Smart Irrigation System

IOE Mini Project Report submitted in partial fulfilment
of the requirement for the degree of
B.E. (Information Technology)

Submitted By

Roshan Mhatre (19101B0017)
Yash Kadulkar (19101B0018)
Sanket Sawant (19101B0020)

Under the Guidance of
Prof. Ajitkumar Khachane
Department of Information Technology

Vidyalankar Institute of Technology
Wadala(E), Mumbai 400 037

University of Mumbai

2022-2023

CERTIFICATE OF APPROVAL
For
IOE Mini Project Report

This is to Certify that

Roshan Mhatre
Yash Kadulkar
Sanket Sawant

Have successfully carried out Mini Project entitled
“Smart Irrigation System”
in partial fulfilment of degree course in
Information Technology
As laid down by University of Mumbai during the academic year
2022-2023

Under the Guidance of
Prof. Ajitkumar Khachane

Prof . Ajitkumar Khachane
Signature of Guide

Dr Vipul Dalal
Head of Department

Examiner 1

Examiner 2

Dr. S. A. Patekar
Principal

ACKNOWLEGEMENT

We express our profound gratitude and sincere thanks to **Prof. Ajitkumar Khachane** our respectable project guide, for his gigantic support and guidance. Without his counselling our project would not have seen the light of the day.

We extend our sincere thanks to **Dr. Vipul Dalal** Head of the Department of Information Technology for offering valuable advice at every stage of this undertaking. We would like to thank all the staff members who willingly helped us. We are grateful to VIDYALANKAR INSTITUTE OF TECHNOLOGY for giving us this opportunity.

1. **Roshan Mhatre**
2. **Yash Kadulkar**
3. **Sanket Sawant**

Table of Contents

List of Figures

Abstract

In this project, we introduce automatic plant watering system, which is considered as one of the most used and the most beneficial automated systems nowadays, which help people in their daily activities by reducing or completely replacing them effort. This system uses sensor technology along with microcontroller and other electronics to behave like smart switching system which senses soil moisture level and irrigates the plant if necessary. Purpose of this work is automating your plant watering system with help of technology.

The system made in that way would be the most appropriate for home usage as solution for some daily and usual issues, there is a wide spectrum of possibilities of implementing these systems as a long-term solution for many agricultural and medical problems, some of which are undernourishment and air pollution as most prominent, dangerous, and important ones.

1. Introduction

In daily operation related to watering the plants is the most important cultural practice and the most labor-intensive task. No matter whichever weather it is, either too hot or cold or too dry and wet, it is very crucial to control the amount of water that reaches the plants. So, it will be effective to use the idea of automatic plant watering system which waters plants when they need it. An important aspect of this project is that: “when and how much water”. To reduce manual activities for humans to water plants, and idea of a plant watering system is adopted. The method employed is to monitor the soil moisture level continuously and to decide whether watering is needed or not, and how much water is needed in the plant’s soil. This project can be grouped into subsystems such as power supply, relays, solenoid valve, NodeMcu, Soil moisture sensor and temperature and humidity sensor.

Essentially, the system is designed and programmed in such a way that soil moisture sensor senses the moisture level of plants at instance of time if moisture level of sensor is less than the specified value of threshold which is predefined according to the plant’s water needs then the desired amount of water should be supplied till it reaches the predefined threshold value.

2. Aim and Objectives

The aim of the project is to develop a plant monitoring system which will monitor all the plant conditions like humidity, temperature and soil moisture and report the same to mobile application.

The core objectives are:

- Gather system requirements
- Evaluate and study the platform required for the system
- Evaluate and study suitable development language, technologies, and tools.
- Monitor plant requirements like soil moisture, humidity, temperature.
- Based on the above parameters control LED lights and water pump.
- Evaluate and test the system
- Maintain system.

