Embedded Systems Lab

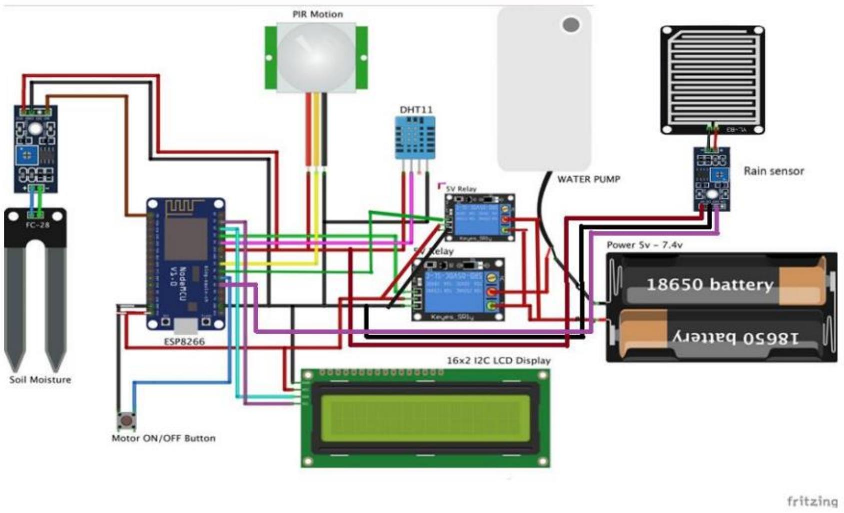
Problem Statement - SMART PLANT IRRIGATION SYSTEM

Agriculture, the backbone of many societies, faces a critical challenge: **inefficient water management**. Traditional irrigation methods often **lack precision**, leading to **overwatering** or **underwatering**, which wastes precious resources and harms crop yields.

This project proposes a **solution** – an **Automated Irrigation System** powered by embedded systems. By incorporating sensors that measure **soil moisture**, **temperature**, and **humidity**, the system can deliver water only when necessary and in optimal quantities. This targeted approach aims to significantly improve water usage efficiency. With less water wasted, more resources are available for agriculture and other needs. Additionally, by providing the right amount of water at the right time, the system can **enhance crop yields**, leading to **increased food production** and **economic benefits** for farmers.

Ultimately, this approach promotes sustainable agricultural practices, ensuring a healthier environment and a more secure food supply for the future.

Schematic Diagram



C Code

```
#define BLYNK_TEMPLATE_ID "TMPL3b7xD9arn"
#define BLYNK_TEMPLATE_NAME "Plant monitoring system using IoT"
#include <LiquidCrystal_I2C.h>
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <DHT.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
char auth[] = "YckrZE3eotjCioRh9rhFoULoW3jndYZx";
char ssid[] = "AKiPhone";
char pass[] = "ri0336699";
DHT dht(D4, DHT11);
BlynkTimer timer;

#define soil A0
#define PIR D5
#define RAIN_SENSOR D8 // Digital pin for rain sensor
int PIR_ToggleValue;
int soilThreshold = 30; // Set the soil moisture threshold to 30%

void checkPhysicalButton();
int relay1State = LOW;
int pushButton1State = HIGH;
#define RELAY_PIN_1 D3
#define PUSH_BUTTON_1 D7
#define VPIN_BUTTON_1 V12
double T, P;
char status;
void updateLCD();
void setup() {
        Serial.begin(9600);
        lcd.begin();
        lcd.backlight();
```

```
    pinMode(PIR, INPUT);
    pinMode(D8, INPUT); // Set rain sensor pin as input
    pinMode(D6, OUTPUT);
    pinMode(RELAY_PIN_1, OUTPUT);
    digitalWrite(RELAY_PIN_1, LOW);
    pinMode(PUSH_BUTTON_1, INPUT_PULLUP);
    digitalWrite(RELAY_PIN_1, relay1State);

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

    lcd.setCursor(0, 0);
    lcd.print(" Initializing ");
    for (int a = 5; a <= 10; a++) {
        lcd.setCursor(a, 1);
        lcd.print(".");
        delay(500);
    }
    lcd.clear();
    lcd.setCursor(11, 1);
    lcd.print("W:OFF");

    timer.setInterval(100L, soilMoistureSensor);
    timer.setInterval(100L, DHT11sensor);
    timer.setInterval(500L, checkPhysicalButton);
}

void DHT11sensor() {
    float h = dht.readHumidity();
    float t = dht.readTemperature();
    if (isnan(h) || isnan(t)) {
        Serial.println("Failed to read from DHT sensor!");
        return;
    }

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

    Blynk.virtualWrite(V0, t);
    Blynk.virtualWrite(V1, h);
    lcd.setCursor(0, 0);
    lcd.print("T:");
    lcd.print(t);
    lcd.setCursor(8, 0);
    lcd.print("H:");
    lcd.print(h);
}
void soilMoistureSensor() {
    int value = analogRead(soil);
```

