Mohawk College of Applied Arts and Technology
Department of Electrotechnology

CENG-10024 – Capstone Project

Automated Plant Pot

Project submitted April 8th, 2022

Project Completed By:
Hajar Farag

Abstract

This report is about the capstone project that was assigned as the final project of the program. The purpose of the project is to leave a plant unattended so it can find optimal sunlight and water the plants when required. This is done using an Arduino to control different components to achieve the goal of the project.

Table of Contents

Table of Figures

1. Introduction

There are artificial lamps that could supply artificial sunlight to indoor plants, however artificial lamps do not supply the same wavelength of light as sunlight. Which would lead the plants to miss out on the full benefits of the sunlight. Nowadays everything is automated which is what gave me the idea to create an automated sunlight detecting pot. While working at Evertz, I learned how to create scripts to do automated testing. This opened my eyes to how everything around us is slowly being changed into an automated system. From smart fridges to smart cars, everything is being automated. My mother always moves her plants towards sunlight three times a day for the plants to get the optimal amount of sunlight. This system will allow the pots to move towards sunlight and waters the plants without any supervision.

2. Theory of Operation and Background Information

The automated plant pot will supply a sufficient amount of sunlight as well as water the plants when needed. An Arduino will be used to control the motors and collect inputs from the sensors. If the Arduino receives a value of less than 50% on the Moisture sensor, it will actuate the servo motor and move the water gate allowing water to flow to the soil. The LDR sensors send constant data to the Arduino and the Arduino checks the data in a loop. If one LDR sensor has a greater value compared to all the LDR sensors, the Arduino sends a PWM to the H-bridge which then controls the DC motors speed and direction of movement. Once the Pot starts moving, the Arduino code checks the IR sensor every loop to see if there is an object in front of the sensor. If the value on either IR sensor is above 250, the motor would move the opposite direction of the where the object is.

One of the technical issues encountered were the rover tracks. The tracks did not have enough traction since it was made from PLA filament which resulted in the pot not being able to move.

One of the possible solutions would be to replace the tracks with a rubber belt which would have enough traction to move the pot. This solution was not used since the right track size was not available online.

Another solution would be replacing the tracks with rubber wheels which would make the whole pot more stable and would have enough traction to move the pot. This solution was used for demonstration since it was the optimal solution, however rubber tracks were not found tires online, so tires from Remote-Controlled Car were used instead.

Similar projects to my project would be a sumo bot: the sumo bots use motors and IR sensor which are some of the materials that the automated plant pot uses. Another similar project would be AutoPot Watering system. This is a similar application to my automated plant pot due to the watering system which waters the plants on its own. The contributions to the project would be 3D designing the housing, pot, and water reservoir, adding a moisture sensor which would detect the moisture of the plants soil, watering the plant using a water reservoir, and moving the pot towards sunlight.

