Main Project: Micromouse

In this devlog, I will document the processes I followed to complete the micromouse project. This includes excerpts from various workshops that significantly contributed to different aspects of the build, as well as my independent work outside of the workshops. Both the workshops and deeper design details will be referenced and linked to their respective pages.

Initial Thoughts

From the outset thismodule looked to allow a lot of creative thinking and problem solving to reach the literal and figurative goals. I knew I wanted to add a compass and found the magnetometer GY-271 HMC588L. This would become a stretch goal later on.

Introduction to Components

First we were introduced to the basic components that would comprise our micromouse.

1. Pi Pico - Micro Controller
2. TCRT500 - Infra red sensor
3. L7805 - Linear voltage regulator
4. L293D - H-Bridge

We went about connecting these up with breadboards and using the Arduino's integrated development enviroment (IDE) and installed a plugin in order to use the

Pi Pico with it. From here we tested each individual components to ensure they worked correctly and for debugging experience.

https://prod-files-secure.s3.us-west-2.amazonaws.com/6d327d71-a800-4bb7-a305-2462de459e8c/dc5a5b70-5988-48d9-8b37-66a5bb072acc/Video.mov

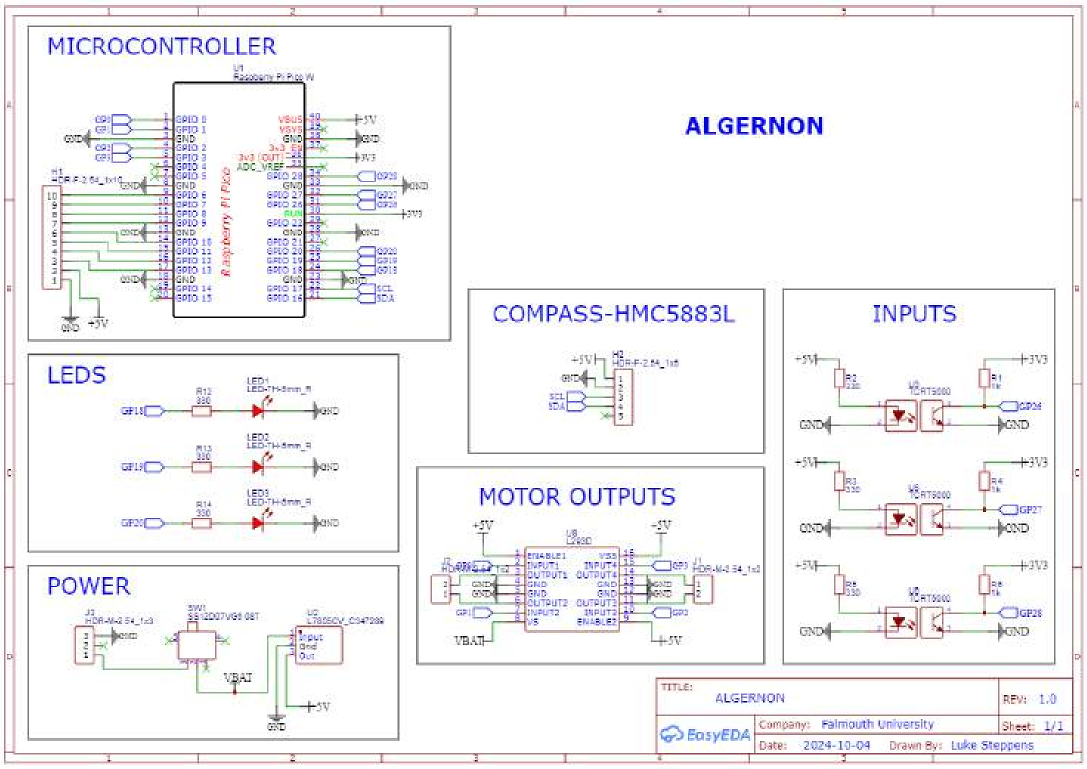
I go into more detail about each component in the corresponding workshop page Workshop 1 - Build a Robot

---

Printed Circuit Boards - Schematics

With our components chosen and tested we moved on to creating a circuit board for them. The first stop was to use the program EasyEDA to create a schematic and add all our components to it. Once added every point needed to be connected to the correct place on the Pi Pico.

Components were often found in the user contributions section of the search.



This went well except for one mistake which was that on the compass I took information from the sellers page that was incorrect. The 5V on the compass should be 3.3V, I have learned to always go to the datasheet for each item rather than assume every seller has correct information.

I go into more detail of this process on the corresponding workshop page

Workshop 2 - PCB Design (Schematics)

Printed Circuit Boards - Layout

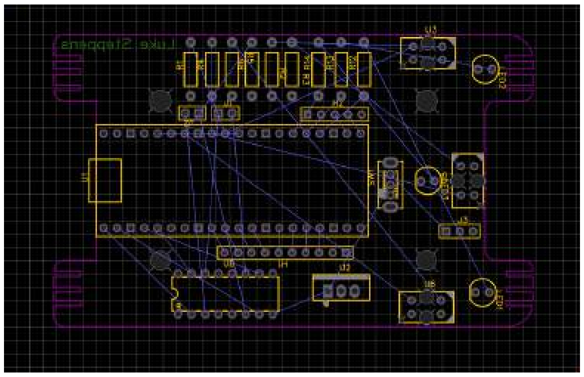
Before we could arrange the layout of the circuit board we need a shape for it to fit within, we used the program Fusion 360 to design the general outline of our Micro

Mouse base.

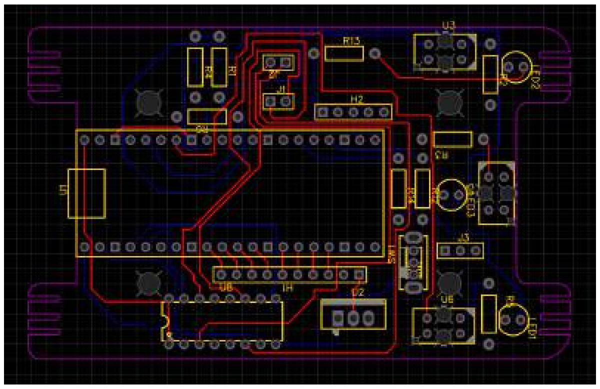
The process is detailed here PCB Base - CAD - Fusion 360

With the file exported as a DXF file it was then imported into EasyEDA. This is where the project took on a puzzle-like element as we had to arrange all the components within the confines of the mouse shape, and have them fit in a logical manner. Such as the TCRT5000's be near the edges because they need to access light, this sounds simple enough but when twinned with having to connect up all the components tracks it becomes a bit of a juggling act to find a happy medium between the two.

Initial arrangement without considering track between components.



Final arrangement after considering component placement and track lines.



I created a problem of my own making when I i thought it would be clever to make the track lines thinner to make it easier to navigate around the board, this threw up fault after fault and I had to go back to the beginning and start again before landing on the final track design above. All that was left was to add some designs and info onto the underside of the PCB

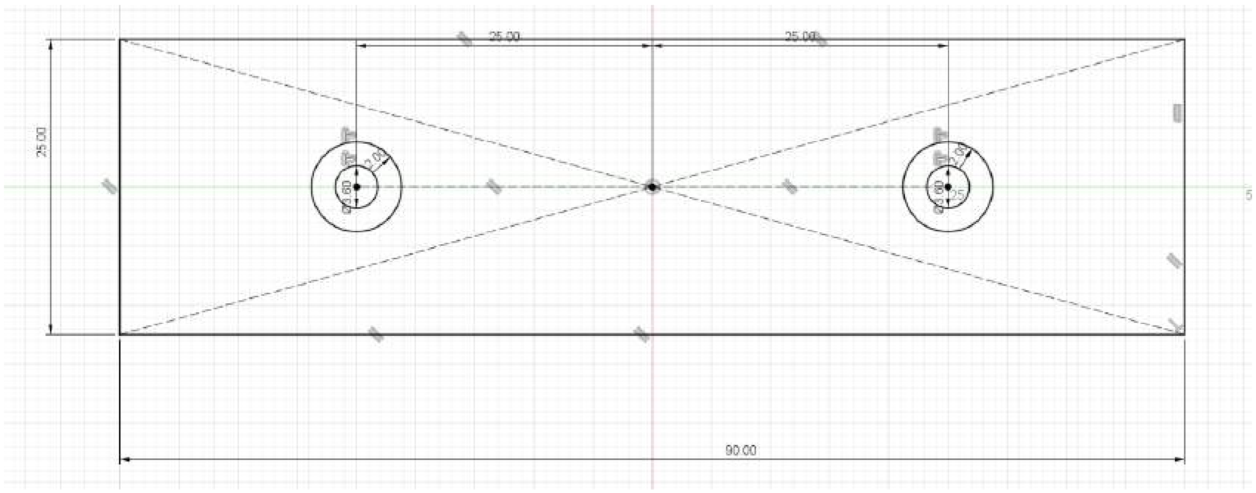
.

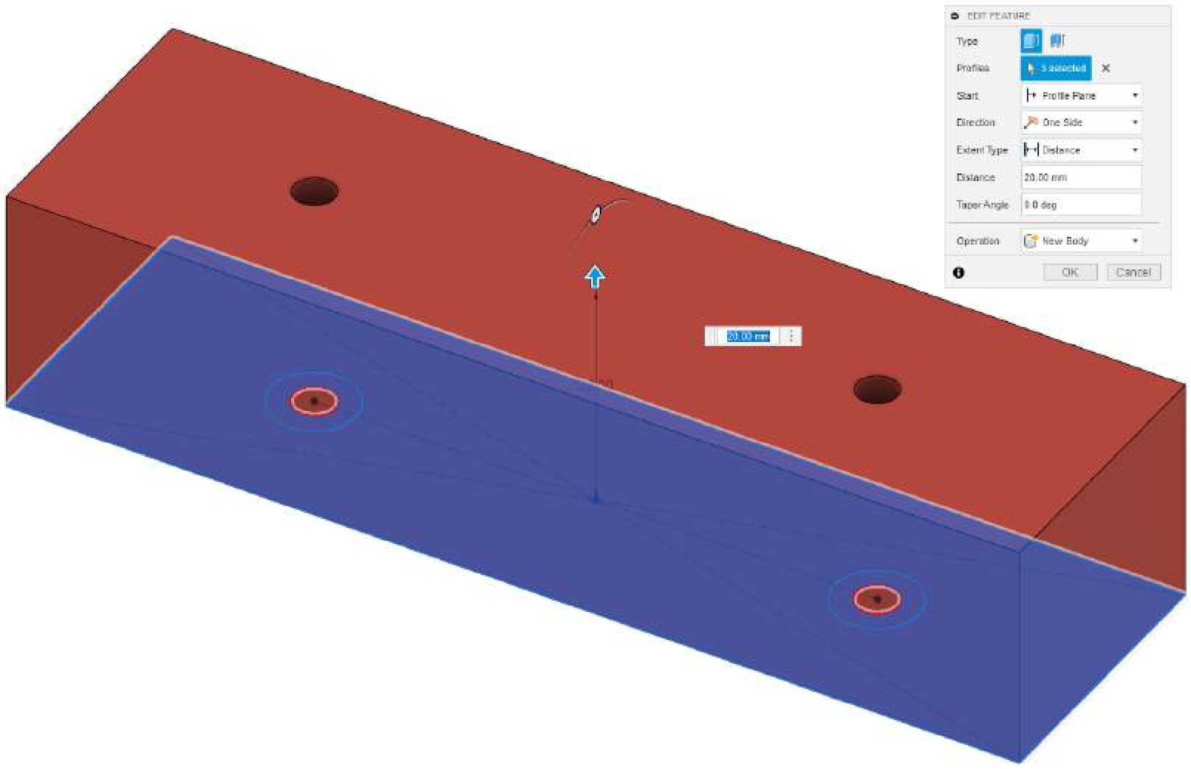
I go into further detail on these processes in the corresponding workshop

Workshop 3 - PCB Design (Layout)

Motor Mount - Fusion 360

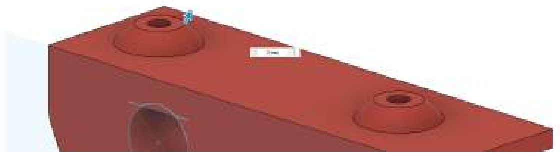
Designing the mounts had to follow some basic rules in order to make sure they would fit both the DC motors and PCB's. Firstly a sketch is placed and we begin building a basic block with holes with which to attach the PCB. The centre rectangle was made 90mm by 25mm giving lots of space, and then I used the construction line tool to find the spots 25mm away from the centre east and west. These would be the places for the PCB holes, the circle tool was used at 3.60 diameter and then used the offset tool to add another circle 2mm larger around it. The rectangle was extruded 20mm with the two circles deselected which leaves the holes going through the block.

