ECET 430 - Final Report

Natwarin Padtha, Jonathan Jou

2025-05-14

1. ABSTRACT

The Smart Plant Care project is an embedded system project with the goal of helping plant owners maintain conditions for plant growth with the added benefit of continuous environmental data collection. The device uses sensors to collect real-time environmental data to measure soil moisture, light, temperature, humidity, and alert users via visual indicators (RGB LED), and textual displays (OLED display). The power supply consists of a rechargeable 3.7V Li-ion battery, which is regulated to produce a 5V supply to the Arduino Uno microcontroller and sensors. The development of Smart Plant Care focused on user simplicity, reliability, and expandability. Our group developed and manufactured a custom PCB using KiCad software, developed performance testing of the devices to ensure all systems identified functions, and used a logic-based alert system using Arduino IDE to guide the users to appropriate corrective action. The Smart Plant Care features a real-time display, color condition alerts, and an additional optional path to allow future mobile app integration/interfacing with mobile devices. Overall, the cost-effective, beginner-friendly Smart Plant Care device is ideal for plant monitoring and creates an environment for users who grow plants with visually attractive devices. The final product successfully finds the balance between technical depth and usability suggestions for casual users and matches common technical breadth and usability for home gardeners, students, busy people who want to have plants, and tech lovers alike.

2. INTRODUCTION

The Smart Plant Care device allows users to continuously monitor and communicate real-time plant health parameters based on important environmental factors, namely soil moisture, temperature, humidity, and light intensity. Indoor plants often suffer from inconsistent care or a lack of feedback about what they need. By simple sensors and a user feedback mechanism, we hope to provide a solution to the challenges indoor plants endure so users can react when needed, like watering or moving their plant.

The system is powered via an Arduino Uno microcontroller, where the sensors read data and update the user on the condition of the plant via an OLED screen. The Smart Plant Care device was designed for usability, cost-effectiveness, while not sacrificing modularity. Thus, it can be used by novice gardeners, tech enthusiasts, and time-crunched plant owners alike. The device is encased in a small, water-resistant enclosure with a rechargeable battery so it can be easily moved and used in a practical context. In the end, this project powers embedded electronics and sustainable living by creating a smart product that we hope people find convenient. This project could also be useful for individuals who don't have enough time to look after the plant but like the idea of having a plant. It would also be ideal for students themselves as this type of device could help them keep track, monitor, and study their plant.

3. OBJECTIVE

This project, Smart Plant Care, is the class project for ECET 430: Electronics Design for Manufacturing and Production in Spring 2025. The purpose of the Smart Plant Care project is to create a low-cost, reliable, and easy-to-use system that helps plant owners keep their plants healthy by constantly monitoring their environment. The project consists of building a device that utilizes the sensors to monitor soil moisture, temperature, humidity, and light intensity, and makes it possible for users to make informed decisions, at the right time, to encourage healthy plant growth. The system will be designed with simplicity in mind so that users will only have to insert it in a plant pot, activate it, and the system will collect data and provide feedback.

4. PROJECT DEFINITION

The original goal of the Smart Plant Care project was simply to provide people with a reliable and simple device upon which they could monitor the relevant environmental conditions that affect plant health. The device is intended to monitor the following environmental conditions: soil moisture, light intensity, temperature, and humidity, and allow individuals to be informed with a response to the data gathered by the device in a visual, auditory, and text format. The intention of this device was aimed at simplifying plant care and monitoring, especially for a beginner, busy person, or someone who often forgets to care for their indoor plants.

User Interaction Stories

The Smart Plant Care device is designed for easy use. It is easy to unpack because all the parts are well-marked. To set it up, the user only needs to put the device into the pot and switch it on. The Advanced Model can be connected with a phone using Bluetooth or Wi-Fi. During its work, it constantly monitors the plant, presenting real-time data on the OLED display. It will even light up or send a notification to the user when it needs attention.

User Interface

• Displays
  o OLED Display: An OLED display allows the device to present real-time data on soil moisture, light intensity, temperature, and humidity for clear visual feedback.
• Buttons
  o Power Button: It has a straightforward power button that could turn on or off.
  o Reset Button: This is used to reset the device when required; for instance, this may be used after an operating system update or troubleshooting of the device.
• Indicators
  o RGB LED: The device features a multi-color LED for visual indication of alerts. Green means good condition, and red for a critical issue like dry soil or too hot/cold.
  o Buzzer: This includes an onboard buzzer that allows for signals in case the plant needs immediate care, for instance, if it gets too dry or too hot/cold.
  o Charging Indicator: This is a small LED indicator showing the status of charging of the rechargeable battery-for instance, blinking red for charging and green when it is full.

