ENGR 290: Introductory Engineering Team Design Project

Course Instructor: Dr. Rastko Selmic

Section N, Team #23

By:
Nabila Tabassum - 40109039
Alexandre Fontaine - 40158602
Rayan Alkayal - 40051210
Navid Mehrabi - 27552874

A design report submission in the fulfillment of the requirements of ENGR 290
Concordia University

December 17th, 2022

TABLE OF CONTENTS

1.0 PROBLEM DEFINITION

This report's objective is to conclude the Introductory Engineering Team Design Project course for members of team # 23. The planning, design, and analysis/debugging of the hovercraft design and code will be discussed, concluded by a self-evaluation of the project's management.

It will go in-depth into the thoughts and ideas that led to certain design choices in order to attempt the challenge of making a hovercraft that is physically and logically capable of competing in a maze. The hovercraft must be able to traverse the full length of the maze in under two minutes, while successfully going over three different obstacles that increase in height as the maze progresses.

1.1 Scope

The lecture component of this course taught us physical concepts of statics and dynamics. It can be broken down in two halves: the first half focusing on concepts of physics and a slight introduction into the hovercraft project, and the second half focusing on the project and recalling those concepts from the prior half. Principles of statics and dynamics are crucial in planning for what kinds of forces a hovercraft would experience as it traverses through a maze with obstacles.

The objective is to design and build a functioning autonomous hovercraft that is capable of following a specified course. The final score is represented by the following formula:

$$S = kD/(T * P)$$

Where:	S denotes the score
	T denotes the time (in seconds) to complete the maze
	P denotes the number of components utilized
	D denotes the distance along the track that was navigated autonomously
	k denotes a scaling factor for all teams

In order to maximize the score, the number of sensors should be at a minimum without compromising performance. Finally, under no circumstances should the hovercraft exceed 70% of the passage width in any dimension. That means neither the length, width, or diagonals of the hovercraft can exceed 45.5 cm.

1.2 Definition

A hovercraft is a type of vehicle that moves by creating an air cushion underneath itself. This is done by using a vertical fan that generates lift by leveraging the pressure difference between the atmosphere and the air bubble underneath (Woodford, 2021). For horizontal movement, a thrust fan is used to propel the hovercraft forward. In order to turn, a rudder is typically found behind the thrust fan that deflects air off to the side, resulting in a torque. It is crucial to have a balanced hovercraft, as tilt in any of the axes impacts its heading.

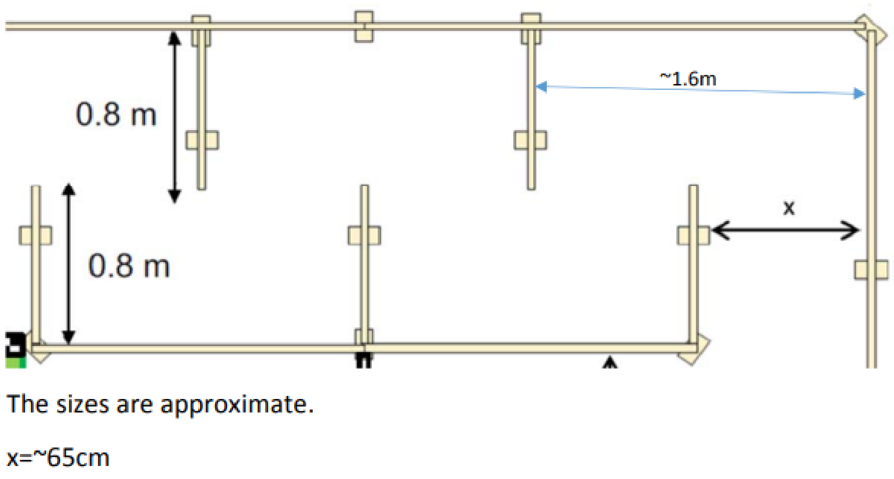


Figure 1. Maze dimensions

Figure 1 shows the expected maze dimensions. Using these, alongside with the supplied components, it is possible to create criterions for how large, heavy and fast the craft should be. It also steers how the code should be implemented.

2.0 SPECIFICATIONS

Components selected for the competition:

Sensors

- 2x HC-SR04 US sensors
- 1x SHARP 2Y0A21 IR sensor

Fans

- 2x MEC0251V1-00U-A99 fans

Battery

- 2x 460 mAh batteries in series

Servo

- 1x HS-311 servo

Using past knowledge from the technical assignments, it was decided that the IMU should not be implemented in the final design. The decision came due to a critical problem encountered during technical assignment #2 where the yaw would gradually offset itself from the origin. Thus, a triple sensor layout was opted for. With an infrared sensor in the front and two ultrasonic sensors perpendicular to the sides of the hovercraft.

Power draw estimates (referenced from datasheets):

1x IR + 2x US sensors	30 + 2 * 15 = 60mA at 5V
1x HS-311 servo	180mA at 5V
2x MEC0251V1 fans	2400mA at 5V
1x Arduino Nano	19mA at 5V
Total	2660mA at 5V = 13.3W
2x 460 mAh at 7.4V gives 30 minutes	

3.0 CONCEPTUAL DESIGN

3.1 System Functional Block Diagram

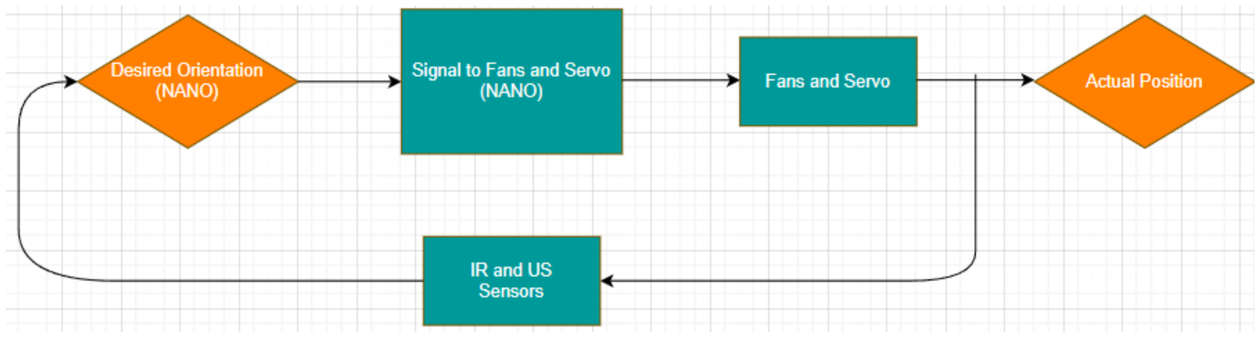
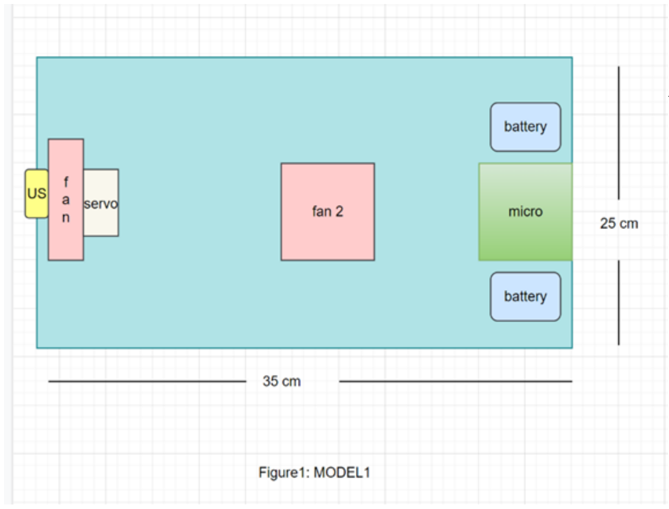


Figure 2. Functional block diagram

3.2 Alternative Concepts and Evaluation

3.2.1 Model 1



The first model was using one US sensor at the front. The US sensor was attached to the fan in front of the hovercraft. Both were installed to the servo and were able to rotate 180º. The sensor was calculating the distance as it rotated. Additionally, we used the lift fan in the middle, which helped to lift the hovercraft, by taking the airflow and releasing it through the holes of the skirt. Moreover, the microcontroller was placed at the back of the hovercraft alongside the two batteries placed on the two sides of the microcontroller.