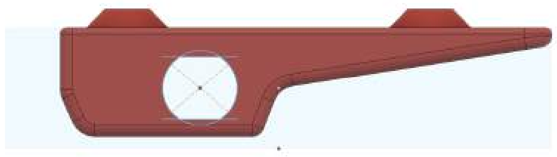




Type	
Profiles	3 selected
Start	Profile Plane
Direction	One Side
Extent Type	Distance
Distance	20.00 mm
Taper Angle	0.0 deg
Operation	New Body

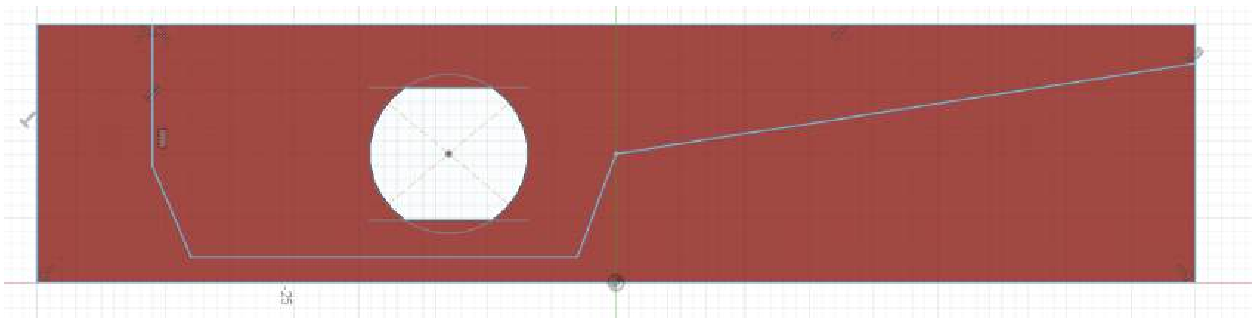
We were told that between the PCB and the mount we should leave 3mm for clearance, so the circles created by the offset on the surface were extruded to 3mm. Then I used the chamfer tool at 3mm on those extrusions to give a ramping down effect.

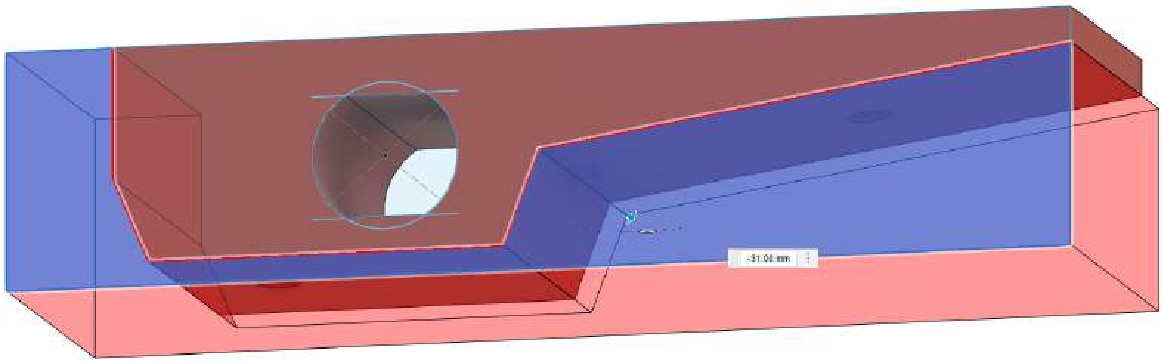




From here I used the fillet tool and rounded off most of the edges to 4mm.

The next step was to carve out the hole for the DC motor and to create the shape of the mount itself. Making the general shape was easy enough, make a sketch on the side and then used the line tool just selecting points then extruding through. The motor hole had to follow the specification of its real life counterpart so on the same sketch a circle of 12mm was made with a square of 10mm was drawn from the circles centre, this left the centre part ready for the extrusion process.

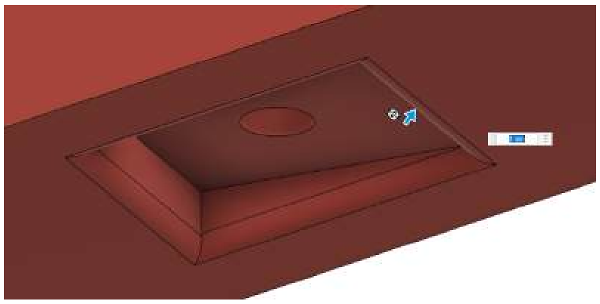




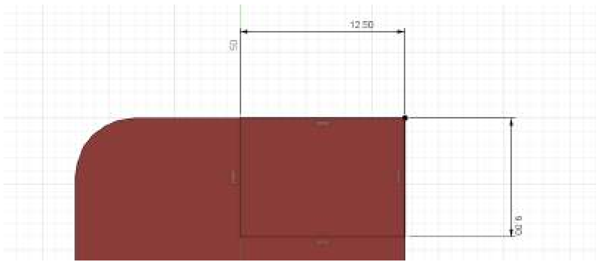
At this point I noticed the cylindrical hole going through on the right would where the object is at an angle, so if a screw or bolt had to sit against it, it wouldnt be able to tighten correctly.

https://prod-files-secure.s3.us-west-2.amazonaws.com/6d327d71-a800-4bb7-a305-2462de459e8c/830f8a8e-e3c7-4f86-b5bd-2d26a9f33255/IMG_3076.mov

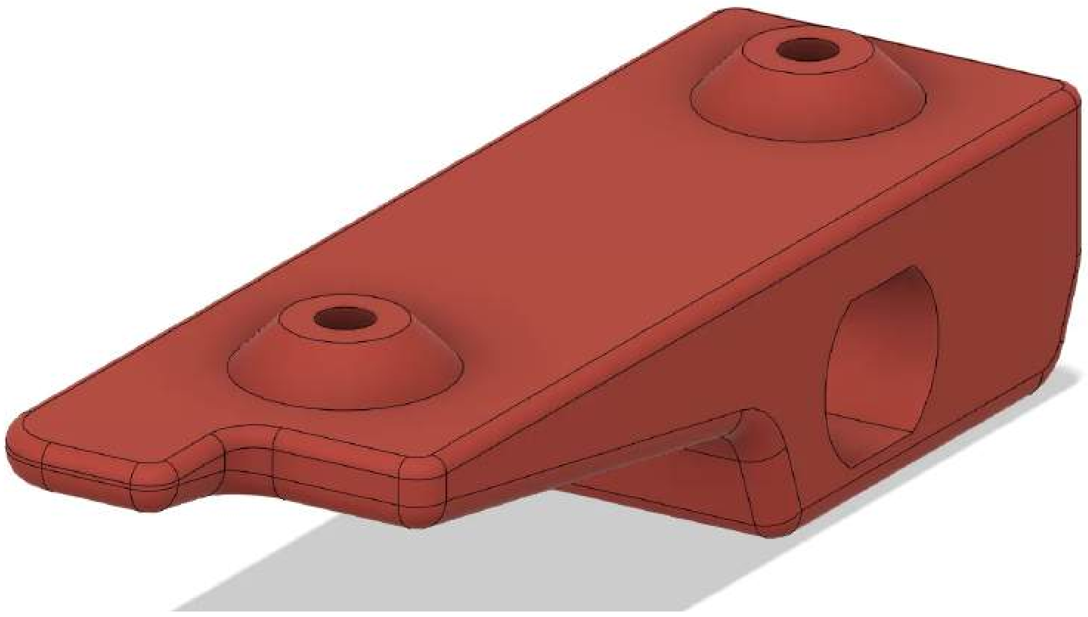
To remedy this a sketch was made on the side of the mount what would intersect horizontally and then the two sides edit feature was used with the extrude. This way I was able to keep the gap made within the width of the motor and keep space either side. A fillet was added to the new edges of 1mm.



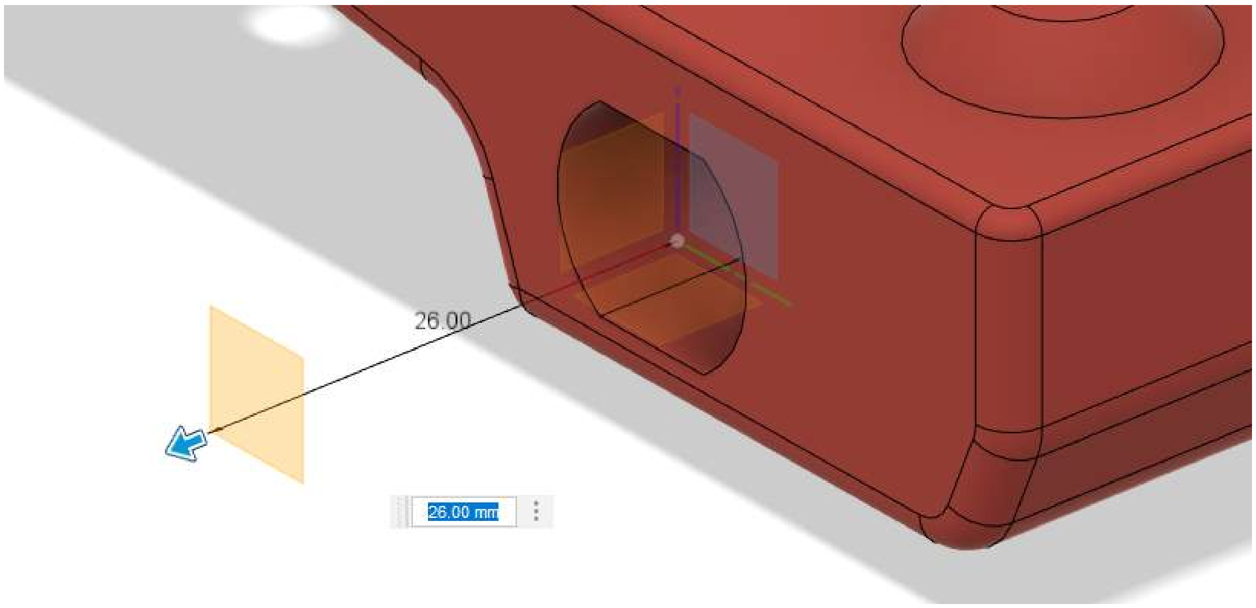
From here I wanted to add some character so I used the inspect tool and found out the dimensions that would have to be cut out in order to have the motor mount match the shap of the mouse poor on the PCB. 12.5mm by 9mm. This was then extruded through.

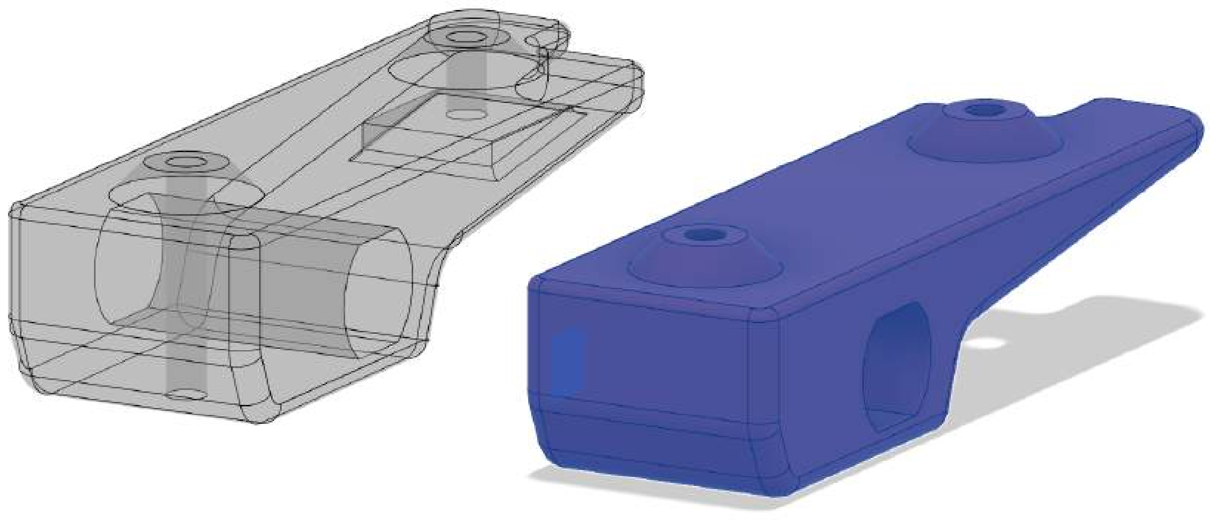


A 1mm fillet was added to the new angles resulting in the final shape.