3. Problem Statement

During day-to-day activities may people forget to water their plants or don’t have time to pay attention to how much water a plant needs at a particular time. Not providing the right amount of water may prove hazardous to the plant. Hence, we came up with a plant watering system that monitors the soil moisture of the plant, the temperature and humidity of the environment. Using this information, providing the plant with the right amount of water becomes easy.

4. Proposed System

4.1 Block Diagram

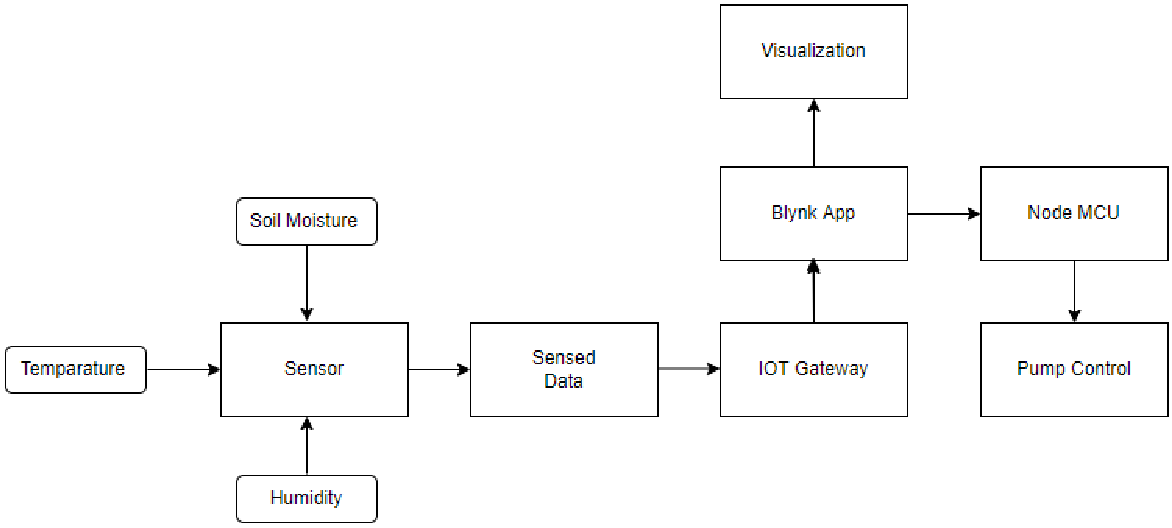

Figure 4.1 Block Diagram

4.2 Flow Chart

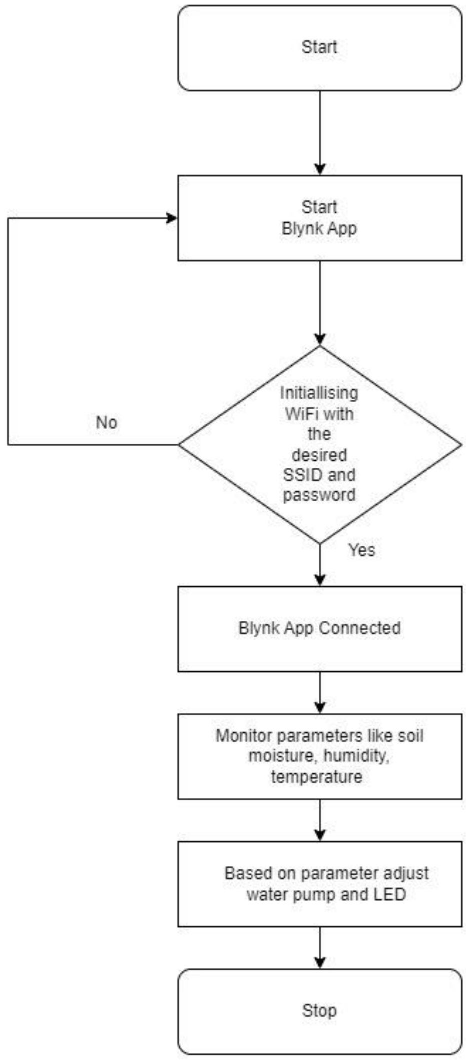

Figure 4.2 Flow Chart

5. Components

5.1 Hardware:

5.1.1 NodeMCU

NodeMCU v3 is a development board which runs on the ESP8266 with the Espress if non-OS SDK, and hardware based on the ESP-12 module. The device features 4MB of flash memory, 80MHz of system clock, around 50k of usable RAM and an on chip WIFI Transceiver.

