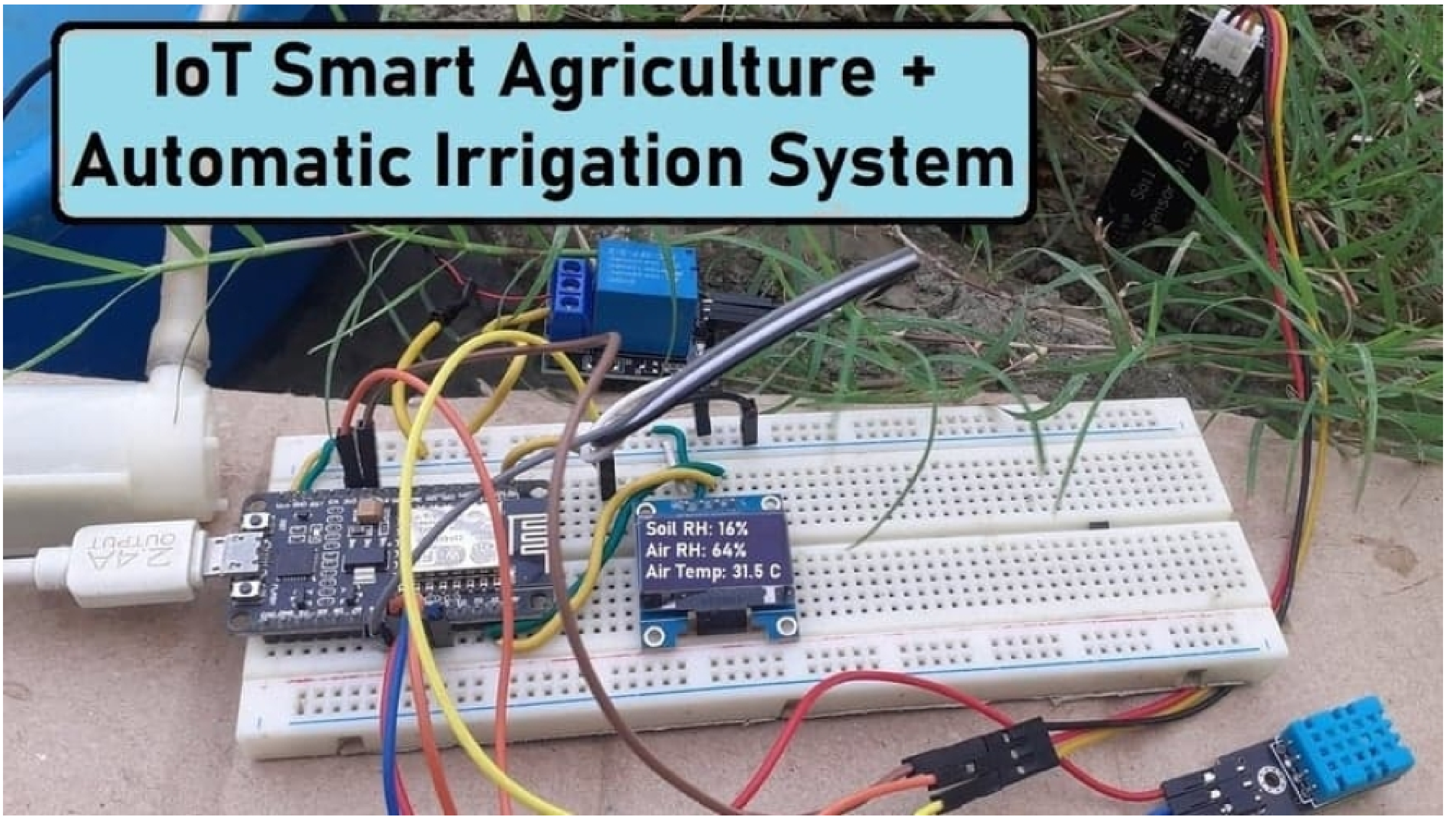


Overview: IoT Based Smart Agriculture & Automatic Irrigation System

In this project, we will learn about the IoT Based **Smart Agriculture** & **Automatic Irrigation System** with **Nodemcu ESP8266**. Agriculture plays a vital role in the development of agricultural countries. Some issues concerning agriculture have been always hindering the development of the country. Consequently, the only solution to this problem is **smart agriculture** by modernizing the current traditional methods of agriculture.

