Section: 0302

Team: HydroQuack

Assignment: MS9

Contributions

Kemisola Benson: Working on pump, Organization and Labeling for Design Brief 1
Matthew Frenia: Coding, Navigation
Haley Smith: MS9, project management, Design Brief 6 and 7
Charlie Jeckell: Machine learning, Depth sensor, Color Sensor, Mission Brief
Anunithaa Rajakumaresan: Propulsion Modeling
David Li: CAD, Coding, design brief 5, providing models and exploded view for anyone who needed, project management. Helping on sustainability brief 8. Few revisions and additions to ms8
Allison Hiller: electronics calculations, MS8 Slides, Design Brief 1, project management, Labeling and editing
Julia Tisaranni: Electronics, machine learning

Milestone 9: Final Design Briefs

Water Sampling Mission - Team HydroQuack

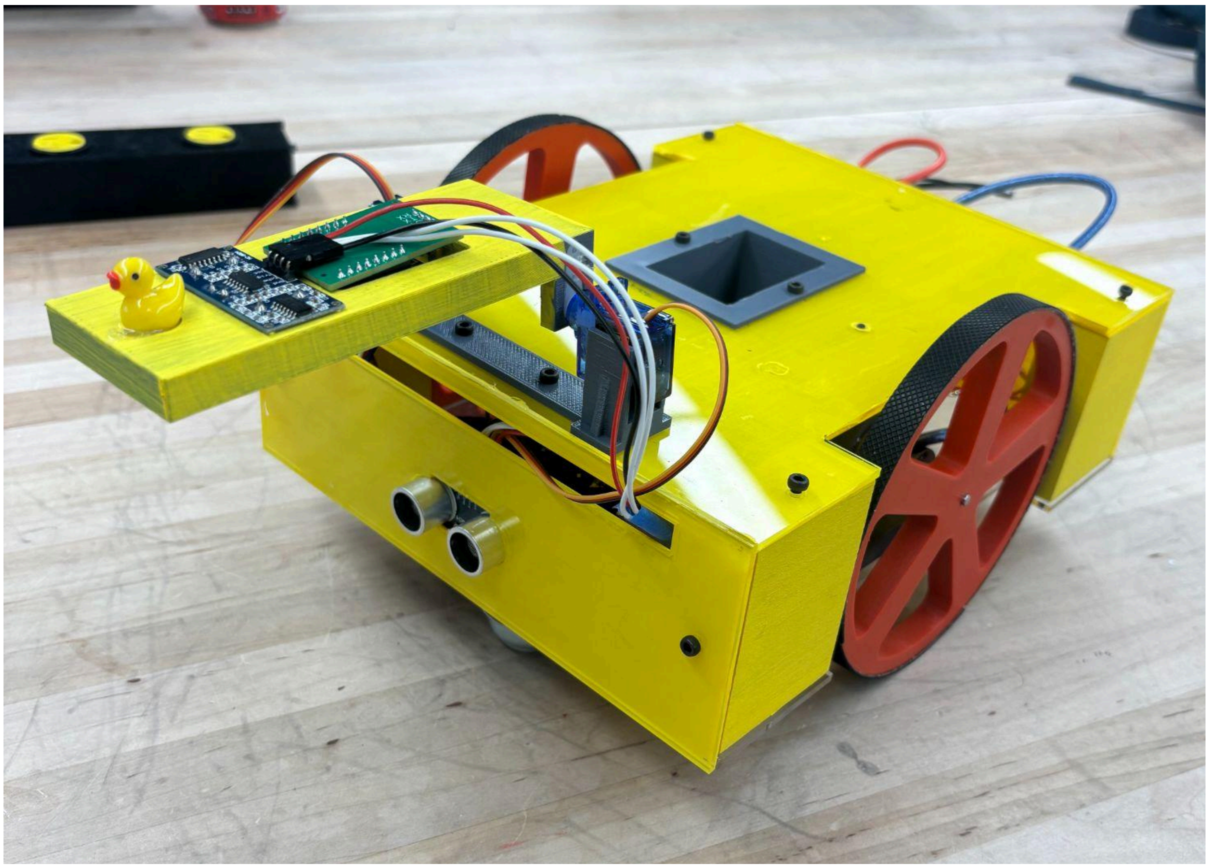


Table of Contents

Constructing a Prototype:

Manufacturing:

Laser Cut	Front chassis wall, back chassis wall, top chassis, and bottom chassis
3-D print	Water reservoir, water reservoir attachment, wheels, motor casing, servo bracket, arm
Wood shop	4 (⅞ in. deep x 2 ⅞ in. high x 2 in. wide) wooden posts 1 (⅝ in. height x 1 ¼ in. deep x 2 ⅛ in. wide) wooden block

Table 1

After the selected laser cut parts and the 3-D parts referenced in Table 1 were produced, the wood posts were created in the wood shop. A planer and table saw to create 4 (⅞ in. deep x 2 ⅞ in. high x 2 in. wide) wooden posts for OTV corners. A drill press was also used to create holes for the top, bottom, and side of the wooden blocks. The drawing below shows the position of the holes in mm.

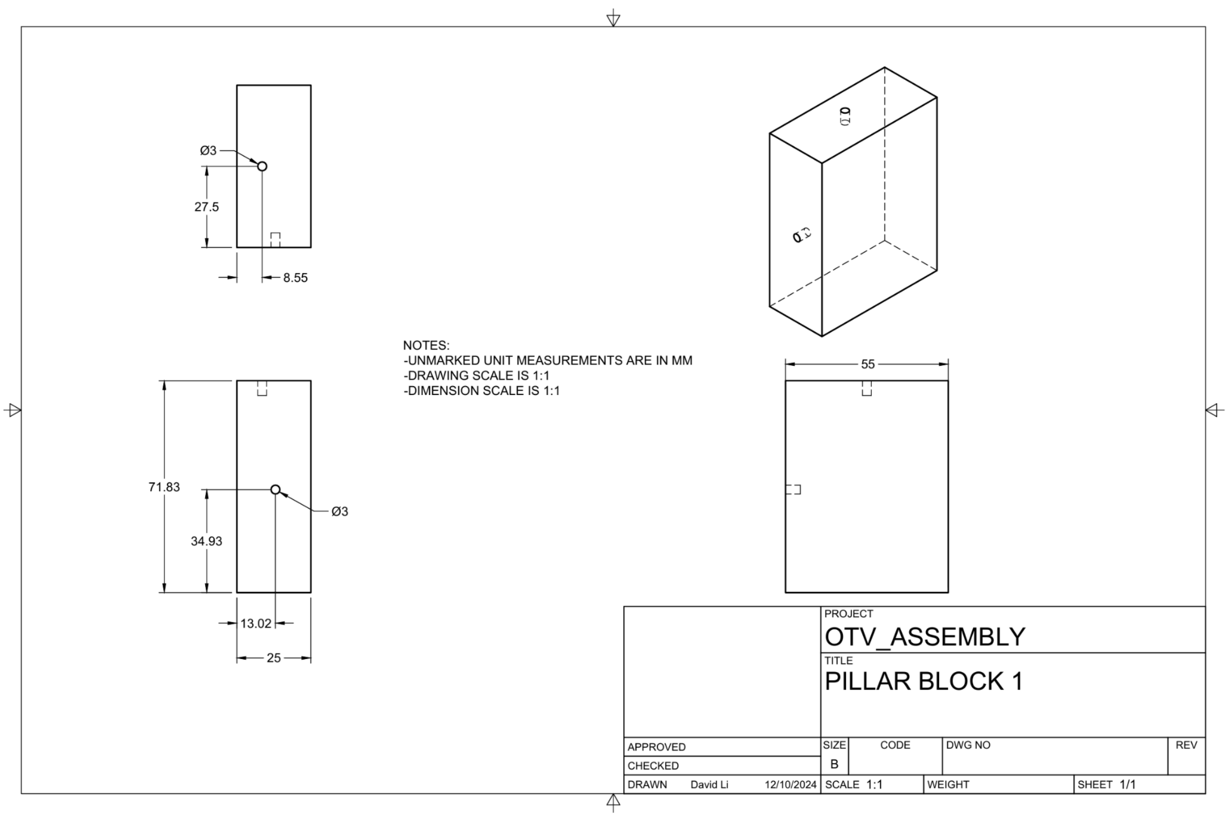


Figure 1: Dimensions of the wooden blocks

Additionally, the bandsaw was used to cut 1 (⅝ in. height x 1 ¼ in. deep x 2 ⅛ in. wide) piece of wood for rollerball mount. The roller ball was then drilled on top of the small piece of wood.

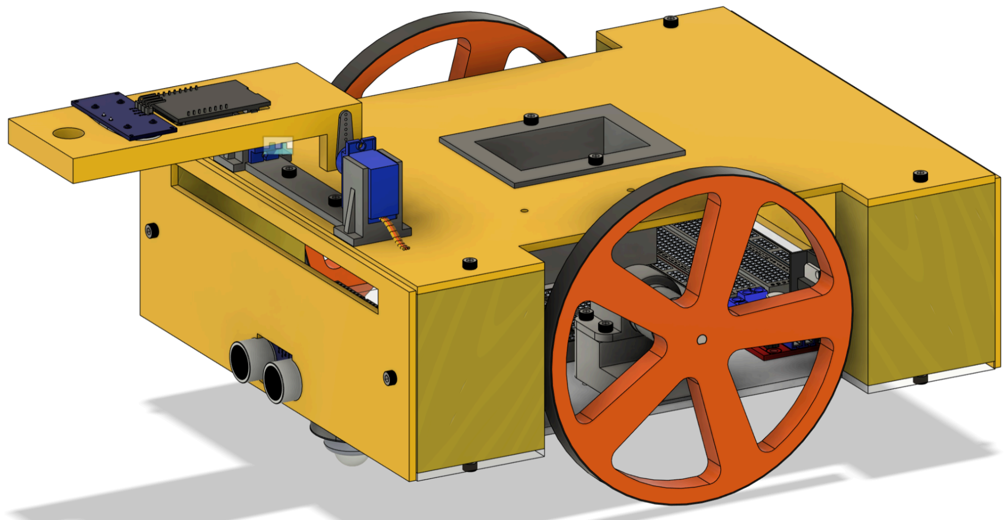


Figure 2: Final full assembly with lid fastened

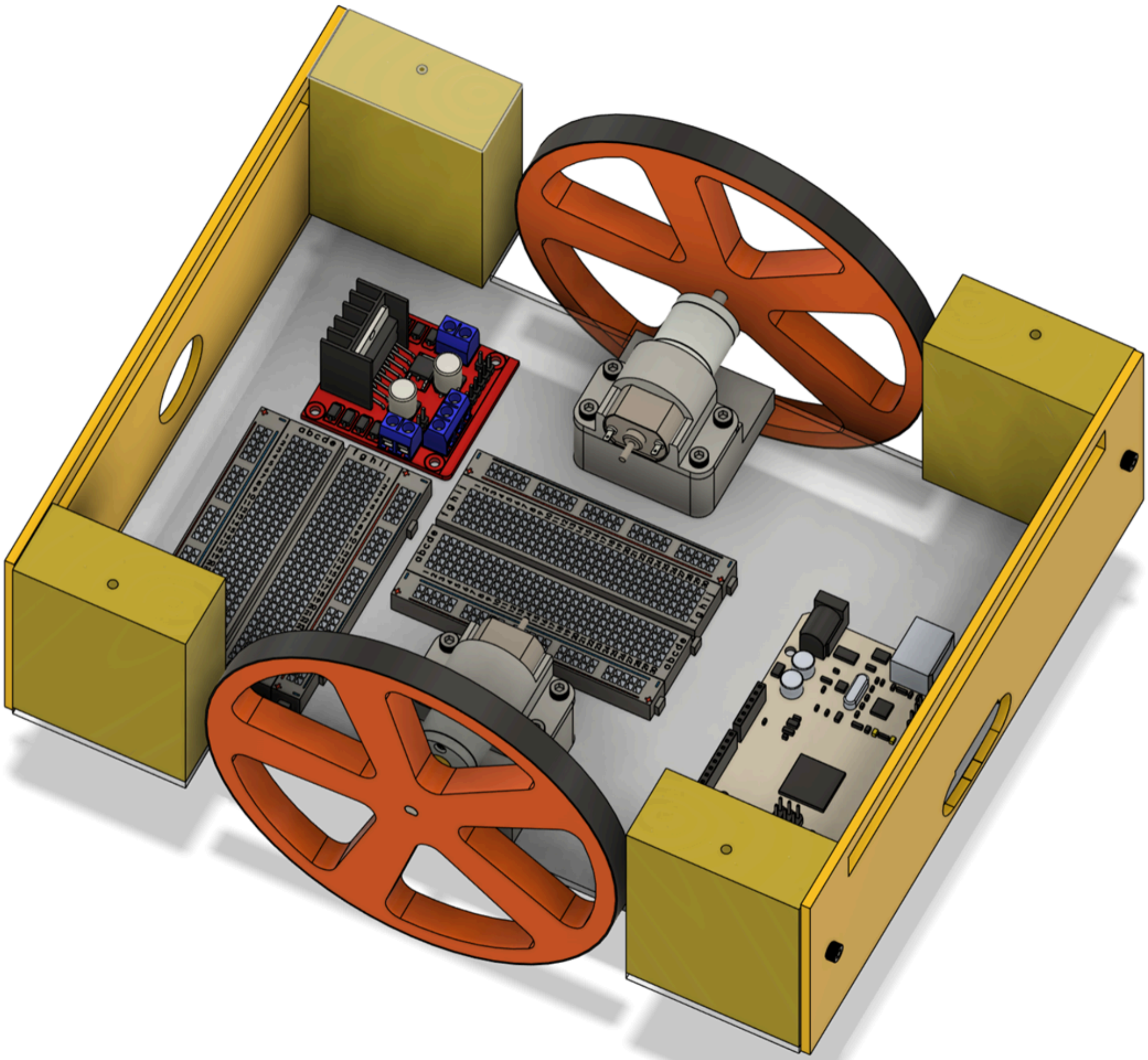


Figure 3: Full assembly close-up

<u>Assembly:</u>

1. Use hex head M3 screws to attach both motor mounts to the acrylic chassis base on the sides
2. Insert the DC motors into the motor casing
3. Wrap the 3-D printed wheels with 0.5mm thick rubber
4. Position the wheels on the motors
5. Use epoxy to connect the rollerball block to the bottom of the bottom chassis in the middle (4 inches near the front).
6. Screw the bottom of the wood blocks to the chassis base
7. Screw the front and back chassis into the wooden blocks

8. Position the bread boards, h-bridge, and arduino mega inside the OTV's interior
9. Screw each corner of the top chassis into the holes in the wooden blocks
10. Screw the servo mount into the top chassis
11. Screw the water reservoir basket into the top chassis
12. Hot glue the servos to the 3-D printed arm
13. Position the servos into the servo mount

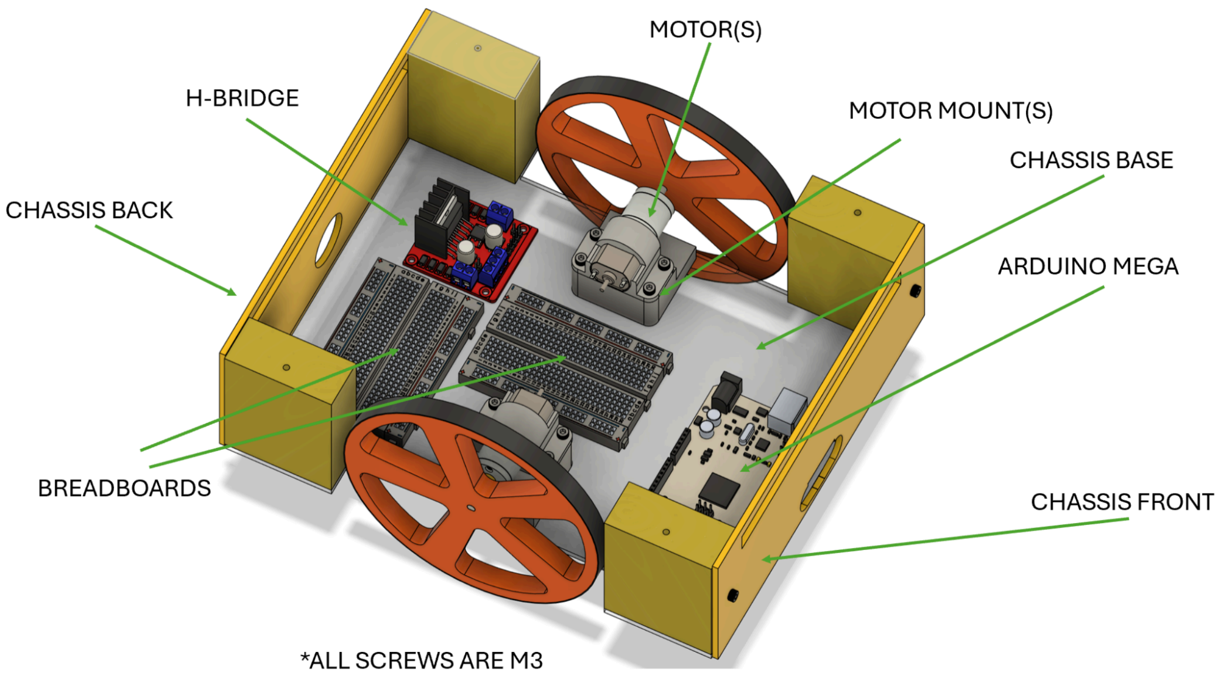


Figure 4: Final full assembly with lid unfastened

<u>Aesthetics:</u>

- The wooden blocks and the acrylic chassis top, bottom, and walls were painted yellow to resemble the body of a duck. The servo holders and the arm were also painted yellow.
- The wheels are a vibrant orange to symbolize the feet of the duck.