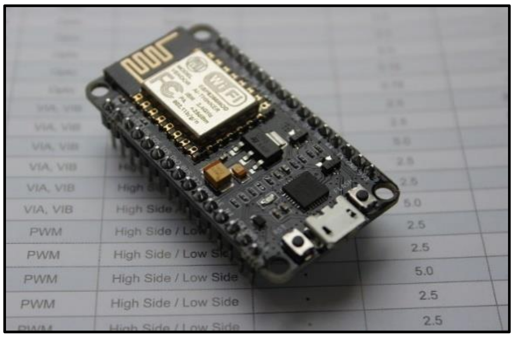

Figure 5.1.1A NODEMCU

Features:

Microcontroller: Tensilica 32-bit RISC CPU Xtensa LX106

Operating Voltage: 3.3V	Input Voltage: 7-12V
Digital I/O Pins (DIO): 16	Analog Input Pins (ADC): 1
UARTs: 1	SPIs: 1
Flash Memory: 4 MB	SRAM: 64 KB
Clock Speed: 80 MHz	Wi-Fi: IEEE 802.11 b/g/n:

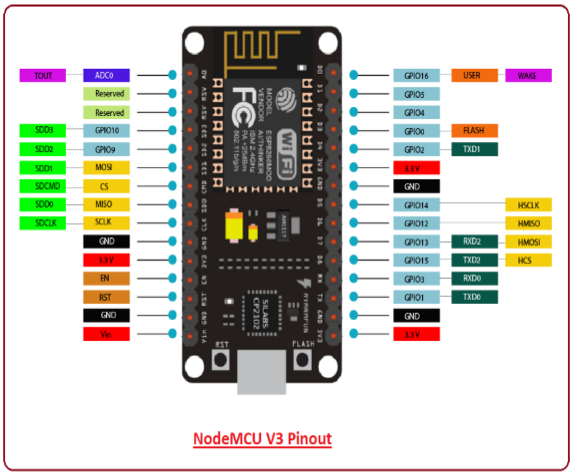

Figure 5.1.1B NodeMCU PIN Diagram

5.1.2 Soil Moisture Sensor

The moisture of the soil plays an essential role in the irrigation field as well as in gardens for plants. The soil moisture sensor is one kind of sensor used to gauge the volumetric content of water within the soil. As the straight gravimetric dimension of soil moisture needs eliminating, drying, as well as sample weighting. These sensors measure the volumetric water content not directly with the help of some other rules of soil like dielectric constant, electrical resistance, otherwise interaction with neutrons, and replacement of the moisture content. The FC-28 soil moisture sensor includes 4-pins:

- VCC pin is used for power
- A0 pin is an analog output
- D0 pin is a digital output
- GND pin is a Ground

This module also includes a potentiometer that will fix the threshold value, & the value can be evaluated by the comparator-LM393. The LED will turn on/off based on the threshold value.

Specifications:

• Working voltage: 5V

• Working current: <20 mA

• Interface: Analog

• Working Temperature: 10°C-30°C

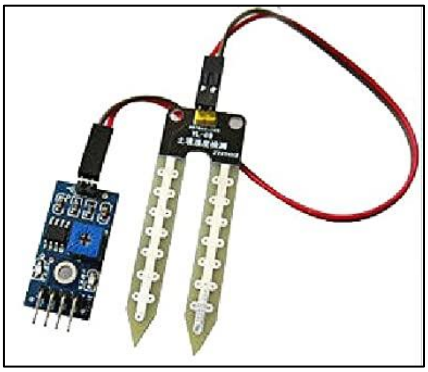

Figure 5.1.2 Soil Moisture Sensor

5.1.3 DHT 11

DHT11 is a low-cost digital sensor for sensing temperature and humidity. This sensor can be easily interfaced with any micro-controller such as Arduino, Raspberry Pi etc… to measure humidity and temperature instantaneously. DHT11 humidity and temperature sensor is available as a sensor and as a module. The difference between this sensor and module is the pull-up resistor and a power-on LED. DHT11 is a relative humidity sensor. To measure the surrounding air this sensor uses a thermistor and a capacitive humidity sensor. DHT11 sensor consists of a capacitive humidity sensing element and a thermistor for sensing temperature. The humidity sensing capacitor has two electrodes with a moisture holding substrate as a dielectric between them. Change in the capacitance value occurs with the change in humidity levels. The IC measure, process this changed resistance values and change them into digital form. The temperature range of DHT11 is from 0 to 50 degree Celsius with a 2-degree accuracy. Humidity range of this sensor is from 20 to 80% with 5% accuracy. The sampling rate of this sensor is 1Hz . i.e. it gives one reading for every second. DHT11 is small in size with operating voltage from 3 to 5 volts. The maximum current used while measuring is 2.5mA. DHT11 sensor has four pins- VCC, GND, Data Pin and a not connected pin. A pull-up resistor of 5k to 10k ohms is provided for communication between sensor and micro-controller.

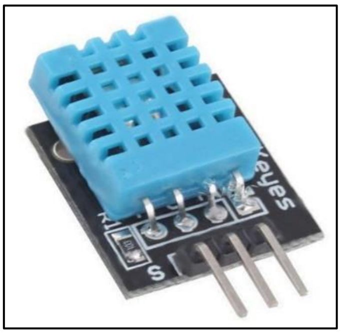

Figure 5.1.3 DHT 11

5.1.4 12V Battery

The nine-volt battery, or 9-volt battery, is an electric battery that supplies a nominal voltage of 9 volts. Actual voltage measures 7.2 to 9.6 volts, depending on battery chemistry.

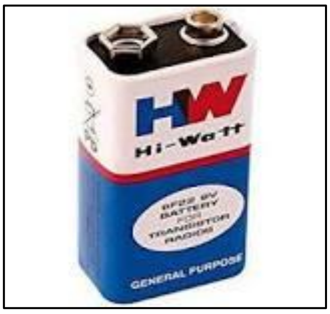

Figure 5.1.4 12V Battery

5.1.5 Breadboard:

A breadboard is a solderless device for temporary prototype with electronics and test circuit designs. Most electronic components in electronic circuits can be interconnected by inserting their leads or terminals into the holes and then making connections through wires where appropriate. The breadboard has strips of metal underneath the board and connect the holes on the top of the board. Note that the top and bottom rows of holes are connected horizontally and split in the middle while the remaining holes are connected vertically.

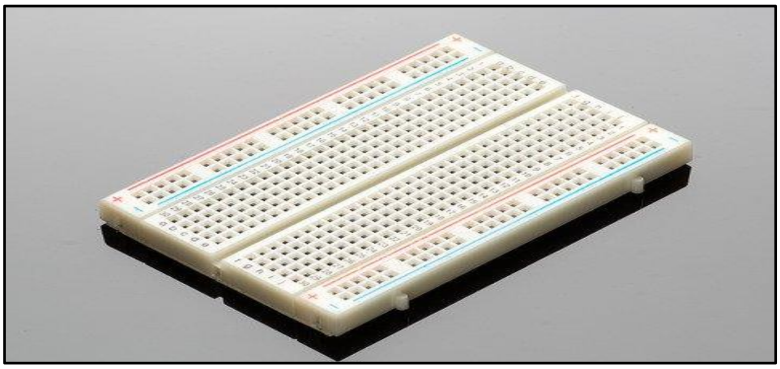

Figure 5.1.5 Breadboard

5.1.6 Jumper Wire

Jumper wires are simply wires that have connector pins at each end, allowing them to be used to connect two points to each other without soldering.

