ECE 445

Senior Design Laboratory

Final Report

DIY PLANTIFY

A Self-Moving Potted Plant System For Ideal Sunlight Exposure

---

Team No. 70

Maya Kurup

Joshmita Chintala

Hongshang Fan

TA: Raman Singh

Professor: Olga Mironenko

May 3, 2023

Abstract

This document will provide a more in-depth look into our senior design project, DIY Plantify. DIY Plantify is a self-moving potted plant system for ideal sunlight exposure. This document will consist of our problem statement, solution, design procedure and details, cost and schedule, ethical/safety considerations, and any other research in relation to our project.

Table of Contents

1 Introduction

1.1 Problem

At the root of every plant, it needs 5 different components for it to grow, survive, and thrive: light, air, water, nutrients, and space to grow. There are not many systems put in place to help individuals understand how much sunlight and water a plant needs, when the plant needs it, and how much of it they need. So, a solution to resolve these issues can be very beneficial in people's day-to-day lives when growing plants (simple leaf plants, trees, fruits, or even vegetables) on a smaller scale, but can also be extended to a professional/advanced level that farmers and larger industries can use.

1.2 Solution

Our solution to this problem is DIY Plantify. DIY Plantify is an integrated system using a light sensor, moisture sensor, and notification system to move a potted plant where optimal sunlight is needed for the plant. As well, it alerts the user when moisture levels are low in the soil through a blinking LED notification system.

In this light detecting system, it uses a light sensor, and more specifically in our case, a photocell, to detect where the optimal light is located throughout a certain distance along a straight path. Once the highest light intensity is detected in that straight path after moving a certain amount of steps forward (which can differ based on the size of the room), the moving robot below the potted plant and sensor system will move backwards to that optimal location. With the combination of this light sensor and a moving chassis, we feasibly made a product that can be applied and used in people's day-to-day lives that will optimize the amount of light a plant should receive.

In the notification system that is integrated with the moisture sensor, we are able to detect levels of moisture and alert the user through a blinking LED, which tells them when moisture levels are low in the soil of the potted plant.

All in all, we would create a system that optimizes sunlight, and also warns the user about low moisture levels for small plants and can further be extended to larger systems, which would help everyone at a local level and professional/worldwide level.

1.3 High-Level Functionality

The following are our high-level project functionalities.

1. The moving robot should be able to carry a potted plant with a weight between 2-8 pounds and diameter 10-15 centimeters.
2. When placed indoors in front of a window or on a balcony, without objects obstructing its path, the moving robot should “follow the light” by moving in a straight path to find optimal light within 1 hour.
3. The robot should detect when the moisture level is low in the plant and alert the user when this occurs with a blinking LED notification system.

The above high-level functionalities will allow our robot to determine the amount of sunlight and water the plant needs, when it needs them, and how much of them it needs. Overall, the high-level functionalities are working towards our goal of keeping a plant alive without the need for human intervention.

1.4 Subsystem Overview

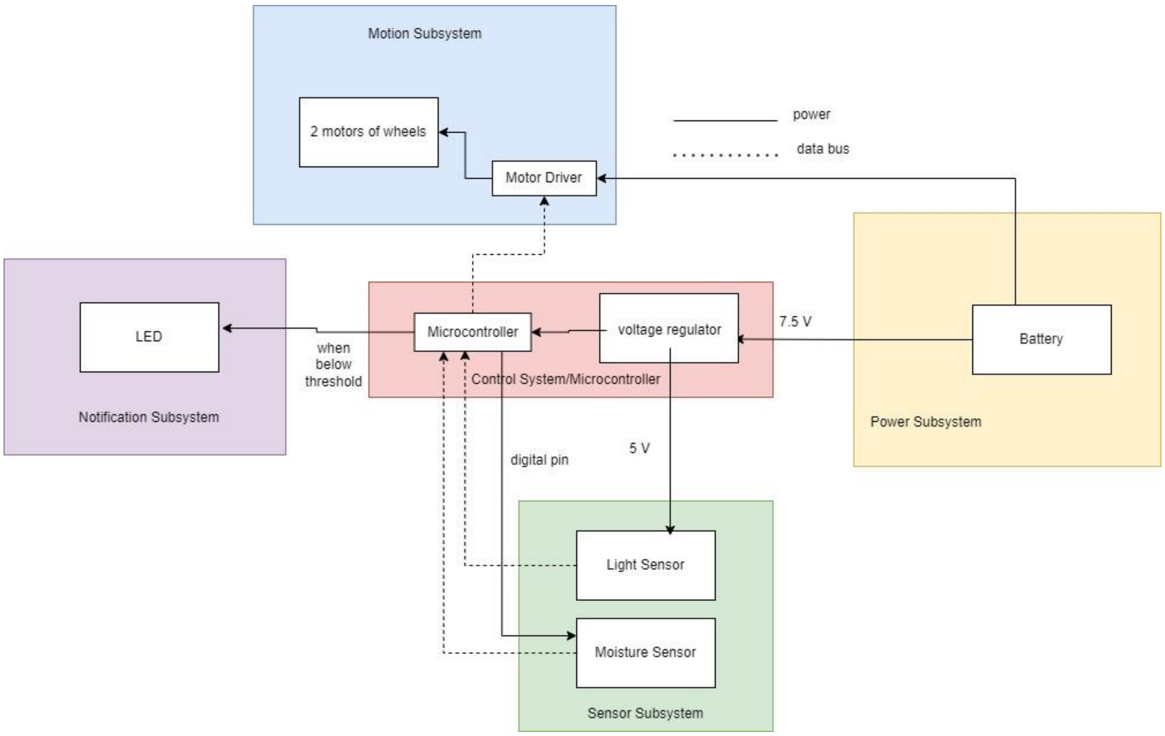


Figure 1: Top-level Block Diagram of the DIY Plantify System

Sensor subsystem

As seen in Figure 1, our sensor subsystem consists of a light sensor and moisture sensor. The light sensor will record light intensity data in a certain room along a straight path. The moisture sensor will watch out for low moisture levels, and will alert (Notification subsystem) the user when the plant needs to be watered.

Motion subsystem

The motion system is a chassis robot carrying the pot. The microcontroller will send commands to it to move the plant to the desired position.

Notification subsystem

The notification subsystem will consist of a LED. The LED will blink when the moisture sensor detects dry soil in the potted plant.

Power subsystem

The power subsystem consists of 5 AA batteries (7.5 Volts), and a voltage regulator. This handles power regulation of the microcontroller, light sensor, and moisture sensor, and powering the motor driver.

Control System/Microcontroller

The microcontroller integration in our dev board controls the motion, sensor, and notification subsystems. It is able to send signals to the motor driver, moving the robot to the desired location where optimal light is. As well, it analyzes the data from the moisture sensor in order to send a signal to the LED to alert the user the plant needs to be watered.