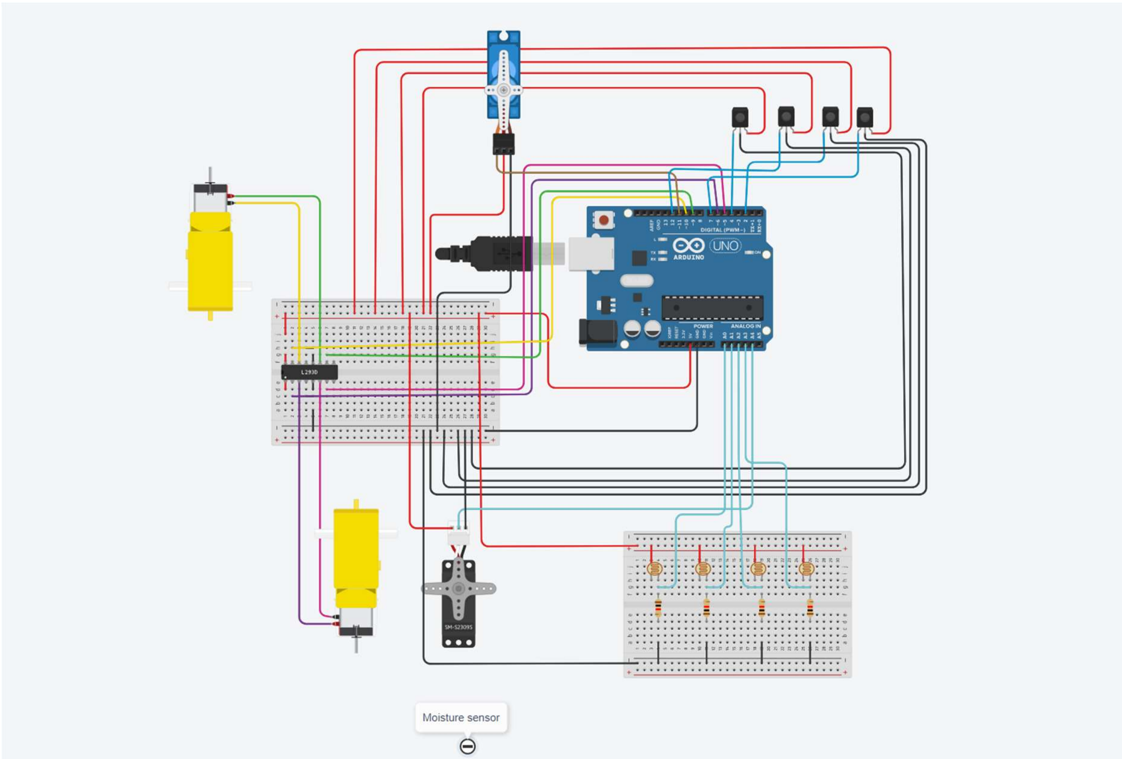


Figure 1: Theoretical circuitry connections

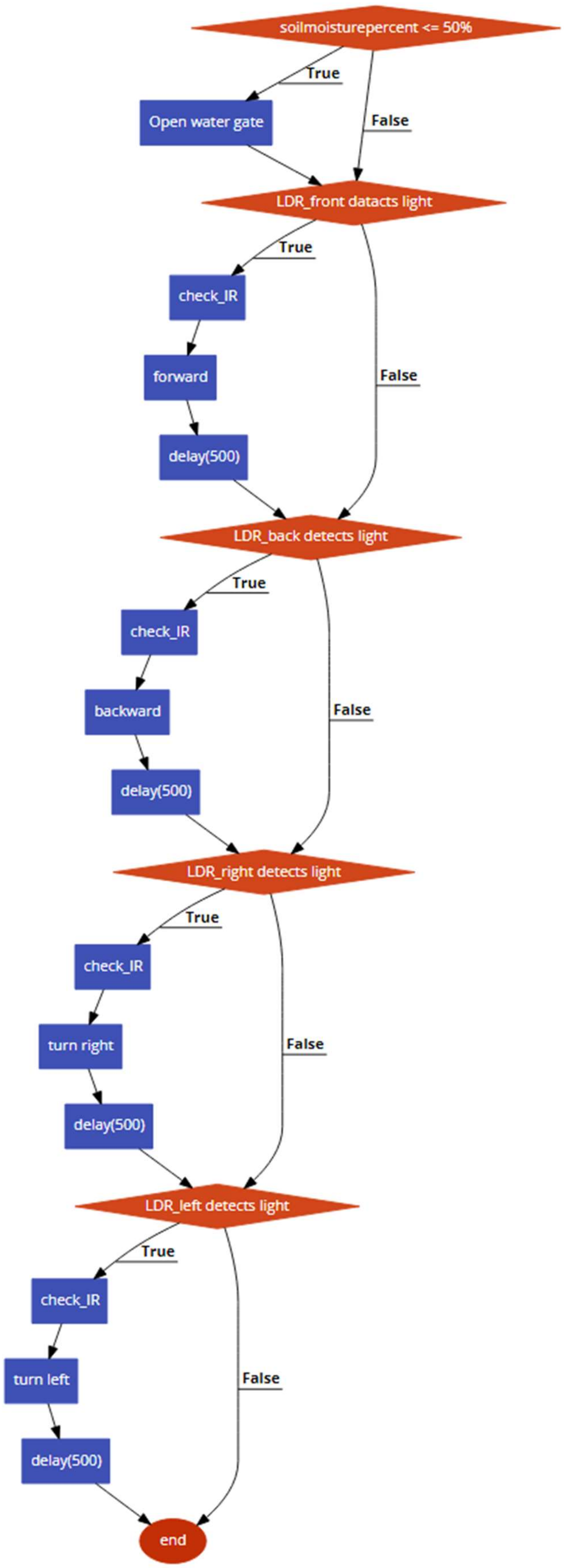


Figure 2: Flow chart logic

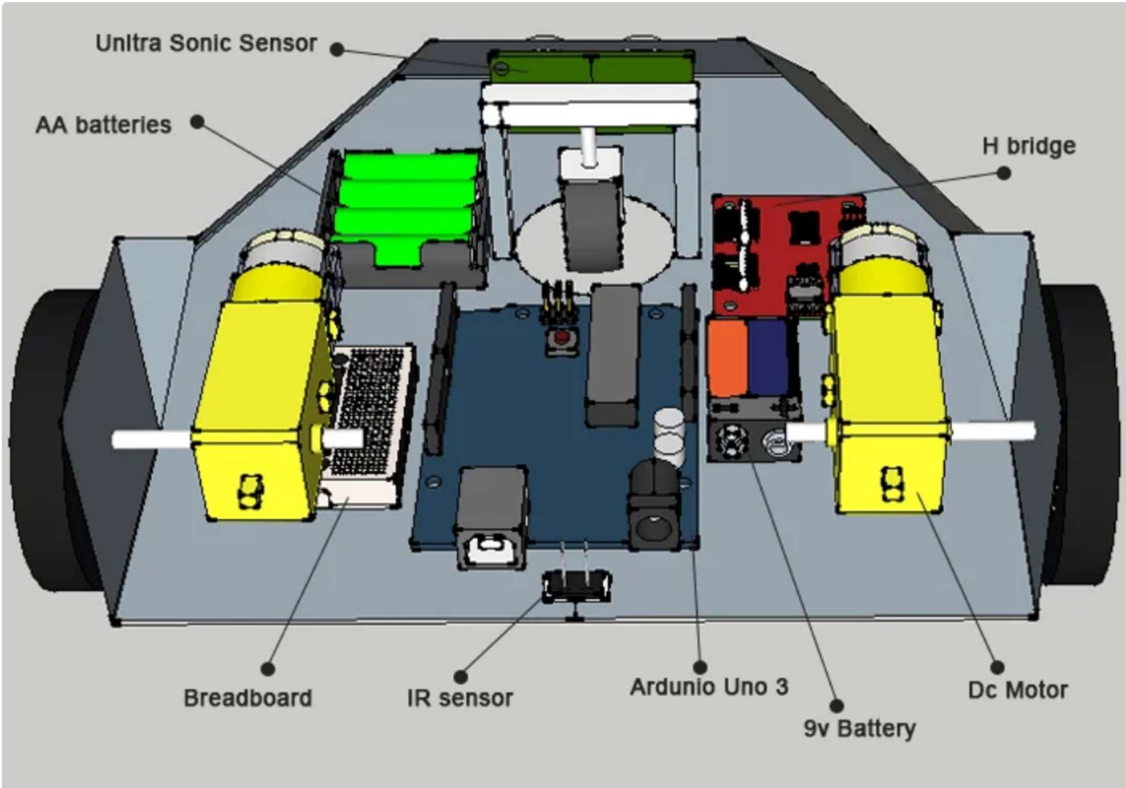


Figure 3: Sumo Bot Schematic

Figure 4: Auto Watering System

3. Project Description

The Automated Plant Pot follows sunlight and waters the plants when the moisture of the soil decreases below a certain percentage. An Arduino based controller will be mounted on the pot which will control where the pot will move. Motors will be placed on the bottom of the pot to move the pot towards sunlight. LDR sensors will be used to detect the sunlight. IR sensors will be used to make sure the pot will not bump into anything while moving. A pulley system will be used to make the pot move to the optimal point where the sunlight is. A soil moisture sensor will be used to measure the moisture of the soil. If the moisture is low, the servo motor will rotate allowing the water to flow to the soil.

4. Bill of Materials

Budget Item	Quantity	Cost
DC Motor	2	15
Servo Motor	1	10
LDR	4	10
Breadboard	1	3
Capacitive Soil Moisture sensor	1	10
M3x10mm Screw	15	13
9V Batteries	1	5
Battery House	1	4.5
PLA Filament	1	30
H-Bridge	1	15
Total		**$115.5**

5. Project Construction

5.1. Materials needed

- 1.75mm PLA Filament
- Lead Free Solder
- 1x Arduino Mega 2560
- 1x SG90 Micro Servo Motor
- 1x L298N Motor Drive
- 1x Capacitive Soil Moisture Sensor
- 1x Breadboard
- 1x Battery House (4 AA Battery Holder)
- 1x 9V Batteries
- 2x DC motors
- 4x AA Batteries
- 4x IR sensor
- 4x LDR sensors
- 4x 1k ohm resistors
- 12x M3x10mm screw
- 10x wires

5.2. Tools needed

- FDM Printer

- Screwdriver
- Soldering Iron

5.3. 3D Prints needed

Note: The STL files that are in a blue font are an external STL files created by a 3rd party.

- 1x Lower_Housing.stl
- 1x Upper_Housing.stl
- 1x Pot.stl
- 1x Water_Reservoir.stl
- 1x Water_Gate.stl
- 2x MotorMount_Enclosire.stl
- 2x MotorMount.stl
- 2x Wheels_Enclosure_A.stl
- 2x Wheels_Enclosure_B.stl
- 2x Wheel_Drive_A.stl
- 2x Wheel_Drive_B.stl
- 4x Wheel_Slave.stl
- 4x Wheel_Enclosure_Connector.stl
- 44x Track.stl

Procedure

The following procedure was used to create the project.

1. The STL Files listed above were printed individually.
2. A wire was attached to each pin of the DC motor using the soldering iron and solder.
3. The DC motor was placed in the Motor Mount Enclosure and ensured the wires were coming out of the 10 mm opening.
4. The motor was then placed in the Motor Mount on top to secure the motor.
5. 2 wheels enclosure connectors were placed on the motor mount and motor mount enclosure to hold the parts together.
6. Wheel Drive A and Wheel Drive B were attached to the DC motor shaft by sliding them into the shaft.
7. A wheel slave was then placed on the motor mount enclosure and secured with a M3x10mm screw then another wheel slave was placed on the other side of the motor mount and secured with M3x10mm screw.
8. Wheels Enclosure B was then aligned with the opening on the motor mount enclosure and 2 M3x10mm screws were used to attach the parts together.
9. The 22 Tracks were attached to each other using the 1.75 mm PLA filament and then melting the tips to make sure the PLA does not slide off.
10. The tracks were then slid onto the wheel drive and wheel slave
11. Wheel Enclosure A was then placed on the Wheel Enclosure connector facing the Motor Mount and 4 M3x10mm screws were used to attach the parts together.
12. Steps 2-11 were done again using the second DC motor.
13. AutoDesk Inventor was then used to design the bottom section of the housing with the consideration of where the components will be placed.

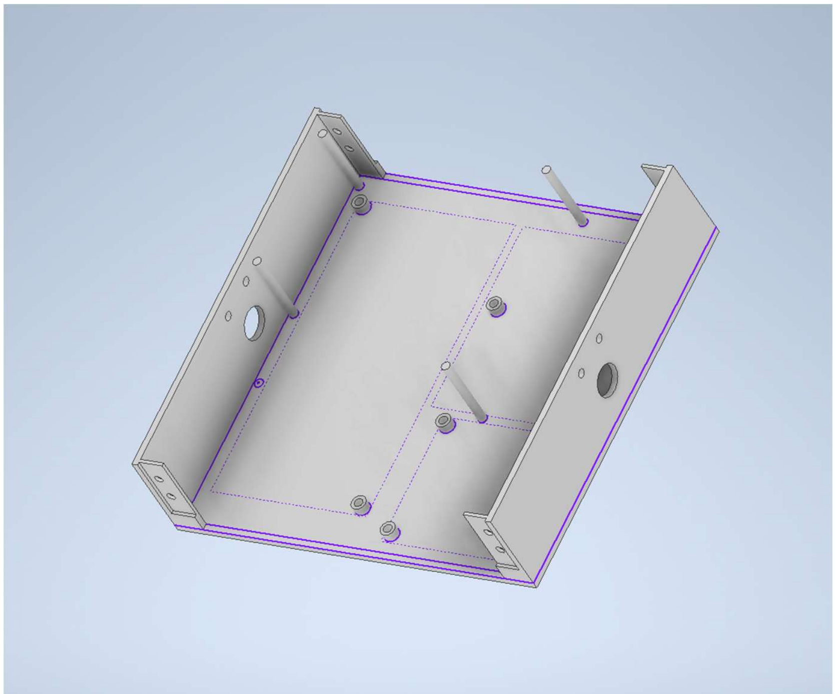


Figure 5: Lower section of the housing in AutoDesk Inventor

14. The ipt file was then exported as a STL file then sliced using Ultimaker Cura. The lower section of the housing was then printed using a 3D printer.

