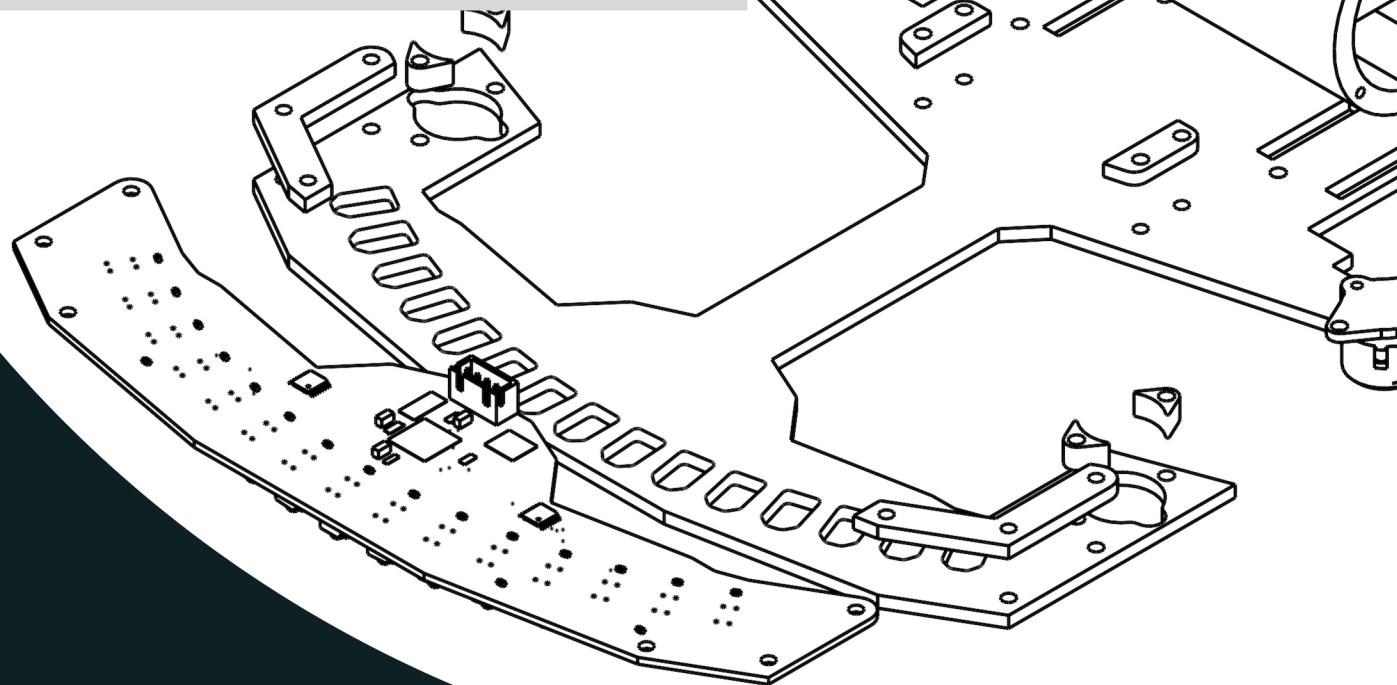


Line Follower

A High Performance
Line Following Robot



By
Chang Seng Chon (5C05)
Tam Chon Man(5C31)

Abstract

This project set out with the aim to construct an autonomously guided line-following robot capable of high-speed navigation while maintaining stability and precision on the track. Leveraging a suite of 15 infrared sensors and a robust carbon fiber frame, we meticulously engineered a robot that fulfills stringent competition criteria. The selected Nidec 24H brushless motors, known for their lightweight and integrated driver circuits, contributed to the robot's agile yet stable performance. Our result was a streamlined robot that successfully completed track cycles in under 10 seconds, with a weight of 2.32 kg and dimensions within the 200*200*150 mm limit. It achieved speeds over 2 m/s, demonstrating the effective synergy between sophisticated hardware design and user-centric software development.

Table of Contents

Abstract	1
Table of Contents	2
Introduction and Motivation	4
Relative Work/background	5
Rule of competition	6
Aim	6
Bill of materials.....	7
Hardware and Electronics Design	8
Mechanism	8
Selection of Materials.....	8
Selection of motor	9
Gear set design.....	12
Follower Design	14
Electronics	19
Selection of Sensor	19
Sensor board	21
Mother Board	26
Prototype.....	30
Software Development.....	31
Integration.....	31
Communication between sensor board and mother board:.....	33
Calibration	33
Error Value Transmit	33
User Interface	34
Result.....	37
Challenges	38

<i>Future Development</i>	39
<i>Conclusion</i>	40
<i>References</i>	41
<i>Appendix</i>	<i>Error! Bookmark not defined.</i>

Introduction and Motivation

In general, the line follower robot is one of the self-operating mobile machines that follows a ‘line’ or ‘path’ already predetermined by the user. This line or path may be as simple as a physical white line on the floor or as complex path marking schemes e.g. embedded lines, magnetic markers and laser guide markers. In this project we are using certain number of sensors to detect these specific lines, also outlining the design construction and testing of our line follower robot.

In this project, we used 15 reflective optical sensors (TCRT5000) which include an infrared emitter and phototransistor in a leaded package which blocks visible light. In terms of accuracy, we implement a calibration algorithm to eliminate or minimize factors that cause inaccurate measurements. It is a fundamental aspect of instrumentation design simply by taking the maximum and minimum of the sensors. Due to the limitation of Arduino board, we decided to use two multiplexers in order to reduce the usage of pin.

The innovation within the project that are seeking to be acquired and recognized by external parties:

- Engaging way to understand concepts related to sensors, control systems, and autonomous navigation
- Enhance our creativity and problem-solving abilities (e.g. process of designing printed circuit board (PCB), designing a gear box, drafting and producing CAD)
- Further exploration of algorithms for line detection and tracking
- Endeavor to present a fully developed line following system

Relative Work/background

Line-sensing robots, also known as line-following robots, are autonomous robots that use sensors to detect and follow a line on the ground. They are widely used in industrial automation, robotics competitions, and educational projects. Early versions of these robots used simple analog circuits and light sensors to detect the line's position. They typically had wheels or tracks for locomotion. In recent years, with the advancement of technology and the availability of affordable microcontrollers and sensors, line-following robots have become popular educational tools for teaching programming, electronics, and robotics. They provide a hands-on approach to learning about sensors, control systems, and autonomous navigation. Infrared (IR) sensors and proportional-integral-derivative (PID) control are commonly used in these robots.

Line following robot in industrial application:

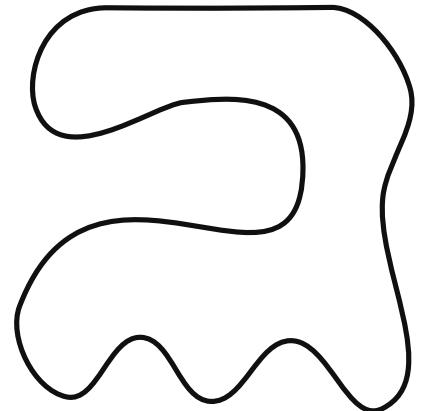


Rule of competition

2 students in a group, design and make an autonomously guided robot that follows a line drawn on the ground to detect a dark line on a white surface and participate in inter-class competition while the size of robot is limited 200(L)x200(W)mm. There are 3 rounds in the match, each round the robots are able to run 3 loops. The final result is taken the best single loop time.

These are the following violations who commit serious infractions will be disqualified from the competition:

- Intentional damage to the competition area or target objects
- Touching or tampering with other participants' works or target objects without permission from the staff during the competition
- Participating with a work that does not meet the specified requirements



The track of the competition

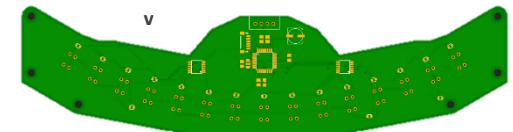
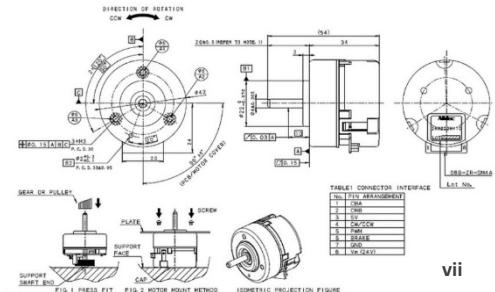
Aim

This project aims to perform an autonomously guided robot with the capability of moving at high speeds while maintaining stability and accuracy. Nevertheless, the robot should be extremely sensitive to the changes of brightness, ensuring that the robot stays on track and avoids deviating from the path.

- Time used to travel one cycle: under 10 seconds
- Weight: under 2.5kg
- Dimension: within 200*200*150 mm
- Maximum speed: over 2 m/s (7.2 km/h)
- No. of Infrared sensor: 15
- Power supply: 11.1V lithium-ion battery (3 cell Li-po Battery)
- Develop an easy way for monitoring, PID tuning, and debug

Bill of materials

- i. 3D printed gearbox x 2
- ii. Carbon fiber frame x 1
- iii. Tcrt5000 IR sensor x 15
- iv. Mother Board x 1
- v. Sensor Board x 1
- vi. NRF24 x 2
- vii. Nidec 24h brushless motors x 2
- viii. Ball caster x 2
- ix. Washer x Multiple
- x. Cups of coffee x 13



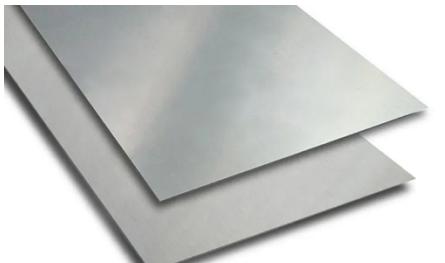
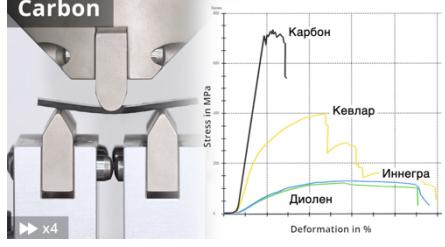
Hardware and Electronics Design

Mechanism

Selection of Materials

An appropriate selection of materials while designing a robot can make a big difference in the performance including durability, stability, weight and cost of your robots

In this project, we undertake a comprehensive analysis from various perspectives as above, and gain a thorough understanding of different subject matter. Through evaluation, we have reached a consensus on choosing "Carbon fiber".

Steel	Acrylic (PMMA)	Carbon Fiber (✓)
		
<p>Advantages:</p> <ul style="list-style-type: none">➤ Strong➤ Durable➤ Able to withstand heavy loads➤ Rigid <p>Drawbacks:</p> <ul style="list-style-type: none">➤ Corrodes easily➤ Extremely heavy (increase the energy consumption in the motor)➤ Conducts heat (complicated the cooling effect)	<p>Advantages:</p> <ul style="list-style-type: none">➤ Commonly used➤ Same qualities as glass without the fragility issue (will not break into piece easily)➤ Safe material➤ Lightweight <p>Drawbacks:</p> <ul style="list-style-type: none">➤ Brittle when under load (especially under an impact force)➤ Low wear resistance (obtain scratches easily)	<p>Advantages:</p> <ul style="list-style-type: none">➤ High mechanical strength➤ Light weight➤ Low thermal expansion➤ Affordable➤ High stiffness  <p>The graph plots Stress in MPa on the y-axis against Deformation in % on the x-axis. Four curves are shown: Carbon (black), Kevlar (yellow), Diolene (blue), and Innegra (green). Carbon shows the highest peak stress and stiffness. Kevlar has a lower peak stress and higher deformation before failure. Diolene and Innegra show the lowest peak stresses and highest deformations.</p> <p>Drawbacks:</p> <ul style="list-style-type: none">➤ Conduct electricity

- | | | |
|--|---|--|
| <ul style="list-style-type: none"> ➤ Considerably more expensive than other materials | <ul style="list-style-type: none"> ➤ Stiffness (not rigid enough to hold the main body of our robot) | |
|--|---|--|

→ In summary, considering the massive size and weight of our gearboxes and motors, carbon fiber emerges as the most suitable material due to its unique properties.

Selection of motor

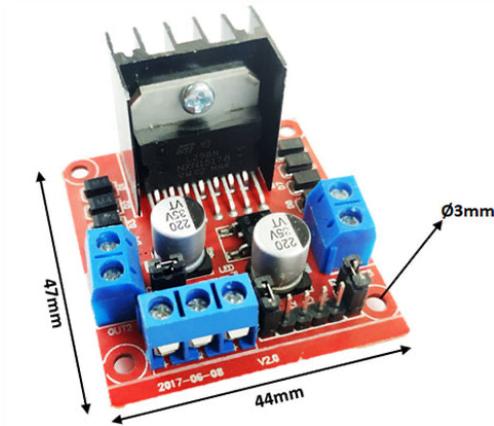
In this project, we have chosen to use “Nidec 24H055M020” (Nidec 24H), a brushless DC motor, as the actuator of the line-following robot. Here are several factors we have considered:

Name	Tamiya 70093 3-Speed Crank-Axle Gearbox Kit	Namiki 22CL-3501PG	N20 Motor	Nidec 24H055M020	RoboMaster M2006 P36
Category	Brushed			Brushless	
Gear ratio	17:1/58:1/203:1	80:1	298:1	1:1	36:1
Min Speed (rpm)	17:1⇒ 481(2V) 58:1⇒ 141(2V) 203:1⇒ 40(2V)	N/A	100 (6V)	150	N/A
Max Speed (rpm)	17:1⇒ 1202(5V) 58:1⇒ 353(5V) 203:1⇒ 101(5V)	120	200 (12V)	3900	416
Voltage (V)	3 - 6	12	6 - 12	12	12
Driver Required	External H-Bridge DC <u>driver</u> required			Built-in driver	External ESC/FOC driver required
Weight (g)	40	5000	9.5	110	90
Photo					

1. Usability

Unlike the Nidec 24H motor, many commercial motors typically require external H-Bridge or Electronic Speed Control (ESC) drivers for operation. This setup required additional space, especially since two drivers are needed to manage both left and right motors.