Hence the method is making agriculture smart using **automation** and **IoT technologies**. Internet of Things (IoT) enables various applications of **crop growth monitoring** and selection, **automatic irrigation** decision support, etc. We proposed **ESP8266 IoT Automatic irrigation system** to modernize and improve the productivity of the crop.

This post explains how to make **IoT Smart Agriculture with Automatic Irrigation System** using some simple sensors that are available in the market. We will use **Capacitive Soil Moisture Sensor** to measure moisture content present in the soil. Similarly to measure Air Temperature and Humidity, we prefer **DHT11 Humidity Temperature Sensor**. Using a **5V Power relay** we will control the **Water Pump**. Whenever the sensor detects a low quantity of moisture in the soil, the motor turns on automatically. Hence, will automatically irrigate the field. Once the soil becomes wet, the motor turns off. You can monitor all this happening remotely via **Thingspeak Server** online from any part of the world.

Bill of Materials

We will need the fllowing electronics **components** for making this project.You can purchase all the components online from **Amazon**. The component purchase link is given below.

S.N.	Components Name	Quantity
1	NodeMCU ESP8266 Board	1
2	Capacitive Soil Moisture Sensor	1
3	0.96" I2C OLED Display	1
4	DHT11 Sensor	1
5	1 Channel 5V Relay Module	1
6	5V DC Motor Pump	1
7	Connecting Wires	10
8	Breadboard	1

Capacitive Soil Moisture Sensor

This is an **analog capacitive soil moisture sensor** which measures soil moisture levels by **capacitive sensing**. This means the capacitance is varied on the basis of water content present in the soil. You can convert the capacitance into voltage level basically from 1.2V minimum to 3.0V maximum. The advantage of Capacitive Soil Moisture Sensor is that they are made of a **corrosion-resistant material** giving it a long service life. The sensor can be used to make **Automatic Plant Watering** System.

Features & Specifications

1. Supports 3-Pin Sensor interface
2. Analog output
3. Operating Voltage: DC 3.3-5.5V
4. Output Voltage: DC 0-3.0V
5. Interface: PH2.0-3P
6. Size: 99x16mm/3.9×0.63″

To learn more about Capacitive Soil Moisture Sensor you can follow this post: [Capacitive Soil Moisture Sensor Basic Interfacing Guide]

DHT11 Humidity Temperature Sensor

The **DHT11** is a basic, ultra low-cost digital temperature and humidity sensor. It uses a capacitive **humidity sensor** and a **thermistor** to measure the surrounding air. It spits out a digital signal on the data pin.

It's fairly simple to use, but requires careful timing to grab data. The only real downside of this sensor is you can only get new data from it once every 2 seconds. So when using the library, sensor readings can be up to 2 seconds old. In this project, we will use this sensor to measure the air temperature and humidity.

To learn more about DHT11 Humidity/Temperature Sensor you can follow this post: DHT11

[Basic Interfacing Guide]

DC 3-6V Micro Submersible Mini Water Pump

The **DC 3-6 V Mini Micro Submersible Water Pump** is a low cost, small size Submersible Pump Motor. It operates from a **2.5 ~ 6V power supply**. It can take up to 120 liters per hour with a very low current consumption of **220mA**. Just connect the tube pipe to the motor outlet, submerge it in water, and power it.