User Acceptance

1. **Given** the device is on and in a plant pot,
When the soil is dry,
Then the LED turns red, and a buzzer alerts the user to check the display for the plant status.
2. **Given** the device is connected to a phone,
When the light is too bright for the plant,
Then the user gets a notification on their phone.
3. **Given** the device is in the plant pot,
When the temperature and humidity are ideal,
Then the display shows "Optimal" and the light turns green.
4. **Given** the battery is low,
When the device is charging,
Then the charging light will blink red and green when full.

Parameters

• Dimensions: Width 4 inches, Height 1.5 inches
• Weight: 200 grams
• Electromagnetic compatibility (EMC) and electromagnetic interference (EMI): The Smart Plant Care will be designed to meet the necessary requirements of Electromagnetic Compatibility to avoid interfering with other gadgets. It shall be shielded appropriately to limit unwanted emissions, hence not interfering with the operation of other gadgets around it. The device shall also be designed to minimize EMI, which creates minimal possible interference that might affect other devices, including a Wi-Fi router, smartphone, or any other household electronic device. It ensures that the Smart Plant Care device works perfectly at home or in the office, without interfering with other technologies that may be nearby.
• Protections:
o Water Resistance: safe from splashes.
o Shock Protection: Durable, shock-resistant casing.
o Overcharge Protection: Prevents battery damage during charging.

Functions

• Soil Moisture: Monitors and alerts in case the plant needs water.
• Light Level: Monitors the light level for perfect growth.
• Temperature and Humidity: Measures and keeps conditions optimal.
• Alerts: Sends sound and phone notifications to the user for needs of care needs.
• Rechargeable Battery: Long-lasting with overcharge protection.
• Data Display: Real-time plant data on an OLED display for easy monitoring.
• App Feedback: The device sends updates and alerts right to your phone via the app.
• Easy Setup: Very easy to set up and use.

Integration

• Interfaces:
1. OLED Display: displayed in real-time: Soil moisture, Light levels, Temperature, and Humidity
2. RGB LED: Visual feedback given- Green for optimum, and Red when urgent care is required.
3. Buzzer: A buzzer makes a sound as an alert regarding a critical plant condition.
4. Mobile Application Interface FOR THE Advanced model: Real-time data, notification display, and adjusting settings using smartphones can be done easily.

• Protocols:
1. Wi-Fi for Advanced Model: Wi-Fi connectivity could send data to applications or alert at the same time to sync to the cloud.

Operational

• Restriction

o Temperature Limits: Although this device can monitor temperature in real time, it's best advice not to operate in extreme heat or cold (0°C to 40°C).

o Power Supply: The rechargeable battery requires regular recharging, and cannot be replaced by the user.

o Water Resistance: It's water resistant, but it's best not to be submerged in water.

Environment

• Temperatures

o Operating Temperature: The Smart Plant Care device functions well within 0°C to 40°C; anything surpass 40°C or below 0°C might damage the device.

• Hazards

o Electrical Hazard: Avoid charging the device in wet conditions to prevent electrical short circuits.

o Battery Hazard: Do not try to replace the battery on your own, to prevent leakage or fire risks.

• Power

o Power Source: The device uses a rechargeable battery.

o Battery Life: The battery lasts up to 10 days on a full charge, depending on usage.

Key Concerns

• Most Important Parameters

o Battery Life: Lasts up to 10 days after a full charge.

o Operating Temperature: Works between 0°C and to 40°C.

o Water Resistance: Splash proof but advise not to submerge with water.

o User Alerts: Sends alerts through LED, buzzer, and phone notifications (only in advanced model).

• Set-in-Stone Parameters

o Rechargeable Battery: Charging using USB.

o Dimensions: About 4 inches in width and 1.5 inches in height.

o Connectivity: Bluetooth and Wi-Fi for app use (only in advanced model).

Future

• Plans:

o Software Updates: Create different needs for different plants.

o Better Features: Add sensors for soil pH and nutrients.

o Cost: Bring down the cost.

• Ideas

o AI Support: Use AI to research the plant, and using the information, come up with the best way to take care of it.

o Community Sharing: Build an online platform for users to share tips and advice.