2 Design

2.1 Design Procedure

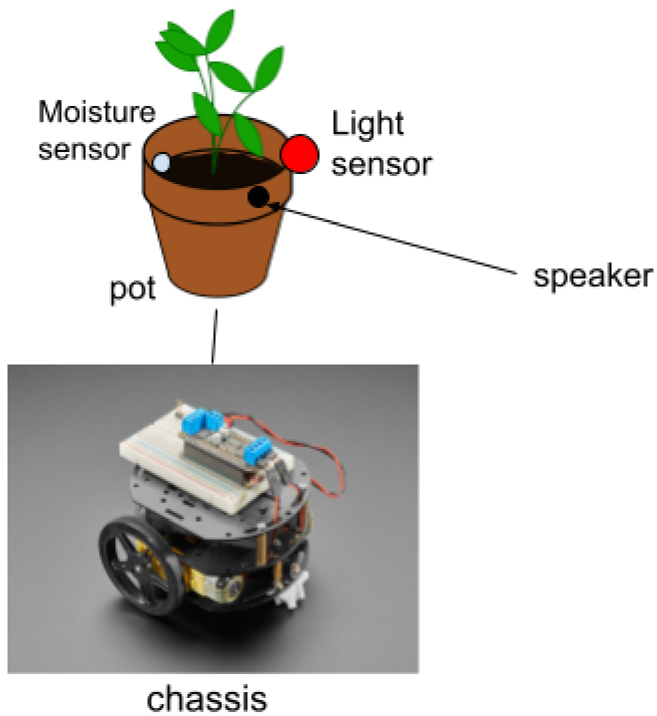


Figure 2: Original Overview of the DIY Plantify System

Figure 2 shows the original design for the DIY Plantify System. In this design, we had planned to place the pot directly centered on the chassis, and the light sensor on the pot of the plant itself. This design also has a speaker on the pot, which would beep to notify the user to water the plant.

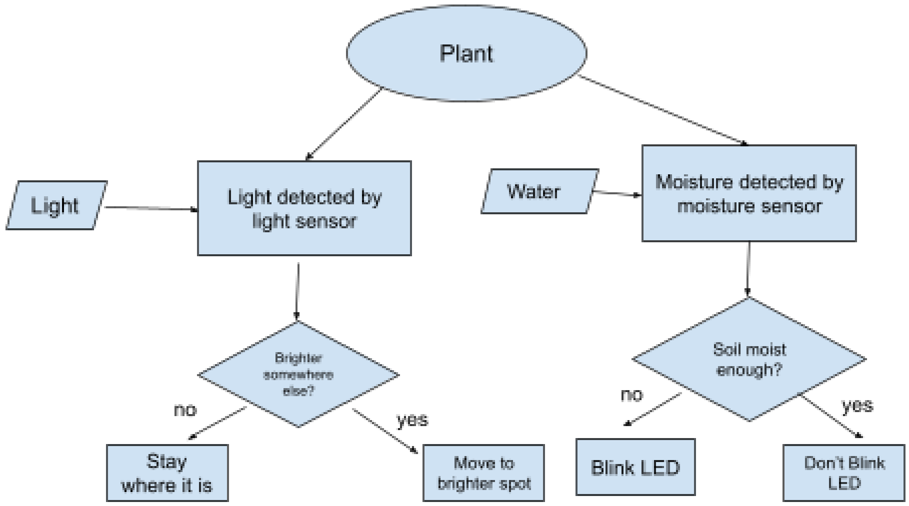


Figure 3: Basic Functionality of the System

Figure 3 is a flowchart explaining the basic functionality of the DIY Plantify system. When light is detected by the light sensor, if the light is brighter somewhere else, it must move to the brighter spot. If not, it should stay where it is. When moisture is detected by the moisture sensor, if the soil is moist enough, it should not blink the LED, whereas if it isn't, it must blink the LED.

Figure 4: Final Product

Figure 4 shows our final product. As shown in Figure 3, we decided to replace the speaker in the notification subsystem, to an LED. This way, the user would not be disturbed with the beeping noises of the speaker. As can be seen in Figure 4, we did not have the plant on the chassis' center, as we had planned originally. It is now on a platform at the back of our chassis so that it would not block the light sensor, which we had to implement on the breadboard rather than on the pot of the plant.

Below we will walk through our original schematics of our PCB design, to what we are now using for our final product's design.

Initial PCB Design:

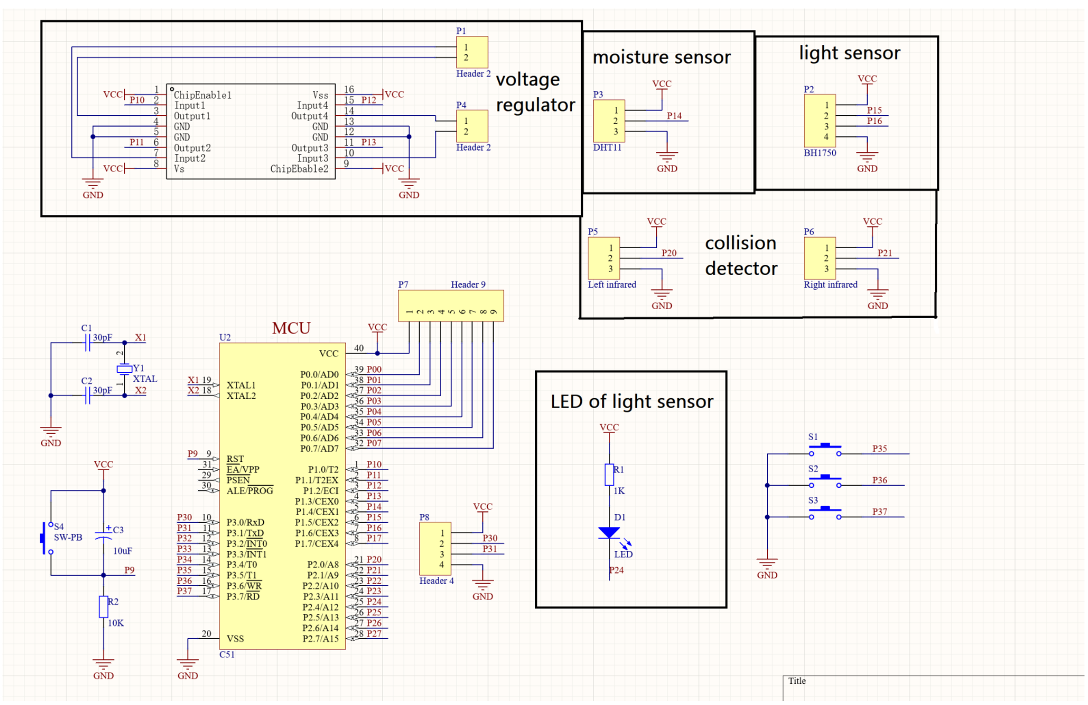


Figure 5: First PCB Schematic

As shown in Figure 5, in our original PCB schematic we incorporated all of our necessary components like the microcontroller, the light and moisture sensors, the voltage regulator, and LED notification system. The main component we did not have originally was the motor driver. As well, we did not have a proper integration of the microcontroller.

