Pre Project Report

Automated Plant Watering System

EE331
Embedded Systems [Laboratory]

[Group 3]

Members: Divyanshu Lakra, Nitin, Vinod Kumar

Roll No: 2104210, 2104222, 2104241

School of Electrical Science and Engineering
Indian Institute of Technology, Goa

Automated Plant Watering System

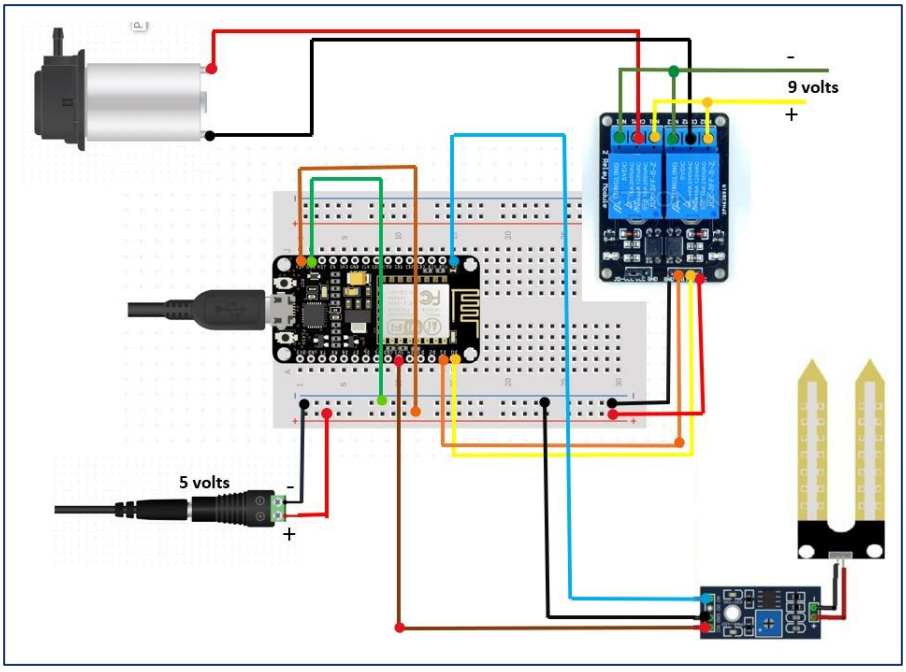
❖ Selection of Components:

I. ESP8266 Microcontroller
II. Soil Moisture Sensor
III. 9V Relay
IV. Water Pump
V. 9V Battery
VI. LED (for testing)
VII. BreadBoard
VIII. Jumper wires

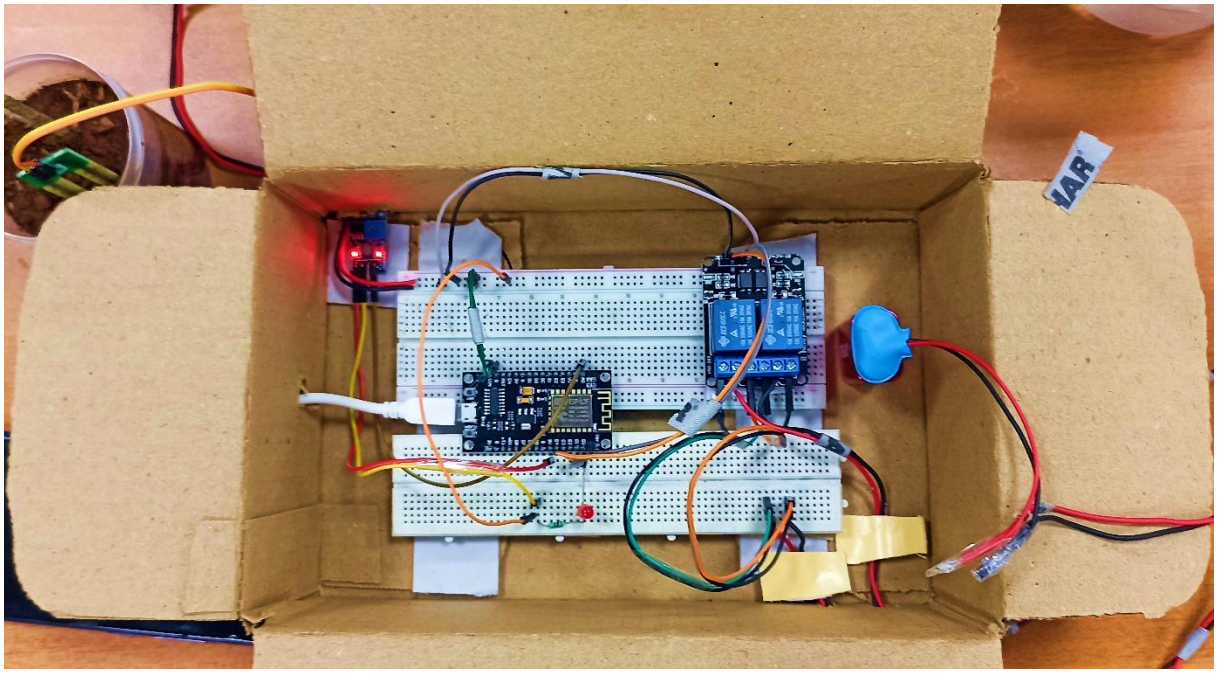
❖ Distribution of work:

I. Divyanshu (2104210): Coding in Arduino IDE and Report
II. Vinod (2104241): Circuit Diagram and Connections
III. Nitin (2104222): Blynk Connections

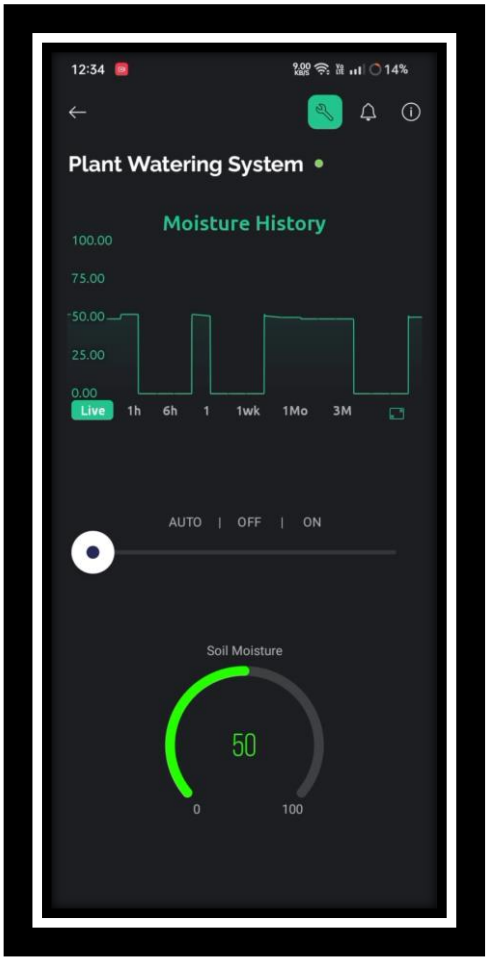
❖ Circuit Diagram:



❖ Real Life Circuit connections:



❖ Blynk App Interface:



❖ Explanation of hardware or circuit connections:

Important connections Explanation (Excluding VCC and GND)

1. ESP8266 Connections:

I. **Pin A0**: Connected to the analog output of the soil moisture sensor to take the moisture value
II. **Pin D0:** Connected to the control input of the 9V relay module to give the signal according to the code to turn on and off the pump.

2. Relay Connections:

I. **NO1 and NO2**: Connected to +ve of 9V Battery
II. **NC1 and NC2:** Connected to -ve of 9V Battery
III. **COM1 and COM2:** Connected to +ve and -ve of Pump

A two-channel relay can independently control the direction of a motor. Each channel has a Normally Open (NO) contact. To control the motor's forward direction, connect one motor terminal to NO of channel 1 and the other to the power supply's positive terminal. To reverse, connect one terminal to NO of channel 2 and the other to the positive terminal. Apply a control signal to the relay's input pins to switch between forward and reverse

Explanation of Working of System:

The IoT plant watering system functions by using ESP8266 microcontroller as its central processing unit. The system uses a soil moisture sensor, connected to the ESP8266's analog pin A0, to measure soil moisture levels.

The ESP8266 communicates wirelessly with the Blynk app over Wi-Fi to get the commands by the user and Upload the value of Sensor to Display there Enabling users to monitor and control the watering process remotely. In Blynk We developed two mode for the system to have one Auto and one Manual.

