Development Log - 207:

26/9/12:

Today we begun learning about the Pico pro, we connected it to a breadboard and the a L7308 which regulates the voltage down to 5v.

We also used a TCRT5000 which emits and detects IR light, this will be used to detect obstacles.

04/10/2024:

We used learnt about the physical design of PCB’s and the design process behind it. Going from just the components to a schematic then from there to a digital version of the PCB layout and finally the manufacturing.

For this module we are using Easy EDA, on of the reasons for his is because of the usefulness of being able to buy the components required through affiliated companies.

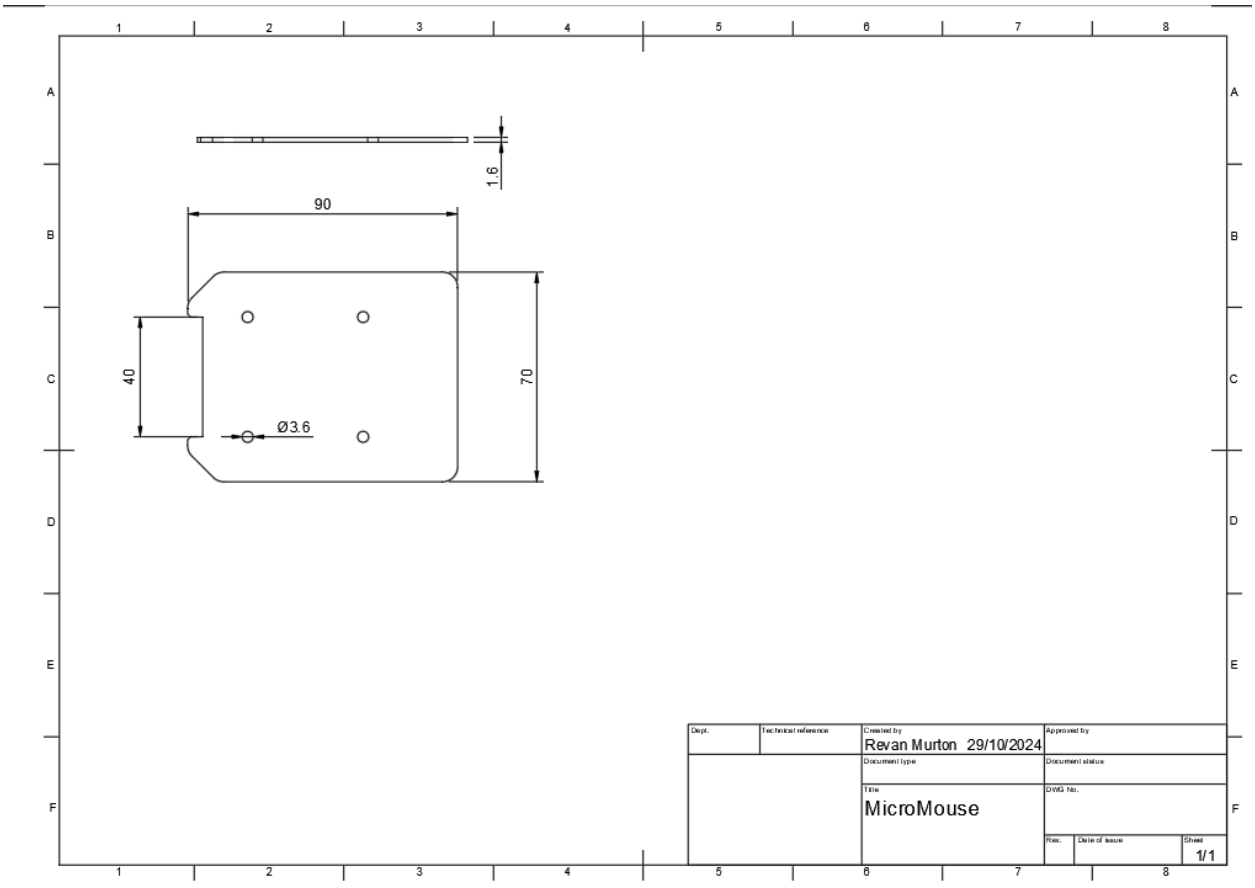
We revised the ICE standards for the symbols of electrical components. as well as the math behind finding the resistance on a system with resistors in parallel and in series.

We did a simple schematic and layout for a basic PCD that had a 1×2 Header. LED and 1k Resistor.

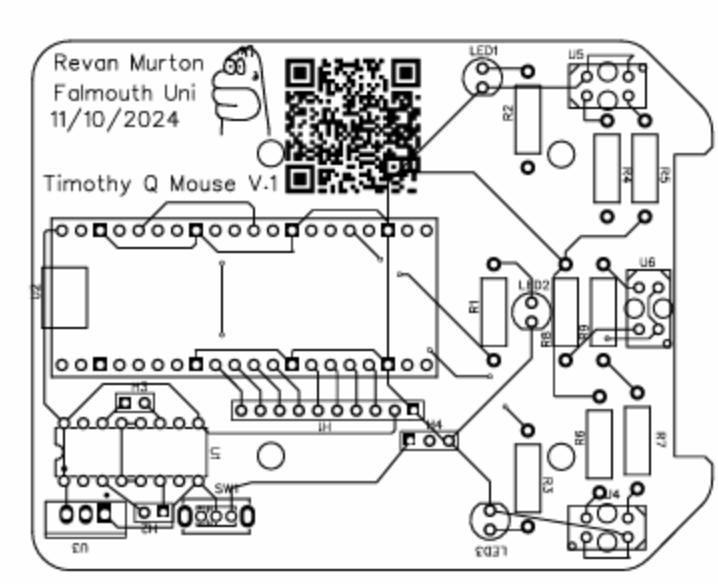
I finished the initial MicroMouse schematic and will finish the layout before next Friday.