3.2.2 Model 2

Model 2 represented the most innovative concept. The propulsion fan would push air into two tubes that would redirect the airflow to the sides. The idea was to be able to select which tube received the air flow in order to induce a torque. This, paired with a circular body, would allow the craft to easily turn, even if stuck in a corner. The main drawback was that this is an untested concept. It would have required the team to quickly pivot to another build in the event that proved unsuitable.

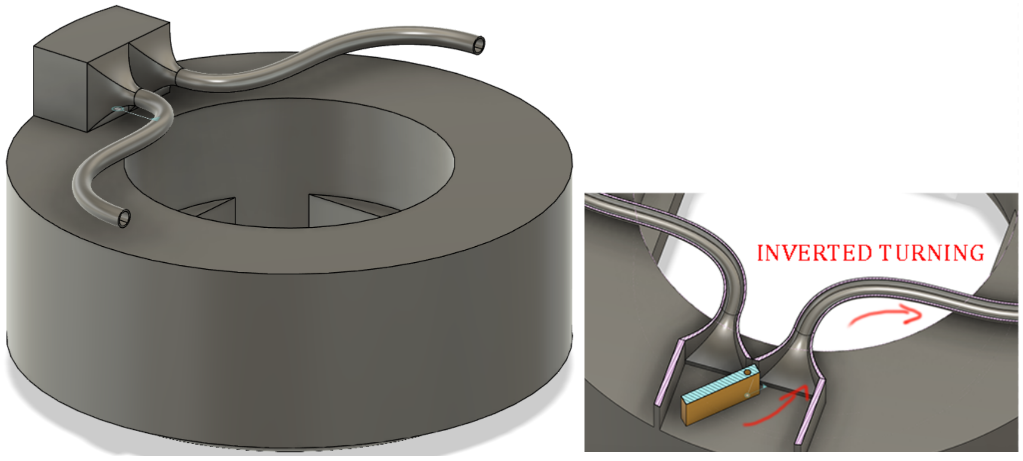
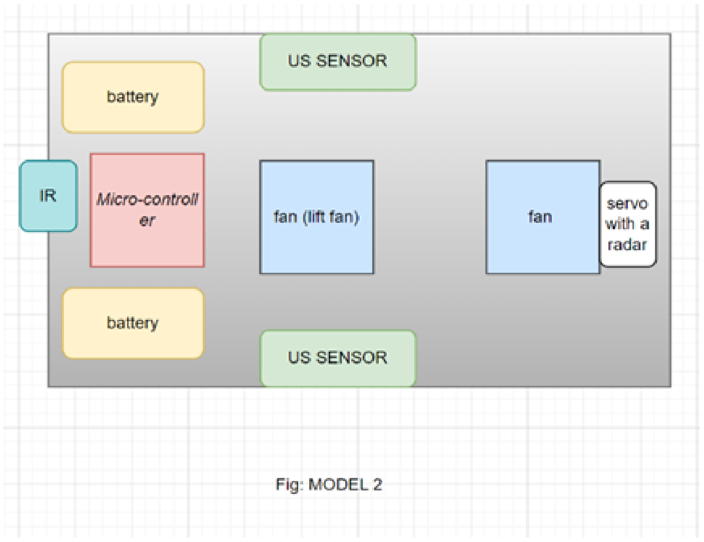


Figure 4. Conceptual Drawing of Model 2

3.2.3 Model 3

Model 3 was designed to have 3 sensors (1 IR and 2 US) and 2 fans. The base would be rectangular to make it easy to build. Like the other models, a lift fan would be placed horizontal and in the middle.A 2-fin rudder at the back would redirect the air and create a torque. Nonetheless, since the model was a propulsion fan at the back, its turning ability was limited.



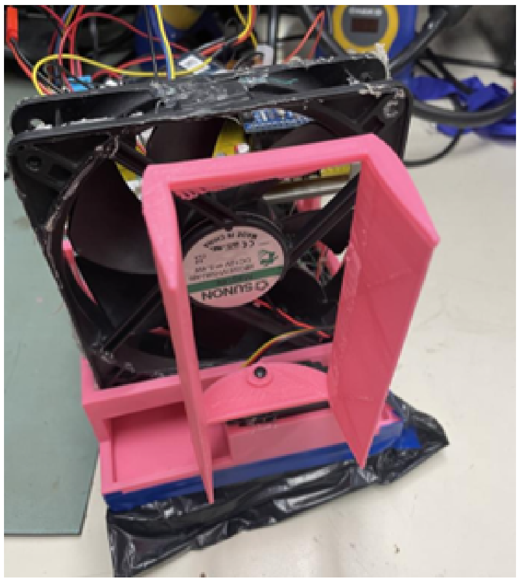


Figure 6. Initial rudder design

3.3 Decision Matrix

A decision was obtained by taking the highest value in the decision matrix, which ended up being model 3. The weight (Wt) was obtained by taking five important criteria from the hovercraft and assigned importance in a decreasing fashion. Here, 5 denotes the most important criteria, whereas 1 is the least important criteria. The value has been calculated by prioritizing its value out of 5.

Evaluation criteria	Model 1			Model 2			Model3		
	Wt1	Val1	Wt1* val1	Wt 2	Val2	Wt2* val2	Wt3	Val3	Wt3* val3
Skirt design	5	4	20	5	4	20	5	5	25
Position of fan	4	3	12	4	3	12	4	5	20
Turning ability	3	4	12	3	2	6	3	4	12
Sensors	2	3	6	2	4	8	2	4	8
Weight	1	4	4	1	3	3	1	3	3
Total			∑54			∑49			∑68