As the project progressed, the fundamental definition of the project shifted as more solid design practices were added, implemented, and tested. While the initial concept showed the bare minimum functionality, the description of the prototype is now comprehensive enough to contain a custom PCB in KiCad, a sealed housing that protects the electronics from moisture, and improved sensor calibration so that readings can be obtained consistently. Additionally, we decided not to use a buzzer at the end of the project due to the shortage of components. An Arduino Uno was chosen as the microcontroller as it provided the easiest interface with multiple sensors and peripherals. User feedback is provided by an OLED display, an RGB LED, and a buzzer that allows alerts and notifications to be received without necessarily using a mobile app.

Although the more advanced version that could provide both Bluetooth and WiFi capabilities was proposed for future enhancements, the current redesign incorporates the baseline project elements of local monitoring, notifications, and low-power operation from a 3.7V rechargeable battery. The project has addressed issues such as water resistance, electromagnetic immunity shielding, and basic protections such as overcharge protection. The modifications made throughout the semester were translated into this updated revision, demonstrating that the project affords the opportunity to correctly place sensors, manage power, and focus on a good user experience.

In the revision, Smart Plant Care is now a completed integrated device that aligns with the original deliverable, with options for future development, including mobile tracking, data logging, or machine learning/AI-based suggestions for plant care.

5. DESIGN DOCUMENT

The development of this project focused on designing a device that was functional, compact, and reliable in measuring plant health in real time with an embedded design. This section outlines the key hardware and software decisions made throughout the project and the reasons for those decisions.

List major components:

Microcontroller: Arduino UNO

The Arduino Uno was chosen for the main controller because it is easy to program, there is a large community to build off of, and it has a bigger than required number of GPIO pins for the many sensors and output devices. Other more compact options were considered, for example, the Arduino Nano or ESP32. However, at the time of development, the Arduino Uno was the easiest option to use for prototyping and debugging.

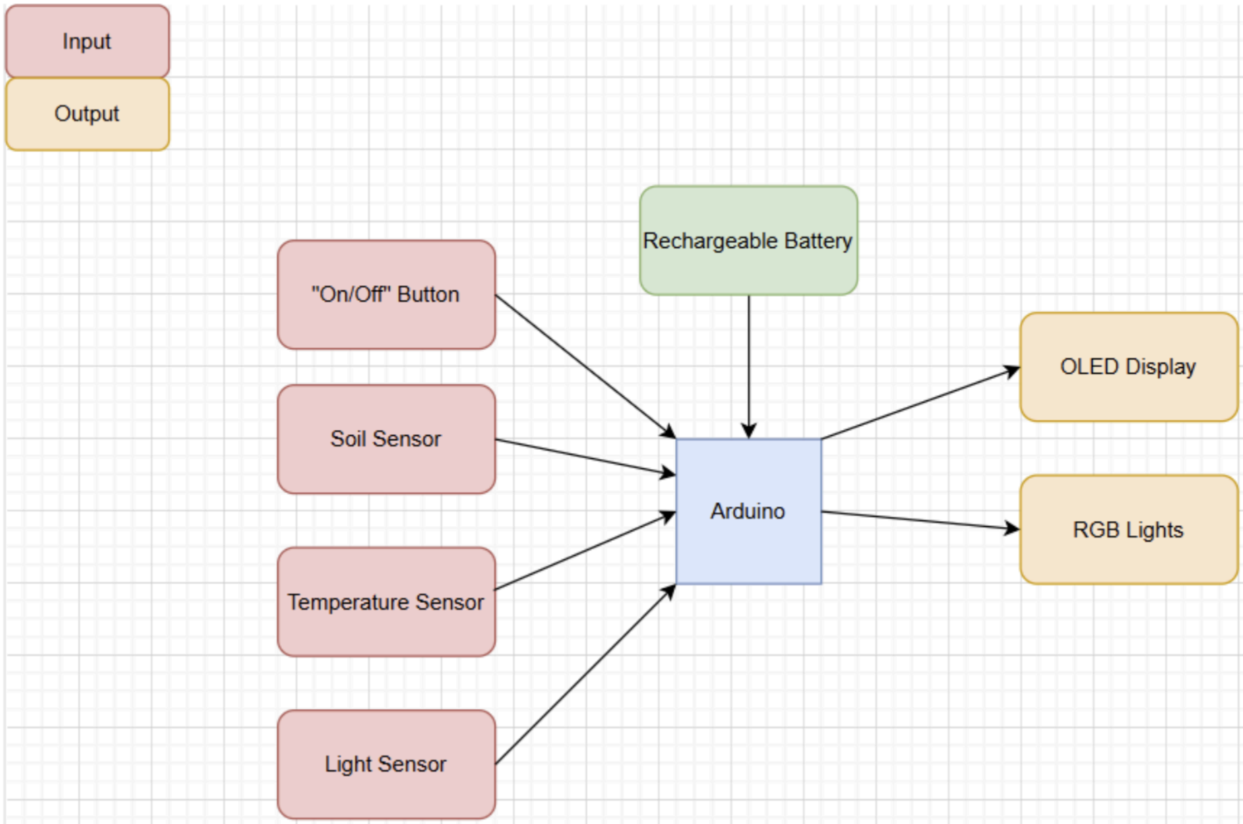
Sensors: DHT11 Temperature and Humidity Module, Digital LDR Photosensitive Light Sensor Module, LM393 3.3V-5V Soil Moisture Detect Sensor, Soil Moisture

