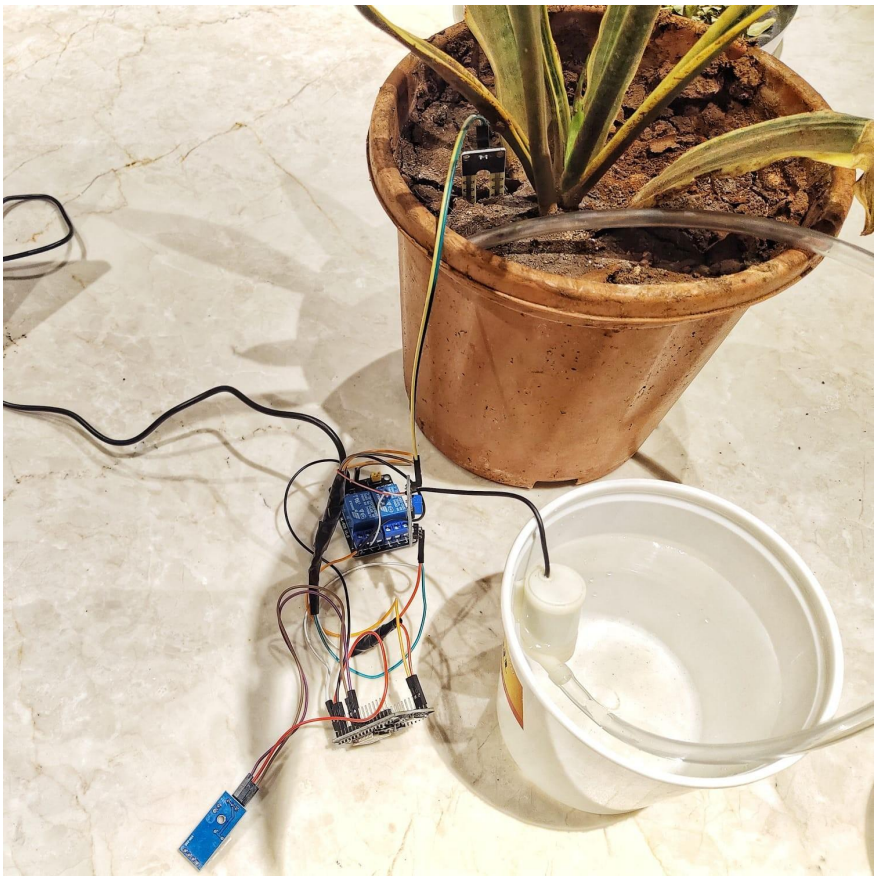
Smart Irrigation And Plant Monitoring IOT Project



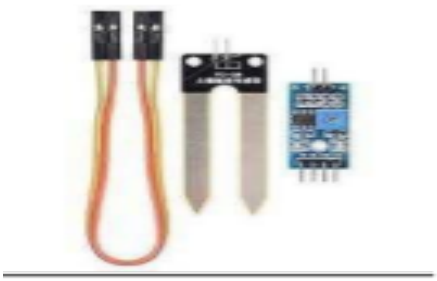
Description:

Ours is a smart irrigation system and plant monitoring project in which we have used soil moisture sensor to detect if the moisture of the soil is appropriate enough for the plant to live if the soil is too dry the motor automatically starts pumping water and notifies us of the same on our phones. When the soil becomes moist enough for the plant to survive healthily the pumps automatically stops the flow of water notifying the same on our phone. For the plant monitoring plant we have used DHT temperature and humidity sensor which detects the temperature and humidity of the surrounding, the data is then stored on Blynk, we can review the data there and deduce if the conditions are sufficient for healthy life of the plant.

Devices Used:

- Submersible mini water pump
- Soil moisture sensor
- NodeMCU
- DHT11 temperature and humidity sensor
- Jumper wires
- 5V,10A 2-Channel Relay Module

A) Soil Moisture Sensor:



Soil moisture sensors measure or estimate the amount of water in the soil. These sensors can be stationary or portables such as handheld probes. Stationary sensors are placed at the predetermined locations and depths in the field, whereas portable soil moisture probes can measure soil moisture at several locations

B) NodeMCU:

NodeMCU is an open-source platform based on ESP8266 which can connect objects and let data transfer using the Wi-Fi protocol. In addition, by providing some of the most important features of microcontrollers such as GPIO, PWM, ADC, etc., it can solve many of the project's needs alone.