When in **Auto mode** If the soil moisture level drops below a predefined threshold, the ESP8266 activates a 9V relay connected to its digital pin D0. This

relay, in turn, controls the power supply to the water pump. The 9V relay is powered by a 9V battery. In **Manual Mode** we can switch on and off the motor remotely by a switch in blynk app.

Additionally, We connected an LED to D0 of the ESP8266 which is used in system testing and debugging.

❖ The IDE Code:

Plant_Watering_System_final.ino

```
/*Plant watering system with new blynk update
    Home Page

*/
#define BLYNK_TEMPLATE_ID "TMPL3ho8CSsug"
#define BLYNK_TEMPLATE_NAME "Plant Watering System"
#define BLYNK_AUTH_TOKEN "uBIF7hQRgfXQ22zdj3400o4Mrcb7a3qk"

//Include the library files
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

char auth[] = "uBIF7hQRgfXQ22zdj3400o4Mrcb7a3qk";  //Enter your Auth token
char ssid[] = "Redmi";                             //Enter your WIFI name
char pass[] = "72580306";                          //Enter your WIFI password

BlynkTimer timer;
bool Relay = 0;

//Define component pins
#define sensor A0

void setup() {
  Serial.begin(9600);
  pinMode(D0, OUTPUT);
  pinMode(D1, OUTPUT);
  digitalWrite(D1, 0);

  Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);

  //Call the function
  timer.setInterval(100L, soilMoistureSensor);
}

//Get the button value
BLYNK_WRITE(V1) {
  int Relay = param.asInt();

  if (Relay == 0) {               //Automatic mode
    int moisture = analogRead(sensor);
    moisture = map(moisture, 0, 1024, 0, 100);
    moisture = (moisture - 100) * -1;
    Serial.println(moisture);
    if (moisture < 20) {    //Threshold value
      digitalWrite(D0, 1);  // turn on the motor
    } else {
      digitalWrite(D0, 0);  // turn off the motor
    }

  } else if (Relay == 1){         //Manual mode
    digitalWrite(D0, 0);          //motor manually turned off by switch
    Serial.println(Relay);
  } else{
    digitalWrite(D0, 1);
    Serial.print(Relay);          //motor manually turned on by switch
  }
}

//Get the soil moisture values for widget
void soilMoistureSensor() {
  int moisture = analogRead(sensor);
  moisture = map(moisture, 0, 1024, 0, 100);
  moisture = (moisture - 100) * -1;

  Blynk.virtualWrite(V0, moisture);
}


void loop() {
  Blynk.run();
  Blynk.syncVirtual(V1);  //Run the Blynk library
  timer.run();  //Run the Blynk timer
}
```

❖ Explanation of programming aspects:

```
#define BLYNK_TEMPLATE_ID "TMPL3ho8CSsug"
#define BLYNK_TEMPLATE_NAME "Plant Watering System"
#define BLYNK_AUTH_TOKEN "uBIF7hQRgfXQ22zdj3400o4Mrcb7a3qk"
```

This block defines the Blynk template id.

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

This block includes the necessary libraries for the ESP8266 Wi-Fi module and the Blynk IoT platform.

```
char auth[] = "uBIF7hQRgfXQ22zdj3400o4Mrcb7a3qk";  //Enter your Auth token
char ssid[] = "Redmi";                             //Enter your WIFI name
char pass[] = "72580306";                          //Enter your WIFI password
```

This block defines the Blynk authentication token (auth), the Wi-Fi network SSID (ssid), and the Wi-Fi password (pass). It then initializes the Blynk library with these credentials and connects to the Blynk cloud server.

```
BlynkTimer timer;
bool Relay = 0;

//Define component pins
#define sensor A0
```

This block declares a BlynkTimer object timer for scheduling periodic tasks and a boolean variable Relay to track the state of the relay (0 for off, 1 for on).
And defines the analog pin A0 for connecting the soil moisture sensor.

```
void setup() {
  Serial.begin(9600);
  pinMode(D0, OUTPUT);
  pinMode(D1, OUTPUT);
  digitalWrite(D1, 0);

  Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);

  //Call the function
  timer.setInterval(100L, soilMoistureSensor);
}