Therefore, motors with integrated/built-in drivers are preferred due to their more compact and efficient design.



L298N H-Bridge Driver



AGFRC 13A ESC

2. Motor Category:

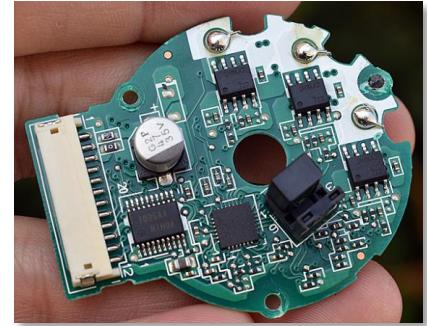
There are two typical types of motor, brushed and brushless.

Comparison of Features between brushed and brushless motor

- BRUSHLESS MOTOR:
 - **Longer Lifespan:** Without brushes, brushless motors experience less wear and tear, resulting in a longer operational life.
 - **Higher Efficiency:** Brushless motors are more efficient, generating less heat and using less energy for the same power output.
 - **Improved Performance:** They offer superior speed control and torque (compared to brushed motor).
 - **Reduced Low-Speed Torque:** BLDC motors can struggle to provide high torque at low speeds, which can be a limitation in applications that require strong torque from a standstill or during slow-speed operations.

- BRUSHED MOTOR:
 - **Lower Initial Cost:** Brushed motors are generally less expensive upfront than BLDC motors. This is due to their simpler design and fewer component requirements.
 - **Simpler Control and Drive Electronics:** They require less complex control and drive electronics, making them easier to integrate and operate in various applications.

Comparing the merits of both types of motor, we consider **brushless DC motor** a suitable actuator for our application because brushless motor offers a handful of merits, including a faster speed, a higher efficiency, and a longer lifespan. Despite the fact that BLDC required a driver (ESC/FOC) to operate, some brushless DC motors, such as Nidec 24H, are integrated with the driver circuits.



Integrated BLDC driver in Nidec 24H

This integration not only simplifies the system design but also enhances overall performance and reliability, making Nidec 24H an ideal choice for our line follower which aims to travel at high speed with a compact body.

3. Weight

The weight of our selected motors directly factors the nimbleness and maximum speed of our application. The Namiki 22CL-3501PG weighs 5kg, which is a considerable factor in determining the overall agility and speed capabilities of our system. Whereas Nidec 24H weighed merely 110g, which is a suitable motor weight for our robot.

In conclusion, we've chosen the Nidec 24H as the actuator for its numerous advantages: its brushless design ensures a longer lifespan and higher efficiency, reducing wear and energy consumption.

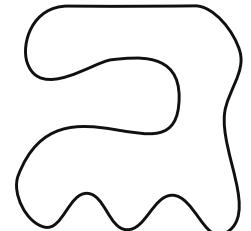
Additionally, the motor offers improved performance with superior speed control and torque. The integration of a driver circuit in the Nidec 24H simplifies system design and boosts performance and reliability, making it perfect for our compact, high-speed line follower, especially when considering the weight and agility factors of our system.

Gear set design

1. Desired Motor RPM

We aim to design a follower that travel the entire track within 10 seconds (0.167 minute)

- Track length: 8000mm
- Tire diameter: 65mm
- $Desired\ Motor\ RPM = \frac{TrackLength}{TireCircumference} \div Time\ (min)$



The track

- $Desired\ Motor\ RPM = \frac{8000mm}{65mm \times \pi} \div 0.167\ min = 234.6\ rpm$

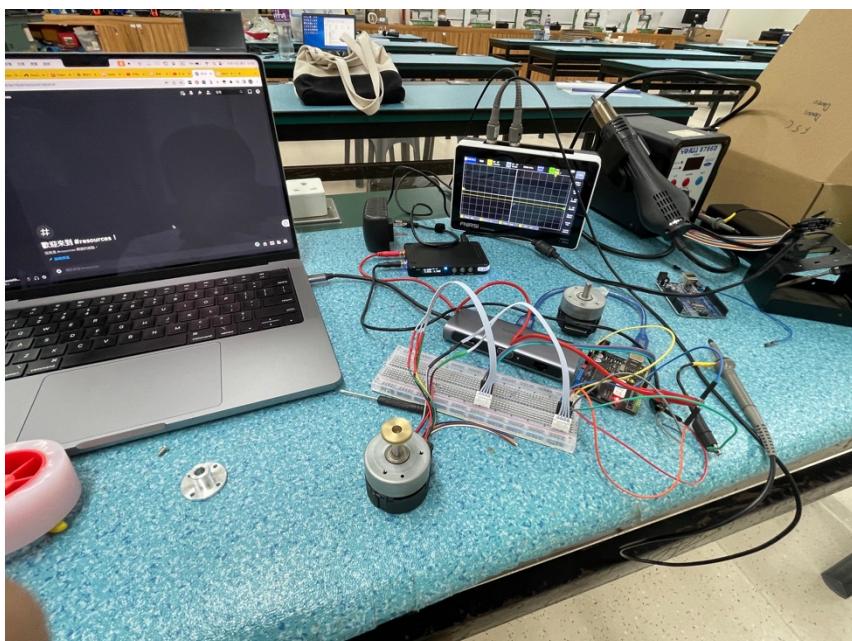
After selecting the ideal motor (Nidec 24H) and evaluating the desired motor speed, we discovered that the Nidec 24H operates at a relatively high-speed range (150 – 3900 rpm). However, it cannot provide sufficient torque during low-speed operations (150 – 500 rpm), a limitation that has been discussed in the “motor selection” section. Furthermore, its maximum speed of 3900 rpm is somewhat excessive for our follower's aim.

Therefore, a gear set is needed for:

- Speed Reduction
- Torque Enhancement

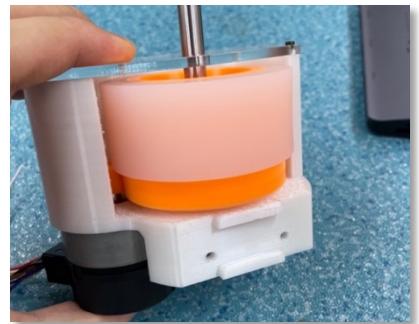
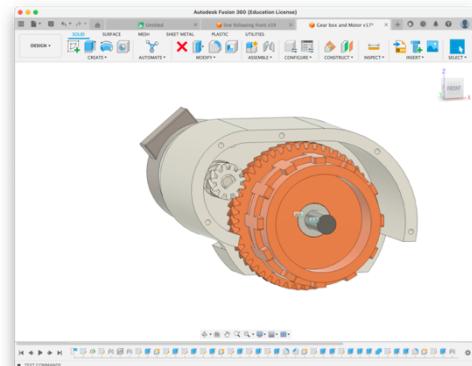
Nidec 24H motor operation and properties test:

[Testing Video \(Click Me!\)](#)



2. Gear ratio

We eventually designed a gearbox with a gear ratio of 4.55:1, featuring an output gear with 50 teeth and an input gear with 11 teeth.

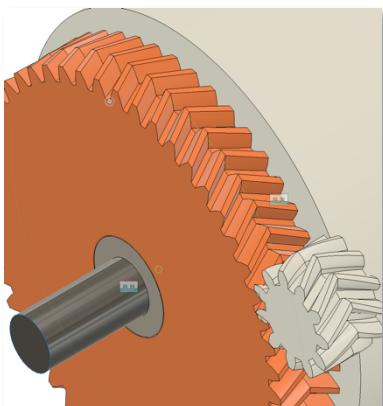


This gear ratio effectively converts the rotational speed of the Nidec 24H motor from 150 – 3900 rpm to an output rotational speed ranging from 33.7 – 876.4 rpm, which matches our desired rotational speed of 234.6 rpm.

Video of motor with gear box equipped: <https://photos.app.goo.gl/NPo4kNCyhGdWwo8V7>

3. Design Detail

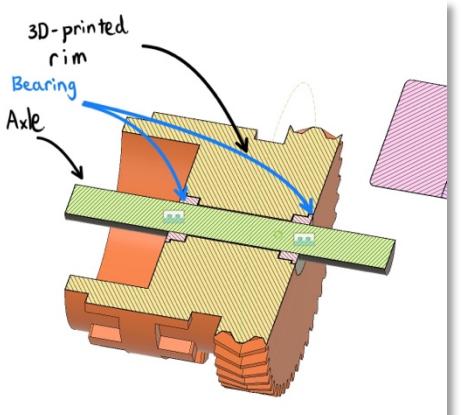
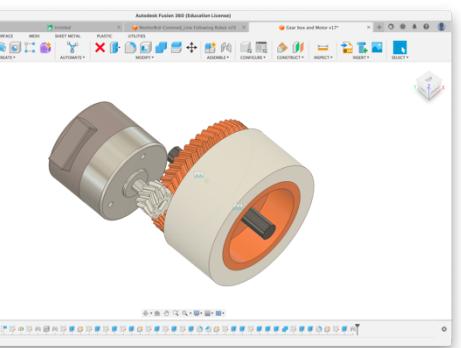
In order to align the input gear and output gear firmly, a pair of herringbone gear is used. Herringbone gears are different from regular spur gears because they have a unique V-shaped double helical design. This design allows for more teeth to be in contact at any given time, providing smoother operation and greater load capacity.



The V-shaped double helical design of a herringbone gear

Additionally, the mirrored helical teeth of herringbone gears effectively cancel out the lateral forces that typically occur in single helical gears. This feature eliminates the need for large thrust bearings, making herringbone gears ideal for high-torque applications in compact spaces, as they can maintain alignment and stability under heavy loads.

Moreover, to hold the wheel in place and align with the axle, two flanged ball bearings are mounted on each side of the 3D-printed rim.

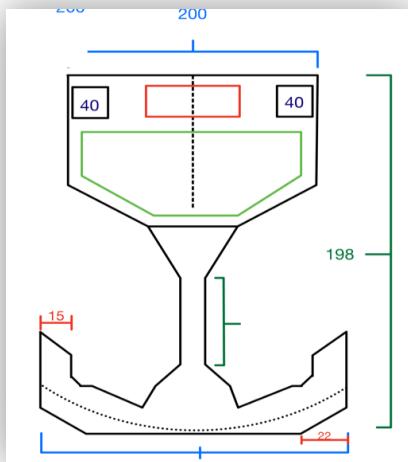


Follower Design

To create a Line Follower capable of efficiently tracking and following a line, we recognized the need to consider several key aspects of the robot's design namely:

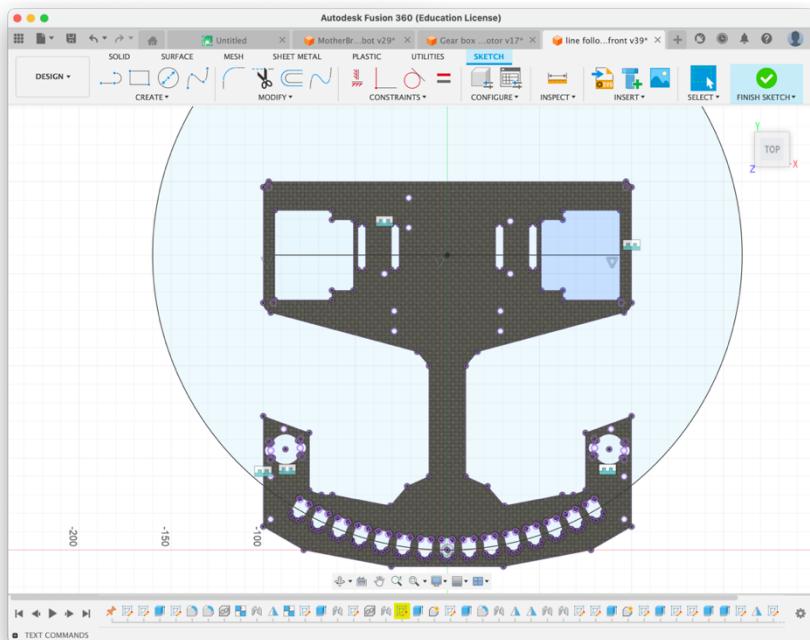
- Width of Robot
- Width of Sensor Array
- Gap Distance between Sensors
- Spacing from Sensors to Motors.

However, constrained by time limitations, we opted to adopt the parameters of [a Line Follower design](#) uploaded to GrabCAD by Andri Setiawan. We are grateful to Andri for this valuable resource. 🙏



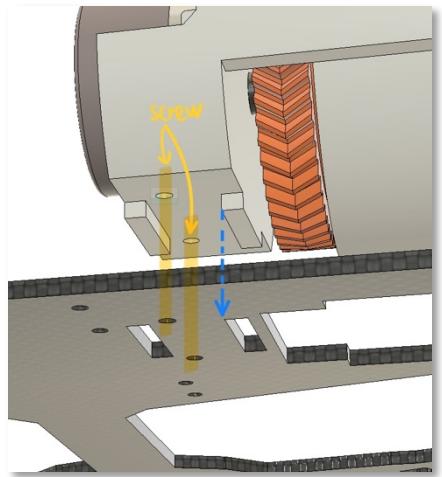
Before moving on to the design and creation of a 3D CAD model for the follower, we engaged in brainstorming and sketched a preliminary draft of the line follower.

We aimed to align the centre of the sensor array's curvature with the robot's rotational centre, which helps add a degree of linearity to the data captured by the sensors.