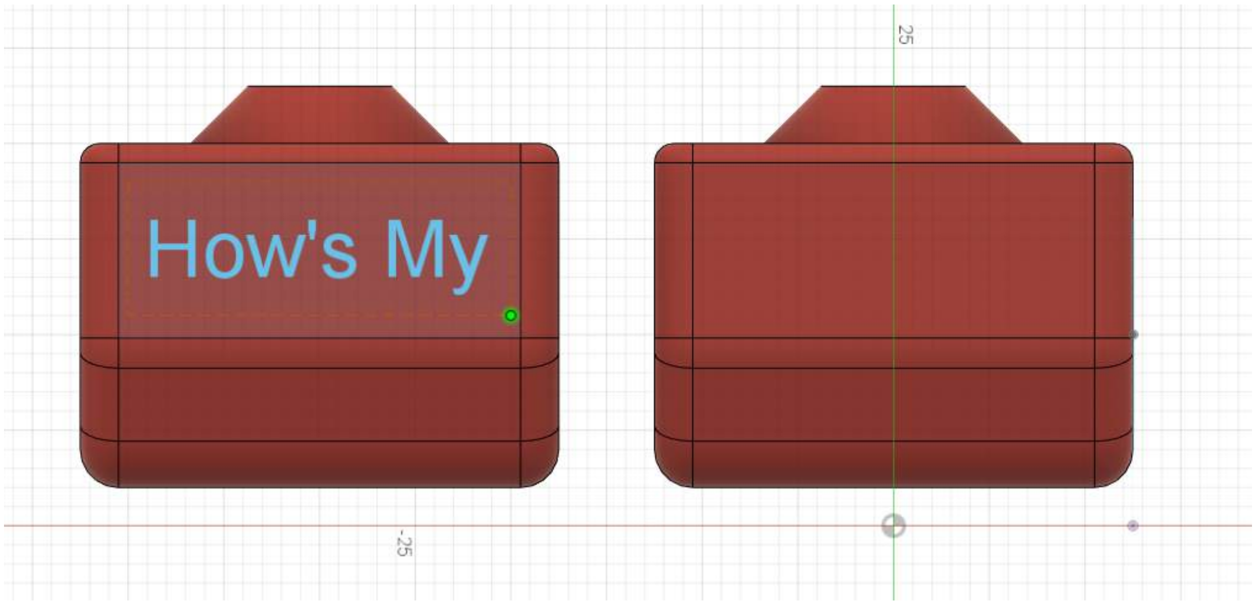


This motor mount is ready to fit one side of the PCB and so I used the mirror tool to make its counterpart. In order to do this I first made a construction plane and moved it 26mm parralel to the mount side and used it as a plane to mirror from.



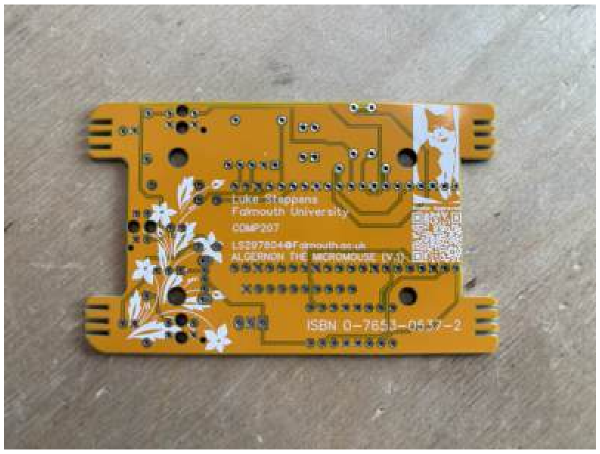


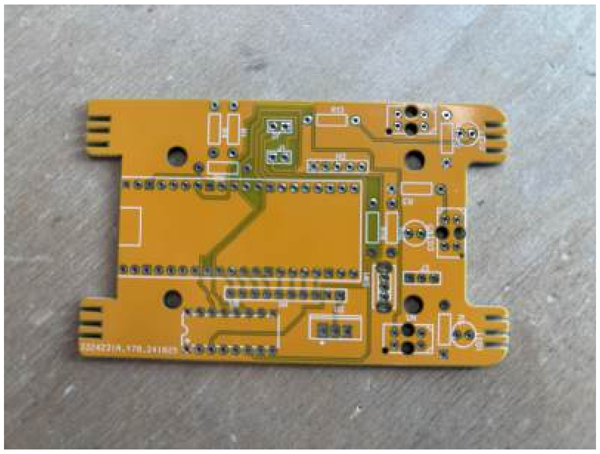
The last touch I put on it was to add some text to the flat surfaces on the back of each mount. This was done by putting sketches on the areas and choosing the text option. The text was moved to the centre and then extruded into the mount by 1mm



PCB Assembly & Testing - Arduino IDE

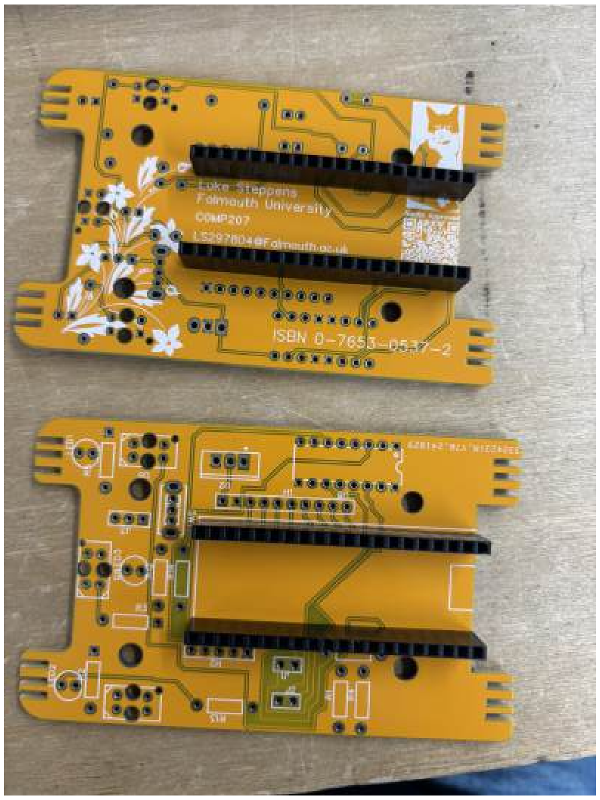
Upon recieving the PCB we were asked to give it a visual once over to check for obvious defects, then using a multimeter I checked the tracks both with themselves (one end to another) to see if it beeped and other tracks to check they dont beep. If they dont beep they arent connected and if they do then they are.



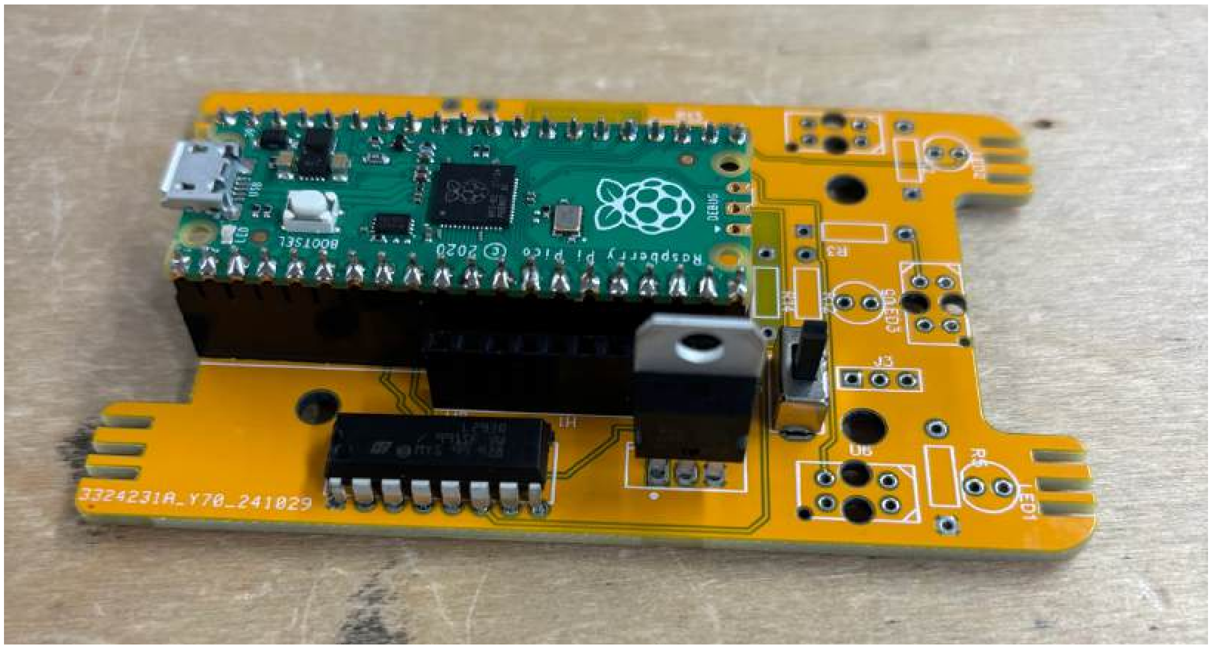


The next thing was to start attaching parts, this is where i made my first mistake and accidentally attached the Pi Pico to the wrong side effectively reversing the placement of all future components. So i started again as we recieved four PCB's each.

The bottom PCB was the correct orientation.



From here I continued the solder components. The placement I had chosen meant that I had to add parts that were a little too close to the Pi Pico first otherwise there wouldnt be enough space later.

