Name of Student: Pushkar Sane		
Roll Number: 45	**Lab Assignment Number: Mini Project**	
Title of Lab Assignment: Mini Project (Arduino Based Plant monitoring system.)		
DOP: 05-03-2024	**DOS: 12-03-2024**	
CO Mapped: **CO3, CO4**	**PO Mapped:** **PO1, PO2, PO5, PO7, PSO1**	**Signature:**

IOT Mini-Project

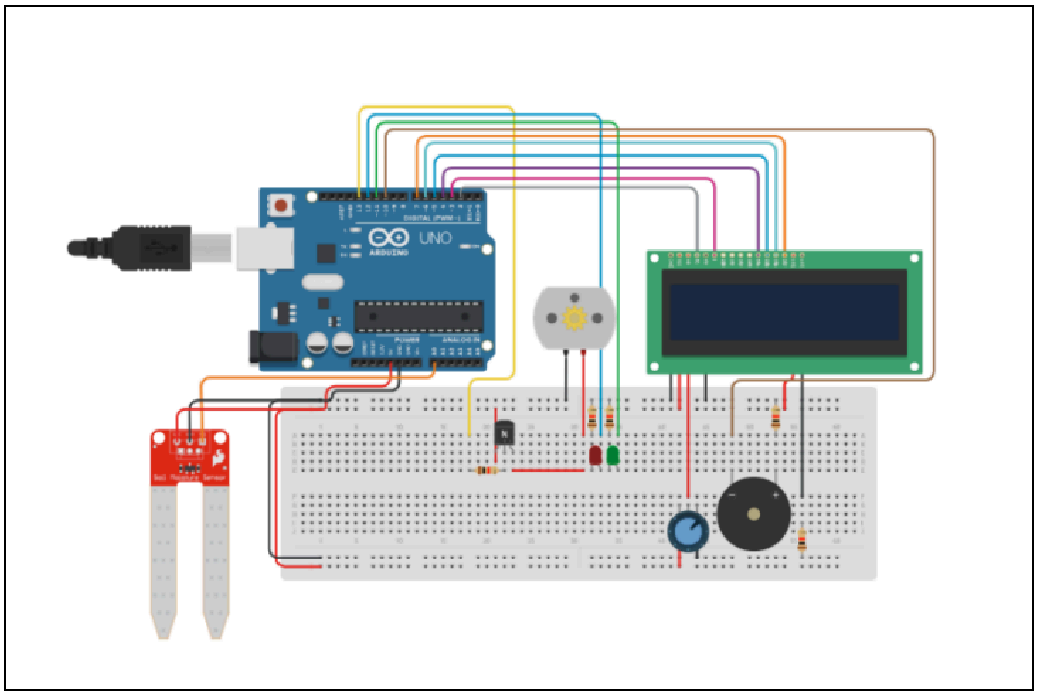
Topic: Arduino Based Plant monitoring system.

Theory:
An Arduino-based plant monitoring system is a project aimed at providing real-time data about the environment and conditions surrounding a plant. By using sensors connected to an Arduino board, this system can measure parameters such as soil moisture, temperature, light intensity, and humidity. This data can then be used to monitor the health of plants and optimize their growth conditions, helping gardeners and farmers to ensure that their plants are thriving.

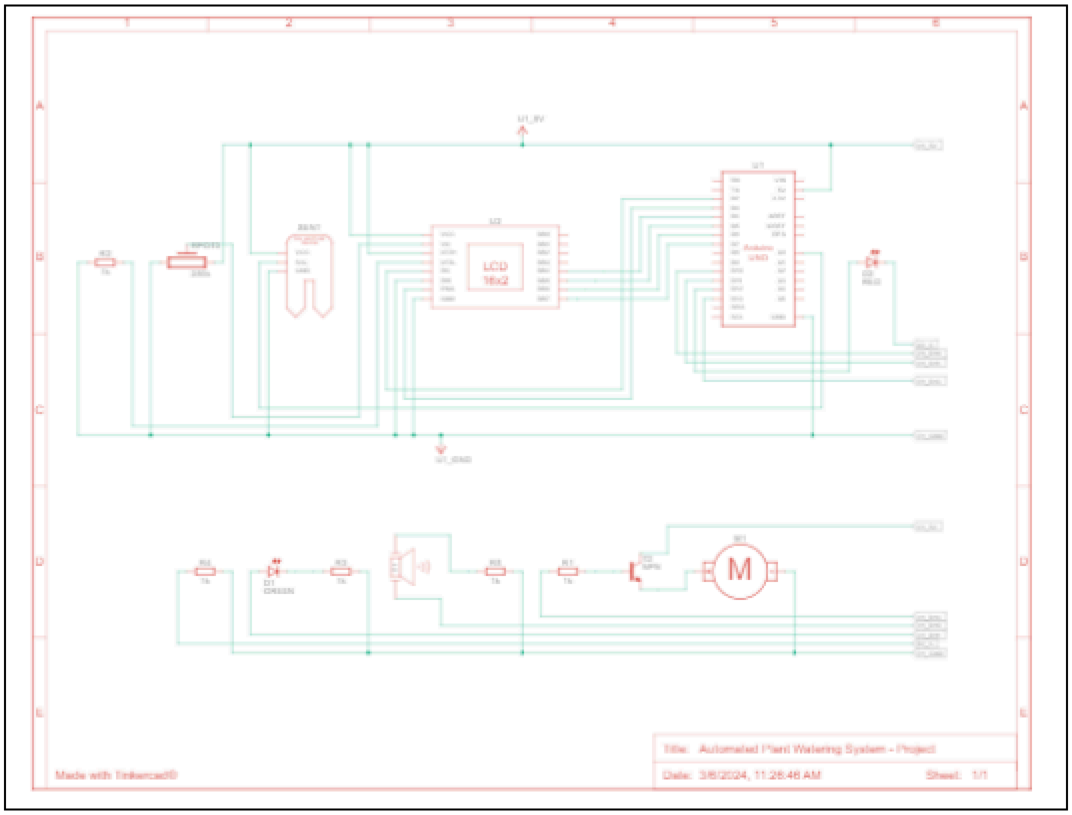
Project Objective:

- Monitor Soil Moisture: Measure the moisture level in the soil to ensure plants are adequately watered.
- Measure Temperature and Humidity: Monitor the environmental conditions to ensure they are within the optimal range for plant growth.
- Monitor Light Intensity: Measure the amount of light available to the plants to ensure they are receiving sufficient light for photosynthesis.
- Real-time Data Display: Display the collected data in real-time, allowing users to easily monitor the conditions.
- Alert System: Implement an alert system to notify users when certain conditions (e.g., soil too dry) are met, so they can take action.
- Data Logging: Log the collected data over time to analyze trends and optimize plant care strategies.
- Remote Monitoring: Enable users to monitor the plant conditions remotely, using a web or mobile interface.
- Automation: Implement automation features to control watering systems or adjust light sources based on sensor readings.
- Energy Efficiency: Design the system to be energy-efficient, using low-power components and sleep modes when not in use.
- Expandability: Design the system with expandability in mind, allowing users to add more sensors or features in the future.

Circuit Diagram:



Schematic Diagram:

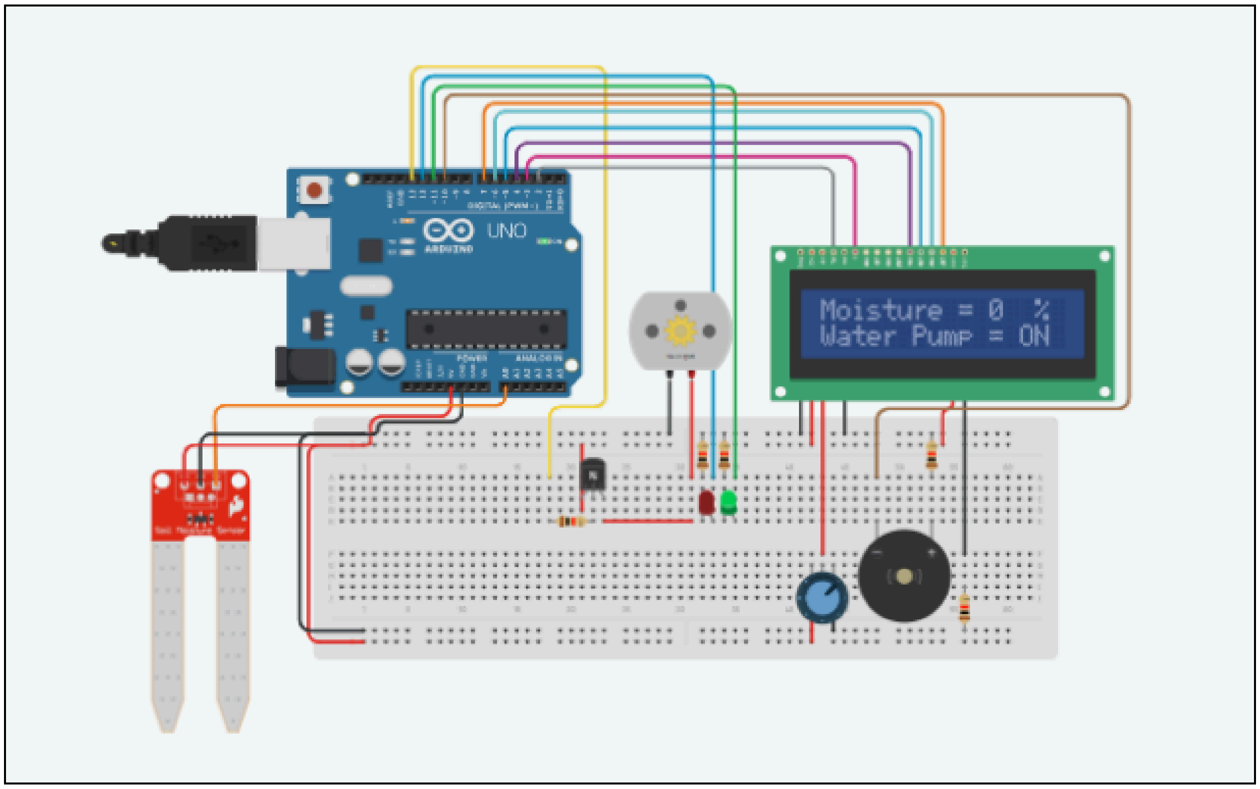

U1_5V
U1_GND
LCD 16x2
Arduino Uno
M
Title: Automated Plant Watering System - Project
Date: 3/8/2024, 11:28:48 AM
Sheet: 1/1
Made with Tinkercad®

Code:

```
#include <LiquidCrystal.h>
// Include the LiquidCrystal library, which provides functions to control the LCD display.
// Declaring variables
const int BuzzerPin = 10; // Pin connected to the piezo buzzer
const int LedRed = 12; // Pin connected to the red LED
const int LedGreen = 11; // Pin connected to the green LED
const int SoilMoistureSensor = A0; // Pin connected to the soil moisture sensor
const int WaterPump = 13; // Pin connected to the water pump relay
LiquidCrystal lcd(2, 3, 4, 5, 6, 7); // Create an instance of the LiquidCrystal class to control the
LCD
void setup() {
  pinMode(WaterPump, OUTPUT); // Set the water pump pin as output
  pinMode(LedRed, OUTPUT); // Set the red LED pin as output
  pinMode(LedGreen, OUTPUT); // Set the green LED pin as output
  pinMode(BuzzerPin, OUTPUT); // Set the piezo buzzer pin as output
  Serial.begin(9600); // Initialize serial communication at 9600 bps
  lcd.begin(16, 2); // Initialize the LCD with 16 columns and 2 rows
  pinMode(BuzzerPin, OUTPUT); // Set the piezo pin as output
  lcd.clear(); // Clear the LCD display
  lcd.setCursor(0, 0); // Set the cursor to the first column and first row