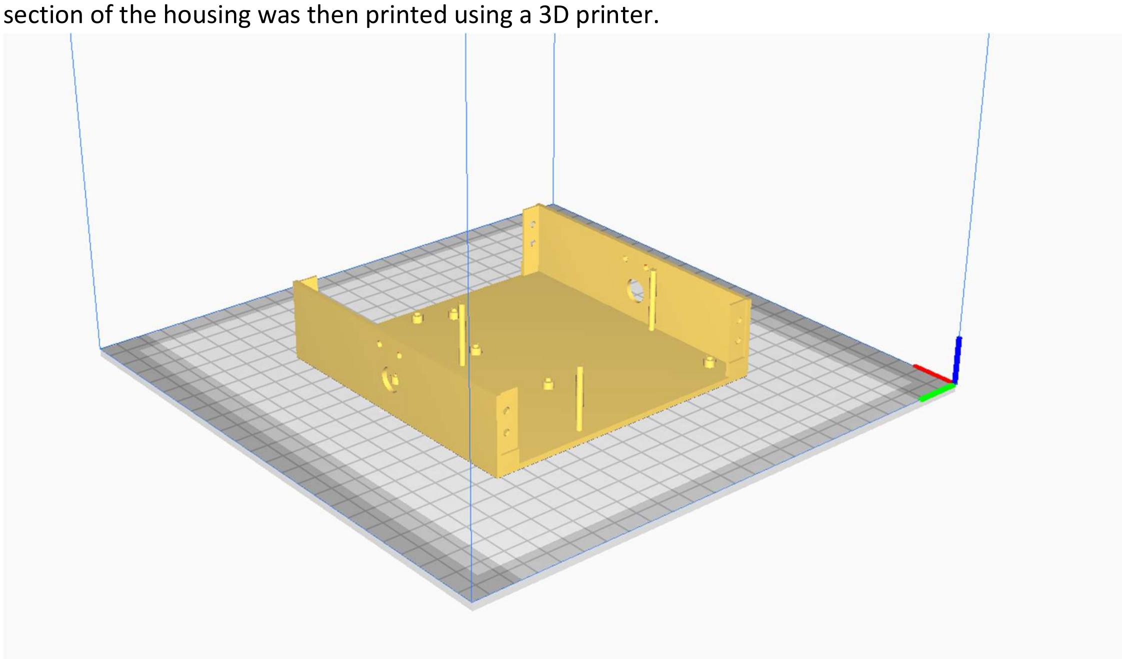


Figure 6: Ultimaker Cura Orientation for Lower housing

15. The motor was then attached to the Lower Housing using 2 M3x10mm screws.
16. The components were then placed onto the housing as shown in figure 8.

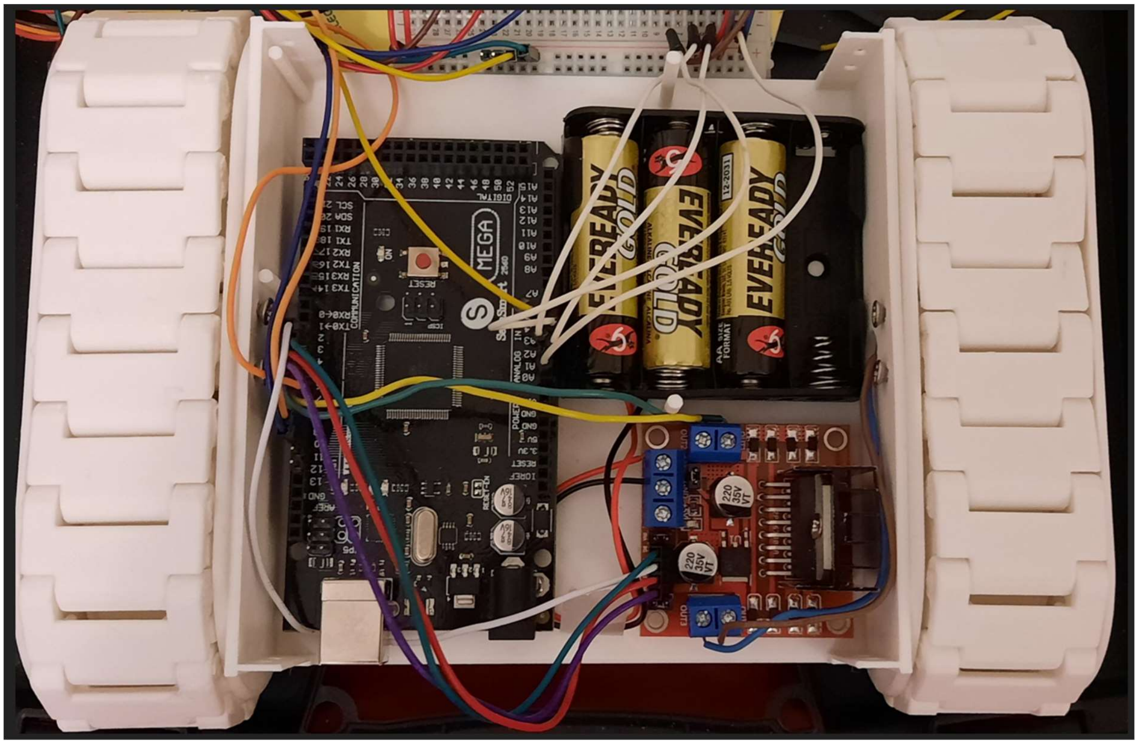


Figure 7: Components placement onto the housing

17. The circuitry was then connected using the components listed in Table 1 and was connected as shown in figure 3.

Table 1: Components needed for circuitry

2x DC Motor	1x Servo Motor
1x H-Bridge	1x Arduino Mega 2560
1x Moisture Sensor	4x IR Sensor
4x 1k ohm resistor	4x LDR Sensors

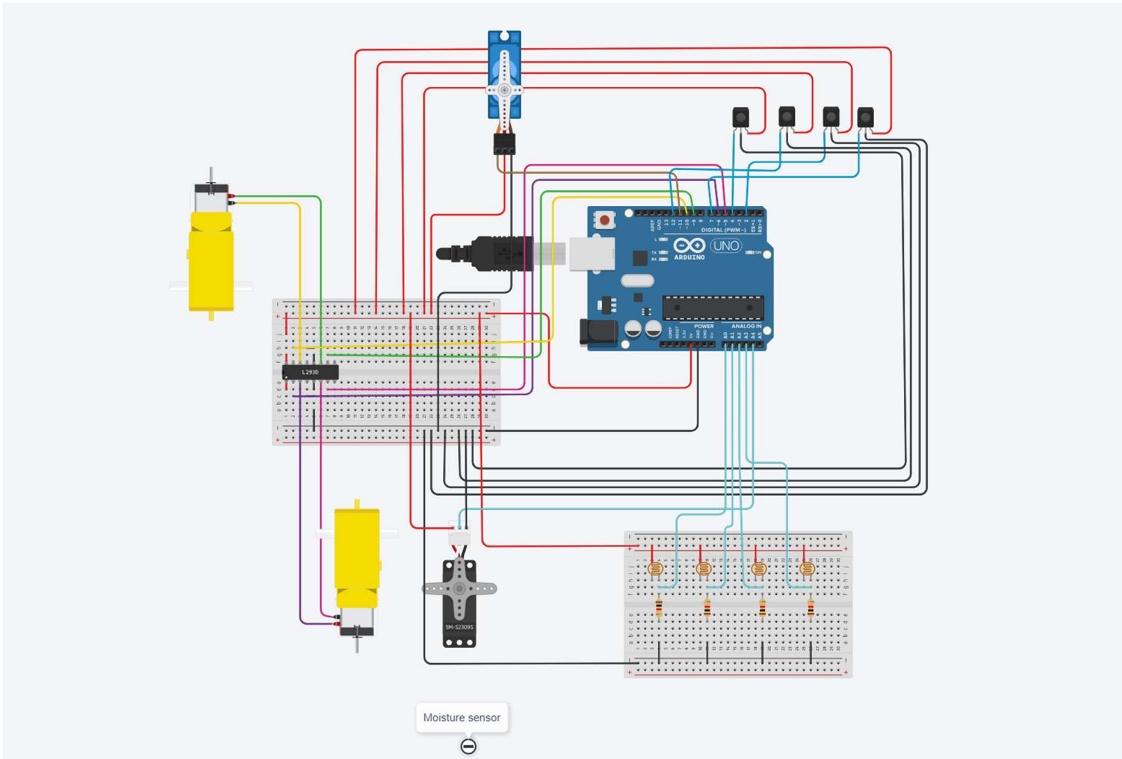


Figure 8: Circuitry connections (Note A Micro server was used in the diagram since a moisture sensor was not found in Tinker Cad)

18. The application Arduino was then used to program the Arduino and control the Automated plant pot. The code was then uploaded to the Arduino.
19. AutoDesk Inventor was then used to design the upper section of the housing with the consideration of where the pot will be placed.

