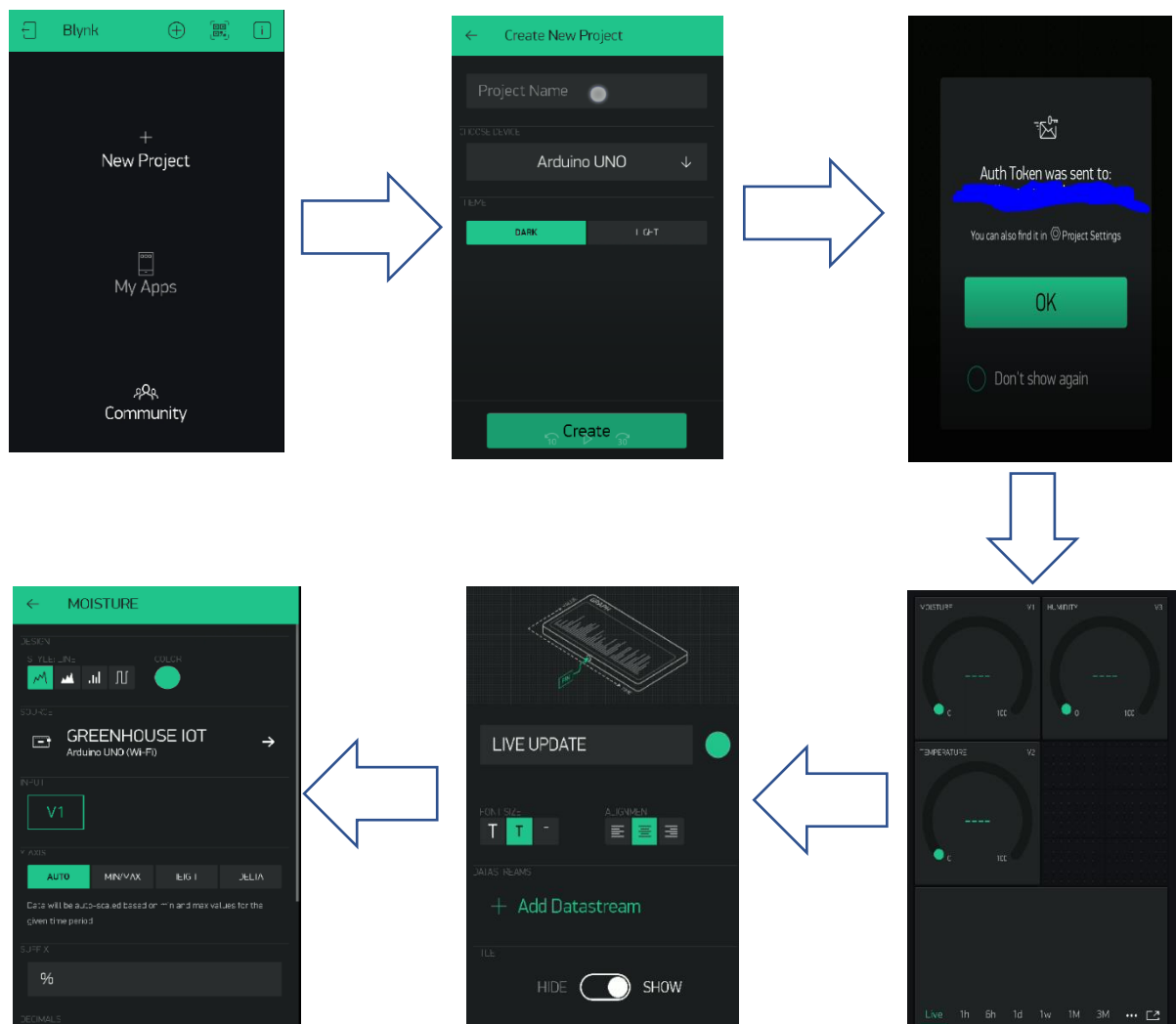
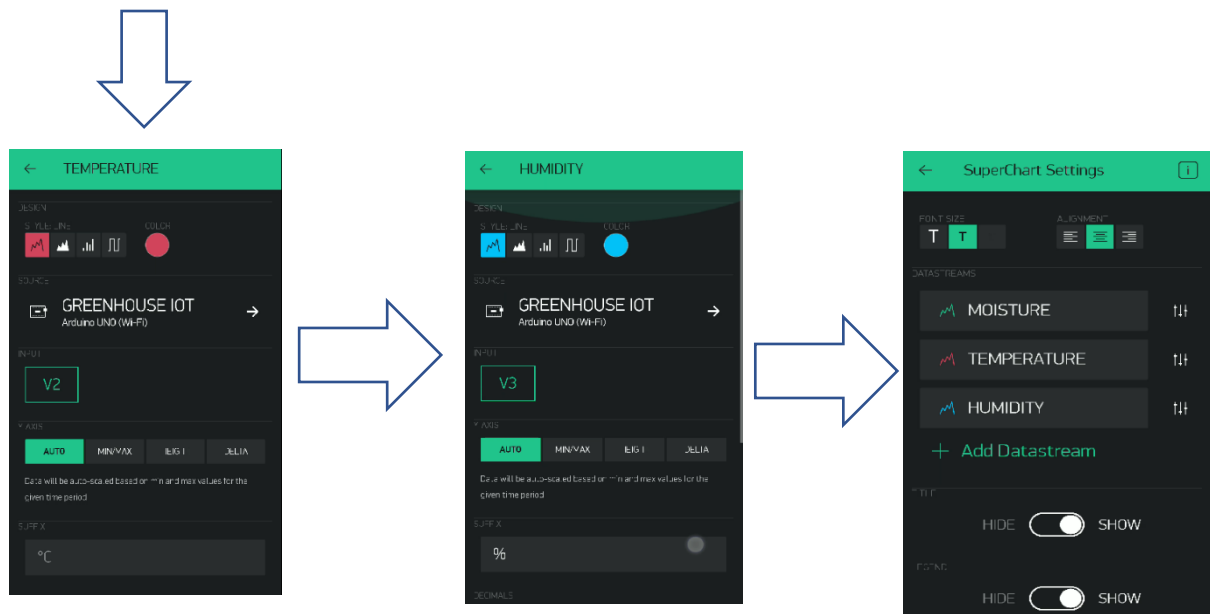