Second PCB Design:

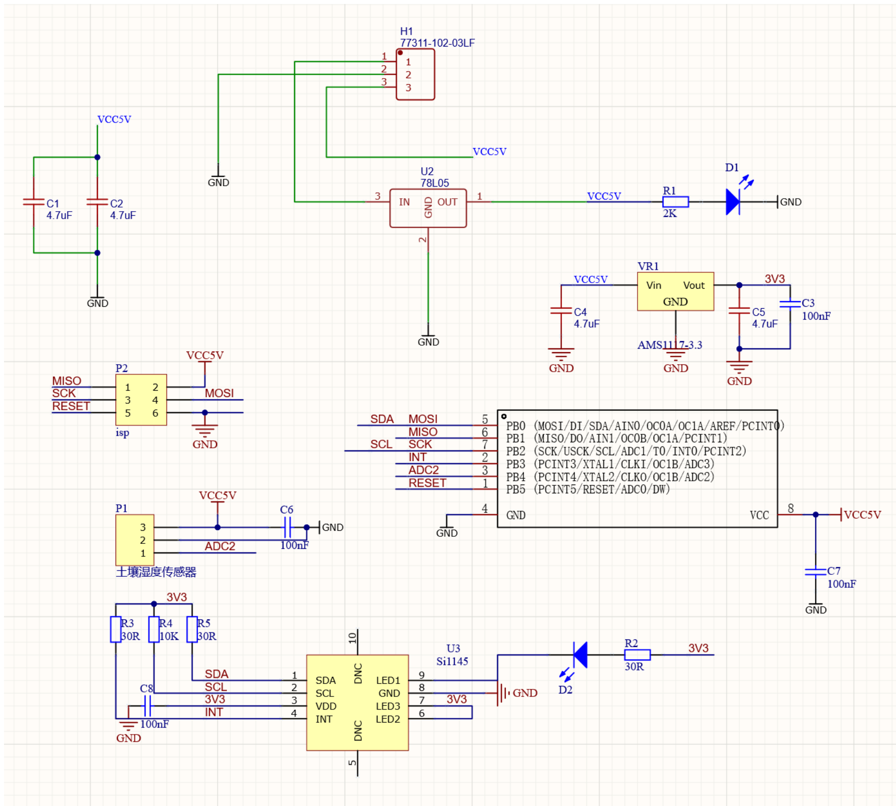


Figure 6: Second PCB Schematic

In our second PCB schematic we incorporated the same components as before, however we re-implemented our microcontroller, as our first one was not resulting in a proper function. As well, we did not fully implement our motor driver/motion subsystem.

Third PCB Design:

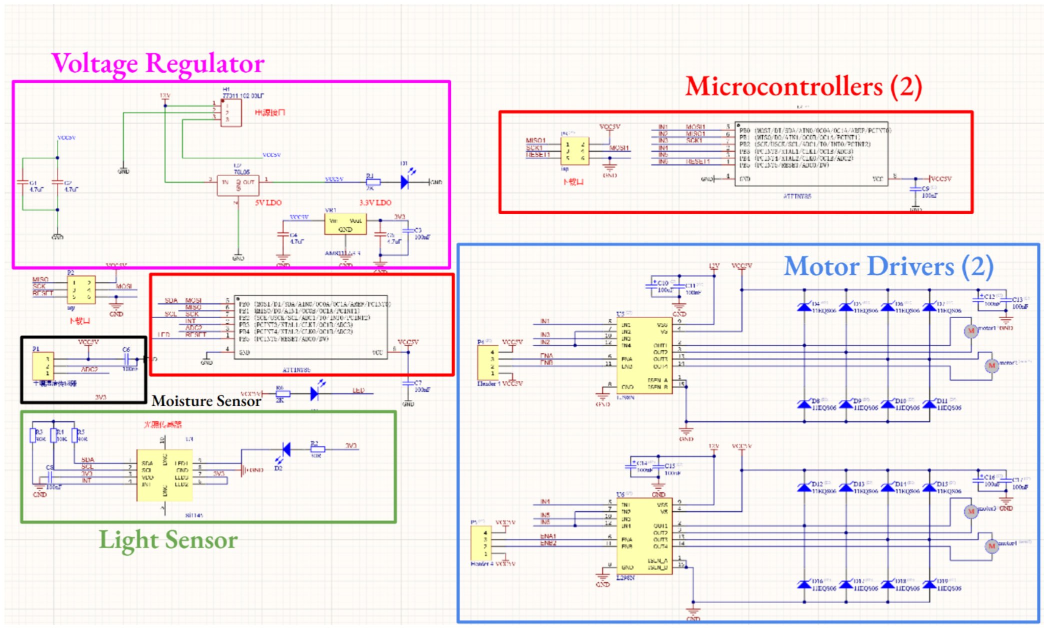


Figure 7: Third PCB Schematic

In our third and last PCB schematic, we had all components needed for the final product design, however we implemented 2 motor drivers, and ended up using 1. All in all, the functionality of our final design would have been able to work, if it wasn't for issues when soldering and making those specific connections.

Final Design:

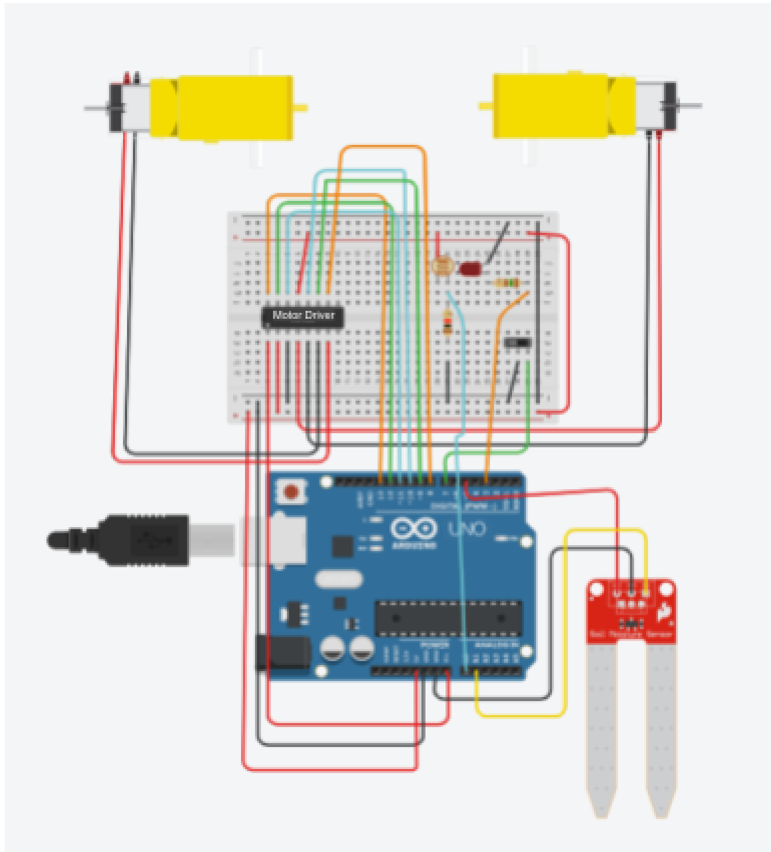


Figure 8: Final Product Design

Here in our final design, we ended up implementing a breadboard with a bought dev board. This is due to the fact that we were having issues getting our PCB to work, so we opted to use a combination of a breadboard and dev board. Within this design schematic we have implemented our motor driver in the top left, which is able to connect and drive our 2 motors located in the back of the chassis. Next, we have our light sensor subsystem to the right of the motor driver, which is connected using a photocell and resistor to create a voltage division circuit. To the right of the light sensor subsystem, we have our LED notification system which incorporates a blinking LED and resistor in series. Connected to our arduino directly, we have our moisture subsystem which uses a moisture sensor, testing and collecting data of moisture levels based on what the metal prongs are connected to. One last implementation we added was a switch, which is able to turn on the system and move the chassis where maximum light is. As well, this button will turn on the moisture sensor to detect low moisture levels, and flash the blinking LED when below the threshold.