Mission

- Mission apparatus:

Design Choice	Justification
Separate Water Reservoir	Keeps water collected away from the electronics of the OTV without risk of spilling
Arm apparatus	Control the tubing positioning, as well as position the cameras in a spot that allows for a clear reading.

Table 2 - Planned design choices for our OTV mission apparatus

Materials/Construction method: Water reservoir and arm apparatus were both 3D printed in order to let them be fully customizable. The water reservoir had to fit in the center of our OTV and out of the way of the electronics housed in the chassis, and the arm had to house the two sensors used for depth and pollution sensing

Attachment: Base of mission apparatus connects at front edge of OTV chassis, and hangs over edge when lowered down with servo motors

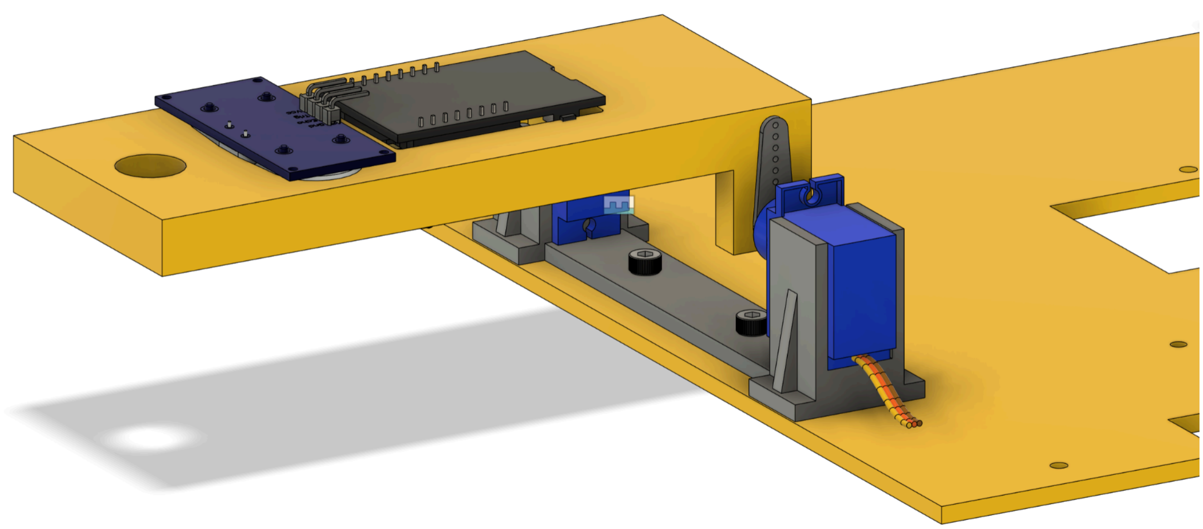


Figure 5 - OTV mission apparatus attachment to chassis

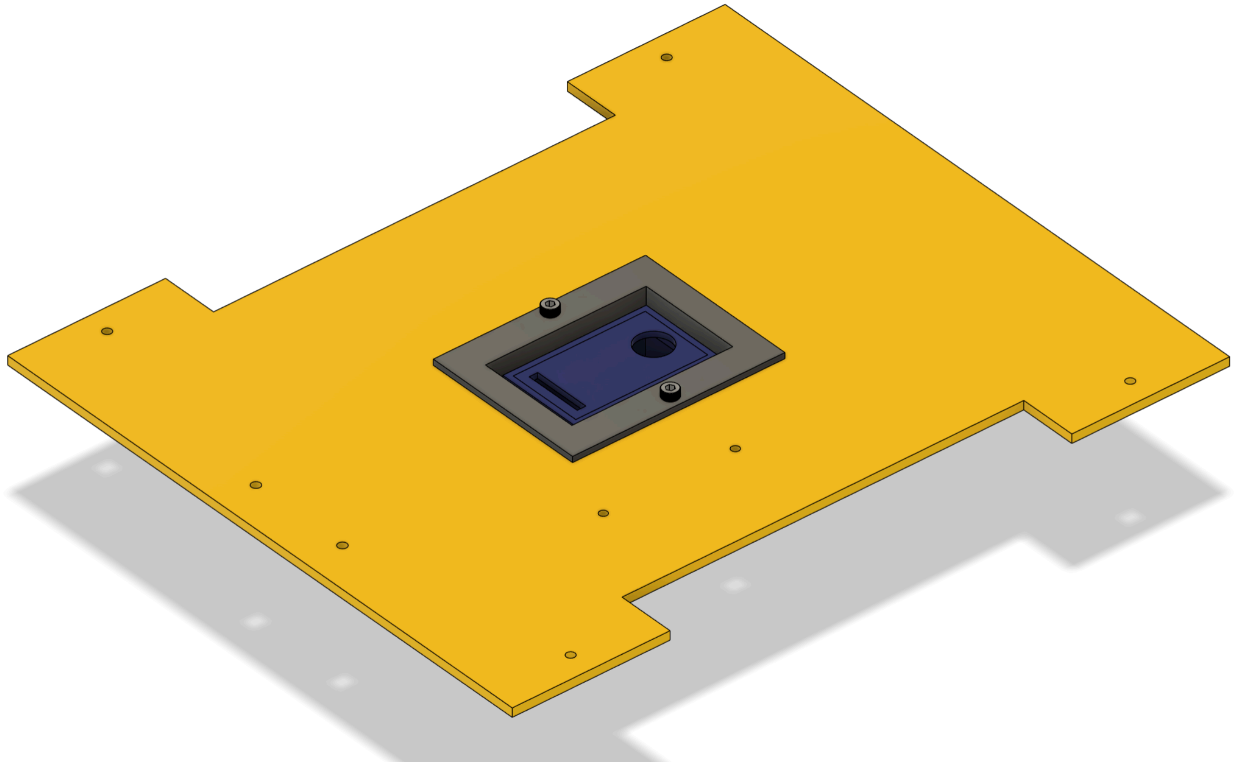


Figure 6 - Water reservoir housed within chassis

- Sensors, Actuators and power:

Design Choice	Justification
ESP-32 Machine Learning Camera	Used for the pollution detection requirement - trained through the ENES model workbook
Ultrasonic Sensor	Used for the depth sensing requirement of the mission detail - either 20, 30, or 40 mL depth of water
DC Water Pump	Used to collect the water sample required for the mission - at least 20 mL
GY-31 Color Sensor	Switched to the color sensor after the ML camera had problems with wifi connectivity. Remained unused due to lack of time to

	implement.
Dual Servo Motors	Raise and lower mission arm to position sensors and navigate under limbo obstacle
Arduino Mega	Allows for enough pins with breadboards to house all needed electronics

Table 3 - Mission-specific electronics chosen for the OTV

Attachment:

- Servo Motors were placed at the base of the mission apparatus, where it would be attached to the chassis in order to raise and lower the arm for testing and navigation
- ESP-32 Camera and Ultrasonic sensor attached with a clearance fit, having designed cutouts in the arm structure to fit the sensors dimensions
- DC Water Pump was attached to the top of the chassis, with the tubing running from the reservoir in the center of the OTV through the mission arm, with its own cutout made for the dimensions of the tubing
- GY-31 Color Sensor was planned to be attached in the area vacated by the ESP-32 Camera once plans were changed

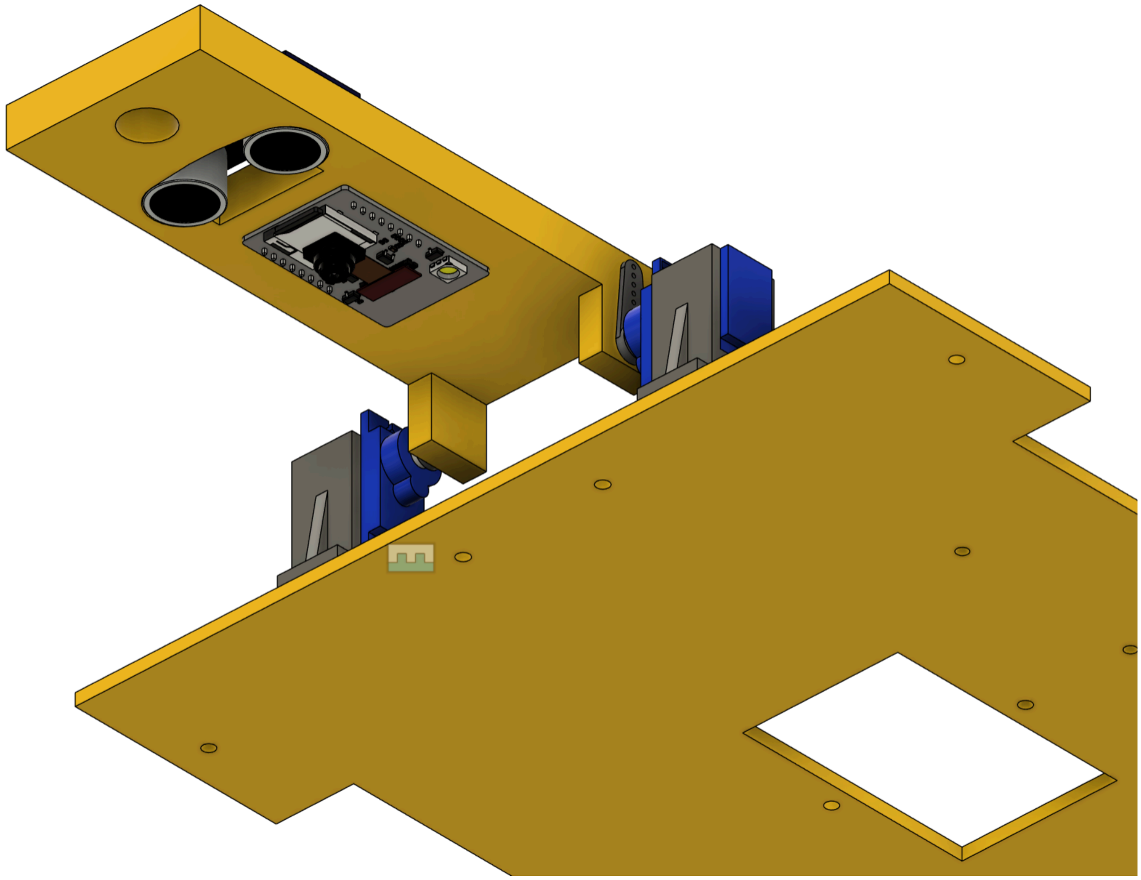


Figure 7 - Sensors housed in the OTV mission apparatus

Power:

	Max Current Draw(A)	Operating Voltage (V)	Operating Power (W)	Time in Use (h)	Energy Consumption (mWh)
ESP-Cam	0.17	3.3	0.561	1/60	9,350
Ultrasonic Sensor	0.02	5	0.2	5/60	16,700
2 Servo Motors	2×0.2	5	0.2	1/60	3300
DC Water Pump	0.08	12	0.96	30/3600	8

Table 4 - Power chart for mission specific models

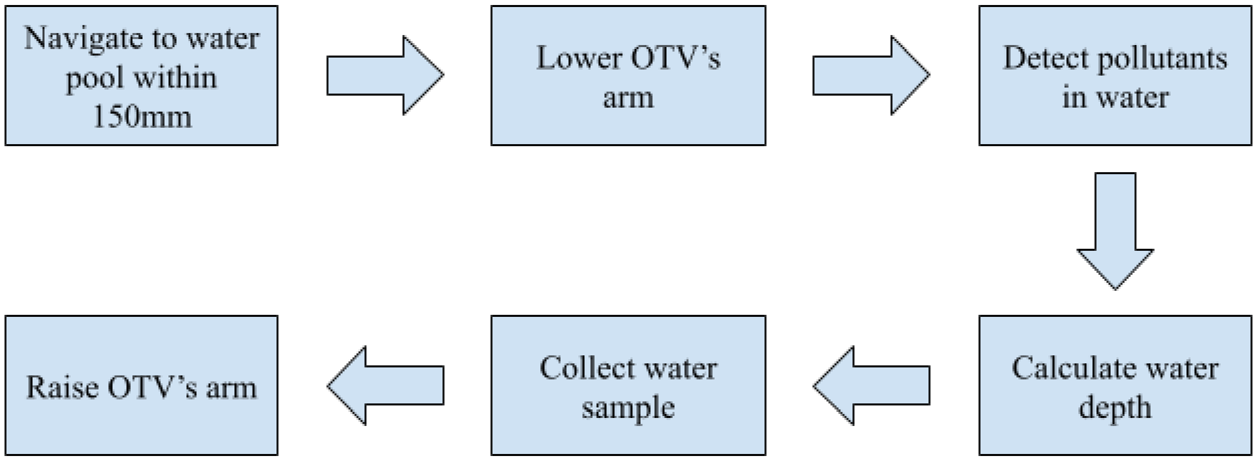


Figure 8 - Flowchart of Control Algorithm for Mission-Specific Tasks

Changed from original:

- Switched ML camera for a color sensor camera
- Got rid of servo motors(keeping arm stationary) to save pin space to focus on navigation
- Ended up not doing any mission connections for the OTV final run; stayed focused on trying to get navigation to work

<u>Propulsion Modeling</u>

Component	Design Choice
Configuration	• Two wheels on either side • Two individual motors connected to a single H-bridge
Wheel/Tread Size	• 3D Printed Wheels → 13 cm • Rubber Treads → 0.1 cm
Motors	• Two 12V DC motors (550 RPM)
Drive Train	• Two motors for each wheel →

	forward propulsion and turning • Independent moving wheels
Steering System	• H-bridge that controls the direction • Autonomous robot → dependent on code

Table 5 - Propulsion design choices with explanations

Motor Calculations:

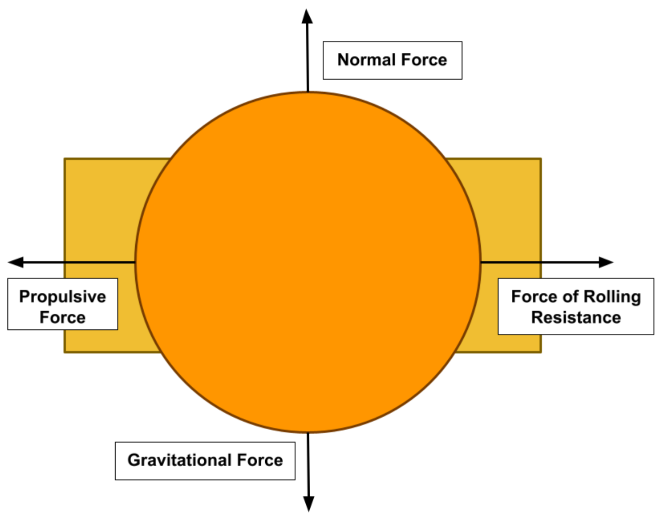


Figure 9 - Free body diagram for single wheel

- General Values
  - R = 0.066 cm
  - M = 1.6 kg
  - C_{rr} = 0.01
- Torque Requirements:

$$\tau_{req} = F_T \times R = 0.0784\,N \times 0.066\,m = 0.0051744\,[N - m] \rightarrow$$

0.528 [kg - mm]

$$F_{rr} = C_{rr} \times \frac{mg}{3} = 0.01 \times \frac{(1.6\,kg)\,(9.8\,m/s^2)}{3} = 0.0523\,N$$

$$2F_T = 3F_{rr} \rightarrow F_T = \frac{3}{2}F_{rr} = \frac{3}{2}\,(0.0523\,N) = 0.0784\,N$$

Motor Characteristic Graph:

- Linear Speed Goal: 0.18 m/s

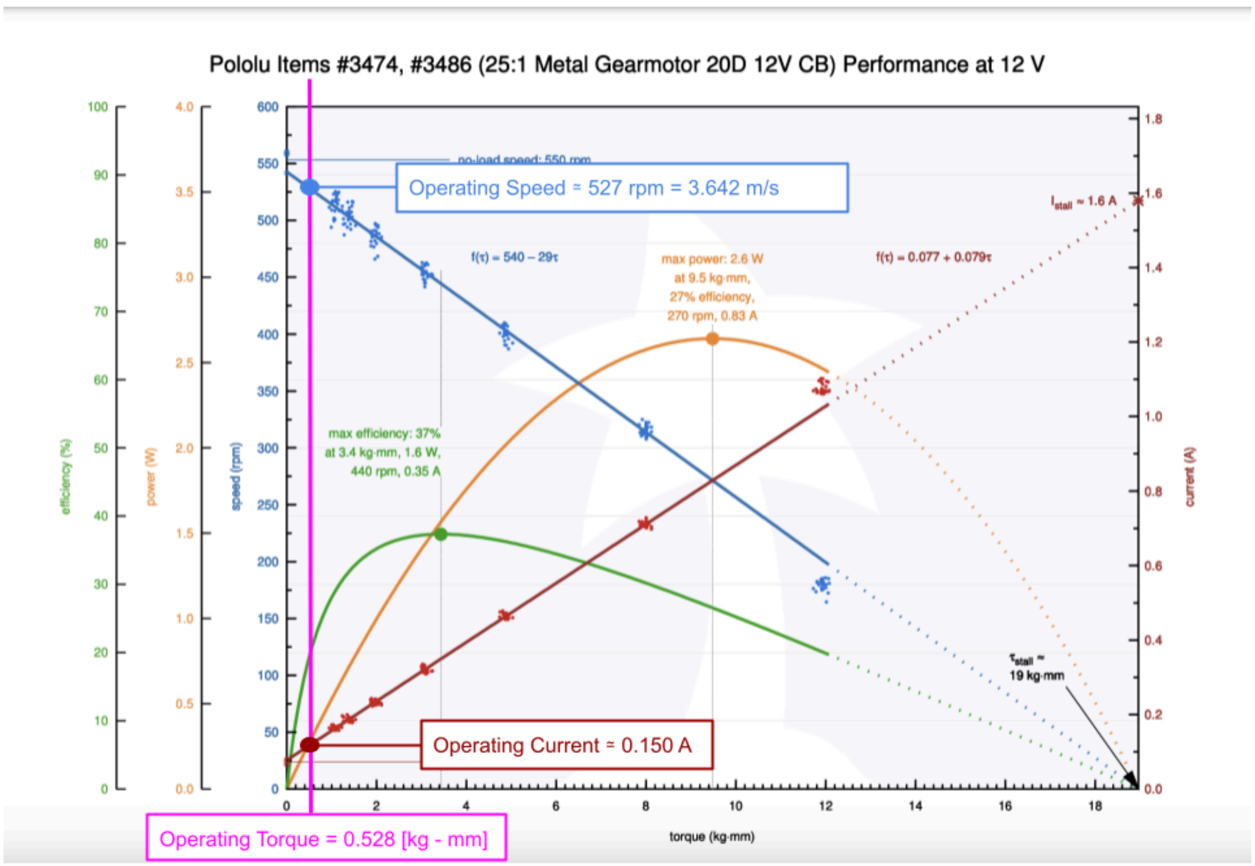


Figure 10 - Motor Curve Displaying Expected Behavior

- Operating Angular Speed:

$$\omega = \frac{v}{r} = \frac{3.642\ m/s}{0.066\ m} = 55.182\ rad/s = 527\ rpm$$

Actual Motor Behavior:

- Actual Linear Speed:
  - During the run, OTV only traveled in a circle so the linear speed was 0 m/s
  - Considering code values and speed of single wheel:
    - $v = \frac{6.096\ m}{31.6\ s} = 0.193\ m/s$

- Actual Angular Speed:

- Considering a linear speed of 0 m/s

$$\omega = \frac{v}{r} = \frac{0\ m/s}{0.066\ m} = 0\ rad/s = 0\ rpm$$

- Considering the linear speed of 0.193 m/s

$$\omega = \frac{v}{r} = \frac{0.193\ m/s}{0.066\ m} = 2.924\ rad/s = 27.922\ rpm$$

- Operating Torque:

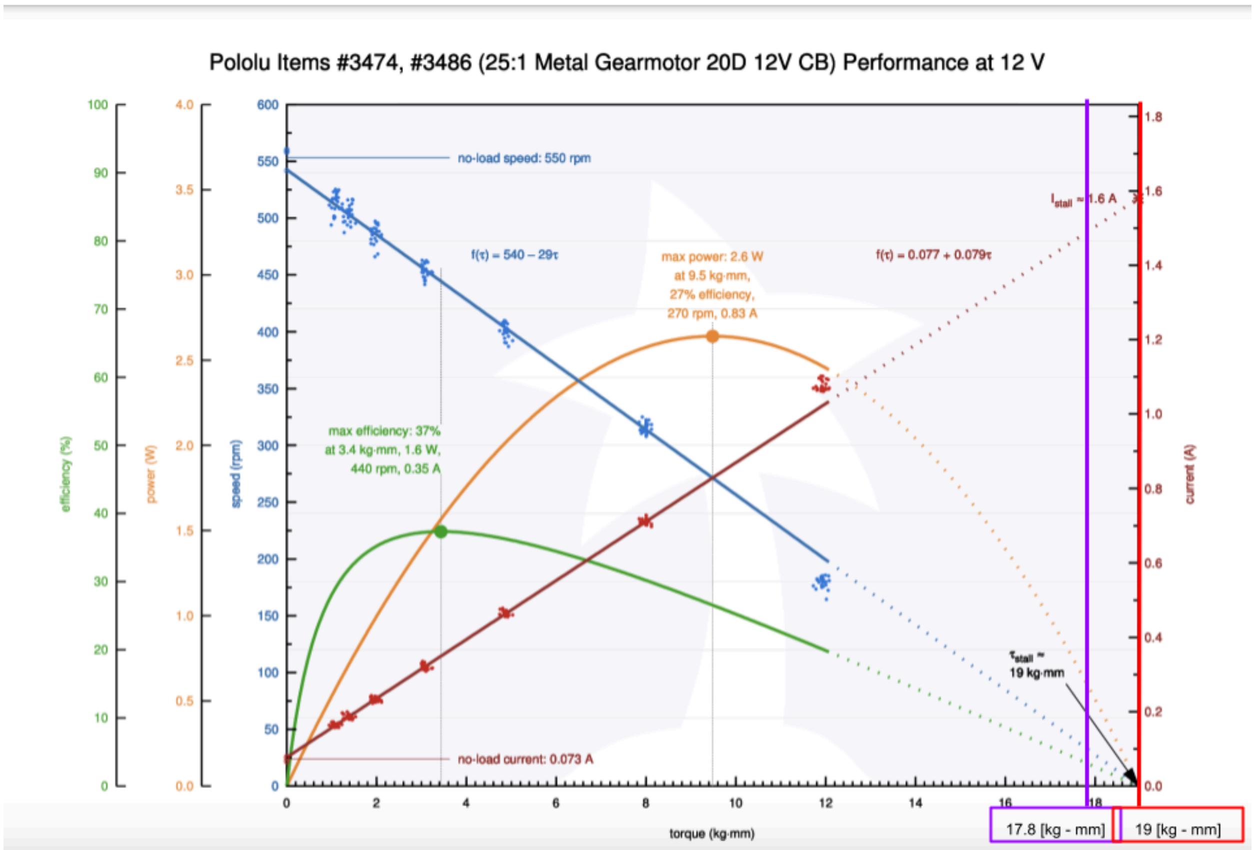


Figure 11 - Motor curve displaying actual behavior

- Operating Torques:
  - 0 RPM Line → 19 [kg - mm] (stall torque)
  - 29.922 RPM Line → 17.8 [kg - mm]
- Comparison:
  - The calculated and actual values are extremely different, which may be due to:
    - One of our motors not working during the run
    - H-bridge limiting the voltage the motors received

- Coefficient of Rolling Resistance Being Off

Electronics

	Max Current Draw(A)	Operating Voltage (V)	Operating Power (W)	Time in Use (h)	Energy Consumption (mWh)
Arduino Mega	0.2	5	1	5/60	83.3
ESP-Cam	0.17	3.3	0.561	1/60	9.35
Wifi Module	0.17	3.3	0.561	5/60	46.7
2 Ultrasonic Sensors	2×0.02	5	0.2	5/60	16.7
2 Servo Motors	2×0.2	5	0.2	1/60	3.3
2 DC Motors	2×0.5	12	12	4/60	800
DC Water Pump	0.08	12	0.96	30/3600	8
Total:	1.6				967.35

Table 6 - Planned Electronics

	Max Current Draw(A)	Operating Voltage (V)	Operating Power (W)	Time in Use (h)	Energy Consumption (mWh)
Arduino Mega	0.2	5	1	12/3600	3.33

2 DC Motors	2×0.5	12	12	12/3600	40
Total:	1.2				43.33

Table 7: Actual Energy Calculations for Final Run

Planned Calculations

Total Current Draw:

$0.20\,A + 0.17A + 0.17A + (2 \times 0.02A) + (2 \times 0.2A) + 0.02A + (2 \times 0.5A) + 0.08A =$ **2.08 A**

Power Calculations:

P = IV

Arduino Mega: (0.2 A)×(5 V) = 1 W = 1000mW

ESP-Cam: (0.17 A)×(3.3 V) = 0.561 W = 561 mW

Wifi Module: (0.17 A)×(3.3 V) = 0.561 W = 561 mW

Ultrasonic Sensor: (0.02 A)×(5 V) = 0.1 W = 100 mW

Servo Motor: (0.02 A)×(5 V) = 0.1 W = 100 mW

Line Follower: (0.02 A)×(5 V) = 0.1 W = 100 mW

DC Motor: (0.5 A)×(12 V) = 6 W = 6000 mW

DC Water Pump: (0.08 A)×(12 V) = 0.96 W = 960 mW

Energy Consumption Calculation:

E = Pt → mWh

$(1000mW) \times (\frac{5}{60}h) + 561mW \times (\frac{1}{60}h) + 561mW \times (\frac{5}{60}h) + 200mW \times (\frac{5}{60}h) + (200mW) \times (\frac{1}{60}h) + (100mW) \times (\frac{30}{3600}h) + (12000mW) \times (\frac{4}{60}h) + (960mW) \times (\frac{30}{3600}h) =$ **967.35 mWh**

Run time calculations before recharging:

Current Drawn from one run: (1600 mA)× $(\frac{5}{60}h)$ = 133.3 mAh

Possible Run Attempts 1 battery: $\frac{2000mAh}{133.3\ mAh}$ ≈ **15 attempts**

Battery Details

- 2000mAh - battery capacity
- 12V - voltage
- 1A - recommended discharge rate

Actual Calculations

Power Calculations:

P = IV

Arduino Mega: (0.2 A)×(5 V) = 1 W = 1000mW

DC Motor: (0.5 A)×(12 V) = 6 W = 6000 mW

Energy Consumption Calculation:

E = Pt → mWh

$(1000mW) \times (\frac{12}{3600}h) + (12000mW) \times (\frac{12}{3600}h)$ + (960mW) × $(\frac{30}{3600}h)$ = **43.33 mWh**

Battery Details

- 2000mAh - battery capacity
- 12V - voltage
- 1A - recommended discharge rate

Run time calculations before recharging:

Current Drawn from one run: (1200 mA)× $(\frac{12}{3600}h)$ = 4 mAh

Possible Run Attempts 1 battery: $\frac{2000mAh}{4\ mAh}$ ≈ **500 attempts**

Overall Schematic*:

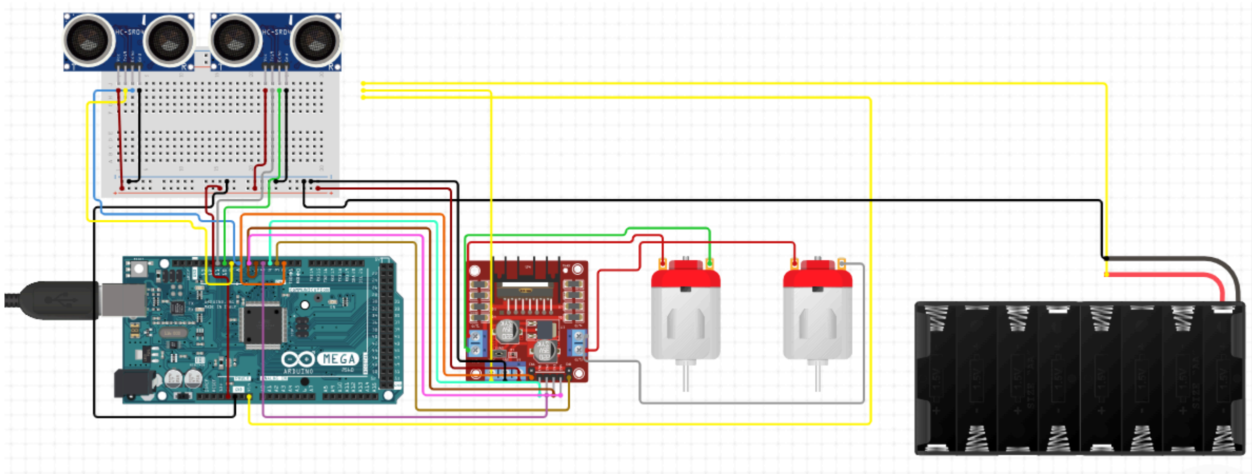


Figure 12: Final Overall Schematic

*Unable to display kill switch due to the circuito.io software. It is connected to Vin through the battery's power wire.

Ultrasonic Sensors:

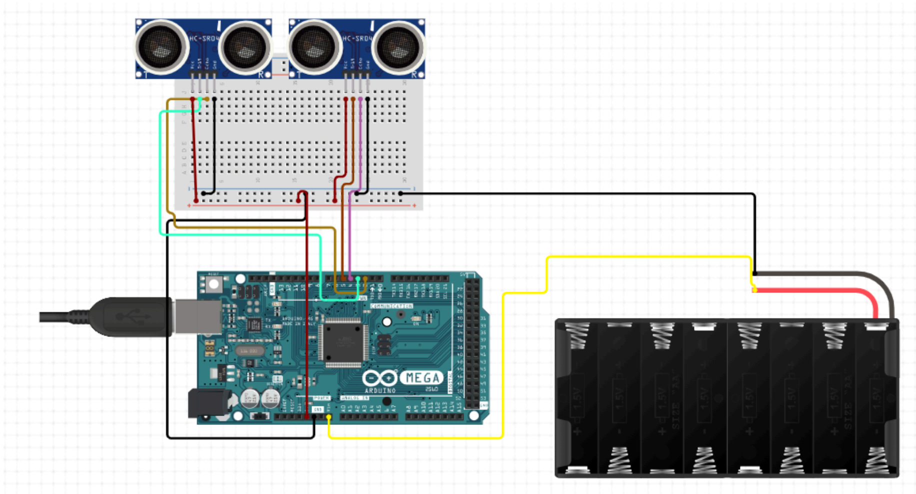


Figure 13: Ultrasonic Schematic

Pin Diagram:

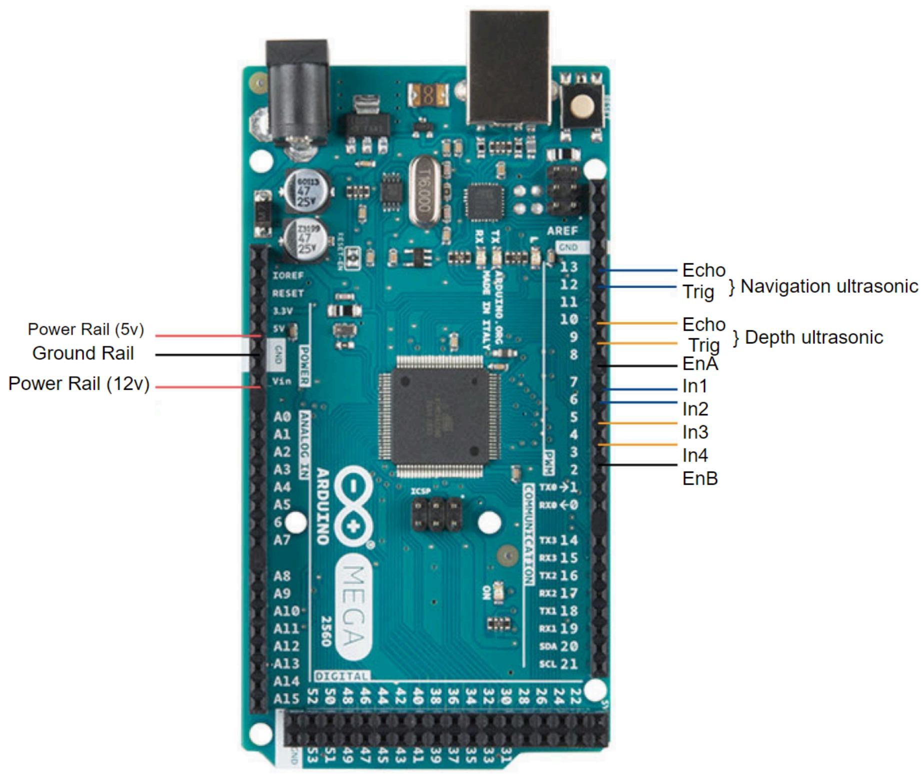


Figure 14: Pin Diagram

H- Bridge + Motor Schematics:

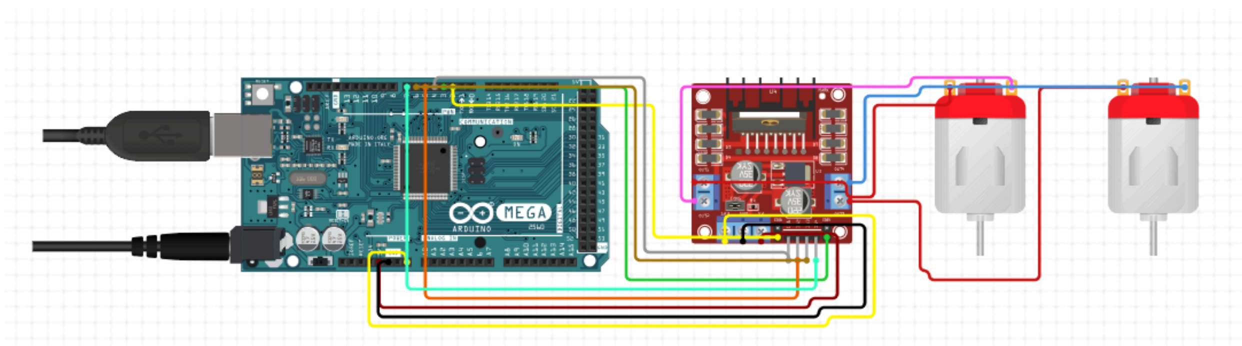


Figure 15: Driving Diagram

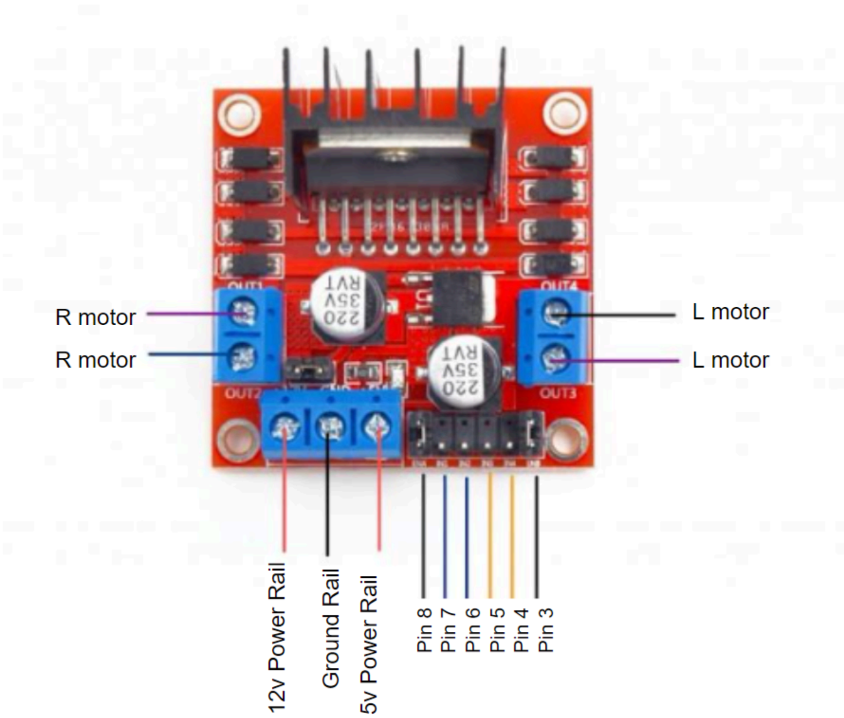


Figure 16: H-Bridge Pin Diagram

Changes from original:

-Removed the wifi-module due to its incompatibility with our arduino mega

-Removed the machine learning camera functionality as we were not able to find a working one

-Removed the pump as it wasn't self priming and thus unable to collect the water with our design

-Removed the servos as we switched between different Arduinos and had to save pin space

Engineering Drawings

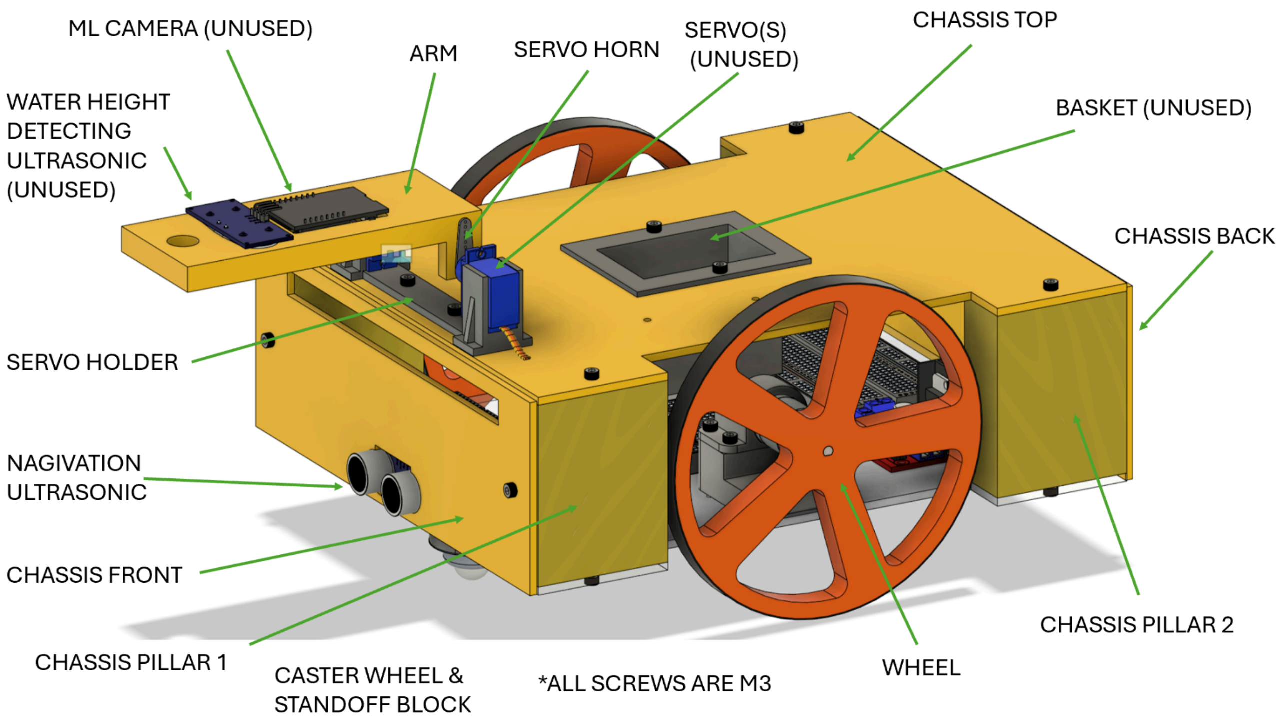


Figure 17: Exterior Labeled

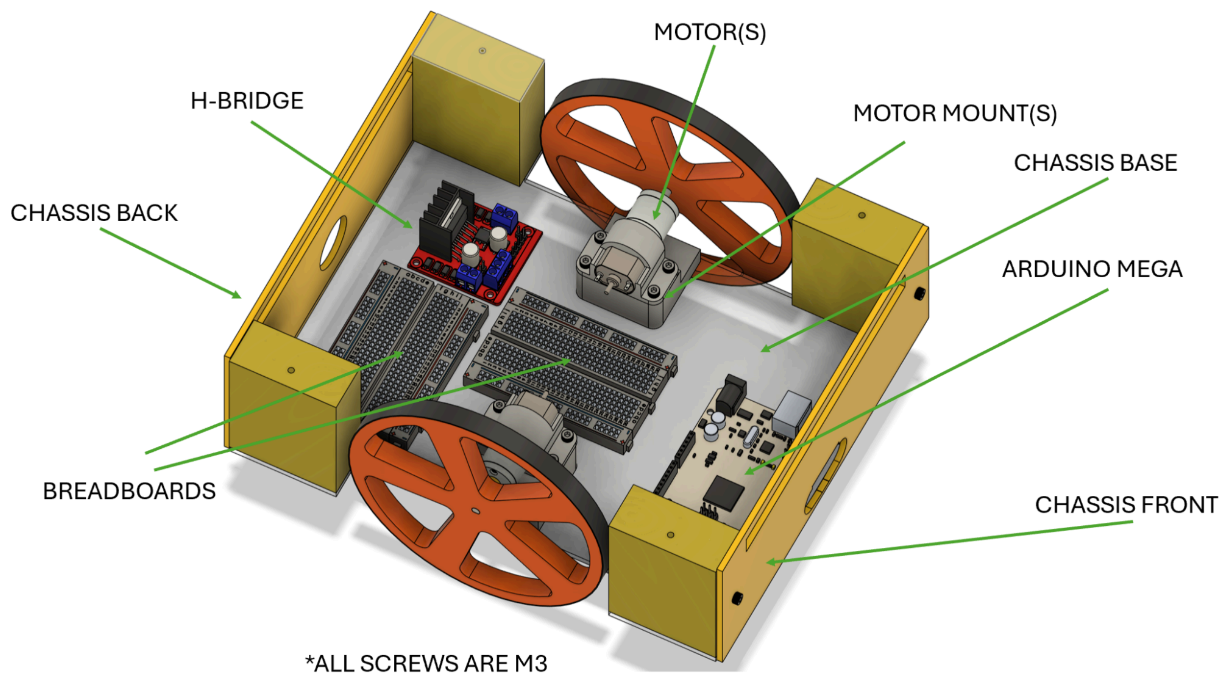


Figure 18: Interior Labeled

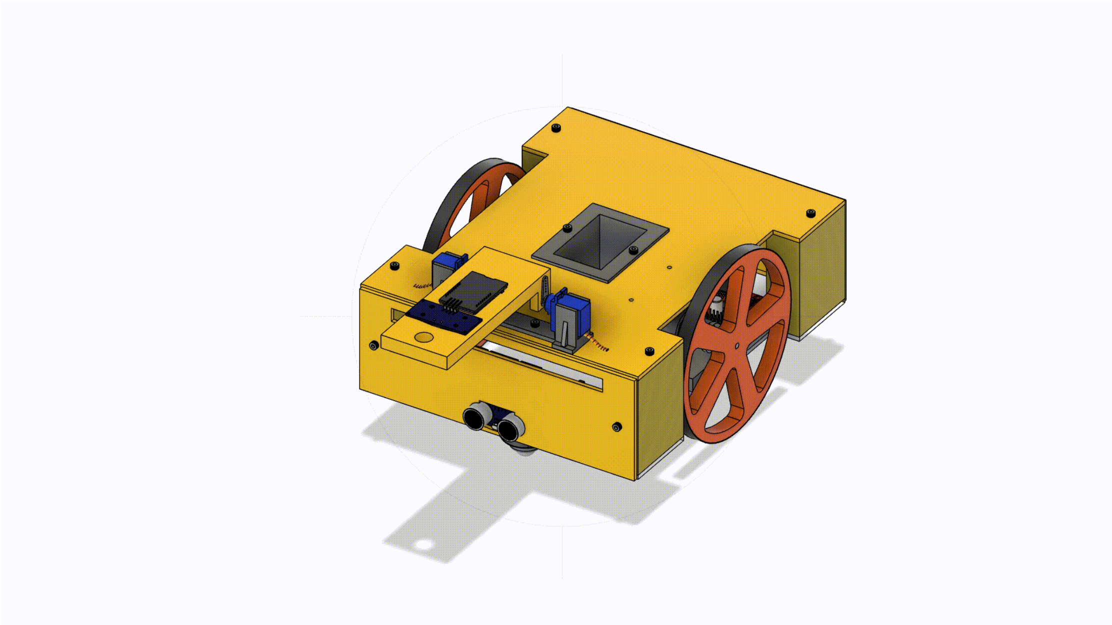
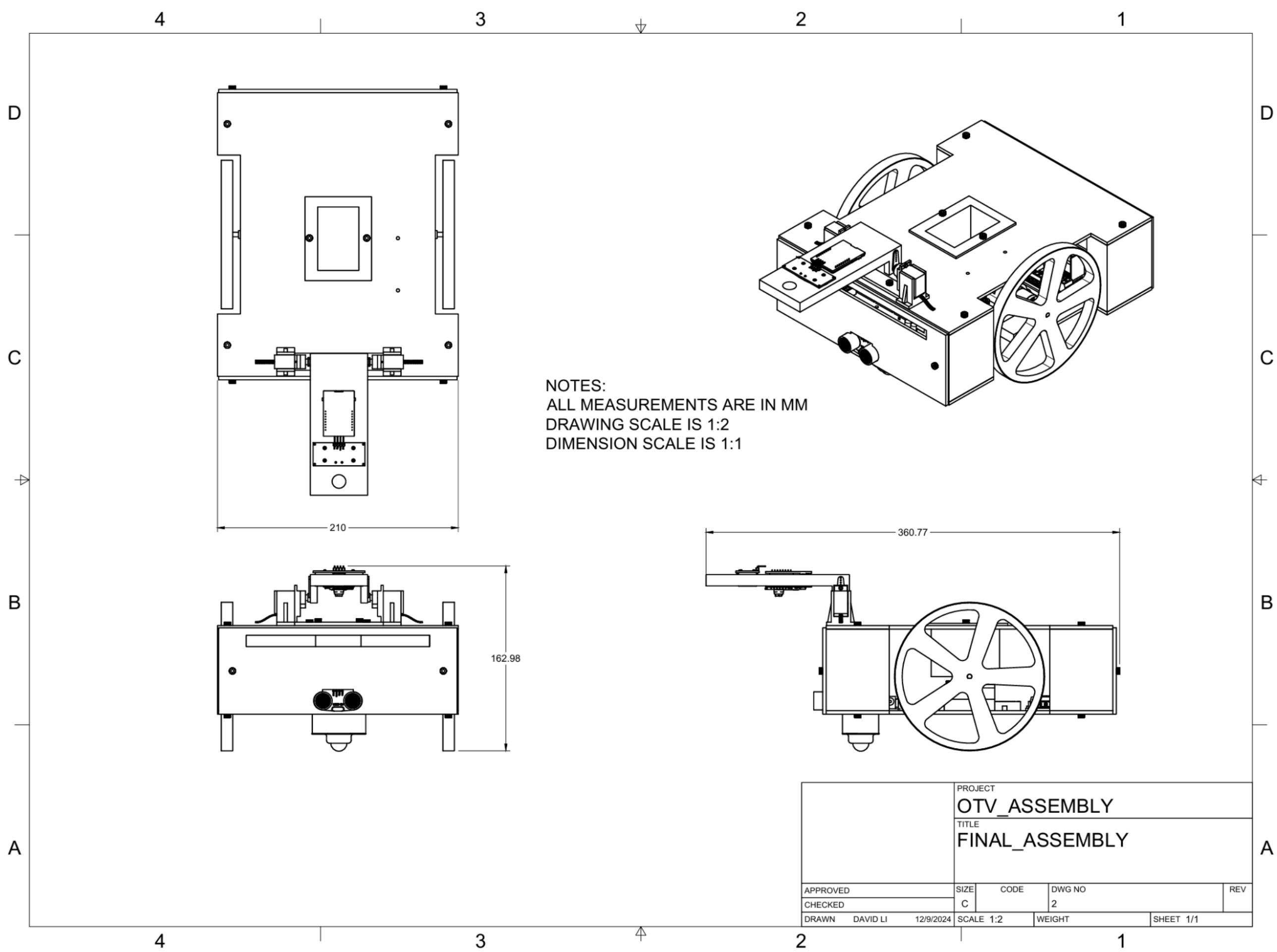
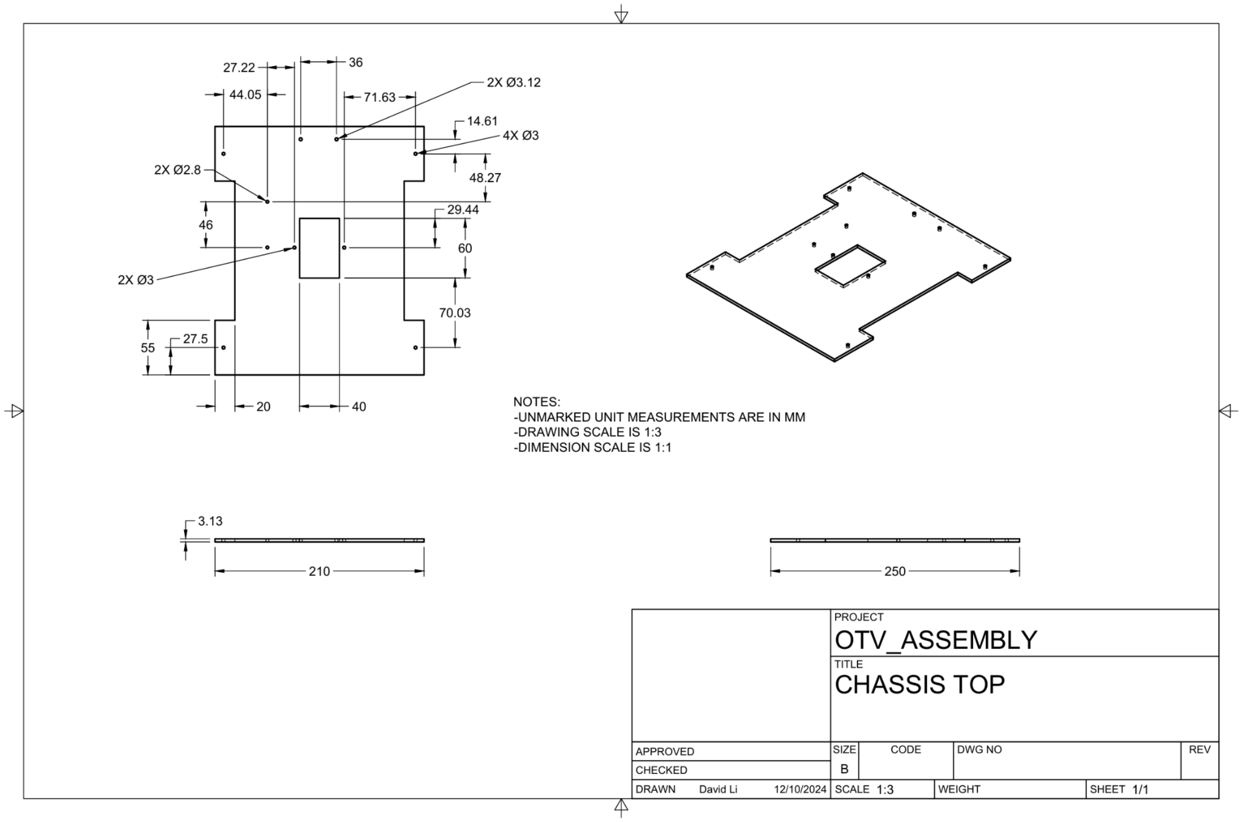