  String message1 = "Automated Plant";
  String message2 = "Watering System";
   // Display "Automated Plant" on the first row of the LCD with a delay of 100ms between
characters
  for (int i = 0; i < message1.length(); i++) {
    lcd.print(message1.charAt(i));
    delay(100);
  }
  lcd.setCursor(0, 1); // Set the cursor to the first column and second row
    // Display "Watering System" on the second row of the LCD with a delay of 100ms between
characters
    for (int i = 0; i < message2.length(); i++) {
        lcd.print(message2.charAt(i));
        delay(100);
    }
    delay(2500); // Delay for 2.5 seconds to display the messages
    lcd.clear(); // Clear the LCD display again
    lcd.setCursor(0, 0); // Set the cursor to the first column and first row
    lcd.print("Moisture ="); // Display "Moisture =" on the first row
    lcd.setCursor(0, 1); // Set the cursor to the first column and second row
    lcd.print("Water Pump ="); // Display "Water Pump =" on the second row
}

void loop(){
      int Sensor = analogRead(SoilMoistureSensor); // Read the value from the soil moisture
sensor
     int mappedValue = map(Sensor, 0, 876, 0, 99); // Map the sensor value to a range from 0 to
99
    lcd.setCursor(11, 0); // Set the cursor to the 12th column and first row
    lcd.print(mappedValue); // Display the mapped value (moisture percentage) on the first row
    lcd.setCursor(14, 0); // Set the cursor to the 15th column and first row
    lcd.print("%"); // Display the percentage symbol on the first row
    lcd.setCursor(13, 1); // Set the cursor to the 14th column and second row
    // Control the water pump and LEDs based on the moisture percentage
    if (mappedValue < 50) {
        digitalWrite(WaterPump, HIGH); // Turn on the water pump
        digitalWrite(LedGreen, HIGH); // Turn on the green LED
        digitalWrite(LedRed, LOW); // Turn off the red LED
        lcd.print("ON "); // Display "ON" on the second row
        playSound(); // Play a sound using the piezo buzzer
    }
    else {
        digitalWrite(WaterPump, LOW); // Turn off the water pump
        digitalWrite(LedGreen, LOW); // Turn off the green LED
        digitalWrite(LedRed, HIGH); // Turn on the red LED
        lcd.print("OFF"); // Display "OFF" on the second row
    }
}
void playSound() {
    tone(BuzzerPin, 87, 100); // Play a tone of 87 Hz for 100 milliseconds on the piezo buzzer
    delay(1000); // Delay for 1 second
}
```

Output:



Conclusion:

Successfully implemented the IOT Mini Project on Arduino Based Plant monitoring system.