NTNU
Norwegian University of
Science and Technology

Department of ICT and Science

AIS1104 - Automation and mechatronics with project

Concept Prometheus

Author:
Oscar Andreas Friksen Andersen

Abstract

This report presents the design and development of a concept for a line-following robot, detailing the components chosen, CAD model, wiring diagram and functionality of the code. Discussion on strengths and weaknesses of the concept is provided, followed by a conclusion.

05.12.24

Table of Contents

1 Introduction

This project focuses on developing a concept for a line-following robot capable of autonomously navigating a predefined path. The primary objective is to create a simple and lightweight robot that optimally balances accuracy, speed, and stability while tracking the line. Additionally, a language model (LLM) was utilized to assist in writing LaTeX code and some functions for the robot.

2 Concept

2.1 Description

The line-following robot concept integrates Simultaneous Localization and Mapping (SLAM) principles. Using an IR sensor array to detect the line and an odometry sensor to map the track, the robot optimizes its route for subsequent runs by effectively 'cutting corners' and reducing unnecessary travel.

2.2 Key Motivations

- **Performance:** By mapping the track beforehand, the robot can use algorithms to change the mapped track to find the fastest route, prioritizing speed without merely increasing velocity.
- **Learning SLAM:** This project also serves as an opportunity to explore SLAM, which is also something that I want to learn more about in general for future projects.

3 Component Selection and Justification

For this project, the following components were chosen based on this track:

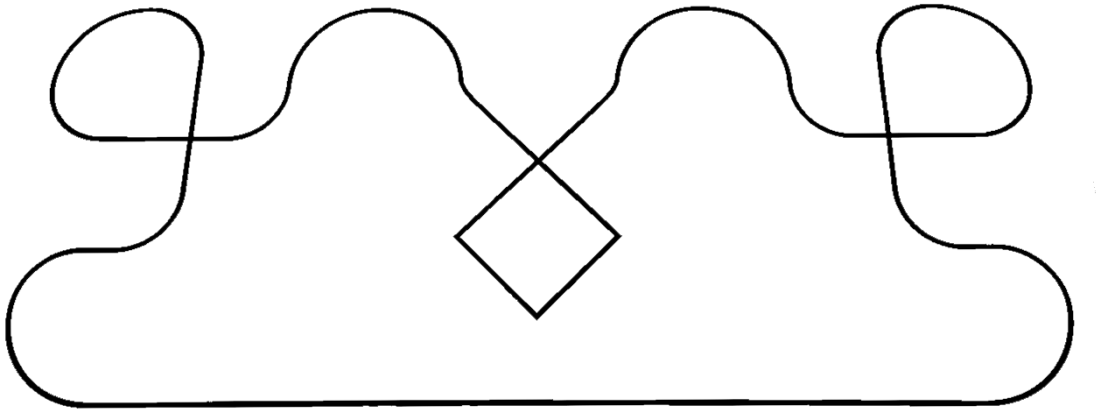


Figure 1: Sketch of the competition track

3.1 Microcontroller

- **Arduino Nano 33 IoT:** For the previous line follower, a standard Arduino provided sufficient computing power. However, adding position logging introduced significant limitations, as it could only store approximately 200 points. Given that the track is calculated to be around 10 meters long, the result is a maximum map resolution of 5 cm per point, which could be too small for precise navigation. To address these challenges, I chose a microcontroller that the school already has access to, which offers the following advantages:

  1. **Increased Memory and Clock Speed**
     - This microcontroller can store 100 times more points, increasing map resolution.
     - It also allows for more computationally heavier code and algorithms for more advanced navigation and mapping tasks.
  2. **Integrated Wireless Communication**
     - Built-in Wi-Fi and Bluetooth capabilities make it easier to adjust parameters wirelessly during testing.
  3. **Compact Form Factor**
     - Its form factor matches that of a standard Arduino Nano, leaving more space for other components.