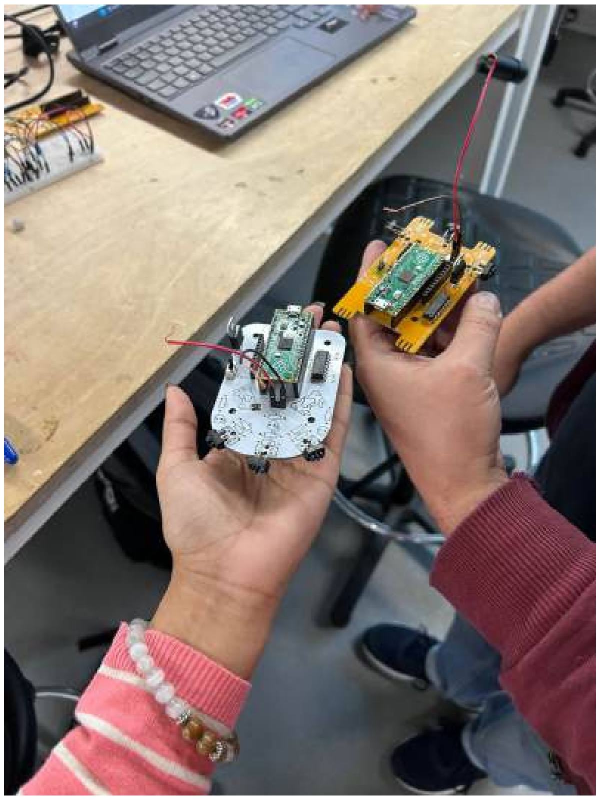


with basic parts added we connected to a power supply and tested the built in LED

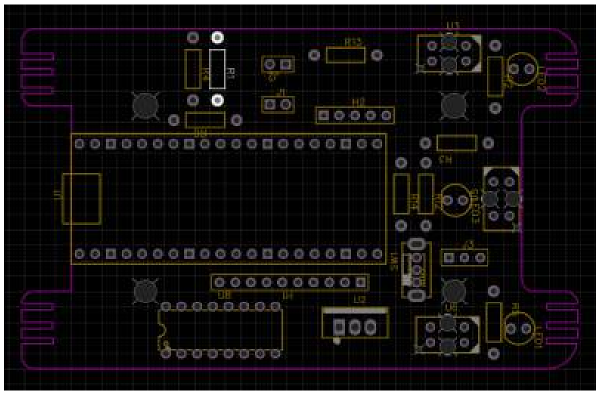
```
void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
  digitalWrite(LED_BUILTIN, HIGH)
  delay(200);
  digitalWrite(LED_BUILTIN, LOW);
  delay(200);
}
```

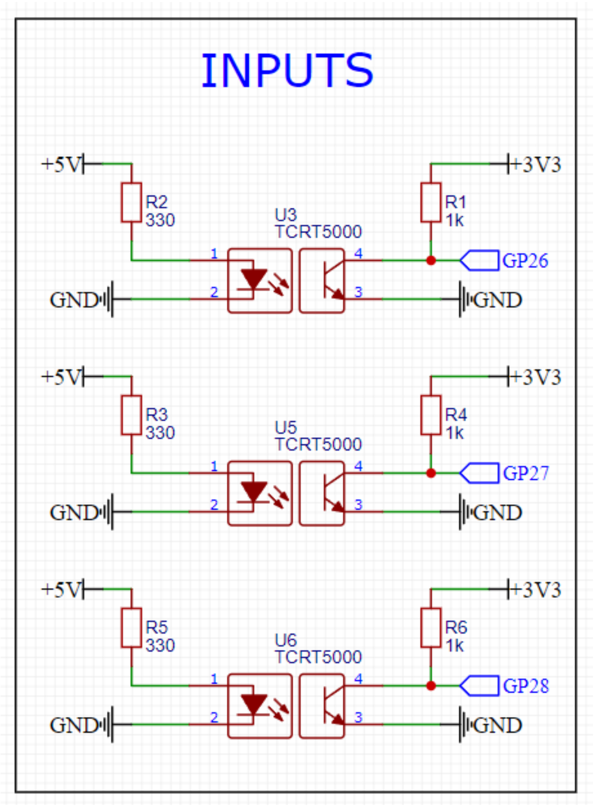
https://prod-files-secure.s3.us-west-2.amazonaws.com/6d327d71-a800-4bb7-a305-2462de459e8c/a9a1be55-ab34-4b52-babc-e18f56ccf936/IMG_3200.mov

The next thing was to add the TCRT5000's and their fuses, the TCRT5000's were fiddly and one of the LEDs came out of one so it was replaced.



For the fuses I had to make sure to check the scematics as both 1k and 330 ohms resistors were needed and had to be placed correctly.





Using the sheet on the right I was able to twin each resistor with its placement on the PCB.

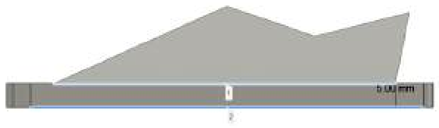
After this I soldered all of the components left over onto the PCB including some headers for the optional compass.

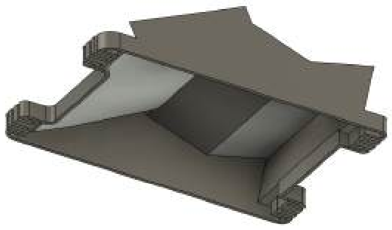
Designing Chassis - Fusion 360

By far the most time was put into this section, Multiple iterations were made. The first chassis didnt have enough clearance for the voltage regulator so I lifted it. This solved one problem but highlighted another, the chassis sides were too thin and prone the snapping, so I thickened the walls only to find another problem which was that the mouse didnt fit upon the PCB base securely. So after many redesigns and additions such as LED holes and space for wires to come through the back the final design was ready.

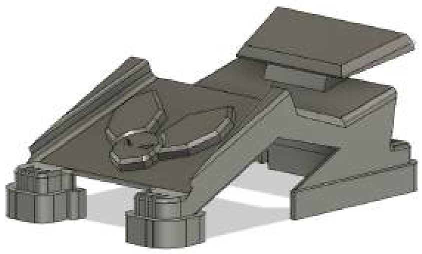
Basic Steps

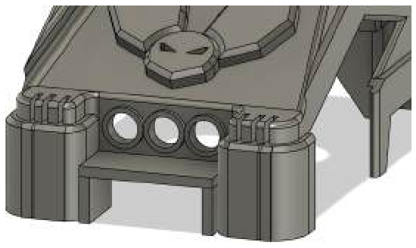
Initially I took the base PCB shape imported it into a new sketch design and extruded to 5mm, and then created a sketch on the side and used the line tool draw the general chassis shape and extruded across the base. To save time I hollowed it using the shell command and chose a thickness of minimum of 2mm.



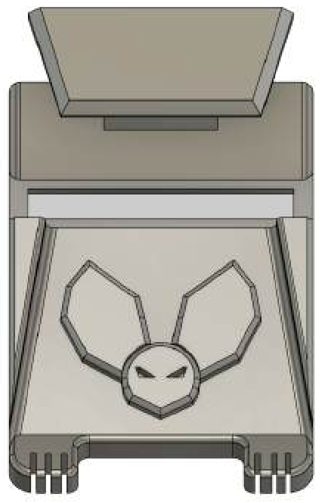


Because the TCRT5000's have to be postitioned center, left and right we need to make sure there is a place for them so we extrude to cut a gap for them. Again this was done by creating a side sketch and extruding after drawing with the line tool.

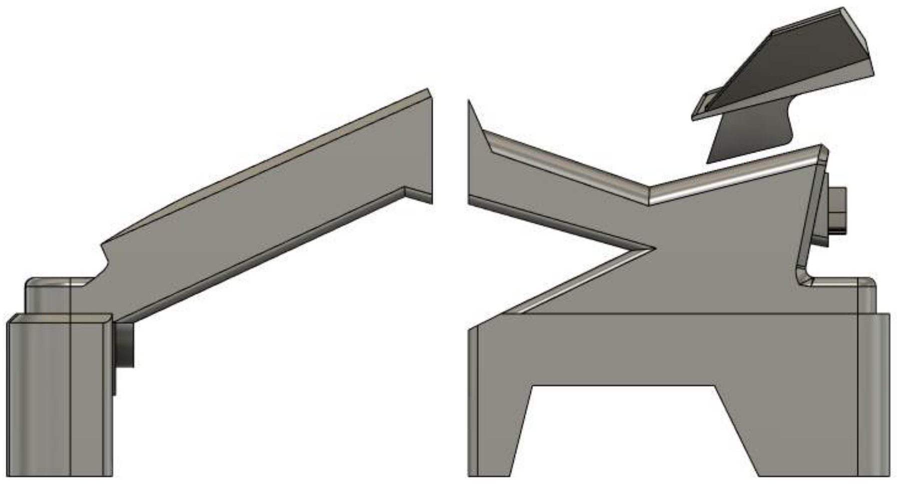




For fun I created a mouse design on the bonnets face, I did this by drawing with circle and line tools on the face and then using the mirror function so I could ensure symmetry.



When fixing the lngth of the mouse I used the split body function and then extruded from one split body to the other extending the chassis by 1cm



The many iterations of the first chassis

Assembly and Testing 2 - Arduino IDE

With the code below all three LED slots were tested and worked as hoped. I was made aware that the Blue LED was a matte type and so gave off a different more diffuse light compared to the Red and Green. The green seemed dimmer so the code was changed to lower the Blue's intensity.

https://prod-files-secure.s3.us-west-2.amazonaws.com/6d327d71-a800-4bb7-a305-2462de459e8c/6ca0a650-490a-47b0-9da9-d4c40ea07aa9/IMG_3267.mov

https://prod-files-secure.s3.us-west-2.amazonaws.com/6d327d71-a800-4bb7-a305-2462de459e8c/14a91e12-e77e-44f5-8e82-6840050bb713/IMG_3266.mov

I gave the LED's their own headers because I wanted to be able to move them around and not have to worry about my already overcomplicated chassis design

```
int ledR = 20;
int ledG = 19;
int ledB = 18;

void setup() {
  pinMode(ledR, OUTPUT);
  pinMode(ledG, OUTPUT);
  pinMode(ledB, OUTPUT);
}

void loop() {
  digitalWrite(ledR, HIGH);
  digitalWrite(ledG, HIGH);
  analogWrite(ledB, 30);    //lower the Blue LED to match intens
  delay(200);
  digitalWrite(ledR, LOW);
  digitalWrite(ledG, LOW);
  digitalWrite(ledB, LOW);
  delay(200);
}
```

TCRT5000 Test Code

```
int TCRT_Left = 26;
int TCRT_Center = 27;
int TCRT_Right = 28;
int TCRT_Output;


void setup() {

  Serial.begin(9600);
  pinMode(TCRT_Left, INPUT);
  pinMode(TCRT_Center, INPUT);
  pinMode(TCRT_Right, INPUT);
}

// gets reading from TCRT on the right and outputs to serial mor
void loop() {
  TCRT_Output = analogRead(TCRT_Right);
  Serial.print("Right Reading ");
  Serial.println(TCRT_Output);
  delay(500);
}
```

This basic code was then updated to trigger the LED's when distance was further than 8cm with each sensor. Each sensor is slightly different which is shown by the slight difference in the readings used to change the LED's off and on.

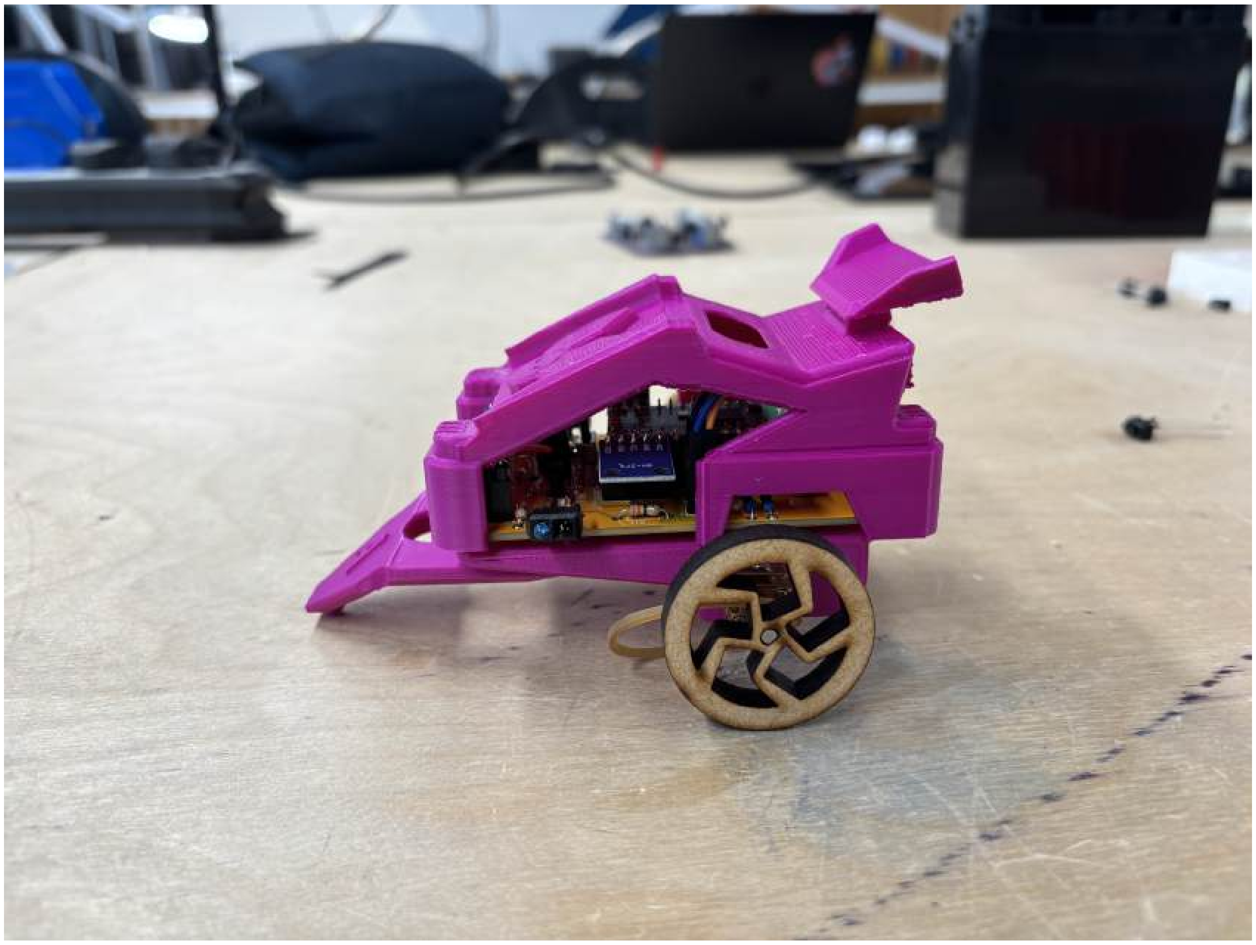
```
void checkDistance(){
  if (analogRead(TCRT_Right) < 980){
    digitalWrite(ledW, LOW);
  }
  if (analogRead(TCRT_Right) > 980){
    digitalWrite(ledW, HIGH);
  }
  if (analogRead(TCRT_Left) < 990){
    digitalWrite(ledB, LOW);
  }
  if (analogRead(TCRT_Left) > 990){
    digitalWrite(ledB, HIGH);
  }
  if (analogRead(TCRT_Center) < 1005){
    digitalWrite(ledR, LOW);
  }
  if (analogRead(TCRT_Center) > 1005){
    digitalWrite(ledR, HIGH);
  }
}
```