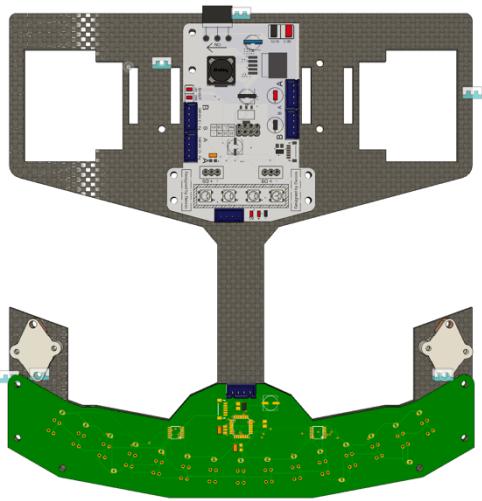


Alignment of the center of rotation and the center of curvature of the sensors

After designing the bottom frame, we implement the gear box into the design. We mount the gear box on the frame with a mortise and tenon joint and two screws.



A mortise and tenon joint and screw mount.



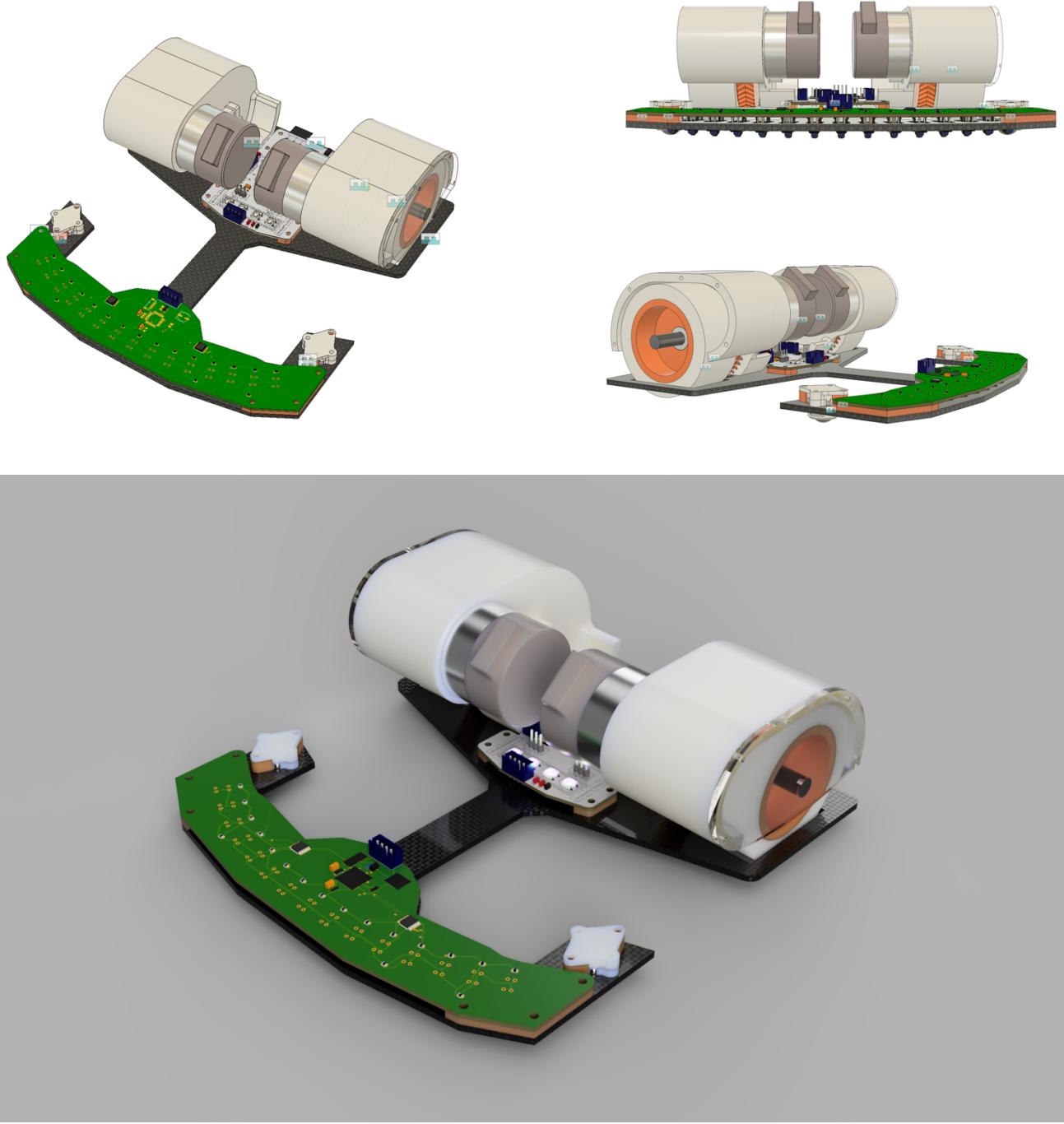
Mother Board (upper part) Sensor Board (bottom part)

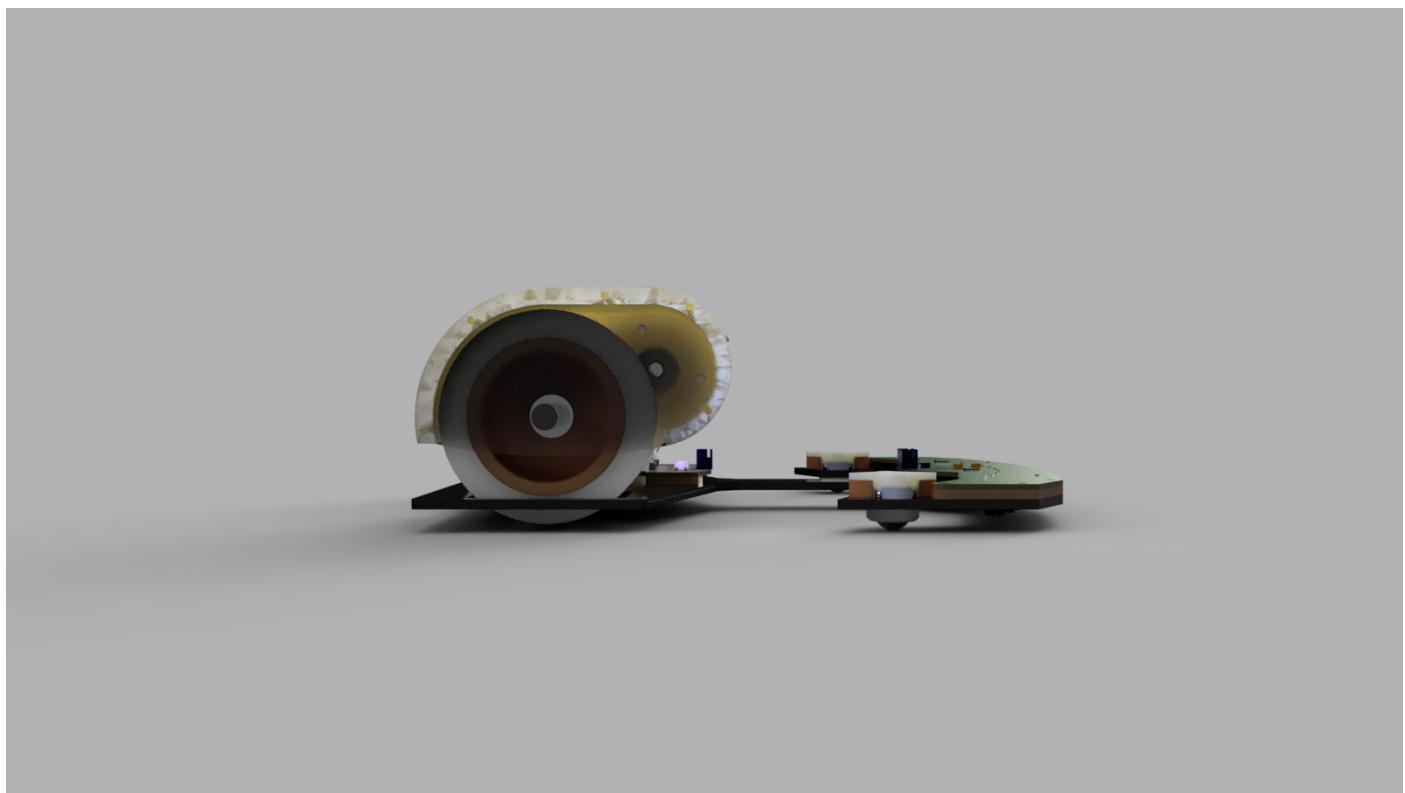
Afterwards, we implement two customized circuit boards, including one sensor array circuit board and one mother circuit board. The electronic designs of the boards will be discussed in the upcoming sections.

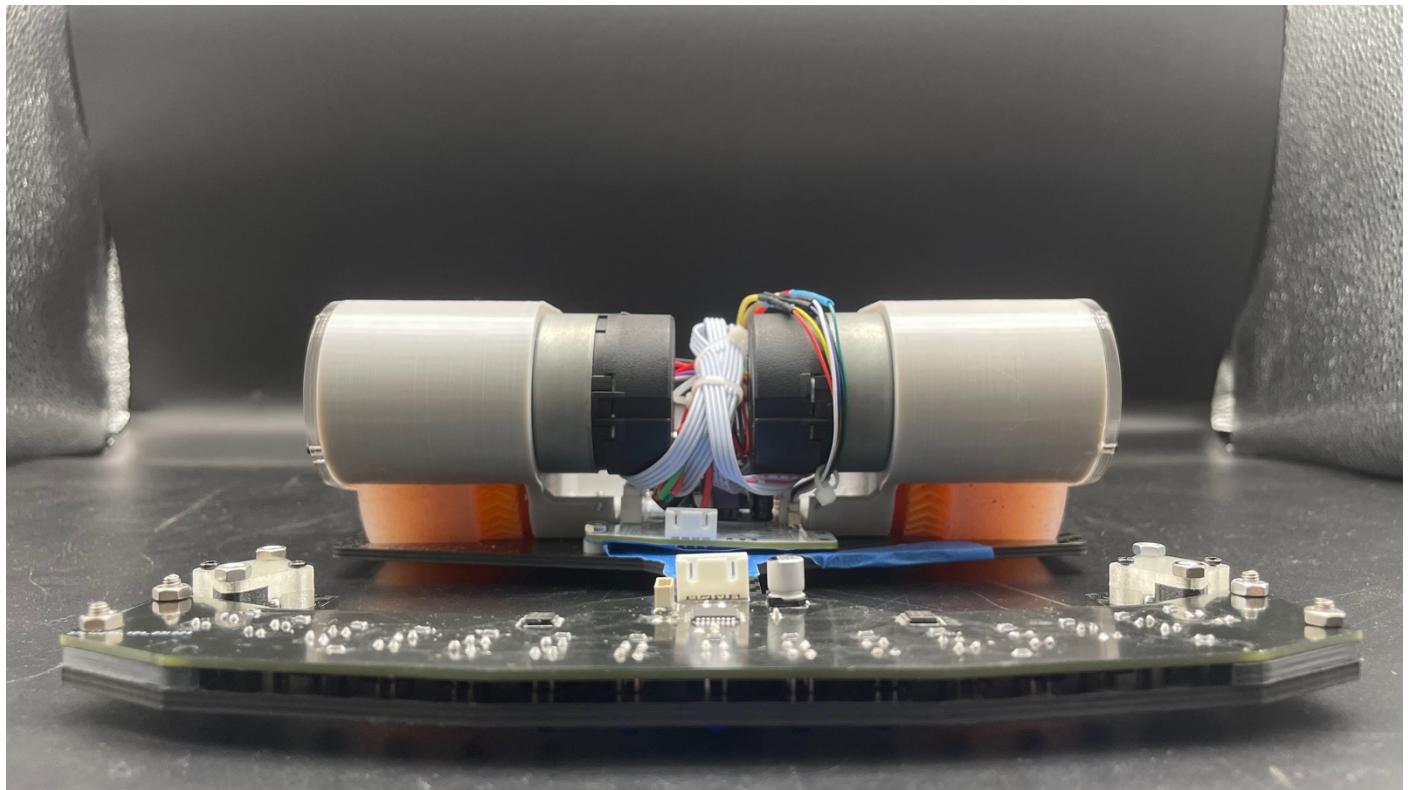
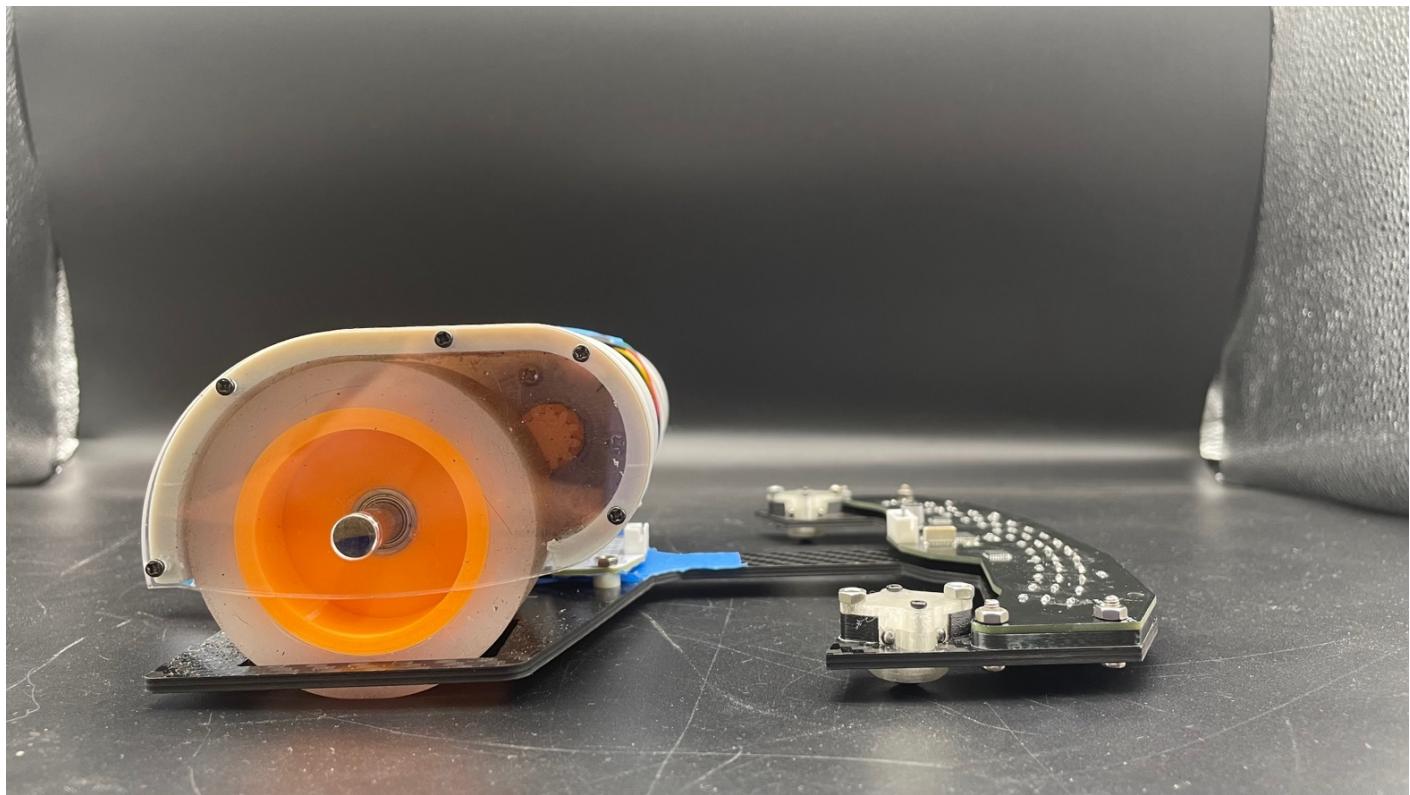
To ensure both stability and agility, especially when navigating sharp turns, our design incorporates a pair of silicone wheels. These wheels provide the follower with enhanced traction and flexibility, allowing for smooth movement and precise control even at higher speeds or during sudden directional changes.