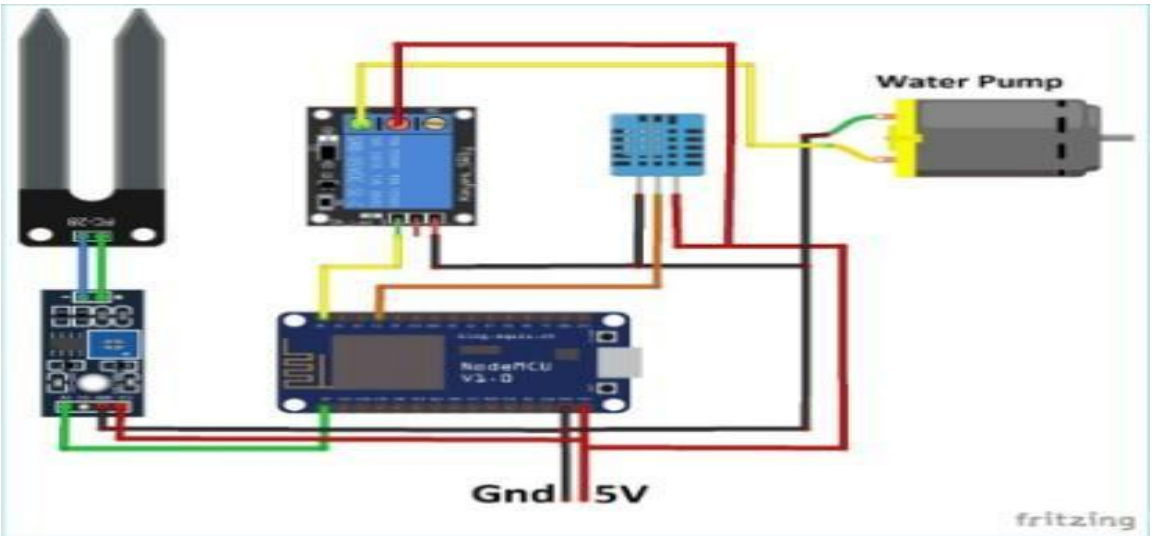
C) DHT11 temperature and humidity sensor:

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, and spits out a digital signal on the data pin (no analog input pins needed). Its fairly simple to use, but requires careful timing to grab data.

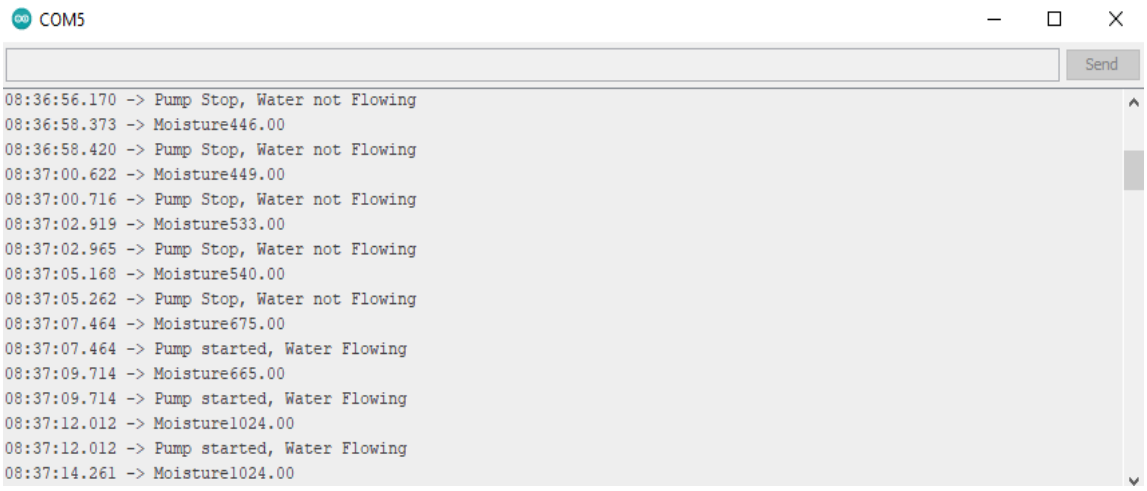
D) 5V,10A 2-Channel Relay Module:

2-Channel 5V Relay Module is a relay interface board, it can be controlled directly by a wide range of microcontrollers such as Arduino, AVR, PIC, ARM and so on. It uses a low level triggered control signal (3.3-5VDC) to control the relay. Triggering the relay operates the normally open or normally closed contacts.