Soil Moisture Sensor (LM393): Chosen for its analog and digital output, allowing flexibility in threshold-based water level detection.

Temperature and Humidity Sensor (DHT11): Although less accurate than the DHT22, the DHT11 was selected for its low cost and sufficient precision for this application.

Light Sensor (APDS-9306): This digital LDR-based sensor provides reliable ambient light readings, essential for determining if a plant receives enough sunlight.

Block Diagram:



Priority items:
- Accuracy: It ensures reliable plant monitoring
- Battery Life & Power Management: Keeps the device running for extended periods
- Electrical Hazard: Avoid any safety risks
- Water Resistance: Prevent environmental exposure

Dependent tasks:
What will require additional research? What needs to be completed before something else can be started? What feels unclear or not obvious?
- The appropriate soil moisture, humidity, and light intensity ranges for different types of plants
- At what moisture level should the device trigger a watering alert?
- How do environmental conditions affect sensor accuracy?
- The best microprocessor that balances power efficiency, processing capability, and connectivity
- How will the design prevent electrical hazards, such as short circuits, from moisture exposure?

Architecture:
Options to implement the project:
I chose a Li-ion 14500 battery or a Lithium Polymer (LiPo) 3.7V battery because they are both 3.7V and can output 5V with a voltage regulator, which is what an Arduino, sensor, and display need. Secondly, they are small enough to fit into the smart plant device.

Three possible prototypes:
1. Temperature Sensor
2. Soil Moisture Sensor
3. Light Sensor

Output Components:
1. A 0.96-inch OLED Display to display the temperature, humidity, soil moisture, and light status in real time.
2. an RGB LED to provide quick feedback - green for good, red for critical

3. a buzzer to give audio alerts when immediate action is required (like very dry soil when a spray is needed)

Not all features could be tested at the same time due to the number of I/O pins available on the Arduino Uno. Future designs may use a microcontroller with more I/O pins or add multiplexers to expand options for other features.

Enclosure Design

The physical design was 3D-printed using PLA and is approximately 6" x 6" x 6" in size. The electronics are housed in a water-resistant inner compartment to protect them from outside influences, but still allow air and light into the sensors.

PCB Design in KiCad

A custom PCB was created using KiCad, replacing the generic Arduino Uno footprint with the ATmega328P microcontroller and supporting circuitry to optimize space. The PCB was designed with trace clearance, voltage rails, and short-circuit protection in mind, and DRC/ERC checks confirmed design reliability.

Software Design

The Arduino code was written using the Arduino IDE. It includes modular functions for reading each sensor, evaluating thresholds, and triggering corresponding outputs. The system checks conditions every few seconds and responds accordingly, balancing accuracy and power consumption.