3.4 Final Model

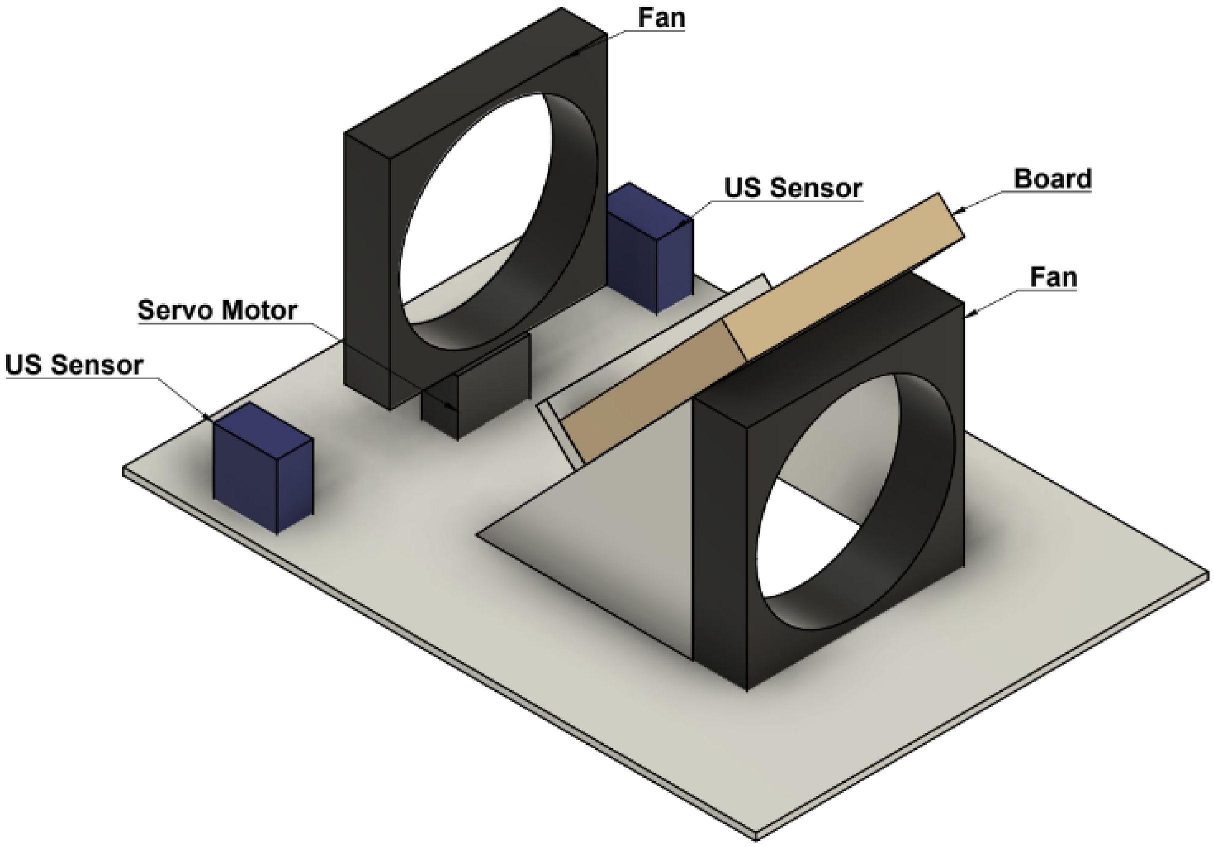
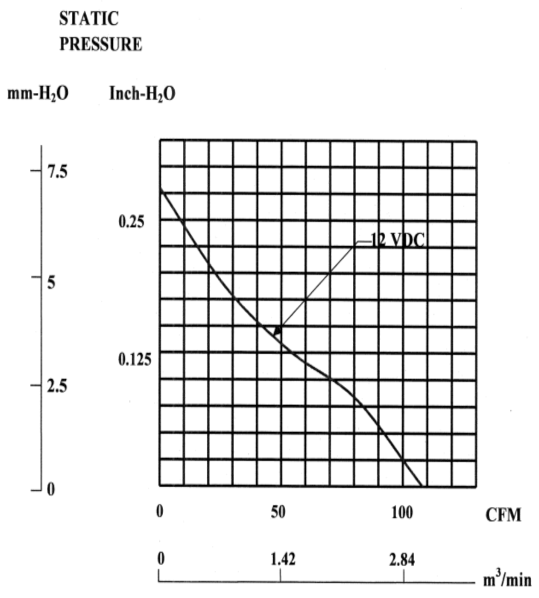


Figure 7. 3D Design of final implementation. Model 3 with certain modifications.

Model 3 ranked highest in the decision matrix and was chosen. The actual construction of the hovercraft, seen in figure 7, deviates from the model sketches. Several challenges were encountered throughout the experiment leading to these modifications. They are discussed throughout the report.

3.5 Design Calculations

Below is the CFM to pressure graph for the fan.



At 12V:

3.5 mm H2O = 9.8 Pa

Weight of hovercraft:

Board 116 g

- Fans 324 g
- Servo 35 g
- Sensors 22 g

Total: 497g

Needed surface area:

Area = m*g / P

= 497 * 9.8 / 9.8

= 497 cm^2

Final model surface area:

Area = l * w

= 35 * 25

= 875 cm^2

The final model's surface area is within specification in order to hover. Testing showed there was a considerable loss in efficiency and that the needed area was much larger than laid out by the calculations.

Pressure required to levitate:

$$\Delta p = mg/A_p$$

$$\Delta p = 9.81/0.85^2 = 13.58\ Pa$$

Flow rate required to clear a 3 mm obstacle:

$$Q = lh\sqrt{2\Delta p/\rho}$$

$$Q = 2 * 0.003\sqrt{(2 * 13.58)/1.18} = 0.02878\ m^3/s$$

$$Q = 60 * 0.02878 = 1.7268\ m^3/min$$

Center of Mass:

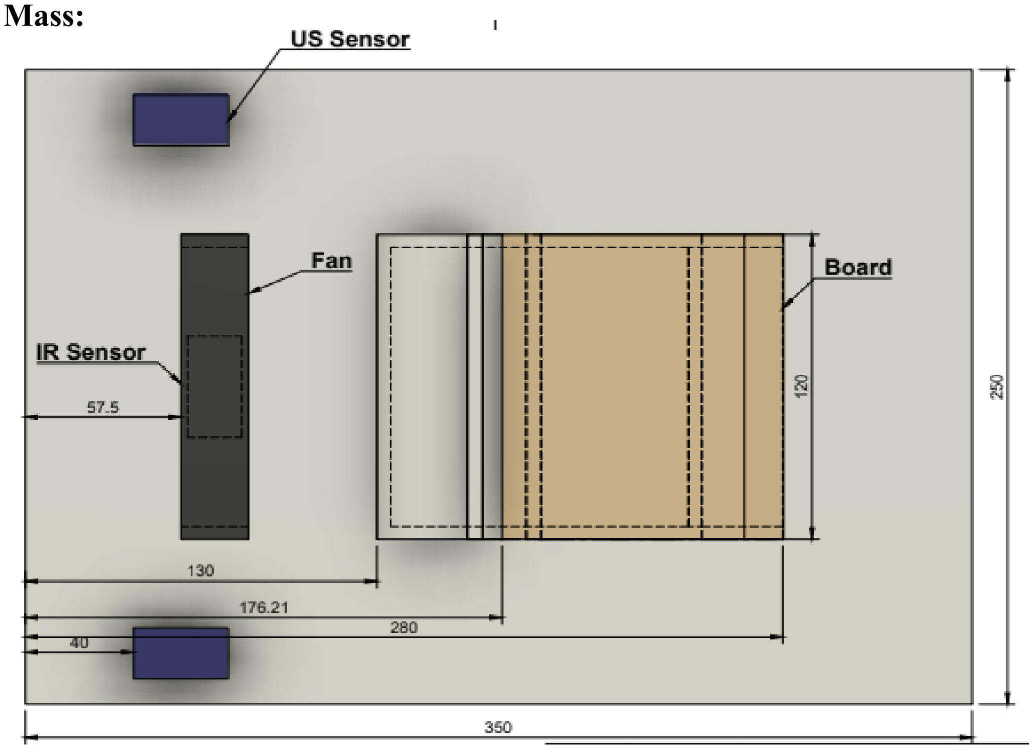


Figure 8. Dimensions of the final craft

Assumptions:

The craft was built symmetrical along the y-axis. As a result, the center of mass is in the middle of that axis. Those calculations have been omitted. It is assumed the components' center of mass is at their geometric center. The sensors' mass is sufficiently small and are not included in the calculations.

$X_{x\text{-axis}} = (M_{fan}X_{fan} + M_{servo}X_{servo} + M_{board}X_{board} + M_{liftFan}X_{liftFan} + 2 * M_{battery} + X_{battery}) / \text{Total}$

$X_{x\text{-axis}} = (100g*6.8cm + 43g*6.8cm + 116g*22.8cm + 100g*27cm + 2*(70g*34cm)) / 556g$

$\mathbf{X_{x\text{-axis}} = 19.92cm}$ $\mathbf{X_{y\text{-axis}} = 12.5cm}$

The center of mass is 2.4cm behind the geometric center. If this causes poor handling, the batteries can be quickly repositioned.

4.0 DETAILED DESIGN

4.1 Main Features and How They Work

4.1.1 Skirt

Throughout the project, it became apparent that proper skirt construction was essential. A soft skirt was chosen. Instead of making a large hole in the middle of the skirt, it was opted to do several small holes underneath.

These conclusions came after extensive testing. At first, a bottom plate was added that would protrude from the bottom of the craft and create an air filled cavity.