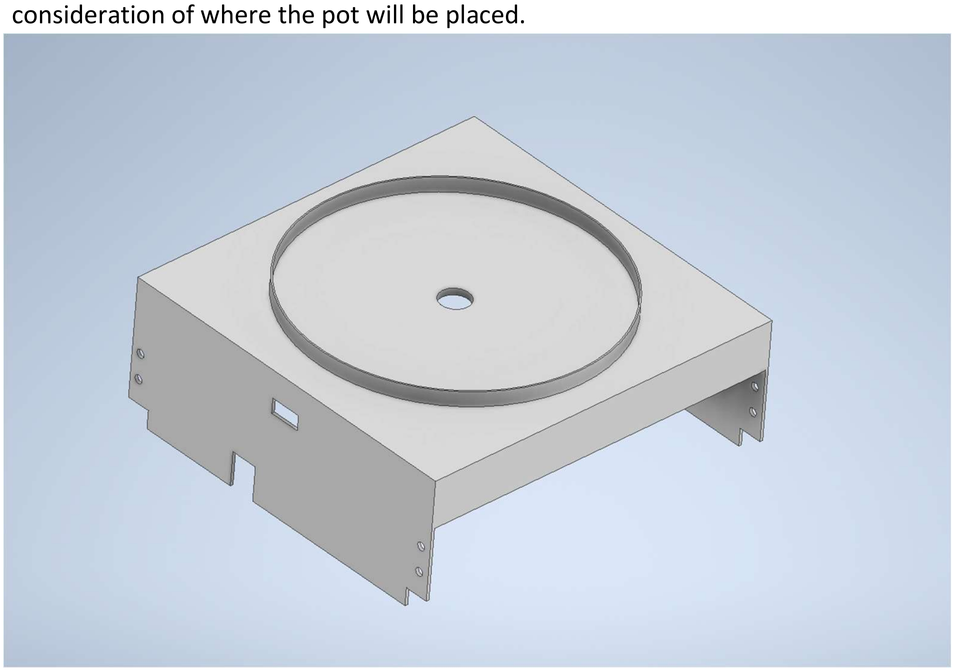


Figure 9: Upper section of Housing in AutoDesk Inventor

20. The ipt file was then exported as a STL file then sliced using Ultimaker Cura. The Upper section of the housing was then printed using a 3D printer.

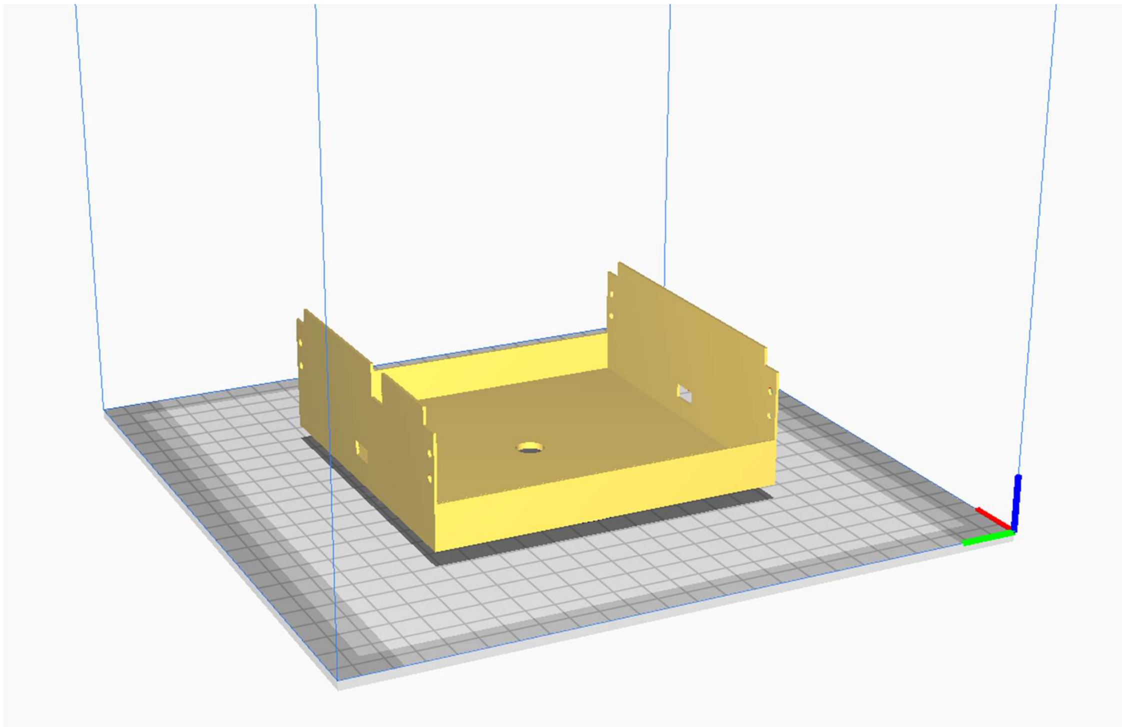


Figure 10: Ultimaker Cura Orientation for Upper housing

21. The IR sensors, Servo Motor, and Moisture sensor were then put through the rectangular openings on the side of the upper housing.
22. The LDR wires were then slid into the opening ontop of the housing then 8 M3x10mm screws were used to secure the Upper Housing to the Lower Housing.
23. AutoDesk Inventor was then used to design the pot with the consideration of where the Water reservoir will be placed.

Figure 11: Pot in AutoDesk Inventor

24. The ipt file was then exported as a STL file then sliced using Ultimaker Cura. The Pot was then printed using a 3D printer.

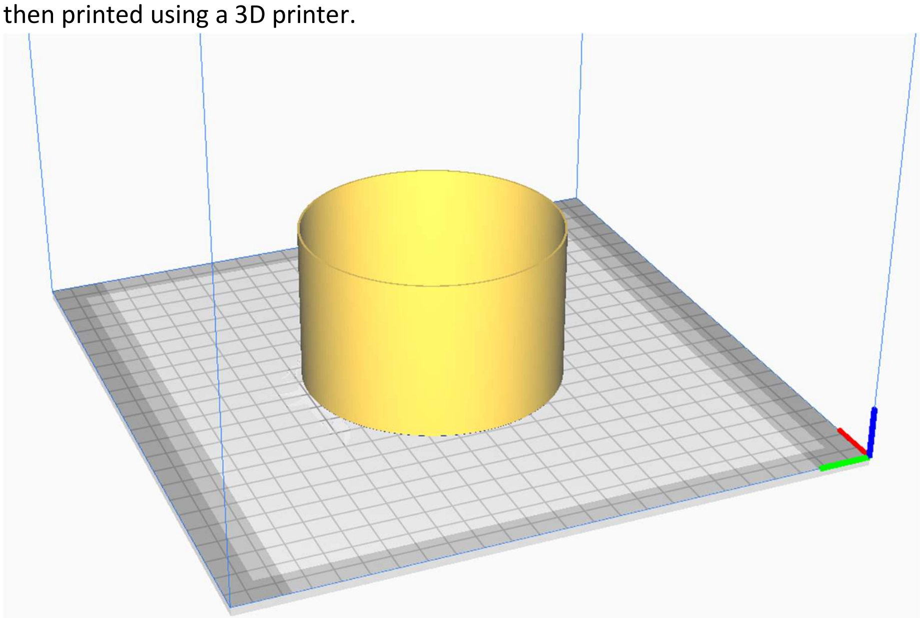


Figure 12:Ultimaker Cura Orientation for Pot

25. The Pot was placed on top of the Upper housing. The LDR sensors were then moved so that each one is facing either north, south, east, or west.

26. AutoDesk Inventor was then used to design the Water Reservoir with the consideration of how it will be placed on top of the Pot.

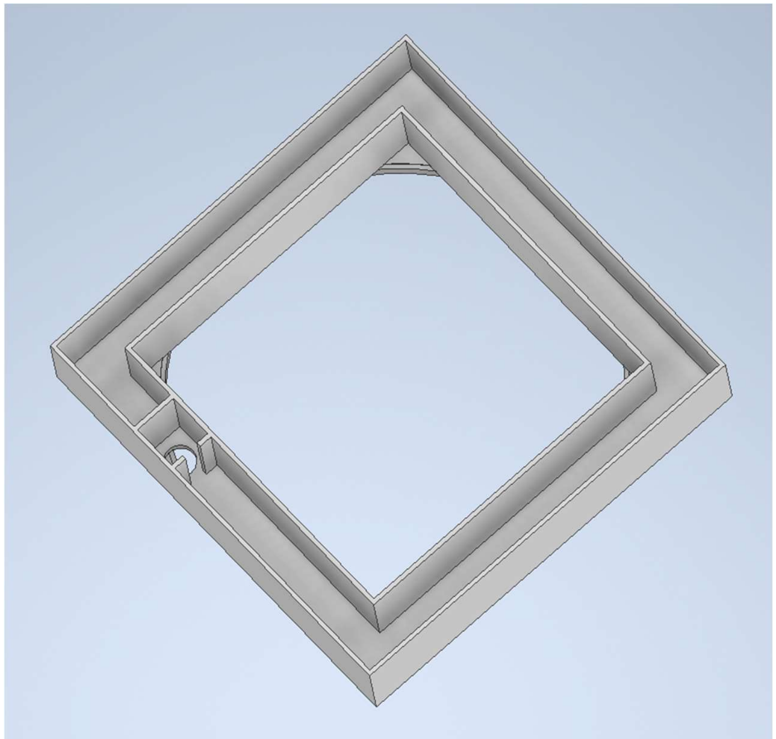


Figure 13: Water Reservoir in AutoDesk Inventor

27. The ipt file was then exported as STL file then sliced using Ultimaker Cura. The Water Reservoir was then printed using a 3D printer.