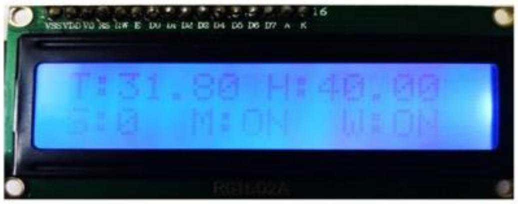
```
       value = map(value, 0, 1024, 0, 100);
       value = (value - 100) * -1;
       bool isRaining = digitalRead(RAIN_SENSOR) == LOW;
       Serial.println(value);
       if (value < soilThreshold) {
              digitalWrite(D6,LOW); // Turn on motor if soil moisture is below
              threshold relay1State = LOW;
              lcd.setCursor(11, 1);
              lcd.print("W:ON ");
       }

       else {
        digitalWrite(D6,HIGH); // Turn off motor if soil moisture is above threshold
        relay1State = HIGH; lcd.setCursor(11, 1);
        lcd.print("W:OFF");
       }

       if(isRaining == HIGH) {
             digitalWrite(D6,HIGH);
             lcd.setCursor(11, 1);
             lcd.print("W:OFF");
             clearAndDisplay("Rain is detected");
       }

       Blynk.virtualWrite(V3, value);
       lcd.setCursor(0, 1);
       lcd.print("S:");
       lcd.print(value);
       lcd.print(" "); // Check if soil moisture is below threshold

       Serial.print(value);
       Serial.println(soil);
       }

 void PIRsensor() {
       bool value = digitalRead(PIR);
       if (value) {
              Blynk.logEvent("pirmotion", "WARNING! Motion Detected!");
              WidgetLED LED(V5);
             LED.on();
       } else {
             WidgetLED LED(V5);
             LED.off();
               }
       }
 BLYNK_WRITE(V6) {
       PIR_ToggleValue = param.asInt();
 }
BLYNK_CONNECTED() {
       Blynk.syncVirtual(VPIN_BUTTON_1);
}
```

```
BLYNK_WRITE(VPIN_BUTTON_1) {
        relay1State = param.asInt();
        digitalWrite(RELAY_PIN_1, relay1State);
}
void checkPhysicalButton() {
        int buttonState = digitalRead(PUSH_BUTTON_1);
        if (buttonState == LOW && pushButton1State == HIGH) {
                relay1State = !relay1State; // Toggle relay state
                digitalWrite(RELAY_PIN_1, relay1State);
                Blynk.virtualWrite(VPIN_BUTTON_1, relay1State);
                updateLCD();
        }

        pushButton1State = buttonState; // Update pushButton1State for next iteration }
        void loop() {
                if (PIR_ToggleValue == 1) {
                        lcd.setCursor(5, 1);
                        lcd.print("M:ON ");
                        PIRsensor();
                } else {
                        lcd.setCursor(5, 1);
                        lcd.print("M:OFF");
                        WidgetLED LED(V5);
                        LED.off();
                }
                Blynk.run();
                timer.run();
 }
 void clearAndDisplay(const char* message) {
        lcd.clear();
        lcd.setCursor(1, 0);
        lcd.print(message);
        delay(5000); // Display message for 5 seconds
        lcd.clear();
 }

 void updateLCD() {
        if (relay1State == LOW) {
               lcd.setCursor(11, 1);
               lcd.print("W:ON ");
        } else if (relay1State == HIGH) { // corrected

        lcd.setCursor(11, 1);
        lcd.print("W:OFF");
        }
```