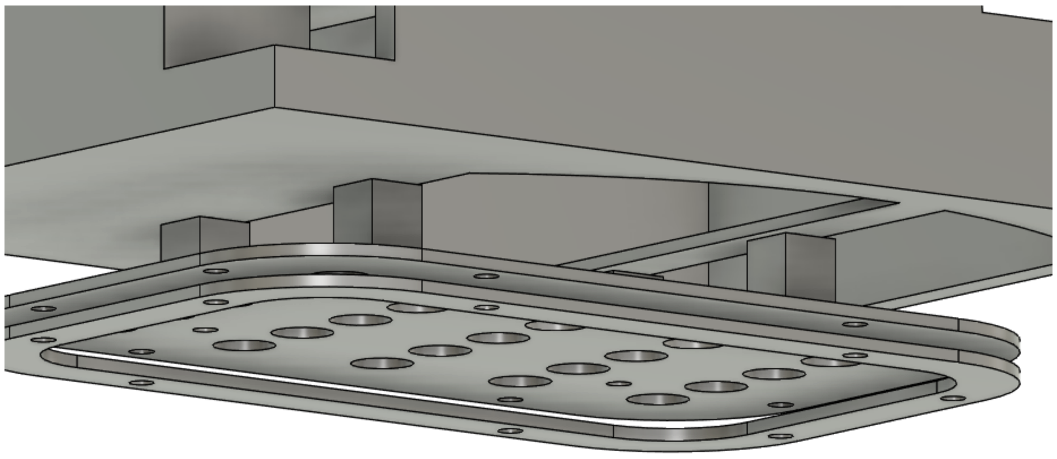


Figure 9. Initial bottom plate. The skirt would attach from the sides of the craft to the plate.

The skirt was made large enough that the bottom plate would not touch the ground. Contrary to initial calculations, the fan was not able to create enough pressure to keep the skirt fully inflated. Removing the plate allowed the craft to slightly hover. It was then a process of optimization to determine the optimal number, size and placement of holes. It appears the craft does not need to completely hover nor the skirt be fully inflated, in order to sufficiently reduce the friction and start moving.

4.1.2 Lift fan

Initially, the lift fan was placed horizontally. It blew air directly downwards to fill the skirt. However, it became immediately apparent that it was creating a torque in the opposite direction of its rotation. This made it difficult to keep the craft moving in a straight line.

The solution was to orient the fan vertically and have a conduit redirect the air flow downwards. The induced torque was then in the roll axis where the craft is much more stable.

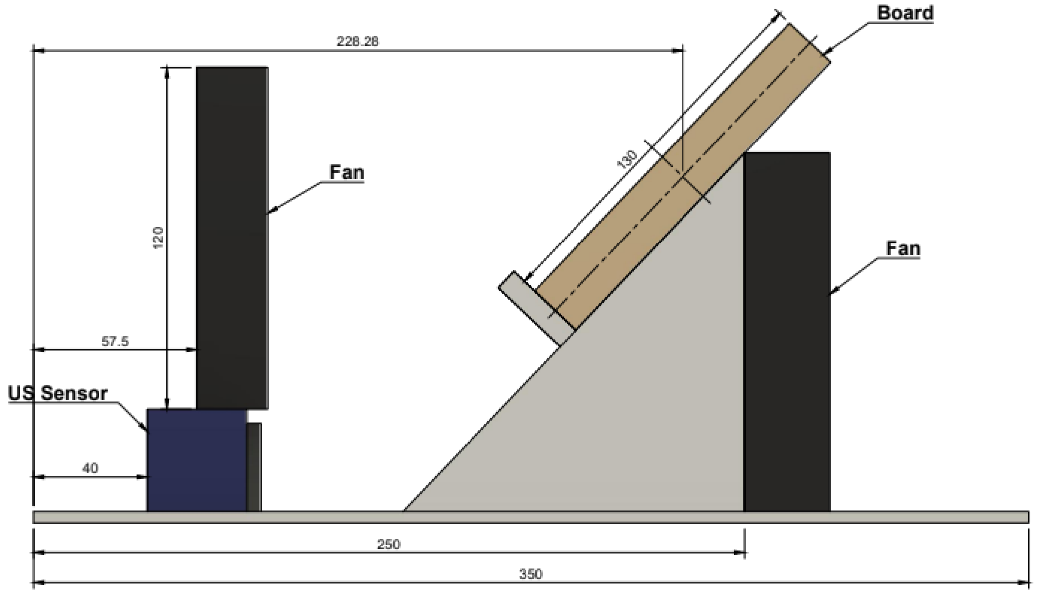


Figure 9. Side view showing the angled conduit to redirect air flow downwards

However, the energy loss redirecting the air was considerable. The craft was much slower afterwards which demanded a redesign of the skirt.

4.1.3 Propeller fan

The chosen Model 3 had the propulsion fan at the back with rotating rudders that would rotate hee craft. Initial testing made it clear that the setup could not orient the craft quickly enough to maneuver through the tight maze. As a result, the rudders were removed and the fan was placed directly on the servo at the front. This way, the craft could rotate from a standstill, even if placed in a corner.

4.1.4 Sensors

The final design utilizes two types of sensors: HC-SR04 ultrasonic and SHARP GP2Y0A21YK0F infrared sensors. Two ultrasonic sensors will be placed at ± 90 degrees on the front corners of the hovercraft, while an IR sensor is mounted on the servo at the front.

The infrared sensor is only good for readings between 10-80 cm in distance away from the sensor (SHARP). With the hovercraft's width of 25 cm, that means the IR sensor would not be suitable for the sides as the distance to the walls would be right on the edge at 10 cm. However, it should be good to use in the front. The ultrasonic sensors work by triggering an ultrasonic pulse and measuring distance based on the amount of time it takes for it to receive an echo of the pulse. During implementation, it was discovered that the ultrasonic sensors should work on a rotational basis, as they would otherwise interfere with each other.

4.1.5 Algorithm

The key concept behind the code was to have the craft react differently if it was going through a straight section versus in a turn. It should not correct intensely if it is close to a side with no obstruction in the front. It should correct aggressively as an obstruction approaches in the front. These requirements led to the formula in the figure below.

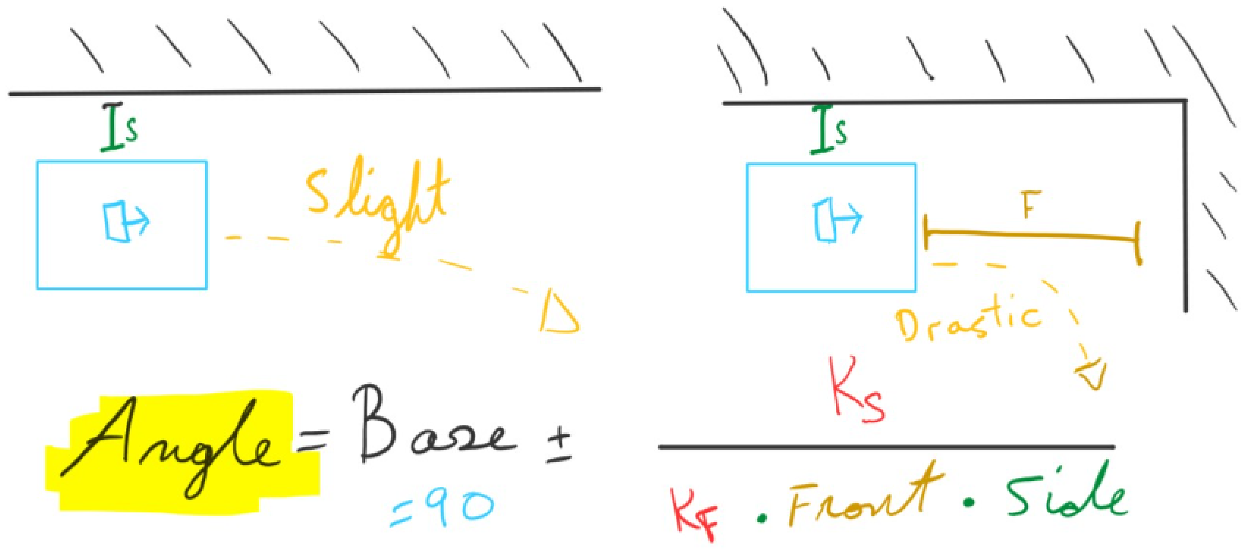


Figure 10. Conceptual sketch for the algorithm

In figure, Angle represents the angle of the servo that orients the fan. The advantage of this implementation is that two parameters, K_S and K_F , could be adjusted to modify how intensely the craft would react depending if it had a wall approaching in the front or not.