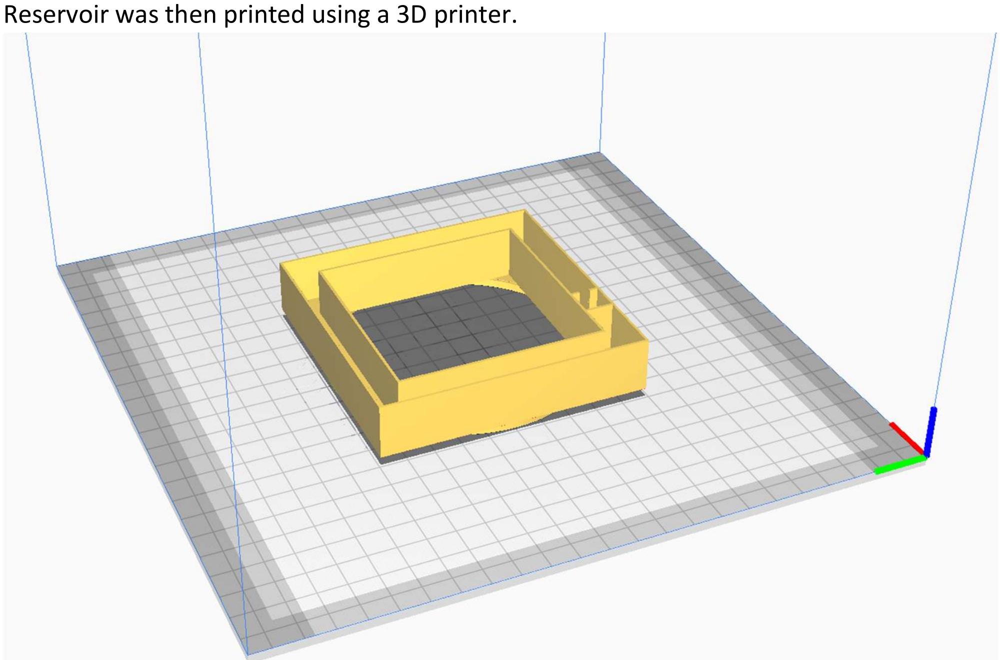


Figure 14: Ultimaker Cura Orientation for Water Reservoir

28. Once the Water Reservoir was printed, It was placed on top of the pot and secured so that the water does not spill.

29. AutoDesk Inventor was then used to design the Water Gate with the consideration of how it will be placed in the Water Reservoir.

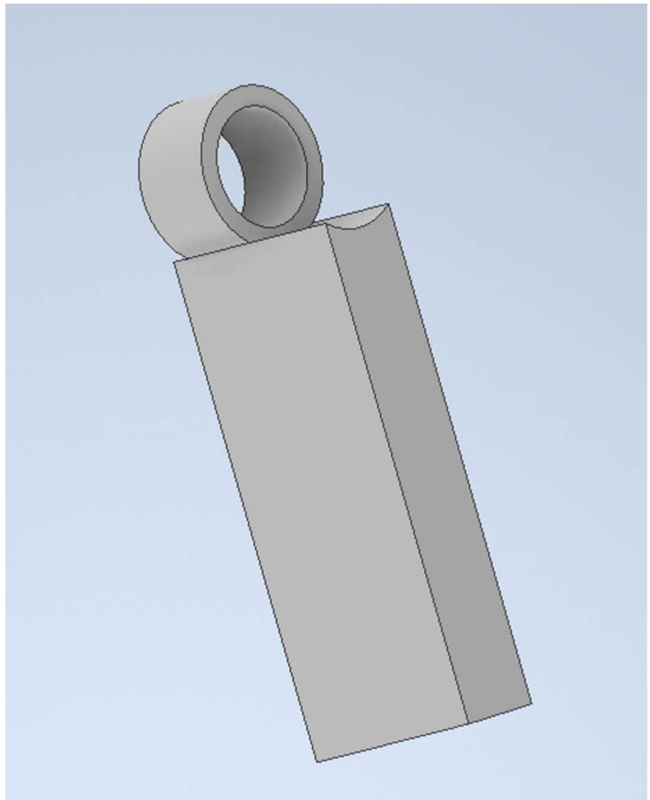


Figure 15: Water Gate in AutoDesk Inventor

30. The ipt file was then exported as STL file then sliced using Ultimaker Cura. The Water Gate was then printed using a 3D printer.

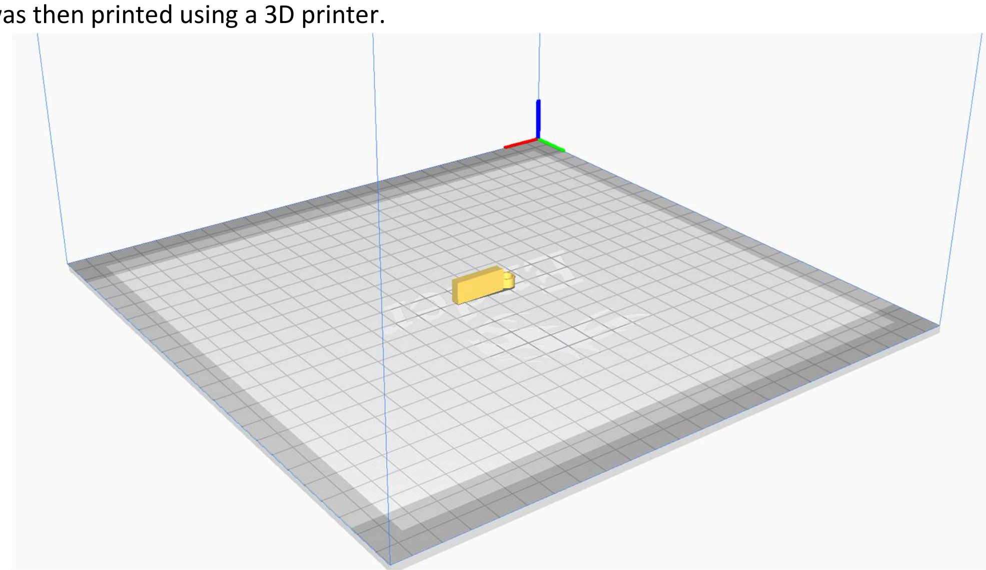


Figure 16: Ultimaker Cura Orientation for Water Gate

31. Once the water gate was printed, it was attached to the servo motor. The servo motor was then Glued to the Water reservoir in an angle that the gate can move in.

32. The moisture sensor was then placed into the soil.

6. Testing and Observations

6.1. Testing

4 series of testing were done to ensure that the product is fully working for demonstration.

One test was the Circuitry and code testing using TinkerCAD. The Circuitry was created on TinkerCAD as well as the code. Once all the components were added to TinkerCAD and code was created, a simulation mode was then selected to test the code and ensure that the circuitry will work as needed.

Another test was done when the project was fully assembled. Once the code was uploaded, a source of light was put stationary and the pot moved towards the light as intended. This test was done for all 4 LDR's.

Another test was done to ensure that the IR sensor is working. Once the pot started moving, an object was placed in front of one IR sensor to ensure that the pot does stops and does not keep driving towards an object.

The last test that was done was testing the water gate. The moisture sensor was placed in a dehydrated environment where the humidity was low. The humidity was decreased to ensure that the servo motor of the water gate moves so that it waters the soil.

6.2. Observations

The motors used for this project are DC motors used for RC cars which means that it did not have enough power to move the pot with the soil and plants on it at the expected speed. Since the weight of the Pot increased with the soil, water and plants that would require a motor with higher torque to be able to move the pot.

LDR (Light detecting resistors) sensors were used to detect sunlight, however LDR sensors detect light rather than sunlight which resulted the pot to move towards any source of light rather than sunlight.

7. Project Outcome

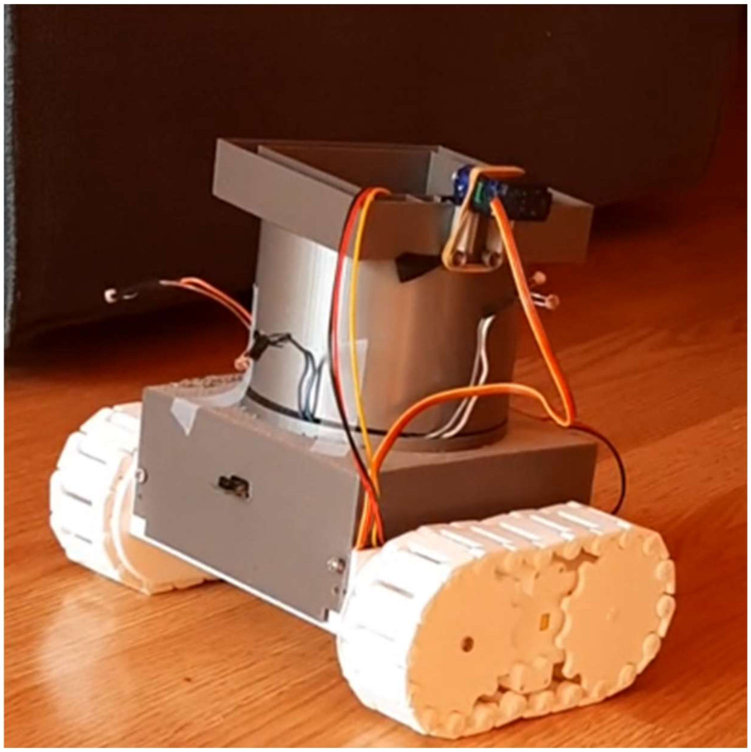


Figure 17: Automated Plant Pot Fully Assembled

The capstone project was fully completed on time and demonstrated on time. The automated plant pot follows the light and waters the plant when the moisture sensor's value decreases below 50%. The code compares all values of LDR sensors to decided which direction the robot will move in. The code also checks the IR sensor values to ensure that there are no objects in front of the Pot.

8. Timeline and Milestones

The making of this project went smoothly from the start. A Gantt chart was created to stay on track and help set milestones for important tasks to finish the capstone project in an orderly and time efficient manner.

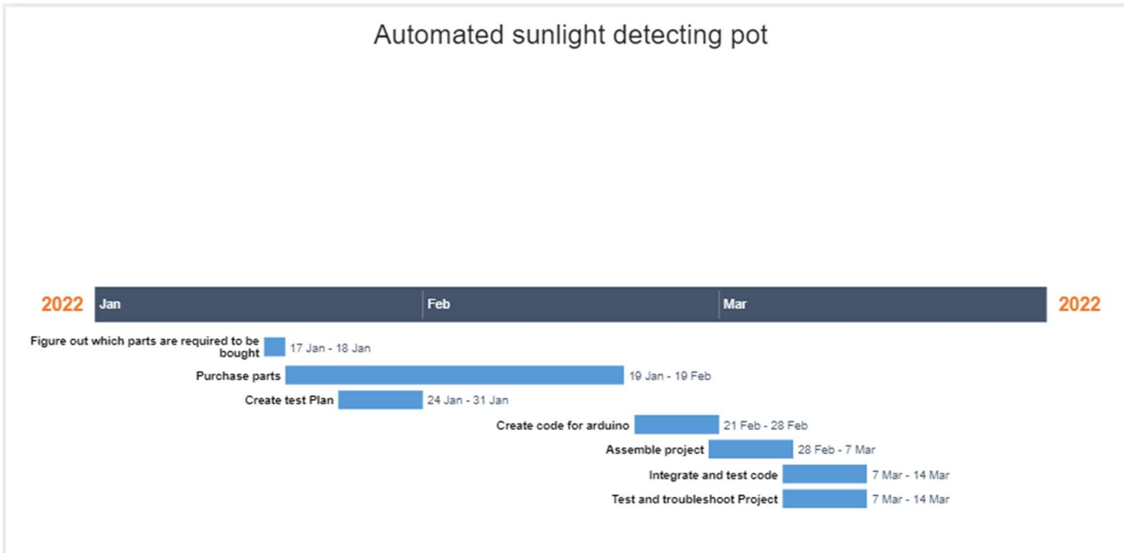


Figure 18: Gantt Chart of Expected Timeline

- Proposal: Ahead of Schedule
- Purchase Items: Ahead of Schedule
- Make Schematic: Ahead of Schedule
- Circuit Assembly: Ahead of Schedule
- Arduino Script: Ahead of Schedule
- Test Script: Behind Schedule, Completed on 07/03/2022
- Design the pot: Behind Schedule, Completed on 10/03/2022
- Print the pot: On Schedule
- Assemble Project: Behind Schedule, Completed on 22/03/2022
- Test Project in real life setting: Behind Schedule, Completed on 22/03/2022
- Project Demonstration: On Schedule
- Final Report: On Schedule

The project was mostly ahead of schedule excluding a couple of milestones. Designing the pot was a bit behind schedule due to trying to optimize the pots design. Multiple prints failed or did not meet the requirements.

The key dates for accomplishing my project were

1-**Planning**. After figuring out what parts were needed, purchasing them was the next step. Purchasing parts was a Milestone since the right parts had to be chosen before purchasing them for the project to work

2- **Coding.** Creating the code is a milestone since the logic is needed to make the Automated plant pot work.

3- **3d Designing** project was also a milestone for the capstone project. Creating a design that fit all the parts neatly while maintaining optimal operation was a major focus before moving on to the next step.

4- **Assembling and testing** the project was the last milestone for the capstone project. After creating a sufficient design, assembling the automated pot was the final step in the process before testing. Making sure each component fit neatly without interfering with each other before closing the component housing was key to avoid potential breakages.

9. Conclusion

The Automated plant pot fully functions as proposed. A flow chart was created to plan out how the logic will look like and how the automated pot will operate. Testing was made to ensure that the pot functions as proposed and all the parts work. Project management skills were used to meet the deadlines and well as plan out how the project will be constructed. Autodesk Inventor was also used to create the structure of the pot to store the components as well as function as required. Knowledge applied in this project are from courses learnt throughout the years at mohawk. The course Applied Programming helped program the Arduino to function as needed. The course Additive manufacturing helped design the housing, pot and water reservoir.

10. Future Work/Recommendation

Currently the pot can hold water that will last for 3 days, an improvement that can be added would be different sizes of water reservoirs to fit the needs of the user.

One other improvement that can be added would be a bigger base that can hold multiple pots. Only one pot can be placed on the current setup which is not optimal since people tend to have more than one plant in their houses. Having a bigger base would involve using stronger motors and a bigger printer bed to print parts.

Another improvement is adding idle time. Since there will be no sun after sunset the pot should stay idle and not move till sunrise. The user will be able to set the idle time since sunset and sunrise time differs depending on the season.

LDR sensors can be replaced with solar panels, which would be more optimal since Solar panels would detect sunlight rather than just light. Solar panels would also be used to set an idle time. If after 30 mins there was no sun, then Pot would go into idle mode to save the batteries.

11. References

A. Azouz, “Arduino Sumo Robot,” *Instructables*, 24-Apr-2019. [Online]. Available: https://www.instructables.com/How-to-Make-Arduino-Sumo-Robot/ . [Accessed: 05-Feb-2022].

B. Taibo, N., 2022. *Modular Rover Tracks by nahueltaibo*. [Online] Thingiverse.com. Available at: https://www.thingiverse.com/thing:2804287 [Accessed 5 February 2022].

C. “Datasheet - components101.com,” *Light Detecting Resistor*. [Online]. Available: https://components101.com/sites/default/files/component_datasheet/LDR%20Datasheet.pdf [Accessed: 05-Feb-2022].

D. “L298N motor driver,” *L298N Dual H-Bridge Motor Driver*. [Online]. Available: http://www.handsontec.com/dataspecs/L298N%20Motor%20Driver.pdf . [Accessed: 06-Feb-2022].

E. “Arduino® MEGA 2560 REV3 - docs.arduino.cc.” [Online]. Available: https://docs.arduino.cc/static/a5318e27baa5c7076e37e01d8a03072a/A000067-datasheet.pdf . [Accessed: 06-Feb-2022].

