Sumo Robot 5

Team Members: Kevin Ng, Jacob McCray, and Corbin Krecklow

May 27, 2022

Class: Junior Design 2

Lab Section: 002

Contents

1 Introduction

The goal of this project is to create a sumo robot. Sumo robots are a product of Robot-sumo, where two robots face each other and attempt to push each other out of a flat ring. For this assignment, the team is required to design, manufacture, and construct a sumo robot from scratch that meets the following **customer requirements:**

1. Able to find and push an object

At least 9 out of 10 times, must be able to push a sumo-bot sized object out of the regulation* ring without human intervention

2. Debuggable

Ability to read sensor values of each sensor (navigation, wheel encoders, battery sensors, and other sensors) while the robot is competing in a match. Only one sensor is required to be displayed at a time

3. Heavy enough to be competitive

Must weigh at least 95% of the maximum robot weight (500 grams for the mini-sumo robot).

3. Heavy enough to be competitive

Must weigh at least 95% of the maximum robot weight (500 grams for the mini-sumo robot).

4. Reasonable Battery Life

The system will move for at least 15 continuous minutes on a single charge

5. Aesthetically pleasing

Wiring must be grouped with split loom or other wiring organization material. Wiring will be routed through zip ties secured to the chassis. No cardboard or tape will be visible on the final project

