Abstract

Nowadays world is fully overtaken by the internet and internet of things. The Internet of Things (IoT) is the network for physical objects. Agriculture is the backbone of our country, most of the people depend on agriculture. The main issue in agriculture is water scarcity. The water resource is not used in an effective manner, so the water is wasted. The use of internet of things in this field will be helpful to reduce the wastage of water. We proposed a system called Smart Plant Monitoring System that will capture all the details about the soil,humidity and the temperature by means different sensors. IoT permits objects to be sensed or controlled remotely across the network infrastructure. The result improves accuracy, economic benefits, efficiency and reduces intervention of human.

Introduction

Smart Plant Monitoring System is a IOT based plant operating system which is designed and built to watering and monitoring plant via cloud server..This project highlights various features such as smart decision making based on soil moisture real time data.The system works through soil moisture sensor, humidity and temperature sensor whose are set in the root zone of the plants. The data will gather from the sensors and send to the web server.In agriculture,the major problem is the water scarcity because of low rainfall and wastage of water. Many techniques are proposed in IoT in terms of providing a better irrigation to the crop. This IoT device can also be used in home for monitoring the garden real time.

Requirements:

1) NodeMCU ESP8266 x 1
2) Temperature and Humidity Sensor x 1
3) Soil Moisture Sensor x 1
4) Jumper Wires x 1
5) Relay 5v x 1
6) 5v DC Submersible pump with pipe x 1

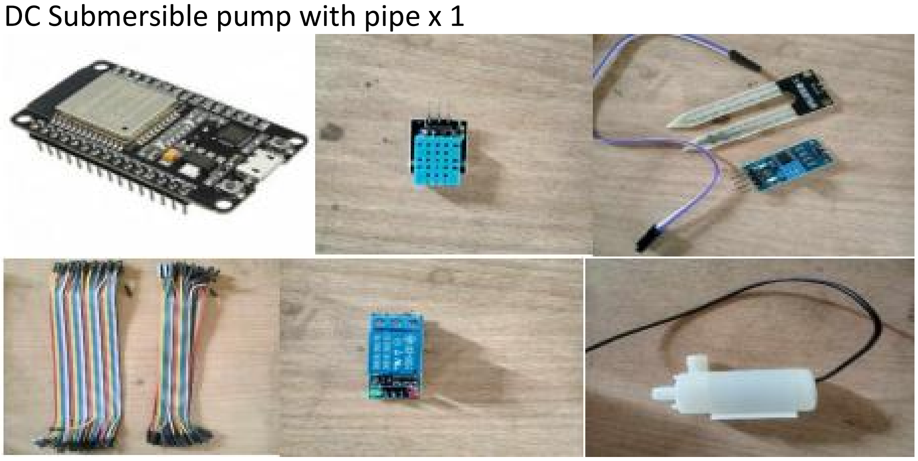

Fig 1: Equipments

Design:

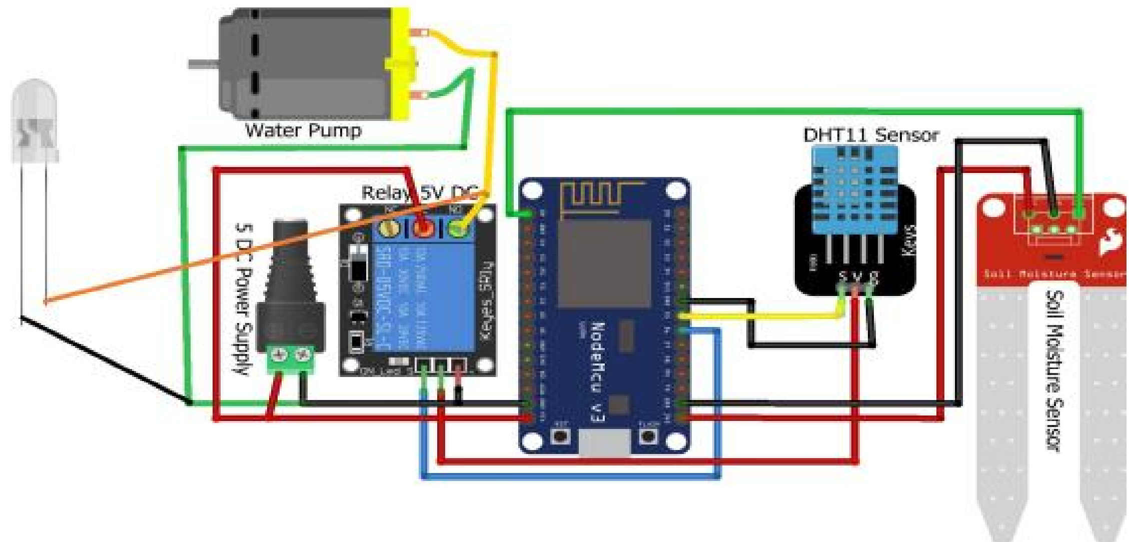

Fig 2.1: Circuit Diagram

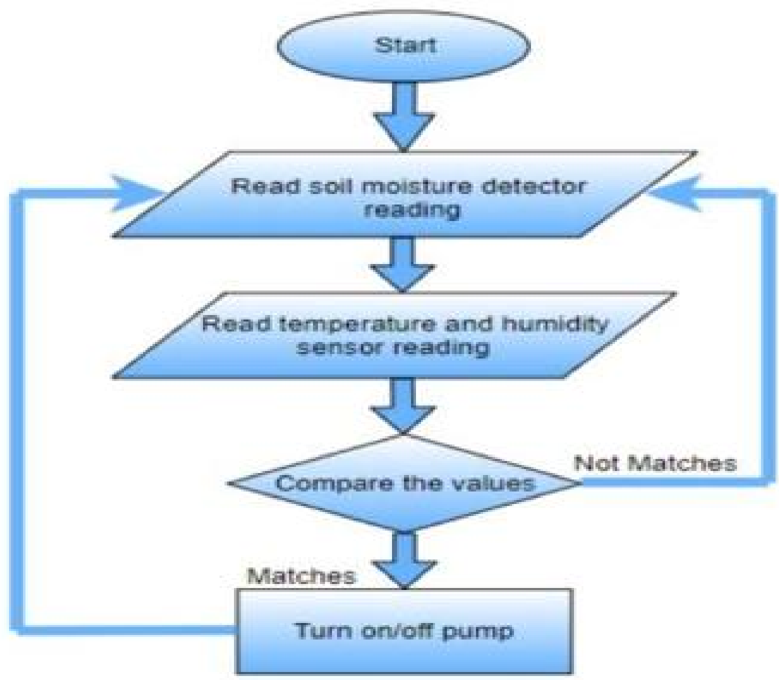

Fig 2.2: Data Flow Diagram

Working Procedure:

Soil moisture sensor will collect the moisture present in soil and DHT11 sensor will collect the temperature along with humidity near to the plants. Then it will send the data to ESP8266 based NodeMCU, which will send the data further to Blynk 2.0 platform.

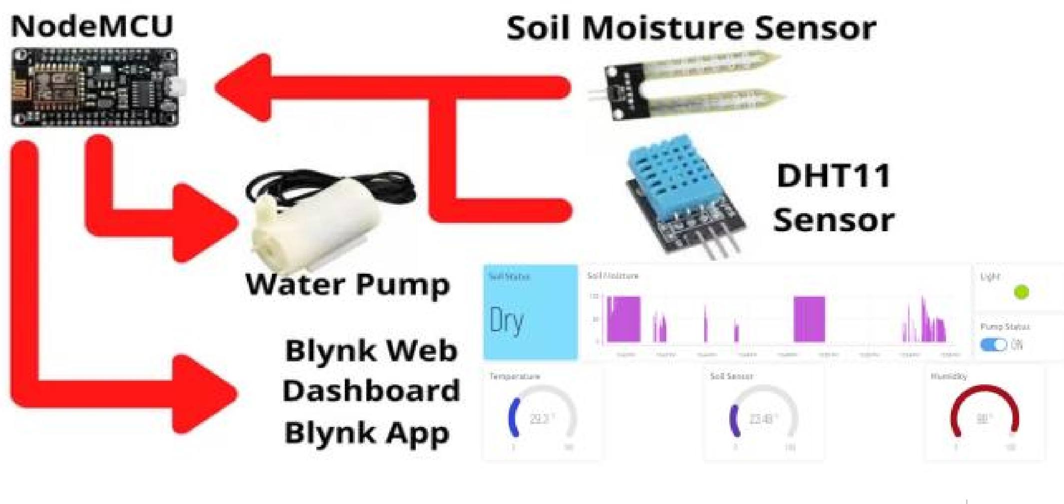

Fig 3.1: Overall working process

Hardware Setup:

1) Connect the soil moisture sensor to the analog pin A0 of the NodeMCU.