11/10/2024:

In today's session we went through creating the layout for our PCBs, we used fusion 360 in order to create the shape of our boards, in order to connect the different components this involves creating tracks on both sides of the PCB, using the via tool in order to circumnavigate wires that are in the way and ensuring there are no faulty connections



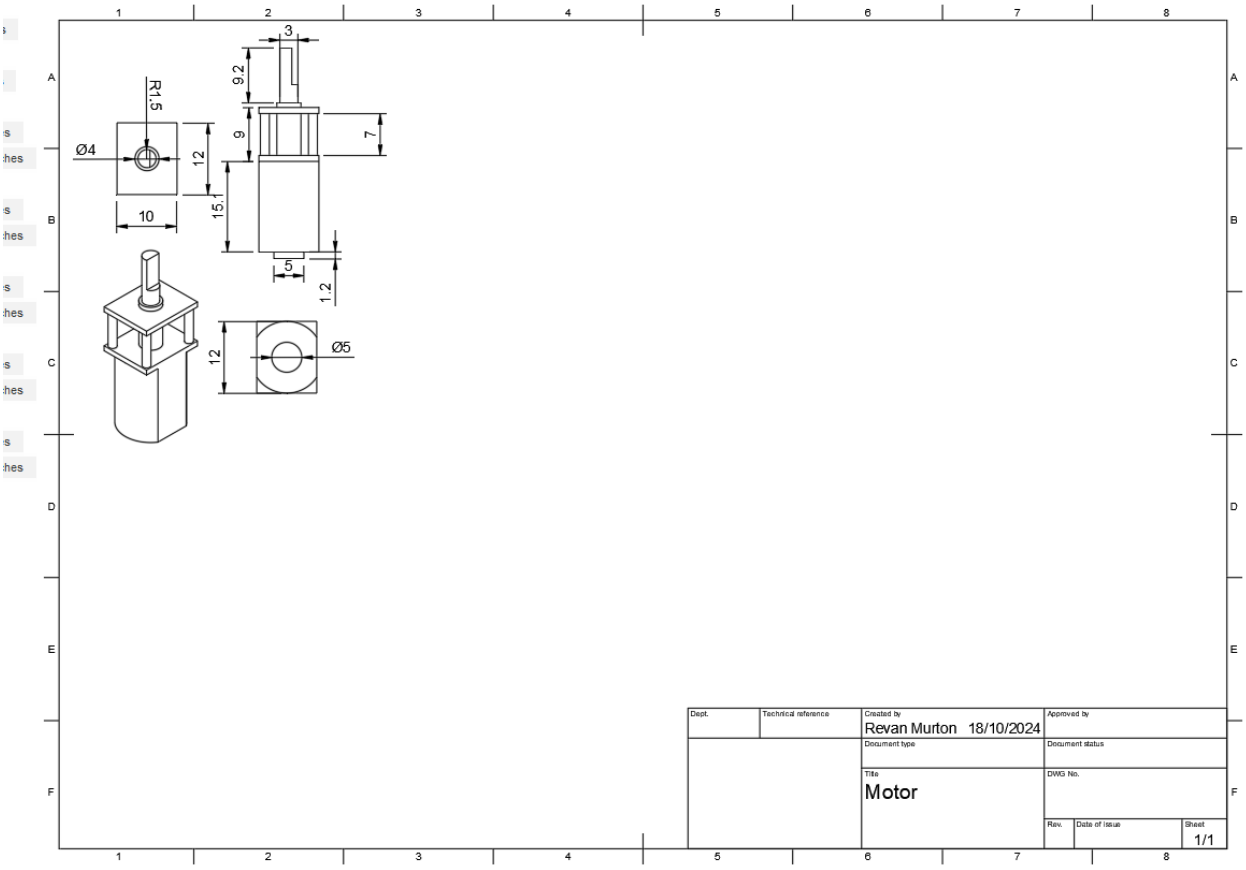
I got all the connections completed and added the required info to the PCB as well such as a qr code, name and date



18//10/2024:

We went over the design again, I had to redesign my PCB due to my poor placement of the mounting holes, however the fix was quick and hopefully my PCB works once it arrives.

I designed the motor in Fusion 360 this session, once I had the design modelled I added colours to more accurately match the physical copy I had as reference. I then made a drawing of the part so I could have a quick reference guide to it's dimensions.



We also went over the formula for power P = IV and how to solve for multiple different scenarios.

V = IR

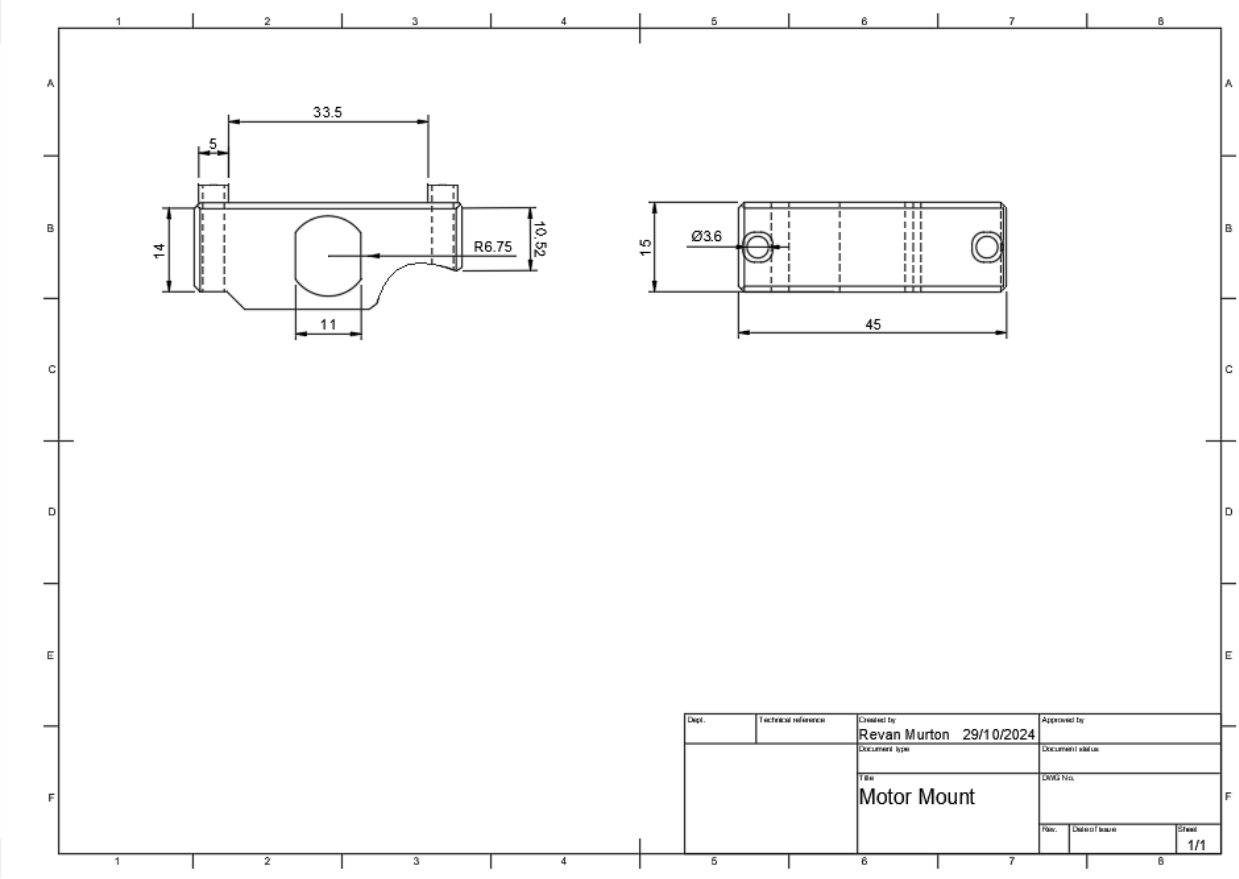
P = V/R * V

P = V/I * V

V = P/V * V

25/10/2024:

I finished the first design of my motor mouse today, this allowed me to make an assembly of the PCB board, motor mounts and the motors. We were inducted on the 3D printers and I got a test print of my motor mount printed off. Unfortunately I sized the hole for the motor incorrectly so the motor was too big to fit inside. I will redesign the mount and reprint after the reading week so I can get more of the micro mouse components ready for when the PCB's arrive.

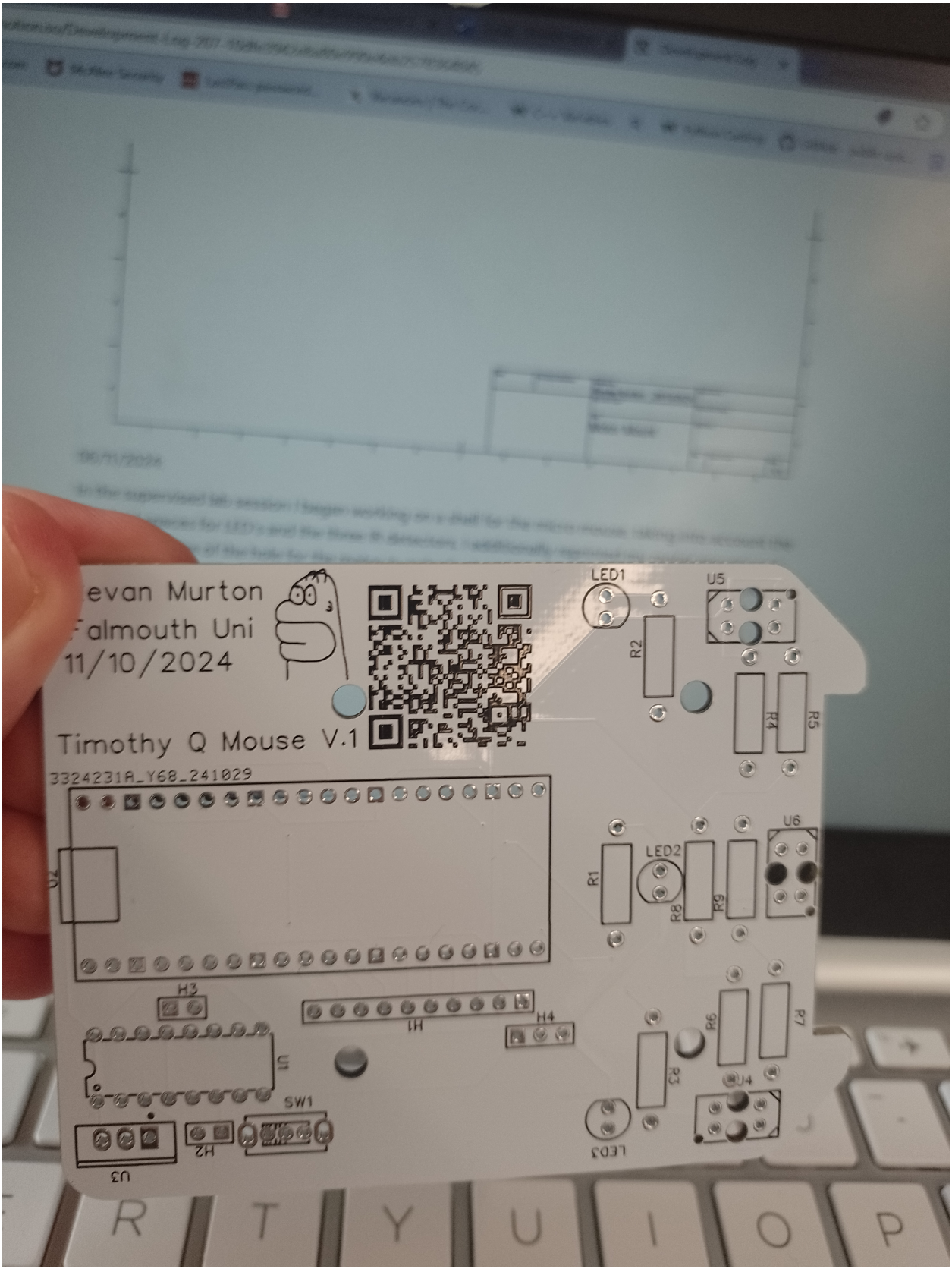


06/11/2024:

In the supervised lab session I began working on a shell for the micro mouse, taking into account the required spaces for LED's and the three IR detectors. I additionally reprinted my motor mounts and adjusted the size of the hole for the motor in order for it to fit inside. I will add hot glue or another adhesive to ensure the motor cannot fall out as the micro mouse is moving.

08/11/2024:

We received the PCB’s and began a visual inspection, I could notice no copper that had been exposed, the wiring looks good and the silkscreen is the exact same.


evan Murton
Falmouth Uni
11/10/2024
Timothy Q Mouse V.1
3324231A_Y68_241029
LED1
U5
R2
R4
R5
U6
LED2
R1
R8
R9
H3
H1
H4
R6
R7
R3
U4
U2
U1
SW1
H2
U3
LED3


LIMITED RUN
ARMAGEDDON
alt gr

Next I did a probe test on all of the connections to ensure there the correct components were connected and to ensure there were no errors. This went well and there were no wrong or missing connections on the PCB it's self.

The next step was to solder header female headers onto the place the Pico would go, this was to ensure that the Pico could be removed if there was an issue with it and so it could be reused in another project. This went smoothly and I double checked that the Pico was working by running the Blink code example code.

```
// the setup function runs once when you press reset or power th
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is
  delay(1000);                      // wait for a second
  digitalWrite(LED_BUILTIN, LOW);   // turn the LED off by makin
  delay(1000);                      // wait for a second
}
```

The next step was to solder the battery connector headers (for this I used male headers), the voltage regulator and a switch to turn on and off the Pico. The solders for these components went well, however I ran into two errors that I caused by not double checking my soldering earlier and changing my technique to account for this. The voltage regulator was not fully soldered, instead of one continuous mass of solder I had two separate clumps, this reduced the strength of the bond holding the voltage regulator to the board and prevented the electricity from flowing through. The next mistake was to not double check the soldering on the Pico itself that it didn't suffer from the same problem.