6. Wireless Charging

Must be able to be charged wirelessly

7. Aggression

Must have toggle-able “Aggressive” mode

2 Electrical Schematic

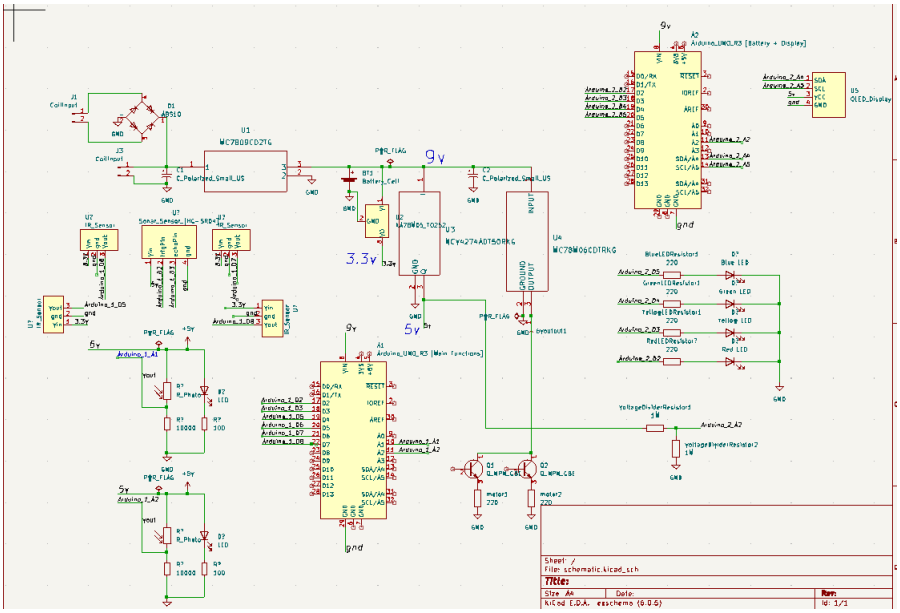

Figure 1: Entire Electrical Schematic

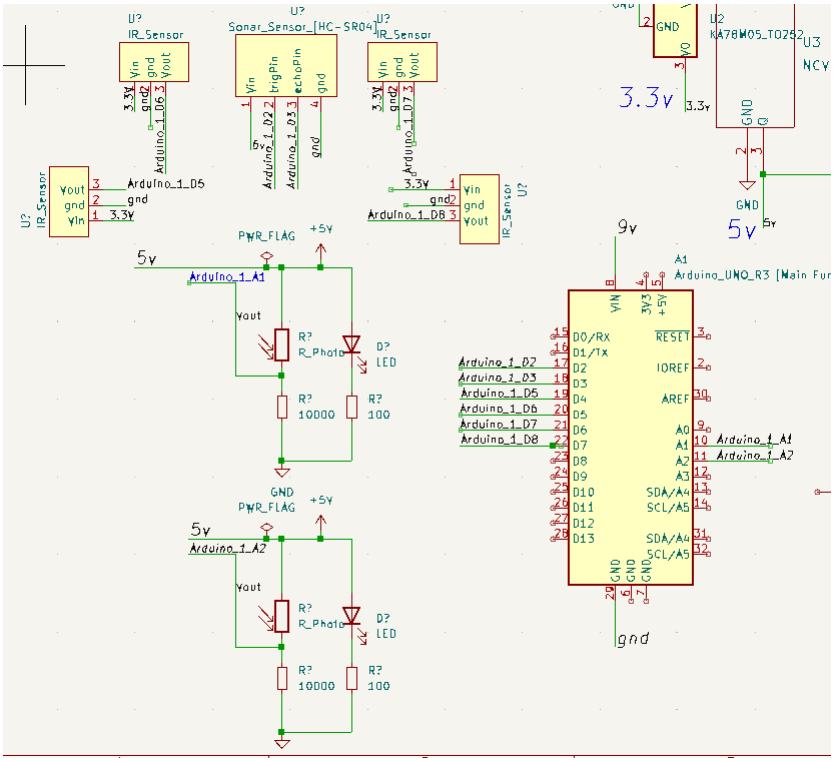

Figure 2: Arduino 1 Electrical Schematic

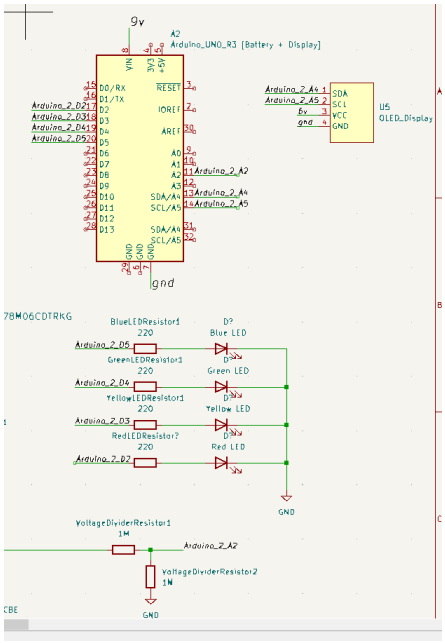

Figure 3: Arduino 2 Electrical Schematic

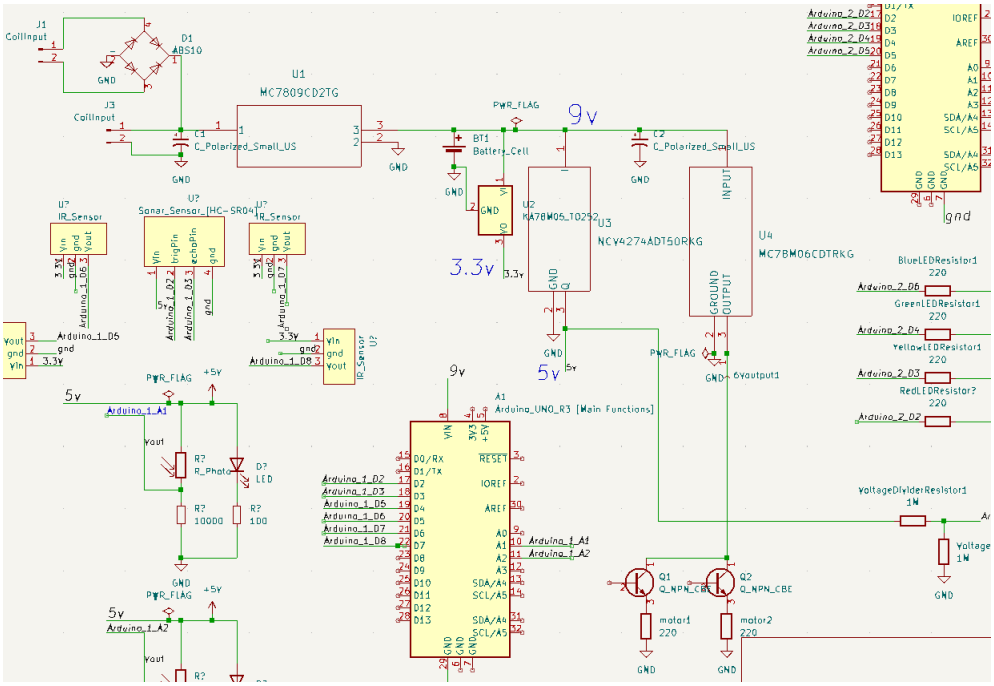

Figure 4: Electrical schematic Zoomed Center

KiCAD Schematic File Attached

3 All Code

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 32 // OLED display height, in pixels

const unsigned char PROGMEM anger [] = {0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
const unsigned char PROGMEM sunglasses [] = {0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x
const int analogPin = A2;
float analogValue;
float input_voltage;

int ledPins[] = { 1, 2, 3}; // The LEDs attach to this array of pin numbers
int pinCount = 3;           // the number of pins used for LEDs

int xx = 40;
int yy = 0;