2) Connect the DHT11 temperature and humidity sensor to digital pin D2

3) Connect the relay module to the digital pin D8

4) Connect the motor and led light to the 1channel 5v relay module and 9v battery.

Blynk App Setup:

Configuring a project in Blynk involves several steps. After installing the Blynk app or go to Blynk website, we need to create an account and start a new project.

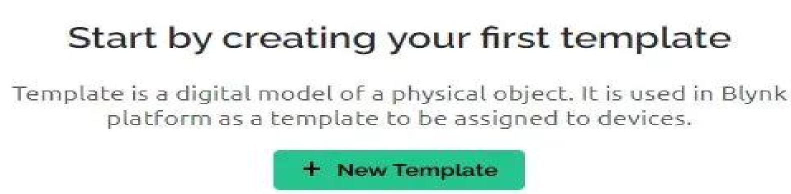

Fig 3.2: Creating Template

Then,create datastreams, through which we are going to send the data from hardware to blynk servers.

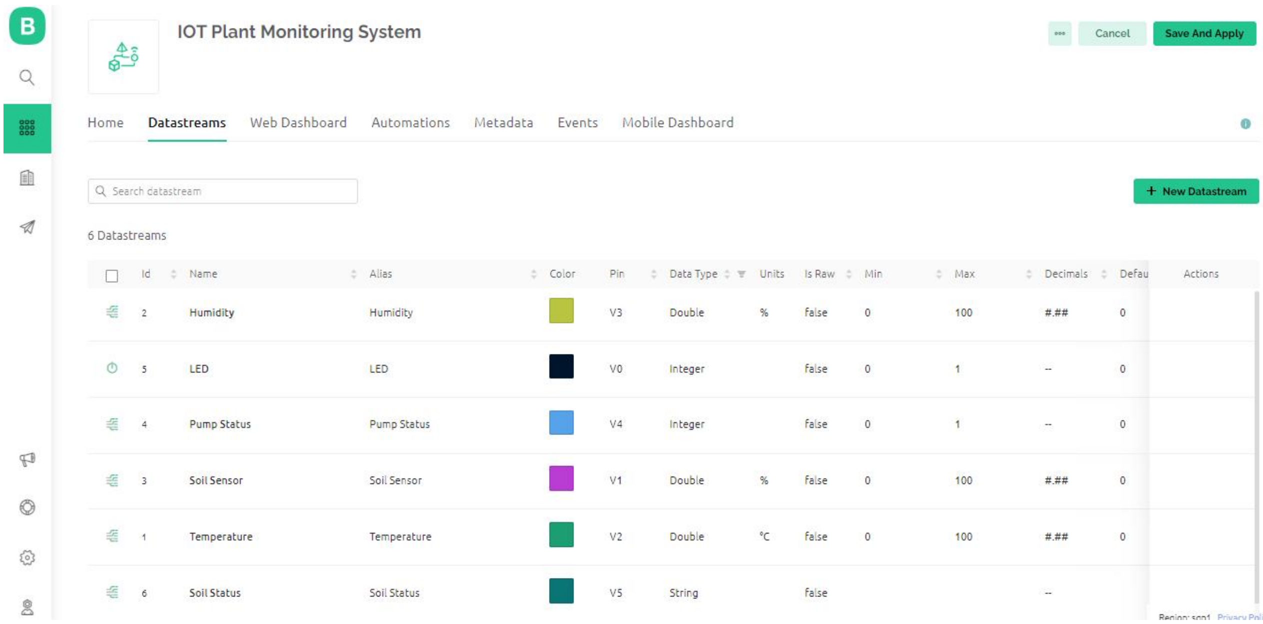

Fig 3.3: Setup Virtual Pins

Then,add widgets like buttons or graphs to the dashboard, assigning them to the relevant virtual pins (V1, V2, V3, etc).