3.2 Sensors

- **IR array QTR-MD-08RC:** In the first iteration of the line follower, an IR array with six sensors provided good accuracy and stability. However, small spikes were observed when transitioning from one sensor to the next, causing slight disruptions. The updated array, featuring two additional sensors, is expected to minimize these issues by offering finer resolution for detecting the line. A digital version was selected over the analog alternative due to its significantly lower signal processing time.

- **Optical Tracking Odometry Sensor PAA5160E1:** To address the issue of slippage when using wheel encoders for positional tracking, I explored alternative methods to track ground movement without direct contact. One promising solution was, quite literally, in the palm of my hand. By repurposing a mouse sensor (optical flow sensor), I was able to measure movement in both axes effectively. Test code is provided in the zip file.

```
if (mouse1.available()) {
    deltaX = static_cast<float>(mouse1.getMouseX());
    deltaY = static_cast<float>(mouse1.getMouseY());
    mouse1.mouseDataClear();
} else {
    deltaX = 0;
    deltaY = 0;
}

x += deltaX;
y += deltaY;
```

However, accurate tracking failed if the sensor rotated, as this caused the movement data to flip. To resolve this, I used an IMU to measure the Z-axis angle, or heading. By combining the change in movement with a slightly modified rotation matrix, I could fix the issue and restore correct tracking.

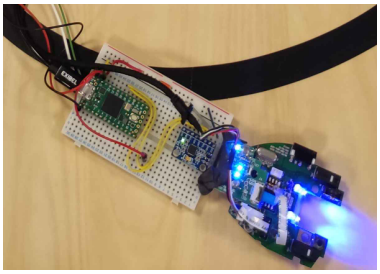
The position update can be written in matrix form as:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \Delta x & \Delta y \\ -\Delta x & \Delta y \end{bmatrix} \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix}$$

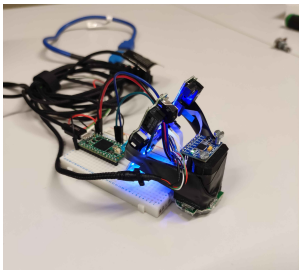
```
angleDeg = mpu.getAngleZ();
angleRad = (angleDeg * 3.14159) / 180.0;

 x += deltaX * cos(angleRad) + deltaY * sin(angleRad);
 y += -deltaX * sin(angleRad) + deltaY * cos(angleRad);
```

Below are some pictures showcasing the various iterations of this solution:



(a) Version 1



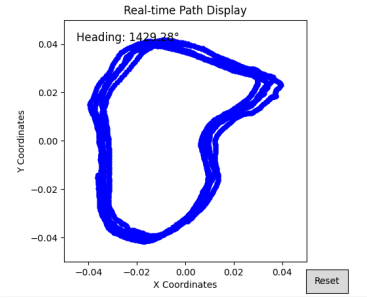
(b) Version 2

Figure 2: Test setups

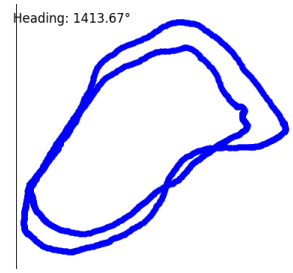
(a) Map after 3 laps using version 1

(b) Map after 3 laps using version 2

Figure 3: Test runs



(a) Map after 3 runs.



(b) Example of how the robot would traverse the track after mapping it.

Figure 4: Latest test runs.

The latest tests demonstrate how the mapping improved with additional calibration and better filter for estimating the angle. While significant progress has been made, the results are still far from ideal for this specific application. With more time and further modifications, this approach might become viable. However, the sensor selected for this concept already achieves accurate tracking at speeds of up to 2.5 m/s, all in a single compact package with easy to use libraries.