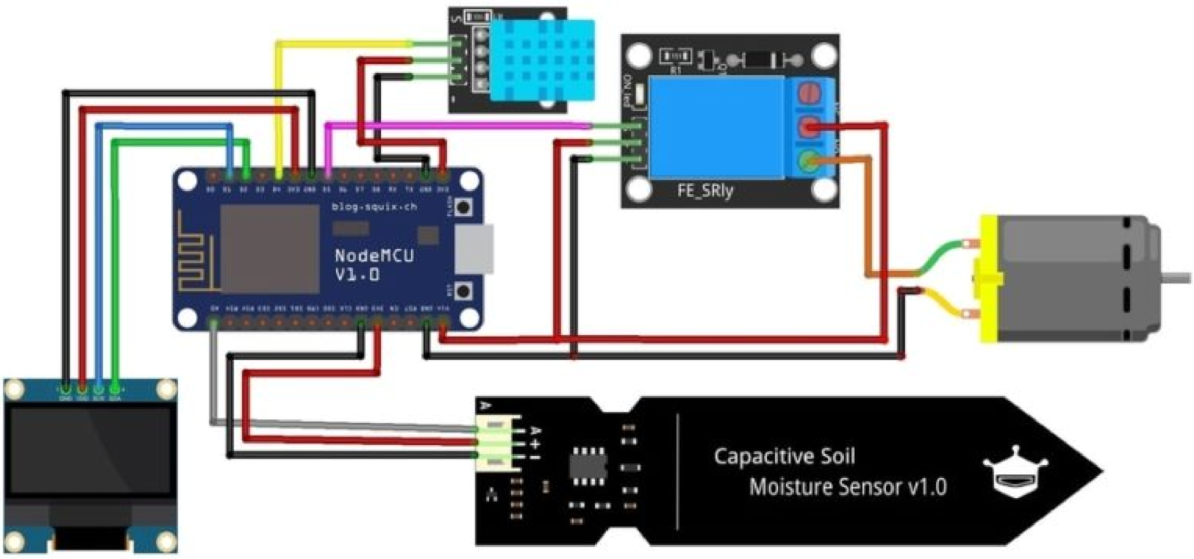
Features & Specifications

1. Operating Voltage : 2.5 ~ 6V
2. Operating Current : 130 ~ 220mA
3. Flow Rate : 80 ~ 120 L/H
4. Maximum Lift : 40 ~ 110 mm
5. Outlet Outside Diameter: 7.5 mm
6. Outlet Inside Diameter: 5 mm