https://prod-files-secure.s3.us-west-2.amazonaws.com/6d327d71-a800-4bb7-a305-2462de459e8c/99e4a925-6f37-48dd-aa91-491fda8f6dea/IMG_3395.mov

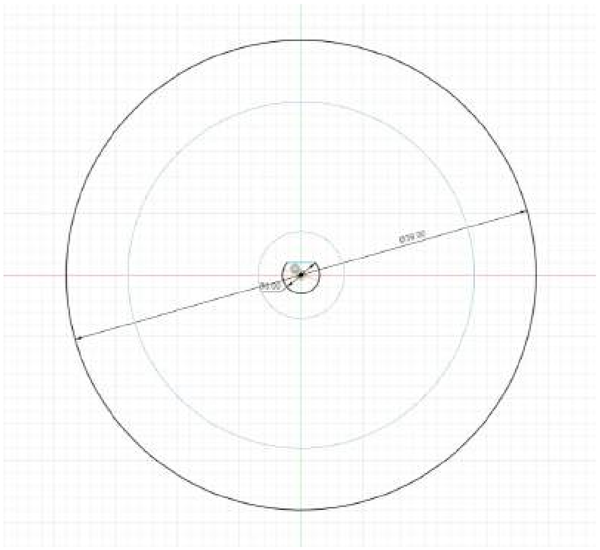
---

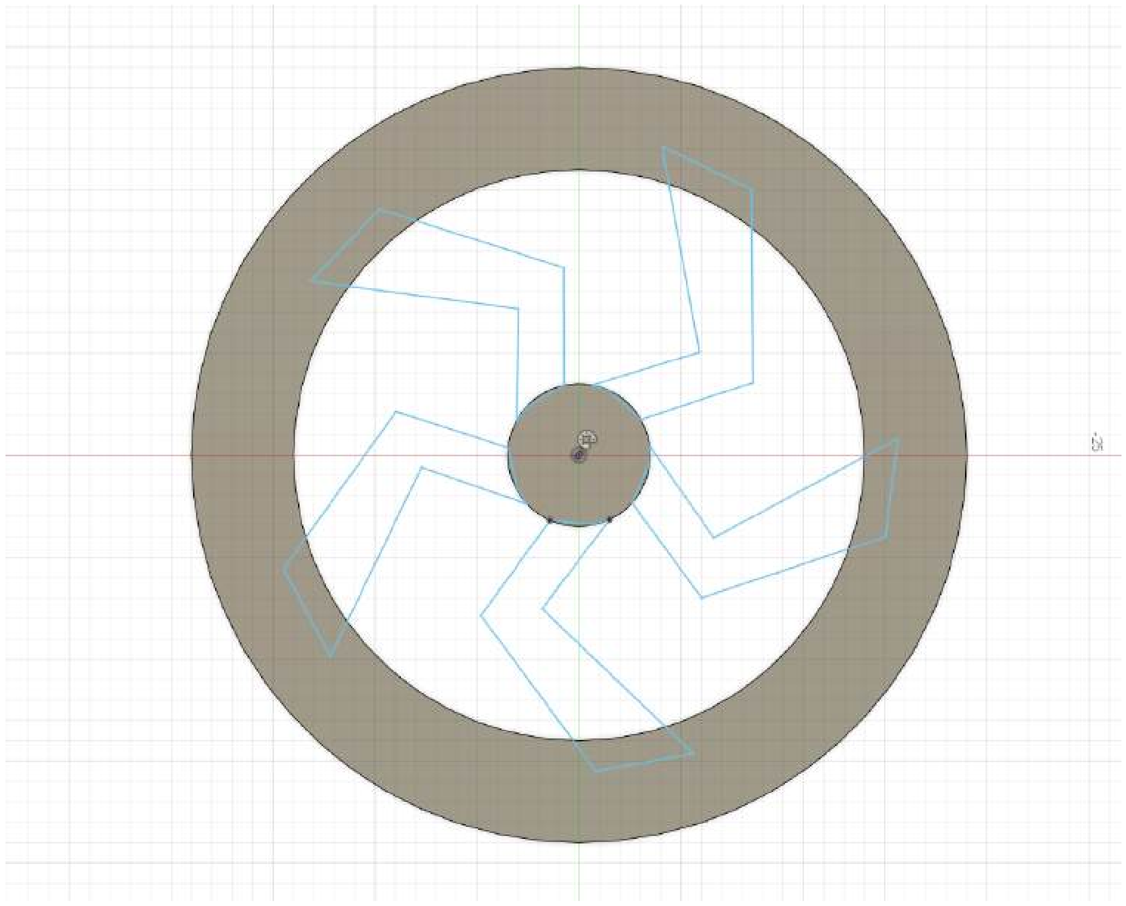
Wheel Design - Fusion 360 + Laser Cutter

The wheels went through several iterations as I kept finding problems with each. First off I designed wheels where measurements were exact, but because PLA isnt always perfect they didnt fit on shaft properly. After trying this two or three times I decided to move to laser cutting MDF as it is incredibly fast and less wasteful.



To make this wheel I first drew basic circles, one for the inner section to connect to the shaft and then one for the outer rim. The outer rim had a diameter of 38mm and the internal had 5mm. The shaft hole would be added later but the design was made with a base 3mm diameter circle, i added a construction light 1mm up from the centre and cut purpendicular to it.



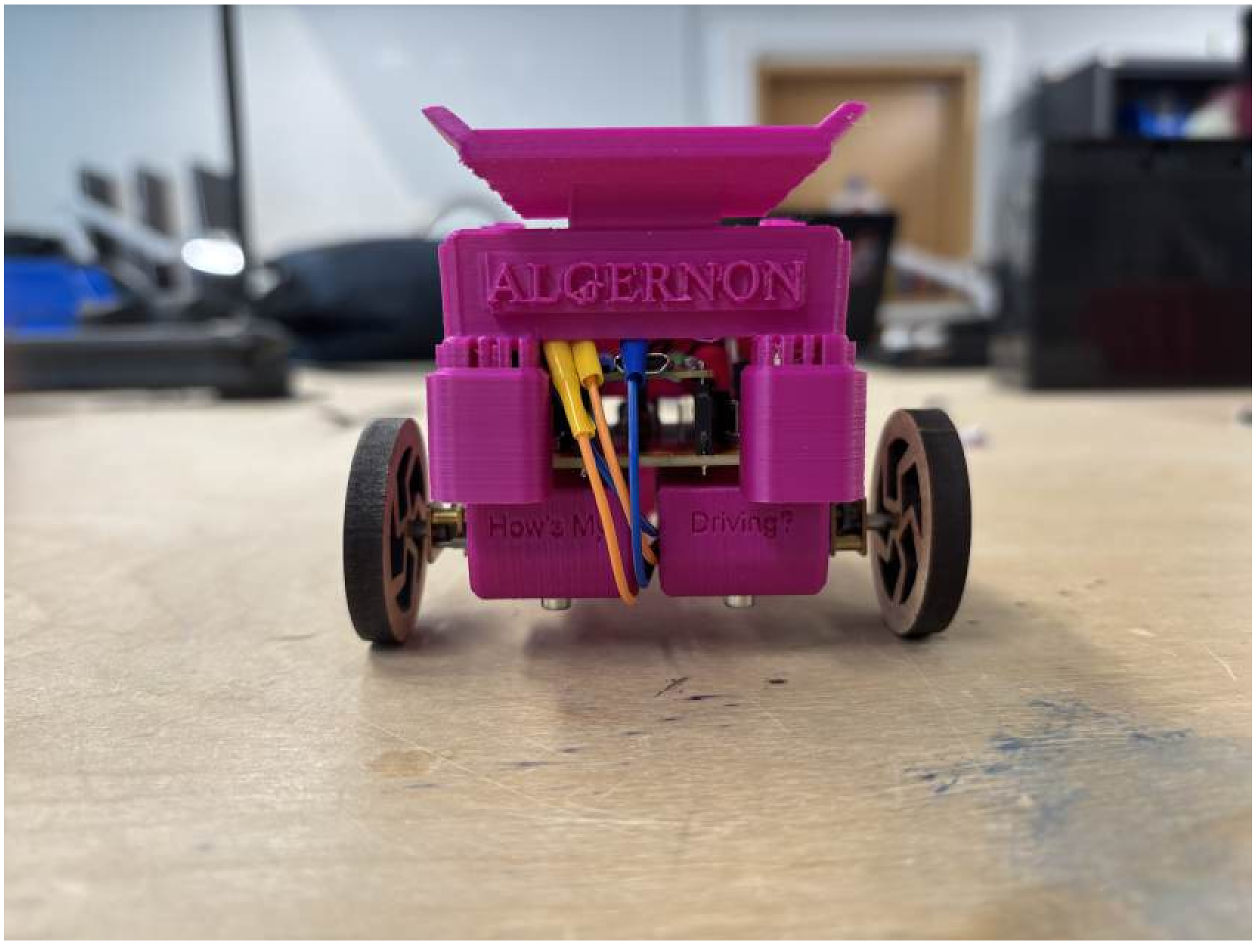


I then made a single bolt like shape as a wheel spoke and then using the circular pattern feature duplicated it 4 more times around the centre and then extruded through.

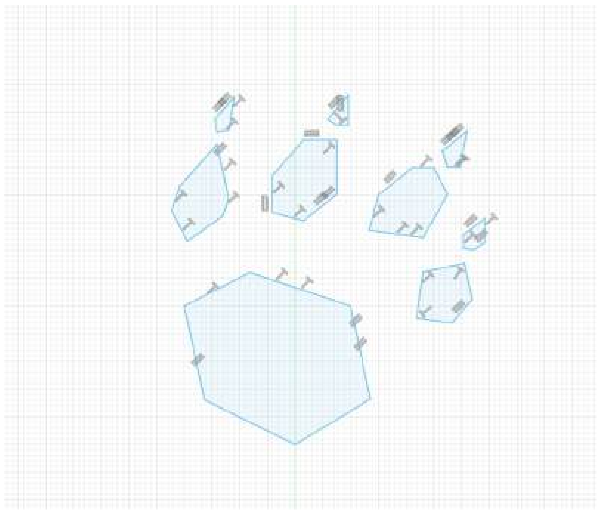
This was then exported as a DXF file and laser cut on 6mm MDF.