These mistakes made it take a lot longer to debug the board itself. I did this by checking if the connections were correct by running the blink code on the Pico which is being powered by an external power supply.

In the end I finally got it working with Ben's help, it turned out the biggest mistake was the GND pin on the Pico was not properly soldered, in the future I will ensure that I double check each of my solders to ensure they will work.

Finally I added the IR detectors to the front with solder, I bent them to hopefully make them more successful at detecting walls to the sides and front.

14/11/24:

Today in the Supervised Lab I soldered on all of the components for the MicroMouse. I made sure to check each solder for good smooth connections as last time that cost me a lot of time. Because of the limited number of solder stations and the limited time. I got all the resistors, motor driver and LED's in place before I soldered. This sped up the process massively. I tested the TCRT sensors and they work, giving a lower signal the closer an object is.

15/11/24:

I designed a dice holder in the workshop today, I've yet to laser cut it off but will do over the coming weeks, I don't plan on using laser cutting for anything else other than the wheels this was good practise and helped me to understand how to get multiple laser cut pieces to join together without using any adhesives.

22/11/24:

I spent the supervised session today printing both the shell and motor mounts again. Both of them will need improvement as the motor mount is too wide for the motors as I tried to account for the consequences of 3D printing and overcompensated. On the other hand the shell covers all three of the LED's so I'll need to redo this as well.

I managed to do a little more code too.

23/11/24:

Today we started in the IT room but quickly moved to the lab, I began work on the code for the MicroMouse and spending far too long trying to get the pins for the LED's and TCRT's correct again. I realise now I should of just checked the PCB and use the pins from there and in the future I will do this to save time and my sanity.

Here is the code I have at the end of today.

```
int blueLED;
int redLED = 20;
int yellowLED = 18;
int leftTCRT = 26;
int middleTCRT = 27;
int rightTCRT = 28;
int motorRA = 0;
int motorRB = 1;
int motorLA = 2;
int motorLB = 3;

enum mouse_States {FORWARD, RIGHT, LEFT, MIDDLE};

int motorDriving(int speed, int motorA, int motorB){ //function
 //Make sure speed is within the bounds
  if (speed < -255){
    speed = -255;
  }
  if (speed > 255){
    speed = 255;
  }

  //sets motors to spin forward, backwards or not at all depend
  if(speed > 0){
    analogWrite(motorA, -speed);
    analogWrite(motorB, speed);
  }
   if(speed < 0){
```

```
    analogWrite(motorA, speed);
    analogWrite(motorB, -speed);
  }
   if(speed == 0){
    analogWrite(motorA, 0);
    analogWrite(motorB, 0);
  }

}
void testLED(){
  digitalWrite(redLED,HIGH);
  digitalWrite(yellowLED, HIGH);
  digitalWrite(blueLED, HIGH);
  delay(1000);
  digitalWrite(redLED,LOW);
  delay(1000);
  digitalWrite(yellowLED, LOW);
   delay(1000);
  digitalWrite(blueLED, LOW);
  delay(1000);
}
void testTCRT(){
  if(analogRead(leftTCRT) < 900){
   digitalWrite(redLED, HIGH);
  }
   if(analogRead(leftTCRT) > 900){
   digitalWrite(redLED, LOW);
  }

   if(analogRead(middleTCRT) < 900){
   digitalWrite(yellowLED, HIGH);
  }
   if(analogRead(middleTCRT) > 900){
   digitalWrite(yellowLED, LOW);
  }
```

```
   if(analogRead(rightTCRT) < 900){
   digitalWrite(blueLED, HIGH);
  }
   if(analogRead(rightTCRT) > 900){
   digitalWrite(blueLED, LOW);
  }

}
void motorTest(){
  digitalWrite(motorRA, HIGH);
  digitalWrite(motorRB, LOW);
  digitalWrite(motorLA, HIGH);
  digitalWrite(motorLB, LOW);
}
void turnLeft(){
  motorDriving(-120, motorLA, motorLB);
  motorDriving(255, motorRA, motorRB);
  delay(500);
  motorDriving(120, motorLA, motorLB);
  motorDriving(120, motorRA, motorRB);
}
void turnRight(){
  motorDriving(255, motorLA, motorLB);
  motorDriving(-120, motorRA, motorRB);
  delay(500);
  motorDriving(120, motorLA, motorLB);
  motorDriving(120, motorRA, motorRB);
}
void wallInfront(){
  if(analogRead(leftTCRT) < analogRead(rightTCRT)){
    motorDriving(0, motorLA, motorLB);
    motorDriving(255, motorRA, motorRB);
    }
    else{
    motorDriving(255, motorLA, motorLB);
    motorDriving(0, motorRA, motorRB);
```

```
    }
}
void forwardMovement(){
  int left = map(analogRead(leftTCRT), 400, 1040, 255, 0);
  int right = map(analogRead(rightTCRT), 400, 1040, 255, 0);
  motorDriving(left, motorLA, motorLB);
  motorDriving(right, motorRA, motorRB);

}

void setup() {
  Serial.begin(9600);
  pinMode(leftTCRT, INPUT);
  pinMode(middleTCRT, INPUT);
  pinMode(rightTCRT, INPUT);
  pinMode(blueLED, OUTPUT);
  pinMode(redLED,OUTPUT);
  pinMode(yellowLED, OUTPUT);
  pinMode(motorLA, OUTPUT);
  pinMode(motorLB, OUTPUT);
  pinMode(motorRA, OUTPUT);
  pinMode(motorRB, OUTPUT);

}

void loop() {
  /*testTCRT();
  motorTest();*/
  movement();
}
void movement(){
  forwardMovement();
  if( analogRead(leftTCRT) && analogRead(middleTCRT) < 970){
    turnRight();
  }
   if( analogRead(rightTCRT) && analogRead(middleTCRT) < 970){
```

```
    turnLeft();
  }
  if(analogRead(middleTCRT) < 650){
    wallInfront();
  }
}
```

28/11/24:

I did more coding today in the supervised session, although I've now set up states that should in theory work I couldn't get them working. I had an issue where even though each state would run when it was detecting no specific scenarios the wheels just spun at max power despite the map function being there that should increase and decrease the speed of the motors depending on the distance output by the left and right TCRT's. I also need to comment a lot more in my code.

```
int blueLED = 19;
int redLED = 20;
int yellowLED = 18;
int leftTCRT = 26;
int middleTCRT = 28;
int rightTCRT = 27;
int motorRA = 1;
int motorRB = 0;
int motorLA = 3;
int motorLB = 2;
unsigned long timer;
enum Mouse_enum {FORWARD, TURNRIGHT, TURNLEFT, CROSSROAD};
enum Sensor_enum {RIGHT, LEFT, MIDDLE};
static Mouse_enum mouse;
static Sensor_enum sensor;

void motorDriving(int speed, int motorA, int motorB){ //functio
 //Make sure speed is within the bounds
```

```
  if (speed < -255){
    speed = -255;
  }
  if (speed > 255){
    speed = 255;
  }

  //sets motors to spin forward, backwards or not at all depend
  if(speed > 0){
    analogWrite(motorA, -speed);
    analogWrite(motorB, speed);
  }
   if(speed < 0){
    analogWrite(motorA, speed);
    analogWrite(motorB, -speed);
  }
   if(speed == 0){
    analogWrite(motorA, 0);
    analogWrite(motorB, 0);
  }

}

void turnLeft(){ //Turns left and then continues for a short tim
  motorDriving(-225, motorLA, motorLB);
  motorDriving(255, motorRA, motorRB);

}
void turnRight(){ //Turns right and then continues for a short t
  motorDriving(255, motorLA, motorLB);
  motorDriving(-255, motorRA, motorRB);



}
void wallInfront(){ //Decides which way to turn depending on the
  if(analogRead(leftTCRT) < analogRead(rightTCRT)){
```

```
    motorDriving(-255, motorLA, motorLB);
    motorDriving(255, motorRA, motorRB);
    delay(500);
    }
    else{
    motorDriving(255, motorLA, motorLB);
    motorDriving(-255, motorRA, motorRB);
    delay(500);
    }
}
void forwardMovement(){ //Moves the mouse forwards should accom
  int left = map(analogRead(leftTCRT), 300, 1100, 255, 0);
  int right = map(analogRead(rightTCRT), 300, 1100, 255, 0);
  Serial.println(left);
  Serial.println(right);
  motorDriving(left, motorLA, motorLB);
  motorDriving(right, motorRA, motorRB);

}


void setup() {
  Serial.begin(9600);
  pinMode(leftTCRT, INPUT);
  pinMode(middleTCRT, INPUT);
  pinMode(rightTCRT, INPUT);
  pinMode(blueLED, OUTPUT);
  pinMode(redLED,OUTPUT);
  pinMode(yellowLED, OUTPUT);
  pinMode(motorLA, OUTPUT);
  pinMode(motorLB, OUTPUT);
  pinMode(motorRA, OUTPUT);
  pinMode(motorRB, OUTPUT);

}
```

```
void sensing(){ //Sets the state of the sensors
  if( analogRead(leftTCRT) && analogRead(middleTCRT) < 950){
    sensor = LEFT;
  }
   if( analogRead(rightTCRT) && analogRead(middleTCRT) < 950){
    sensor= RIGHT;
  }
  if(analogRead(middleTCRT) < 800){
    sensor= MIDDLE;
 }
}

void movement(){

  if( analogRead(leftTCRT) && analogRead(middleTCRT) < 970){
    turnRight();
  }
   if( analogRead(rightTCRT) && analogRead(middleTCRT) < 970){
    turnLeft();
  }
  if(analogRead(middleTCRT) < 650){
    wallInfront();
  }
  else{
     forwardMovement(); //Moves the mouse forwards should accom
  }
}

void mouseStates(uint8_t Sensor_enum){
  switch(mouse){
    case FORWARD:
      if(sensor == LEFT){
      timer = millis();
      mouse = TURNRIGHT;
      }
      if(sensor == MIDDLE){
```

```
      mouse = CROSSROAD;
      }
      if(sensor == RIGHT){
      timer = millis();
      mouse = TURNLEFT;
      }
      else{
      forwardMovement();
      }
      break;

    case CROSSROAD:
       if(sensor = MIDDLE){
        mouse = FORWARD;
        digitalWrite(blueLED,LOW)
       }
      else{
        wallInfront();
        digitalWrite(blueLED,HIGH)
       }
      break;

    case TURNLEFT:
      if( millis() - timer >= 200){
        digitalWrite(redLED,LOW);
        mouse = FORWARD;

       }
      else{
        digitalWrite(redLED,HIGH)
        turnLeft();
       }
      break;

    case TURNRIGHT:
      if( millis() - timer >= 200){
```

```
        digitalWrite(yellowLED,LOW);
        mouse = FORWARD;
       }
      else{
        digitalWrite(yellowLED, HIGH);
        turnRight();
       }
      break;

  }

}


void loop() {
  /*
  testTCRT();
  motorTest();
  testLED();
  mouseStates(sensor);
  sensing();
  */
  Serial.println(analogRead(middleTCRT));
  Serial.println(analogRead(leftTCRT));
  Serial.println(analogRead(rightTCRT));
  if( analogRead(leftTCRT) && analogRead(middleTCRT) < 900){
    sensor = LEFT;
  }
   if( analogRead(rightTCRT) && analogRead(middleTCRT) < 900){
    sensor= RIGHT;
  }
  if(analogRead(middleTCRT) < 800){
    sensor= MIDDLE;
 }
  mouseStates(sensor);
```

```
}


//Test code

void testLED(){ //Tests the LED's
  digitalWrite(redLED,HIGH);
  digitalWrite(yellowLED, HIGH);
  digitalWrite(blueLED, HIGH);
  delay(1000);
  digitalWrite(redLED,LOW);
  delay(1000);
  digitalWrite(yellowLED, LOW);
   delay(1000);
  digitalWrite(blueLED, LOW);
  delay(1000);
}
void testTCRT(){ //Tests the TCRT's by turning LED's on if they
  if(analogRead(leftTCRT) < 900){
   digitalWrite(redLED, HIGH);
  }
   if(analogRead(leftTCRT) > 900){
   digitalWrite(redLED, LOW);
  }

   if(analogRead(middleTCRT) < 900){
   digitalWrite(yellowLED, HIGH);
  }
   if(analogRead(middleTCRT) > 900){
   digitalWrite(yellowLED, LOW);
  }

   if(analogRead(rightTCRT) < 900){
```