Figure 19: Exploded View

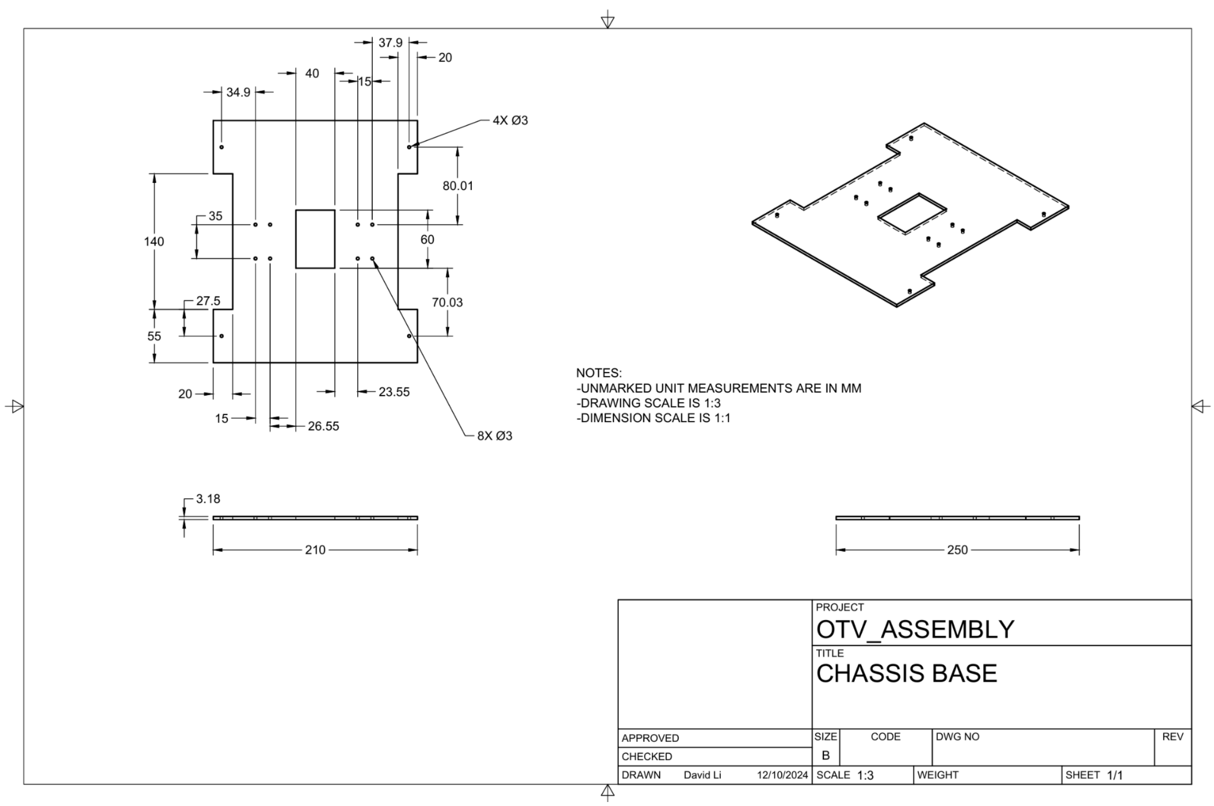
-Extents:



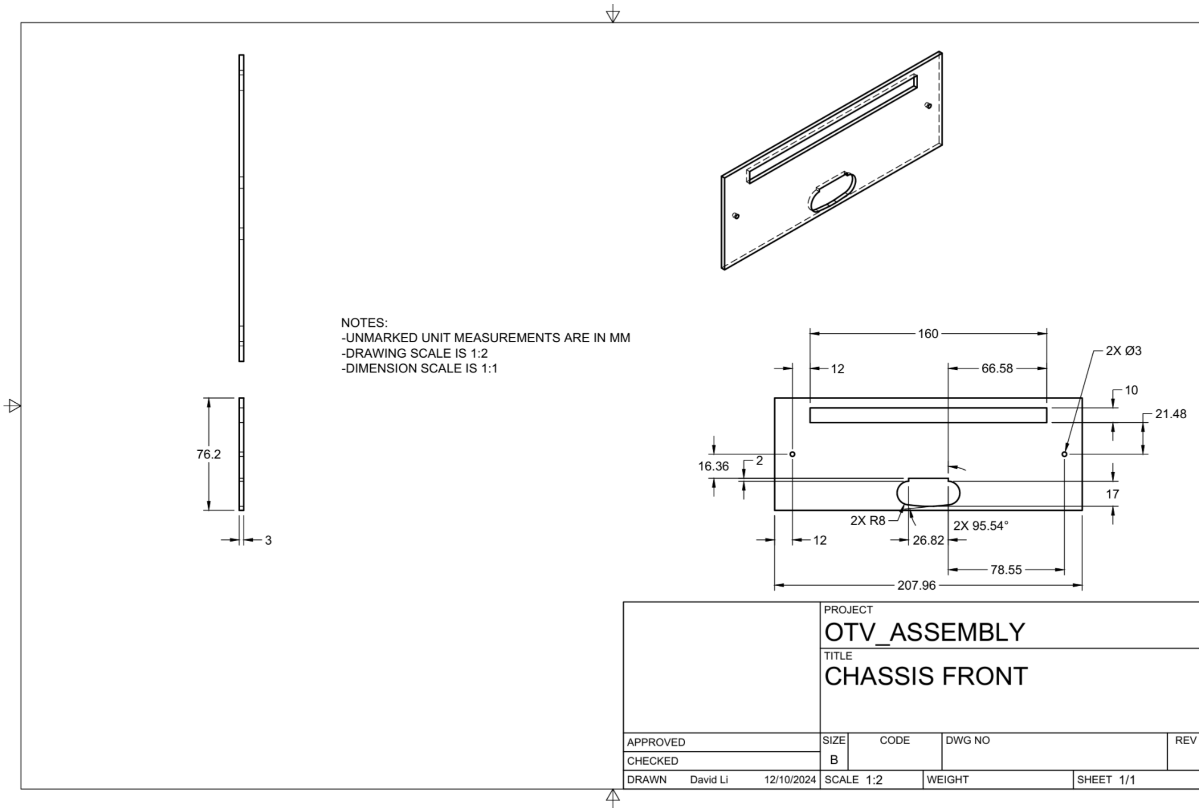
-Chassis Top



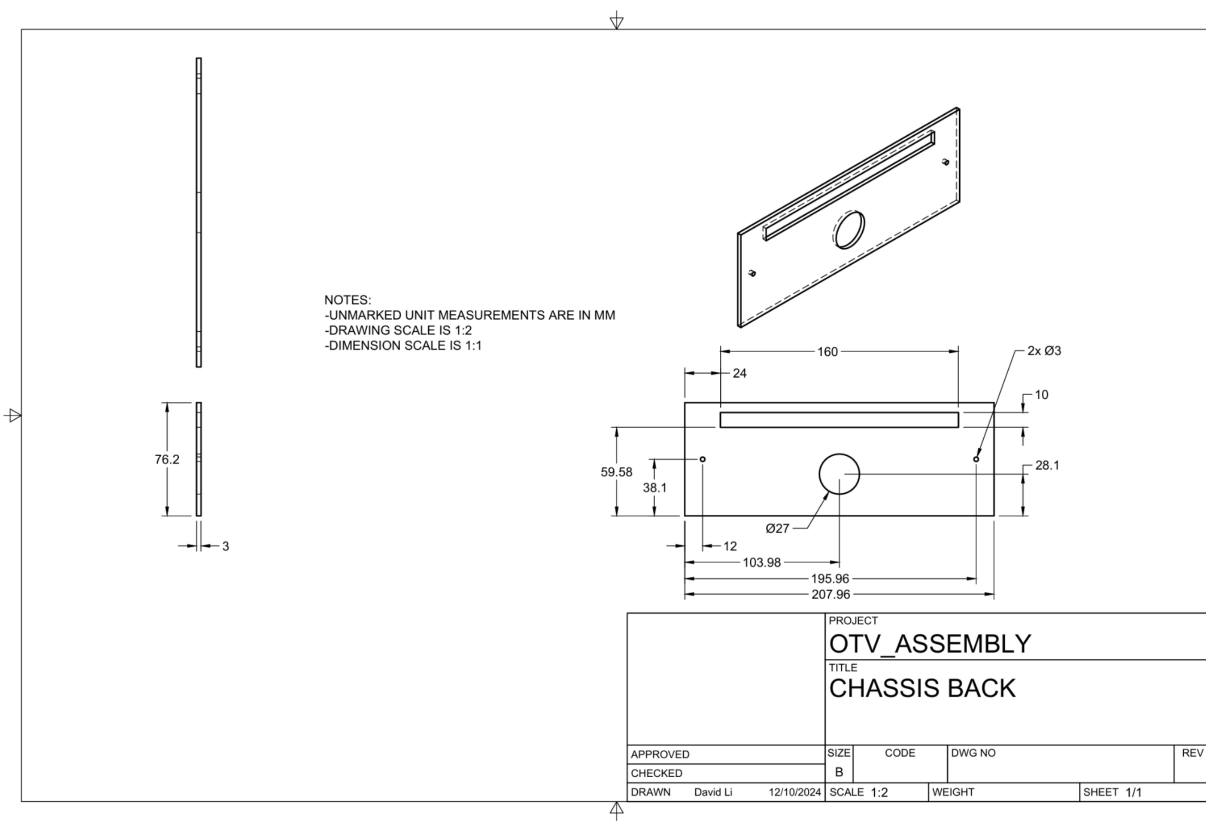
-Chassis Base



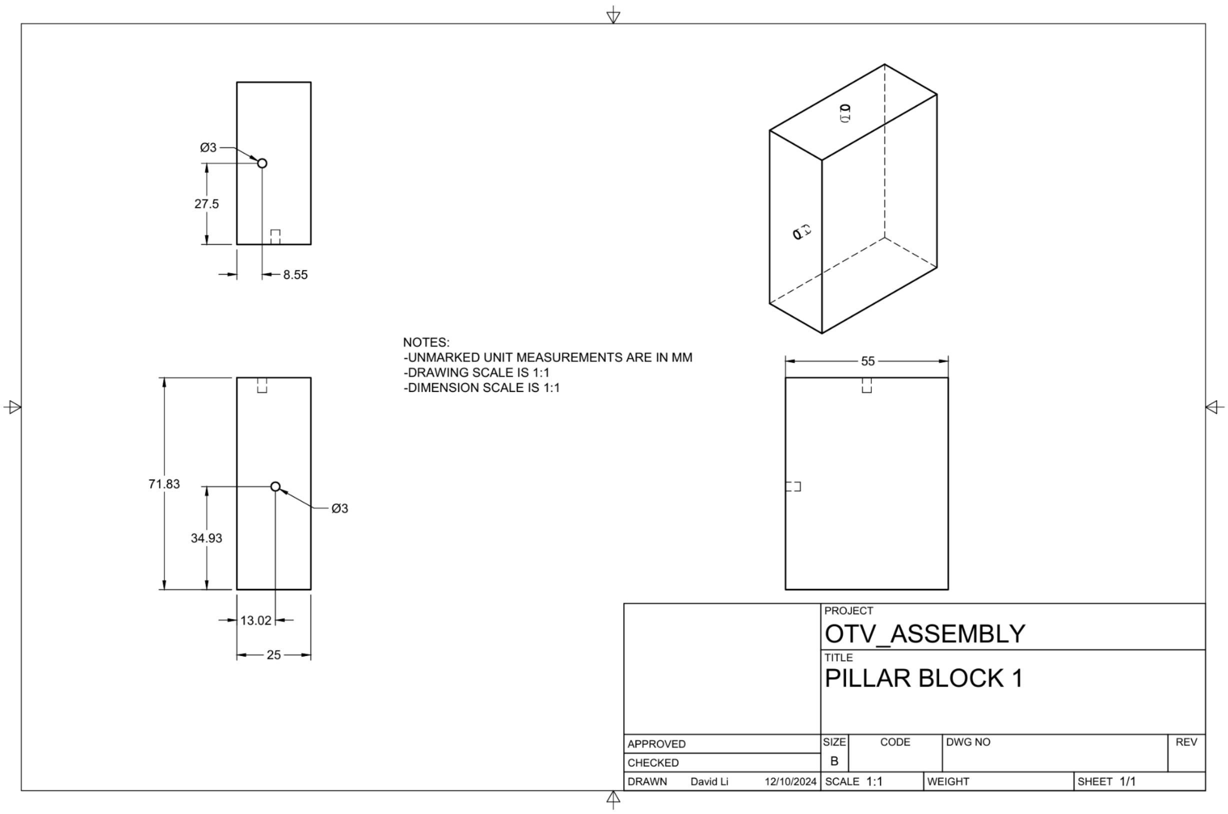
-Chassis Front



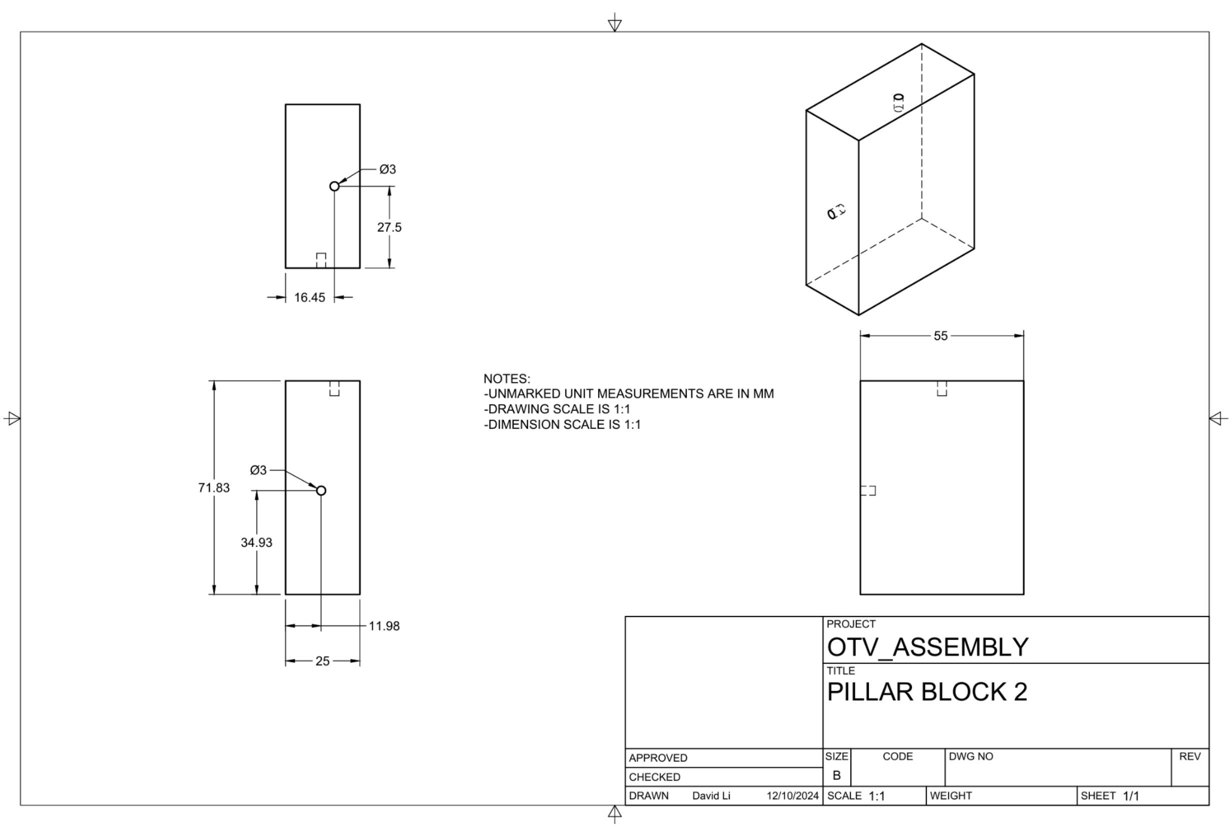
-Chassis Back



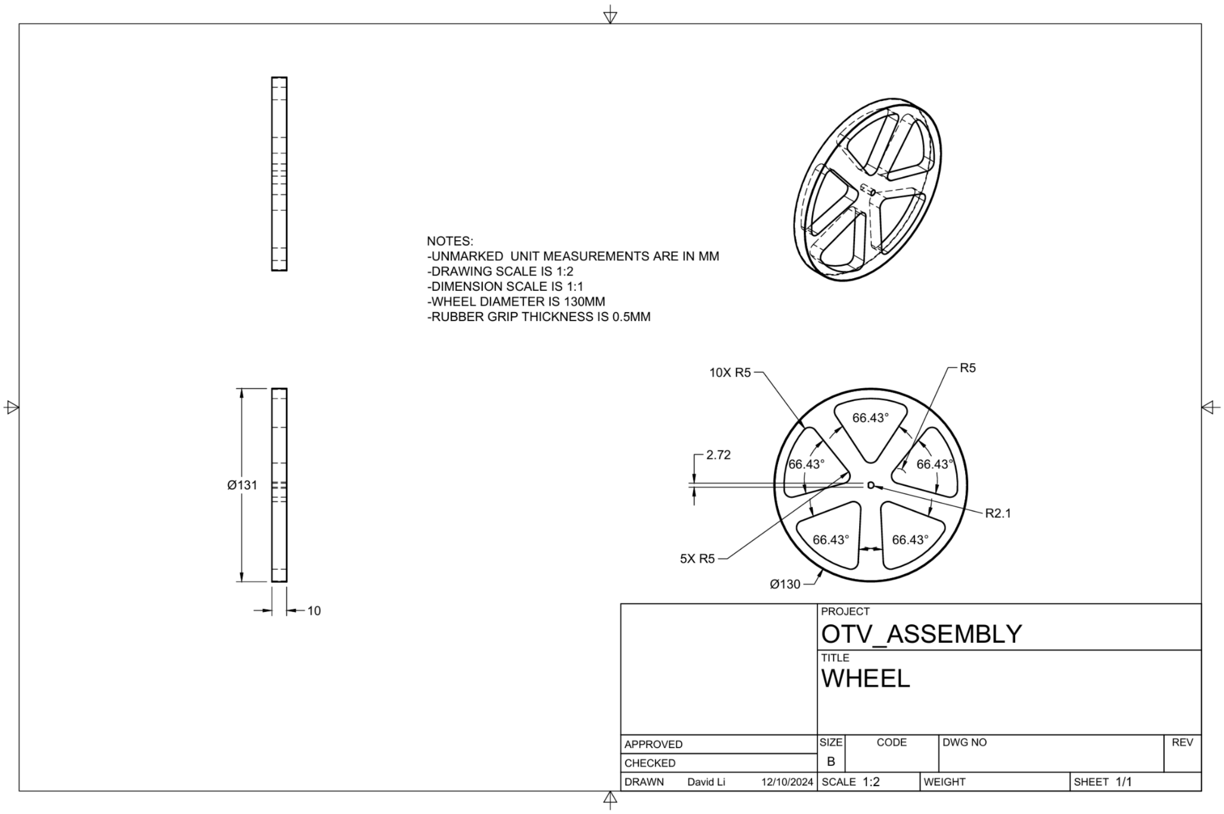
-Chassis Pillar 1



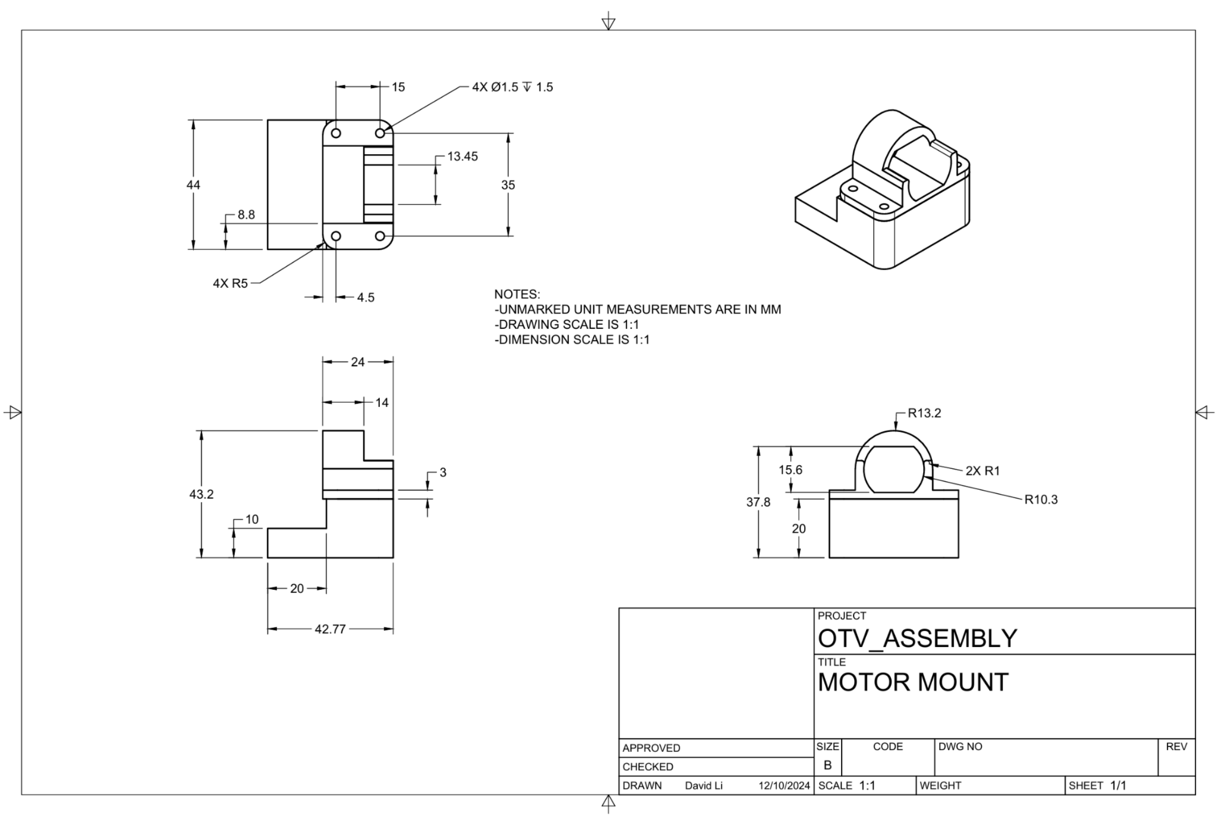
-Chassis Pillar 2



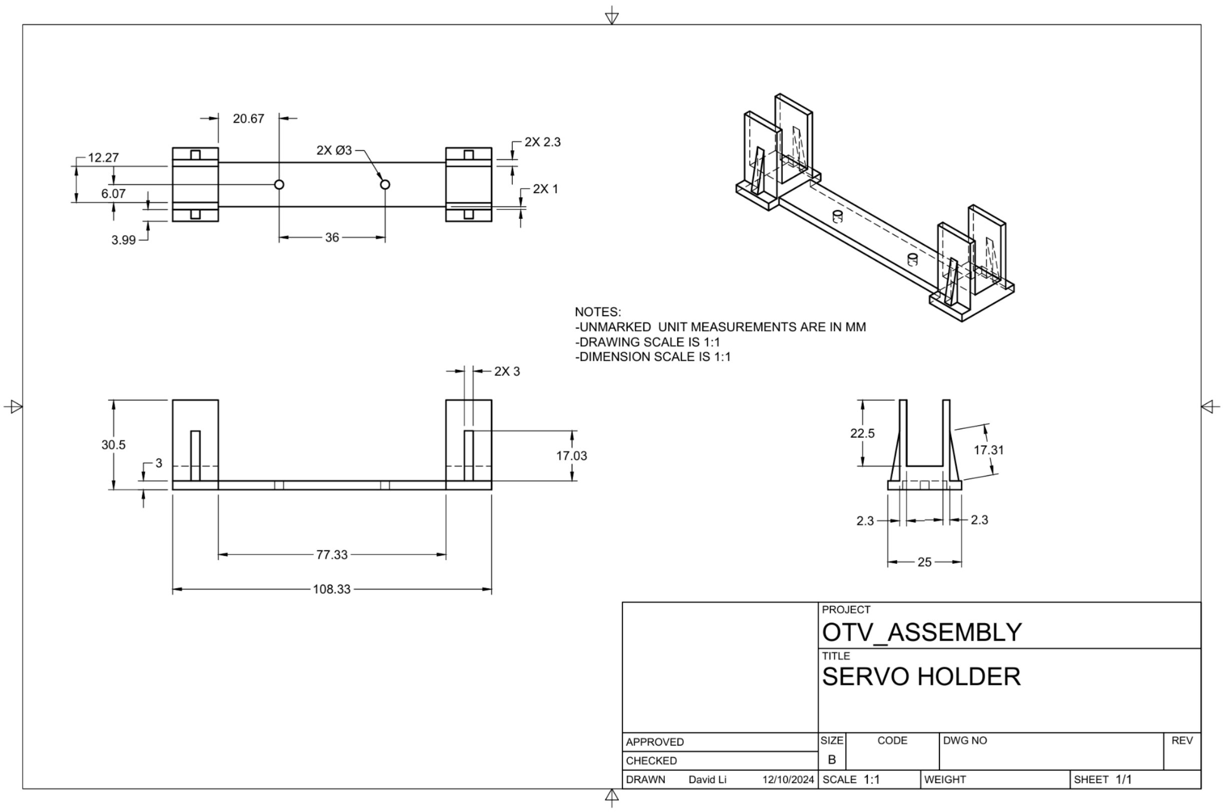
-Wheel



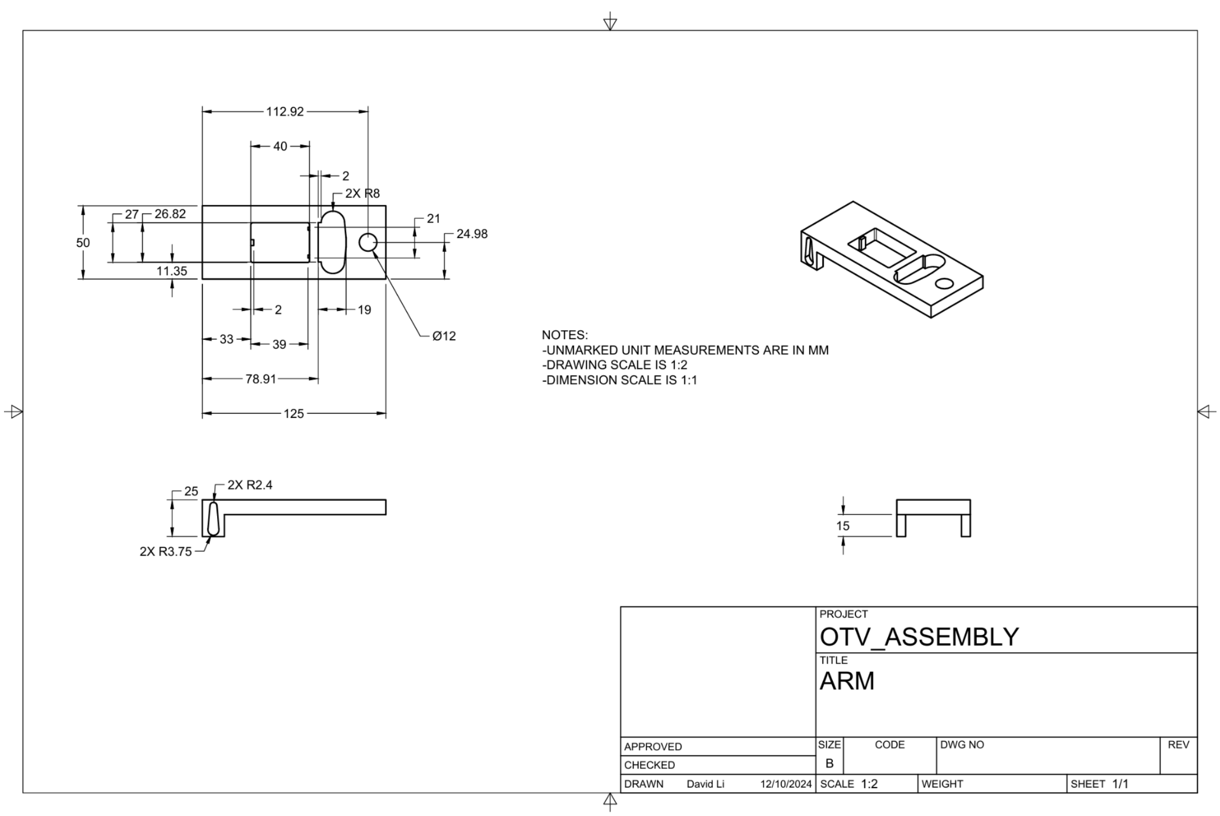
-Motor Mount



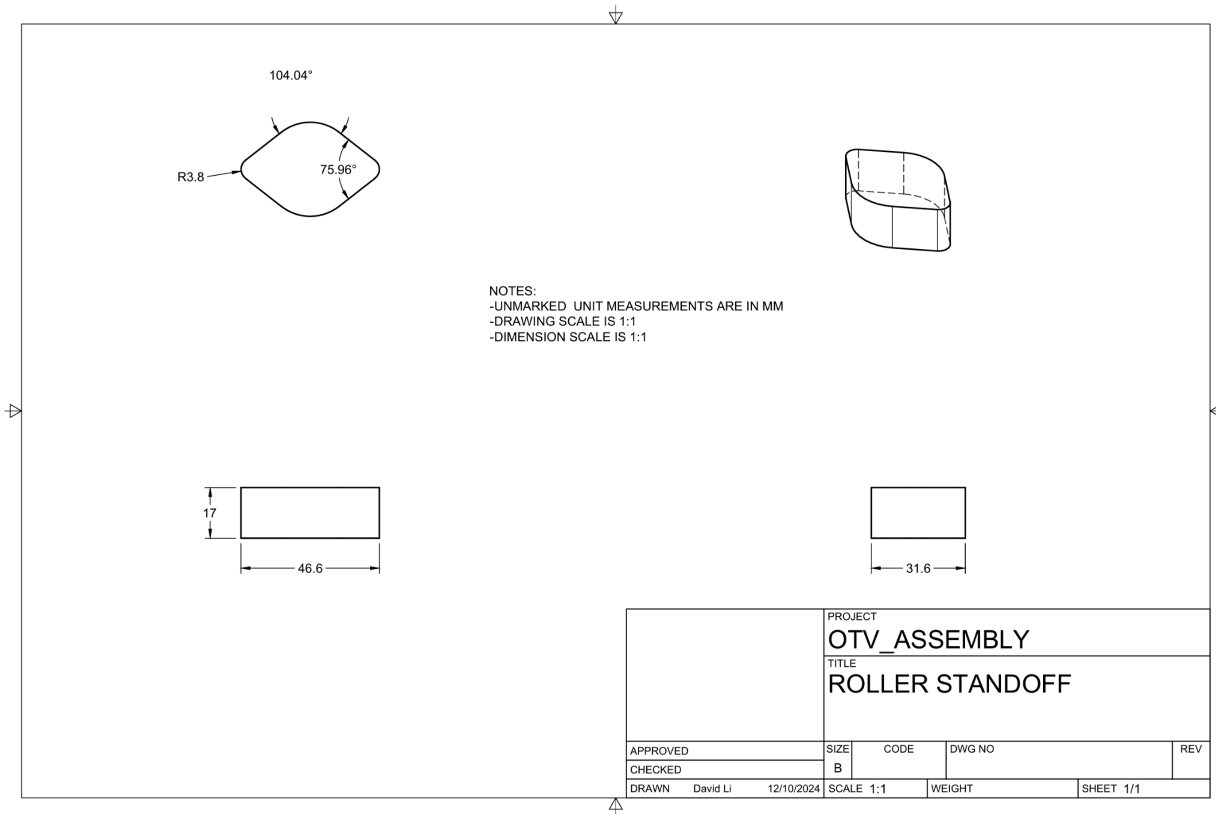
-Servo Holder



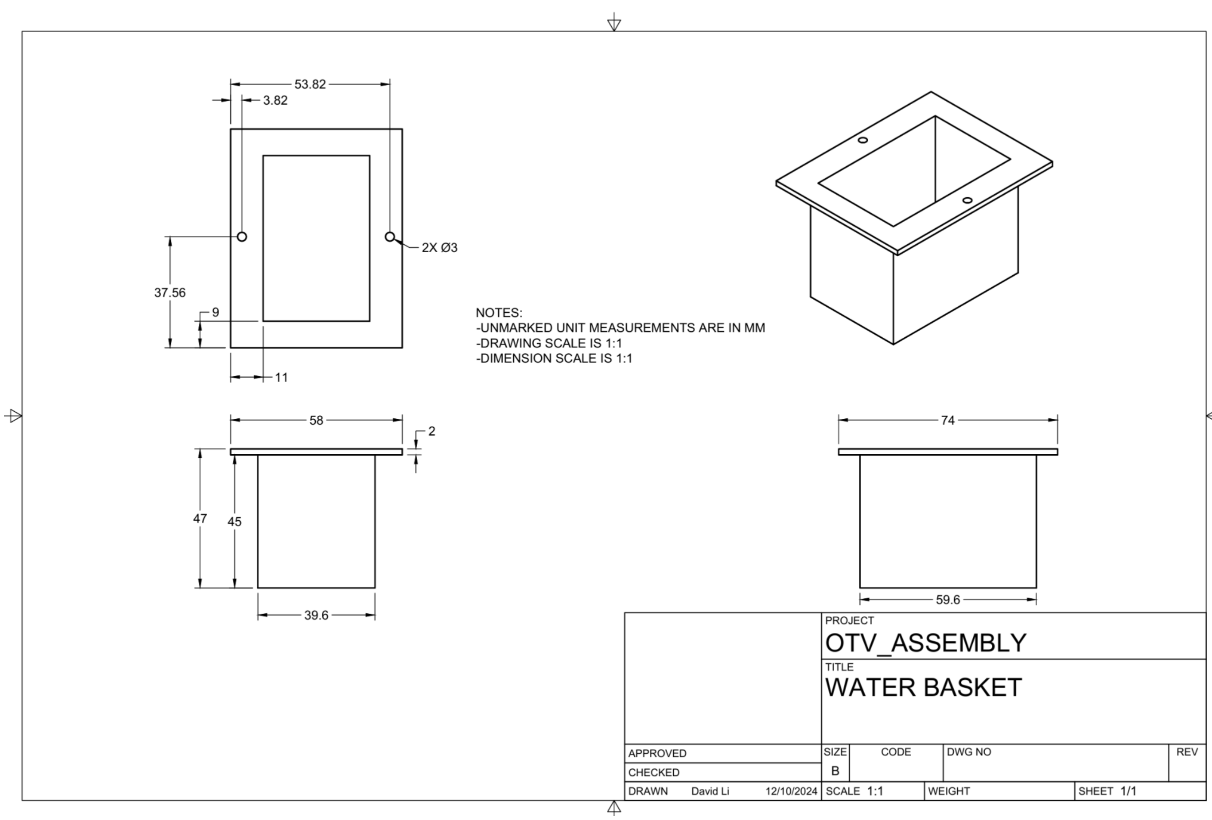
-Arm



-Caster Standoff



-Water Basket



Troubleshooting and Iteration

a) Share a specific example of how your team creatively approached problem solving. What tools, resources, techniques, and people were used in this creative approach?