3.3 Motors

- **N20 1100 rpm DC motor with encoder:** The OTOS has a maximum speed of 2.5 m/s, making it a suitable—albeit slightly optimistic—target speed for this track. To reach this speed, the required RPM for the motors using 40mm wheels was calculated as follows:

$$\begin{aligned} \text{Given:} \quad v &= 2.5\,\text{m/s}, \quad d = 40\,\text{mm} = 0.04\,\text{m} \\ C &= \pi \cdot d = \pi \cdot 0.04 \approx 0.1256\,\text{m} \\ \text{RPM} &= \frac{v}{C} \cdot 60 = \frac{2.5}{0.1256} \cdot 60 \approx 1193.7\,\text{RPM} \end{aligned}$$

The robot weighs under 150g, including all components, chassis, and screws. To calculate the required torque, we consider an acceleration of at least 2 m/s^2, ensuring that the robot can complete the longest line in 1 second from a standstill. Since the wheels are small in diameter and weigh only 4g, the moment of inertia is neglected in the following calculations.

Given the robot's mass $m <= 0.150\,\text{kg}$ and acceleration $a = 2\,\text{m/s}^2$, the required force to accelerate the robot is:

$$\overrightarrow{F} = m \cdot \overrightarrow{a} = \frac{0.150}{2}\,\text{kg} \cdot 2\,\text{m/s}^2 = 0.15\,\text{N}$$

The required torque at the wheel is calculated by:

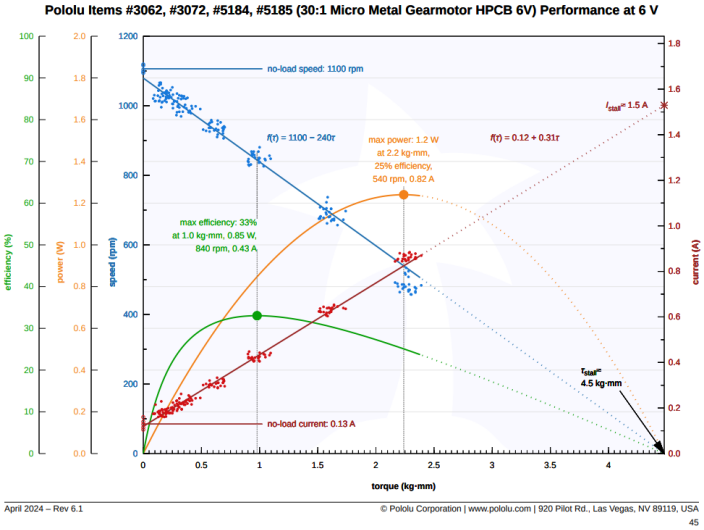
$$\overrightarrow{\tau} = \overrightarrow{F} \cdot \overrightarrow{r} = 0.15\,\text{N} \cdot 0.02\,\text{m} = 0.003\,\text{Nm}$$

To convert the torque from newton meters (Nm) to kilogram-centimeter (kg·cm), we use the conversion factor $1\,\text{Nm} = 101.9716\,\text{kg·cm}$:

$$\tau_{\text{kg·cm}} = 0.003\,\text{Nm} \cdot 101.9716\,\text{kg·cm/Nm} \approx 0.306\,\text{kg·cm}$$

The closest motor from Pololu that meets all the requirements is the 30:1 Micro Metal Gearmotor HPCB 6V, which has been chosen for this concept. The torque rating was selected based on the extrapolated stall torque specified in the datasheet. A look at the RPM-to-torque graph shows that the rotation speed is lower than the rated speed, which will result in reduced acceleration. However, since the motor only starts from zero velocity once during the race and will have already gained some speed before crossing the start line, I believe the loss will be negligible.

Encoders are needed to ensure the speed output from the line following PID controller is linear and symmetrical (when driving forwards).