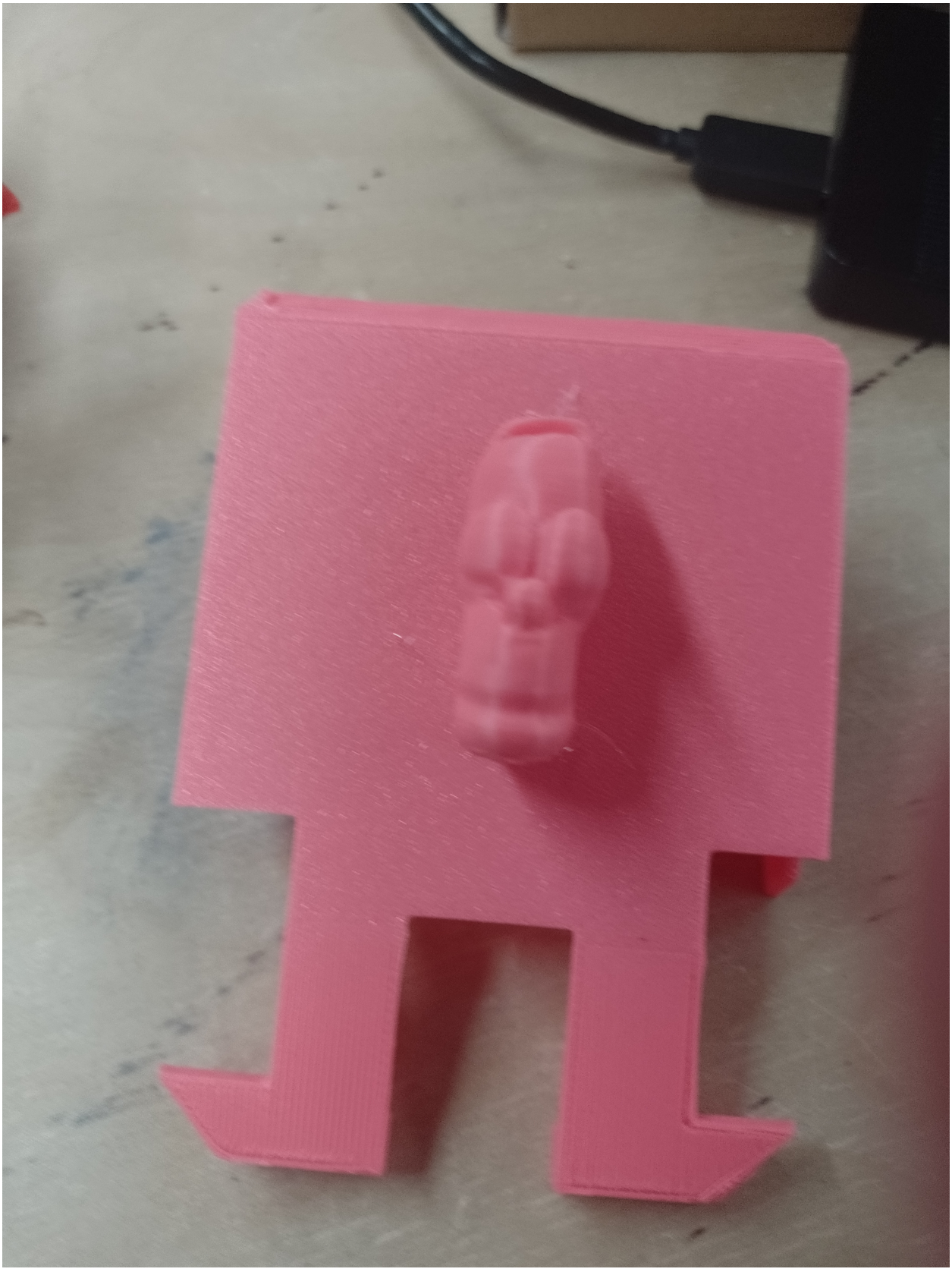
```
    digitalWrite(blueLED, HIGH);
   }
    if(analogRead(rightTCRT) > 900){
    digitalWrite(blueLED, LOW);
   }

}
void motorTest(){ //Tests the motors
   digitalWrite(motorRA, HIGH);
   digitalWrite(motorRB, LOW);
   digitalWrite(motorLA, HIGH);
   digitalWrite(motorLB, LOW);
}
```

29/11/24:

Today we used clay in order to create sculptures/designs to 3D Scan. My biggest issue was due to my phone being older I couldn't use the recommended app of PolyCam. However I got Kiri working and managed to get two poor scans completed. I will rescan the design later, scale it down and 3D print in to sit somewhere on my MicroMouse.

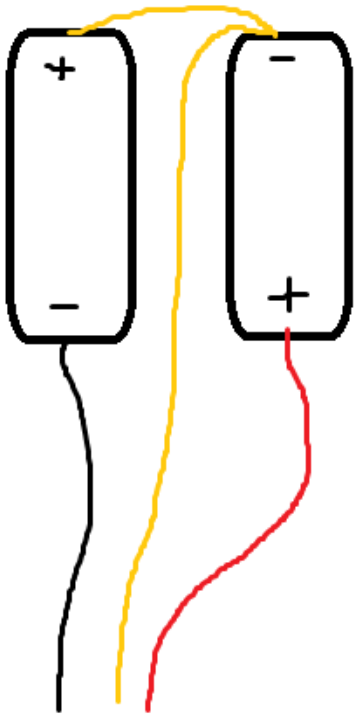
5/12/24:

Today I just tested the code again and made minor changes, along with 3d printing and rescanning the clay design I made. The print came out surprisingly well and I hot glued it to the MicroMouse shell. I did it like this so I could easy change where it is or put it on a another shell if I have time to redesign. At the moment it looks very out of place but I'm hoping to fix that before the review.



6/12/24

I focused on getting the battery soldered up and charged, the hardest part of this was trying to solder the wire that connects both positive and negative together, outlined in yellow in this diagram I made to help explain.



Once fully soldered, I taped it up to ensure that the connection points of the wires to the battery itself are not moving. This will hopefully prevent the wires from coming loose and allows the rest of the wire to be manipulated as normal.

12/12/24:

Today I went over the code and spent the hour supervised session testing and tweaking the code to go around the loop that had been set up.

13/12/24:

Today we got feedback from our peers and I'm going to use this to improve my project so far.