6. KiCAD PROJECT

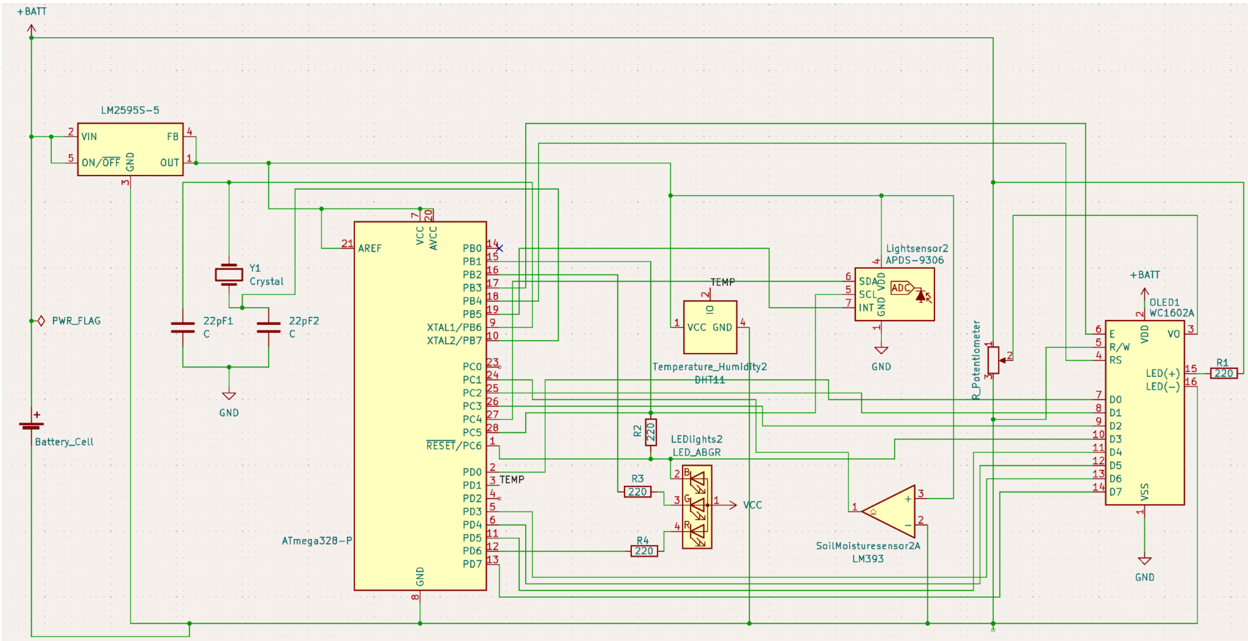
The printed circuit board (PCB) for the Smart Plant Care device was designed in KiCad, a free and open-source electronics design automation software. The PCB design was intended to consolidate all of the necessary components into an efficient, manufacturable board with the microcontroller, sensor inputs, display connectors, voltage regulator, and power supply terminal in a combined area for the enclosure.

This design incorporated the ATmega328P microcontroller, the same chip as seen in the Arduino Uno, which is only slightly smaller than the full Arduino Uno module. Utilization of the microcontroller chip provides a decrease in form factor and thus improves overall layout efficiency per unit area. The supporting components for the microcontroller, including the crystal oscillator, pull-up resistors, and decoupling capacitors, were added to allow for stable operation.

The three sensors were connected to the PCB via analog or digital connections. The connections were rerouted through headers that were individually labeled for external connections. The power lines were separated and included a regulated 5V voltage output using a 5V linear voltage regulator, powered by a 3.7V Li-ion battery, which is a rechargeable battery and is commonly used in consumer electronics.

*KiCAD zip file on CANVAS

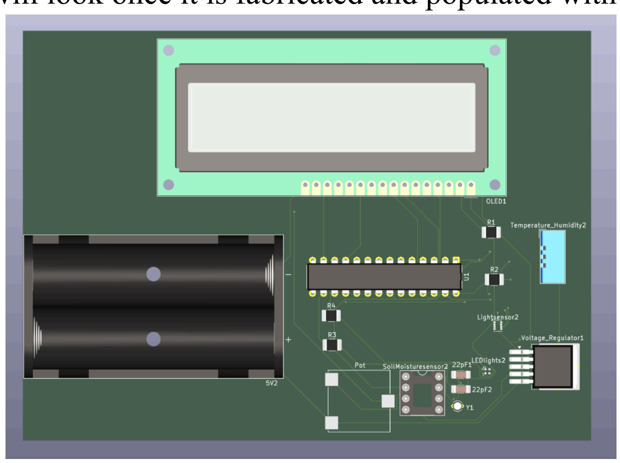
7. SCHEMATIC



*KiCAD PDF Plot on CANVAS

8. 3D BOARD RENDERING

To help picture the physical layout of the Smart Plant Care device's custom PCB, a 3D board rendering was produced using KiCad's 3D Viewer tool. The rendering provides a realistic depiction of how the board will look once it is fabricated and populated with components.