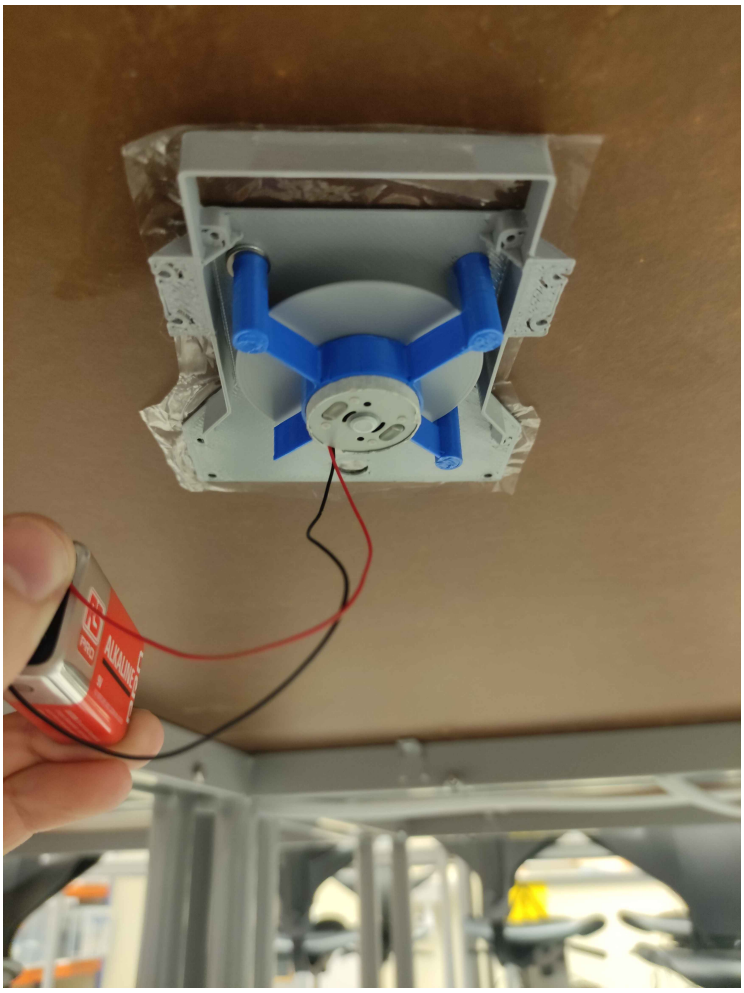


- **CD-rom motor:** Since the only force that stops the wheels from slipping is the friction coefficient, the grip can be calculated like this:

$$R = \mu \cdot N = \mu \cdot m \cdot g$$

Since increasing the mass or gravitational force is not an option, an alternative solution to enhance the force on the wheels—besides using ones with a higher friction coefficient—is to introduce an additional downward force acting on the robot. One way to achieve this is by creating a vacuum, a technique inspired by similar approaches used in micromouse competitions [4].

The motor I selected, combined with an optimized fan I designed to generate a low-pressure chamber beneath the robot, provided at least 50g of additional downforce using this test setup. This increases stability and grip, improving the robot's performance.



- **TB6612FNG Motor driver:** Capable of driving two separate motors, this driver delivers over 1.2A per channel, which is more than sufficient for this application. Its compatibility with both 3.3V and 5V logic, as well as its support for motor supply voltages of up to 15V, allows for testing a wide range of microcontrollers and motors.

3.4 Power Supply

- **Li-Ion 18650 3.7V:** I chose this battery due to its high energy density, Wh/g and small size. Based on the data sheets for the motors, sensors, and microcontroller, the average current draw is estimated to be approximately 600mA.

  To calculate how long a 3.7V, 2800mAh battery will last at a 6V load of 600mA, we need to adjust for the difference in voltage and consider the capacity in terms of watt-hours.

$$\text{Wh} = \text{V} \times \text{Ah}$$

$$\text{Wh} = 3.7V \times 2800\text{mAh} = 3.7V \times 2.8\text{Ah} = 10.36\text{Wh}$$

$$\text{P} = \text{V} \times \text{I} = 6V \times 600\text{mA} = 6V \times 0.6A = 3.6\text{W}$$

$$\text{Runtime} = \frac{\text{Battery Energy (Wh)}}{\text{Power Consumption (W)}} = \frac{10.36\text{Wh}}{3.6\text{W}} \approx 2.88 \text{ hours}$$

  This means the runtime will be around 2.88 hours, which provides ample time for testing. The current draw, including startup currents, is within the battery's maximum discharge current.

  The battery's form factor makes it easy to mount on the robot and remove for recharging. Since the school has multiple of these batteries available, the cost of the battery can be excluded from the project budget.