#define OLED_RESET     4 // Reset pin # (or -1 if sharing Arduino reset pin)
#define SCREEN_ADDRESS 0x3C ///< See datasheet for Address; 0x3D for 128x64, 0x3C f
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

#define echoPin 13
#define trigPin 10
#define SonarVCC 5
#define sonarLED 4
#define pushToggle 9
int Motor_1B = 11;
int Motor_2B = 12;
long duration;
bool toggle = true;
```

```
void setup() {
  // put your setup code here, to run once:
  pinMode(analogPin, INPUT);
  Serial.begin(9600);
  pinMode(Motor_1B, OUTPUT);
  pinMode(Motor_2B, OUTPUT);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(SonarVCC, OUTPUT);
  pinMode(sonarLED, OUTPUT);
  pinMode(pushToggle, INPUT);
  digitalWrite(SonarVCC, HIGH);

  if(!display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS)) {
    Serial.println(F("SSD1306 allocation failed"));
    for(;;); // Don't proceed, loop forever
  }

  // Clear the buffer
  display.clearDisplay();

  drawsunglasses();
  delay(5000); //so as to meet regulation and wait 5 secs before beginning
}

void Forward() {
  digitalWrite(Motor_1B, HIGH);
  digitalWrite(Motor_2B, HIGH);
}
void Left(){
  digitalWrite(Motor_1B, LOW);
  digitalWrite(Motor_2B, HIGH);
}
void Right(){
  digitalWrite(Motor_1B, HIGH);
  digitalWrite(Motor_2B, LOW);
```

```
}
void Backward(){
  digitalWrite(Motor_1B, LOW);
  digitalWrite(Motor_2B, LOW);
}
void Stop() {
  digitalWrite(Motor_1B, LOW);
  digitalWrite(Motor_2B, LOW);
  }

int distance(){
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  //Reads the echoPin
  duration = pulseIn(echoPin, HIGH);

  return duration *0.034 / 2;
}

void LED_function(int stage)
{
    for (int j=2; j<=4; j++)
    {
    digitalWrite(j,LOW);
    }
    for (int i=1, l=2; i<=stage; i++,l++)
    {
    digitalWrite(l,HIGH);     //delay(30);
    }
}

void drawsunglasses(void) {
  display.clearDisplay();
```

```
  display.drawBitmap(xx, yy,sunglasses,48,47, 1);
  display.display();
  delay(100);
}

void drawmad(void) {
  display.clearDisplay();
  display.drawBitmap(xx, yy,anger,48,48, 1);
  display.display();
  delay(100);
}

void isawthat(void){
  digitalWrite(sonarLED, HIGH);
  delay(1000);
  digitalWrite(sonarLED, LOW);
}

void loop() {
  int ButtonState = digitalRead(pushToggle);
  analogValue = analogRead (analogPin);
  input_voltage = (analogValue * 5.0) / 1024.0;
  Serial.println(input_voltage);

  Serial.print("\n");
  if (input_voltage < 0.6 ){
    toggle = !toggle;
  }
  Serial.print(toggle);
  //Moods Portion
  switch (toggle){
    //Sensible Mode
    case true:
    {
        drawsunglasses();
        Serial.print ("Default\n");
```

```
    int range = distance();
    Serial.print(range);
    digitalWrite(sonarLED, LOW);
    int photoResistor1 = analogRead(A1);
    if (range < 50 && range!=0){

      isawthat();
      do{
        digitalWrite(sonarLED, HIGH);
        photoResistor1 = analogRead(A1);
        Serial.print("Lightsensor 1:");
        Serial.print(photoResistor1);
        Serial.print("\n");
        Forward();
        delay(500); //FOR TESTING
      }while(photoResistor1 < 600); //|| photoResistor2 > 520
      Stop();
       digitalWrite(sonarLED, LOW);
      Serial.print("Line Reached");
      Left();
    }
    else{
      Left();
      digitalWrite(sonarLED, LOW);
      Serial.print("Searching \n");
    }
    break;
}
case false:
{
    //Aggressive Mode
    drawmad();
    Serial.print("Aggresive Mode\n");
    int range = distance();
    Serial.print(range);
    int photoResistor1 = analogRead(A1);
```