In order to estimate the initial values of the constants: the craft was placed in various scenarios it would experience in the maze. Distances and desired fan angles were measured. In

Matlab it was then possible to calculate the necessary values of K_S and K_F that would give similar results. This provided a starting point that could be refined with further testing.

4.2 Results of Experiments

During initial trials, the skirt proved unsuccessful. The hovercraft would not move unless nudged. The bottom skid plate was swiftly eliminated and replaced with a perforated plastic bag skirt. It was measured, cut, and taped under the hovercraft with an excess of 5 cm on each side to allow for it to inflate. After which, the hovercraft was flipped and holes were cut out at different points around the edges of the hovercraft to support it.

At this point, the lift fan had been positioned horizontally and an observation was made regarding the weight distribution. The center of mass was vastly off what the numbers would otherwise indicate. On one of the corners, the hovercraft would have a severe inclination to turn right. This was fixed by standing the lift fan upright and using styrofoam to redirect the air downwards. If using the right hand rule, it was noticed that when the lift fan was laid down flat to push air down, the net torque force would turn the base rightwards.

4.3 Manufacturing Results Including Photos

The manufactured result of the models ended up being very close, as the weight of the cables was not taken into account. All excess cables were taped together and put on one side of the hovercraft, and instead of placing one battery on each rear corner of the hovercraft as planned, they were placed together on the rear left side. This was not an ideal fix, but it was enough to allow for movement in straight and rightward/leftward directions. It was definitely a lot better than it was before, and actually managed to make a few turns.

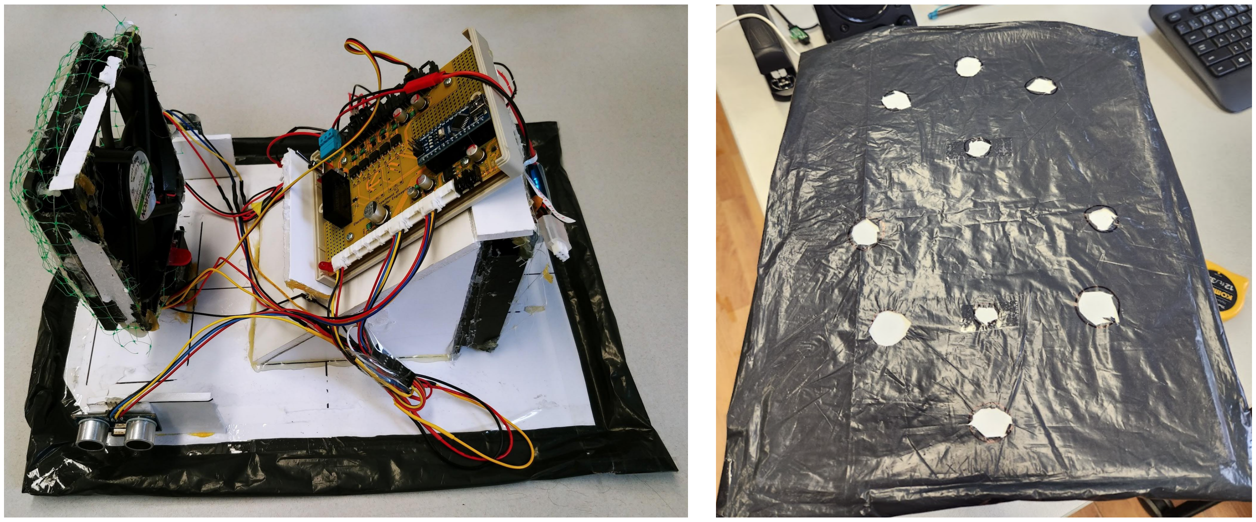


Figure 11. Final hovercraft build.

Standing the lift fan upright was the best modification to the hovercraft throughout this project. The hovercraft became very stable, and would no longer turn right on its axis. Extra attention was given to sealing the edges with hot glue, which was pushed in using a scrap piece of styrofoam to form an airtight seal.

Holes were gradually added to the bottom of the skirt as deemed fit, and that seemed to balance out the hovercraft quite well. They were measured using a Canadian quarter, which has a diameter of approximately 24 mm. There were 10 holes in total, and this design was certainly a lot better than with the bottom skid plate. In principle, the skid plate would be fixed to the base of the hovercraft, which would allow the skirt to inflate and lift the whole body up, minimizing the friction resistance and allowing the thrust fan to propel the hovercraft forward.

4.4 Gantt Chart

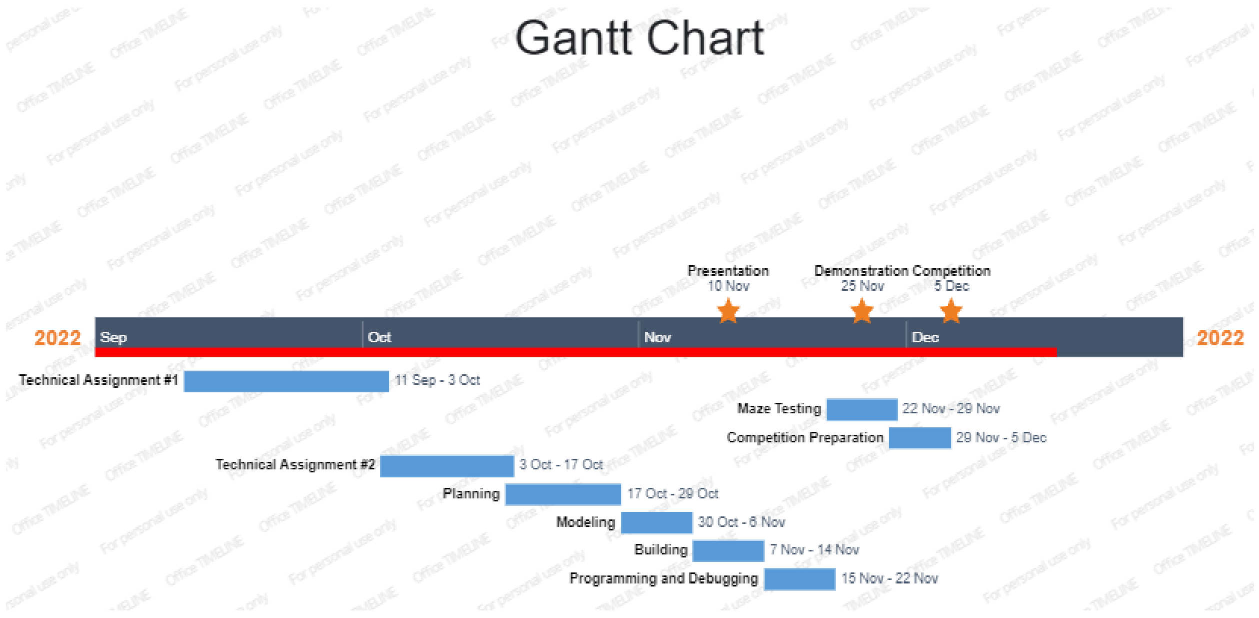


Figure 12. Gantt Chart

5.0 PERFORMANCE EVALUATION

On the day of the competition, the hovercraft performed as expected from previous testing. The code had parameters to allow for quick modifications for testing. The hovercraft performed best when the response intensity was mild. It was a delicate balance between not overreacting while going through a straight section and being able to veer away from corners. Unfortunately, it overcorrected during both attempts. On the first run it made it through two lengths. On the second run it ended up turning back around completely.