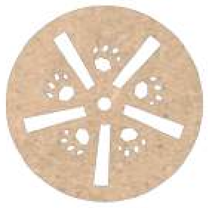
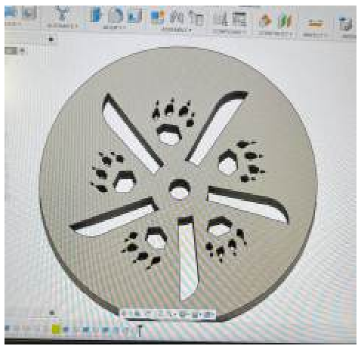
With it cut I then put all the pieces together and tested with early code

https://prod-files-secure.s3.us-west-2.amazonaws.com/6d327d71-a800-4bb7-a305-2462de459e8c/4d3e9c9b-f06c-46db-9733-b98f09f7593e/01B18EB3-F997-4ED0-A71A-163023AFC560.mp4



It was looking good but sadly I felt that it was now too wide. I was advised to not worry about fitting the wheel to the part of the shaft with the cut out section and just make the wheels fit tight. With this information I went to work on what i hoped to be the final design (it was)



To make this design I used the same outer diameter but for fun decided to create a crude paw design, this was basic shapes made with the line tool and then using the circular pattern tool rotated it around the centre. I followed this with a rectangle with a filleted corner.

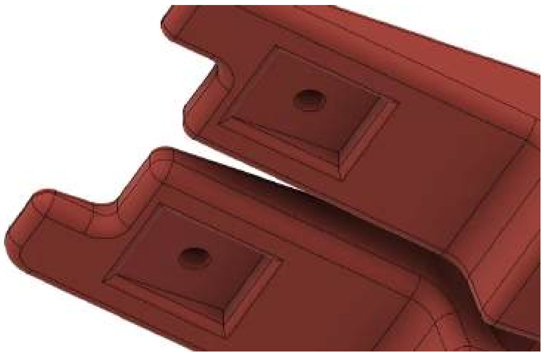
I made the hole just smaller than the shaft so rather than 3mm i made it 2.82mm. this was a snug fit, but to make it extra secure I used hot glue on the outside.

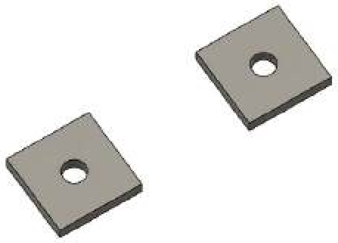
Third Point Balance - Fusion 360

The third point of balance was created at the same time as the above and then was modified again before the following section.

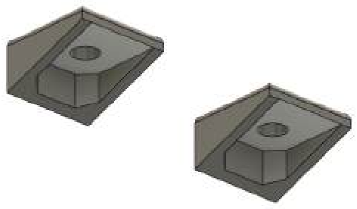
With the wheels being two points of balance for the micro mouse, so in order for it to keep its balance I designed a small protruding nub that comes from spaces I made on the motor mount.

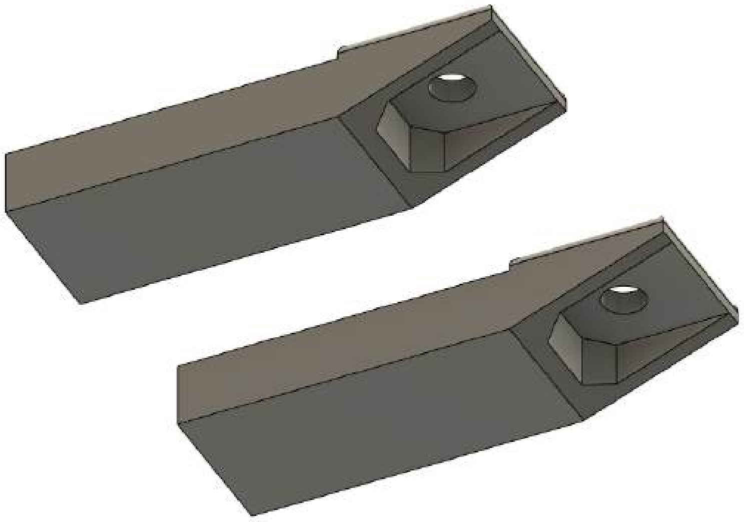
First i took the measurements from the underside of the motormounts, including their distance apart.



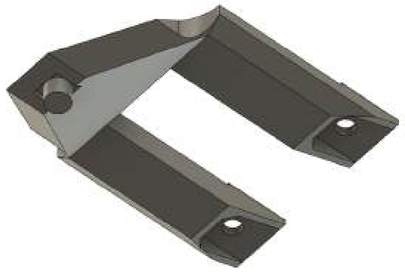


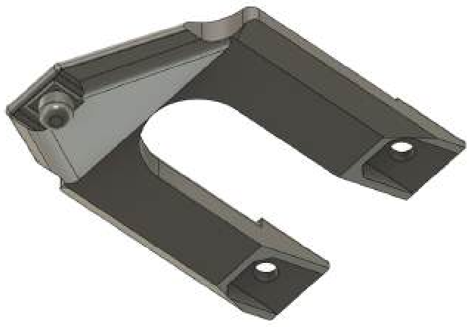
After recreating the squares I drew and arch shape and extruded the arch shape downward before creating a sketch on the side at an angle and extrude cutting away through both.



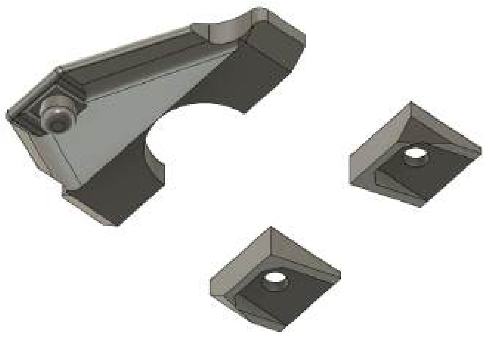


A simple extrude out to create legs was made and then on a a side sketch i created a downward slope, this would leave a small amount of clearance for the nub. The nub was made by cutting a flat square surface on the underside, I did this with a construction plane and a simple extrusion a couple mm deep. From this a 5mm circle was made and then itself extruded. Next I filleted corners and chamfered the longer edges for a nice look.

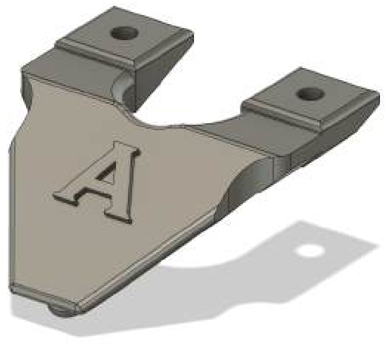


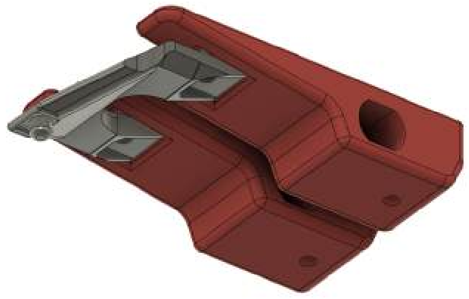


The above and to the right picture was the original design which worked well but I found it to be too long as the form factor being smaller makes it easier for the mouse to navigate without hitting anything. So I used the split body function to break the legs off from the initial square sections and then extruded the legsback until they hit the curved section before the slop down. Then i simply moved the parts together and used the combine tool to make them one piece again.



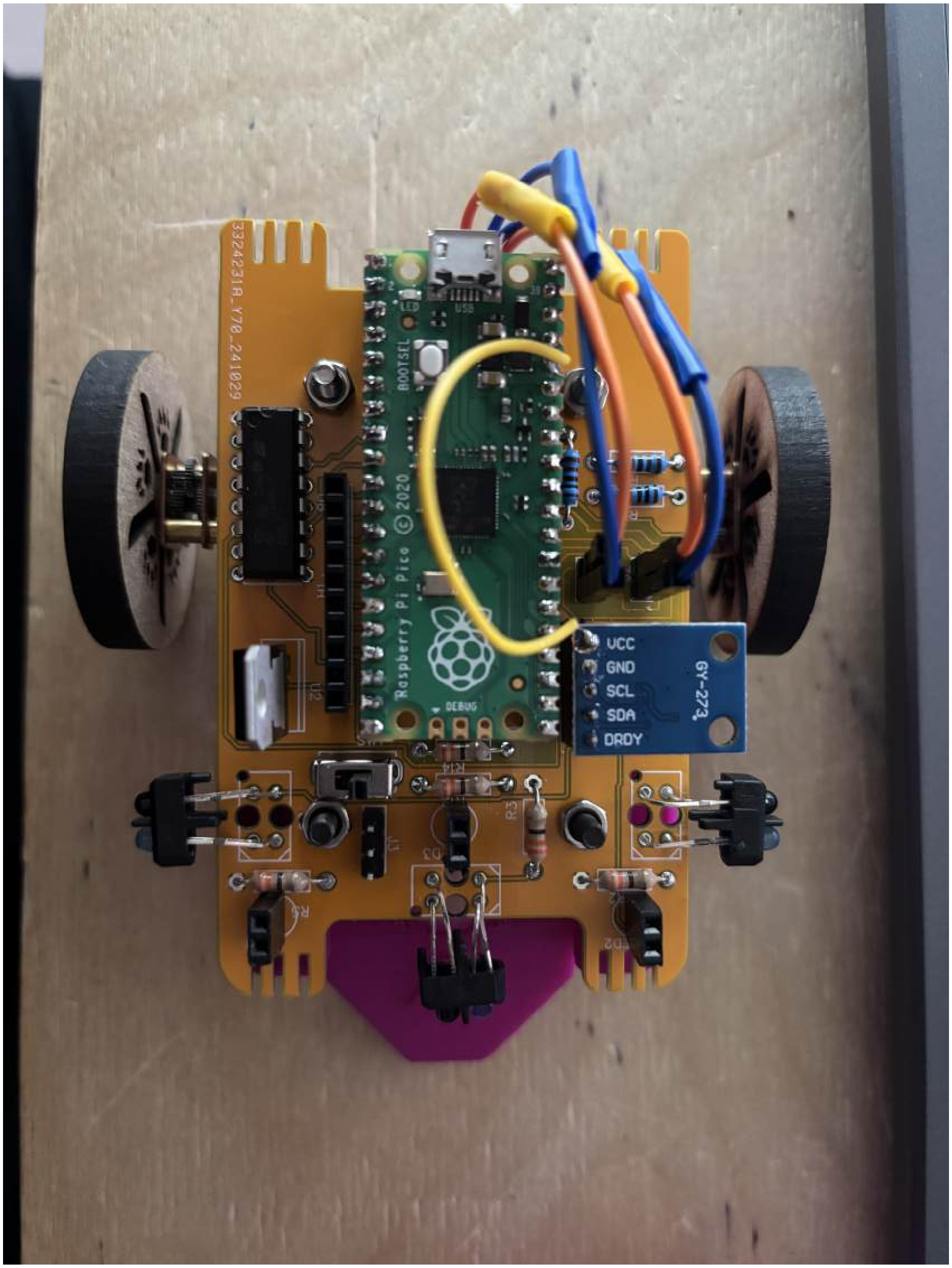
Added a nice big letter A on the surface and it was good to be attached.





Fitting the Compass - Multimeter

The compass was initially easy to fit, I soldered pins to the bottom then attached it to the headers I had already added to the Pi Pico. But there was a problem, I had used the settings mentioned on the sellers page rather than from the data sheet. VCC needed 5v and I had designed the PCB to be 3v. With sage advice from Jamie rather than remaking the whole PCB and other drastic measures i was imagining he said to solder from the top of the Pi Pico (5v pin) directly to the compass.