```
if (range < 50  && range!=0){
  isawthat();
  if (range%2 == 0){
    Right();
    delay(750);
    Forward();
    delay(4000);
    Left();
    delay(600);
    do{
      digitalWrite(sonarLED, HIGH);
      photoResistor1 = analogRead(A1);
      Serial.print("Lightsensor 1:");
      Serial.print(photoResistor1);
      Serial.print("\n");
      Forward();
      delay(500);
    }while(photoResistor1 < 600); //&& photoResistor2 > 520
    Stop();
    digitalWrite(sonarLED, LOW);
    Serial.print("Line Reached");
    Left();
   }
 else{
    Left();
    delay(750);
    Forward();
    delay(4000);
    Right();
    delay(600);
    do{
      digitalWrite(sonarLED, HIGH);
      photoResistor1 = analogRead(A1);
      Serial.print("Lightsensor 1:");
      Serial.print(photoResistor1);
      Serial.print("\n");
```

```
            Forward();
          }while(photoResistor1 < 600); // && photoResistor2 > 520
          Stop();
          digitalWrite(sonarLED, LOW);
          Serial.print("Line Reached");
          Left();
         }
        }
      else{
        Left();
        Serial.print("Searching \n");
      }

      break;
  }
}
//Lights portion

if (input_voltage < 3.25)
  {
  //LED_function(3); //2 pins lit up
  //digitalWrite(3, HIGH);
   Serial.print("HERE");
  }
else if (input_voltage < 3.5 && input_voltage >= 3.25)
  {
 LED_function(3); //2 pins lit up
 //digitalWrite(3, HIGH);
  //Serial.print("HERE");
  }
else if (input_voltage >= 3.5)
  {
 //LED_function(3);  //3 pins lit up
 //digitalWrite(3, HIGH);
  Serial.print("HERE");
  }
```

```
  else
    {
    //LED_function(3);
    Serial.print("HERE");
    //digitalWrite(4, HIGH);
      delay(1000);
     // LED_function(0);
      delay(1000);
    }
  //Use delay if debugging
  //delay(1000);