Participants	1: Ana	2: conor	3:
How could the laser cutting, other material usage be improved to ensure that they are the most appropriate use case possible.	4/5 The laser cutting & 3d modelling is good. I think the only thing that the mouse can benefit from is having technical drawings of the designs used.	5/5 The 3D printed parts were made well and conform nicely to the direction you were going. The use of laser cut wheels was smart with the design being unique	5/5 Wheels are excellent and show a good understanding of CAD. Could possibly be implemented for different parts
How could the physical form be improved; shell, motor mount, homer, pcb size and shape	3/5 Making a more stable way for the wheels to be attached to the motors would be good. Currently they look like they're held on by hot glue and prayers. The shell would benefit from some changes, maybe with a way to incorporate homer into the print without hot glueing him on. Maybe even add etching on the top to show leaves without having to paint it.	3/5 The overall profile of the PCB is rather blocky and may not be able to take around corners as great but the component placement is quite orderly. While homer is a nice touch, I do have to question why he wasn't stenciled on the actual board itself. The pieces seem to be attached via glue and tight pockets which may lose out on stability.	3/5 The general design is fun and functional but could use more stability. Maybe a chamfer or fillet. Perhaps more ways to secure it to the PCB.
How could the micromouse navigation and code be improved. -code readability, TCRT reading, speed, turning, general logic	4/5 The code logic looks good, the cases are well laid out. There is some comment duplication, and some parts of the code aren't commented at all.	4/5 The coding side of thing is not really my area of expertise so i am unable to critique much but it looks good for comments if a little copy-paste.	5/5 Including states is excellent. Could use a little more comments

One consistent feedback was that the shell should be changed as now only is it unstable but it looks out of place on the micromouse.

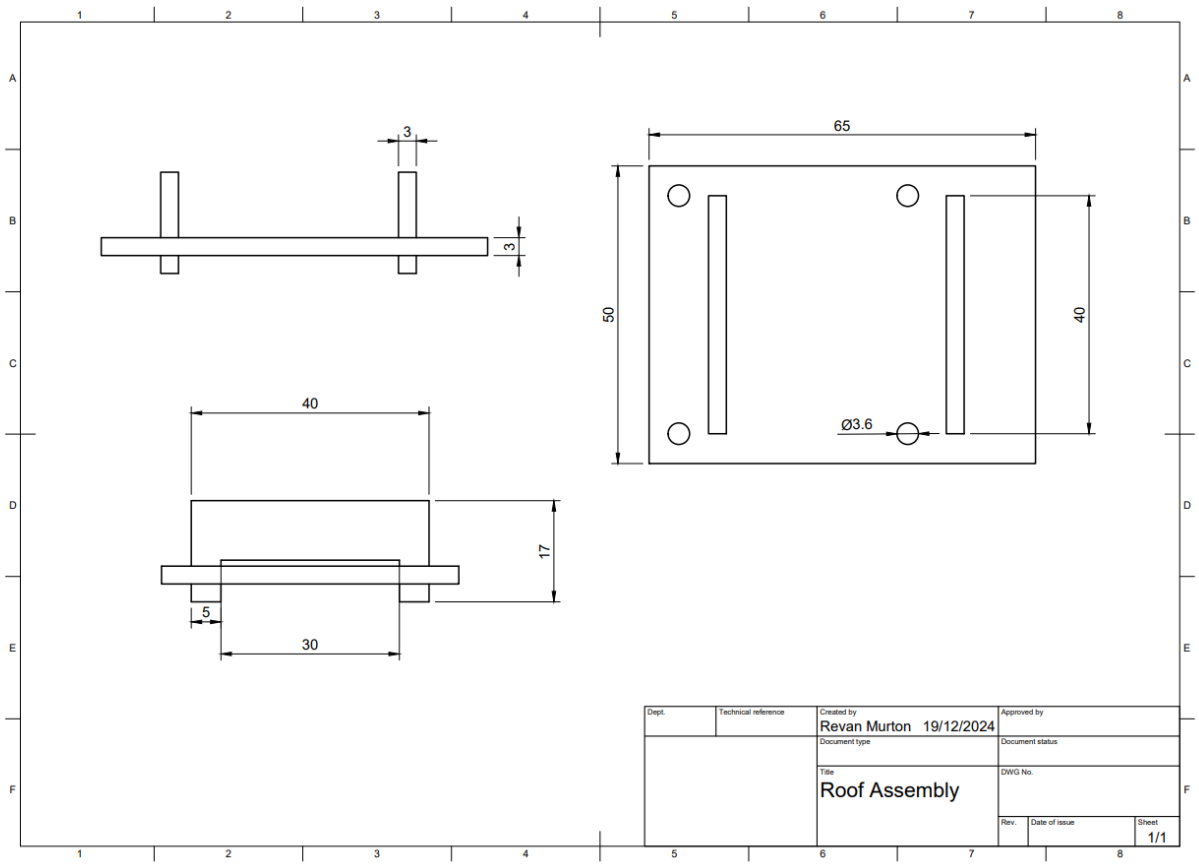
Another piece of feedback is how parts of the code is uncommented, I'll fix this as it can make understanding what the code does much easier and will help if I need to go back and change the code.

Need to add technical drawings, as at the moment there little documentation for the CAD inside the github, this helps with making the project much easier to replicate.

19/12/2024:

I've gone over the recieved feedback and have put the changes I made here

-To hopefully remedy this I've moved onto a new idea of making a roof that is held up using the m3 bolts and should provide room for the battery, be more secure and allow the robot to be powered on with the switch much easier.



