ECE 203: Electronic Circuit Design, Analysis and Implementation
Final Project Lab Report

Gloria Liu and Ella Simmons

Table of Contents

1. Project & Team Information	**2**
2. Statement of Objective	**2**
3. System Overview	**2-3**
4. Circuit Documentation	**3**
5. Performance Analysis	**4**
6. Design Challenges and Solutions	**4**
7. Future Improvements	**4**
8. Appendices	**5-6**

1. Project & Team Information

• Project title: Plant-Sitter: A Soil Moisture and Light Level Monitoring System

• Full names of all team members: Gloria Liu, Ella Simmons

• University email addresses: gl1480@princeton.edu, es5889@princeton.edu

• Submission date: 5/2/2025

2. Statement of Objective

The primary objective of this project is to design and build a "Plant-Sitter" device that monitors the soil moisture and light levels of a plant to ensure optimal growing conditions. By integrating moisture and light sensors with a microcontroller, the device aims to automate the monitoring process and provide feedback to users. This project holds significant practical relevance for plant enthusiasts, as it reduces the need for constant manual monitoring. Specific objectives included measuring soil hydration levels, assessing light intensity, processing sensor data through a microcontroller, and indicating results through LED lights in real-time.

3. System Overview

The Plant-Sitter system consists of three main subsystems: sensors (water and light sensors), a microcontroller, and an output mechanism. The water sensor (from Keyestudio) measures hydration levels in the soil by detecting changes in resistance, which is then converted into a voltage signal and sent through one of the analog input pins of the Arduino. This water sensor model (shown in Figure 1) was chosen because its shape enables easy measurement of soil moisture. Similarly, the light sensor measures ambient light intensity and transmits the data through one of the analog input pins to the microcontroller for analysis. The Arduino Uno microcontroller was selected due to its compatibility, ease of use, and sufficient processing capability. It processes sensor inputs, applies threshold comparisons, and triggers the output mechanisms. The output mechanism includes LEDs (connected to the Arduino through GPIO pins) to visually indicate soil moisture and light levels. The color of the LEDs indicate soil moisture: white LED indicates that the soil is wet, yellow LED indicates that the soil is moist, and red LED indicates that the soil is dry and that the plant needs to be watered. Light level is indicated by the brightness of the LEDs: if there is sufficient light in the environment, the LED

would be dim, and if light in the environment is insufficient, the LED would shine bright, supplying extra light to the plant.

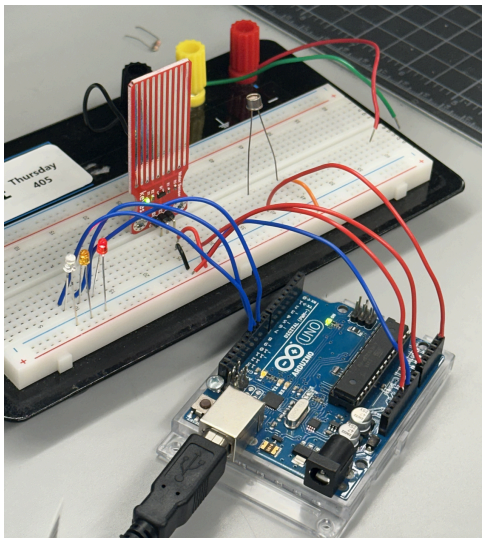


Figure 1: Picture of system setup.

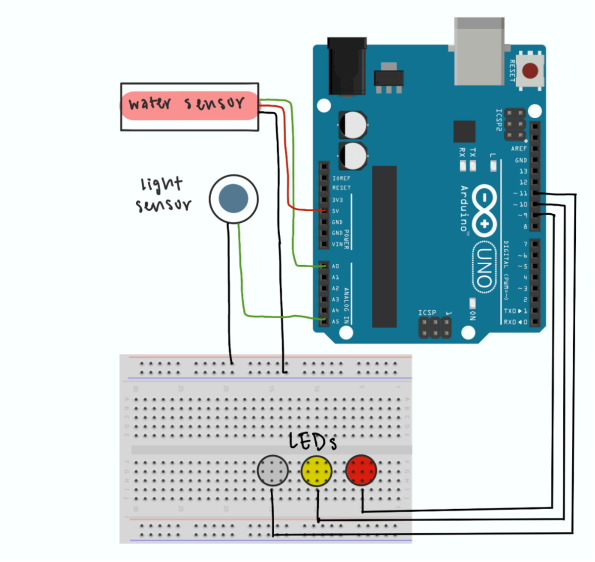


Figure 2: Block diagram of system setup.