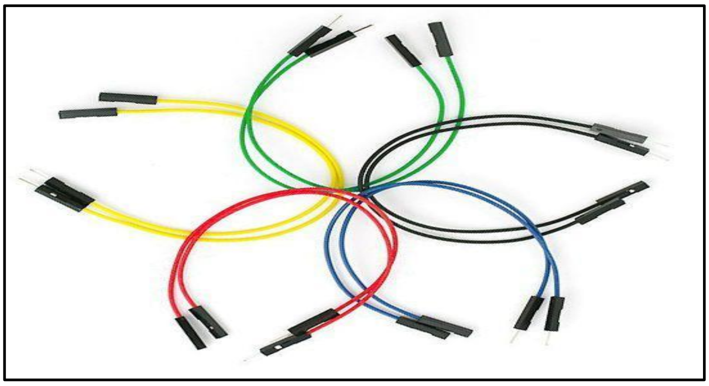

Figure 5.1.6 Jumper Wire

5.1.7 Relay Module

The relay is the device that open or closes the contacts to cause the operation of the other electric control. It detects the undesirable condition with an assigned area and gives the commands to the circuit breaker to disconnect the affected area through ON or OFF. It consists of: Electromagnet, Mechanically movable contact, Switching points and Spring along with:

COM: common pin

NO: Normally open – there is no contact between the common pin and the normally open pin. So, when you trigger the relay, it connects to the COM pin and power is provided to the load.

NC: Normally closed – there is contact between the common pin and the normally closed pin. There is always connection between the COM and NC pins, even when the relay is turned off. When you trigger the relay, the circuit is opened and there is no supply provided to the load.

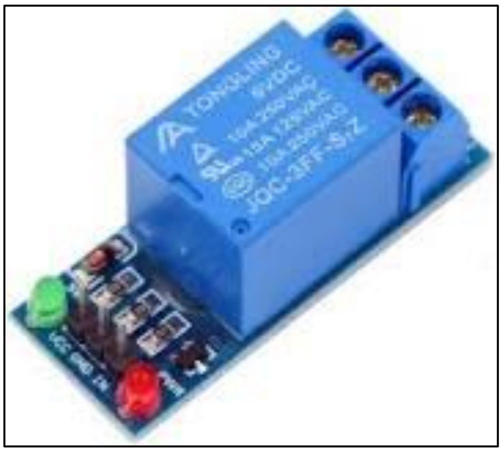

Figure 5.1.7 Relay Module

5.1.8 Submersible Water Pump

The solenoid valve is normally closed. It requires 12V DC to completely open. Therefore, the degree of opening can be decided by the voltage supplied to it. It is connected to the pipes in the project, which direct the water to a part of the field. To select which valve has to be opened, a logical circuit has to be used which is again a combination of Demultiplexer IC 74139 and IC 7404 as defined and stated earlier.

Specification:

- Maximum Flow Rate: 100 Litters Per Hour

- Power Source: Battery Powered

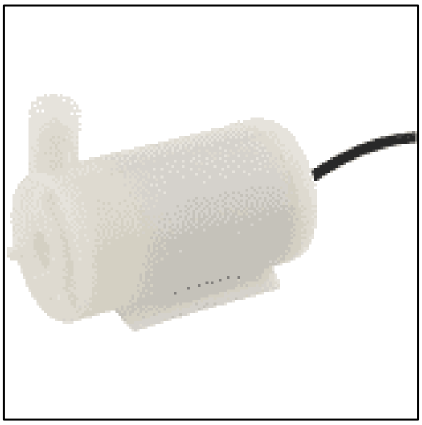

Figure 5.1.8 Submersible Water Pump

5.1.9 LED

A light-emitting diode (LED) is a semiconductor device that emits light when current flows through it. Electrons in the semiconductor recombine with electron holes, releasing energy in the form of photons.