}
```

Summary:

There are two primary modes that the robot runs in. In aggressive mode, the robot performs a pit maneuver to attempt to hit its target in the side. On the other hand, in sensible mode the maneuver will simply go straight towards an object until it sees the white line. The code also uses an analog pin to read in the battery level and display it using three LEDs. Moreover, while in either two modes the robot will display a corresponding emote on it's display. This uses two libraries plus two arrays with pixel assignments.

4 The Chassis of the Robot

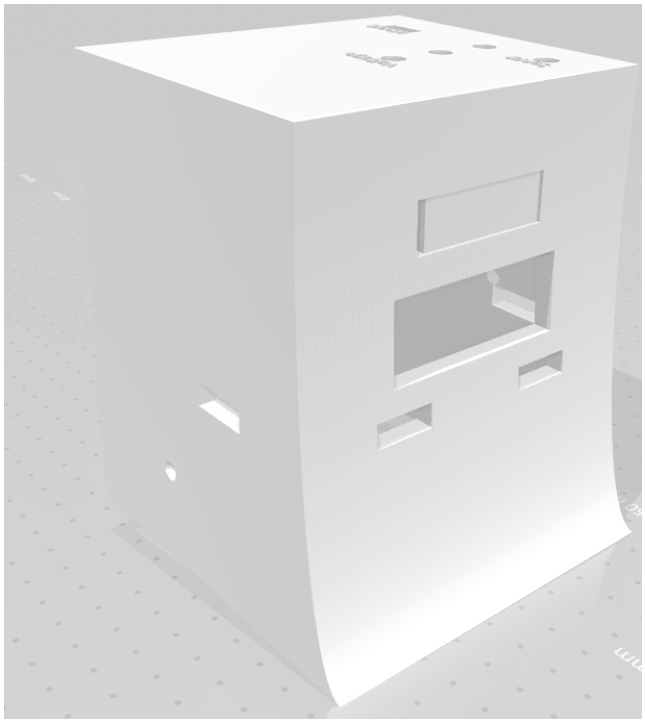

Figure 5: General View of Chassis

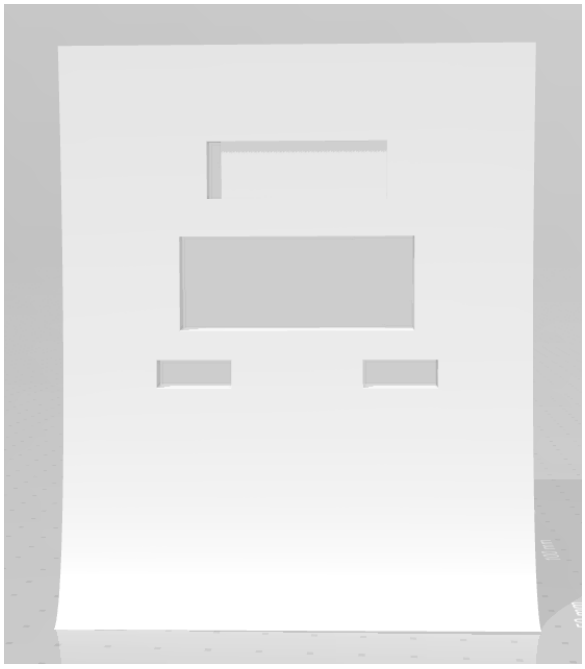

Figure 6: Front View of Chassis

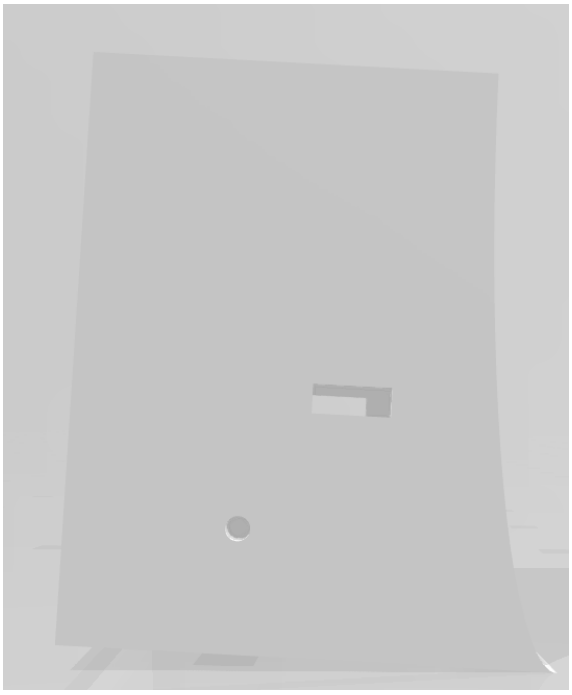

Figure 7: Side View of Chassis

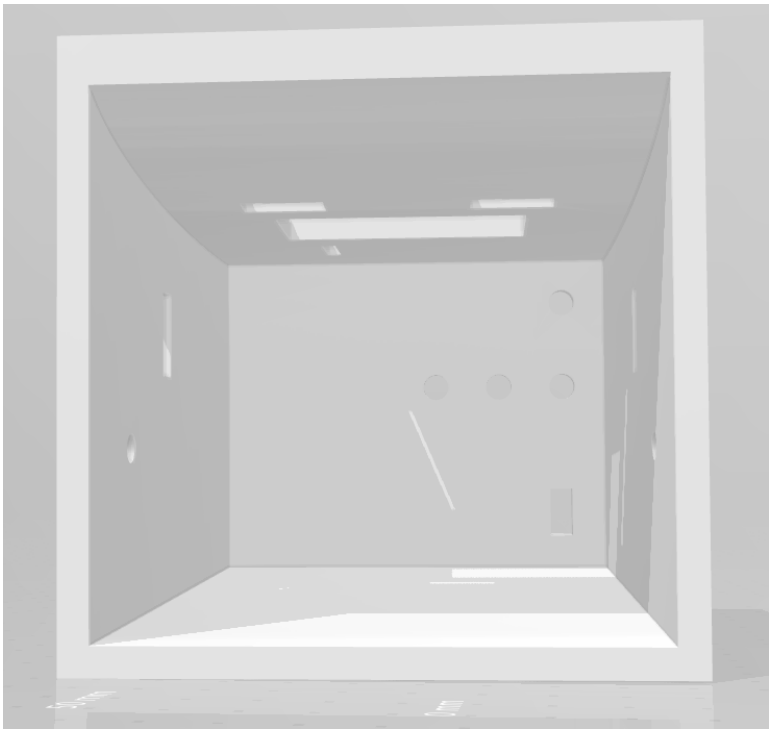

Figure 8: Bottom View of Chassis

Figure 9: Top View of Chassis

3D Chassis Model Is Attached To This Submission. Cost unknown due to fee being waived.

5 Top Level System Diagram

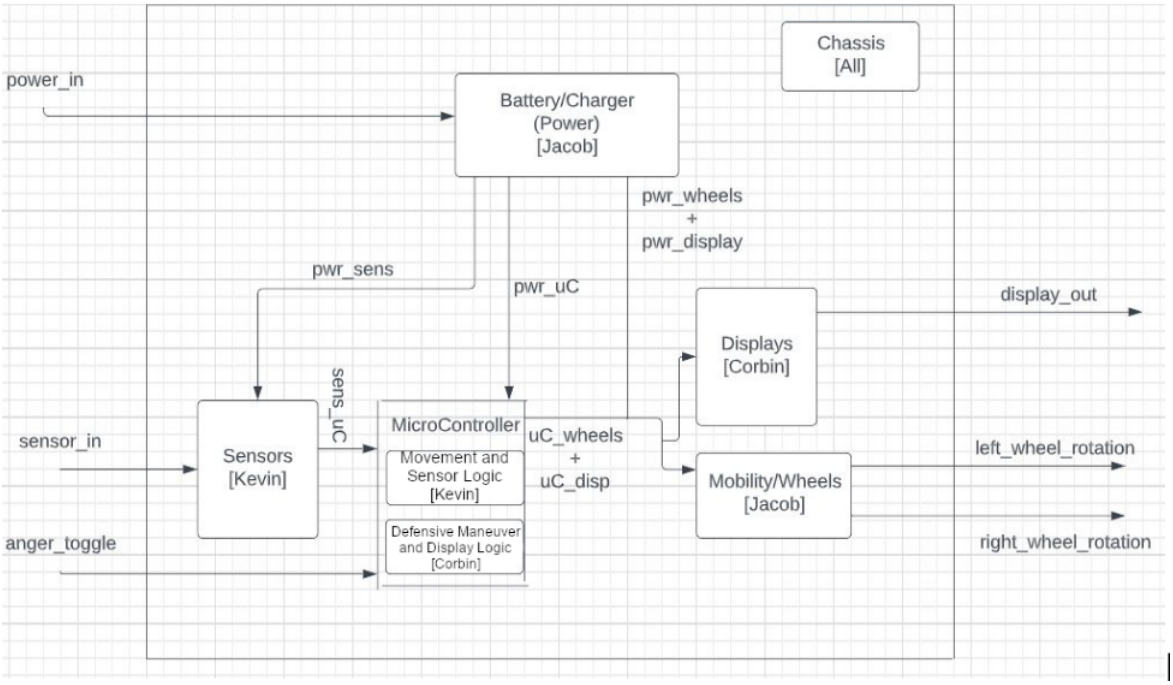

Figure 10: Top Level System Diagram

6 Interfaces and Properties

Below is the interface and properties table of our robot. (These were taken as screenshots due to formatting issues).