3 Design Verification

3.1 Sensor Subsystem

3.1.1 Light Sensor

For our light sensor, our main requirement is that the light sensor should collect light data correctly. To do this, we must show that the photocell light measurements decrease as distance from the light source increases. To verify this, we tested the light sensor with the phone's flashlight with different intensity, and checked if the light sensor’s data could indicate which intensity is most bright.

First, we show the results when holding a flashlight at a distance, then move it closer to the photocell. Figure 9 and 10 show that when increasing the light intensity as the source comes closer to the photocell, the voltage response of the light sensor system increases, as the resistance decreases.

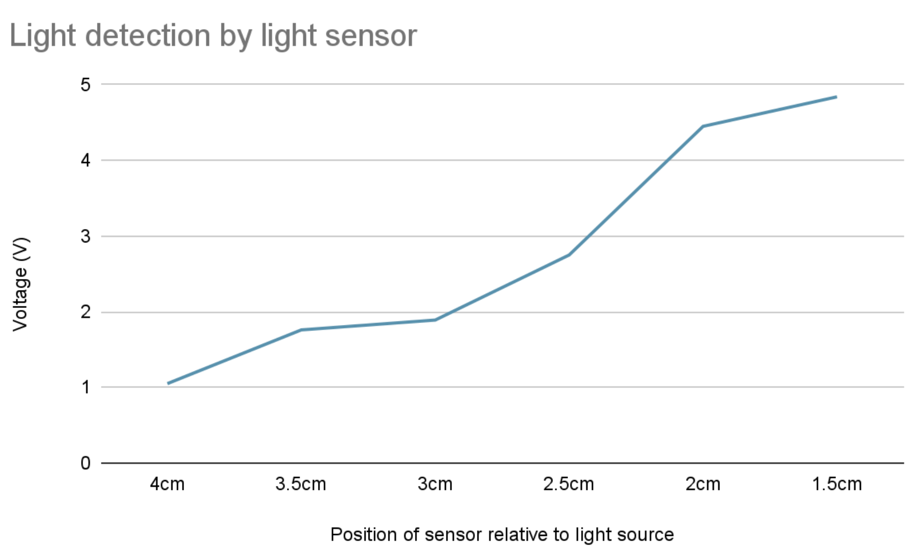


Figure 9: Light Data (Position vs. Voltage)

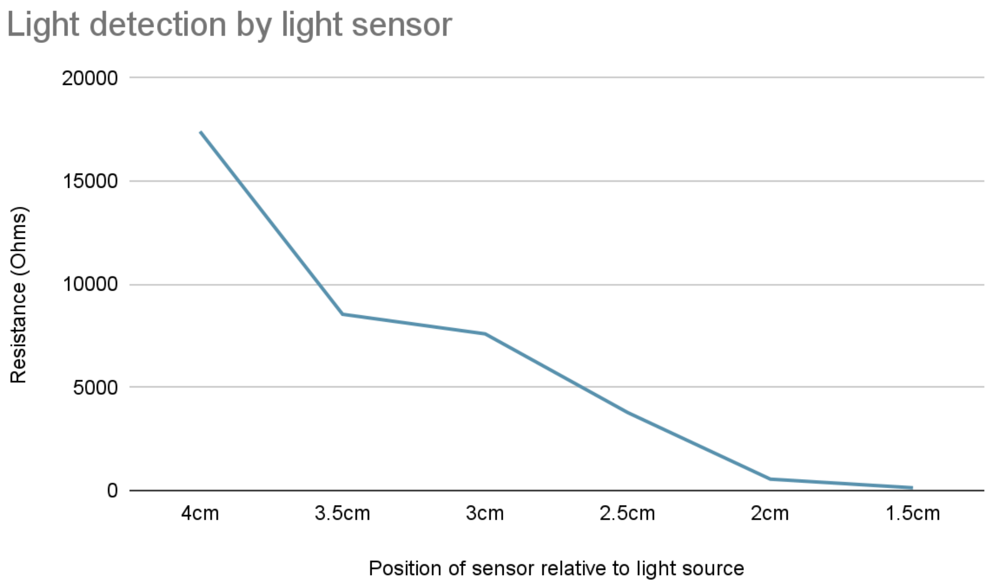
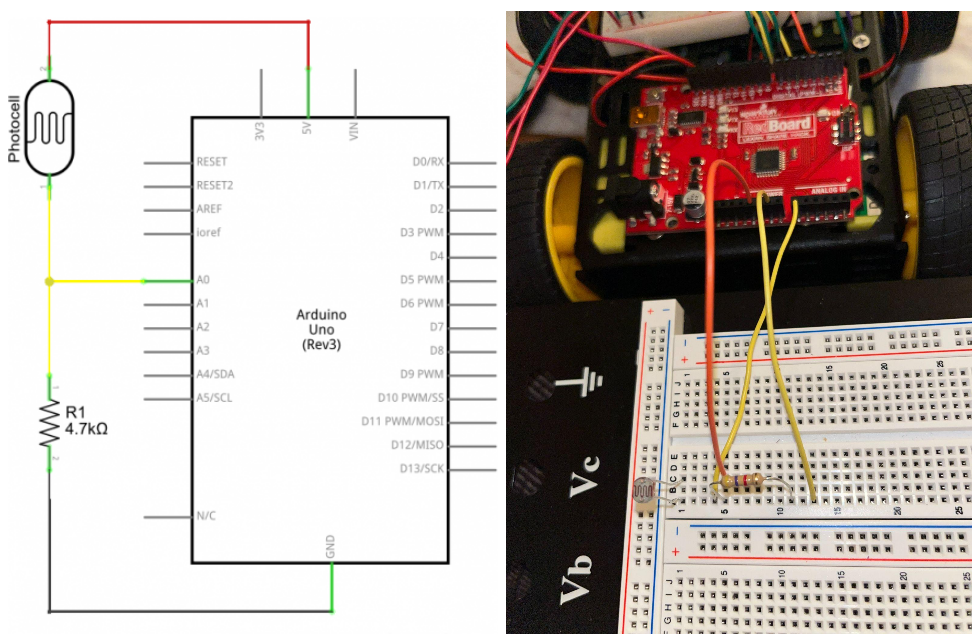


Figure 10: Light Data (Position vs. Resistance)

We tested the photocell with the voltage divider circuit below which produces a voltage, dependent on the photocell's resistance. Here we have a 4.7kΩ resistor on the ground side and the photocell on the 5V side. This means that as the cell's resistance increases, (meaning the sensor's surroundings are getting darker) the voltage on A0 will decrease.