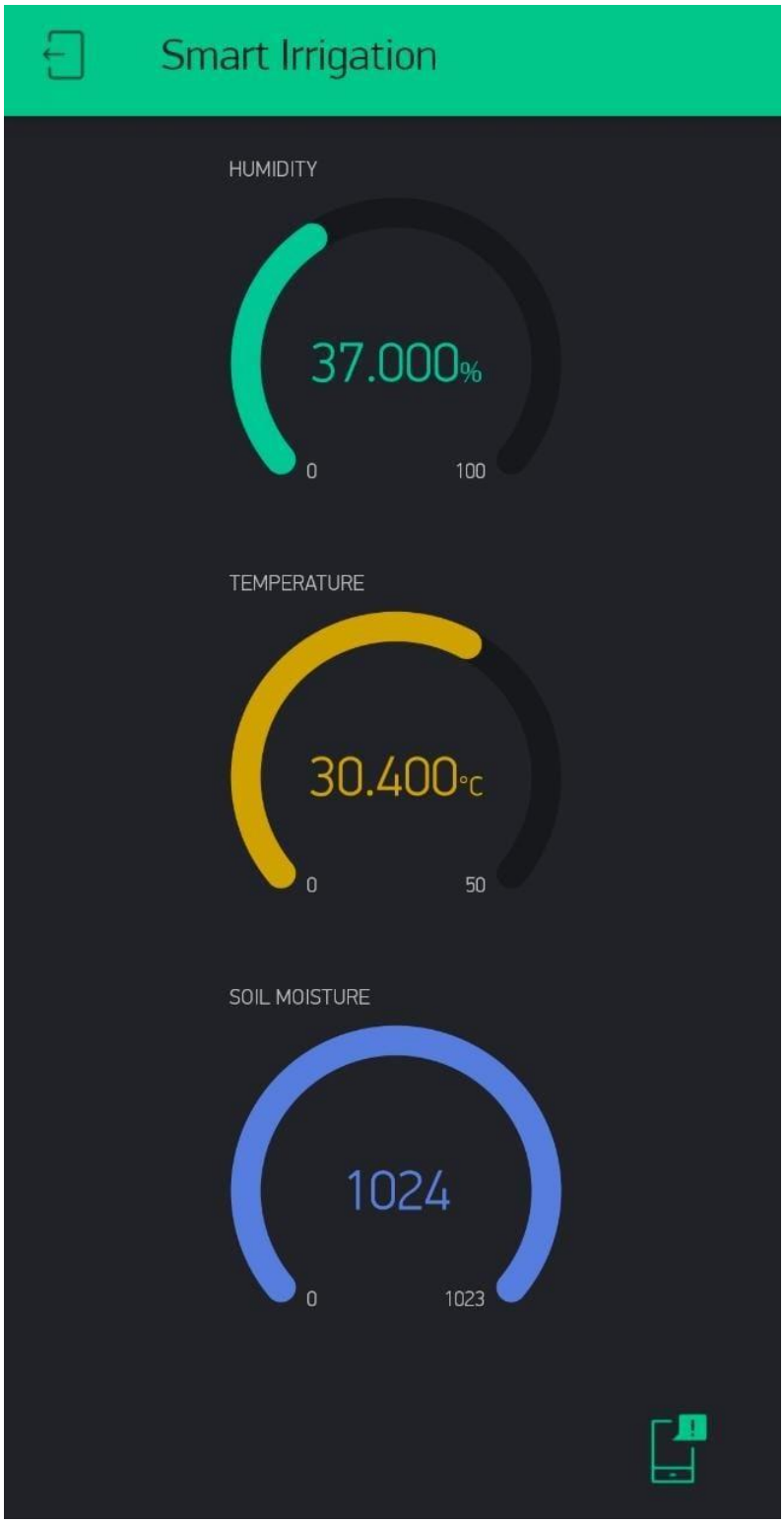
CIRCUIT DIAGRAM



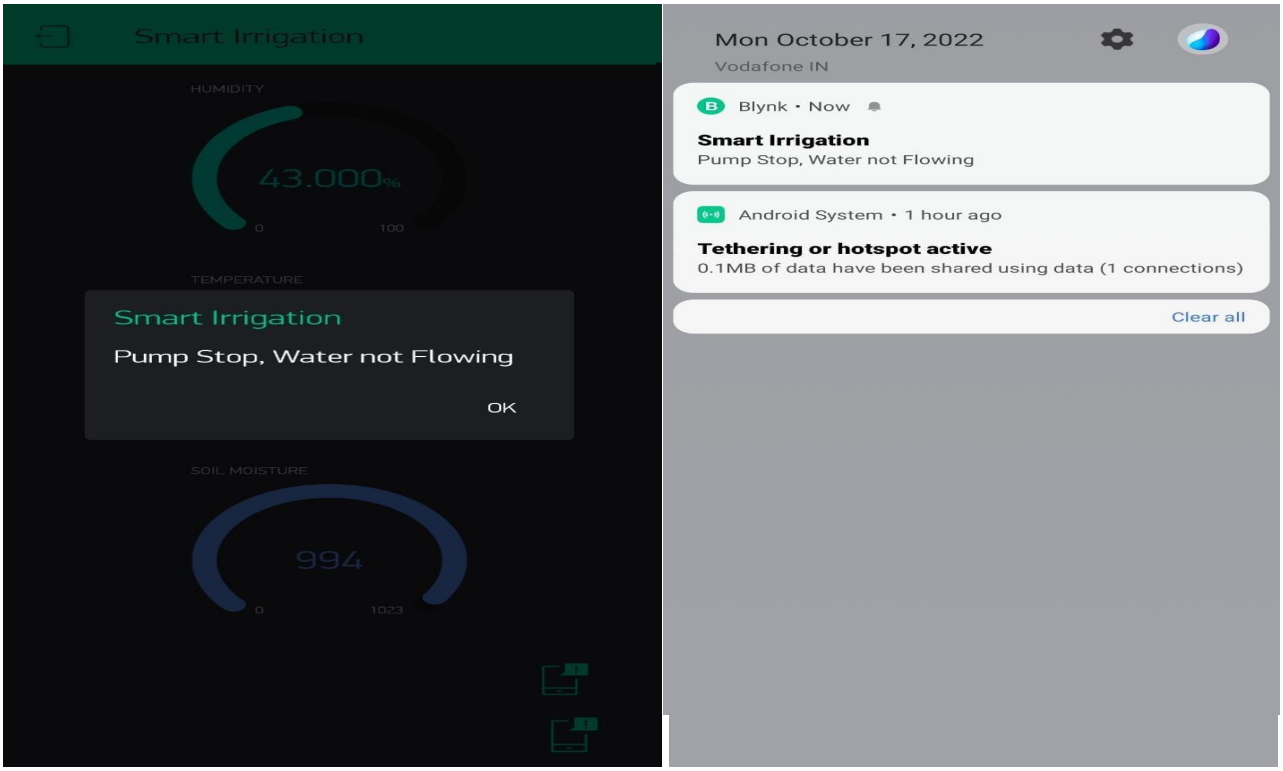
SERIAL MONITOR



BLYNK APP ON MOBILE PHONE



NOTIFICATION FEATURE



Code:

```
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <SPI.h>
#include <DHT.h>

// You should get Auth Token in the Blynk App.
char auth[] = "Y1wPunpUXU8NPEiUFjzLb_PMkFIHrWO8";

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "vivo 1818";
char pass[] = "purva1818";


//DHT11 for reading temperature and humidity value

#define Light D2
#define DHTPIN D1
#define DHTTYPE DHT11
```

```
DHT dht(DHTPIN, DHTTYPE);
BlynkTimer timer;
float moisture;

void sendSensor()
{
  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;
  }

  Blynk.virtualWrite(V2, t);
  Blynk.virtualWrite(V1, h);

}

void setup()
{
  // Debug console
  Serial.begin(9600);
```

```
  pinMode(Light,OUTPUT);


  Blynk.begin(auth, ssid, pass);
  digitalWrite(Light,HIGH);


  dht.begin();

  // Setup a function to be called every second
  timer.setInterval(1000L, sendSensor);
}

void loop()
{
   Blynk.virtualWrite(V3,analogRead(A0));
   moisture=analogRead(A0);
   Serial.print("Moisture");
   Serial.println(moisture);

  if(analogRead(A0) <= 600)
   {
```

```
   digitalWrite(Light,HIGH);
   Blynk.notify("Pump Stop, Water not Flowing");
   Serial.println("Pump Stop, Water not Flowing");
   delay(2000);
  }
 else if(analogRead(A0)>= 650)
 {
   Serial.println("Pump started, Water Flowing");
   Blynk.notify("Pump Started, Water Flowing");
   digitalWrite(Light,LOW);
   delay(2000);
 }
  Blynk.run();
  timer.run();
}
```