After finalizing the design, including mounting the ball caster, designing spacers for the circuit boards, here is our final design:





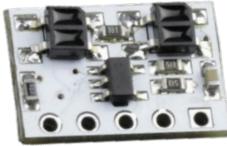


Electronics

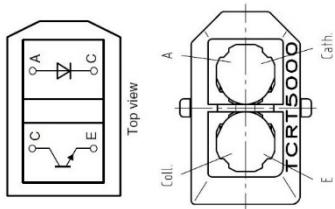
Selection of Sensor

Sensor Consideration:

- Output signal
- Weight
- Size (is it easy to integrate to our robot)
- Suitability of the competition arena

JSumo Mz80 Infrared Sensor	JSumo Dual Micro Line Sensor ML2
 (1)	 (2)
<p>Features:</p> <ul style="list-style-type: none">● Height: 4.5 cm● Voltage: 5V● Weight: 20g● Widely used in mini sumo robots and sumo robots● Range can be decreased or increased with trimpot head screw● 15 ° detection angle <p>Pros:</p> <ul style="list-style-type: none">▫ Digital output (easier to process)▫ Can be integrated into various robotic projects (sumo robot and line following robot)▫ Adjustable sensitivity <p>Cons:</p> <ul style="list-style-type: none">▫ Low resolution (comparing to analog signal)▫ Not convenient▫ Low accuracy (adjustable)	<p>Features:</p> <ul style="list-style-type: none">● Dimensions: 9.80 x 14.00 x 3.30 mm● Weight: 0.4g● Voltage: 3.3-5V● Integrated fabrication● Two separated IR sensor <p>Pros:</p> <ul style="list-style-type: none">▫ Tiny size of the sensor▫ Easy integration (integrate with various of device)▫ Consist of two individual sensors (processing more efficient and display of information) <p>Cons:</p> <ul style="list-style-type: none">▫ Limitation of sensor (difficult to track line efficiently in high speed)

Tcrt5000



Feature:

- Dimensions: 10.2 x 5.8 x 7
- Voltage: 5V
- Sensing range: 0.2to 15 mm
- Peak operating distance: 2.5mm
- Typical output current under test: IC = 1 mA

Pros:

- Presence of different physical object
- Clear observation of signal
- Good waveform
- Common

Cons:

- Easily affected by environment conditions (sunlight and lights from surrounding)

In contrast, other groups who use the official line tracking sensor module (fig b), will generates only a digital signal, we have chosen to implement the tcrt5000 directly on our PCB board.

This decision was made to enhance the accuracy of line detection by utilizing the analog signals produced when the sensor receives infrared signals.

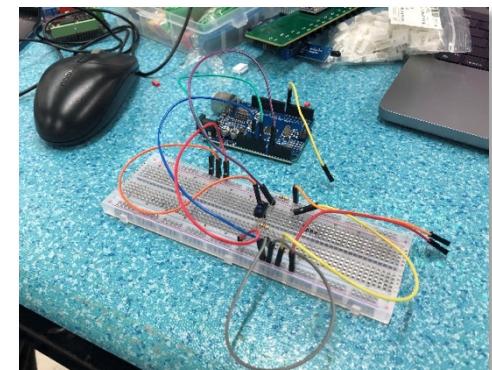
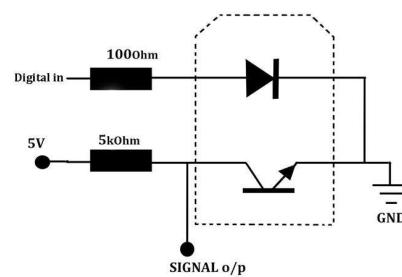
By collecting and analyzing different analog readings, it will benefit us from using the calibration command and employ a map function, which will be further discussed in subsequent sections.



fig b

Process of testing the sensitivity of the TCRT5000 Sensor

IR sensor: TCRT 5000
Circuit Diagram



To sum up, sensor (1) and (2) can only generate a digital signal. However, in this project we desire to make use of the benefit of analog signal. Furthermore, those sensors (1) are too large in size, so it restricts us from implementing multiple sensors. Regarding sensor (2), considering its performance while following line, it will be more suitable for us to select the tcrt5000 as our line following sensor.

Sensor board

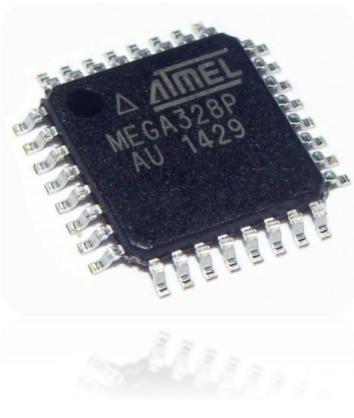
PCB design

MCU (Marvel Cinematic Universe)

In this project, we picked ATmega328P as our microcontroller unit, it can operate at a clock speed of up to 20 MHz and has 32KB of flash memory for program storage. By executing powerful instructions in a single clock cycle, the device achieves throughputs approaching one MIPS per MHz, balancing power consumption and processing speed.

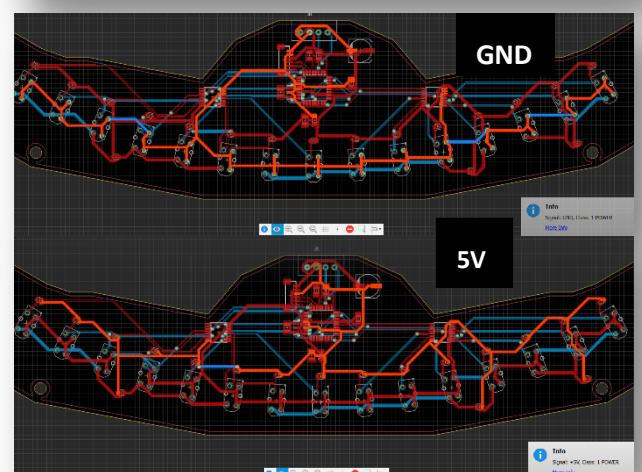
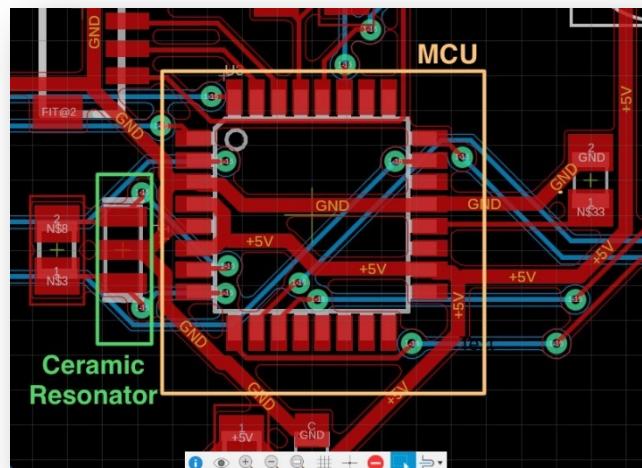
Reason for selecting this MCU are:

- Obtain 1024B EEPROM
- Easy integration with Arduino boards
- Abundance of libraries (since it supports various functionalities and many pre-written code, such as EasyTransfer)



Ceramic Resonator

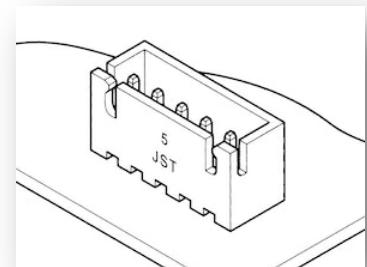
Since the ceramic resonator provides accurate timing of frequency reference, it can be used as the source of the clock signal for digital circuits such as microprocessors. As such, we placed it extremely close to the MCU.



PCB Layout

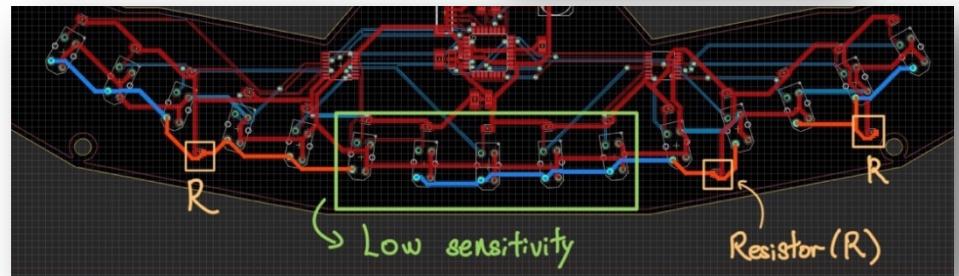
- ✓ The 5V and ground circuit are thickened due to the high consumption of current.

- ✓ A connector (through-hole type) is applied, for power connection and most importantly, communication between two boards.



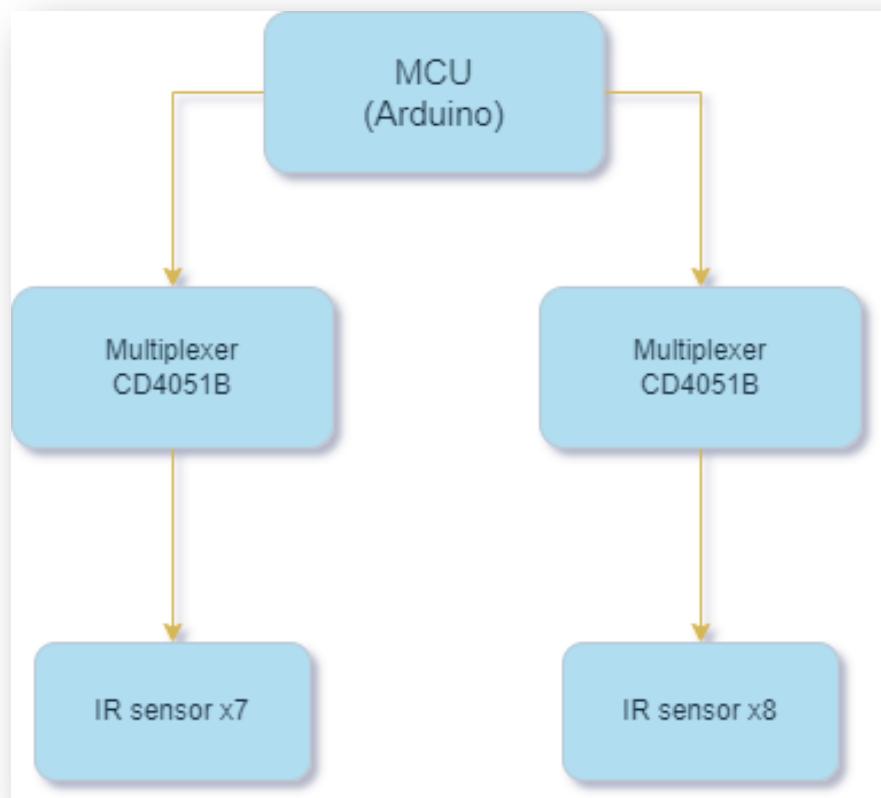
Reflection

Drawback: the circuit traces between the three 100Ω resistors and those IR sensors are forgotten to enlarge in thickness, causing the sensitivity of sensors (particularly near the middle area) relatively low. Therefore, the values of different sensors are not comparatively the same while detecting the line.



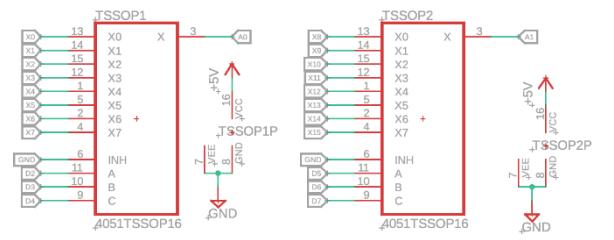
System

As the diagram shown below, the MCU (Microcontroller Unit) is sending data to two multiplexers, specifically the CD4051B, which are connected to seven IR sensors each.