In our team's original plans and prototype, we were going to have 3D printed walls along all 4 sides (front and back) to protect the OTV and cover wiring and the water reservoir. This image portrays the original prototype for the OTV, including the 4 3D printed gray walls.

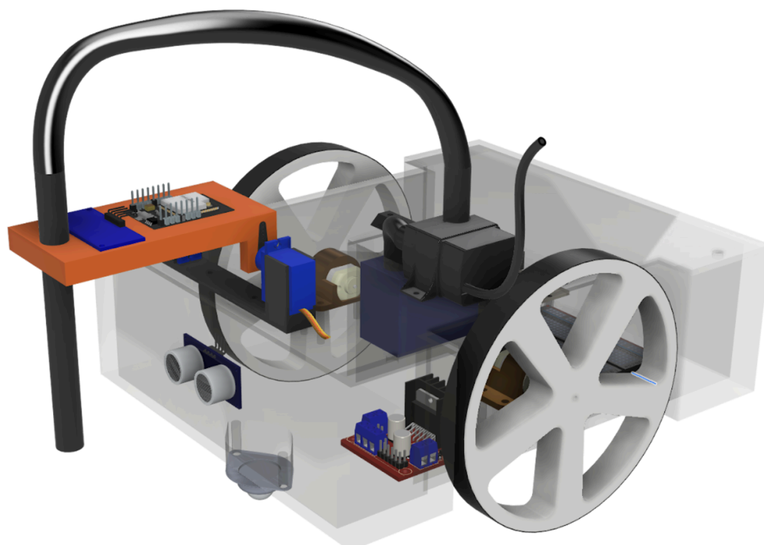


Figure 20 - Original Render

We sent a request to terrapin works on the first day of the build phase to 3D print our four walls since this print would be too large and take too long to do in the printers available in the classroom. We gave them all the information needed to print the walls including dimensions and weight. We kept checking to see if they processed our request and started working on it but after 2 weeks they still had not. This is when we realized we had to abandon this idea and think of a different solution for the walls. We canceled the terrapin works order and started thinking about what we could do instead. This delay put us behind in building and meant that we had to go back to the drawing board to figure out what we were going to do for the walls. We decided to change the material we were going to use, instead of 3D printing the walls, we decided to laser cut acrylic. We also decided to create walls for just the front and back. This would take less time and less material plus with the wheels on the other two sides, creating walls there didn't seem necessary. We didn't want the walls on the two sides to interfere with the wheels or for a team member to have to take time to figure out the shape needed for those walls. These are the pdfs for the laser cuts of the front and back walls.

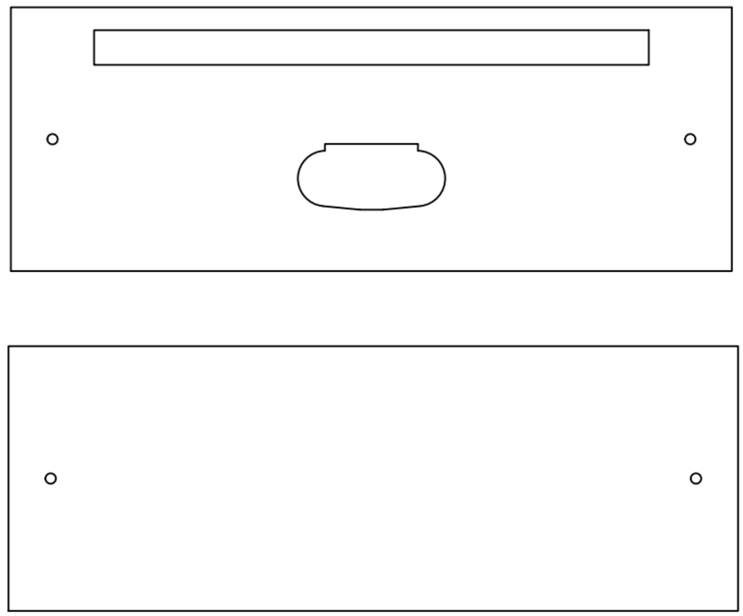


Figure 21 - Chassis front and back laser cuts

This would ensure we could finish the walls in one class period and we already had an acrylic sheet we could use. We quickly had to redesign the walls to take the new material into account. We created drawings with the correct cutouts and dimensions which we then made into the PDFs shown above. Finally we were able to laser cut the acrylic and create our walls. Additionally, we used the Terrapin Works wood shop to create support blocks for our OTV. Because the acrylic was thin and could not be stabilized by itself, we crafted wooden supports with the bottom and top chassis sandwiched together and the walls drilled into the wood. The following photos demonstrate how the wood and acrylic joined to form an insulated space for our OTV's interior.

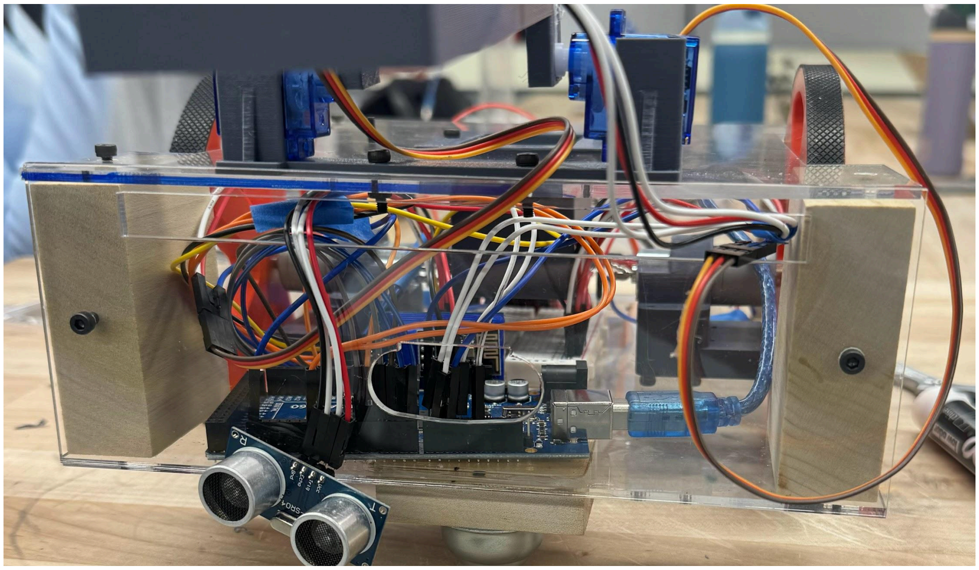


Figure 22 - In Progress OTV Construction

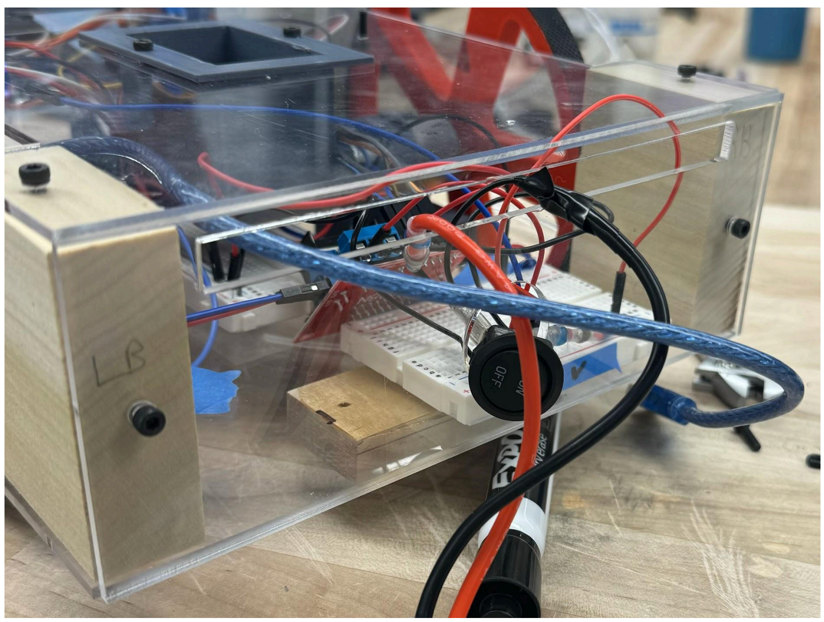


Figure 23 - Angle 2 of in progress construction

After laser cutting the walls and finalizing the supports, the construction of our OTV was complete, and those working on electronics started wiring the OTV. We could now also start fitting sensors into the wall cutouts made for them. The main people involved in this creative approach were those in charge of building the OTV. Once the problem surfaced, the team grouped up, consulted the white board, and determined a solution.

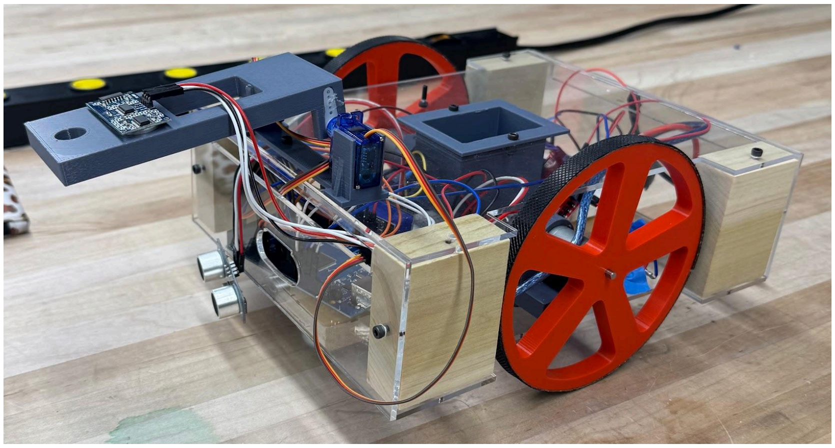


Figure 24 - Angle 3 of in progress construction

Lastly, we painted the walls, the chassis, and the arm yellow to fit our rubber duck theme. This also hid the wiring in our OTV's interior and made the OTV look better overall.

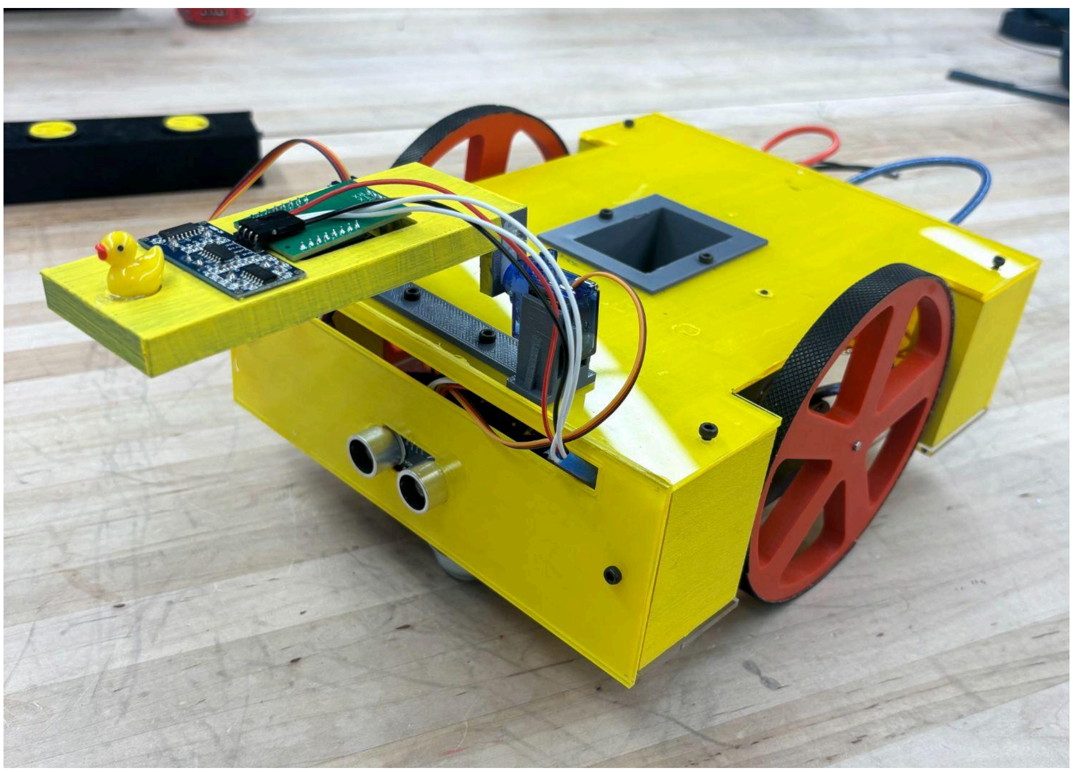


Figure 25 - Final OTV. Version used for MS7

Design Brief 7 - Teamwork and Project Management

a) The proudest moment for our team was getting our OTV to move forward.

For a few weeks of class our team was dedicated to trying to figure out forward locomotion of the OTV. During this time we had to do a lot of problem solving and had to put in a lot of lab hours outside of class. Our main issue we were having was with getting our wheels to move. At first only one wheel worked and the other did not move at all no matter if we changed the code and pins. If both wheels did happen to randomly work, they did not move at the same speed. We also could not decipher what we were doing that changed whether or not both wheels worked. A main problem of ours was figuring out which pins were controlling the left wheel and which pins were controlling the right wheel.

```
int In4 = 2;
int In3 = 4;
int In2 = 5;
int In1 = 6;
int EnA = 7;
int EnB = 3;


//#include "Enes100.h"


void setup() {
  Serial.begin(9600);
  pinMode(2, OUTPUT);  //Motor
  pinMode(3, OUTPUT);  //Motor
  pinMode(4, OUTPUT);  //Motor
  pinMode(5, OUTPUT);  //Motor
  pinMode(6, OUTPUT);  //Motor
  pinMode(7, OUTPUT);  //Motor
  pinMode(14, INPUT); // wifi
  pinMode(15, INPUT); // wifi
  //pinMode();
  //pinMode();
  //pinMode();
```

Figure 26 - Snippet of master code file

To try to fix this problem we changed all the pins around multiple times to no avail. After asking for the help of our TA Kevin we figured out that the wheels were only connected to 5V instead of the 12V battery. This means they were not getting enough power and it made sense to why they were not working correctly. Because of this only one wheel could move using the 5V of power and the other was left receiving nothing to power it. To solve this problem we connected the motors to the 12V battery after switching their pins to a different breadboard. This breadboard was connected and receiving power from the 12V battery. After this we redid the pins and coding for forward locomotion to continue to try to get the wheels to move at the same speed.

```
//Enes100.begin("Hydroquack", WATER, 376, 9, 8);
delay(2000);
/*
float x = Libraries.Enes100.getX();
float y = Enes1001.getY();
float t = Enes100.getTheta();
bool v = Enes100.isVisible();
bool c = Enes100.isConnected();
*/
//right90();
//left90();
```

Figure 27 - Snippet of master code file

```
  //go();
  //motorsOff();

  delay(2000);
  go();
/*
  Enes100.println("Visibility: " + String(v));
  Enes100.println("Connection: " + String(c));
  Enes100.println("X: " + String(x) + " Y: " + String(y) + " Degree: " +
String(t));
*/
 }
void loop(){
  //right90();
  //left90();
}

void go(){
  Serial.println("Going");
  digitalWrite(In3, HIGH);
  digitalWrite(In4, LOW);
  digitalWrite(In1, LOW);
  digitalWrite(In2, HIGH);
  analogWrite(EnA, 180);
  analogWrite(EnB, 200);
  delay(2000);
}
```

Figure 28 - Snippet of master code file

After doing lots of testing of changing pins around and altering the code, we got both wheels to turn together at our station. They still were not moving at exactly the same speed but it was much less of a difference now. Once we accomplished getting both wheels to turn at almost the same speed, every member of the team went over to the OTV testing site to see our OTV move forward. The images below show the correct coding for our wheels.

```
void right90(){
  Serial.println("Right");
  digitalWrite(In3, LOW);
  digitalWrite(In4, HIGH);
  digitalWrite(In1, LOW);
  digitalWrite(In2, HIGH);
  analogWrite(EnA, 100);
  analogWrite(EnB, 100);
  delay(1600);
  motorsOff();
}
```

Figure 29 - Snippet of master code file

```
void left90(){
  Serial.println("Left");
  digitalWrite(In3, HIGH);
  digitalWrite(In4, LOW);
  digitalWrite(In1, HIGH);
  digitalWrite(In2, LOW);
  analogWrite(EnA, 100);
  analogWrite(EnB, 100);
  delay(1700);
  motorsOff();
}

void motorsOff(){
  Serial.println("Off");
  digitalWrite(In3, LOW);
  digitalWrite(In4, LOW);
  digitalWrite(In1, LOW);
  digitalWrite(In2, LOW);
  analogWrite(EnA, 0);
  analogWrite(EnB, 0);
}
```

Figure 30 - Snippet of master code file

Once our entire team was at the testing zone, the OTV was set down inside of it and the kill switch was turned on. The second it started moving forward everyone on the team started jumping up and down and cheering. It was a major accomplishment for our team that we had been trying to complete for multiple class sessions. Every member of the team had a hand in making this possible so everyone was equally excited. Below is the video of our OTV moving forward for the first time in the testing zone.