Circuit Diagram & Connection

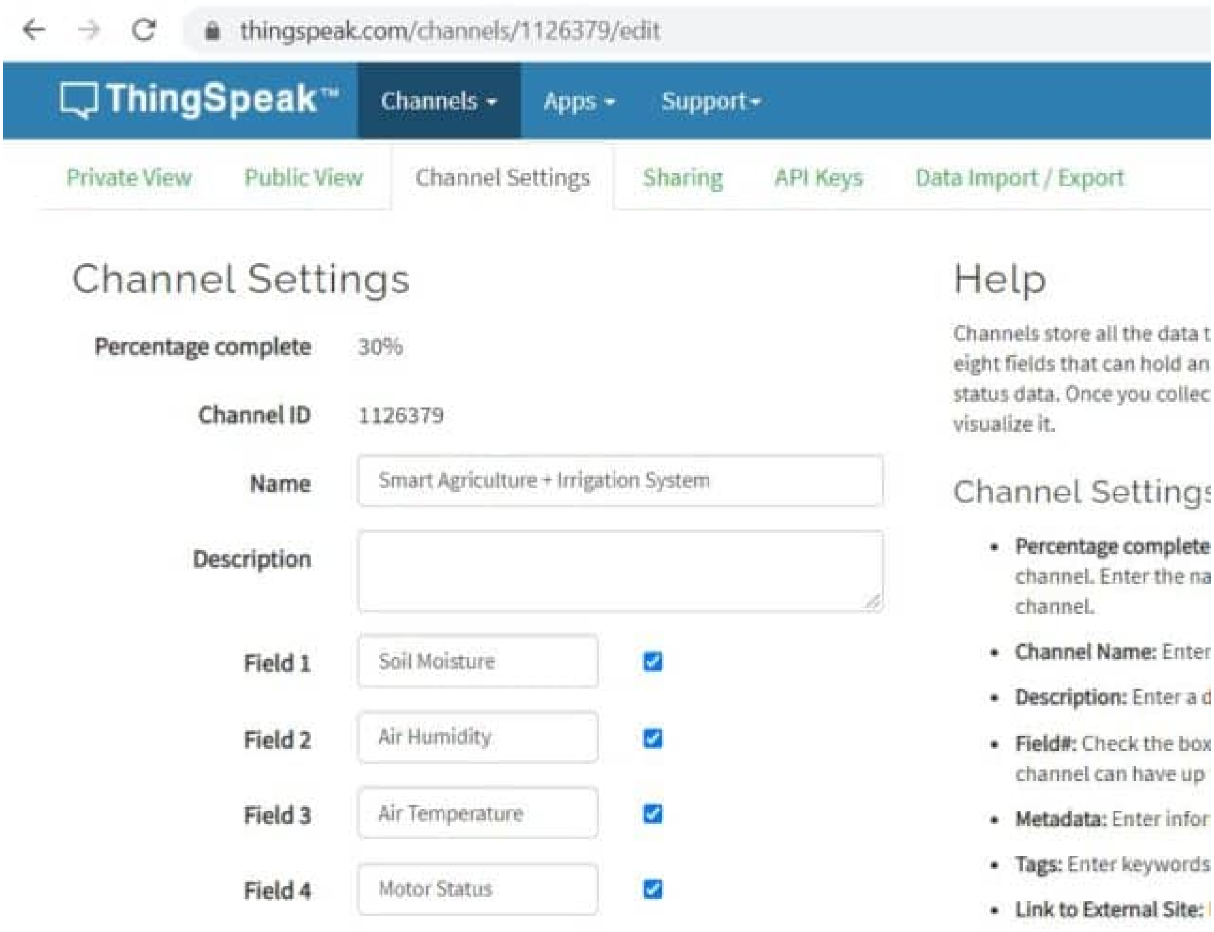
Let us see the schematic of the IoT Smart Agriculture & Automatic Irrigation System project. I use Fritzing to make an schematic for most of my projects. All you need is to place and connect a component that is super easy.

Connect the soil moisture sensor to A0 of Nodemcu and DHT11 to D4 Pin. The motor connects to Relay. To control the relay, we use the D5 Pin of NodeMCU. Connect the OLED display to the I2C pin of NodeMCU. You can power the Motor and Relay using the 5V pin of NodeMCU. The DHT11 Sensor, Capacitive Soil Moisture Sensor, and OLED Display require a 3.3V Supply only.



Setting Up Thingspeak Server

Now we need to **setup the Thingspeak Account**. To set up Thingspeak follow the following Steps:



Step 1: Visit https://thingspeak.com/ and create your account by filling up the details.

Step 2: Create a New Channel by Clicking on "Channel" & fill up the following details as shown in the image below.
Step 3: Click on API Key, you will see the "**Write API Key**". Copy the API Key. This is very important, it will be required in Code Part.
Step 4: You can click on the "**Private View**" & customize the display window as you want.

So, that's all from the Thingspeak Setup Part. Now let us move to the programming Part.

Source Code/Program

The Source Code for IoT Smart Agriculture & Automatic Irrigation System with ESP8266 is very simple. The code can be directly uploaded to the NOdeMCU Board. But before that make sure to add OLED Display Library from the below link.

1. Adafruit GFX Library: Download
2. SSD1306 Library: Download

In the below code part, change the Thingspeak API Key, WiFi SSID & Password.