Figure 5.1.9 LED

5.2 Software

5.2.1 Arduino IDE

The Arduino integrated development environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in the programming language Java. It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards.

Figure 5.2.1 Arduino IDE

5.2.2 Blynk App

Blynk is a platform that allows you to control Arduino, Raspberry Pi, and other devices via the Internet using IOS and Android applications. It's a digital dashboard where you may drag and drop widgets to create a graphic interface for your project. Blynk is a programme that allows you to create your own apps. It can be applied to a single project or several of them. Virtual LEDs, buttons, value displays, and even a text terminal, as well as the ability to interact with one or more devices, may be incorporated in any project.

Figure 5.2.1 Blynk App

6. Implementation

6.1 Circuit Diagram

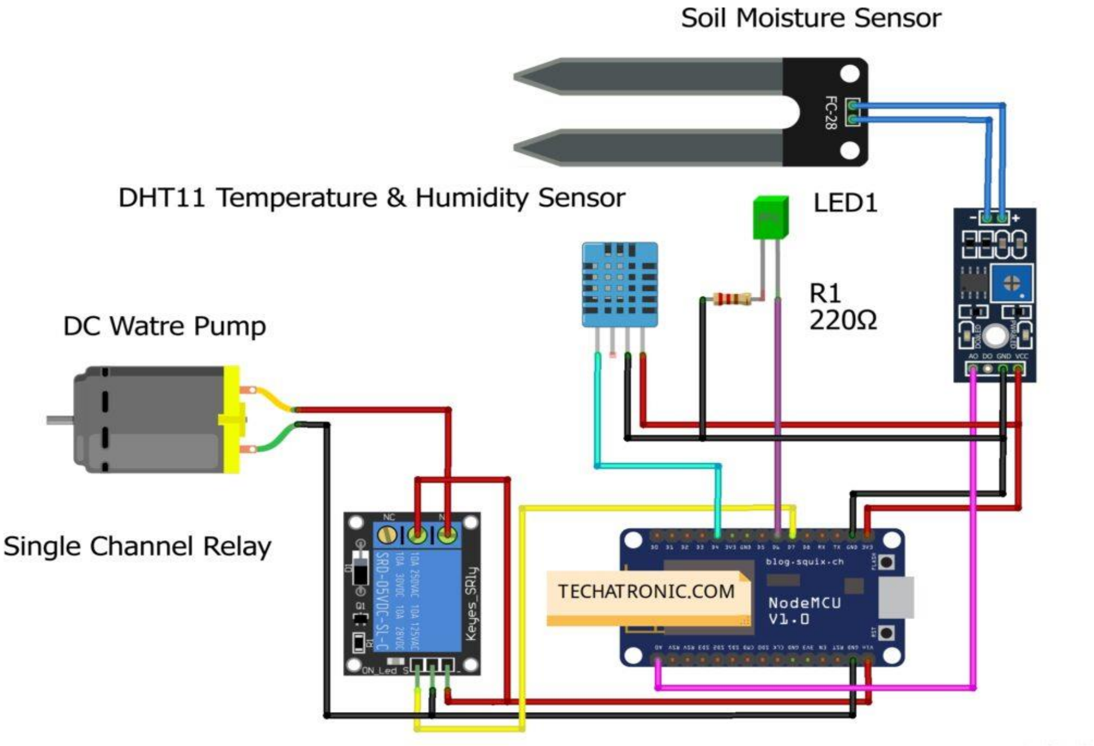

Figure 6.1 Circuit Diagram

6.2 Working

With the help of the plant watering system, we can monitor the moisture level of the soil using a soil moisture sensor along with the temperature and humidity levels of the environment the plant is in using the temperature and humidity sensor (DHT). The project also enables us to control the heat in case the plant is indoors through mobile interface.

