### **Code**

### #include <Servo.h>

#include <LowPower.h>

const int soilMoisturePin = A0;

const int potentiometerPin = A1;

const int pumpPin = 9;

const int buzzerPin = 8;

const int ledPin = 7;

const int buttonPin = 2;

Servo servoMotor;

void setup() {

pinMode(soilMoisturePin, INPUT);

pinMode(potentiometerPin, INPUT);

pinMode(pumpPin, OUTPUT);

pinMode(buzzerPin, OUTPUT);

pinMode(ledPin, OUTPUT);

pinMode(buttonPin, INPUT\_PULLUP);

servoMotor.attach(10);

// Attach interrupt for wake-up button

attachInterrupt(digitalPinToInterrupt(buttonPin), wakeUp, FALLING);

Serial.begin(9600);

}

void loop() {

enterSleepMode();

int soilMoisture = analogRead(soilMoisturePin);

int threshold = map(analogRead(potentiometerPin), 0, 1023, 0, 1023);

Serial.print("Soil Moisture: ");

Serial.print(soilMoisture);

Serial.print(" | Threshold: ");

Serial.println(threshold);

if (soilMoisture < threshold) {

startWatering();

} else {

delay(1000); // Check soil moisture every second

}

}

void startWatering() {

digitalWrite(pumpPin, HIGH);

servoMotor.write(90);

feedback(true);

delay(5000); // Watering duration

digitalWrite(pumpPin, LOW);

servoMotor.write(0);

feedback(false);

}

void feedback(bool watering) {

if (watering) {

for (int i = 0; i < 2; i++) {

digitalWrite(ledPin, HIGH);

tone(buzzerPin, 1000);

delay(200);

digitalWrite(ledPin, LOW);

noTone(buzzerPin);

delay(200);

}

} else {

digitalWrite(ledPin, HIGH);

tone(buzzerPin, 500);

delay(200);

digitalWrite(ledPin, LOW);

noTone(buzzerPin);

}

}

void enterSleepMode() {

// Enter sleep mode

Serial.println("Entering sleep mode...");

LowPower.powerDown(SLEEP\_FOREVER, ADC\_OFF, BOD\_OFF);

}

void wakeUp() {

// This will be called when the button is pressed

Serial.println("Waking up...");

}

### Explanation of Changes:

1. Custom Threshold using Potentiometer:
   * Added a potentiometer connected to A1.
   * The threshold variable is set based on the potentiometer value using the map function.
2. Feedback Mechanism:
   * Added a buzzer (buzzerPin) and an LED (ledPin).
   * The feedback function provides feedback using the buzzer and LED when watering starts and stops.
3. Power Efficiency:
   * Integrated LowPower library to manage sleep mode.
   * Added a button connected to digital pin 2 to wake up the system from sleep mode using an interrupt.
4. Serial Communication:
   * Added Serial.begin for debugging purposes, printing the soil moisture and threshold values.