A thing to note is that the hovercraft had no problems navigating over any of the obstacles. This is a clear indicator that the skirt was well designed, which was a relief after the number of revisions it had gone through.

The size of the batteries was a concern, and because the team stayed overnight the prior night testing and revising the algorithm, when the time for the competition came the batteries were not fully charged. It was observed during testing that the hovercraft performed better when the batteries were more charged, and that is expected because the fans would run better on higher voltages.

6.0 CONCLUSION

The experience obtained from completing the project was great. Working with other engineering students with different backgrounds allows for learning opportunities in fields which people have little to no experience in. Communication is crucial in engineering design projects, as there can be several components that different teams develop, but have to do so in a way that brings a single refined product forward.

To tackle this project alone with no background knowledge of any of the components (i.e. programming, construction, and/or circuitry) would be an impossible feat. Even if done, there will be some oversights which would otherwise be spotted and addressed in a team setting. Another key take-away is that real-world testing is just as crucial as running calculations and simulations to obtain theoretical data. It is easier to forget certain things such as the weight of the cables, which could have a huge impact on performance in a weight-sensitive application such as building a hovercraft.

Modular code makes the final stages less stressful as further modifications could be as easy as changing numerical values, which was exhilarating to have the day before the competition when fine-tuning took place.

7.0 REFERENCES

Airbus (2022, July 1). *Safety innovation #2: The Fenestron*. https://www.airbus.com/en/newsroom/stories/2022-07-safety-innovation-2-the-fenestron

ITead Studio (2010, March 11). *Ultrasonic ranging module : HC-SR04*. Electro Schematics https://www.electroschematics.com/wp-content/uploads/2013/07/HC-SR04-datasheet-version-2.pdf

Jackson, D. (2001, July 8). *Helicopter Yaw Control Methods*. Aerospace Web. http://www.aerospaceweb.org/question/helicopters/q0034.shtml

SHARP *GP2Y0A21YK0F Datasheet.* Sheet No. E4-A00201EN https://global.sharp/products/device/lineup/data/pdf/datasheet/gp2y0a21yk_e.pdf

Woodford, C. (2021, April 21). *How do hovercraft work?* Explain That Stuff. https://www.explainthatstuff.com/hovercraft.html

8.0 APPENDIX

8.1 Meetings

Every Sunday evening, a 30 minute group meeting was held to discuss the task breakdown for the following week.

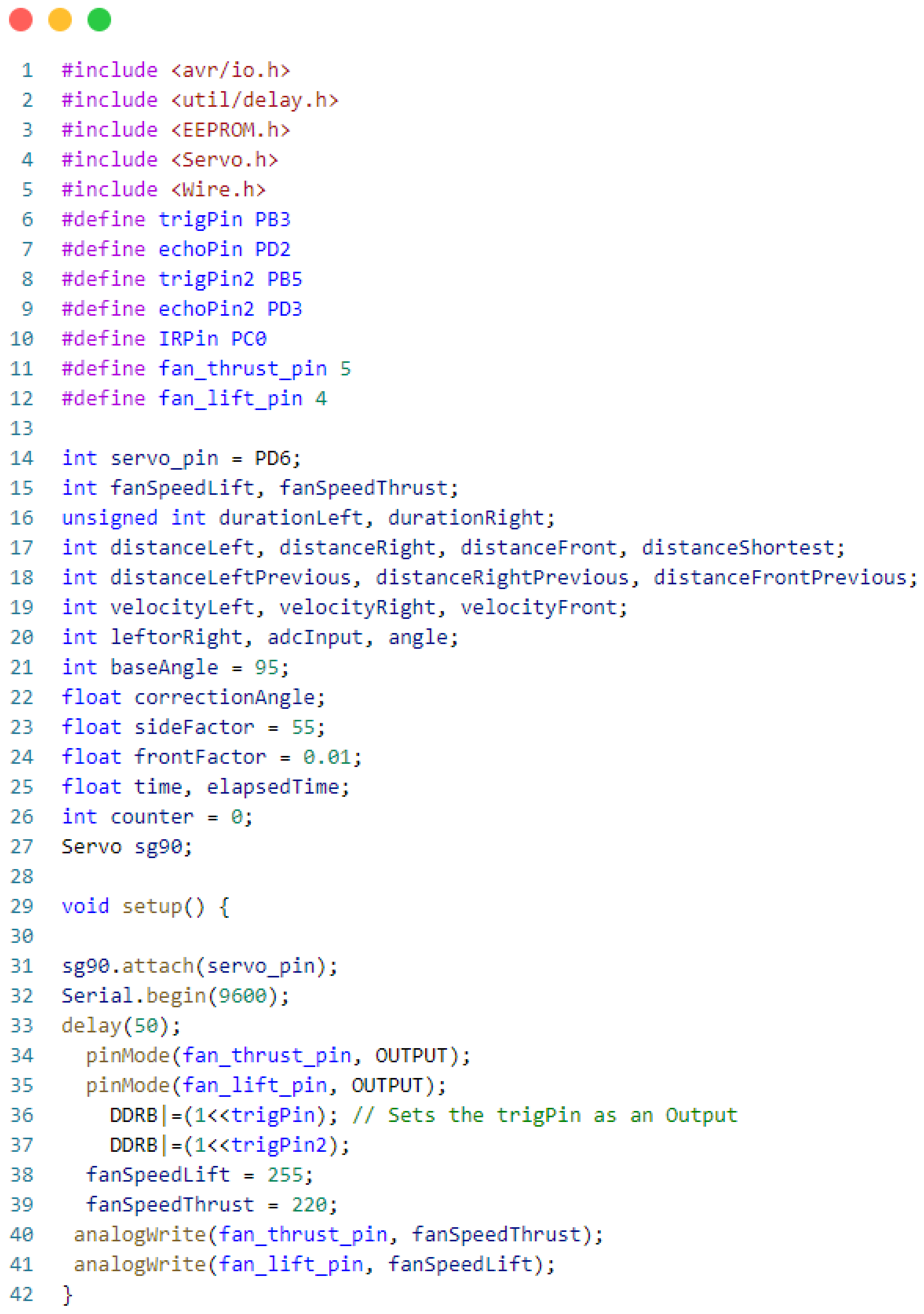
- For 13 weeks of instruction (minus the first week as teams had not been announced yet):
    - 12 * 30 = 360 minutes collectively as a group

Total time spent on various tasks:

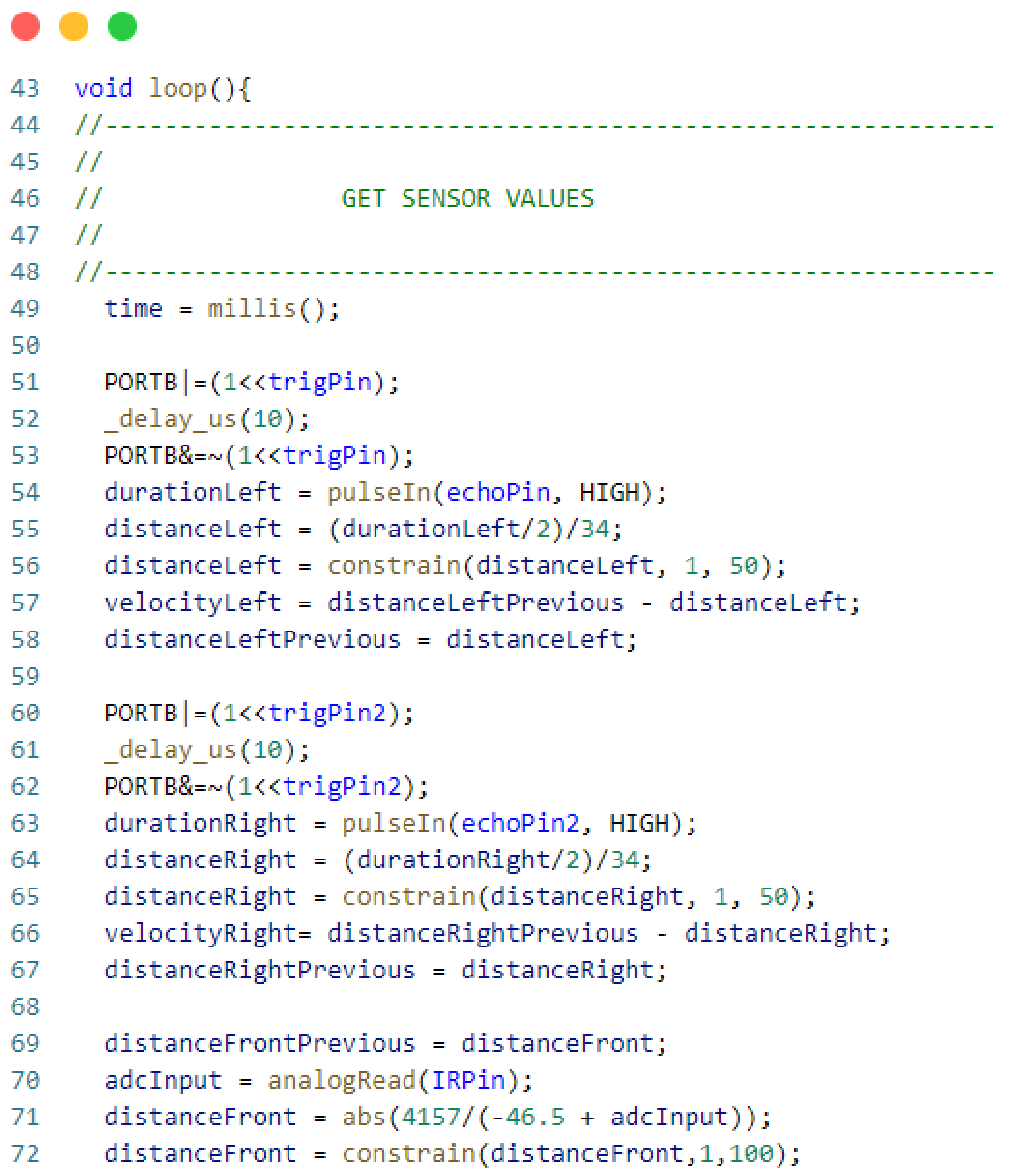
- Modeling: 12 hours during week 10
- Prototyping: 40 hours over weeks 10 and 11
- Programming: 36 hours during week 11
- Building: 24 hours during week 12
- Testing and revising: 40 hours during week 12
- Report: 20 hours during week 13 and beyond

Every member was able to partake in each stage of the hovercraft design and build process. An example of that would be how the code was revised several times by different members who had different implementations, and the most optimal code was selected. All work was done at the Concordia IEEE lab as well as online through Discord. 3D models were created using Fusion 360 running with student licenses for cloud-share collaboration on models.