9. FABRICATION FILES

A set of fabrication files was created from the finalized KiCad PCB design to prepare the Smart Plant Care device for manufacturing. These fabrication files contain everything a PCB fabrication service needs to create the physical board.

10. USER DOCUMENTATION

Purpose of the Product

The Smart Plant Care device is a small, independent system that assists users in monitoring the environmental conditions for indoor plants. By measuring soil moisture, temperature, humidity, and light intensity in real time, the system provides actionable feedback to improve plant care efficacy and decrease guesswork. The device is particularly beneficial for beginners, busy individuals, or those aiming to enhance the health and lifespan of their indoor plants.

How to Operate the Device

- Insert the Device into the Plant Pot: Gently place the device into the soil so the moisture sensor is embedded below the surface.
- Power On: Use the built-in power button to activate the device. It will automatically begin monitoring and displaying sensor data.
- Monitoring: The OLED screen shows live readings of soil moisture, temperature (in °C and °F), humidity, and light conditions.
- Alerts: If the sensor detects issues soil dry, not enough light, too cold, et,c the LED will light up.
- Charging: When the battery is low, connect the USB charging cable. A small LED will blink red while charging and turn green once fully charged.

Wiring

Component	Connection Type	Arduino Pin	Notes
Soil Moisture Sensor	Analog Input	A0	Insert sensor into soil; keep cable dry
DHT11 Temp/Humidity	Digital Input	Pin 8	Place the sensor near the base of the plant
Light Sensor	Digital Input	Pin 13	Should face the natural light direction
RGB LED	Digital Output	Pins 6, 9, 10	Controls color-coded alerts
OLED Display	I2C Interface	SDA/SCL (A4/A5)	Displays live sensor data
Battery	Power Input	JST Connector	Powers the system via a 3.7V battery
Voltage Regulator	Power Management	In line with VCC	Converts a 3.7V battery to a stable 5V

User Interface Elements

OLED Display: Shows temperature (°C/°F), humidity (%), soil status (Dry/Good/Wet), and light status.

RGB LED: Visual status indicator.

Power Button: Turns the device on or off.

Charging Indicator: LED near USB port shows battery charging status.

11. TESTING DOCUMENT

Test Case	Test Method	Expected Result	Actual Result	Pass/Fail
Temperature Sensor Accuracy *	Compare with a thermometer	The result matched the thermometer	The system displays real-time temperature and humidity levels, providing accurate environmental monitoring.	Passed
Soil Moisture Sensor Calibration*	Measure values in dry, moist, and wet soil	Readings match expected thresholds	The system detects soil values and outputs, and determines whether the conditions are classified as "wet" or "dry" based on the measured data.	Passed
Light Sensor Functionality*	Test in both dark and bright environments	Readings correspond to light or no light	The system outputs a "light" or "no light" result based on the detected environmental conditions.	Passed
OLED Display Output	Display text messages & check readability	Text and alerts are clearly visible	It did not show any text on the display.	Passed
Data Logging and Communication	Connect to the computer and check synchronization	Data updates accurately with minimal delay	It displays real-time data from each sensor on the computer.	Passed

12. SOFTWARE

The Smart Plant Care device's software was built with the Arduino IDE. The software's basic function is to continually read values from the soil moisture, temperature/humidity, and light sensors, process those values, and feed back to the user via the OLED display, RGB LED, and serial monitor.