```

This block runs once when the microcontroller is powered on or reset. It initializes the serial communication for debugging, sets the D0 and D1 pins as OUTPUT, and sets D1 LOW (assuming it's connected to an LED for indication). It then begins the Blynk connection and sets up a timer to call the soilMoistureSensor function every 100 milliseconds (0.1 seconds).

```
//Get the button value
BLYNK_WRITE(V1) {
  int Relay = param.asInt();

  if (Relay == 0) {               //Automatic mode
    int moisture = analogRead(sensor);
    moisture = map(moisture, 0, 1024, 0, 100);
    moisture = (moisture - 100) * -1;
    Serial.println(moisture);
    if (moisture < 20) {    //Threshold value
      digitalWrite(D0, 1);  // turn on the motor
    } else {
      digitalWrite(D0, 0);  // turn off the motor
    }

  } else if (Relay == 1){         //Manual mode
    digitalWrite(D0, 0);          //motor manually turned off by switch
    Serial.println(Relay);
  } else{
    digitalWrite(D0, 1);
    Serial.print(Relay);          //motor manually turned on by switch
  }
}
```

The Blynk write function for virtual pin V1 in the provided sketch controls the operation mode of a plant watering system. When the value of V1 changes in the Blynk app, the function reads this value to determine the mode: 0 for automatic, 1 for manual off, and 2 for manual on. In automatic mode, the function reads the soil moisture sensor and turns on the motor (connected to pin D0) if the moisture level is below a threshold (moisture < 20); otherwise, it turns off the motor. In manual mode, the function either turns off (Relay = 1) or turns on (Relay = 2) the motor based on the user's choice, providing flexibility in controlling the watering process to suit the plant's needs.

```
//Get the soil moisture values for widget
void soilMoistureSensor() {
  int moisture = analogRead(sensor);
  moisture = map(moisture, 0, 1024, 0, 100);
  moisture = (moisture - 100) * -1;

  Blynk.virtualWrite(V0, moisture);
}
```

This block defines a function soilMoistureSensor that reads the soil moisture sensor and maps it to 0 to 100 and sends the value to virtual pin V0 in the Blynk app for display.

```
void loop() {
  Blynk.run();
  Blynk.syncVirtual(V1);  //Run the Blynk library
  timer.run();  //Run the Blynk timer
}
```

This block is the main loop function that runs continuously. It runs the Blynk library to handle communication, syncs the virtual pin V1 to ensure the correct state is maintained, and runs the Blynk timer to trigger scheduled tasks.

❖ Challenges faced and their respective solutions:

Problem: Soil Moisture sensor not giving the proper readings.
Solution: We made a circuit to calibrate the sensitivity of soil moisture sensor as mentioned in this site.
https://srituhobby.com/how-to-make-a-plant-watering-system-with-the-nodemcu-esp8266-board-and-the-new-blynk-update/

Problem: We couldn't make two switches to interact with each other as blynk doesn't allow their write function to be called in another one. i.e no isolated switch for turning the mode to manual and auto
Solution: We made a slider with three input which represents 3 states as mentioned in the code to function 1st state as Auto mode and and the other two as Manual mode ON and OFF.

❖ Key Learnings and outcomes:

We were working first time with ESP board and the IOT Blynk so we learned a lot about these in the process. At first our project was Submarine which couldn't be completed for time and resources reasons So we also learned that we must decide a project which could be completed in the given time frame by utilizing the available resources to get the most out of it. Building a project comes with many challenges and we tried facing them which was a good learning experience. Finally when we shifted to Automated Plant Watering System we applied all our past experience and completed it in the time frame.

❖ References, video links, or online resources used:

Videos:

I. https://youtu.be/WLCR0r7rYi8?si=lDGuJd2X6rM85pPO
II. https://youtu.be/Ix3a3ThsHfA?si=ILLyncmYo3o-Z1j7
III. https://youtu.be/17PBh9JFKzE?si=suq4z_AgIxGufjJe

Websites:

I. https://srituhobby.com/how-to-make-a-plant-watering-system-with-the-nodemcu-esp8266-board-and-the-new-blynk-update/
II. https://www.instructables.com/DIY-Relay-switch-motor-controller-Arduino/
III. https://circuitdiagrams.in/iot-plant-watering-system-using-esp8266/
IV. https://projecthub.arduino.cc/Aswinth/soil-moisture-sensor-with-arduino-91c818