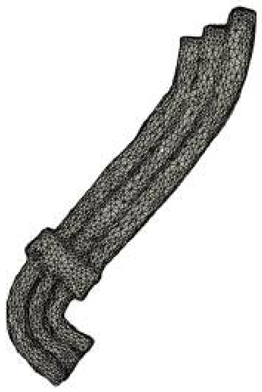

VCC
GND
SCL
SDA
DRDY
GY-273
Raspberry Pi Pico
BOOTSEL

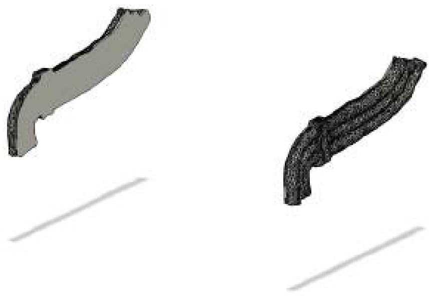
Exhaust Pipes - 3D Scanning / Fusion

For the 3D scanning we used a phone app called Polycam. The shape for the exhaust was made out of clay and then scanned with the Polycam application. For best results I took as many photos as it allows to get greater quality. I downloaded the GLTF file and converted it online (just in a brief browser search) to an STL file which I then opened up in fusion

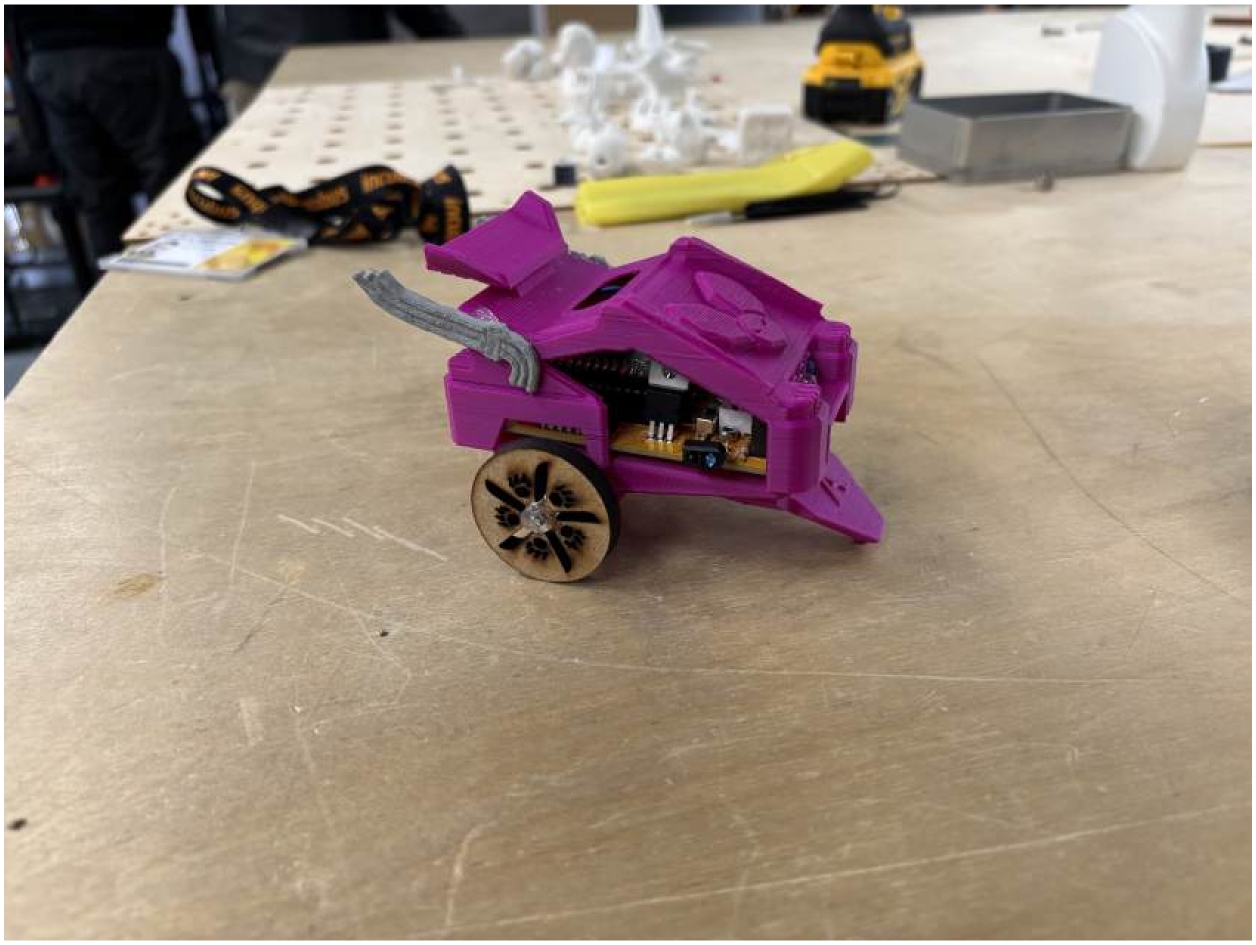
Within Fusion the table had to be removed, so I created a construction plane below the 3D image and drew a rectangle larger than said image. Next I extruded upwards. This filled in the hollowness of the underside of the image. From here I drew around the exhaust shape with the spline (from top down view) and then began the painstaking extruding away of the external segments one by one until I was left with just the exhaust. I attempted to extrude away everything outside the splines but it would crash the fusion, I think fusion didnt like the amount of polygons I was trying to intersect at one time.

https://prod-files-secure.s3.us-west-2.amazonaws.com/6d327d71-a800-4bb7-a305-2462de459e8c/2b36841f-8e58-436a-af10-c23be70ce715/ScreenRecording_01-03-2025_12-00-26_1.mov





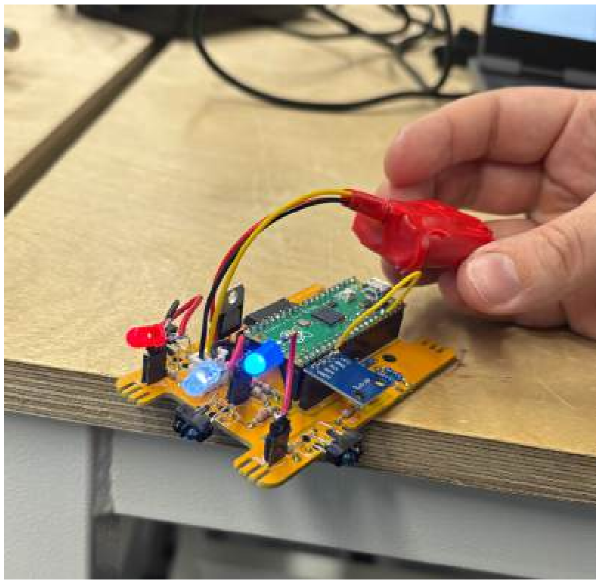
Using the mirror tool I easily created the exhausts counterpart



Battery + Basic Navigation - Arduino IDE

In one of the final workshop sessions we were tasked with breaking down materials in order to retrieve the batteries we then had to attach them to our micro mice to ensure they worked, using our schematics I was able to see which orientation with which to put the wires.

At this point I had also changed the middle LED to a plain white one.



Next I put it all together and had the battery hanging off the top, this way I could test some basic code for navigating the maze

Using the below code and some basic IF statements I was able to get the mouse to move around, although in a very clunky manner.

https://prod-files-secure.s3.us-west-2.amazonaws.com/6d327d71-a800-4bb7-a305-2462de459e8c/eef1ebdd-9530-4601-8190-308ef067bd22/IMG_3484.mov

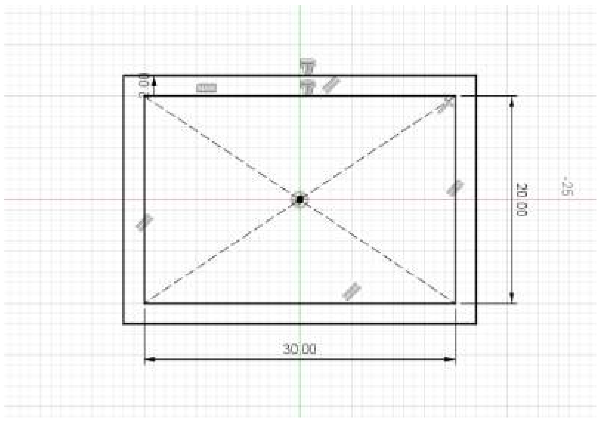
```
void loop() {

  checkTcrt();  // Continuously print sensor readings for debugg


  Front = false;
  Left = false;
  Right = false;

  checkDistance();

  if (analogRead(TCRT_Front) < 950) {
    Serial.println("Front sensor triggered");
    Front = true;

  }
  if (analogRead(TCRT_Left) < 980) {
    Serial.println("Left sensor triggered");
    Left = true;

  }
  if (analogRead(TCRT_Right) < 930) {
    Serial.println("Right sensor triggered");
    Right = true;

  }

  if (!Front){        // if nothing detected ahead move forward
    motorsForward();
  }

  else if (!Left){    // if nothing detected to the left turn le
    turnLeft();
  }
  else{                // if neither of the above turn right
    turnRight();
  }

}
```

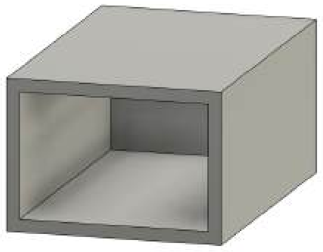
In an attempt to be more specific with my code I actually made things worse so i went back to the code above.

https://prod-files-secure.s3.us-west-2.amazonaws.com/6d327d71-a800-4bb7-a305-2462de459e8c/59ee77ac-caad-49d1-ad4a-d02824596e4a/IMG_3482.mov

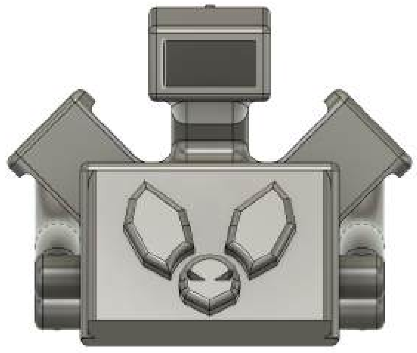
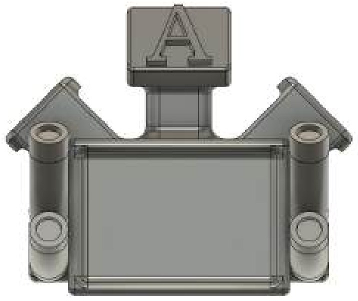
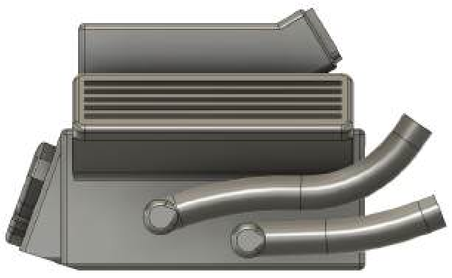
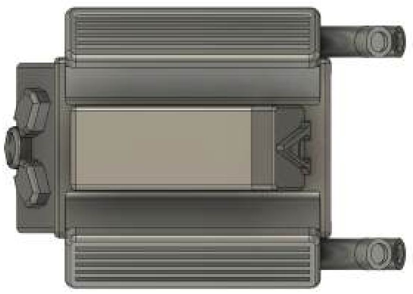
Battery Holder - Fusion 360

With the addition of the battery a final problem was added which was where to place it on the chassis, making it as complicated as I had there was no easy surface to attach it to so i went about designing a battery holder, which i would then incorperate into my chassis design.





I designed a simple box to contain the battery and then I wanted to make it look fun so I took inspiration from the big V8 engines Like the ones below and started working on my own.