Interface	Type	Specifications
power_in	dcpwr	$V_{nominal}$: 5V $I_{nominal}$: 2A
sensor_in	envin	Properties: Distance (cm) • Sonar sensor Light (light sensor) • Photoresistor + LED Touch sensor • Button ○ Gives digital high when pressed
anger_toggle	usrin	active high switch V_{range}: 0V or Floating Voltage $I_{nominal}$: ~20mA
pwr_sens	dcsig	Sonar Sensor: • $V_{nominal}$:5V • $I_{nominal}$: 15mA Light sensor: • LED ○ Forward voltage: 2.7V - 4.2 ○ Nominal current: 30mA ○ Max current: 100mA • Photoresistor: ○ V_{range}: 0.23 - 4.54V ○ I_{range}: 0.024 - 0.45 mA Touch sensor: • $V_{nominal}$:5V • $I_{nomanal}$: 50mA
sens_uC	comm	Sonar Sensor: • Serial input • Code: C++ setup • Data: Output digital high pulse. Read Digital. Records pulse in time, use equation

		to compute distance in cm Light sensor: • Code: C++ setup • Data: Read analog voltage @ pin • Code will print a serial output high or low depending on the voltage. Touch sensor: • Code: C++ Setup • Data: Read Digital High or Low
pwr_uC	dcpwr	$V_{operation}$: 5V V_{input}: 7-12V $I_{nominal}$: ~20mA
uC_wheels	analog	Code: C++ Highest V_{nom}: 5V, 11mA Medium V_{nom}: 2.5V, 11mA Lowest V_{nom}: 0V, 0mA
uC_disp	[Various (see below)]	Code: C++ Serial output