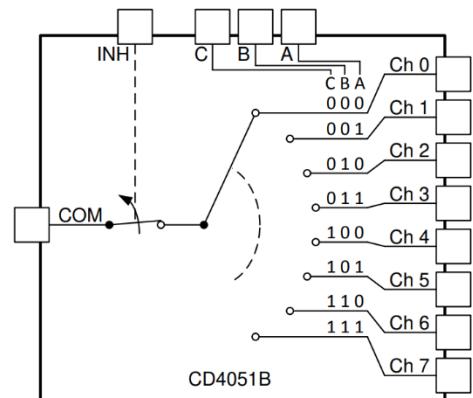
How the "Multiplexer" works

The CD4051B analog multiplexers and demultiplexers are digitally-controlled analog switches having low ON impedance and very low OFF leakage current. These multiplexer circuits dissipate extremely low quiescent power over the full VDD – VSS and VDD – VEE supply voltage ranges, independent of the logic state of the control signals.



This multiplexer can integrate circuit that operates as an 8 channel single switch, it allows one of eight analog input signals to be selected and routed to the output based on control signal (Arduino).

The CD4051B has three control pins (A,B,C), and by setting the appropriate combination, the desired input channel can be selected.

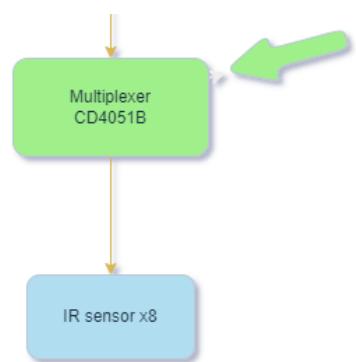


In this robot, there are 2 multiplexers as there are 15 IR sensors (fig a). By connecting 7 out of 15 IR sensor to one multiplexer and the rest to the other one, it will gain the ability to collect the exact number of value of sensors, which the MCU required, by switch different cases and returning the values. Furthermore, due to the limitation of analog pinouts on the Arduino board, which only obtain A0 – A5 in total. This restricts us from navigating various numbers of analog output signal. As a result, applying multiplexers are more suitable for us.

Data Processing

i. Min-Max Calibration Algorithm

In the beginning, all the sensor maximum and minimum value will be set to 0 and 1023 respectively. When the calibration starts it will read all the sensors one by one. Then, for each reading, it checks if the detected value is higher than the sensor maximum value. If so, it stores that new higher value as the sensor maximum value. On the other hand, if the detected value is lower than the sensor minimum value, it will



save it as the sensor minimum value. After the calibration ends, the most updated values will be the following maximum and minimum point.

- This will benefit us from applying different sensor values as a reference for line following under various kind of situation.

ii. Mapping the values

The map function in this case has convert the actual recorded sensor value into between 0-3. Additionally, if it is 1, that means the following sensor has started to detect a line.

- This lets the code normalize all the sensor readings to a common scale, so they can be compared together easier. It translates the raw readings to a format the rest of the code can work with.

iii. Error calculation

The code first takes readings from all the sensors and assigns them values between 0-3 to get an "average" value. It then calculates the real average by multiplying each sensor value by its number and adding them up, giving more weight to higher sensors. It also keeps track of the total sum of all sensor values. If the sensors are sensing something, it uses the average divided by the sum as the "drift" value. The drift value indicates the location of the average sensor reading, similar to a pointer on a scale. The code aims for the drift to be close to 7, so it calculates the "error" value by measuring how far the drift is from 7. A higher error means the drift is further off from the desired location, while an error of 0 means it is perfect.

- Straightforwardly, this algorithm can be used to adjust the system over time (while following line), and it will be more desired to remain at the center (error = 0 means perfect!).

iv. EEPROM

The EEPROM is more likely a little memory chip that can store data, even when the power is turned off. The code is used to save the minimum and maximum sensor values that are detected during calibration. So, when the robot starts up it can use the old data (sensor maximum and minimum) stored from the last calibration right away.

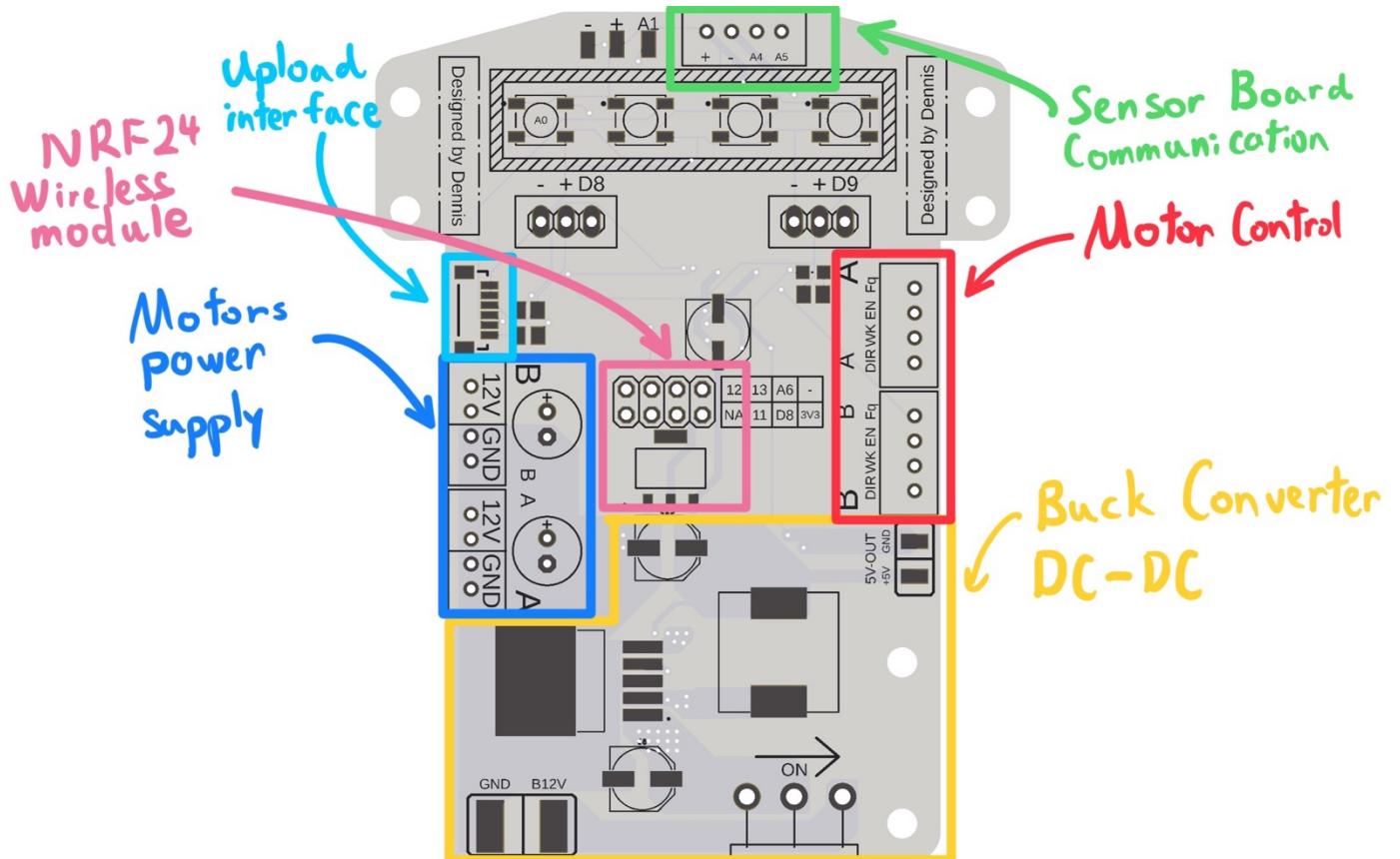
- In summary, the EEPROM allows to remember the sensors values tested last time. This means we do not have to calibrate every time.

Mother Board

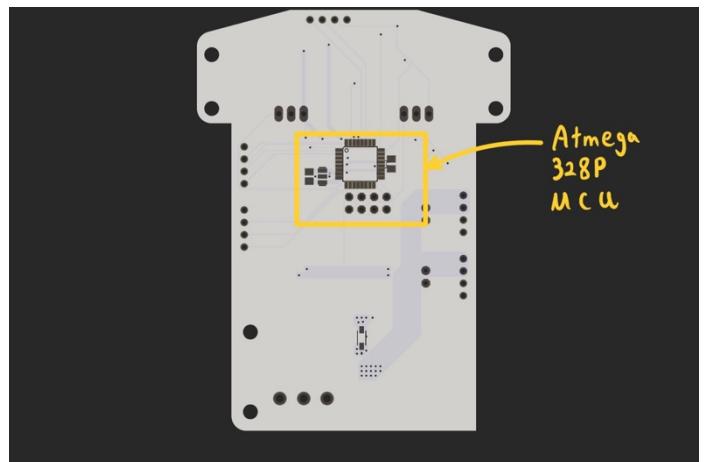
In this project, we have designed and implemented a customized mother board as the robot's brain. Here are the tasks that the mother board have to handle:

- Communication with the sensor board using
 - I2C
- Communication with the wireless console using
 - NRF24
- Motor Controlling
 - Controlling two Nidec Motors
- Power Voltage Step-Down with
 - LM2596 Buck DC-to-DC converter

The diagram provided illustrates the specific pinouts assigned for these functionalities on the motherboard.



To drive these features, we utilize an ATmega328P microcontroller unit (MCU) due to its ease of programming and the extensive range of available libraries that enhance its usability.



PCB Design and Layout

Maintaining a stable power supply within the 1.8 to 5.5 volts operating range is critical for the reliable functioning of the ATmega328P microcontroller. Power fluctuations can cause erratic behavior, so it's vital to keep the voltage steady to ensure the microcontroller operates correctly and avoid system issues.

With the plan to use a 10v – 12v three-cell battery for the robot, employing a LM2596 buck converter to downscale the voltage to 5V is necessary.



LM2596



AMS1117-3.3 V

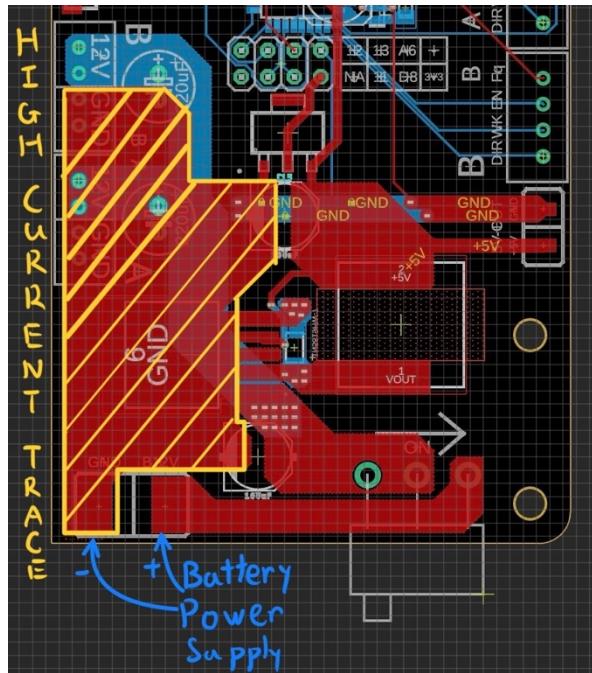
Moreover, the NRF24 wireless radio module runs at a voltage of 3.3V, therefore, an additional 5V to 3V converter is needed. In this case, we decided to implement a small package LDO, AMS1117-3.3 V, to supply a 3.3V power to the NRF24 module.



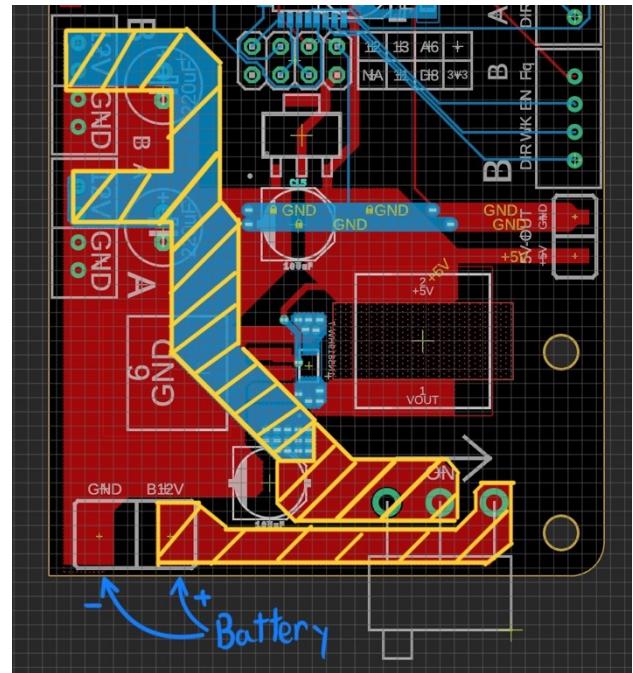
NRF24 Module

Additionally, since the motors draw a large amount of current, designing the circuit with optimized and well-designed high current traces is crucial for delivering robust and stable power to the motors.

The images below show the high current trace designed for the motor ground and the positive.



High current trace from battery ground to motor ground.