-I’ve commented a lot of the code that were vague or confusing before.

```
//pins for the LED's
int blueLED = 19;
int redLED = 18;
int yellowLED = 20;

//pins for the TCRT
int leftTCRT = 26;
int middleTCRT = 28;
int rightTCRT = 27;

//Variables for the motors
int motorRA = 1;
int motorRB = 0;
int motorLA = 3;
int motorLB = 2;
int motorSpeed = 30;

unsigned long timer;

//Initialise mouse states
enum Mouse_enum { FORWARD,
                  TURNRIGHT,
                  TURNLEFT,
                  CROSSROAD };
enum Sensor_enum { RIGHT,
                   LEFT,
                   MIDDLE,
                   NONE };
static Mouse_enum mouse;
static Sensor_enum sensor;
```

```
void motorDriving(float speed, int motorA, int motorB) {  //func
                                                          //Make s
  if (speed < -255) {
    speed = -255;
  }
  if (speed > 255) {
    speed = 255;
  }

  //sets motors to spin forward, backwards or not at all depend
  if (speed > 0) {
    analogWrite(motorA, -speed);
    analogWrite(motorB, speed);
  }
  if (speed < 0) {
    analogWrite(motorA, speed);
    analogWrite(motorB, -speed);
  }
  if (speed == 0) {
    analogWrite(motorA, 0);
    analogWrite(motorB, 0);
  }
}

void turnLeft() {  //Turns left and then continues for a short t
  motorDriving(-100 , motorLB, motorLA);
  motorDriving(motorSpeed, motorRA, motorRB);
}
void turnRight() {  //Turns right and then continues for a short
  motorDriving(motorSpeed, motorLA, motorLB);
  motorDriving(-100, motorRB, motorRA);
}
void wallInfront() {  //Decides which way to turn depending on t
  if (analogRead(leftTCRT) < analogRead(rightTCRT)) {
    motorDriving(-motorSpeed, motorLB, motorLA);
    motorDriving(motorSpeed, motorRA, motorRB);
```

```
    delay(500);
  } else {
    motorDriving(motorSpeed, motorLA, motorLB);
    motorDriving(-motorSpeed, motorRB, motorRA);
    delay(500);
  }
}
void forwardMovement() {  //Moves the mouse forwards should acc
  float right = map(analogRead(leftTCRT), 800, 1030, 30,20 );
  float left = map(analogRead(rightTCRT), 800, 1030, 30,20);
  Serial.println(left);
  Serial.println(right);
  motorDriving(right, motorLA, motorLB);
  motorDriving(left, motorRA, motorRB);
}



void setup() {
  Serial.begin(9600);
  pinMode(leftTCRT, INPUT);
  pinMode(middleTCRT, INPUT);
  pinMode(rightTCRT, INPUT);
  pinMode(blueLED, OUTPUT);
  pinMode(redLED, OUTPUT);
  pinMode(yellowLED, OUTPUT);
  pinMode(motorLA, OUTPUT);
  pinMode(motorLB, OUTPUT);
  pinMode(motorRA, OUTPUT);
  pinMode(motorRB, OUTPUT);
}

void mouseStates(uint8_t Sensor_enum) {//each state has a LED th
  switch (mouse) {
    case FORWARD:  //Set the state depending on which direction
      if (sensor == LEFT) {
        timer = millis();
```

```
        mouse = TURNRIGHT;
      }
      if (sensor == MIDDLE) {
        mouse = CROSSROAD;
      }
      if (sensor == RIGHT) {
        timer = millis();
        mouse = TURNLEFT;
      }
      if (sensor == NONE) {
        forwardMovement();
      }
      break;

    case CROSSROAD:
      if (sensor != MIDDLE) {  //For each of these states a cor
        mouse = FORWARD;
        digitalWrite(blueLED, LOW);
      } else {
         digitalWrite(blueLED, HIGH);
        wallInfront();

      }
      break;

    case TURNLEFT:  //The timer means that the robot will turn
      if (millis() - timer >= 20) {
        digitalWrite(redLED, LOW);
        mouse = FORWARD;

      } else {
        digitalWrite(redLED, HIGH);
        turnLeft();
      }
      break;
```

```
    case TURNRIGHT:
      if (millis() - timer >= 20) {
        digitalWrite(yellowLED, LOW);
        mouse = FORWARD;
      } else {
        digitalWrite(yellowLED, HIGH);
        turnRight();
      }
      break;
  }
}


void loop() {
  /*
  testTCRT();
  motorTest();
  testLED();
  mouseStates(sensor);
  sensing();
  */
  Serial.println(analogRead(middleTCRT)); //reads TCRTs and pri
  Serial.println(analogRead(leftTCRT));
  Serial.println(analogRead(rightTCRT));
  if ((analogRead(leftTCRT) < 990) && (analogRead(middleTCRT) <
    sensor = LEFT;
  }
  else if ((analogRead(rightTCRT) < 990) && (analogRead(middleT
    sensor = RIGHT;
  }
  else if (analogRead(middleTCRT) < 400) {
    Serial.println(analogRead(middleTCRT));
    sensor = MIDDLE;
  }
    else {
    sensor = NONE;
```

```
  }

  mouseStates(sensor); //Run state machine for the mouse states
}


//Test code
/*
void testLED() {  //Tests the LED's
  digitalWrite(redLED, HIGH);
  digitalWrite(yellowLED, HIGH);
  digitalWrite(blueLED, HIGH);
  delay(1000);
  digitalWrite(redLED, LOW);
  delay(1000);
  digitalWrite(yellowLED, LOW);
  delay(1000);
  digitalWrite(blueLED, LOW);
  delay(1000);
}
void testTCRT() {  //Tests the TCRT's by turning LED's on if the
  if (analogRead(leftTCRT) < 900) {
    digitalWrite(redLED, HIGH);
  }
  if (analogRead(leftTCRT) > 900) {
    digitalWrite(redLED, LOW);
  }

  if (analogRead(middleTCRT) < 900) {
    digitalWrite(yellowLED, HIGH);
  }
  if (analogRead(middleTCRT) > 900) {
    digitalWrite(yellowLED, LOW);
  }

  if (analogRead(rightTCRT) < 900) {
```

```
    digitalWrite(blueLED, HIGH);
  }
  if (analogRead(rightTCRT) > 900) {
    digitalWrite(blueLED, LOW);
  }
}
void motorTest() {  //Tests the motors
  digitalWrite(motorRA, HIGH);
  digitalWrite(motorRB, LOW);
  digitalWrite(motorLA, HIGH);
  digitalWrite(motorLB, LOW);
}

void sensing() {  //Sets the state of the sensors, currently unu
  if (analogRead(leftTCRT) && analogRead(middleTCRT) < 950) {
    sensor = LEFT;
  }
  if (analogRead(rightTCRT) && analogRead(middleTCRT) < 950) {
    sensor = RIGHT;
  }
  if (analogRead(middleTCRT) < 800) {
    sensor = MIDDLE;
  }

  else {
    sensor = NONE;
  }
}

void movement() { //Basic Movement function, currently unused

  if (analogRead(leftTCRT) && analogRead(middleTCRT) < 970) {
    turnRight();
  }
  if (analogRead(rightTCRT) && analogRead(middleTCRT) < 970) {
    turnLeft();
```

```
  }
  else if (analogRead(middleTCRT) < 900) {
    wallInfront();
  } else {
    forwardMovement();  //Moves the mouse forwards should accom
  }
}
*/
```

-I’ve added the drawings for all of the components I have so far.

19/12/24 - 06/01/25:

Over the course of the Christmas break, I spent the majority of the time going over the read me and adding more drawings and renders to the repo. The code was modified slightly, mostly additional commenting.