uC_disp: VoltageSig	dcpwr	HIGH or LOW V_{nom}: 3.3V I_{nom}: 15mA
uC_disp: FaceInfo	Data	OLED Arduino display Data: "Face" in form of text determined by uC V_{nom}: 5V I_{nom}: [regulated by chip]
uC_disp: SonarSig	dcpwr	HIGH or LOW V_{nom}: 3.3V I_{nom}: 15mA
pwr_wheels	dcpwr	V_{nom}: 6V $I_{nominal}$: ~70mA I_{max}: ~400mA

display_out	envir	Image/Word displayed on OLED and LEDs
left_wheel_rotation	envin	Motor on
right_wheel_rotation	envin	Motor on

Figure 11: Interface and Properties Table

7 PCB Layers

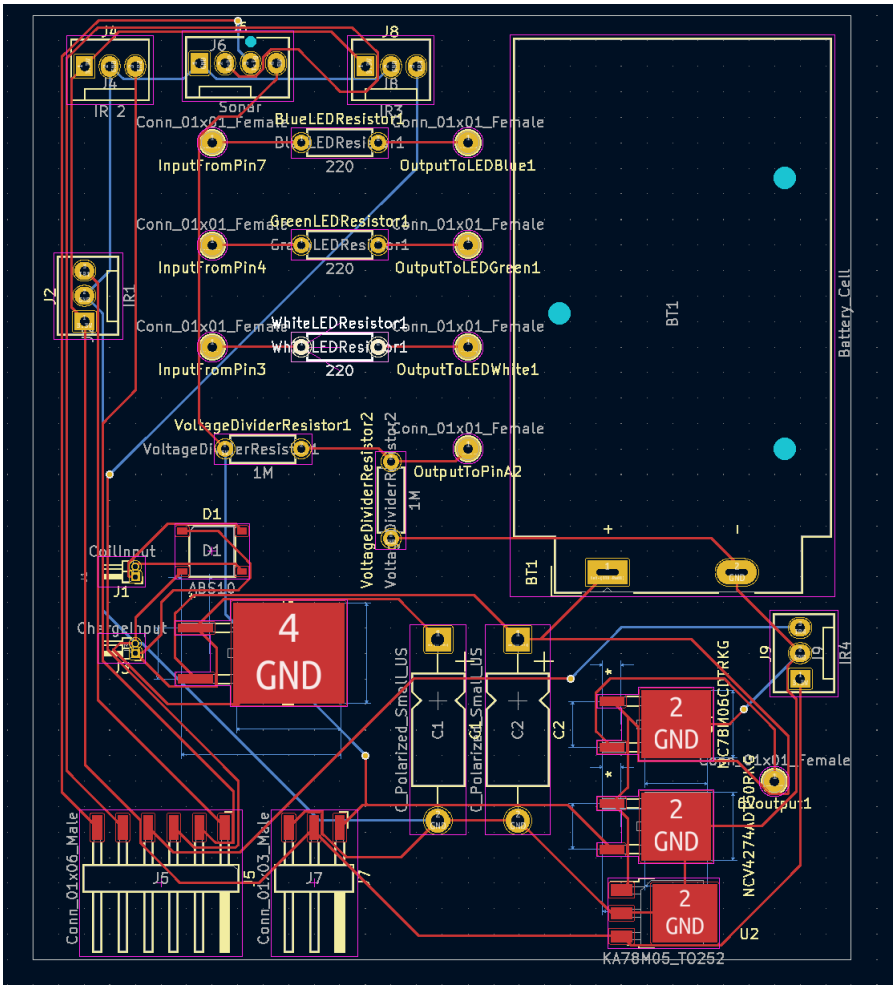

Figure 12: PCB 1

PCB 1 Description:

1. Cost: $50 ($60 including components (resistors, capictors, mosfets..)
2. Houses Main Power Supply (9V Battery)
3. Supplies 5v, 3.3v, and gnd pin out
4. Has female pins to interface with sonar and IR sensors
5. Has Male pins which will be wired to the digital pins of the micro-controller
6. Through-hole pins for both LEDS and Micro controller used in battery level indicator

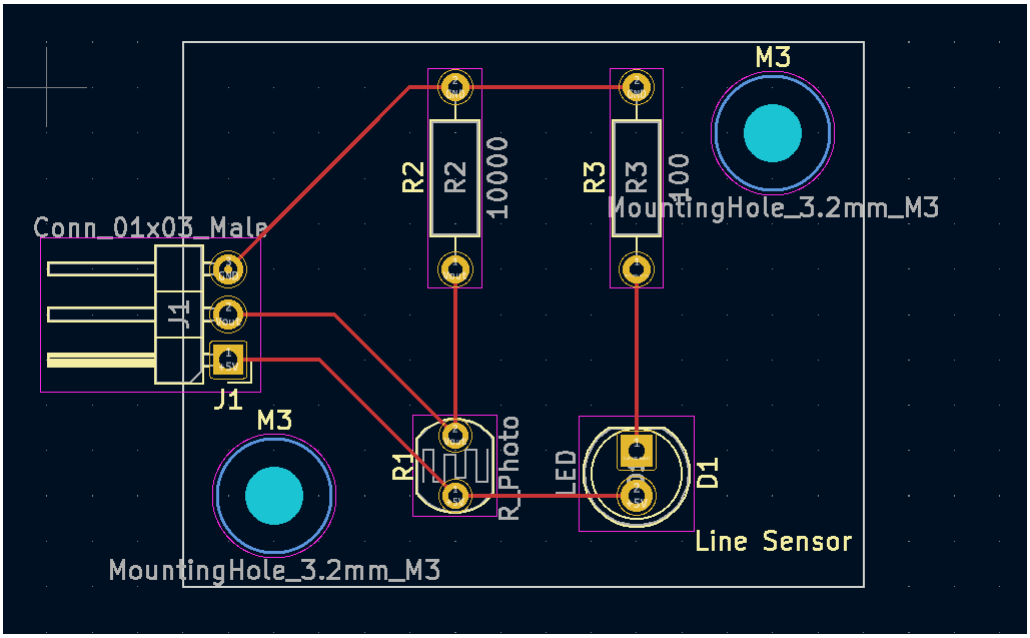

Figure 13: PCB 2

PCB 2 Description:

1. Cost: $20
2. Line Sensor
3. Has three pins: Vin, GND, and Vout
4. Uses photo resistor in a voltage divider format to produce an analogue value that can be read by the micro-controller when attached to the Vout pin