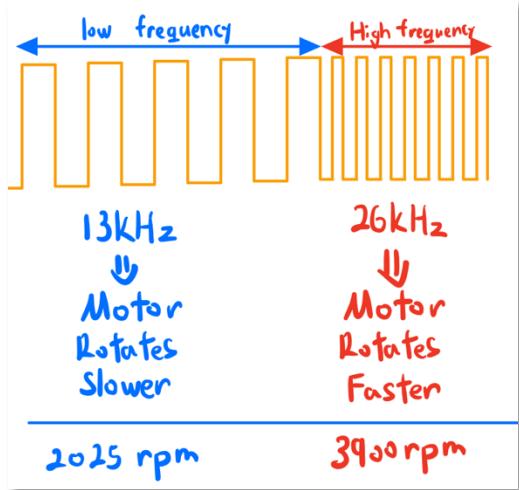
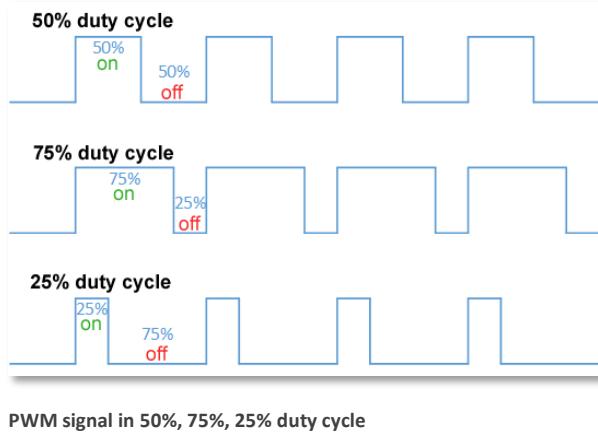
High current trace from battery positive to motor positive.

Motor Control

This section provided a deeper understanding of how the “Nidec 24H motor” is controlled.

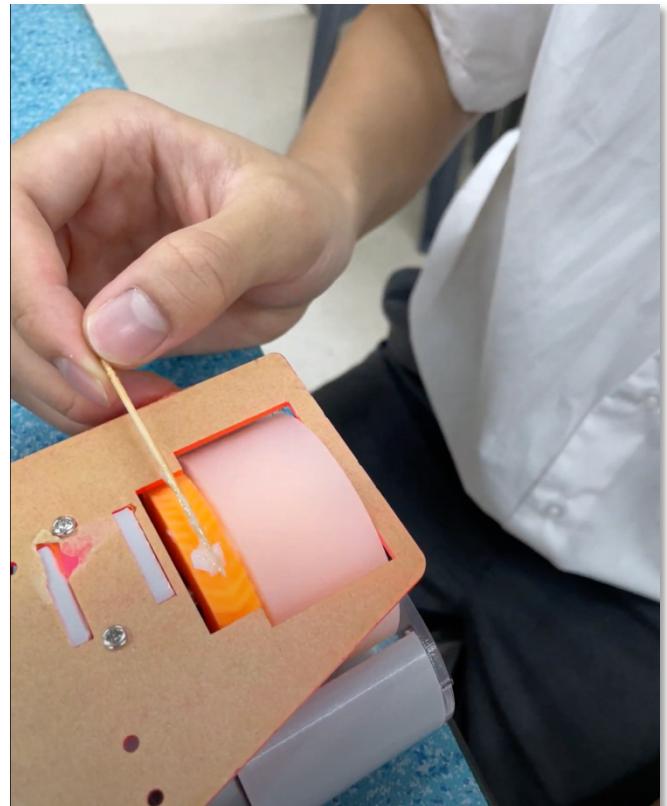
Unlike most commonly found brushed/brushless motors with ESD, FOC and H-Bridge driver, the Nidec 24H motor is controlled under the frequency (instead of PWM) of the signal pin.

The input frequency range of the Nidec motor is 1kHz to 26kHz, which controls the motor spinning from 150 to 3900 rpm. Using this control method, the rotational speed of the motor could be accurately controlled.



Prototype

After designing our line follower, we tested and builded a prototype of our robot.



Our very first prototype

[Click me!](#)



Software Development

After prototyping the follower, we begin to program and develop the firmware of our robot, including the program for the mother board and the sensor board.

Integration

The entire line follower system consists of three major hardware, a mother board, a sensor board and a remote console.

Here is how they collaborate with each other:

The sensor board gathers data from 15 sensors, calculates an “error” value (measure of misalignment of the robot and the line), and transmits it to the motherboard, which uses a PID controller to steer the car, while calibration data is stored in EEPROM; additionally, a remote console with NRF24 allows for wireless control and calibration of the car.

1. Sensor Board

- Get the sensor value from all 15 sensors
 - o Calculate an “error” value from those sensors
 - o Transmit the calculated “error” value to the mother board
- Calibration
 - o Using a min-max calibration algorithm to calibrate each sensor independently
 - o Store those calibrated parameters (min and max value) into EEPROM

2. Mother Board

- Receive the “error” value from sensor board
- Use a PID controller to control the movement of the car according to the “error” value
 - o In simpler words, track and follow the line
- Rotate the robot during calibration so that sensor board could capture more data to eliminate random error

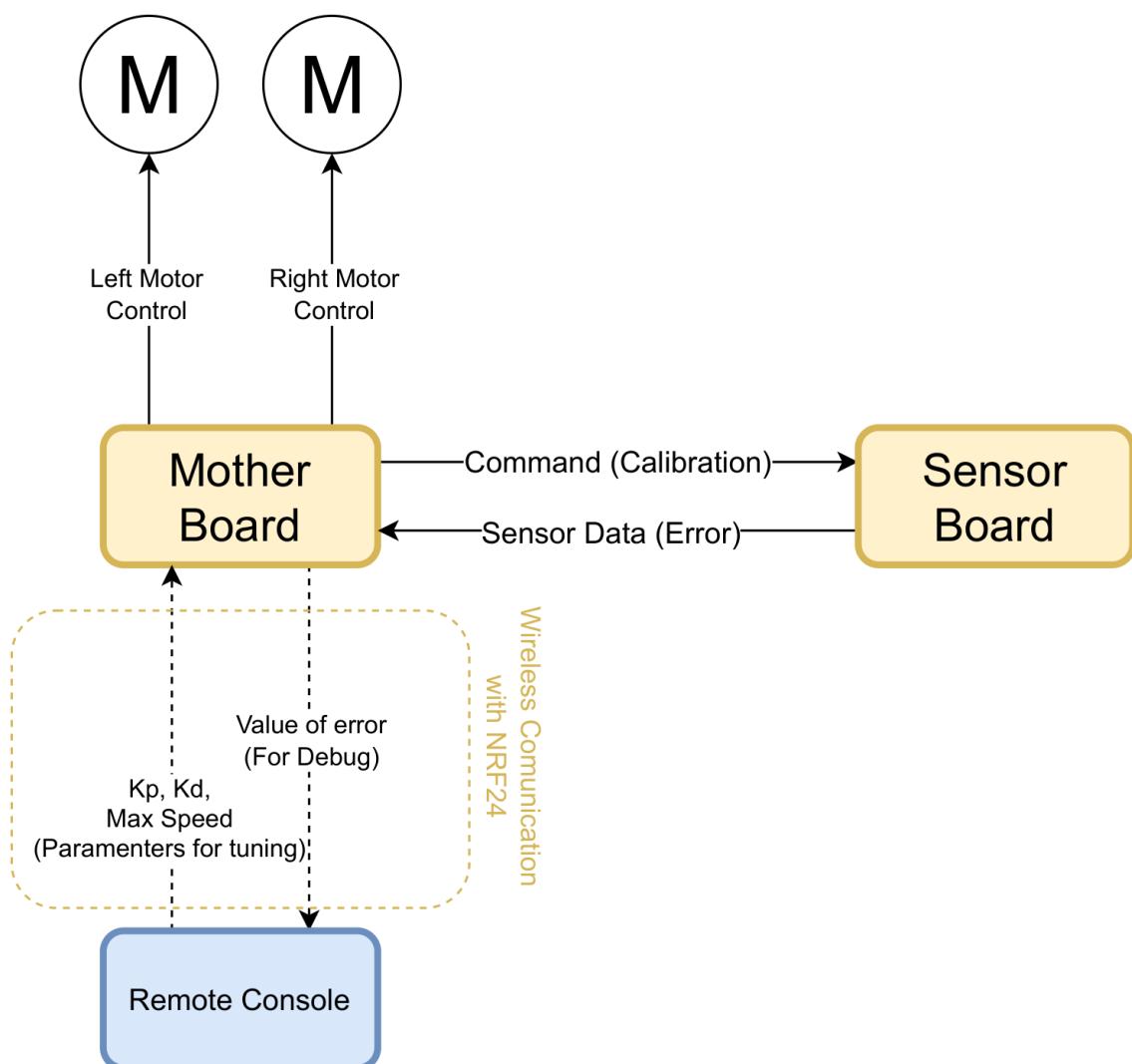
3. Remote Console

This is a console equipped with NRF24 (a wireless module) to communicate with the mother board wirelessly. ([Console-MotherBoard communication test: Click ME!](#))

- User can send command to the follower via the remove console to

- o [Drive the car wirelessly \(Video: Click ME!\)](#)
- o Tune the PID controller wirelessly
- o Calibrate the sensor whenever needed

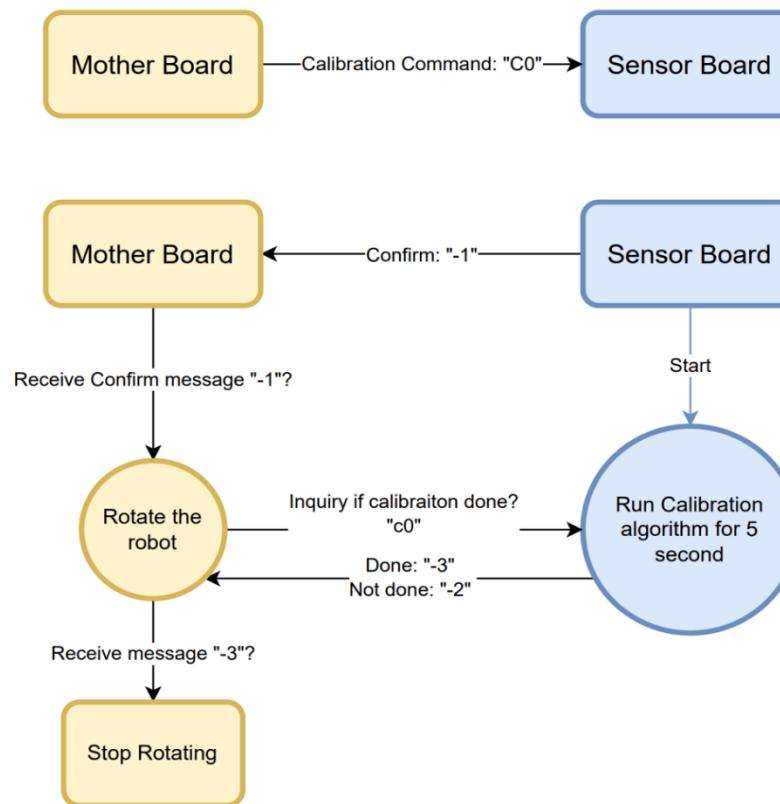
The flowchart below demonstrates the control system of our line follower: The Sensor Board detects the line and sends error data for calibration to the Mother Board, which controls the motors. The Remote Console wirelessly adjusts PID settings and commands calibration.



Communication between sensor board and mother board:

Calibration

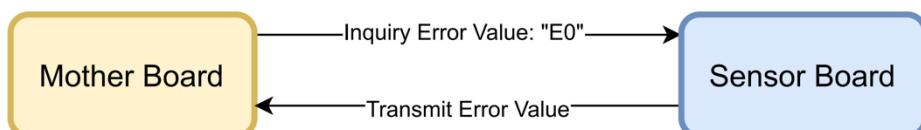
The calibration process for our line follower begins with the Mother Board sending a "C0" command to the Sensor Board. Upon receiving confirmation ("-1"), the robot starts rotating and the Sensor Board runs the calibration algorithm for five seconds. If calibration is complete, indicated by a "-3" message, the robot stops rotating.



Calibration sequence for a line-following robot:

Error Value Transmit

The Mother Board requests an error value with "E0" and the Sensor Board responds by transmitting that value back.



Communication protocol between Mother Board and Sensor Board for error value retrieval.

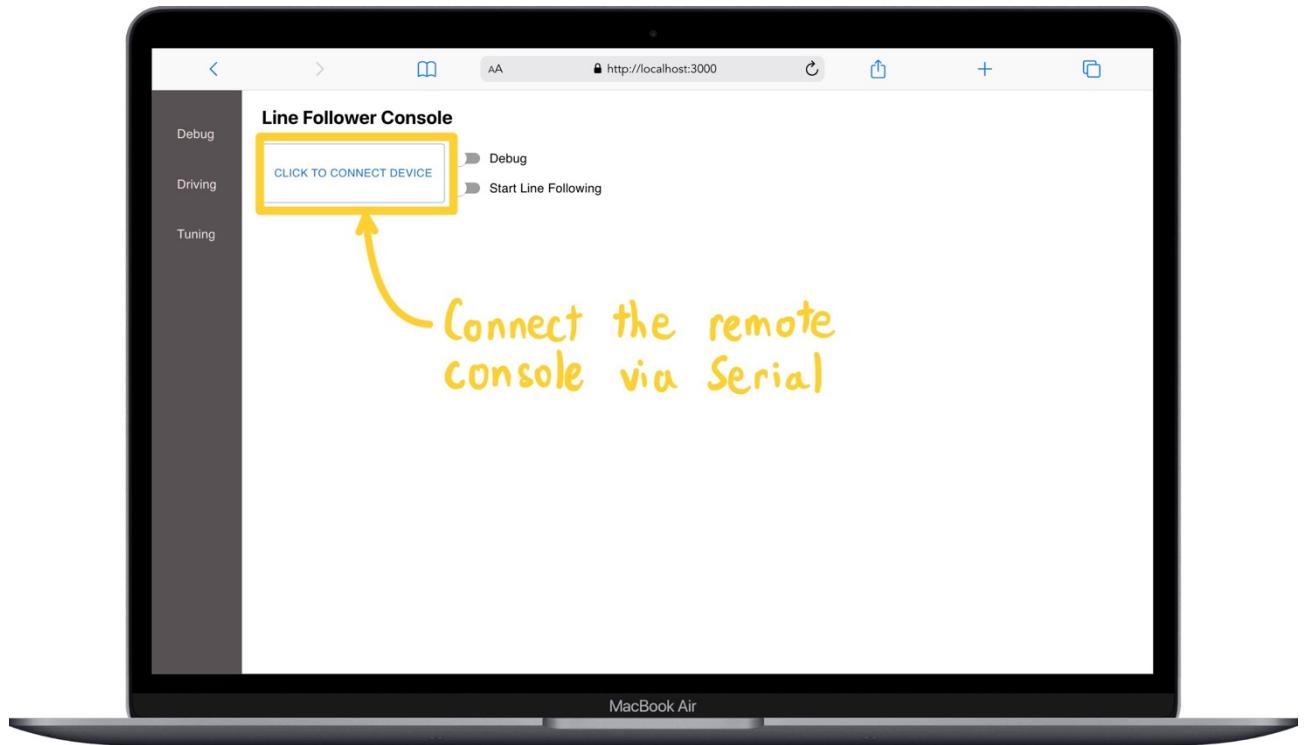
User Interface

To enable user interaction with the line follower, we have developed a web-based user interface that simplifies the control of the robot for any user. Utilizing the ReactJS framework along with the Material-UI library, a responsive and intuitive interface is designed allowing for the navigation and tuning of the robot.