Arduino Code

```
#include <dht.h>
#include <LiquidCrystal.h>

// Sensor and LED Pins
#define outPin 8        // DHT11 sensor
#define DO_PIN 13       // Light sensor (digital)
int sensorPin = A0;     // Moisture sensor (analog)

// RGB LED Pins
const int redPin = 6;
```

```
const int greenPin = 9;
const int bluePin = 10;

// LCD Pins: RS, E, D4, D5, D6, D7
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

// Temperature thresholds (in Fahrenheit)
const float minTempThresholdF = 70.0;
const float maxTempThresholdF = 80.0;

dht DHT;  // DHT11 sensor object

void setup() {
  Serial.begin(9600);
  lcd.begin(16, 2);
  lcd.print("System Starting...");
  delay(2000);
  lcd.clear();

  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);
  pinMode(DO_PIN, INPUT);
  pinMode(sensorPin, INPUT);
}

void loop() {
  int readData = DHT.read11(outPin);
  float temperatureF = (DHT.temperature * 9.0) / 5.0 + 32.0;
  float humidity = DHT.humidity;
  bool lightGood = checkLightSensor();
  int moistureStatus = checkMoistureSensor();

  bool temperatureGood = (temperatureF >= minTempThresholdF &&
temperatureF <= maxTempThresholdF);
  bool moistureGood = (moistureStatus == 0);
  bool allGood = temperatureGood && lightGood && moistureGood;

  // Display sensor values
  Serial.println("============================");
  Serial.print("Temp = ");
  Serial.print(DHT.temperature);
  Serial.print("C | ");
```

```
  Serial.print(temperatureF);
  Serial.println("F");
  Serial.print("Humidity = ");
  Serial.print(humidity);
  Serial.println("%");
  Serial.println(lightGood ? "Light Present" : "No Light Detected");
  Serial.println(moistureStatus == -1 ? "Soil is Dry" : (moistureStatus ==
1 ? "Soil is Too Wet" : "Soil Moisture is Good"));
  Serial.println("===========================");

  // LCD first row = Temp & Humidity
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("T:");
  lcd.print(temperatureF, 1);
  lcd.print("F H:");
  lcd.print(humidity, 0);
  lcd.print("%");

  if (allGood) {
    setGreen();
    lcd.setCursor(0, 1);
    lcd.print("All Good");
  } else {
    if (!temperatureGood) {
      flashRed();
    }
    if (!lightGood) {
      flashBlue();
    }
    if (!moistureGood && moistureStatus == -1) {
      flashPurple();
    }
  }

  delay(2000);
}

bool checkLightSensor() {
  int lightState = digitalRead(DO_PIN);
  return (lightState == LOW);  // LOW = light detected
}
```

```
int checkMoistureSensor() {
  int sensorValue = analogRead(sensorPin);
  sensorValue = map(sensorValue, 0, 1024, 255, 0);
  if (sensorValue < 150) return -1;  // Dry
  else if (sensorValue > 200) return 1;  // Too wet
  else return 0;  // Good
}

void flashRed() {
  lcd.setCursor(0, 1);
  lcd.print("Temp Low");
  for (int i = 0; i < 3; i++) {
    setRGB(HIGH, LOW, LOW);  // Red
    delay(300);
    setRGB(LOW, LOW, LOW);
    delay(300);
  }
}

void flashBlue() {
  lcd.setCursor(0, 1);
  lcd.print("No Light        ");  // Pad with spaces to clear previous msg
  for (int i = 0; i < 3; i++) {
    setRGB(LOW, LOW, HIGH);  // Blue
    delay(300);
    setRGB(LOW, LOW, LOW);
    delay(300);
  }
}


void flashPurple() {
  lcd.setCursor(0, 1);
  lcd.print("Soil Dry");
  for (int i = 0; i < 3; i++) {
    setRGB(HIGH, LOW, HIGH);  // Purple
    delay(300);
    setRGB(LOW, LOW, LOW);
    delay(300);
  }
}

void setGreen() {
```

```
  setRGB(LOW, HIGH, LOW);  // Green
  Serial.println("Everything is good! Green Light ON.");
}

void setRGB(int r, int g, int b) {
  digitalWrite(redPin, r);
  digitalWrite(greenPin, g);
  digitalWrite(bluePin, b);
}
```

13. LOGBOOK

Tasks Completed	Results / Notes
Brainstormed project ideas, selected "Smart Plant Care" as concept	Finalized project scope and key features (sensors, display, alerts)
Built a basic breadboard circuit with Arduino Uno and sensors	All sensors responded to input; OLED not yet integrated
Wrote the first version of Arduino code: read sensors and print to Serial Monitor	The output showed consistent soil moisture and temperature readings
Added RGB LED functionality and logic for color-coded feedback	RGB LED worked correctly; green showed optimal conditions, red for dry soil
Integrated OLED display and attempted I2C communication	Encountered inconsistent display issues; I2C address and wiring were double-checked
Started KiCad schematic and component footprint placement	Replaced Arduino Uno symbol with ATmega328P and supporting components
Finished KiCad PCB design and passed ERC/DRC checks	Gerber files generated; the board was ready for fabrication
3D modeling and printing of a vase enclosure	Fit confirmed; adjustments made to sensor slot positions
Performed individual testing of sensors in varied conditions	Moisture, temperature, and light detection passed test cases
Attempted buzzer integration and full prototype assembly	Could not test buzzer due to lack of available pins
Final code tuning, data logging via Serial Monitor, created repository	Sensor thresholds fine-tuned; GitHub repository updated
Presentation preparation, documentation draft, final testing	Final build working, but OLED display remained unreliable in some tests

Things That Did Not Work

- The OLED display had intermittent failures, possibly due to I2C address conflicts or unstable wiring.
- The buzzer could not be tested in the final prototype due to insufficient digital I/O pins.
- The display readability and update timing needed improvement for a better user experience.