- **Charger and voltage regulator module:** By combining a battery management system (BMS) and a step-up converter on a single board, it makes the system more compact. This setup not only provides a stable high voltage to the rest of the system, but also ensures battery protection.

3.5 Wheels

- **Pololu wheels 40×7mm:** These wheels were selected based on their compatibility with the motor shafts used and the results from tests shown in the following videos: [1], [2], [3]. Additionally, conversations with previous competitors revealed that these wheels offered better grip compared to silicone or other custom-made wheels.

- **Pololu ball caster with 3/8" metal ball:** These are used to minimize friction on the protruding part of the robot. Metal was chosen over plastic based on feedback from previous competitors who used them. I learned that they were susceptible to dust buildup, which could block and grind away the plastic ball.

4 CAD Design

4.1 Pictures

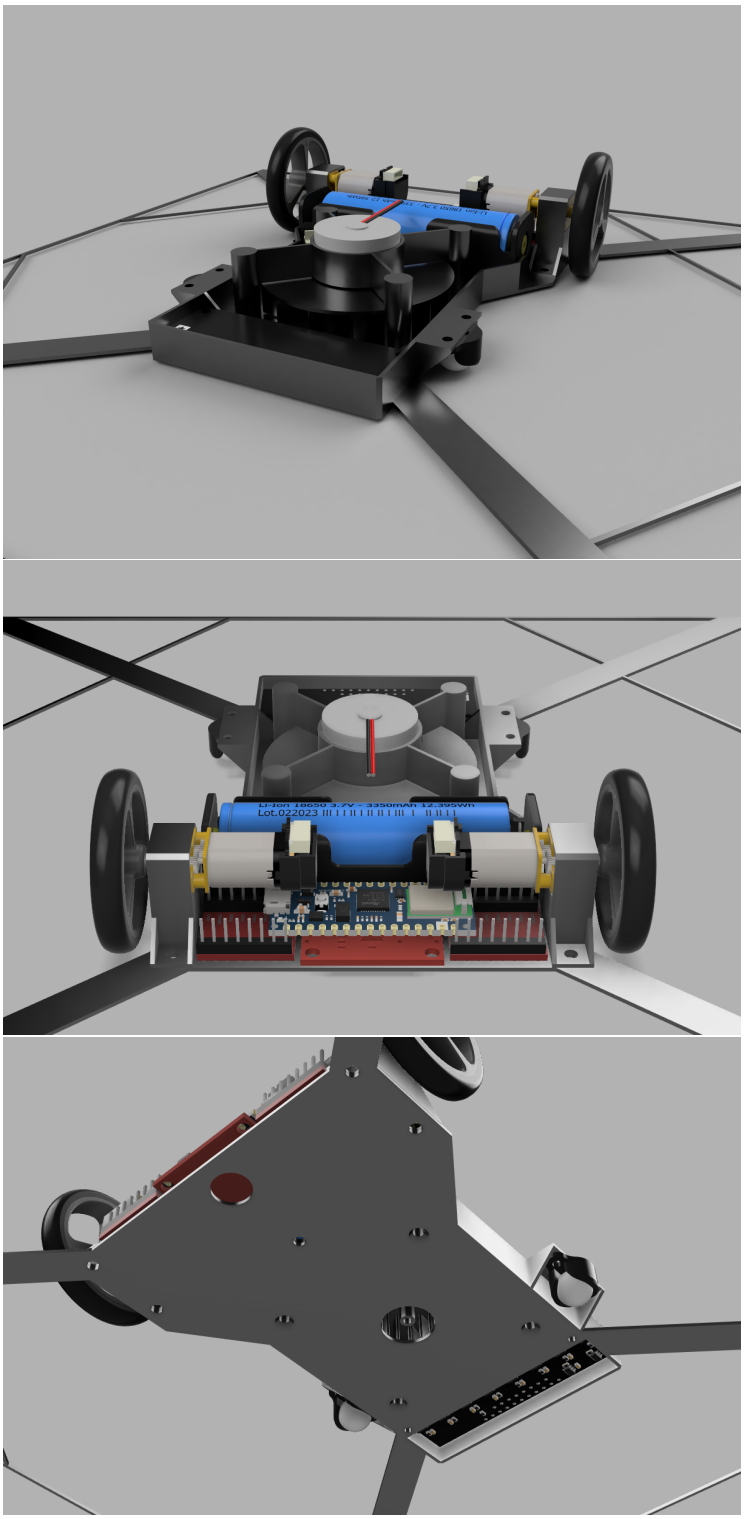


Figure 5: Different angles of the CAD model (breadboard not included). Looking from the back, the power supply module is on the left, OTOS middle and motor driver on the right with the microcontroller in the middle on top. Higher resolution pictures are provided on GitHub at Concept_Prometheus.

4.2 Design Modifications

- **Size Reduction:** The model is nearly half the size of the Phase 1 design in all dimensions, making it lighter and easier to maneuver. The reduced mass, particularly further from the center, significantly decreases the moment of inertia, improving overall handling and responsiveness.
- **Turn Rate Optimization:** During testing in phase 1, I realized that the turn rate was limited by the width of the robot. With the width halved, it should manage faster changes in the track without understeering.
- **Wheel Mounting:** Compared to the previous design, placing the motors on top of the main body makes the whole robot lay horizontally instead at an angle. This reduces friction and puts the center of mass further down, making it more stable.

4.3 Component Layout