Here are the features:

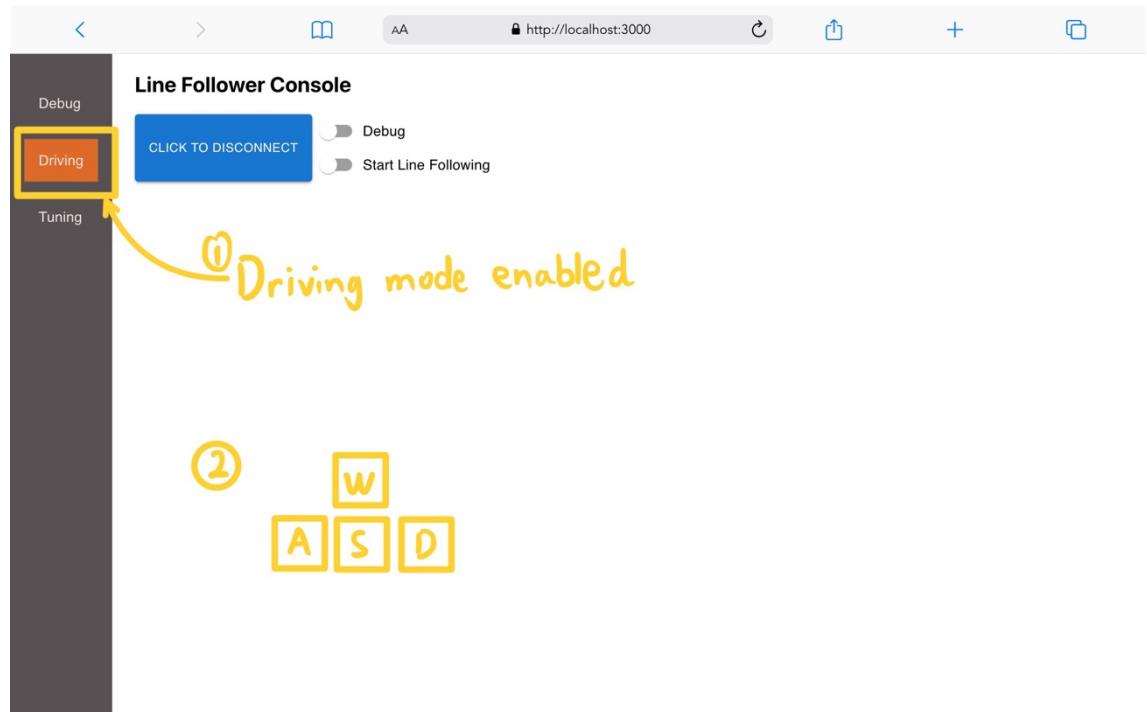
1. Connection

- Press “Click to connect device” button to connect the hardware remote console via Serial.



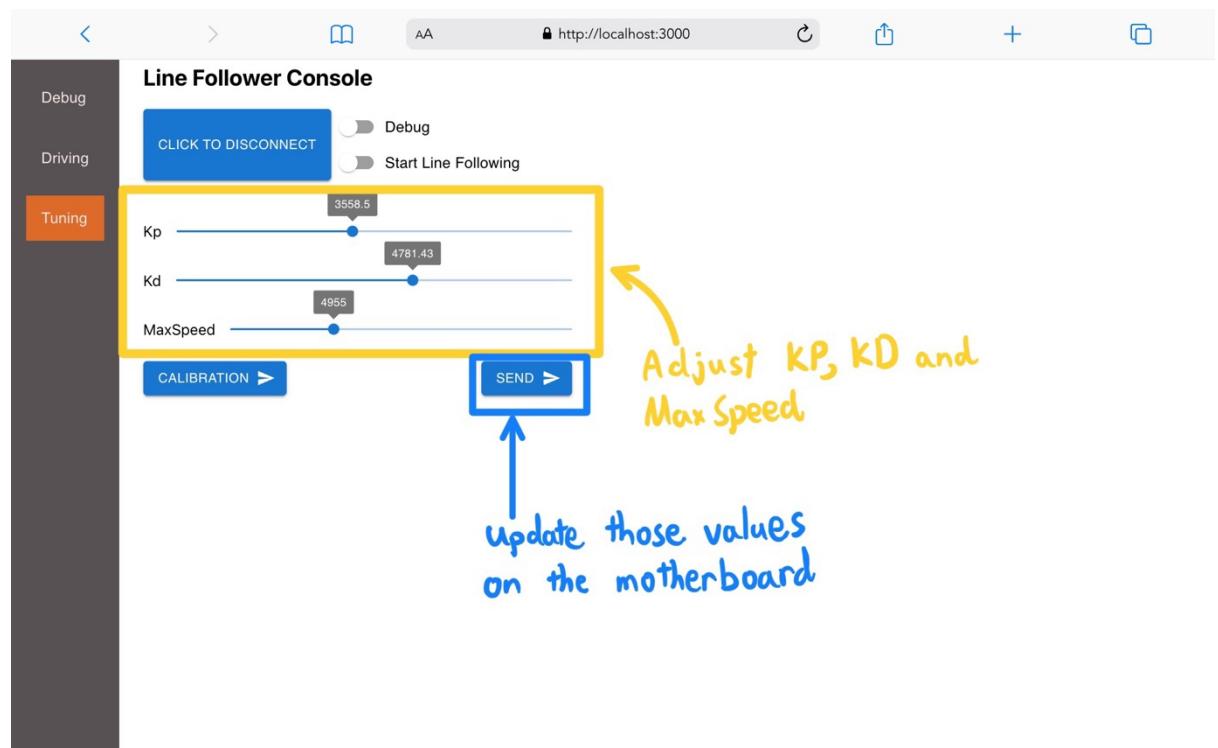
2. Driving

- Press the “Driving” button on the navigator to drive the robot with ‘W’, ‘A’, ‘S’, ‘D’



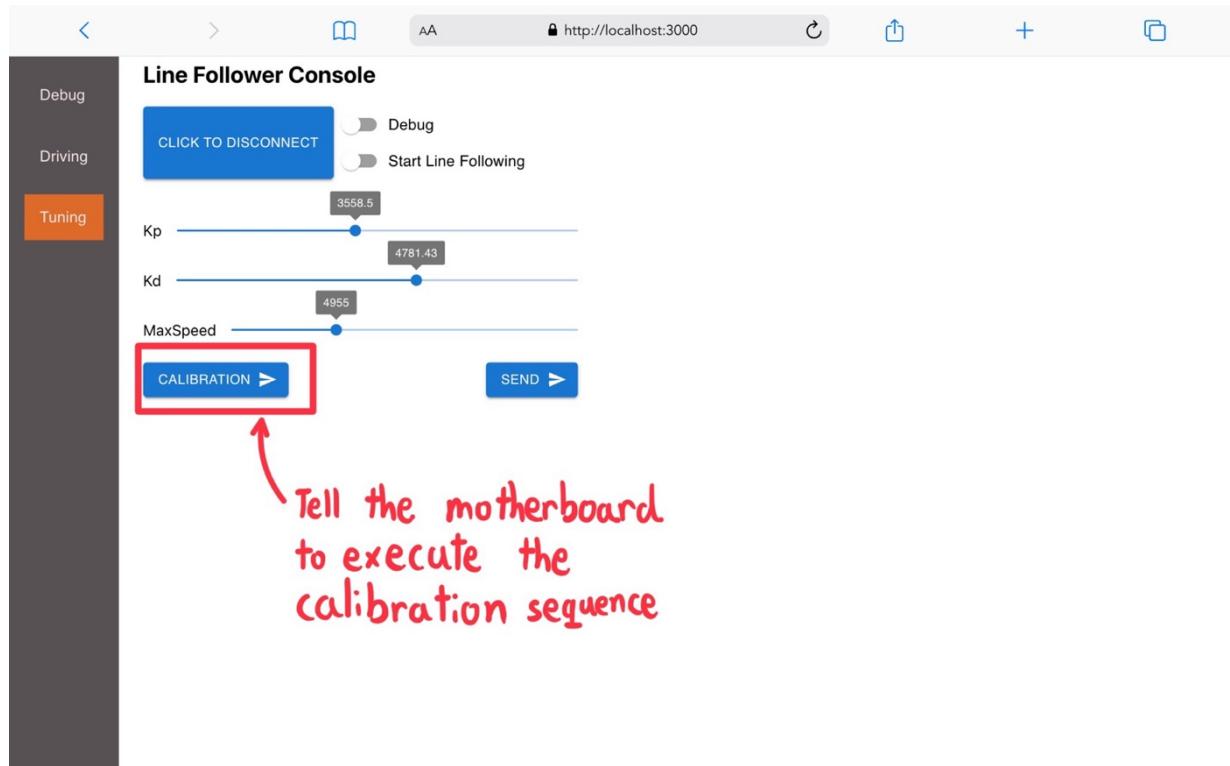
3. Tuning

- By selecting the "Tuning" option, user could modify the Kp, Kd, and MaxSpeed parameters using three interactive sliders. Afterwards, clicking the “Send” button updates those values on the follower.



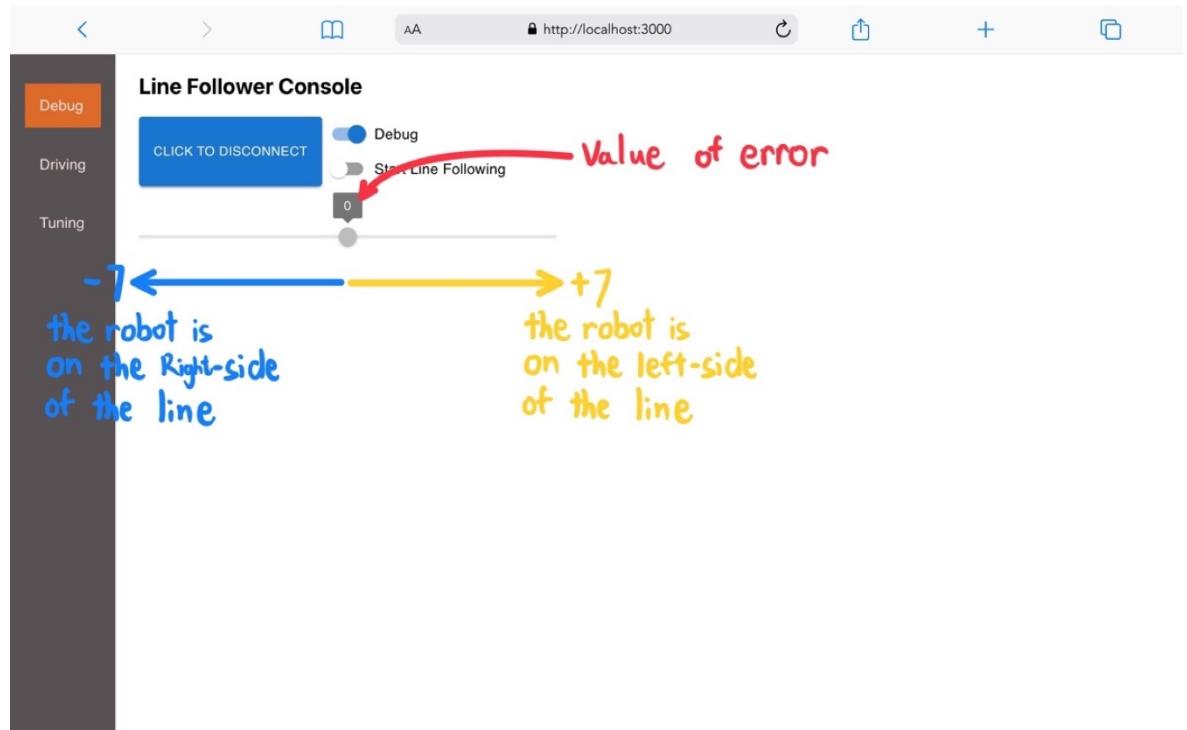
4. Calibration

- User could send a calibration command to the robot to start a calibration sequence of sensors.



5. Debug

- By toggling the "Debug" switch, a bar is shown to present the current value of "error" (measure of misalignment) of the follower



Result

After finalizing the hardware components of the follower, and days and nights of debugging and testing, we eventually approach our goal of building a line following robot that can follow the track for one cycle in 10 seconds.