Figure 1 shows that both DHT11 and Soil Moisture sensors; and ESP8266 (Wifi Module) are connected to Arduino Uno

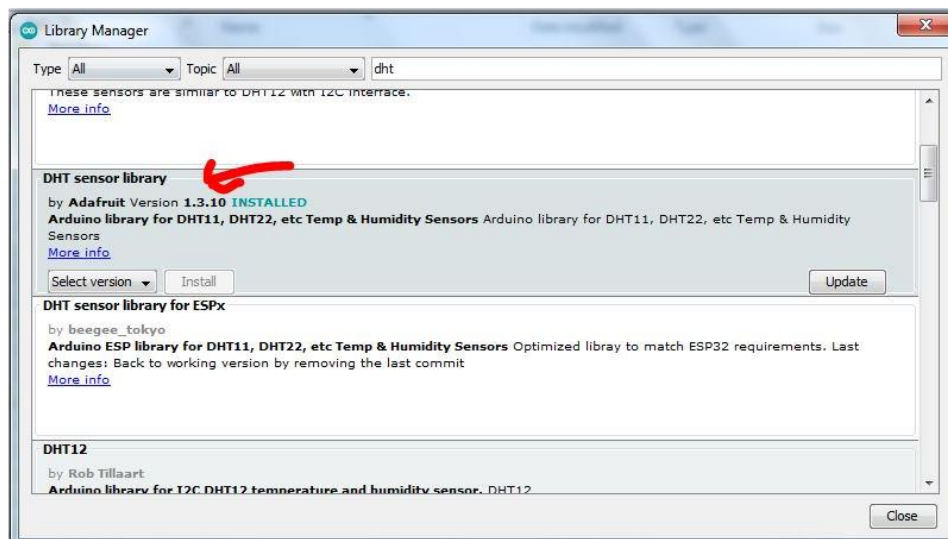
Figure 1

Creating Project in Blynk Appdlication

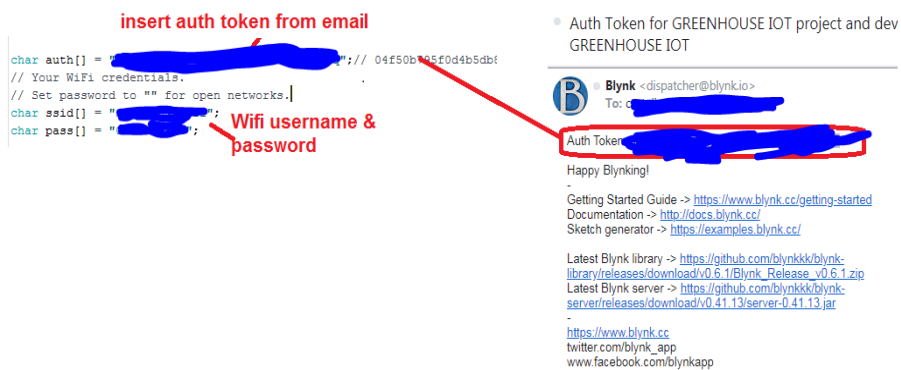




Downloading DHT Sensor Library



Inserting Auth Token from email to code in Arduino IDE



Uploading code into Arduino Uno via Arduino Uno

```
greenhouse | Arduino 1.8.13 (Windows Store 1.8.42.0)
File Edit Sketch Tools Help

greenhouse
#include <BlynkSimpleShieldEsp8266.h>
#include <ESP8266_Lib.h>
#include "DHT.h"
#include <SoftwareSerial.h>

char auth[] = "XXXXXXXXXX";
// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "XXXXXXXXXX";
char pass[] = "XXXXXXXXXX";

SoftwareSerial pc(2, 3);
// Your ESP8266 baud rate:
#define ESP8266_BAUD 115200

#define BLYNK_PRINT pc
#define EspSerial Serial
ESP8266 wifi(&EspSerial);

#define DHTPIN 2 // what digital pin we're connected to
#define DHTTYPE DHT11 // DHT 11
DHT dht(DHTPIN, DHTTYPE);
int moisture = A0;
BlynkTimer timer;

{
  float h = dht.readHumidity();
  float t = dht.readTemperature(); // or dht.readTemperature(true) for Fahrenheit

  if (isnan(h) || isnan(t)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
  float moisturelevel = analogRead(moisture);
  int percentmoisture = (-0.1563*moisturelevel) + 147.98; //equation from calibration
  if(percentmoisture<0) (percentmoisture=0);
  // Serial.print("moisturelevel=");