Figures 11-12: Light Sensor Setup

3.1.2 Moisture Sensor

For our moisture sensor, our main requirement was to show that the moisture sensor is able to detect dryness within 2 days, and send an alarm when soil is dry (below a threshold). We verify this by testing different levels of moisture by gradually watering the plant and noticing the change in moisture levels read by the sensor. *The values provided by the sensor range from approximately 0 to 880 (0 = dry; 880 = wet). At first, the soil was a bit dry, but our specific plant retains water for up to a week. So, immediately after watering it showed a high level of moisture. And, as time went on, the moisture sensor data was responding accordingly, showing that the moisture levels decrease.

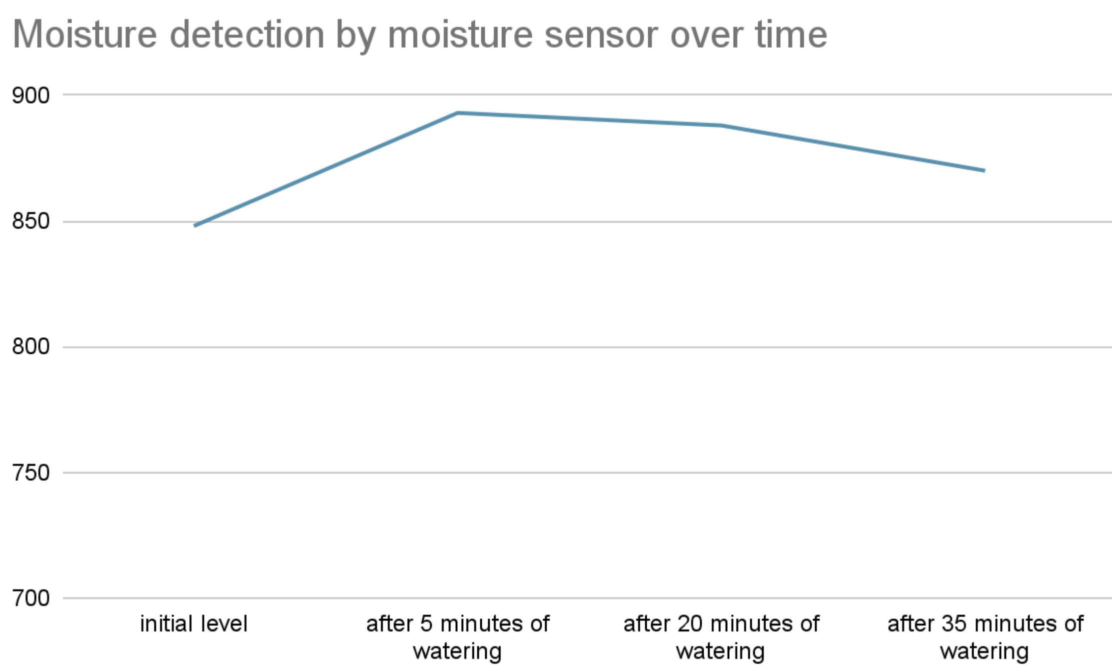
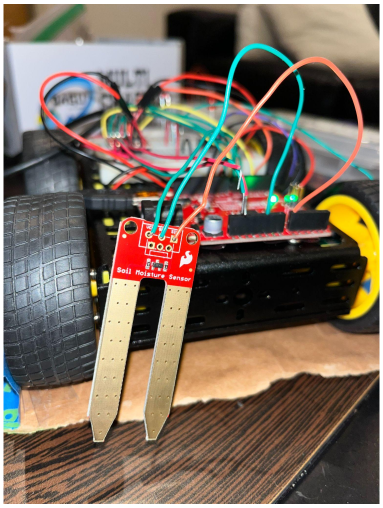
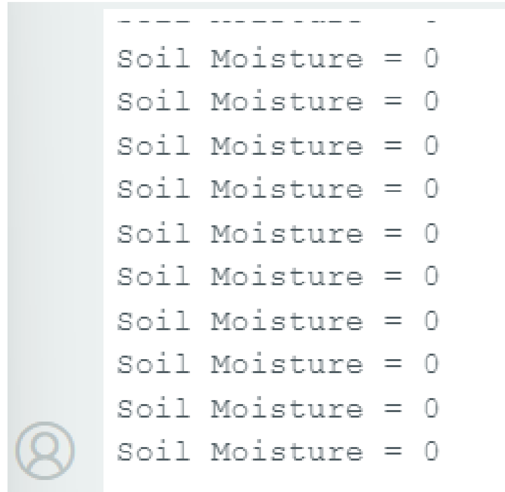


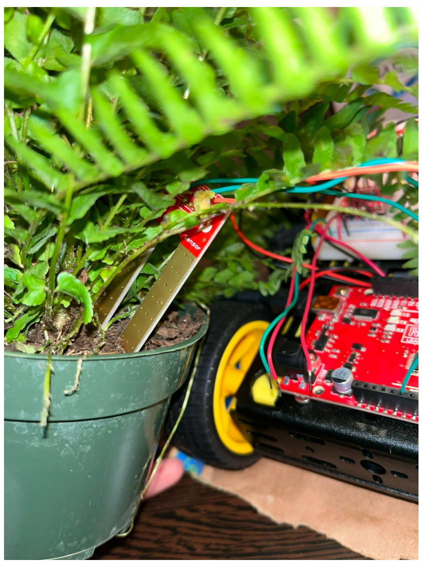
Figure 13: Moisture Data (Time vs. Moisture Level)

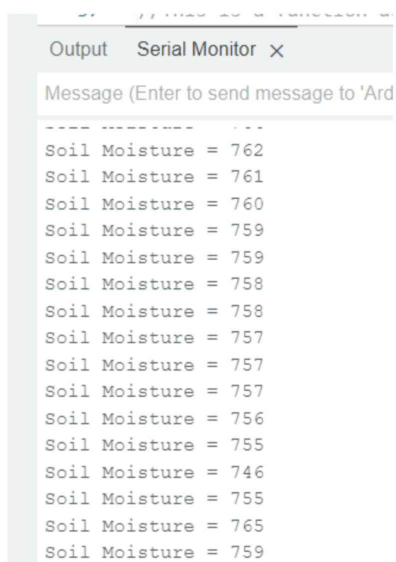
Below are some images of the moisture sensor, testing levels of moisture in low, medium, and high mediums. These three mediums include air (level - 0), semi-dry soil (level - 543), and water (level - 899). It is important to note that the units of this moisture sensor data is in terms of a range in value that the moisture sensor is configured by (0-900). As well, it is important to note that the data that appears on the screen is being tested directly from the moisture sensor at that time, every second.