The most important part of this code is the calibration of the Soil Moisture Sensor Value like AirValue and WaterValue. To learn how to calibrate and get the correct reading follow this post: **Calibrating Soil Moisture Sensor Value**. Please!, do not skip this step. This may result in getting the wrong soil moisture value and sometimes may result in continuous turning ON of the motor.

```
#include <ESP8266WiFi.h>

#include <SPI.h>

#include <Wire.h>

#include <Adafruit_GFX.h>

#include <Adafruit_SSD1306.h>

#include <DHT.h>  // Including library for dht


#define SCREEN_WIDTH 128 // OLED display width, in pixels

#define SCREEN_HEIGHT 64 // OLED display height, in pixels

#define OLED_RESET -1 // Reset pin # (or -1 if sharing Arduino reset pin)


#define DHTPIN D4        //pin where the dht11 is connected

DHT dht(DHTPIN, DHT11);


String apiKey = "C25ICK6FHOR7PST4"; // Enter your Write API key from ThingSpeak

const char *ssid = "MySmartHome";     // replace with your wifi ssid and wpa2 key

const char *pass = "nRF52840";

const char* server = "api.thingspeak.com";


Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);


const int AirValue = 790;   //you need to replace this value with Value_1

const int WaterValue = 390;  //you need to replace this value with Value_2

const int SensorPin = A0;

int soilMoistureValue = 0;

int soilmoisturepercent=0;
```

```
int relaypin = D5;


WiFiClient client;



void setup() {

 Serial.begin(115200); // open serial port, set the baud rate to 9600 bps

 display.begin(SSD1306_SWITCHCAPVCC, 0x3C); //initialize with the I2C addr 0x3C (128x64)

 display.clearDisplay();

 pinMode(relaypin, OUTPUT);


 dht.begin();


 WiFi.begin(ssid, pass);


 while (WiFi.status() != WL_CONNECTED)

 {

  delay(500);

  Serial.print(".");

 }

  Serial.println("");

  Serial.println("WiFi connected");

  delay(4000);

}



void loop()
```

```
{
 float h = dht.readHumidity();
 float t = dht.readTemperature();

 Serial.print("Humidity: ");
 Serial.println(h);
 Serial.print("Temperature: ");
 Serial.println(t);

 soilMoistureValue = analogRead(SensorPin);  //put Sensor insert into soil
 Serial.println(soilMoistureValue);

 soilmoisturepercent = map(soilMoistureValue, AirValue, WaterValue, 0, 100);


if(soilmoisturepercent > 100)
{
 Serial.println("100 %");

 display.setCursor(0,0);  //oled display
 display.setTextSize(2);
 display.setTextColor(WHITE);
 display.print("Soil RH:");
 display.setTextSize(1);
 display.print("100");
 display.println(" %");
 display.setCursor(0,20);  //oled display
```

```
  display.setTextSize(2);
  display.print("Air RH:");
  display.setTextSize(1);
  display.print(h);
  display.println(" %");
  display.setCursor(0,40);  //oled display
  display.setTextSize(2);
  display.print("Temp:");
  display.setTextSize(1);
  display.print(t);
  display.println(" C");
  display.display();

  delay(250);
  display.clearDisplay();
}


else if(soilmoisturepercent <0)
{
  Serial.println("0 %");

  display.setCursor(0,0);  //oled display
  display.setTextSize(2);
  display.setTextColor(WHITE);
  display.print("Soil RH:");
  display.setTextSize(1);
```

```
  display.print("0");
  display.println(" %");
  display.setCursor(0,20);  //oled display
  display.setTextSize(2);
  display.print("Air RH:");
  display.setTextSize(1);
  display.print(h);
  display.println(" %");
  display.setCursor(0,40);  //oled display
  display.setTextSize(2);
  display.print("Temp:");
  display.setTextSize(1);
  display.print(t);
  display.println(" C");
  display.display();

  delay(250);
  display.clearDisplay();
}

else if(soilmoisturepercent >=0 && soilmoisturepercent <= 100)
{
  Serial.print(soilmoisturepercent);
  Serial.println("%");

  display.setCursor(0,0);  //oled display
```