  // Serial.println(moisturelevel);
  // Serial.print("moisturet=");
  // Serial.println(percentmoisture);
  // Serial.print("Humidity=");
  // Serial.println(h);
  // Serial.print("Temperature=");
  // Serial.println(t);
  // Serial.println();
  // You can send any value at any time.
  // Please don't send more that 10 values per second.
  Blynk.virtualWrite(V1, percentmoisture);
  Blynk.virtualWrite(V3, h);
  Blynk.virtualWrite(V2, t);
}

{
  pinMode(moisture, INPUT);

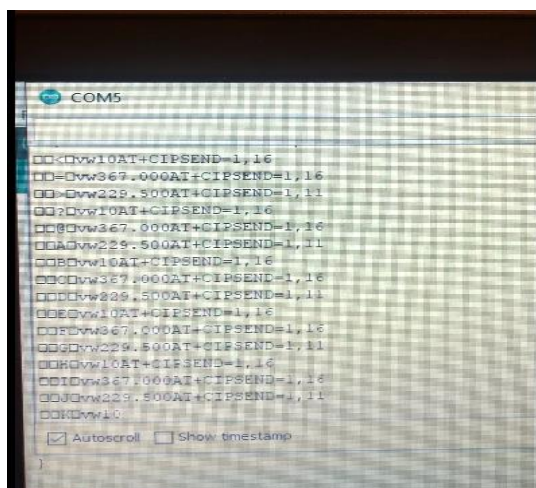
  // Serial.begin(9600);
  pc.begin(115200);
  EspSerial.begin(ESP8266_BAUD);
  Blynk.begin(auth, wifi, ssid, pass);

  dht.begin();

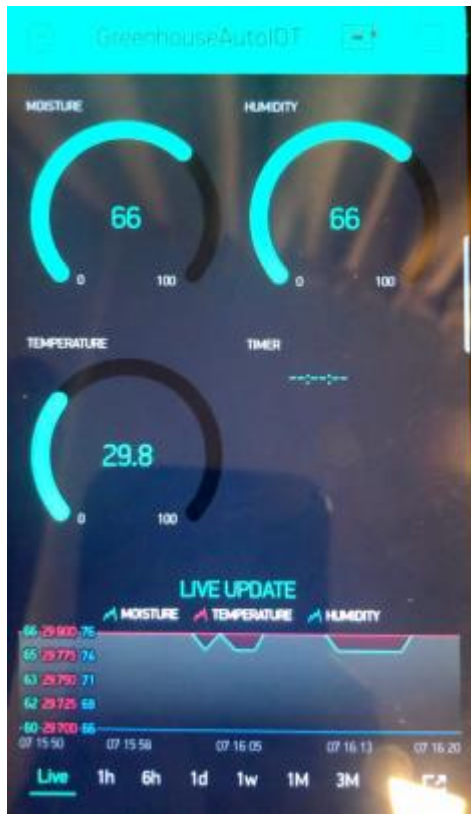
  delay(7000);
  //mySerial.println("AT+CNMI=2,2,0,0,0"); // AT Command to receive a live SMS
  //Serial.println(message);
  timer.setInterval(1000L, sendSensor);
}

void loop() {
  // float moisturelevel = analogRead(moisture);
  // int percentmoisture = (-0.1796*moisturelevel) + 159.15; //equation from calibration
  //
  // Serial.print("moisturelevel=");
  // Serial.println(moisturelevel);
  //
  Blynk.run();
  timer.run();
}
```

Sending data to Blynk



4.1 – Configure Cloud Services



Blynk Application showing the Real-Time/Live Update from DHT11 (Temperature & Humidity) and Soil Moisture in °C and % respectively.

5.1 – Generate Report based on Real-time data

Kindly refer to document titled LIVE PROJECT – Report