- **IR Array and Wheel Configuration:** The distance relation between the IR array and the wheels remains in an isosceles triangle configuration, which ensures accurate tracking without oscillation. Further or closer placement may increase or decrease the sensitivity of the system too much. The array also is now mounted at the correct distance from the ground to avoid unstable readings In addition it has now a wall around it to protect it, while also blocking outside UV light that might affect it.
- **Component Placement:** Most of the components (battery most importantly because it has the highest mass) are located towards the back of the robot, near the wheels, for balanced weight distribution and provide higher downforce on the wheels, improving grip. The boards are arranged in a "sandwich" configuration, stacked on a central breadboard to save space and reduce costs.

4.4 Materials and Build

- **Chassis Material:** The chassis is made of PLA plastic, which allows for easy modifications and rapid prototyping at no additional cost. It also has a high tensile strength for a thermoplastic, making it ideal for a lightweight construction.
- **Lightweight Design:** The model is designed to be as lightweight as possible by using thin walls that maintain structural strength only where needed.

4.5 Functional Design Features

- **Grid Design:** The grid around the main body helps to ensure that the robot is above the track line at all times, even when cutting corners. Using odometry without it would diminish the upsides significantly.
- **Skirt:** Not included in the model, but a thin plastic skirt is taped around the edges. During testing it provided roughly twice the suction force than without. In addition, it was found that the bottom of the robot should have as much area as possible to maximize the vacuum effect.

4.6 Modular Design

- **Motor Mount:** The motor mount is modular and can be easily adjusted or swapped to accommodate different wheel sizes.
- **Circuit Boards:** Only using modules makes it easier to troubleshoot and swap parts if needed.

5 Wiring Diagram

The wiring schematic demonstrates the connections between the IR sensor array, motors, motor driver, battery, and microcontroller. Each signal wire connects to an digital input on the Arduino, except for the pins on the motor controller that needs a PWM signals and the I2C bus for the OTOS.