- **Time used to travel one cycle: under 10 seconds**

✓



[Video of our robot that travels one cycle in 10 seconds. Click ME!](#)

- **Weight: under 2.5kg**

✓

- Measured weight: 2.32 kg

- **Dimension: within 200*200*150 mm**

✓

- Dimension of our robot: 200*198*138mm

- **Maximum speed: over 2 m/s (7.2 km/h)**

✓

- Our robot can travel in a fastest speed of 3 m/s (10.8 km/h)

- **No. of Infrared sensor: 15**

✓

- The sensor board of our robot is equipped with 15 sensors

- **Power supply: 11.1V lithium-ion battery (3 cell Li-po Battery)**

✓

- **Develop an easy way for monitoring, PID tuning, and debug**

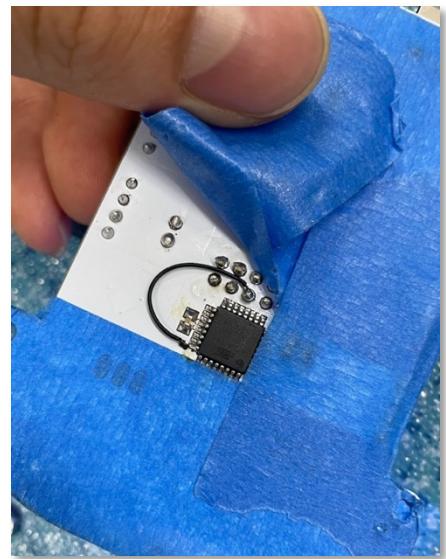
✓

- Using a wireless console incorporated with a web-based UI, we provide a humane interface for debug and tuning

Challenges

1. Wrong wiring of NRF24 on mother board

We have made a mistake in wiring the CSS (chip select) pin of NRF24. Due to a limitation of time, we eventually fix this mistake with a small and thin wire.



2. Motor Speed Control

To control the speed of a motor, we need to tune the frequency of a signal pin ranging from 1kHz to 26kHz.

One quick and simple way to change the output frequency of a pin is to use the built-in “tone” function provided by the Arduino officials. However, this “tone” function could only control the frequency of one pin at a time, which means the robot could not control two motors simultaneously.

To solve this problem, we have struggled with dealing with the hardware and software timers of Arduino but failed. Eventually, we found a third-party [“Tone” library](#) (That's right, it has the same name as the Arduino built-in function “tone”) that could control the output frequencies of two pin simultaneously. (Wish I could find it earlier =_=)

Future Development

1. In future iterations, we could adopt a more scientific method such as the Root Mean Square (R.M.S.) error to more accurately gauge the robot's line-tracking precision.
2. The 3-cell battery currently lacks a secure mount, often falling out during tests; we could design a reliable battery holder in the future.
3. The silicone wheels are great for grip but attract dust, reducing their effectiveness. We could look into creating wheels that resist dust or add a feature to clean the wheels.

Conclusion

In summary, this project has reached a successful conclusion, having meticulously navigated the intricacies of design, electronics, and software development to produce a line-following robot that not only adheres to competition rules but also excels in performance. Through integrating sensors and motors into a durable design, our robot completed the track within the target time and met the competition size and weight criteria. This work has been a valuable learning experience, and we hope it will be useful for future robotics projects.

References

Pulse-width modulation – Wikipedia:

https://en.wikipedia.org/wiki/Pulse-width_modulation

TCRT5000 datasheet

<https://www.vishay.com/docs/83760/t crt5000.pdf>

CMC_24H series | NIDEC CORPORATIO

<https://www.nidec.com/en/product/search/category/B101/M102/S100/NCJ-24H-24-01/>

CD405xB CMOS Single 8-Channel Analog Multiplexer or Demultiplexer With Logic-Level Conversion

<https://www.ti.com/lit/ds/symlink/cd4051b.pdf>

ATmega328P datasheet

https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf

LM2596 SIMPLE SWITCHER® Power Converter 150-kHz 3-A Step-Down Voltage Regulator

<https://www.ti.com/product/LM2596>

RoboMaster M2006 P36 Brushless DC Gear Motor

<https://www.robomaster.com/en-US/products/components/detail/1277>

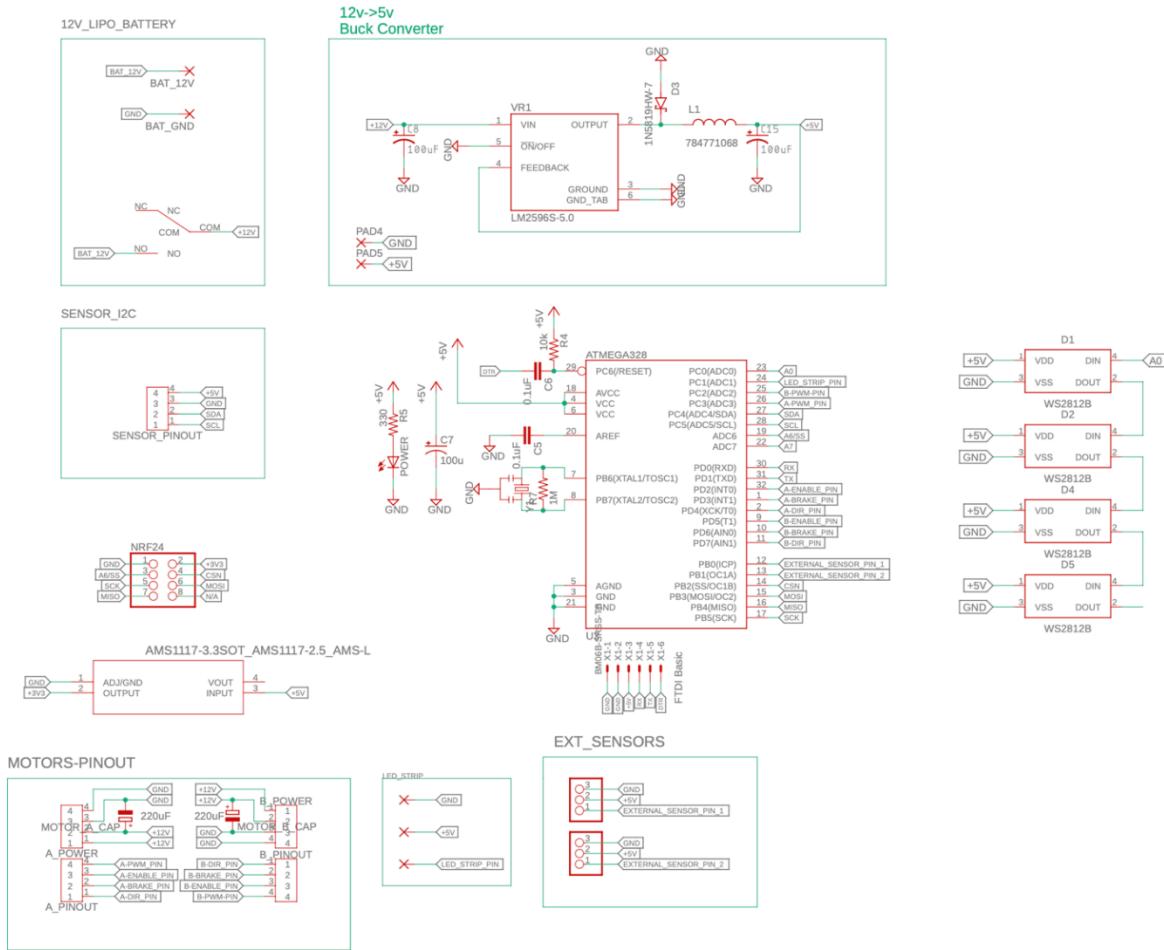
nRF24L01 – How It Works, Arduino Interface, Circuits, Codes

<https://howtomechatronics.com/tutorials/arduino/arduino-wireless-communication-nrf24l01-tutorial/>

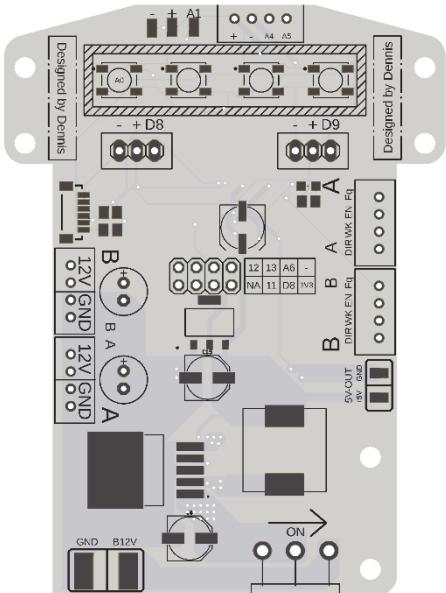
Understanding the I2C Bus

https://www.google.com%252Fhttps://www.ti.com/lit/an/slva704/slva704.pdf?ts=1700970709714&ref_url=https%253A%252F%252Fwww.google.com%252F

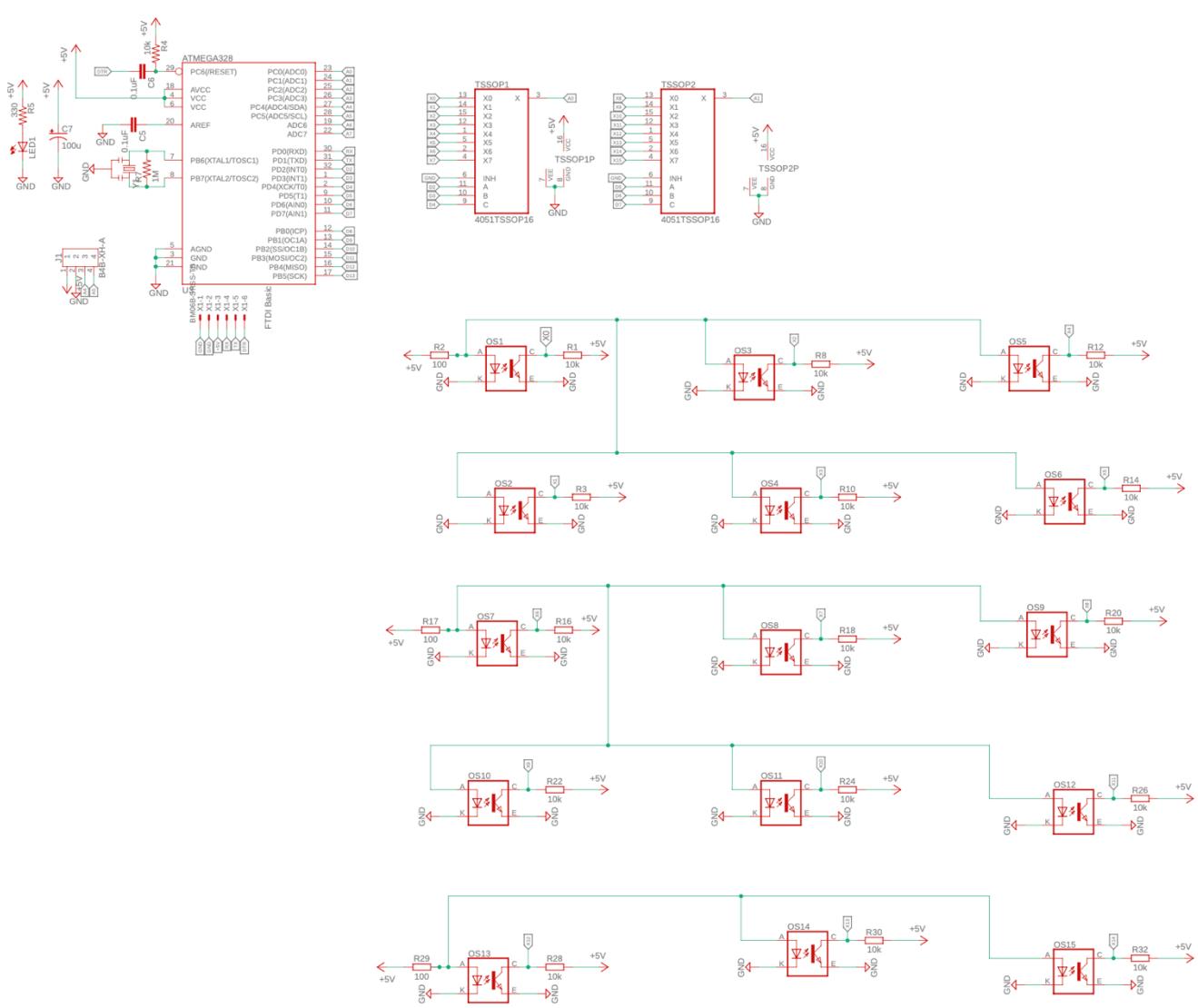
Appendices



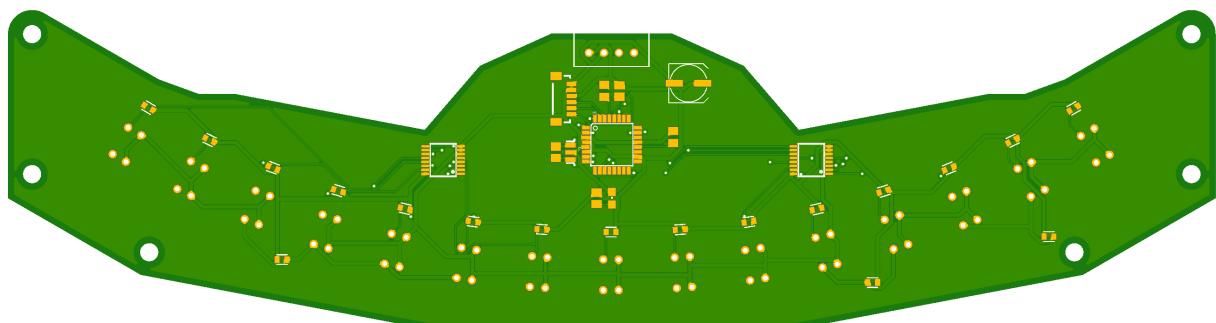
Mother Board Schematic



Mother Board PCB



Sensor Board Schematic



Sensor Board PCB

Our entire project:

<https://github.com/dennischang928/Line-Follower>