4. Circuit Documentation

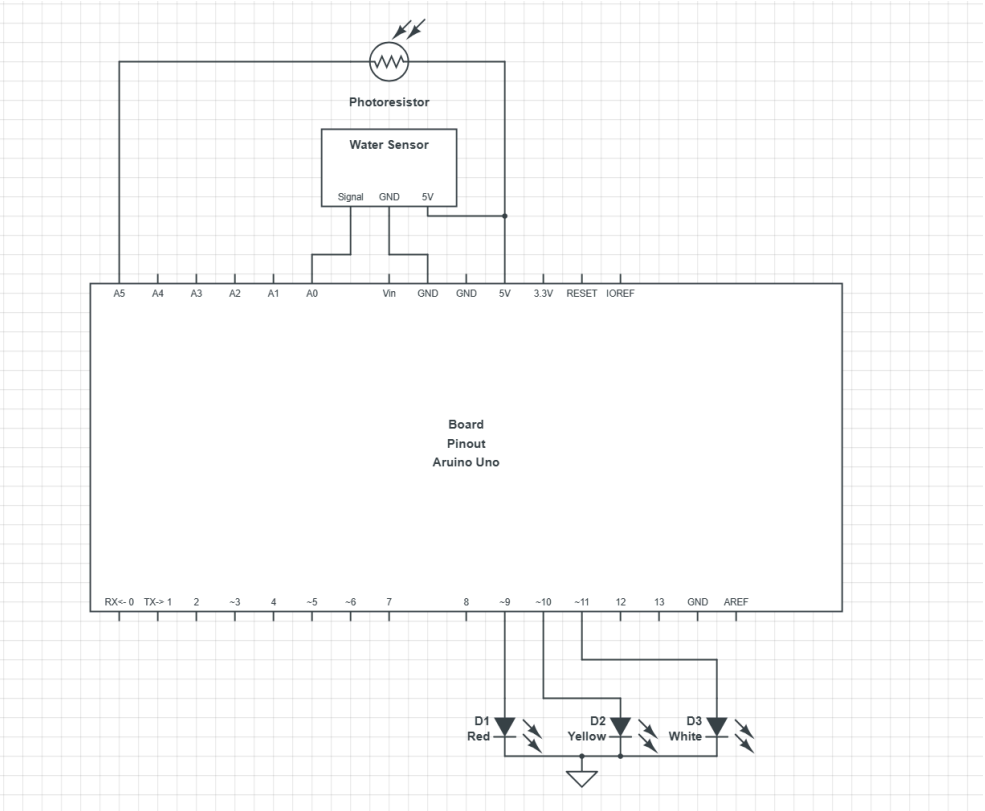


Figure 3: Circuit Diagram (Circuit Lab)

The Arduino block was borrowed from a previously created custom circuit on circuit lab, found here: https://www.circuitlab.com/circuit/6t78uk/arduino-uno/.

5. Performance Analysis

Link to demo video:
https://drive.google.com/file/d/1y-wGqakuBoJlme7EJRtSBJSwSeefAlfx/view?usp=sharing

6. Design Challenges and Solutions

The project faced challenges such as difficulties in determining optimal threshold values for soil moisture and light intensity. To address these issues, different moisture levels and light intensities were tried on the sensors to fine-tune thresholds. Additionally, the moisture sensor was initially not mobile enough to test the soil. The large breadboard was replaced with a smaller breadboard to allow for more physical flexibility in measurement.

7. Future Improvements [5 points]

While the current design meets its intended objectives, there are opportunities for enhancement. The current design combines the indicators (LEDs) for moisture and light levels. Separating them——a set of three colored LEDs to indicate moisture level and a separate LED to indicate light level and supply light to make up for insufficient light in the environment——may allow for more flexibility and better functionality in the aspect of light supply. These improvements may be the most cost effective and feasible in the short term.

Future iterations could also incorporate additional sensors, such as temperature and humidity sensors, to provide a more comprehensive environmental analysis. Solar power could be integrated to improve energy efficiency, particularly for outdoor use. Predictive analytics could also be added to suggest optimal care routines. These improvements have yet to be evaluated for feasibility and cost-effectiveness to ensure practical implementation.

8. Appendices

• Complete code listings with comments

```
int waterPin = 0; //connect water sensor to analog interface 0
int ledWhite = 11; //LED to digital interface 11
int ledYellow = 10; //LED to digital interface 10
int ledRed = 9; //LED to digital interface 9
int waterVal = 0; //define the initial value of variable waterVal as 0
int lightPin = 5; //connect water sensor to analog interface 5
int lightVal = 0; //define the initial value of variable lightVal as 0
int brightness = 0; //define the initial value of variable brightness as 0

void setup()
{
 pinMode(ledWhite, OUTPUT); //define white led as output pin
 pinMode(ledYellow, OUTPUT); //define yellow led as output pin
 pinMode(ledRed, OUTPUT); //define red led as output pin
 Serial.begin(9600); //set baud rate at 9600
}

void loop()
{
 waterVal = analogRead(waterPin); //read and assign analog value to variable waterVal
 lightVal = analogRead(lightPin); //read and assign analog value to variable lightVal

 if (lightVal > 1050) { //decide whether variable lightVal is over 1050
   brightness = 25; //if so, turn on low brightness
 }
 else if (lightVal > 800) { //decide whether variable lightVal is over 800 but lower
than 1050
   brightness = 75; //if so a turn on brightness of 75
 }
 else if (lightVal > 600) { //decide whether variable lightVal is over 600 but lower
than 800
   brightness = 150; //if so a turn on brightness of 150
 }
 else { //decide whether variable lightVal is lower than 600
   brightness = 200; //if so a turn on brightness of 200
 }

 if(waterVal>500){ //decide whether variable waterVal is over 500
```

```
   analogWrite(ledWhite,brightness); //turn on white LED when variable waterVal is
over 500
   analogWrite(ledYellow,0);
   analogWrite(ledRed,0);
}
 else if(waterVal>300){ //decide whether variable waterVal is over 300
   analogWrite(ledWhite,0);
   analogWrite(ledYellow,brightness); //turn on yellow LED when variable waterVal is
over 300
   analogWrite(ledRed,0); //turn on  LED when variable 'val' is over 700
 }
  else{ //decide whether variable waterVal is lower than 300
   analogWrite(ledWhite,0);
   analogWrite(ledYellow,0);
   analogWrite(ledRed,brightness); //turn on red LED when variable waterVal is lower
than 300
 }
 Serial.print("Light level:");
 Serial.println(lightVal); //print variable lightVal by Serial.print
 Serial.print("Water level:");
 Serial.println(waterVal); //print variable waterVal by Serial.print
 delay(100);
}
```