12. Appendix

12.1 Appendix A – Code used for Automated Plant pot

```
#include <Servo.h>

const int AirValue = 600;
const int WaterValue = 350;
int soilmoisturepercent=0;

const int IN1=3; //Input 2 on HBridge
const int IN2=4; //Input 1 on HBridge
const int IN3=5; //Input 3 on HBridge
const int IN4=6; //Input 4 on HBridge

const int IN5=7; //Signal on Servo Motor
Servo waterGate;

const int IR_sensor_front=10; //Output of IR 1
const int IR_sensor_back=12; //Output of IR 3


#define LDR_sensor_front A0
#define LDR_sensor_left A1
#define LDR_sensor_back A2
#define LDR_sensor_right A3

#define soilMoistureValue A4

void setup(){
```

```
 Serial.begin(9600);
 waterGate.attach(IN5);
 //pinMode

}

void loop(){

  int moisture_value = analogRead(soilMoistureValue);

  int LDR_front = analogRead(LDR_sensor_front);
  int LDR_back = analogRead(LDR_sensor_back);
  int LDR_left = analogRead(LDR_sensor_left);
  int LDR_right = analogRead(LDR_sensor_right);

  int IR_front = digitalRead(IR_sensor_front);
  int IR_back = digitalRead(IR_sensor_back);

  soilmoisturepercent = map(moisture_value, AirValue, WaterValue, 0, 100);

  if(soilmoisturepercent <= 50){

    waterGate.write(90);
    delay(1000);
    waterGate.write(0);

  }

  if (LDR_front > LDR_back && LDR_front > LDR_left && LDR_front > LDR_right){

    check_IR(255); // add function
```

```
  delay(50);
  forward(255);
  delay(500);
  Serial.print("FRONT");

}

else if (LDR_back > LDR_front && LDR_back > LDR_left && LDR_back > LDR_right){

  check_IR(255); // add function
  delay(50);
  backward(255);
  delay(500);
  Serial.print("BACK");

}

else if (LDR_right > LDR_front && LDR_right > LDR_left && LDR_right > LDR_back){

  check_IR(255); // add function
  delay(50);
  rotate_r(255);
  delay(500);
  Serial.print("RIGHT");

}

else if (LDR_left > LDR_right && LDR_left > LDR_back && LDR_left > LDR_front){

  check_IR(255); // add function
  delay(50);
```

```
  rotate_l(255);
  delay(500);
  Serial.print("LEFT");

 }

 // testing code
 Serial.print("LDR front: ");
 Serial.println(LDR_front);
 Serial.print("LDR_back: ");
 Serial.println(LDR_back);
 Serial.print("LDR_left: ");
 Serial.println(LDR_left);
 Serial.print("LDR_right: ");
 Serial.println(LDR_right);
 Serial.print("IR_front: ");
 Serial.println(IR_front);
 Serial.print("IR_back: ");
 Serial.println(IR_back);
 Serial.print("moisture percentage:  ");
 Serial.println(soilmoisturepercent);
}



void Stop(){
 Serial.begin(9600);//When we want to  Motor To stop ,
 // just void this part on the loop section .
 analogWrite(IN1,0);
 analogWrite(IN2,0);
 analogWrite(IN3,0);
 analogWrite(IN4,0);
```

```
}

void rotate_r (int Speed)
{
  //When we want to let Motor To Rotate ,
  // just void this part on the loop section .
  analogWrite(IN1,0);
  analogWrite(IN2,Speed);
  analogWrite(IN3,Speed);
  analogWrite(IN4,0);
}

void rotate_l (int Speed)
{
  //When we want to let Motor To Rotate ,
  // just void this part on the loop section .
  analogWrite(IN1,Speed);
  analogWrite(IN2,0);
  analogWrite(IN3,0);
  analogWrite(IN4,Speed);
}

void backward (int Speed){
  //When we want to let Motor To move forward,
  // just void this part on the loop section .
  analogWrite(IN1,Speed);
  analogWrite(IN2,0);
  analogWrite(IN3,Speed);
  analogWrite(IN4,0);
}
```

```
void forward (int Speed){
  //When we want to let Motor To move forward,
  // just void this part on the loop section .
  analogWrite(IN1,0);
  analogWrite(IN2,Speed);
  analogWrite(IN3,0);
  analogWrite(IN4,Speed);
}

void check_IR(int speed){

  int IR_front = digitalRead(IR_sensor_front);
  int IR_back = digitalRead(IR_sensor_back);

  if (IR_front == 1){

    backward(speed);

  }

  else if(IR_back == 1){

    forward(speed);

  }
}
```

12.2 Appendix B – Schematic Diagram for Automated Plant Pot

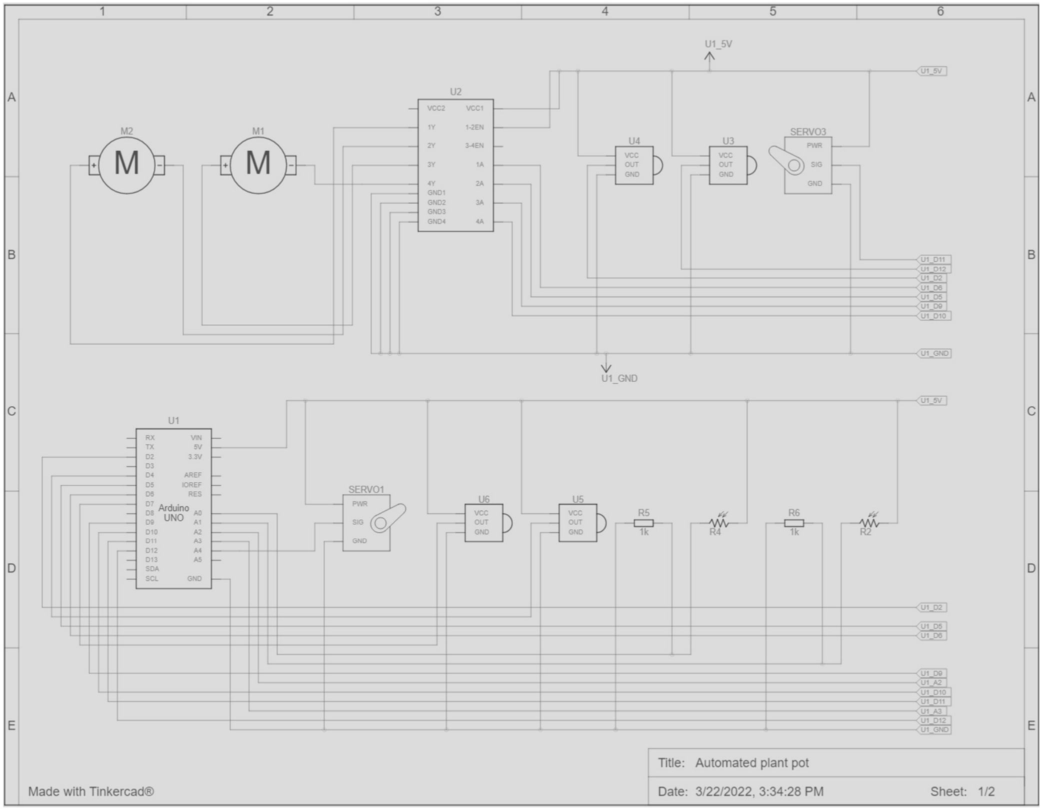


Figure 19: Page 1 schematic for Automated plant pot

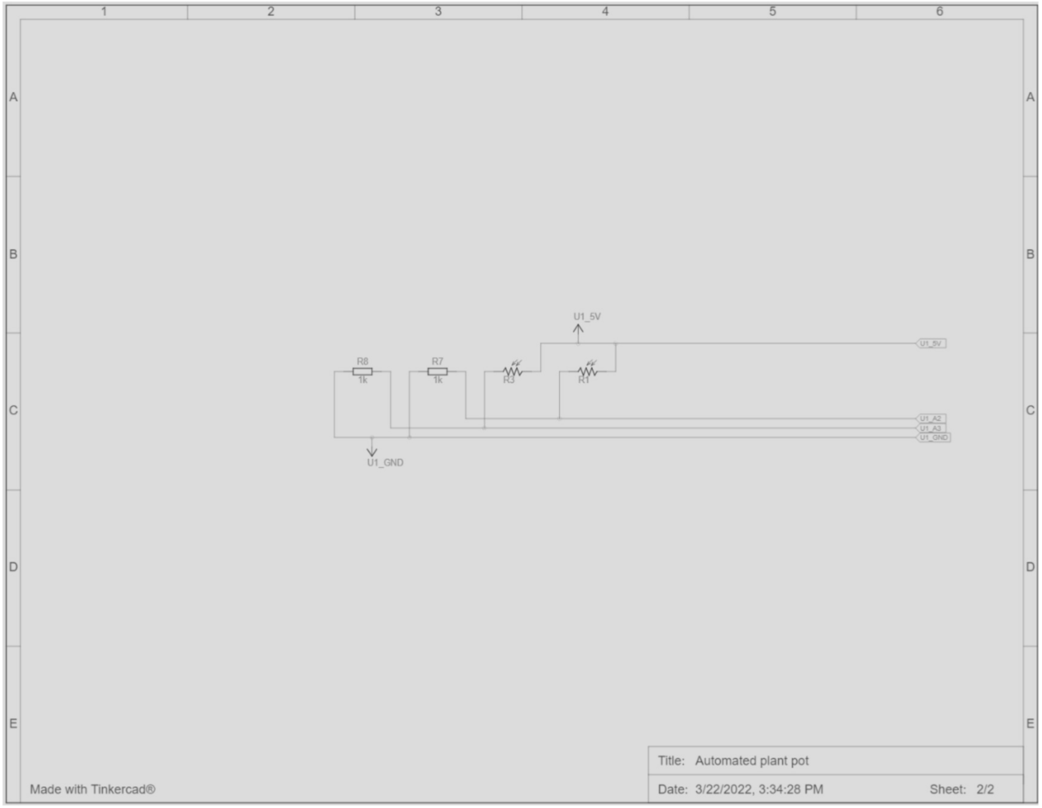
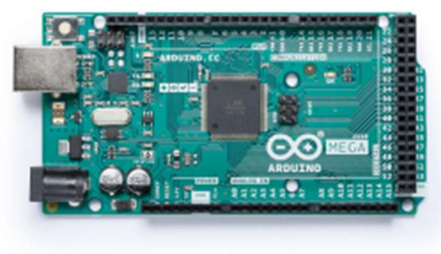


