TRAINING AND INTERNSHIP PROGRAM IN AENEXZ TECH PVT LTD

FINAL PROJECT

NAME : KRISHIKA G

BATCH : JUNE-AUGUST

Soil Moisture Based Smart Irrigation System

# Introduction

This project is a smart irrigation system that uses Arduino UNO, LCD Display, LED, Buzzer, and Potentiometer. It continuously monitors the soil's moisture level using an analog sensor input and makes decisions to water the soil when it is dry. The system helps conserve water and ensures proper irrigation.

# Project Overview

The smart irrigation system addresses the problem of water wastage in traditional irrigation. By measuring soil moisture in real-time, the system decides when to water plants. It alerts users with a buzzer and LED while displaying live data on the LCD screen. This makes irrigation efficient and automated.

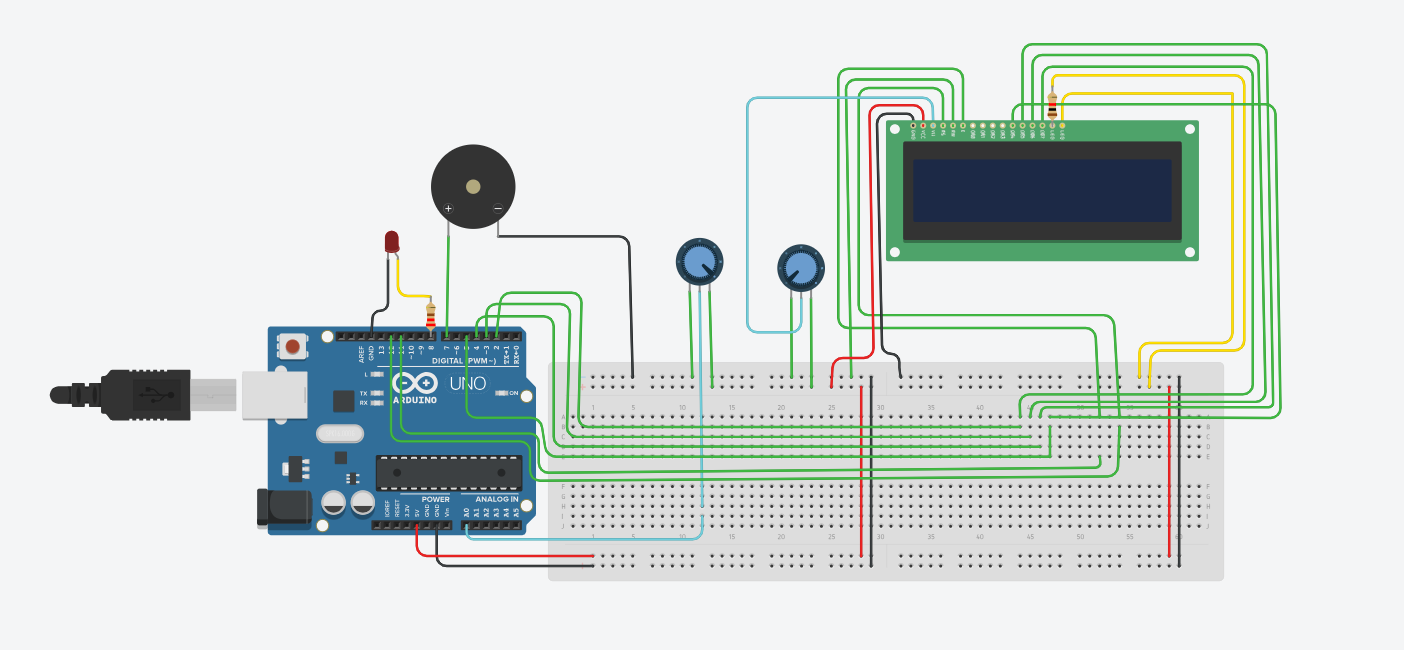
# Components Used

The following components are used in this project:

| **Component** | **Quantity** | **Description** |
| --- | --- | --- |
| Arduino UNO | 1 | Microcontroller board used to process sensor input and control outputs |
| LCD Display (16x2) | 1 | Displays soil moisture readings and system status |
| Potentiometer | 2 | One for adjusting LCD contrast, one simulates the soil moisture sensor |
| LED | 1 | Simulates the water pump (turns ON when soil is dry) |

# Circuit Diagram

The circuit is designed using Arduino UNO as the microcontroller, connected to the LCD, buzzer, LED, and potentiometers. The soil moisture sensor input is simulated with a potentiometer, while the LCD displays moisture level readings.

****

**Arduino code**

#include <LiquidCrystal.h>

// LCD pin configuration: RS, E, D4, D5, D6, D7

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

const int sensorPin = A0; // Soil moisture sensor

const int pumpPin = 8; // LED

const int buzzerPin = 7; // Buzzer

const int threshold = 400; // Moisture threshold

const unsigned long checkInterval = 1000; // Every 5 sec

const unsigned long pumpDuration = 1000; // Run pump for 5 sec

unsigned long lastCheckTime = 0;

void setup() {

lcd.begin(16, 2); // Initialize LCD

pinMode(pumpPin, OUTPUT);

pinMode(buzzerPin, OUTPUT);

digitalWrite(pumpPin, LOW);

digitalWrite(buzzerPin, LOW);

Serial.begin(9600);

}

void loop() {

unsigned long currentTime = millis();

if (currentTime - lastCheckTime >= checkInterval) {

lastCheckTime = currentTime;

int moisture = analogRead(sensorPin);

Serial.print("Moisture Level: ");

Serial.println(moisture);

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Moisture: ");

lcd.print(moisture);

if (moisture < threshold) {

lcd.setCursor(0, 1);

lcd.print("Soil Dry - Watering");

digitalWrite(pumpPin, HIGH); // Turn on pump (LED)

digitalWrite(buzzerPin, HIGH); // Buzz

delay(pumpDuration); // Wait

digitalWrite(pumpPin, LOW); // Turn off

digitalWrite(buzzerPin, LOW); // Stop buzz

} else {

lcd.setCursor(0, 1);

lcd.print("Soil OK - No Water");

digitalWrite(pumpPin, LOW);

digitalWrite(buzzerPin, LOW);

}

}

}

# Working Explanation

1. The soil moisture sensor continuously measures soil moisture.  
2. If the moisture is below a set threshold, the Arduino activates the LED (simulating a water pump) and the buzzer to indicate dry soil.  
3. The LCD displays the soil moisture level and system status in real time.  
4. Once the moisture level is sufficient, the system turns OFF the pump (LED) and buzzer automatically.  
5. This cycle repeats, ensuring smart irrigation with minimal water wastage.

# Applications

- Can be used in agricultural fields for automated irrigation.  
- Useful in gardens and greenhouses.  
- Helps conserve water and maintain proper plant growth.

# Conclusion

This smart irrigation system demonstrates the use of embedded systems for real-life applications. It ensures efficient water usage, reduces human effort, and improves crop management through automation.