Lessons Learned

- Start prototyping earlier to allow more time for debugging hardware interactions (e.g., OLED).
- Pin limitations on Arduino Uno can restrict functionality; future builds should consider microcontrollers with more GPIO pins or use I/O expanders.
- Proper sensor placement and wiring are crucial to ensure accurate readings and system stability.
- KiCad's rule checks are extremely helpful in preventing costly PCB errors and ensuring manufacturability.
- Power supply planning early in the design phase simplifies later integration and improves reliability.

14. REPOSITORY

The complete source files for the Smart Plant Care project are hosted in a public GitHub repository to ensure transparency, collaboration, and reproducibility. The repository contains all files necessary for others to replicate, build, and test the device from start to finish.
Repository Link: https://github.com/jonjonjou/Smart-Plant-Care

15. DISCUSSION

The Smart Plant Care project combined different parts of embedded systems, like connecting sensors, designing circuits, managing power, and building a user interface. One of the biggest successes was how well the sensors worked. The system could reliably measure soil moisture, temperature, humidity, and light, and show the data in real time. Using low-cost sensors turned out to be enough for what we needed, and it kept the device affordable. Also, using the ATmega328P chip directly on our custom PCB (instead of a full Arduino Uno) helped make the design smaller and more efficient.

We ran into some hardware issues since there aren't enough digital I/O pins on the Arduino Uno; we were unable to get accurate testing with the buzzer, as it would conflict with the other I/O pins. In future iterations, we can either implement a microcontroller with a higher number of pins or utilize an I/O expander to fix this issue.

On the software side, functional reasoning with threshold logic to determine health in a plant application worked quite well. The code itself was fairly clean, organized, and easy for other students to read and understand. Configuring the OLED display sometimes didn't function properly, which taught us we should be testing communication protocols like I2C at low power, along with all the other tests.

In totality, we as a team learned a lot beyond the technical. KiCad not only illustrated moving from the schematic to a Working Pinboard (PCB) but also, along the way, taught us about wire routing, spacing, and part selection. Designing the 3D case and placing sensors showed us the importance of balancing aesthetics and form. Managing time and testing step-by-step were key, since adding new parts sometimes caused problems in parts that were already working.

In the end, the Smart Plant Care project reached its goal. We built a working prototype that shows how embedded systems can solve real-life problems in a simple and helpful way.

16. CONCLUSION

The Smart Plant Care project was able to create a working prototype by combining sensor-based monitoring, designs for embedded systems, and user-related feedback to help users care for their indoor plants. Our final prototype was able to create an environment-aware device to provide feedback to the user while measuring the condition of the environment from the perspectives of soil moisture, temperature, humidity, and light conditions. The device is able to provide real-time feedback and uses a clean and effective user interface. An OLED screen and RGB LED, together with a buzzer, were added to indicate to the user when appropriate feedback to the user is required. The user interface for the device is uncomplicated, so it can be easily used by a novice or someone with experience caring for plants.

During the semester, this project did a great job in helping us understand the approaches to applying engineering principles in real life. We learnt how to work with real-world components, solve integration issues between software and hardware, design a PCB from scratch using KiCad, learn how to document and test, and create cycles to solve problems, much like in a professional engineering role.

While the final product faced a few constraints, the OLED display was unreliable, and the plant had insufficient pins to test all of the features, but the main functionality worked well as intended. This experience has allowed us to have a better understanding of the significance of proper planning, modular testing, and adaptability with design decisions. It has also created an opportunity for future expansions, if we were to incorporate wireless connectivity, mobile apps, and AI recommendations for plant care.

Overall, the Smart Plant Care project accomplished everything we wanted to for the main technical objectives we set out to accomplish at the start of the course; it gave us experience with the tangible aspects of engineering, which we hope will assist us in any future engineering opportunities.