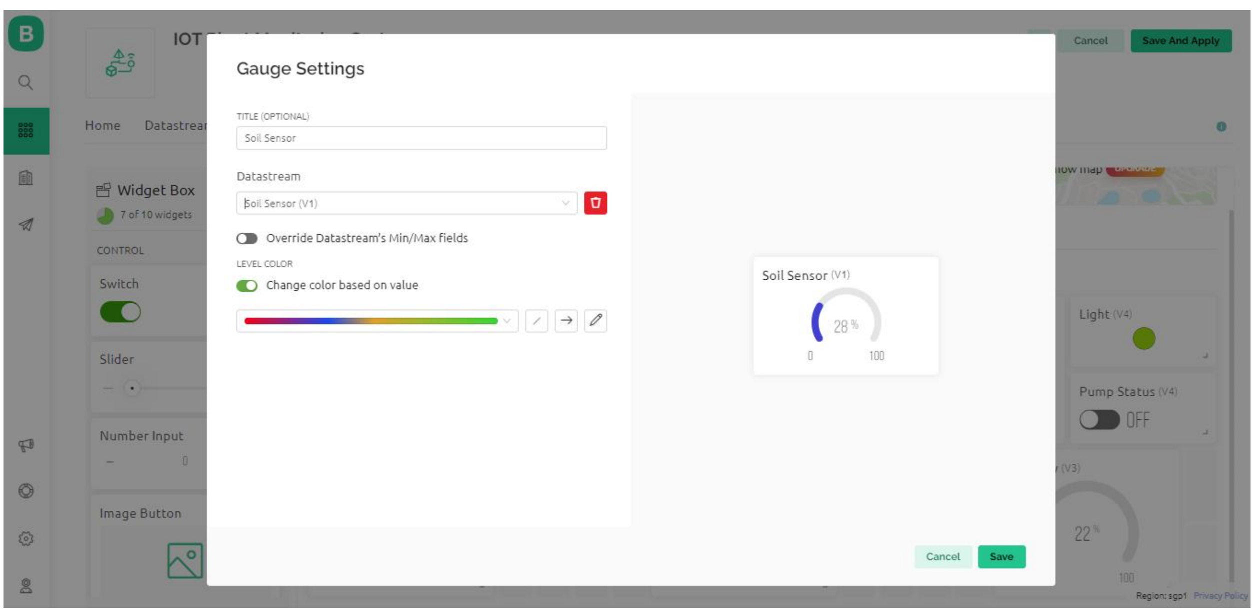

Fig 3.4: Add Widgets

Similarly create for humidity,temparature,pump status,soil status,light and average of soil moistures data chart as well. Write code using the Blynk library and token in Arduino IDE, then upload it to NodeMCU.