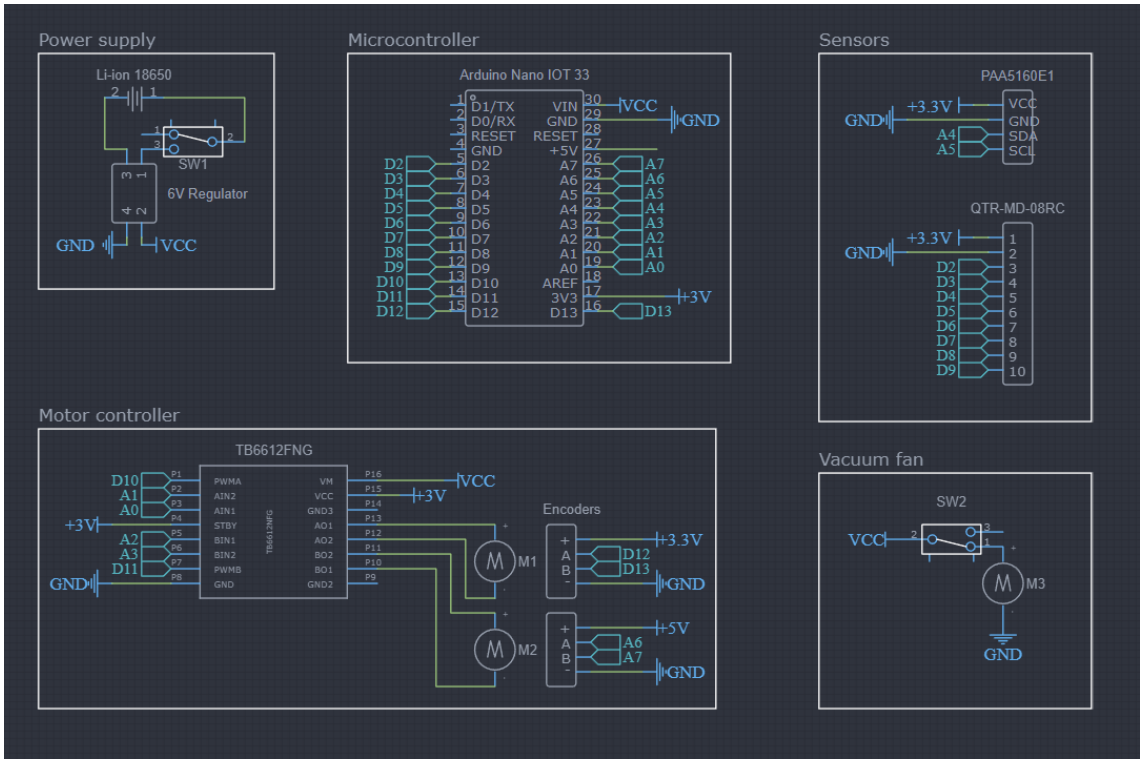


Figure 6: Wiring diagram of the line-following robot

6 Program Structure

The program is a state machine with the following states, which can be set via Bluetooth (not fully implemented yet). The nearly complete code is available on GitHub at Concept_Prometheus.

```
enum State {
    IDLE = 0,
    FOLLOW_LINE = 1,
    MAPPING = 2,
    FOLLOW_MAP = 3,
    RESET_MAP = 4
};
```

6.1 Idle

Stops the motors entirely.

6.2 Follow line

Uses the IR array to detect the line and sends the position to a PID controller. The controller adjusts the angle of the robot accordingly. If the sensor detects a 90-degree turn, one motor stops while the other keeps running until the sensor is back over the line.

6.3 Mapping

Operates similarly to the previous state but also utilizes an odometry sensor. Records positional points at regular intervals to map the track.

6.4 Follow map

Uses the Pure Pursuit algorithm to calculate the angle to a target point on the map that is a fixed distance away from the robot. Continuously adjusts the robot's heading to follow the mapped path with PID control. I think this simple algorithm would work well to skip corners while being easy to tune.

6.5 Reset map

Stops the motors and resets all map data.

7 Discussion

7.1 Strengths and Weaknesses

Microcontroller

- + Compact, lightweight, with sufficient memory and processing power for the program.
- + Integrated Bluetooth and WiFi capabilities.
- - Operates on 3.3V logic instead of 5V, which can limit compatibility with certain components.
- - The number of available ports is near the limit, requiring efficient use of the microcontroller's pins.

IR Array

- + Offers good accuracy with a small form factor.
- - Uses many ports on the microcontroller.
- - The array could technically be reduced to only two sensors.

OTOS

- + High accuracy.
- + Easy to use.
- + Compact design.
- - Very high cost.