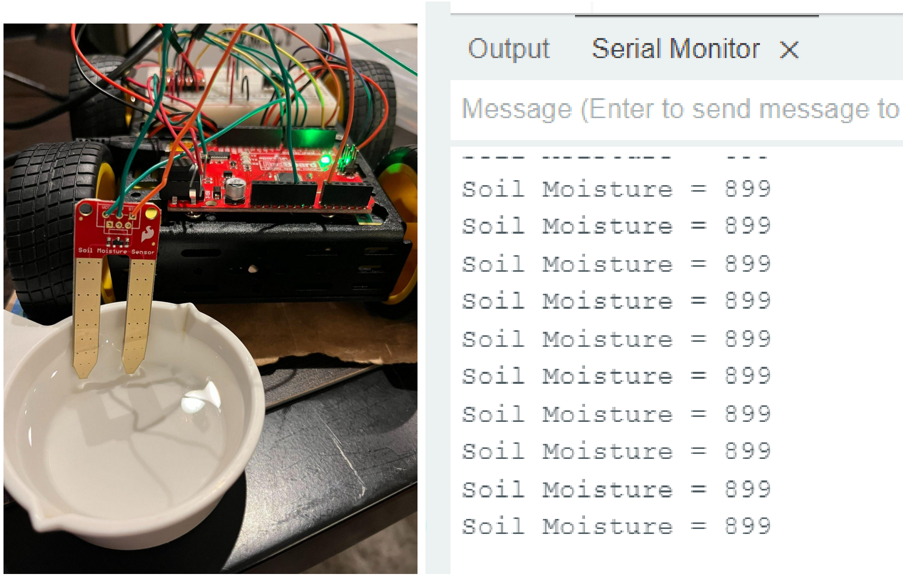


Figures 14-15: Soil moisture level is 0 when moisture sensor is in air





Figures 16-17: Soil moisture level is ~760 when moisture sensor is in soil that has not been watered in 4 days



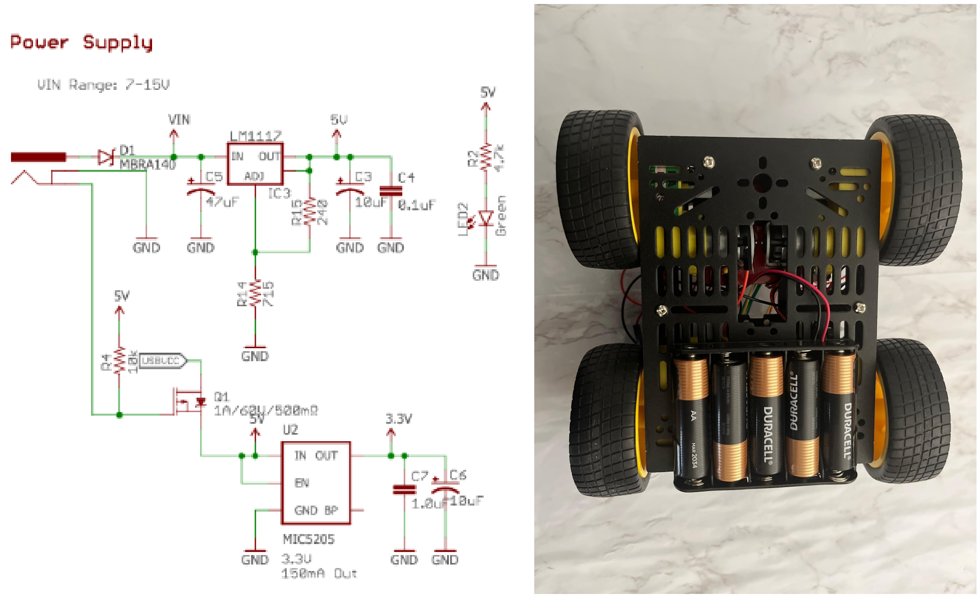
Figures 18-19: Soil moisture level is 899 when sensor is placed in water

3.2 Motion Subsystem

In our motion subsystem, we use 2 motors, which are connected directly to our motor driver. By implementing certain commands in our code, we send signals from the microcontroller to the motor driver, which then drives our 2 motors to where we want it to move. The main requirements for the motion subsystem are that the chassis should be able to move so that the plant does not fall off of it, and the platform should be large and durable enough to hold an indoor, medium-light plant. We verified this by testing that the chassis was able to carry objects ranging in 2-6 pounds. As well, it was able to hold an object with a diameter between 10-15 cm.

3.3 Power Subsystem

Our power subsystem consists of a 7.5V power source from 5 AA batteries, which then gets transmitted through our voltage regulator. This voltage regulator then sends 5V of power to our microcontroller, light sensor, motion sensor, and LED notification system. As well, the 7.5 V of power is directly connected to our motor driver, which directly drives our 2 rear wheels. As seen in our schematic of the microcontroller power supply, 5V is being outputted for the LED, motion, and light sensors. As well, it is important to note that we did not use the 3.3 output voltage, simply because our parts being used needed a higher voltage, hence the use of 5V.



Figures 20-21: Images of Power Supply Schematic & Design

In order to verify our power was accurately being detected in our system, we used the green light implemented on our Dev board, which illuminates when the entire PCB and design is connected to the power source.