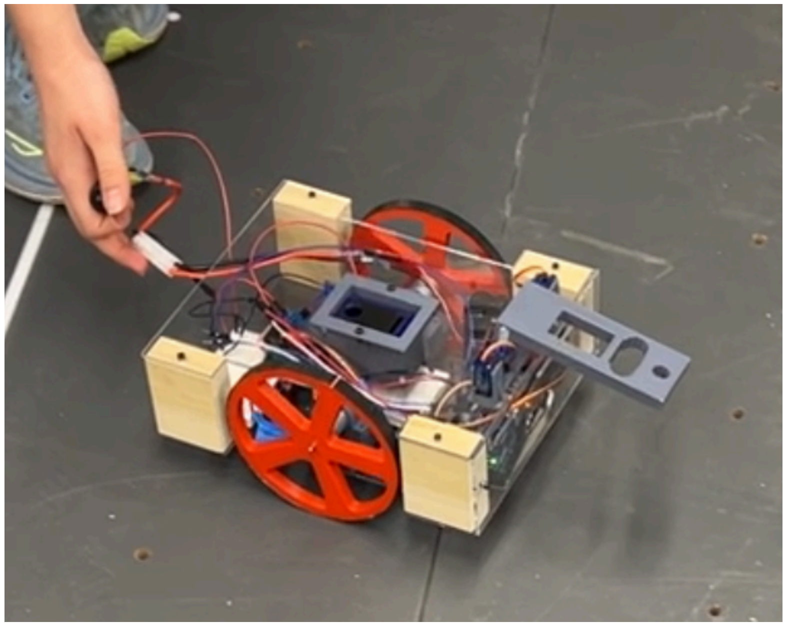


Figure 31 - Final OTV

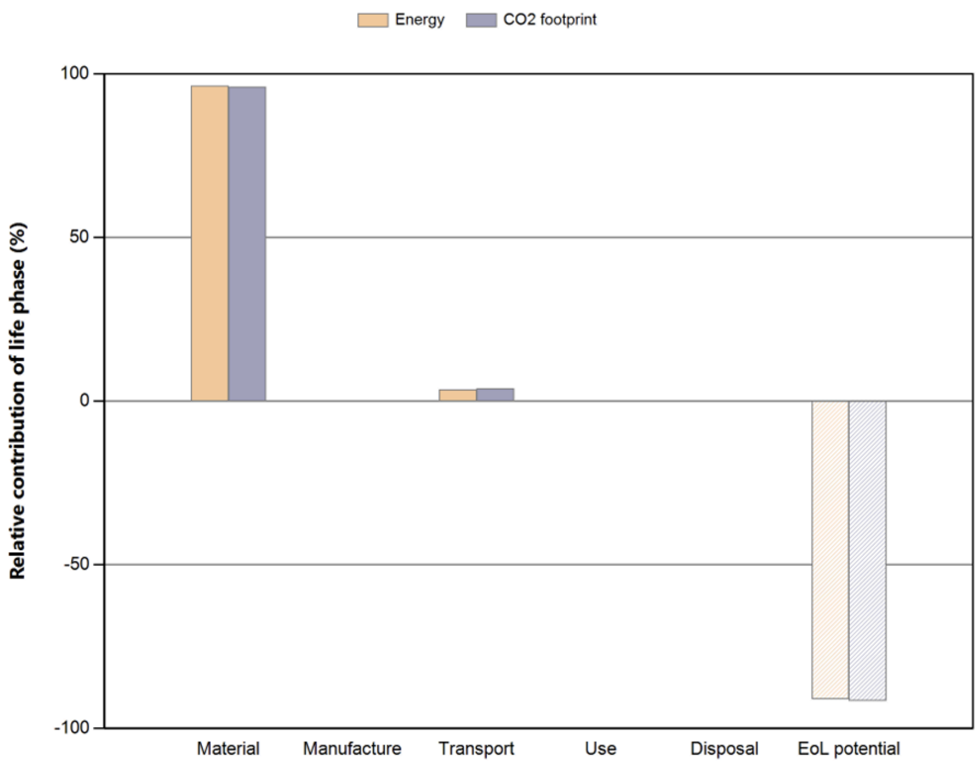
Although it was not moving perfectly- the OTV was veering to the right since the wheels were going at different speeds- everyone was just so proud that we were able to get the OTV to move forward. Everyone on the team had come together and played a part in making this happen so everyone was so proud and overjoyed.

Design Brief 7 - Sustainability

Eco Audit Report

Product name	OTV_HydroQuack
Country of use	United States
Product life (years)	0.25

Summary:



Energy details CO2 footprint details

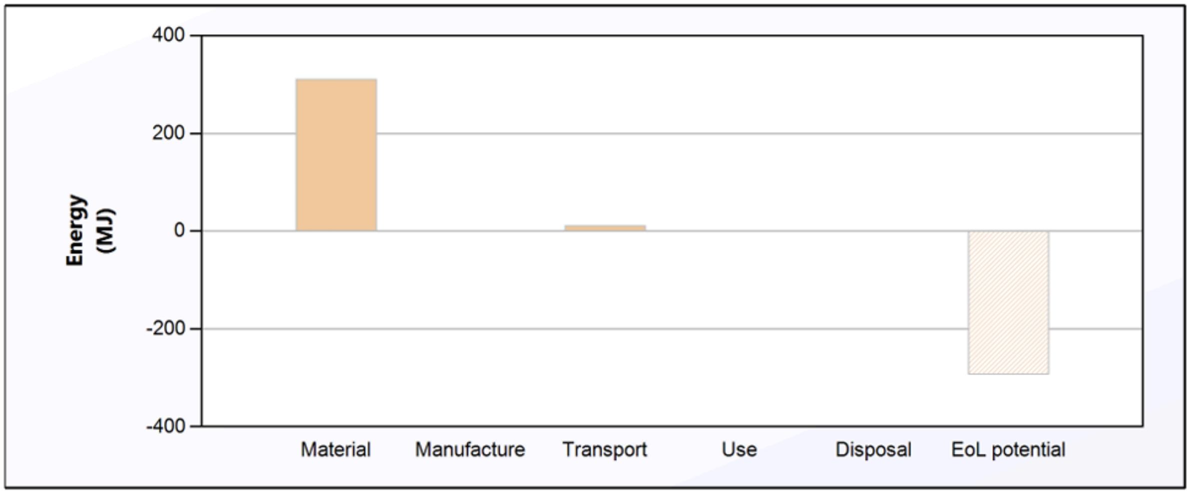
Phase	Energy (MJ)	Energy (%)	CO2 footprint (kg)	CO2 footprint (%)
Material	311	96.4	20.1	96.0
Manufacture	0	0.0	0	0.0
Transport	11.1	3.5	0.802	3.8
Use	0.262	0.1	0.0154	0.1
Disposal	0.344	0.1	0.0241	0.1
Total (for first life)	**322**	**100**	**20.9**	**100**
End of life potential	-294		-19.2	

NOTE: Differences of less than 20% are not usually significant.
See notes on precision and data sources.

Eco Audit Report

Energy Analysis

Summary



	Energy (MJ/year)
Equivalent annual environmental burden (averaged over 0.25 year product life):	1.29e+03

Detailed breakdown of individual life phases

Material:

Summary

Component	Material	Recycled content* (%)	Part mass (kg)	Qty.	Total mass (kg)	Energy (MJ)	%
Ultrasonic Sensor	Printed circuit board assembly	Virgin (0%)	0.01	2	0.02	2.6	0.8
Arduino Mega	Printed circuit board assembly	Virgin (0%)	0.037	1	0.037	4.8	1.5
Breadboard	Printed circuit board assembly	Virgin (0%)	0.04	2	0.079	10	3.3
H-Bridge	Printed circuit board assembly	Virgin (0%)	0.025	1	0.025	3.2	1.0
Motor	Fan	Virgin (0%)	0.044	2	0.088	22	7.0
Wires (Combined)	Cable	Virgin (0%)	0.03	1	0.03	2.7	0.9
Battery	NiMH, rechargeable battery (for laptops)	Virgin (0%)	0.26	1	0.26	2.4e+02	75.7
Screws (Combined)	High alloy steel, AF1410, solution treated & overaged	Virgin (0%)	0.03	1	0.03	10	3.2
Servo	Fan	Virgin (0%)	0.009	2	0.018	4.4	1.4
ESP-32 ML Cam	Printed circuit board assembly	Virgin (0%)	0.02	1	0.02	2.6	0.8
Support Pillars	Fir (l) (md)	Virgin (0%)	0.07	4	0.28	4.3	1.4
Acrylic	PMMA (cast sheet)	Virgin (0%)	0.04	1	0.04	4.5	1.5
Plastic - 3d printed	PLA (general purpose)	Virgin (0%)	0.09	1	0.09	4	1.3
Rubber Wheel Trim	Polyisoprene rubber (unreinforced)	Virgin (0%)	0.001	1	0.001	0.086	0.0
Total				21	1	3.1e+02	100

*Typical: Includes 'recycle fraction in current supply'

***User-defined material

Manufacture:

Summary

Component	Process	Amount processed	Energy (MJ)	%
Total				100

Transport:

Summary

Breakdown by transport stage

Stage name	Transport type	Distance (km)	Energy (MJ)	%
Battery	Ocean freight	1.3e+04	2.4	21.3
Screws	Ocean freight	1.3e+04	2.4	21.3
Motors	55 tonne (8 axle) truck	2.3e+03	1.7	14.9
Rubber WHeel Trim	Ocean freight	1.3e+04	2.4	21.3
ESP-32 ML Cam	Ocean freight	1.3e+04	2.4	21.3
Total		**5.4e+04**	**11**	**100**

Breakdown by components

Component	Mass (kg)	Energy (MJ)	%
Ultrasonic Sensor	0.02	0.22	2.0
Arduino Mega	0.037	0.41	3.7
Breadboard	0.079	0.87	7.8
H-Bridge	0.025	0.27	2.5
Motor	0.088	0.97	8.7
Wires (Combined)	0.03	0.33	3.0
Battery	0.26	2.8	25.2
Screws (Combined)	0.03	0.33	3.0
Servo	0.018	0.2	1.8
ESP-32 ML Cam	0.02	0.22	2.0
Support Pillars	0.28	3.1	27.6
Acrylic	0.04	0.44	3.9
Plastic - 3d printed	0.09	0.99	8.9
Rubber Wheel Trim	0.001	0.011	0.1
Total	**1**	**11**	**100**

Use:

Summary

Static mode

Energy input and output type	Electric to mechanical (electric motors)
Country of use	United States
Power rating (W)	24
Usage (hours per day)	0.5
Usage (days per year)	10
Product life (years)	0.25

Relative contribution of static and mobile modes

Mode	**Energy (MJ)**	**%**
Static	0.26	100.0
Mobile	0	
Total	**0.26**	**100**

Disposal:

Summary

Component	End of life option	Energy (MJ)	%
Ultrasonic Sensor	Landfill	0.004	1.2
Arduino Mega	Landfill	0.0074	2.2
Breadboard	Reuse	0.016	4.6
H-Bridge	Reuse	0.005	1.5
Motor	Reuse	0.018	5.1
Wires (Combined)	Reuse	0.006	1.7
Battery	Reuse	0.051	14.8
Screws (Combined)	Reuse	0.006	1.7
Servo	Reuse	0.0036	1.0
ESP-32 ML Cam	Reuse	0.004	1.2
Support Pillars	Downcycle	0.14	40.7
Acrylic	Downcycle	0.02	5.8
Plastic - 3d printed	Recycle	0.063	18.3
Rubber Wheel Trim	Landfill	0.0002	0.1
Total		**0.34**	**100**

EoL potential:

Component	End of life option	Energy (MJ)	%
Ultrasonic Sensor	Landfill	0	0.0
Arduino Mega	Landfill	0	0.0
Breadboard	Reuse	-10	3.5
H-Bridge	Reuse	-3.2	1.1
Motor	Reuse	-22	7.4
Wires (Combined)	Reuse	-2.7	0.9
Battery	Reuse	-2.4e+02	80.1
Screws (Combined)	Reuse	-10	3.4
Servo	Reuse	-4.4	1.5
ESP-32 ML Cam	Reuse	-2.6	0.9
Support Pillars	Downcycle	-0.028	0.0
Acrylic	Downcycle	-0.6	0.2
Plastic - 3d printed	Recycle	-2.7	0.9
Rubber Wheel Trim	Landfill	0	0.0
Total		**-2.9e+02**	**100**

Notes:

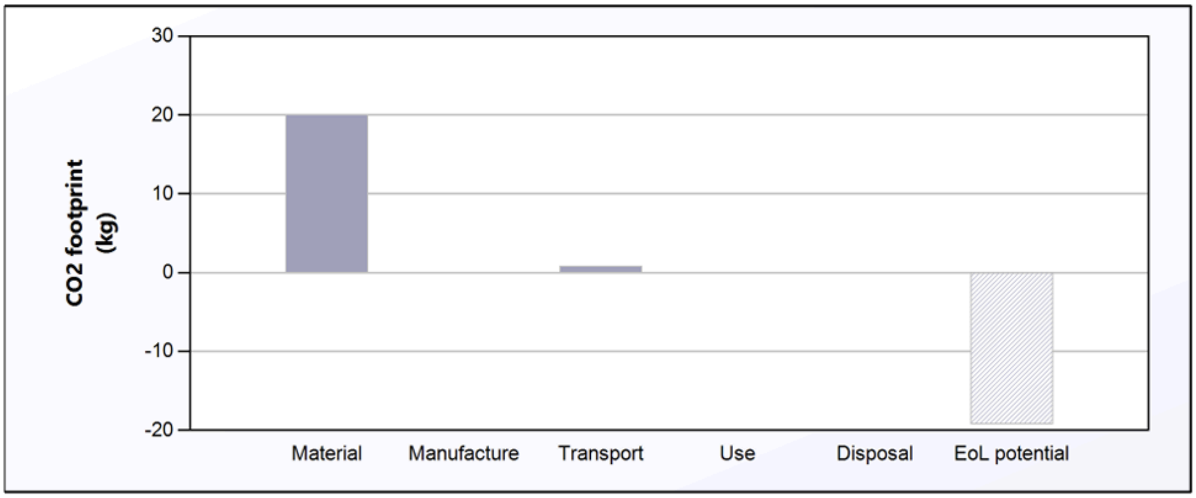
Summary

Ansys GRANTA EDUPACK

Eco Audit Report

CO2 Footprint Analysis

Summary



	CO2 (kg/year)
Equivalent annual environmental burden (averaged over 0.25 year product life):	83.7

Citations:

Part Model	Name / website. (Old Model/Final Model/Shared)
Breadboard	The Sonzie - grabcad (Shared)
Machine learning camera	Dũng Phan - grabcad (Old) Jaime Antunes - grabcad (Final)
Servo motor	Jules - grabCAD (Shared)
Servo Horn	Nikki Egleton - grabcad (Shared)
8mm tubing	McMaster Carr (Old)
Water pump	Ceodor Schumacher/ grabCad (Old)
Ultrasonic sensor	HowToMechatronics/Thangs (Old)

	Phan Nguyen Ngoc Hien/ grabCad (New)
Ball Caster wheels	Ahmed Elprince/grabcad (Shared)
Uno mega	Augustine Aelevanthara/grabcad (Shared)
H-Bridge	Casper Schramme/grabcad (Shared)
M3 Screw	Wakikiprod - printables (New)

Figure 42 - CAD Citations

Part Model	Method of Obtainment
Breadboard x2	Previously owned
Machine learning camera	Amazon - Storefront: WWZMDiB
Servo motor	Amazon - Storefront: WWZMDiB
Servo Horn	Included with Servo Motor
Ultrasonic sensor	Previously Owned
Ball Caster wheels	Amazon - Storefront: Leateck
Uno mega	Previously Owned
H-Bridge	ENES 100 Store
M3 Screw	Amazon - Storefront: Valued Global BUY
Motors	Pololu
Wood	Woodshop Scrap
Acrylic	ENES100 Shop/Provided

PLA (3d prints)	Provided
Cables and Wires	Provided

Figure 42 - Bill of Materials - Final OTV