Figure 20: Page 2 schematic for Automated plant pot

12.3 Appendix C – Datasheet Screenshots

Arduino® MEGA 2560 Rev3

Product Reference Manual
SKU: A000067



Description

Arduino® Mega 2560 is an exemplary development board dedicated for building extensive applications as compared to other maker boards by Arduino. The board accommodates the ATmega2560 microcontroller, which operates at a frequency of 16 MHz. The board contains 54 digital input/output pins, 16 analog inputs, 4 UARTs (hardware serial ports), a USB connection, a power jack, an ICSP header, and a reset button.

Target Areas

3D Printing, Robotics, Maker

1 / 18 Arduino® MEGA 2560 Rev3 Modified: 23/03/2022

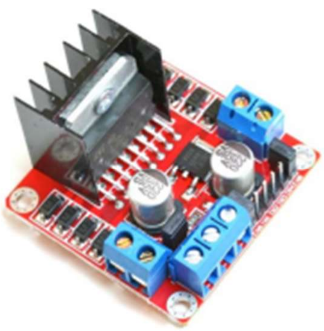
Figure 21: Arduino Mega 2560 Rev3 Datasheet

Handson Technology

User Guide

L298N Dual H-Bridge Motor Driver

This dual bidirectional motor driver, is based on the very popular L298 Dual H-Bridge Motor Driver Integrated Circuit. The circuit will allow you to easily and independently control two motors of up to 2A each in both directions.It is ideal for robotic applications and well suited for connection to a microcontroller requiring just a couple of control lines per motor. It can also be interfaced with simple manual switches, TTL logic gates, relays, etc. This board equipped with power LED indicators, on-board +5V regulator and protection diodes.



SKU: MDU-1049

Brief Data:

- Input Voltage: 3.2V~40Vdc.
- Driver: L298N Dual H Bridge DC Motor Driver
- Power Supply: DC 5 V - 35 V
- Peak current: 2 Amp
- Operating current range: 0 ~ 36mA
- Control signal input voltage range :
- Low: -0.3V ≤ Vin ≤ 1.5V.
- High: 2.3V ≤ Vin ≤ Vss.
- Enable signal input voltage range :
    - Low: -0.3 ≤ Vin ≤ 1.5V (control signal is invalid).
    - High: 2.3V ≤ Vin ≤ Vss (control signal active).
- Maximum power consumption: 20W (when the temperature T = 75 ℃).
- Storage temperature: -25 ℃ ~ +130 ℃.
- On-board +5V regulated Output supply (supply to controller board i.e. Arduino).
- Size: 3.4cm x 4.3cm x 2.7cm

1 www.handsontec.com

Figure 22: L298N H-Bridge Datasheet

Data pack F *Issued March 1997* **232-3816**

Data Sheet

Light dependent resistors

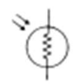
NORP12 RS stock number 651-507
NSL19-M51 RS stock number 596-141

Two cadmium sulphide (cdS) photoconductive cells with spectral responses similar to that of the human eye. The cell resistance falls with increasing light intensity. Applications include smoke detection, automatic lighting control, batch counting and burglar alarm systems.

Guide to source illuminations

Light source	Illumination (Lux)
Moonlight	0.1
60W bulb at 1m	50
1W MES bulb at 0.1m	100
Fluorescent lighting	500
Bright sunlight	30,000

Circuit symbol



Light memory characteristics

Light dependent resistors have a particular property in that they remember the lighting conditions in which they have been stored. This memory effect can be minimised by storing the LDRs in light prior to use. Light storage reduces equilibrium time to reach steady resistance values.

NORP12 (RS stock no. 651-507)

Absolute maximum ratings

Voltage, ac or dc peak	320V
Current	75mA
Power dissipation at 30°C	250mW
Operating temperature range	-60°C to +75°C

Electrical characteristics

T_A = 25°C. 2854°K tungsten light source

Parameter	Conditions	Min.	Typ.	Max.	Units
Cell resistance	1000 lux 10 lux	- -	400 9	- -	Ω kΩ
Dark resistance	-	1.0	-	-	MΩ
Dark capacitance	-	-	3.5	-	pF
Rise time 1	1000 lux 10 lux	- -	2.8 18	- -	ms ms
Fall time 2	1000 lux 10 lux	- -	48 120	- -	ms ms

1. Dark to 110% R_L
2. To 10 × R_L
R_L = photocell resistance under given illumination.

Features

- Wide spectral response
- Low cost
- Wide ambient temperature range.

Dimensions

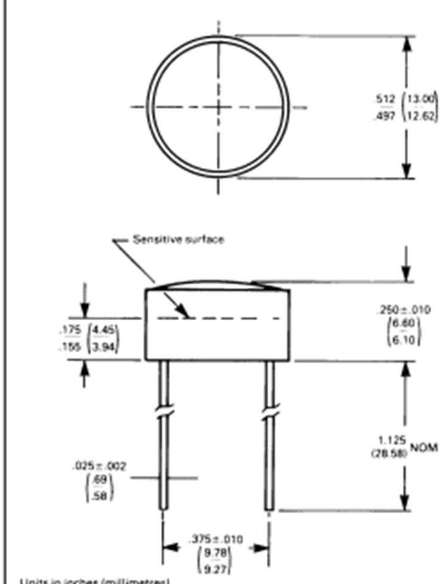


Figure 23: Light Dependant Resistor Datasheet

New Product

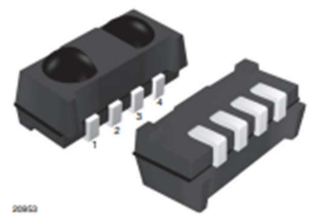
VISHAY.

www.vishay.com

TSSP77038

Vishay Semiconductors

IR Receiver Modules for Remote Control Systems



MECHANICAL DATA

Pinning:

1, 4 = GND, 2 = V_S, 3 = OUT

FEATURES

- Very low supply current
- Photo detector and preamplifier in one package
- Internal filter for 38 kHz IR signals
- Supply voltage: 2.5 V to 5.5 V
- Improved immunity against ambient light
- Capable of side or top view
- Insensitive to supply voltage ripple and noise
- Two lenses for high sensitivity and wide receiving angle
- Narrow optical filter to reduce interference from plasma TV emissions
- Material categorization: For definitions of compliance please see www.vishay.com/doc?99912

Pb-free e3 RoHS COMPLIANT GREEN (5-2008)

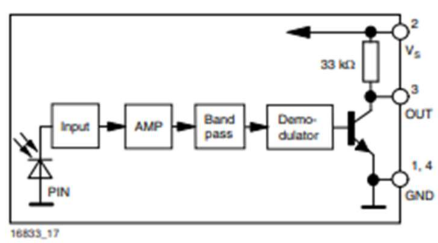
DESCRIPTION

The TSSP77038 is a compact two lens SMD IR receiver for sensor applications. It has a high gain for IR signals at 38 kHz. The detection level does not change when ambient light or strong IR signals are applied. It can receive continuous 38 kHz signals or 38 kHz bursts.

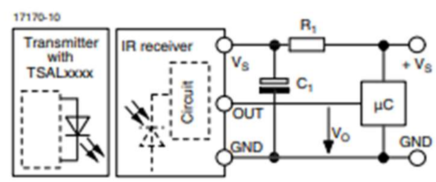
This component has not been qualified according to automotive specifications.

PARTS TABLE	
CARRIER FREQUENCY	SENSOR APPLICATIONS
38 kHz	TSSP77038

BLOCK DIAGRAM



APPLICATION CIRCUIT



The external components R_1 and C_1 are optional to improve the robustness against electrical overstress (typical values are $R_1 = 100\ \Omega$, $C_1 = 0.1\ \mu F$).

Rev. 1.0, 09-Mar-12 1 Document Number: 82470

THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE. THE PRODUCTS DESCRIBED HEREIN AND THIS DOCUMENT ARE SUBJECT TO SPECIFIC DISCLAIMERS, SET FORTH AT www.vishay.com/doc?91000

Figure 24: IR Sensor Datasheet