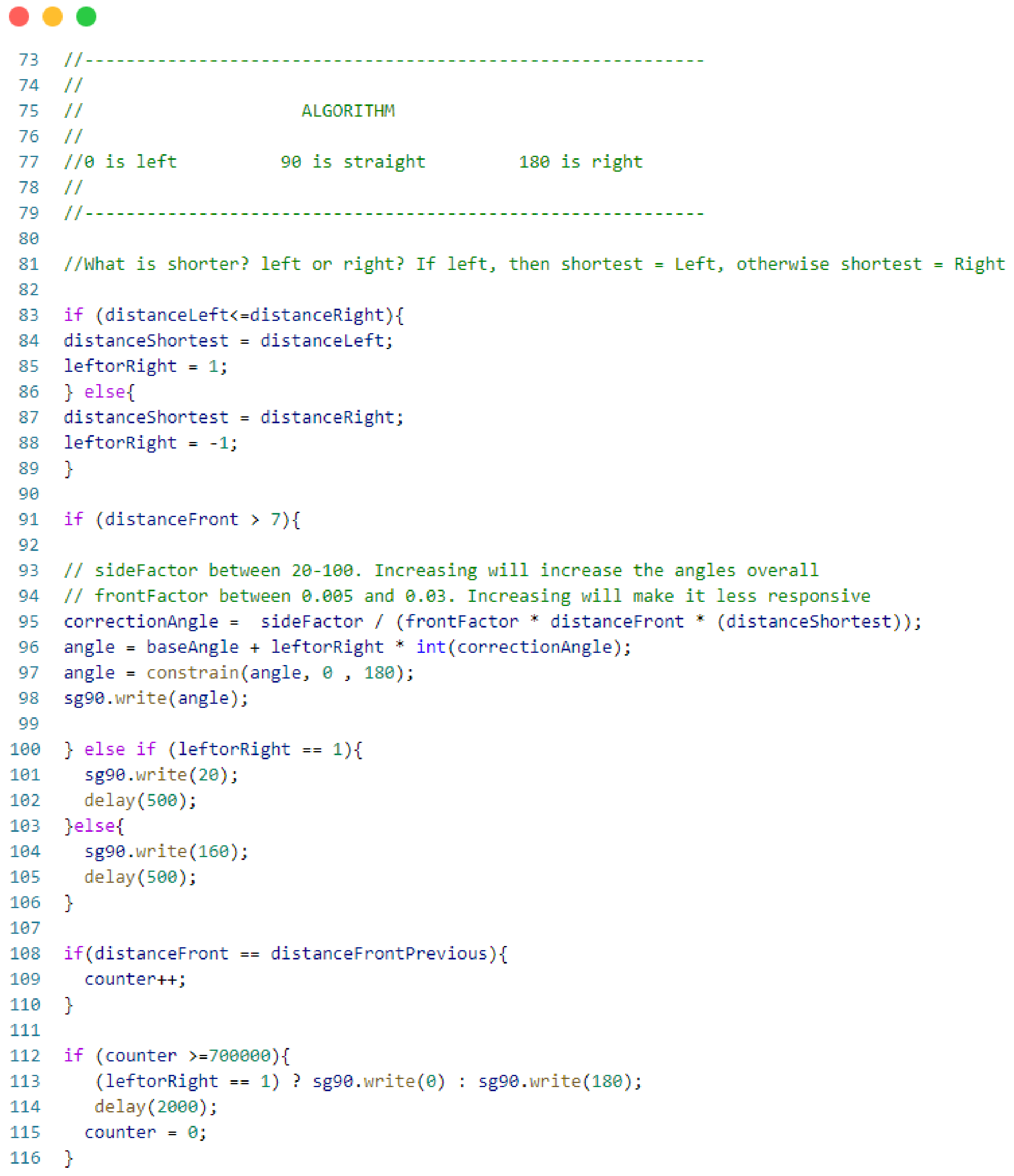
8.2 Code



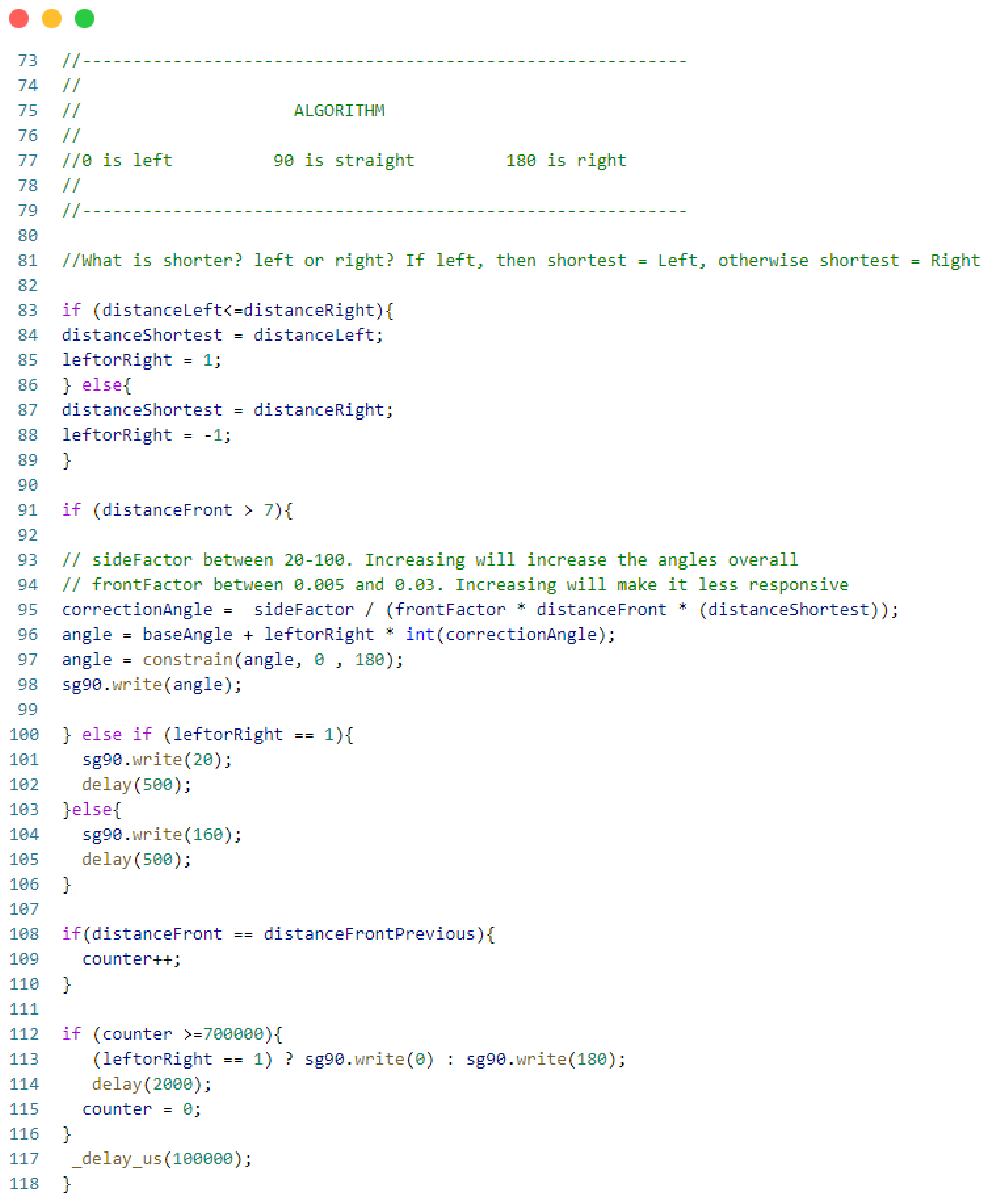
```
#include <avr/io.h>
#include <util/delay.h>
#include <EEPROM.h>
#include <Servo.h>
#include <Wire.h>
#define trigPin PB3
#define echoPin PD2
#define trigPin2 PB5
#define echoPin2 PD3
#define IRPin PC0
#define fan_thrust_pin 5
#define fan_lift_pin 4

int servo_pin = PD6;
int fanSpeedLift, fanSpeedThrust;
unsigned int durationLeft, durationRight;
int distanceLeft, distanceRight, distanceFront, distanceShortest;
int distanceLeftPrevious, distanceRightPrevious, distanceFrontPrevious;
int velocityLeft, velocityRight, velocityFront;
int leftorRight, adcInput, angle;
int baseAngle = 95;
float correctionAngle;
float sideFactor = 55;
float frontFactor = 0.01;
float time, elapsedTime;
int counter = 0;
Servo sg90;

void setup() {

sg90.attach(servo_pin);
Serial.begin(9600);
delay(50);
  pinMode(fan_thrust_pin, OUTPUT);
  pinMode(fan_lift_pin, OUTPUT);
    DDRB|=(1<<trigPin); // Sets the trigPin as an Output
    DDRB|=(1<<trigPin2);
  fanSpeedLift = 255;
  fanSpeedThrust = 220;
 analogWrite(fan_thrust_pin, fanSpeedThrust);
 analogWrite(fan_lift_pin, fanSpeedLift);
}
```



```
void loop(){
//------------------------------------------------------------
//
//                GET SENSOR VALUES
//
//------------------------------------------------------------
  time = millis();

  PORTB|=(1<<trigPin);
  _delay_us(10);
  PORTB&=~(1<<trigPin);
  durationLeft = pulseIn(echoPin, HIGH);
  distanceLeft = (durationLeft/2)/34;
  distanceLeft = constrain(distanceLeft, 1, 50);
  velocityLeft = distanceLeftPrevious - distanceLeft;
  distanceLeftPrevious = distanceLeft;

  PORTB|=(1<<trigPin2);
  _delay_us(10);
  PORTB&=~(1<<trigPin2);
  durationRight = pulseIn(echoPin2, HIGH);
  distanceRight = (durationRight/2)/34;
  distanceRight = constrain(distanceRight, 1, 50);
  velocityRight= distanceRightPrevious - distanceRight;
  distanceRightPrevious = distanceRight;

  distanceFrontPrevious = distanceFront;
  adcInput = analogRead(IRPin);
  distanceFront = abs(4157/(-46.5 + adcInput));
  distanceFront = constrain(distanceFront,1,100);
```



```
//-----------------------------------------------------------
//
//                      ALGORITHM
//
//0 is left          90 is straight         180 is right
//
//-----------------------------------------------------------

//What is shorter? left or right? If left, then shortest = Left, otherwise shortest = Right

if (distanceLeft<=distanceRight){
distanceShortest = distanceLeft;
leftorRight = 1;
} else{
distanceShortest = distanceRight;
leftorRight = -1;
}

if (distanceFront > 7){

// sideFactor between 20-100. Increasing will increase the angles overall
// frontFactor between 0.005 and 0.03. Increasing will make it less responsive
correctionAngle =  sideFactor / (frontFactor * distanceFront * (distanceShortest));
angle = baseAngle + leftorRight * int(correctionAngle);
angle = constrain(angle, 0 , 180);
sg90.write(angle);

} else if (leftorRight == 1){
  sg90.write(20);
  delay(500);
}else{
  sg90.write(160);
  delay(500);
}

if(distanceFront == distanceFrontPrevious){
  counter++;
}

if (counter >=700000){
   (leftorRight == 1) ? sg90.write(0) : sg90.write(180);
   delay(2000);
  counter = 0;
}
```



```
//------------------------------------------------------------
//
//                       ALGORITHM
//
//0 is left          90 is straight         180 is right
//
//------------------------------------------------------------

//What is shorter? left or right? If left, then shortest = Left, otherwise shortest = Right

if (distanceLeft<=distanceRight){
distanceShortest = distanceLeft;
leftorRight = 1;
} else{
distanceShortest = distanceRight;
leftorRight = -1;
}

if (distanceFront > 7){

// sideFactor between 20-100. Increasing will increase the angles overall
// frontFactor between 0.005 and 0.03. Increasing will make it less responsive
correctionAngle =  sideFactor / (frontFactor * distanceFront * (distanceShortest));
angle = baseAngle + leftorRight * int(correctionAngle);
angle = constrain(angle, 0 , 180);
sg90.write(angle);

} else if (leftorRight == 1){
  sg90.write(20);
  delay(500);
}else{
  sg90.write(160);
  delay(500);
}

if(distanceFront == distanceFrontPrevious){
  counter++;
}

if (counter >=700000){
   (leftorRight == 1) ? sg90.write(0) : sg90.write(180);
   delay(2000);
  counter = 0;
}
 _delay_us(100000);
}
```