8 Bill of Materials

ITEM	Quantity	Price
PCB 1 + Components	1	$61.45
PCB 2	1	$22.35
Arduino Uno REV3 [A000066]	2	$45.54
ACEIRMC HC-SR04 Ultrasonic Sensor Distance Module	1	$9.29
HiLetgo 10pcs IR Infrared Obstacle Avoidance Sensor Module for Arduino Smart Car [3-01-0063]	1	$8.78
eBoot 30 Pieces Photoresistor Photo Light Sensitive Resistor Light Dependent Resistor 5 mm GM5539 5539	1	$5.35
Gikfun 6x6x4.3mm TACT Switch Push Button for Arduino PCB (Pack of 50pcs) EK1019	1	$6.88
Chanzon 100 pcs 5mm White LED Diode Lights (Clear Round Transparent DC 3V 20mA) [100F5T-YT-WH-WH]	1	$6.99
Bundle of Wires	1	$2.50
2 pieces I2C display module 0.91 inch I2C OLED display module I2C OLED screen driver dc 3.3V - 5V	1	$9.99
GM3 Gearmotor 6V DC	2	$12.00
10 RGB LEDs	10	$2.40
Total		$193.47

Figure 14: Bill of Materials Table

9 Time spent by each person

For this project, each team member estimates that they spent **at least 20 hours** respectively working on this project.

1. Kevin Ng: 20 Hours working on sensors and micro-controller + time in lab
2. Jacob McCray: 20 Hours working on power and motors + time in lab
3. Corbin Krecklow: 20 Hours working on display and "aggressive' micro-controller + time in lab