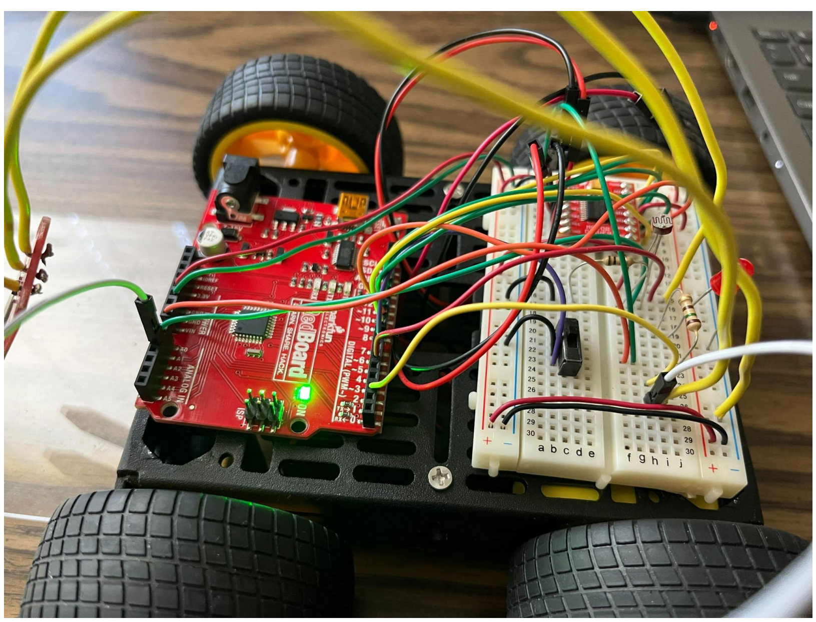
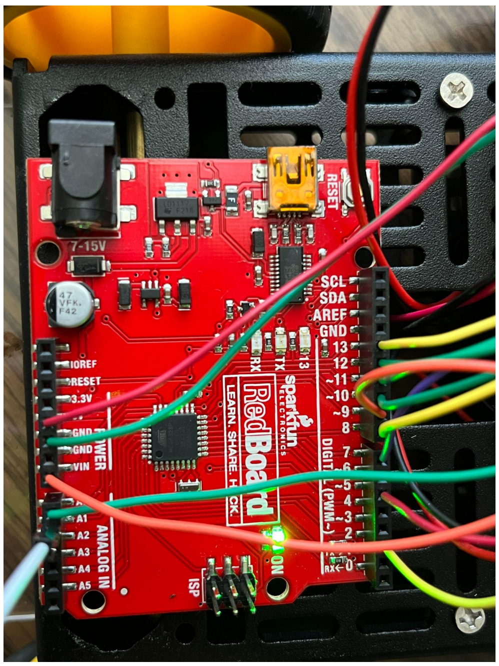
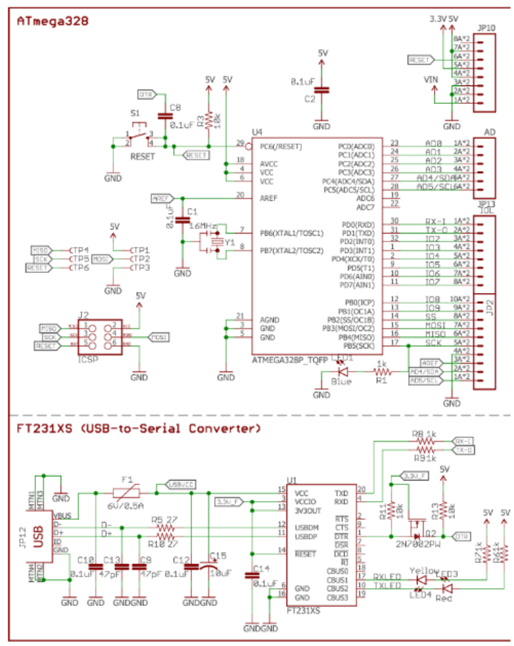


Figure 22: Image of Power Supply Verification

3.4 Microcontroller Subsystem

As previously mentioned, our original PCB and microcontroller schematic was unable to be implemented. So, we bought an arduino, which integrates all of our necessary components of the PCB to have a fully functioning robot. Below, we will show the schematic of the specific arduino we are using, which shows the proper connections that are made from the voltage regulator to the microcontroller, and so forth.





Figures 23-24: Images of Arduino & PCB Schematic

One of the requirements for our microcontroller subsystem is that it should be able to send a signal of the desired location to the chassis, based on the light sensor. We verified this by giving the chassis a set of data and checking whether it arrives at the desired location according to the data.

Another requirement is that it must be able to analyze data from the moisture sensor, and send a signal to the LED when indicated by the data. We verified this by putting the moisture sensor into a dry source (dry soil, paper, air), and checking if the LED will blink.

3.5 Notification Subsystem

The main requirement of the notification subsystem is that it must be able to indicate dryness to the user using a notification system (blinking LED). To verify this, we set a minimum threshold of the moisture sensor in the program, and let the processor send a signal to an LED to blink. We then checked if the LED blinks when below threshold. In Figure 24, we set that threshold to be 600, based on our plant's ability to hold moisture.

The code below demonstrates that functionality:

```
if (count>=total_count){
  digitalWrite(SOILPOWER, HIGH);//turn D5 "On"
  delay(10);//wait 10 milliseconds
  val = analogRead(SOILPIN);//Read the SIG value form sensor
  digitalWrite(SOILPOWER, LOW);//turn D5 "Off"
  if(val<600){
    digitalWrite(LED_PIN, HIGH);
    delay(DELAY_TIME);
    digitalWrite(LED_PIN, LOW);
    delay(DELAY_TIME);
  }
```

Figure 25: Code setting a threshold for moisture sensor value

4 Cost and Schedule Analysis

4.1 Cost Analysis

The total cost of our project was about $176.07.

Description	Manufacturer	Quantity	Price	Link?
Chassis	Sparkfun	1	$40.00	Link
Moisture Sensor	Sparkfun	1	$6.50	Link
Voltage regulator	Found in lab	1	$0.00	
Light Sensor	Adafruit - Found in lab	1	$9.95	Link
Foliage Plant	Home Depot	1	$6.52	
Jumper Wires	ECEB Supply Shop	20	$7.40	
PhotoCell	ECEB Supply Shop	1	$1.50	
Mini Power Switch (SPDT)	ECEB Supply Shop	1	$1.50	
Light Meter	Leaton	1	$14.99	Link
9V Lithium Battery	Walgreens	1	$23.99	

AA Batteries	Walgreens	Pack of 8 - 5 were used	$14.99	
RedBoard Microcontroller	SparkFun	1	$25.66	Link
Motor Driver	SparkFun	1	$23.49	Link

Figure 26: Itemized List of the Components and Costs

4.2 Schedule

Week	Original Task	Person
February 20-27	Order parts for prototyping	Everyone
	Research plant & buy it	Maya
	Research chassis and figure out best way to implement it	Joshmita
	Start basic structure of circuit setup (digital)	Hongshang
February 27 - March 6	Finalize all components being used	Everyone
	Research the best way to acquire all data (light sensor, moisture sensor, movement), determine how the data will be presented,	Everyone

	adapted, and used, and understand which data points are going to be taken	
March 6-13 (Spring Break)	Make sure all the physical components are ready to be put together and implemented	Everyone
March 13-20	Build Mock-up Prototype and start testing data	Everyone
March 20-27	Refine Prototype	Everyone
March 27 - April 3	Test Integrated System	Everyone
April 3-11	Finish all testing (including light sensor, moisture sensor, moving robot/chassis)	Everyone
April 10-17 (Mock Demo - TA)	Test the complete system and make updates based on feedback	Everyone
April 17-24 (Final Demo - Instructor & TAs)	Make updates based on feedback	Everyone
April 24 -	Be prepared :)	Everyone

May 1 (Final Presentation - Instructor & TAs)		

Figure 27: Schedule for Project Progression

5 Conclusion

5.1 Accomplishments

In conclusion, our team was able to successfully design, build, and test a working robot that is able to detect optimal sunlight for a plant, and move to that designated position. As well, we successfully integrated a moisture sensor and notification system to alert a user when moisture levels are low in the soil of the potted plant through a blinking LED notification system. We were able to create a functioning algorithm that behaves based on the specific commands that programmed it to do. Our team as a whole further learned how to properly execute a design in terms of brainstorming ideas from scratch, developing that idea to incorporate a PCB, breadboard, and other technical components, and designing PCB and block diagrams from that idea. We learned to be flexible with our original goals based on the given time, money, and ability constraints. And lastly, we learned how to debug each component in our arduino code, which is similar to Python, when our robot was not functioning as we wanted it to.