Utilising all my prior tools I first created the side angled sections by drawing them on the front sketch then mirroring them and extruding along the body. then to created the ridges I created one thin rectangle, extruded it into the body by 2mm and then use the rectangular pattern function to copy itself 4 more times, using the same procress on the other side to make them match.

https://prod-files-secure.s3.us-west-2.amazonaws.com/6d327d71-a800-4bb7-a305-2462de459e8c/00385046-6c8a-4c09-b43a-719d7574a22b/20250105-0840-20.3509204.mp4

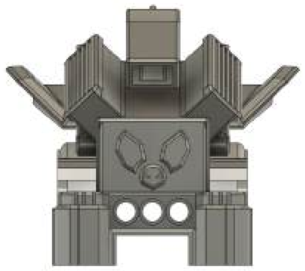
The air intake was a simple couple boxes stacked upon the battery, one small and one large. Then I used the shell command to hollow it out. For the face I went back to the chassis design and copied the sketch with the mouse drawing upon it and then pasted it upon the face of the battery (which i angled a little)

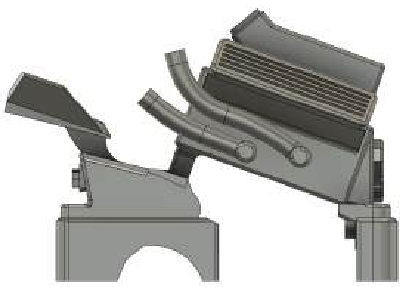
The last touch was the tubes, this was new to me but simple ones I had spent some time with it. First I made protruding circles and then used the tube function purpendicular to the protruding circles and made them follow a curved line upwards, this was done twice and then mirrored across to the other side.

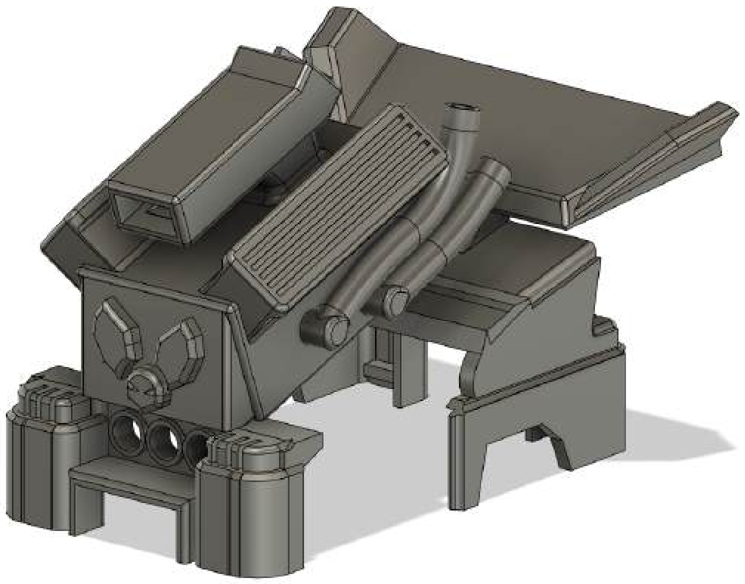
Final Chassis - Fusion 360

In order to incorparate the battery section I had to remove the bonnet and central part of the chassis completely and then add a stanchion closer to the spoiler. This way the wires can easily be connected to the PCB. The only really difficult part here was combining the front of the battery to the paw section containing the LED holders. Because I had overly complicated geometry they initially refused to combine so I had to cut sections away using flat planes in order to connect them.

https://prod-files-secure.s3.us-west-2.amazonaws.com/6d327d71-a800-4bb7-a305-2462de459e8c/4980e32e-18d5-40e6-b4f9-5e739014d779/20250105-1003-28.0350578.mp4







The final touch I increased the size of the spoiler significantly. To do this I use the scale tool and moved it higher before remaking the stanchion connecting it to the rear.

Algernon Versus The Maze - Arduino IDE

This simple code helps Algernon traverse the maze, it is a mixture of basic functions reacting to thresholds set to readings from the TCRT's in CM. It currently works at times but is erratic and prone to spinning in place.

```
// Algernon Current Code



// LED pins
int ledR = 20;
int ledW = 18;
int ledB = 19;

// motor pins
int motor_leftA = 0;   // pin0
int motor_leftB = 1;   // pin1
int motor_rightB = 2;  // pin2
int motor_rightA = 3;  // pin3

// TCRT pins
int TCRT_Left = 26;
int TCRT_Front = 27;
int TCRT_Right = 28;
int TCRT_Output;

bool Front = false;
bool Left = false;
bool Right = false;

int drive_Speed = 35;
int turn_Speed = 30;
int reverse_Speed = 25;

bool firstLoop = true;

void setup() {

  Serial.begin(9600);

  // TCRTS
  pinMode(TCRT_Left, OUTPUT);
  pinMode(TCRT_Front, OUTPUT);
  pinMode(TCRT_Right, OUTPUT);

  // motors
  pinMode(motor_leftA, OUTPUT);
  pinMode(motor_leftB, OUTPUT);
  pinMode(motor_rightA, OUTPUT);
  pinMode(motor_rightB, OUTPUT);

  // LEDS
  pinMode(ledR, OUTPUT);
  pinMode(ledW, OUTPUT);
  pinMode(ledB, OUTPUT);
}

void loop() {

  if (firstLoop) {  // have 4 second delay before starting the l
    delay(4000);
    firstLoop = false;
  }


  checkTcrt();  // Continuously print sensor readings for debugg

  Front = false;  // sensors set to false by deafult so they sta
  Left = false;
  Right = false;

  checkDistance();

  if (analogRead(TCRT_Front) < 981) {
    Serial.println("Front sensor triggered");
    Front = true;

  }
  if (analogRead(TCRT_Left) < 995) {
    Serial.println("Left sensor triggered");
    Left = true;

  }
  if (analogRead(TCRT_Right) < 993) {
    Serial.println("Right sensor triggered");
    Right = true;

  }

// If all three sensors detect something move backwards
  if (Front && Left && Right){
    motorsBackward();
  }

  else if (Front && Left){
    turnRight();
  }

  else if (Front && Right){
    turnLeft();
  }

  else if (Left){
    turnRight();
  }

  else if (Right){
    turnLeft();
  }

  else if (Front){
    turnLeft();
  }

  else {
    motorsForward();
  }
}
```

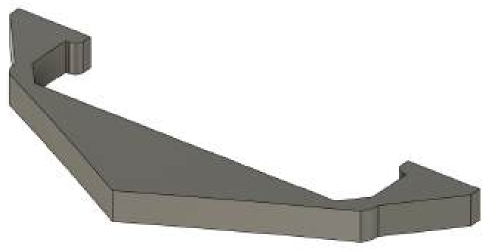
```
// functions
void checkTcrt() {
  // 1cm = 543 - 548, 2cm = 851 - 858, 3cm = 940 - 948,
  // 4cm = 974 - 986, 5cm = 993 - 1002, 6cm = 1002 - 1023,
  // 7cm = 1007 - 1015, 8cm = 1008 - 1018
  Serial.print(" Right Reading ");
  Serial.print(analogRead(TCRT_Right));

  // 1cm = 567 - 569, 2cm = 869 - 876, 3cm = 950 - 959,
  // 4cm = 980 - 990, 5cm = 995 - 1005, 6cm = 1003 - 1013,
  // 7cm = 1010 - 1018, 8cm = 1013 - 1021
  Serial.print("| Left Reading ");
  Serial.print(analogRead(TCRT_Left));

  // 1cm = 691 - 700, 2cm = 892 - 898, 3cm 953 - 964,
  // 4cm = 981 - 990, 5cm = 994 - 1003, 6cm 1001 - 1011,
  // 7cm = 1007 - 1015, 8cm = 1007 - 1019
  Serial.print("| Center Reading ");
  Serial.println(analogRead(TCRT_Front));
  delay(500);
}

void checkDistance() {
  if (analogRead(TCRT_Right) < 980) {
    digitalWrite(ledW, LOW);
  }
  if (analogRead(TCRT_Right) > 980) {
    digitalWrite(ledW, HIGH);
  }
  if (analogRead(TCRT_Left) < 990) {
    digitalWrite(ledB, LOW);
  }
  if (analogRead(TCRT_Left) > 990) {
    digitalWrite(ledB, HIGH);
  }
  if (analogRead(TCRT_Front) < 1005) {
    digitalWrite(ledR, LOW);
  }
  if (analogRead(TCRT_Front) > 1005) {
    digitalWrite(ledR, HIGH);
  }
}

void turnLeft() {
  analogWrite(motor_leftA, -turn_Speed);
  analogWrite(motor_leftB, turn_Speed);
  analogWrite(motor_rightA, turn_Speed);
  analogWrite(motor_rightB, -turn_Speed);
}

void turnRight() {
  analogWrite(motor_leftA, turn_Speed);
  analogWrite(motor_leftB, -turn_Speed);
  analogWrite(motor_rightA, -turn_Speed);
  analogWrite(motor_rightB, turn_Speed);
}

void motorsForward() {

  analogWrite(motor_leftA, drive_Speed);
  analogWrite(motor_leftB, -drive_Speed);
  analogWrite(motor_rightA, drive_Speed);
  analogWrite(motor_rightB, -drive_Speed);

}

void motorsBackward() {
  analogWrite(motor_leftA, -reverse_Speed);  // Reverse left mo
  analogWrite(motor_leftB, reverse_Speed);   // Reverse left mo
  analogWrite(motor_rightA, -reverse_Speed); // Reverse right m
  analogWrite(motor_rightB, reverse_Speed);  // Reverse right m
```

```
}

void allLeds() {
  digitalWrite(ledR, HIGH);
  digitalWrite(ledW, HIGH);
  digitalWrite(ledB, HIGH);
}
```

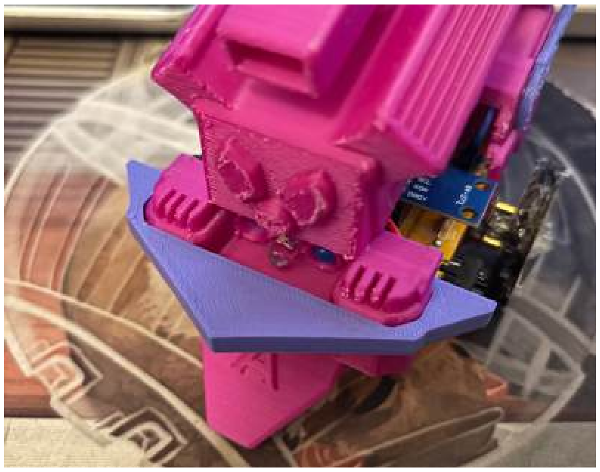
Final additions - Fusion 360

When attempting the maze I found two issues to be solved firstly the MDF had a hard time gripping the ground and seconding the wheels were getting caught when going around a hairpin turn.

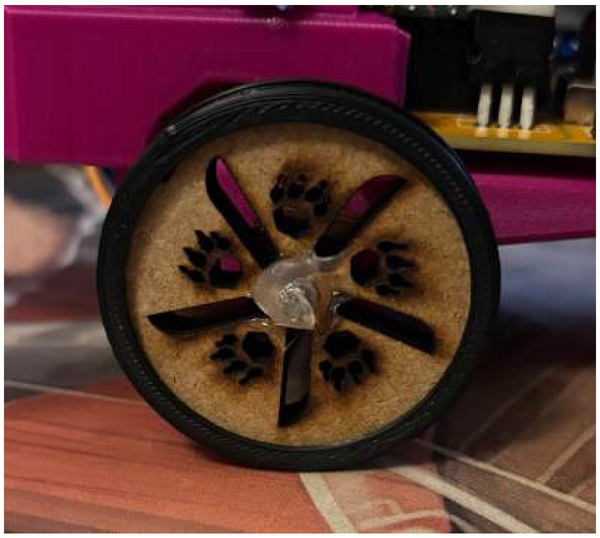
For the latter I designed a bumber that simply clips onto the front of the mouse. To make this I took measurement around the front and simply drew an arrow shape with hooks on the side and then extruded the pattern.



Placing this bumper at the front helps the thin walls not get caught between the chassis and the wheels.



For the Wheels I made a circle the same diameter as the wheels 38mm and then offset this by 1mm and extrudes the width of the MDF which was 6mm. I then offset the larger circle again on either side and extruded inwards by 2mm. Then I decided to make it so the tire couldnt slide off the wheel by making a third offset, but this one would be a smaller circle effectively making them like caps on the wheels. Then the outer edge was filleted.



https://prod-files-secure.s3.us-west-2.amazonaws.com/6d327d71-a800-4bb7-a305-2462de459e8c/3091c0d4-9f88-4465-87fe-123d1ec6af40/20250106-1745-28.3511821.mp4