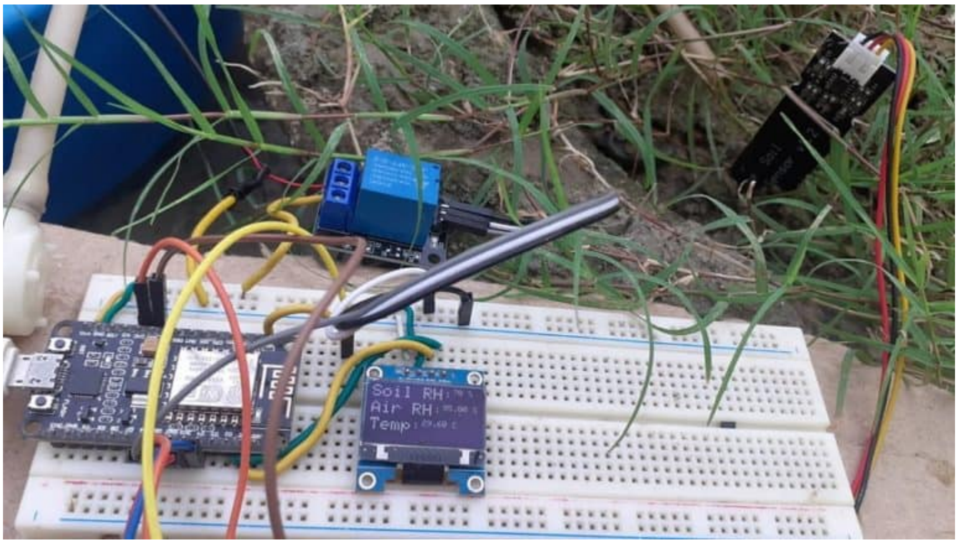
```
 display.setTextSize(2);
 display.setTextColor(WHITE);
 display.print("Soil RH:");
 display.setTextSize(1);
 display.print(soilmoisturepercent);
 display.println(" %");
 display.setCursor(0,20); //oled display
 display.setTextSize(2);
 display.print("Air RH:");
 display.setTextSize(1);
 display.print(h);
 display.println(" %");
 display.setCursor(0,40); //oled display
 display.setTextSize(2);
 display.print("Temp:");
 display.setTextSize(1);
 display.print(t);
 display.println(" C");
 display.display();

 delay(250);
 display.clearDisplay();
}

 if(soilmoisturepercent >=0 && soilmoisturepercent <= 30)
 {
  digitalWrite(relaypin, HIGH);
```

```
  Serial.println("Motor is ON");

}

else if (soilmoisturepercent >30 && soilmoisturepercent <= 100)

{

  digitalWrite(relaypin, LOW);

  Serial.println("Motor is OFF");

}


if (client.connect(server, 80)) // "184.106.153.149" or api.thingspeak.com

{

  String postStr = apiKey;

    postStr += "&field1=";

    postStr += String(soilmoisturepercent);

    postStr += "&field2=";

    postStr += String(h);

    postStr += "&field3=";

    postStr += String(t);

    postStr += "&field4=";

    postStr += String(relaypin);

    postStr += "\r\n\r\n\r\n\r\n";


  client.print("POST /update HTTP/1.1\n");

  client.print("Host: api.thingspeak.com\n");

  client.print("Connection: close\n");

  client.print("X-THINGSPEAKAPIKEY: " + apiKey + "\n");

  client.print("Content-Type: application/x-www-form-urlencoded\n");

  client.print("Content-Length: ");
```

```
  client.print(postStr.length());

  client.print("\n\n");

  client.print(postStr);



}
```