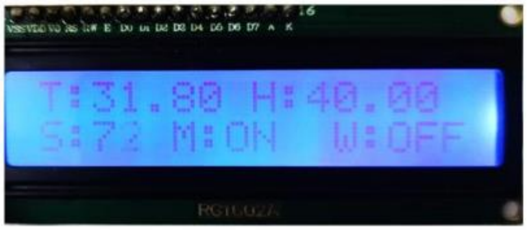
Output Screen Shots

1. Soil Moisture Output

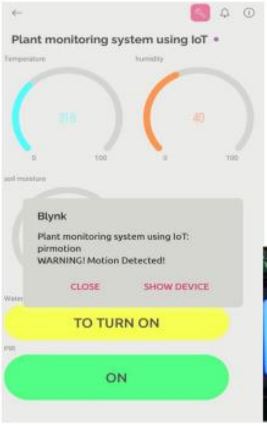
Soil moisture = 0
Motor is on



Soil moisture = 72
Motor is off



2. PIR Sensor



Plant monitoring system using IoT

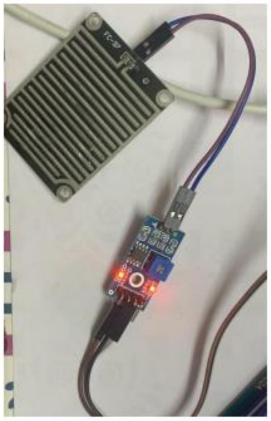
Blynk

Plant monitoring system using IoT: pirmotion
WARNING! Motion Detected!

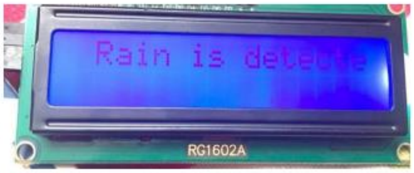
PIR sensors is on
M: on

3. Rain Sensor

Rain is not detected



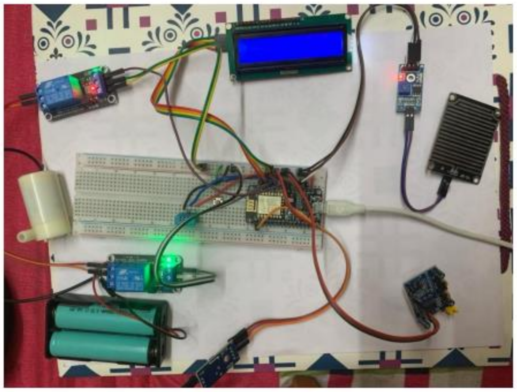
Rain water drop is detected
And displayed in lcd



4. DHT11

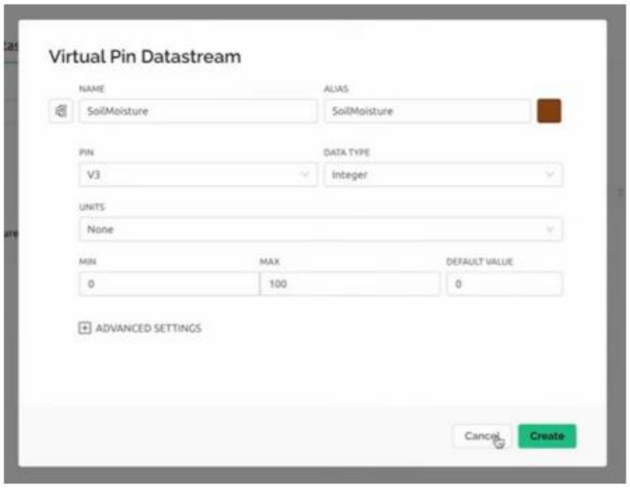
T:31.80'c
H:47'c

T:32.30'c
H:47'c



BLINK

Blink Configuration



Virtual Pin Datastream

NAME: SoilMoisture

ALIAS: SoilMoisture

PIN: V3

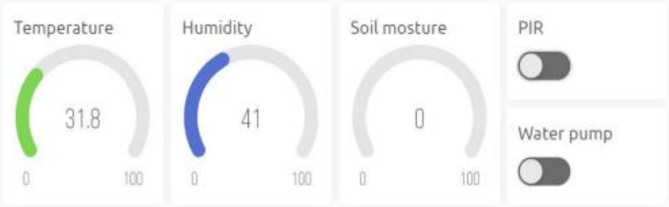
DATA TYPE: Integer

UNITS: None

MIN: 0

MAX: 100

DEFAULT VALUE: 0



Temperature
31.8
0
100
Humidity
41
0
100
Soil mosture
0
0
100
PIN
Water pump