Motors

- + Easy to control.
- - Wheel motors may have insufficient torque under certain conditions.
- - Excess torque for the fan motor.
- - No speed regulation for the fan motor.

Battery

- + High capacity.
- + Easy to secure.
- + High discharge rate.
- - High weight.
- - Cylindrical shape makes it less compact.
- - Low voltage requires step-up conversion.

Wheels

- + High friction coefficient for improved grip.
- + Lightweight.
- + Easy to mount onto the motors.
- - Medium cost compared to alternatives.
- - Fixed diameter limits experimentation.

7.2 Further Improvements

- **Vacuum Fan:** Testing revealed that both the fan and motor were larger than necessary for the application. The fan's size can be significantly reduced, since generating high airflow is not critical in this context. Similarly, the motor can have much lower torque, as the primary factor influencing performance was its RPM rather than raw power.

  To improve the design, using two smaller fans powered by coreless DC motors, similar to those in micromouse competitions, would be better. These could be paired with a suitable driver to enable speed control, which would also help reduce friction when high acceleration is unnecessary. Or alternatively, a small electric ducted fan (EDF).

- **Chassis:** Moving the flat area more towards the center of rotation would improve the down-force placed on the motors instead of the caster wheels.

  For the final version, the material of the base could be swapped with a sheet of thin PCB to reduce the weight further.

  The width should maybe be increased because of the higher RPM of the motors, to avoid too fast turn rate.

  Using only two rods or similar structures of carbon fiber protruding from the center of the robot to its sides, could reduce both the weight and friction of the robot while still ensuring the track line is covered.

- **OTOS:** Consider cost-effective alternatives without compromising performance. Alternatively, the design could utilize only two IR sensors for line-following during mapping, further reducing inertia.

- **Wheels:** Switch to silicone wheels or something similar to be able to test the robot with different wheel diameters. Only using one caster wheel and putting it further back should also decrease the inertia.

- **Battery:** The runtime is way longer than needed for the race. A 7.2V 300mAh LiPo battery would resolve the downsides mentioned earlier while resulting in 15 min of runtime.

- **Code:** Another way to find the shortest path could be done using the Ramer–Douglas–Peucker algorithm to reduce the number of points and then use a B-spline function. Setpoints for different speeds at different lengths of track could also be implemented. The speed could also be optimized by comparing the acceleration between the OTOS and encoders, and adjusting the speed to ensure low wheel slippage.

8 Conclusion

This project demonstrates a concept for a line-following robot capable of autonomous navigation. By integrating a combination of precise sensors, efficient motors, and thoughtful design, the robot achieves a balance between speed, accuracy, and stability. While there are areas for improvement, such as refining the vacuum system, code and optimizing weight distribution, the current design provides a solid foundation.

I believe this concept is worth pursuing further, as the proposed modifications have the potential to create an exceptionally fast and efficient line-following robot.

References

[1] Organized Chaos. *Combat Robotics Wheel Traction Showdown*. Accessed: 2024-10-15. 2021. URL: https://www.youtube.com/watch?v=hJ380d-ZGSs&ab_channel=OrganizedChaos.

[2] Team Panic. *How good is a screw wheel*. Accessed: 2024-11-04. 2022. URL: https://www.youtube.com/watch?v=iY2uiyM_SfI&pp=ygUhV2hlZWwgR3JpcCBTdHJlc3MgVGVzdCB0ZWFtIHBhbmlj.

[3] Team Panic. *Wheel Grip Stress Test*. Accessed: 2024-10-15. 2022. URL: https://www.youtube.com/watch?v=p5tnFrEYutY&t=1s&pp=ygUhV2hlZWwgR3JpcCBTdHJlc3MgVGVzdCB0ZWFtIHBhbmlj.

[4] Veritasium. *The Fastest Maze-Solving Competition On Earth*. Accessed: 2024-09-06. 2023. URL: https://www.youtube.com/watch?v=ZMQbHMgK2rw&pp=ygULbWljcm8gbW91c2U%3D.