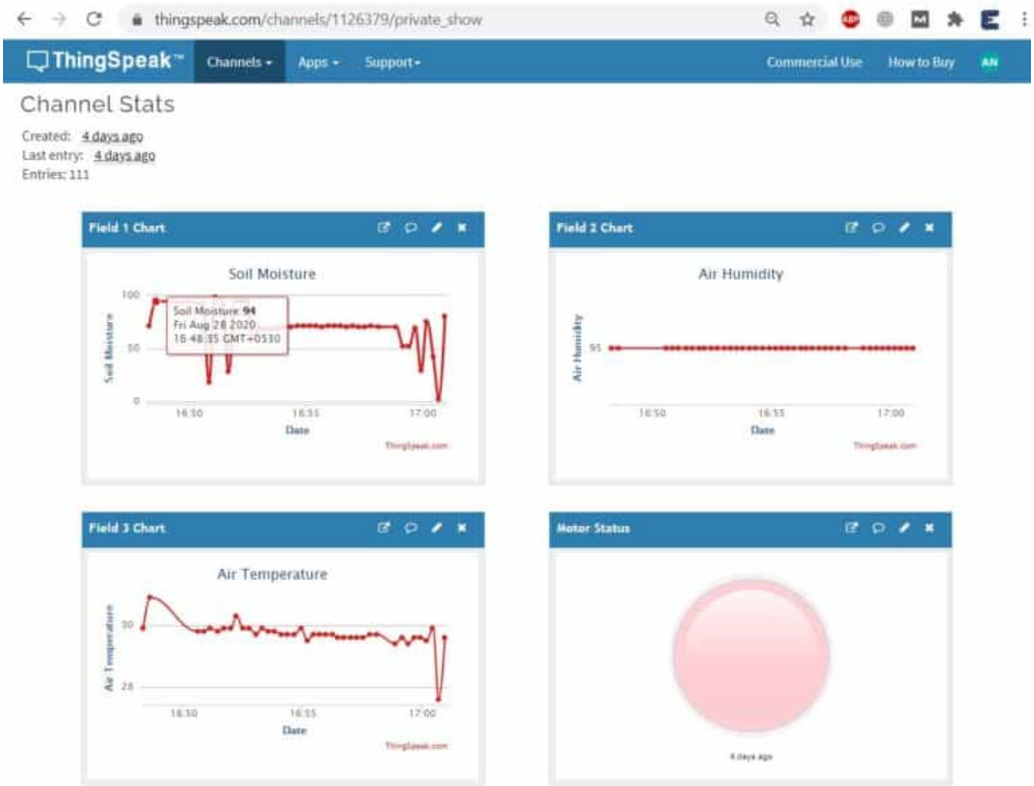
Testing & Results

This water pump need to be fully submerged in water. The outlet pipe is kept in a field for irrigation. Similarly soil Moisture sensor is dipped in soil.



As soon as you power on the device, the OLED will start displaying the Soil Humidity, Air Humidity, and also Air Temperature. It shows the real-Time Data. When the soil moisture content is reduced the water pumps turn on and irrigate the field until the required moisture is achieved. You can check its full working here in this video below.

You can monitor the data online from any part of the world using Thingspeak Server. To do that, go to the private view of the Thingspeak server. You can check the soil Moisture, Humidity, and Temperature as well as relay status.


thingspeak.com/channels/1126379/private_show
ThingSpeak™
Channels
Apps
Support
Commercial Use
How to Buy
Channel Stats
Created: 4 days ago
Last entry: 4 days ago
Entries: 111
Field 1 Chart
Soil Moisture
Soil Moisture: 94
Fri Aug 28 2020
16:48:35 GMT+0530
Date
Field 2 Chart
Air Humidity
Date
Field 3 Chart
Air Temperature
Date
Motor Status
4 days ago