For a mobile interface, we have used Blynk app which allows us to view the moisture, temperature and humidity level and also enables us the control the light (heat) supply and also water supply on low soil moisture levels for the plant.

The project requires an internet connection as it uses a NodeMCU to connect the sensors to the Blynk app. When the soil moisture level is low it indicates that the plant needs watering, this can be one by plugging in the water pump to the power supply and supply water to the plant till the soil moisture level is 100.

7. Results & Discussion

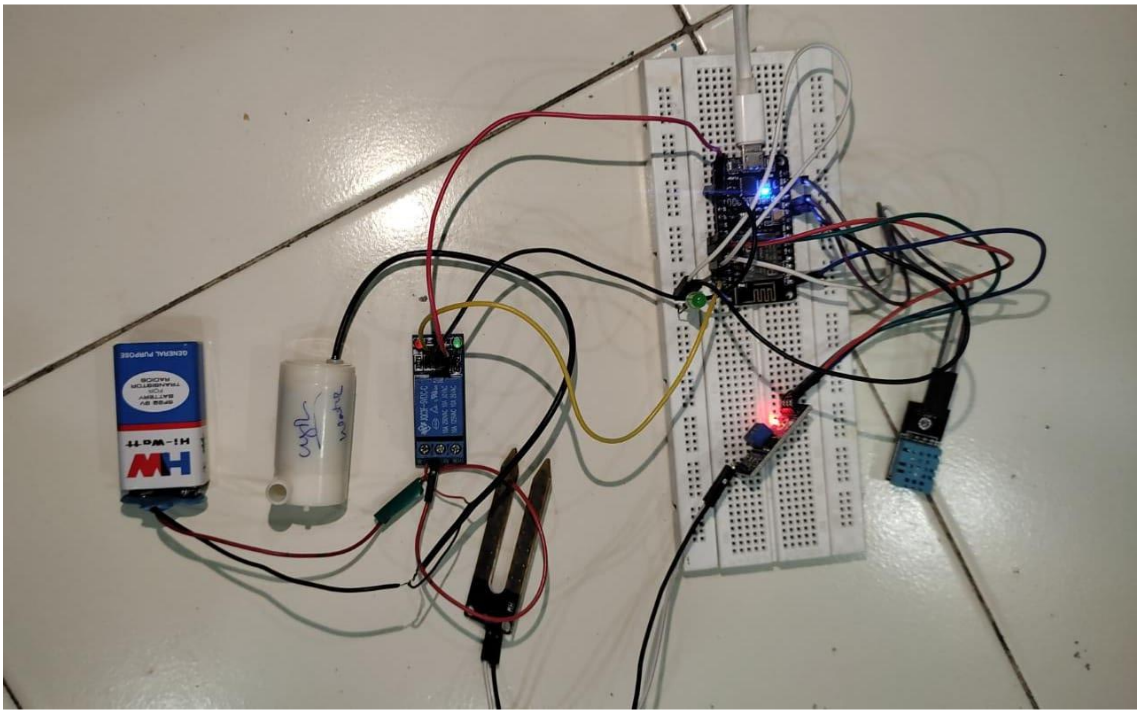

Figure 7.1 Apparatus with connection

7.1 Discussion

As all the testing was done with satisfactory result. The system works with moisture and DHT11 (temperature and humidity) sensor which takes reading according to the current room temperature and humidity. Readings from the moisture sensor in the circuit also depend on what the current moisture level is for the plant. Otherwise, overall result coming out from the circuit in terms of functionality was good for motivation.

8. Conclusion & Future Scope

From this work, we can control the moisture content of the soil of cultivated land. According to soil moisture, water pumping motor turned on or off via the relay automatically. This saves water, while the water level can be obtained in a preferred aspect of the plant, thereby increasing productivity of crops. Submersible water pump from vegetation water uniformly dispersed in water, in order to ensure the maximum utilization of absorption through. Thus, there is minimal waste of water. The system also allows the delivery to the plant when needed based on the type of plant, soil moisture, and observed temperature. The proposed work minimizes the efforts of major agricultural regions. Many aspects of the system can be customized and used software to fine-tune the requirements of the plant. The result is a scalable, supporting technology. Using this sensor, we can see that the soil is wet or dry. If it is dry, the motor will automatically start pumping water.