5.2 Uncertainties

Some challenges that we faced include having difficulties moving the robot on tougher surfaces. We found that our robot moved much smoother, without any hesitation or error on smoother surfaces that did on have any dust or small trash collection. As well, when trying to upload our code onto the arduino, we faced issues with our computers. Sometimes our computer would simply shut down because of the over generation of heat from that system, or it would freeze due to an overload. When figuring out how to move our robot in a straight path, we originally had some difficulties because we had the front 2 wheels also connected to motors, creating further friction of the wheels. But, once taken to the machine shop, they quickly and nicely made a custom axle for our 2 front wheels, removing any friction that was created by those motors we originally had (and couldn't program due to having only 1 motor driver). Another issue we faced was deciding where to attach our platform for our plant, because we originally wanted it to be on the top of our chassis. But, after further considerations, and help from the machine shop, we added a glass platform to the back of our moving robot so that the plant was in a safe spot, not obstructing the photocell from gathering data, as well in a nice spot where the moisture sensor could reach the soil of potted plant.

5.3 Future Work & Alternatives

In the future, given more time and money, we would definitely add further implementations to our project. Firstly, we would add a self-watering/sprinkler system to water the plant its desired amount of water when the moisture sensor detects low levels of moisture. Secondly, we would change the functionality of the code, and apply a motion sensor in order to detect when objects are in obstruction of the robot's path. This would allow us to move the robot in X and Y coordinates, and move throughout the day, without running into someone or something. Lastly, we would want to scale up the system and apply a larger chassis, in order to carry a larger and heavier potted plant. This would also result in the ability to change the design, and have the plant be located on top of the chassis, rather than the back. With this change, it would allow us to center the force of gravity on the system, allowing for heavier and larger plants.

5.4 Ethical Considerations

- In relation to the IEEE Code of Ethics
    - We must comply with ethical design and sustainable development practices and disclose factors that might endanger the public or environment. In relation to our project, since we plan to make an automated robot, we must ensure that it doesn't endanger the environment it is in, in any way.
    - We must mention the societal implications and capabilities of our robot, making sure the end-user is well-informed.
    - Since there is a group doing something similar in terms of measuring values relating to plants (they are measuring moisture values), we must make sure all our ideas are our own and original.
    - We must make sure all our claims and estimates are accurate and realistic, and accept honest feedback and criticism from our TA to make our project as precise as possible.

    - We must credit any sources, code, data, or information we use in the process of making our project.
    - We must use any equipment only if trained or experienced to use them.
    - We must work well with our team, and treat each other fairly and with respect.
- In relation to the ACM Code of Ethics
    - We must consider and provide evaluations of any associated risks of our project.
    - We must design and develop a safe and secure project.
    - We must make sure that the user is the main focus when making and producing this project. The end-user/public should be considered throughout the entire process, and should be made with them in mind.

Addressing Safety Issues:

- Make sure safety protocols are followed while in the lab soldering, using our PCB, and testing our sensors.
- Make sure to be aware when something can put ourselves at risk (when testing and building)
- Set boundaries, and make sure to work collaboratively so that not one person is more at risk for injury than another.
- Make sure that the end-product is safe and does not harm the user of the device, the environment, and the public. Users should be warned to be careful around the operating area despite the low speed of robot.
- Battery will be completely sealed so that moisture and sunlight won't cause any risk for the battery.

6 References

[1] “ACM Code of Ethics and Professional Conduct,” *ACM*. [Online]. Available: https://www.acm.org/code-of-ethics. [Accessed: 09-Feb-2023].

[2] “IEEE Code of Ethics,” *IEEE*. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 09-Feb-2023].

[3] “HOW TO MEASURE LIGHT FOR PLANTS,”Greenery Unlimited. [Online]. Available: https://greeneryunlimited.co/blogs/plant-care/how-to-measure-light-for-plants

[4]“10 Best Indoor Plants for Your Health,”GOOD Housekeeping. [Online]. Available: https://www.goodhousekeeping.com/home/gardening/g40742429/best-indoor-plants-for-health/

[5] “30 Indoor Plants That Like Direct Sunlight,”the Spurce. [Online]. Available: https://www.thespruce.com/best-houseplants-for-sun-4147670

[6] “23 Of The Best Medium-Light Houseplants,”Gardener’s path. [Online]. Available: https://gardenerspath.com/plants/houseplants/medium-light/

[7] “30 Indoor Plants for Low Light,”HGTV. [Online]. Available: https://www.hgtv.com/outdoors/flowers-and-plants/houseplants/14-indoor-plants-for-low-light-pictures

[8] Multi-Chassis,sparkfun. [Online]. Available: https://www.sparkfun.com/products/12089

[9] Ada, L. (n.d.). *Adafruit SI1145 Breakout Board - UV index / IR / visible sensor*. Adafruit Learning System. Retrieved March 29, 2023, from https://learn.adafruit.com/adafruit-si1145-breakout-board-uv-ir-visible-sensor/downloads

[10] Mutlaq, S. A. (n.d.). *Soil Moisture Sensor Hookup Guide*. Soil Moisture Sensor Hookup Guide - SparkFun Learn. Retrieved March 29, 2023, from https://learn.sparkfun.com/tutorials/soil-moisture-sensor-hookup-guide?_ga=2.13092599.1940704549.1680147698-1735945614.1675881698

7 Appendix (R&V)

Subsystem Overview

Sensor Subsystem

Requirements	Verification
The light sensor should collect light data correctly. To do this, show that the photocell light measurements decrease as distance from the light source increases.	Test the light sensor with the phone's flashlight with different intensity, and see if the light sensor's data can indicate which intensity is most bright.
The moisture sensor must be able to detect dryness within 2 days, and send an alarm when soil is dry (below a threshold).	Test different levels of moisture by gradually watering the plant and notice the change in moisture levels read by the sensor. *The values provided by the sensor range from approximately 0 to 880 (0 = dry; 880 = wet)

Motion subsystem

Requirements	Verification
The chassis should be able to move so that the plant does not fall off of it, and the platform should be large and durable enough to hold an indoor, medium-light plant.	Platform & chassis system should be able to carry objects ranging in 2-6 pounds. As well, an object with a diameter between 10-15 cm.

Notification Subsystem

Requirements	Verification
Must be able indicate dryness to the user using a notification system (blinking LED).	Set a minimum threshold of the moisture sensor in the program, and let the processor send a signal to an LED to blink. See if the LED blinks when below threshold.

Power Subsystem

Requirements	Verification
The power system should provide a voltage, and be regulated throughout the entire system.	Use the “ON” LED on the Dev board, making sure it lights up green, indicating power is on.

Microcontroller

Requirements	Verification
Able to send a signal of the desired location to the chassis, based on the light sensor.	Give the chassis a set of data and check whether it arrives at the desired location according to the data.
Able to analyze data from the moisture sensor, and send a signal to the LED when indicated by the data.	Put the moisture sensor into a dry source (dry soil, paper, air), and check if the LED will blink.