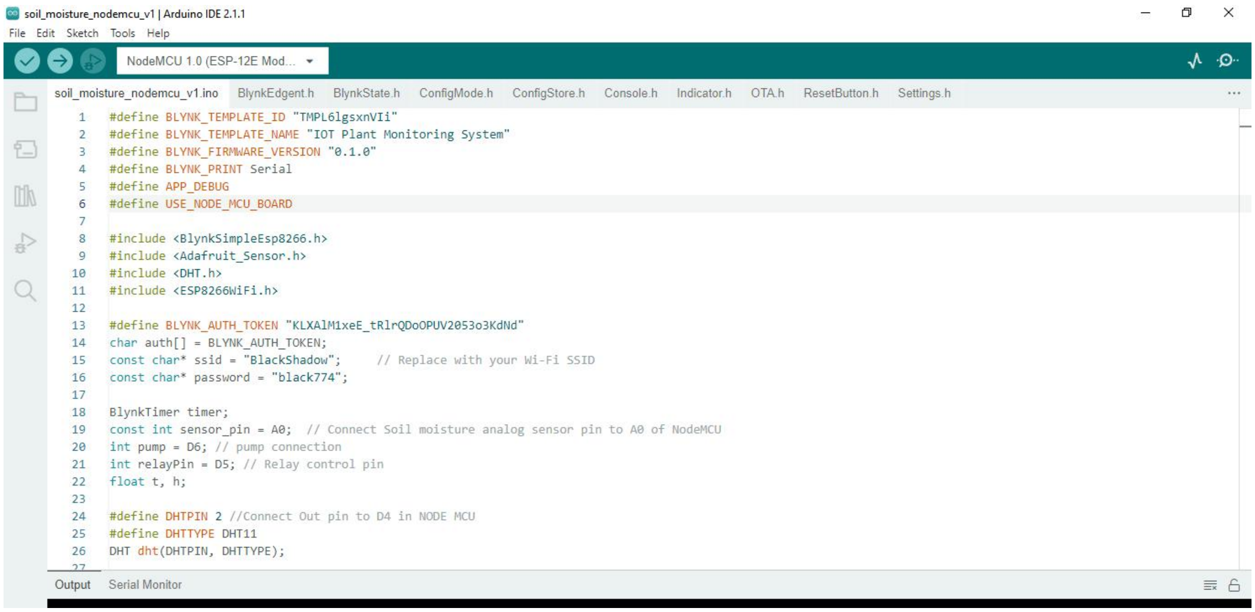

Fig3.5: Arduino IDE

- If soil_moisture < dry_threshold, activate water pump.
- If dry_threshold ≤ soil_moisture < optimal_threshold, keep pump off.
- If soil_moisture ≥ optimal_threshold, delay watering and keep pump off.

Run the project in the app to establish a connection. Use the widgets to monitor and control hardware, and further customize the setup by adding automation and notifications. Once everything works as intended, save the project and share it if desired.

Code:

```
#define BLYNK_TEMPLATE_ID "TMPL6lgsxnVIix"

#define BLYNK_TEMPLATE_NAME "IOT Plant Monitoring System"

#define BLYNK_FIRMWARE_VERSION "0.1.0"

#include <BlynkSimpleEsp8266.h>

#include <Adafruit_Sensor.h>

#include <DHT.h>

#include <ESP8266WiFi.h>

#define BLYNK_AUTH_TOKEN "KLXAlM1xeE_tRlrQDoOPUV2053o3KdNdx"
```

```
char auth[] = BLYNK_AUTH_TOKEN;
const char* ssid = "wifi";
const char* password = "password";
BlynkTimer timer;
const int sensor_pin = A0;
int pump = D6;
int relayPin = D5;
float t, h;
#define DHTPIN 2 //Connect Out pin to D4 in NODE MCU
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
const  int  dry_threshold = 30;
const  int  saturated_threshold = 70;

void sendSensor()
{
  h = dht.readHumidity();
  t = dht.readTemperature();
  Blynk.virtualWrite(V2, t);
  Blynk.virtualWrite(V3, h);
}

void setup() {
  Serial.begin(9600);
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
```

```
  delay(1000);
 }
 dht.begin();
 pinMode(D0, OUTPUT);
 pinMode(relayPin, OUTPUT);
 Blynk.begin(auth, ssid, password, "blynk.cloud", 80);
 delay(5000);
 pinMode(pump, OUTPUT);
 timer.setInterval(5000L, sendSensor);
}

void loop() {
 Blynk.run();
 timer.run();
 float soil_moisture;
 soil_moisture = (100 - (analogRead(sensor_pin)/1023.00) * 100.00 )*3.08;
 Blynk.virtualWrite(V1, soil_moisture);
 if (soil_moisture < dry_threshold) {
  digitalWrite(pump, HIGH);
  digitalWrite(relayPin, HIGH);
  Blynk.virtualWrite(V4, HIGH);
  Blynk.virtualWrite(V5, "Dry");
 } else if (soil_moisture > saturated_threshold) {
  digitalWrite(pump, LOW);
  digitalWrite(relayPin, LOW);
  Blynk.virtualWrite(V4, LOW);
  Blynk.virtualWrite(V5, "Saturated");
 } else {
```

```
  digitalWrite(pump, LOW);
  digitalWrite(relayPin, LOW);
  Blynk.virtualWrite(V4, LOW);
  Blynk.virtualWrite(V5, "Optimal");
 }
}
```

Result with Images:

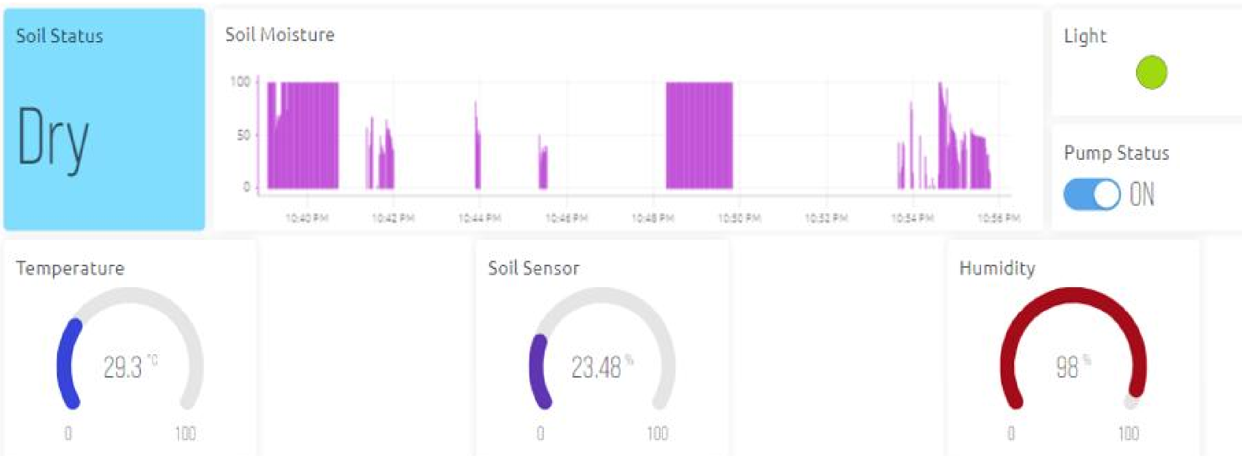

Fig 4.1: Web UI

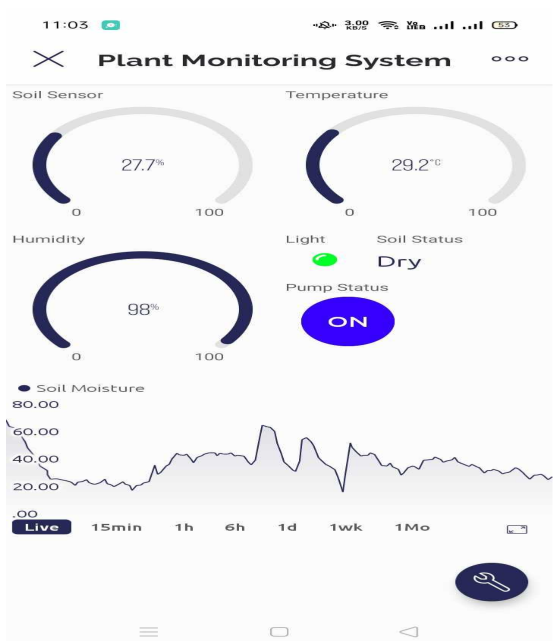

Fig 4.2: Mobile App UI

Fig 4.3: Plant water testing

Application:

1) Irrigation system.
2) Plant watering system in household and house garden.
3) Soil moisture detection and to avoid roughness of soil.
4) Temperature detection and prediction of plants health according to the data.

Future Scope:

1) A speaking voice alarm could be used.
2) Integration with weather forecasts for smarter irrigation.
3) Implementation of AI algorithms for predictive plant care.
4) Expansion to support more sensors for comprehensive monitoring.

Limitation:

1) Stable power supply is essential.
2) Reliable internet connection is needed.
3) Setup requires technical knowledge.
4) Costs can increase with advanced features.

Cost Estimation: The costs for the components needed for the project:

- NodeMCU (ESP8266) Development Board: Provided from MEC
- Soil Moisture Sensor: 120 Taka
- DHT11 Temperature and Humidity Sensor: 150 Taka

- ✧ Relay Module: 99 Taka
- ✧ Water Pump: 145 Taka
- ✧ 9v Battery with connectors: 80 Taka
- ✧ Miscellaneous (led,glue stick,shipping cost, etc.): 76 Taka

Total Estimated Cost: 670 Taka

Conclusion:

The implementation of IoT based smart plant monitoring system using has been verified to satisfactorily work by connecting different parameters of the soil to the cloud and was successfully controlled remotely through a mobile application. The system is designed for monitoring the sensors data, like moisture, humidity and temperature. The system can be elevated to the next level by developing sensor technology. Installation of soil nutrient sensors can modify the system to supply fertilizers to the garden precisely. Providing feedback of the system will improve the implementation of the gardening process. A system to monitor the parameters of soil system was designed and the project provides an opportunity to study the existing systems. Agriculture is one of the most water-consuming activities. The proposed can be used to switch the motor depending on favorable conditions of plants i.e. sensor values, thereby automating the process of irrigation. Which is one of the most time efficient activities in farming, which helps to prevent over irrigation or under irrigation of soil thereby avoiding crop damage. This project concludes that there can be efficient development in farming with the use of IoT and automation.

References:

1) https://iotstarters.com/iot-smart-plant-watering-system-project/?expand_article=1
2) https://www.ijraset.com/research-paper/iot-based-smart-plant-monitoring-system