References

[1] https://www.electronicwings.com/users/vinaybadiger96/projects/181/iot-based-smart-irrigation-system

[2] https://circuitdigest.com/microcontroller-projects/iot-based-smart-agriculture-moniotring-system

[3] https://www.viralsciencecreativity.com/post/iot-smart-plant-monitoring-system

[4] https://extension.okstate.edu/fact-sheets/smart-irrigation-technology-controllers-and-sensors.html

[5] https://nodemcu.readthedocs.io/en/master/

[6] https://blynk.io/

[7] https://examples.blynk.cc/?board=ESP8266&shield=ESP8266%20WiFi&example=GettingStarted%2FBlynkBlink

[8] 2017, IEEE paper : IoT based smart irrigation monitoring and controlling system, (authors : Shweta B saraf, Dhanashree h. Gawali)

Appendix

Code

```
#define BLYNK_TEMPLATE_ID "TMPLETlBi8uz"
#define BLYNK_DEVICE_NAME "Quickstart Template"
#define BLYNK_AUTH_TOKEN "6nGu88ExNXvR-aVt_OHkaDvH8k3Umm70"
#define BLYNK_PRINT Serial
#include <SPI.h>
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <Simpletimer.h>
#include <DHT.h>
#define BLYNK_PRINT Serial
#include <OneWire.h>
#include <DallasTemperature.h>
#define ONE_WIRE_BUS D2
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);

char auth[] = BLYNK_AUTH_TOKEN;              //Authentication code sent by Blynk
char ssid[] = "Sanket";                       //WiFi SSID
char pass[] = "12345678";                       //WiFi Password

#define sensorPin D3
#define relay D0
int sensorState = 0;
int lastState = 0;
#define DHTPIN 2
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
SimpleTimer timer;
void sendSensor()
{
  float h = dht.readHumidity();
  float t = dht.readTemperature();

  if (isnan(h) || isnan(t)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }

  Blynk.virtualWrite(V5, h);  //V5 is for Humidity
  Blynk.virtualWrite(V6, t);  //V6 is for Temperature
}

BLYNK_WRITE(V10) {
```

```
  int switchv10 = param.asInt();
  digitalWrite(D0, switchv10);
  Serial.println(switchv10);
}
void setup()
{
  Serial.begin(9600);
  Blynk.begin(auth, ssid, pass);
   pinMode(sensorPin, INPUT);
   pinMode(relay, OUTPUT);
  dht.begin();

  timer.setInterval(1000L, sendSensor);
  Serial.begin(115200);
  Blynk.begin(auth, ssid, pass);
   sensors.begin();
}
int sensor=0;
void sendTemps()
{
sensor=analogRead(A0);
sensors.requestTemperatures();
float temp = sensors.getTempCByIndex(0);
Serial.println(temp);
Serial.println(sensor);
Blynk.virtualWrite(V1, temp);
Blynk.virtualWrite(V2, sensor);
delay(1000);
}
void loop()
{
  Blynk.run();
  timer.run();
  sendTemps();
  sensorState = digitalRead(sensorPin);
Serial.println(sensorState);

if (sensorState == 1 && lastState == 0) {
  Serial.println("needs water, send notification");
  Blynk.notify("Water your plants");
  lastState = 1;
//  delay(1000);
//send notification

  }
  else if (sensorState == 1 && lastState == 1) {
    //do nothing, has not been watered yet
  Serial.println("has not been watered yet");
//  delay(1000);
  }
```

```
  else {
    Serial.println("does not need water");
    lastState = 0;
//    delay(1000);
  }

  delay(100);
}
```