ZACHERY TAYLOR
EML4804L
08/06/2025

Final Project Report

Precision Horticulture

An Automated Planter System Utilizing Soil Moisture Sensors







ZACHERY TAYLOR
EML4804L
08/06/2025

Project Summary:

Precision Horticulture: An Automated Planter System Utilizing Soil Moisture Sensors

In this project, I explored the design and implementation of an automated planter system that utilized soil moisture sensors to facilitate “precision horticulture” at an attainable price range. My objective was to construct a device capable of independently monitoring two regions and regulating soil moisture, complemented by a water refill station to allow for manual water dispensing activated by an infrared sensor.

I built the system on an Arduino microcontroller, which managed two LM393 soil moisture sensors connected to analog pins (A0, A1). These sensors enabled real-time soil diagnostics and prompted irrigation cycles whenever moisture levels dropped below researched soil thresholds (400 for saturated soil, 550 as the trigger for watering). Each plant zone used a dedicated 5V water pump, controlled by relay modules with digital outputs, to deliver precise water doses and minimize the risk of over- or under-watering.

To add flexibility, I included a proximity-activated manual refill component employing an HC-SR04 ultrasonic sensor to detect hand proximity within 5 centimeters. When activated, the side hose dispenses water after a short delay and maintains flow as long as presence was detected, allowing for supplemental watering as needed.

For control and safety, I included a bundle of (3) SPST rocker switches (wired using INPUT_PULLUP) to independently enable or disable circuits for each soil zone and the manual dispensing station. This approach ensured that soil sensors and pumps were disabled during maintenance or when sensors were removed, effectively preventing unwanted watering that could occur if removed or moving probes registered as “dry.”

In the future I plan to include power management using a low-power operational mode via the LowPower library. When I engage sleep mode through its dedicated switch, the system will enter deep-sleep states for 60-minute intervals between sensor readings, significantly extending battery-powered operating life.

Throughout development, I faced technical challenges in calibrating soil moisture thresholds for consistent plant health and in managing sensor removal scenarios with the safety switch logic. The resulting automated planter system demonstrated reliable, energy-efficient, and modular precision irrigation capabilities. I found that this setup met my goals for smart horticulture while providing a scalable foundation for future enhancements in environmental monitoring and control.

See code below.

ZACHERY TAYLOR
EML4804L
08/06/2025

Arduino Code:

```
int soilPin1   = A0;
int relayPin1  = 6;
int MOIST_DRY1 = 600;
int switchPin1 = 2; // Rocker switch for Pump 1

int soilPin2   = A1;
int relayPin2  = 7;
int MOIST_DRY2 = 600;
int switchPin2 = 3; // Rocker switch for Pump 2

int pumpTime   = 8000;

int irPin      = 9; // IR sensor OUT for Pump 3
int relayPin3  = 8; // Relay for Pump 3
int switchPin3 = 4; // Rocker switch for Pump 3

void setup() {
  pinMode(relayPin1, OUTPUT); digitalWrite(relayPin1, HIGH);
  pinMode(relayPin2, OUTPUT); digitalWrite(relayPin2, HIGH);
  pinMode(relayPin3, OUTPUT); digitalWrite(relayPin3, HIGH);

  pinMode(switchPin1, INPUT_PULLUP);
  pinMode(switchPin2, INPUT_PULLUP);
  pinMode(switchPin3, INPUT_PULLUP);
  pinMode(irPin, INPUT);

  Serial.begin(9600);
}

void loop() {
  // Pump 1 (Soil)
  int reading1 = analogRead(soilPin1);
  if (digitalRead(switchPin1) == LOW) { // Switch ON (closed)
    Serial.print("Pump 1 Switch: ON | ");
    Serial.print("Soil sensor 1 reading: ");
    Serial.println(reading1);
    digitalWrite(relayPin1, HIGH);
    if (reading1 >= MOIST_DRY1) {
      Serial.println("Soil 1 is DRY — Pump 1 ON!");
      digitalWrite(relayPin1, LOW);
      delay(pumpTime);
      digitalWrite(relayPin1, HIGH);
```

```
    Serial.println("Pump 1 OFF\n");
  } else {
    Serial.println("Soil 1 moist — Pump 1 OFF\n");
  }
} else {
  digitalWrite(relayPin1, HIGH); // Always ensure OFF
  Serial.println("Pump 1 Switch: OFF | Pump 1 OFF\n");
}

// Pump 2 (Soil)
int reading2 = analogRead(soilPin2);
if (digitalRead(switchPin2) == LOW) { // Switch ON (closed)
  Serial.print("Pump 2 Switch: ON | ");
  Serial.print("Soil sensor 2 reading: ");
  Serial.println(reading2);
  digitalWrite(relayPin2, HIGH);
  if (reading2 >= MOIST_DRY2) {
    Serial.println("Soil 2 is DRY — Pump 2 ON!");
    digitalWrite(relayPin2, LOW);
    delay(pumpTime);
    digitalWrite(relayPin2, HIGH);
    Serial.println("Pump 2 OFF\n");
  } else {
    Serial.println("Soil 2 moist — Pump 2 OFF\n");
  }
} else {
  digitalWrite(relayPin2, HIGH);
  Serial.println("Pump 2 Switch: OFF | Pump 2 OFF\n");
}

// Pump 3 (IR/manual)
int irValue = digitalRead(irPin); // LOW = hand/object detected
if (digitalRead(switchPin3) == LOW) { // Switch ON
  Serial.print("Pump 3 Switch: ON | IR sensor: ");
  Serial.println(irValue == HIGH ? "Object detected" : "No object");
  if (irValue == HIGH) {
    Serial.println("Hand/object detected by IR — Pump 3 ON!");
    digitalWrite(relayPin3, LOW);
    delay(pumpTime);
    digitalWrite(relayPin3, HIGH);
    Serial.println("Pump 3 OFF\n");
  } else {
    digitalWrite(relayPin3, HIGH);
    Serial.println("No object detected — Pump 3 OFF\n");
```

```
    }
  } else {
    digitalWrite(relayPin3, HIGH);
    Serial.println("Pump 3 Switch: OFF | Pump 3 OFF\n");
  }

  Serial.println("----------------");
  delay(2000);
}
```

ZACHERY TAYLOR
EML4804L
08/06/2025

Circuit Diagram:

