CSC 615 Spring 2023
Final Project Report

Team Footmobile

Axel Biehler
Hann Zhao
Mael Teyssedre
Xiao Deng

GitHub repo: https://github.com/CSC615-2023-Spring/csc615-term-project-xdeng0

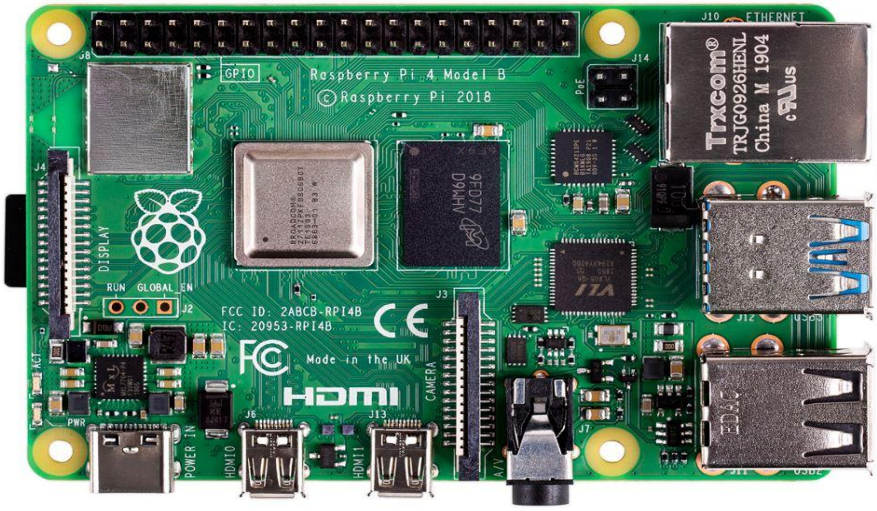
Task Description

Our task for this project is to build a self-driving vehicle that is able to keep itself running on a black line and to detect and avoid obstacles. We're expected to complete this project by applying concepts and techniques we learned earlier this semester, such as how to read from and write to Raspberry Pi GPIO pins, how to run a DC motor, and how to use threading to monitor the states of multiple sensors. We need to write a program that runs on a Raspberry Pi which is the brain of the car to adjust the car's movement in real time based on the states of sensors such that the car is able to follow a path of black tape on the ground and to navigate around obstacles in the path.

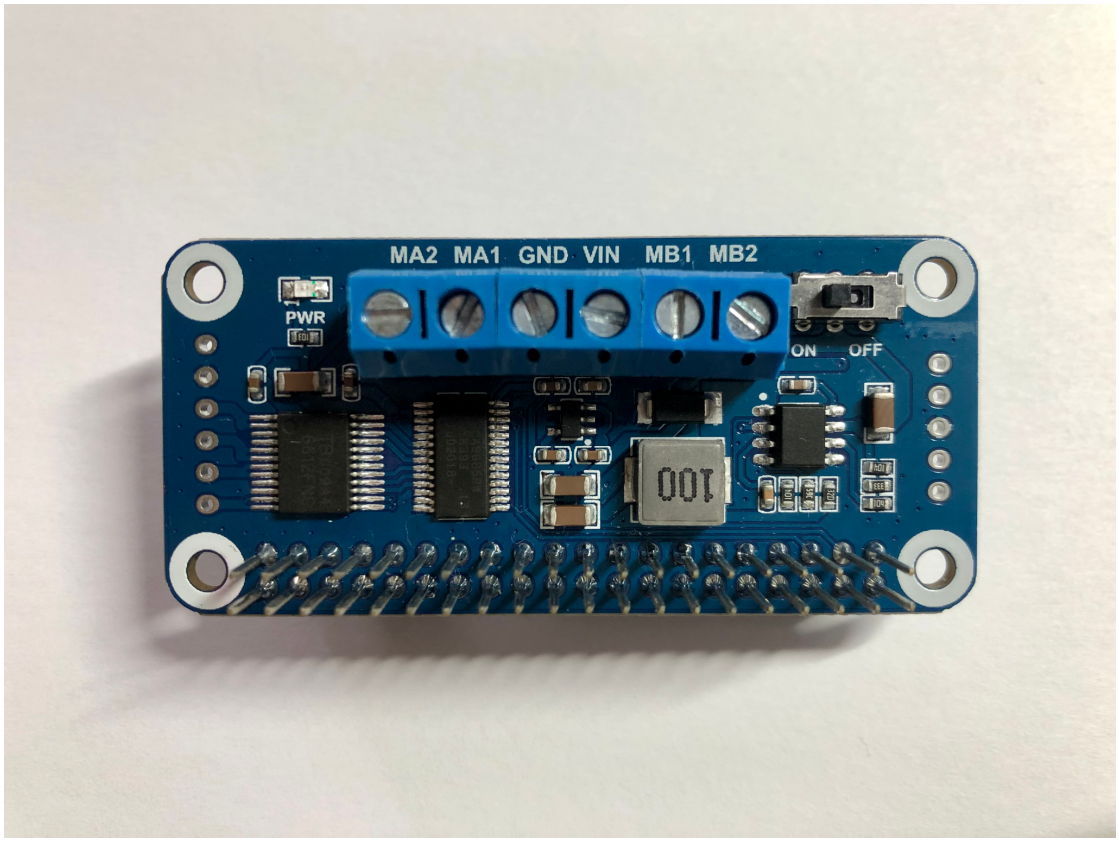
Building the Robot

Parts used

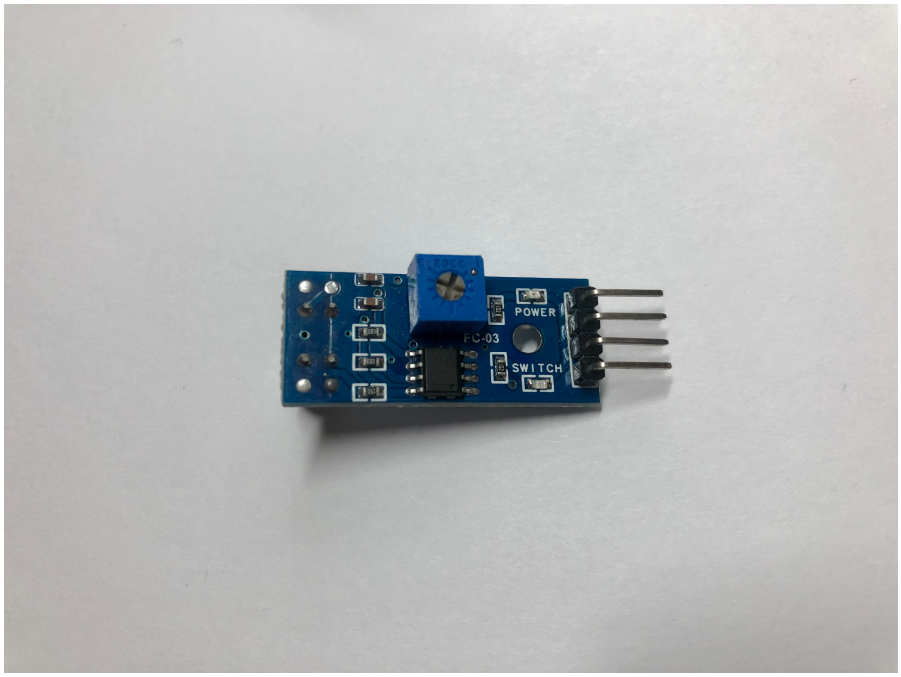
- Raspberry Pi 4



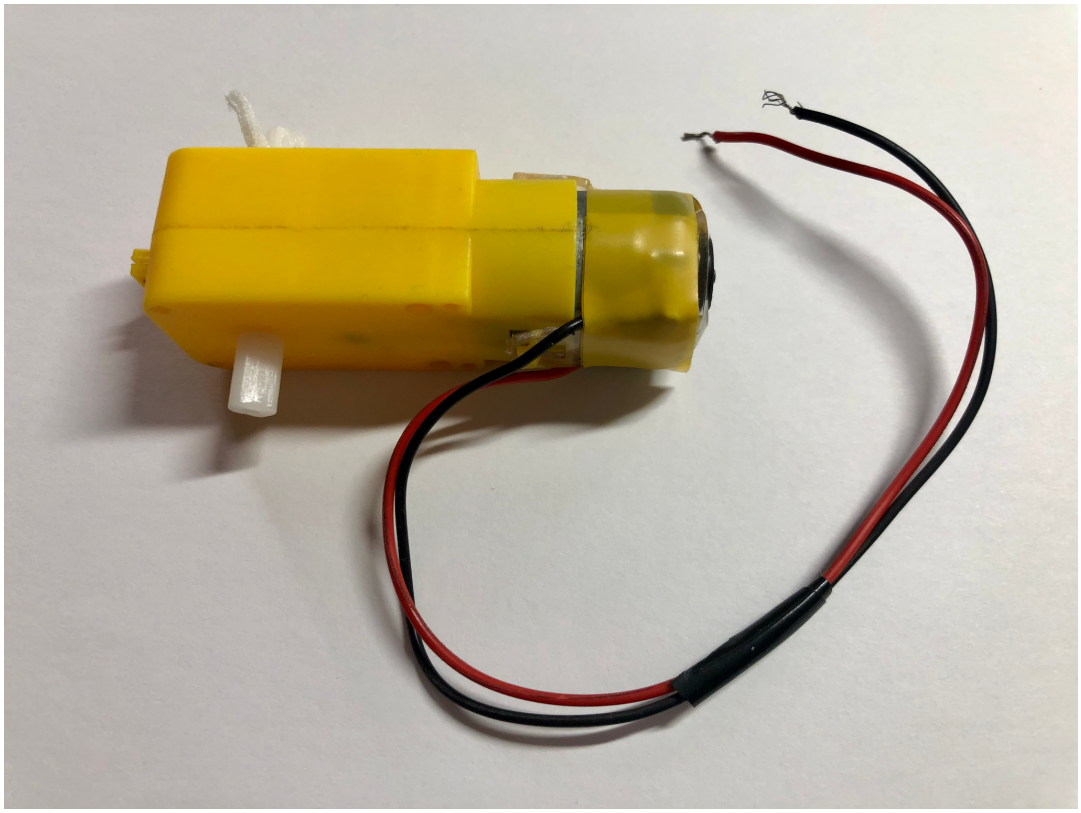
- Waveshare Motor Driver HAT x 2



- TCRT5000 Line Tracking Sensor x 2



- DC Motor x 4



- Omnidirectional Wheel x 4

- battery x 3

- Slamtec RPlidar A1

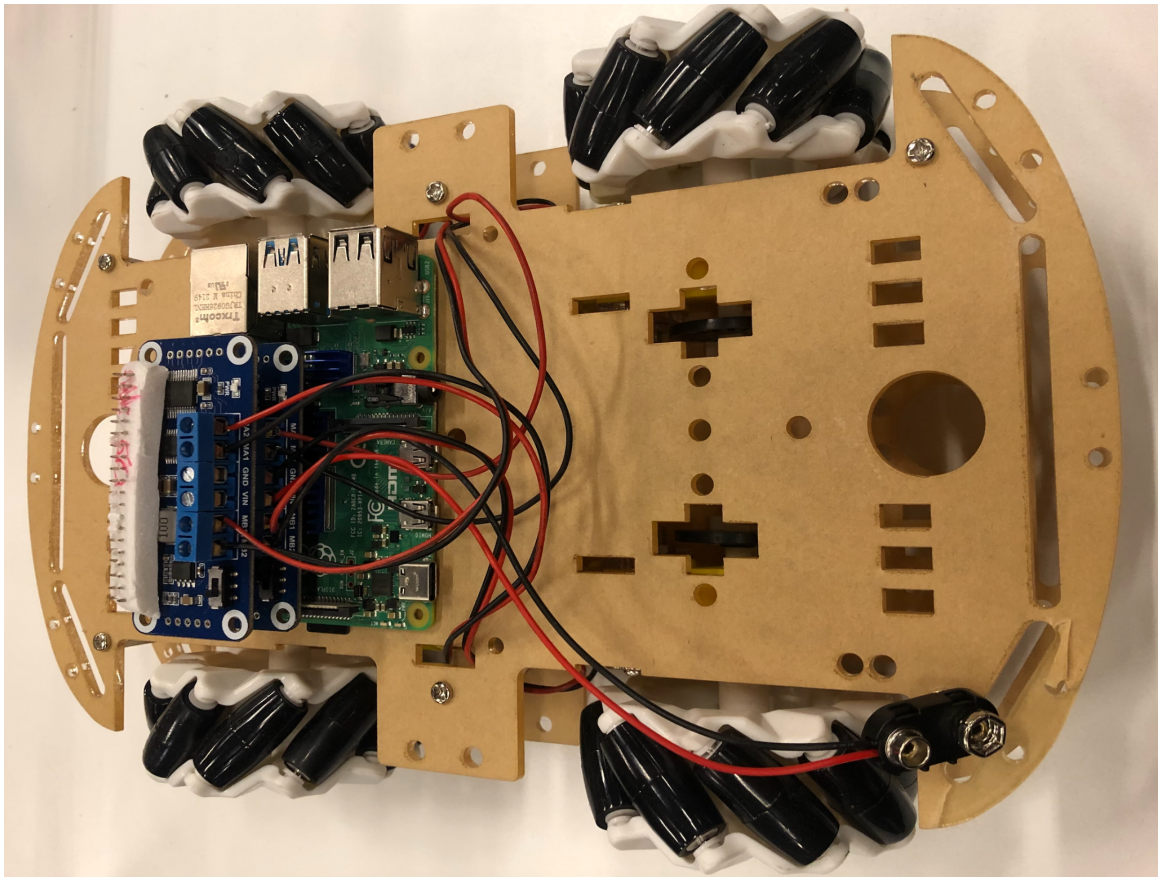
- Power bank

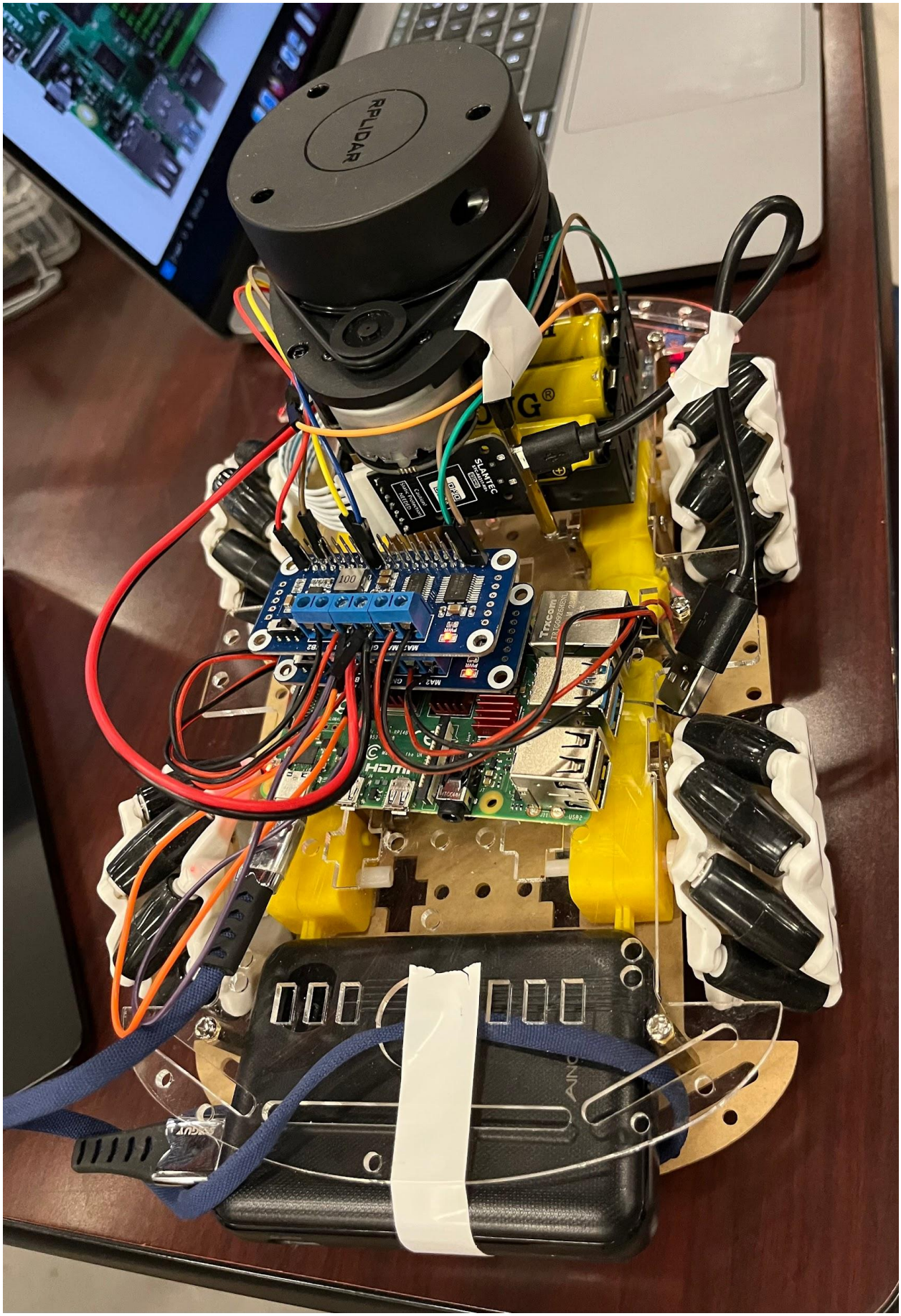
How bot was built

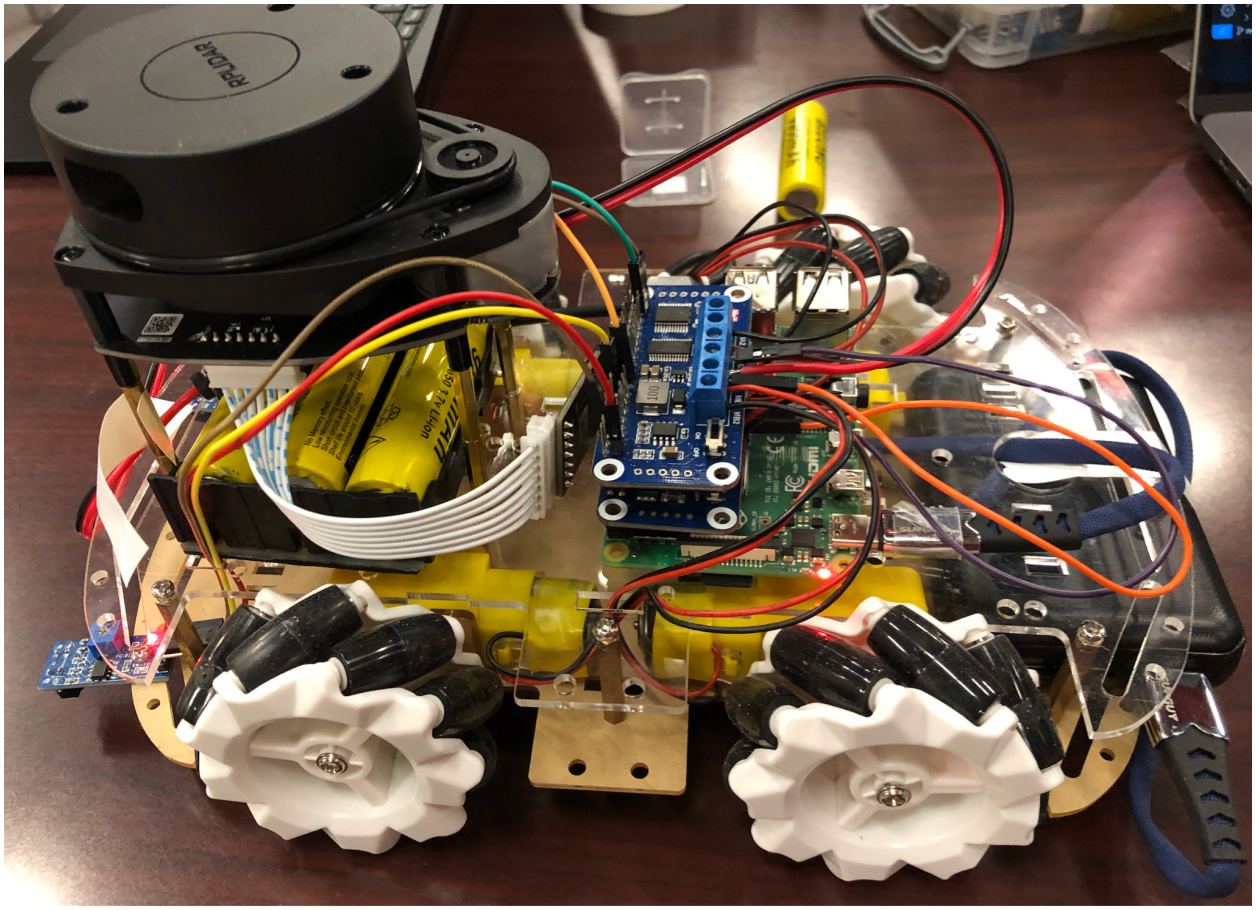
1. Assemble the car
    a. assemble the car chassis
    b. fix motors to the chassis
    c. attach wheels to motors
2. Select sensors / parts and write test program to see how they perform
    a. we first decided that we're going to use 4 omnidirectional wheels, which means we need 2 motor driver HATs to control them
    b. we decided to use 2 line sensors and wrote a simple test program to test sharp turns and slight turns
    c. for object avoidance, we first decided to use 2 obstacle sensors in the front of the car and 1 in the back of the car, but we noticed that these sensors are not very effective for detecting dark objects
    d. then we tried to use the echo sensors, but we were not able to find a way to attach the echo sensors firmly to the car, and none of us has experience with designing 3D-printed parts
    e. after reading the RPlidar documentation and introduction to SDK, we decided to use lidar for object avoidance since it scans 360 degrees and there are standoffs that we can use to install and raise the lidar

3. Connect parts and sensors to the Raspberry Pi
    a. stack 2 Waveshare Motor Driver HATs on the Raspberry Pi's GPIO pins

b. connect front motors to top Motor Driver HAT and rear motors to bottom Motor Driver HAT
c. Install 2 line sensors on the front of the car (bottom chassis, one on the left and one on the right) using screws and connect the sensor to the Raspberry Pi's GPIO pins using jumper wires
d. drill holes on the car's top chassis for lidar standoffs and install lidar
e. connect battery pack to terminals on Motor Driver HATs




RPLIDAR

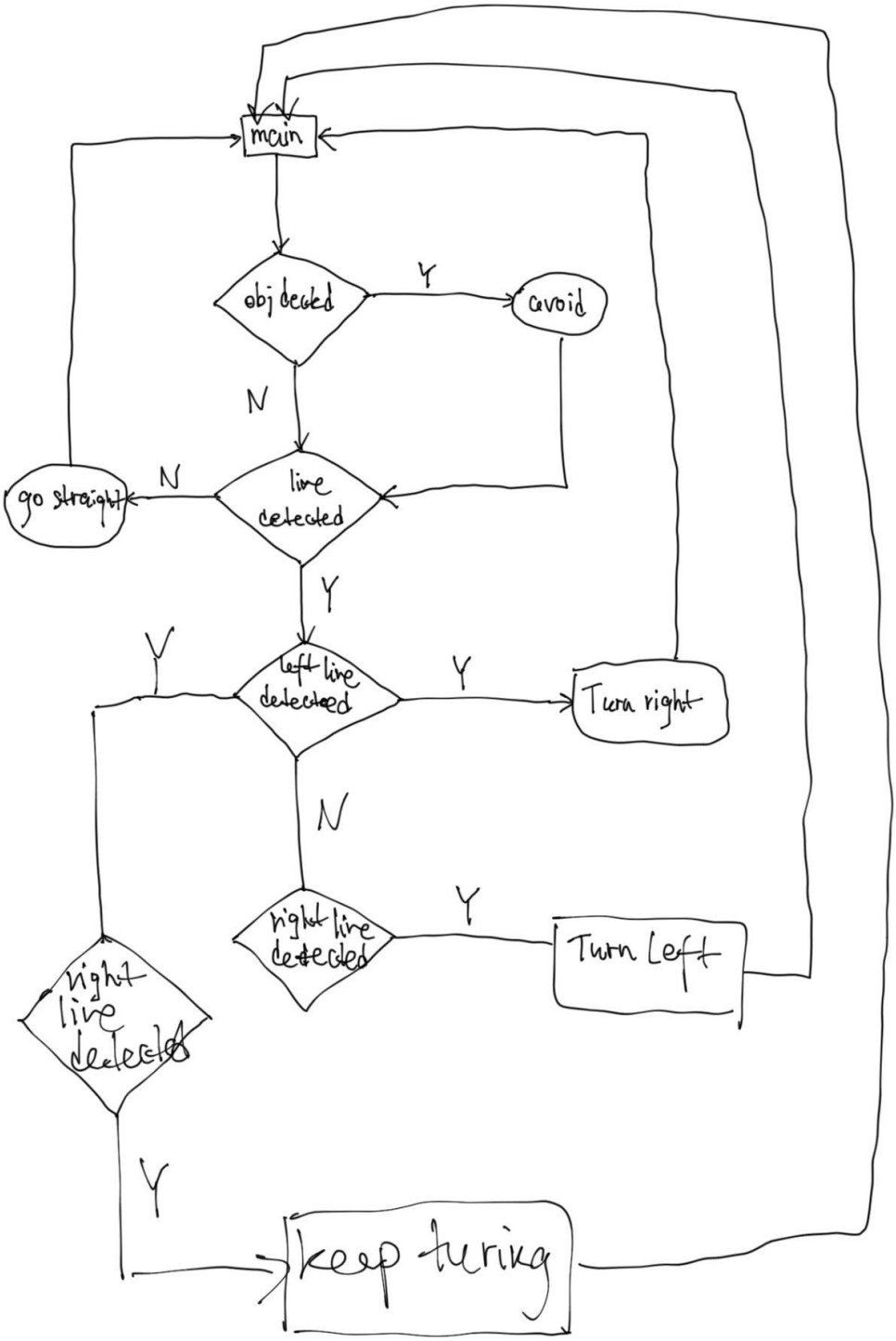


4. Programming the car

Libraries / Software used

- Motor Driver HAT sample code by Waveshare team
    - modified DEV_Config.c and DEV_Config.h
    - modified PCA9685.c and PCA9685.h
- pigpio library
- RPlidar SDK

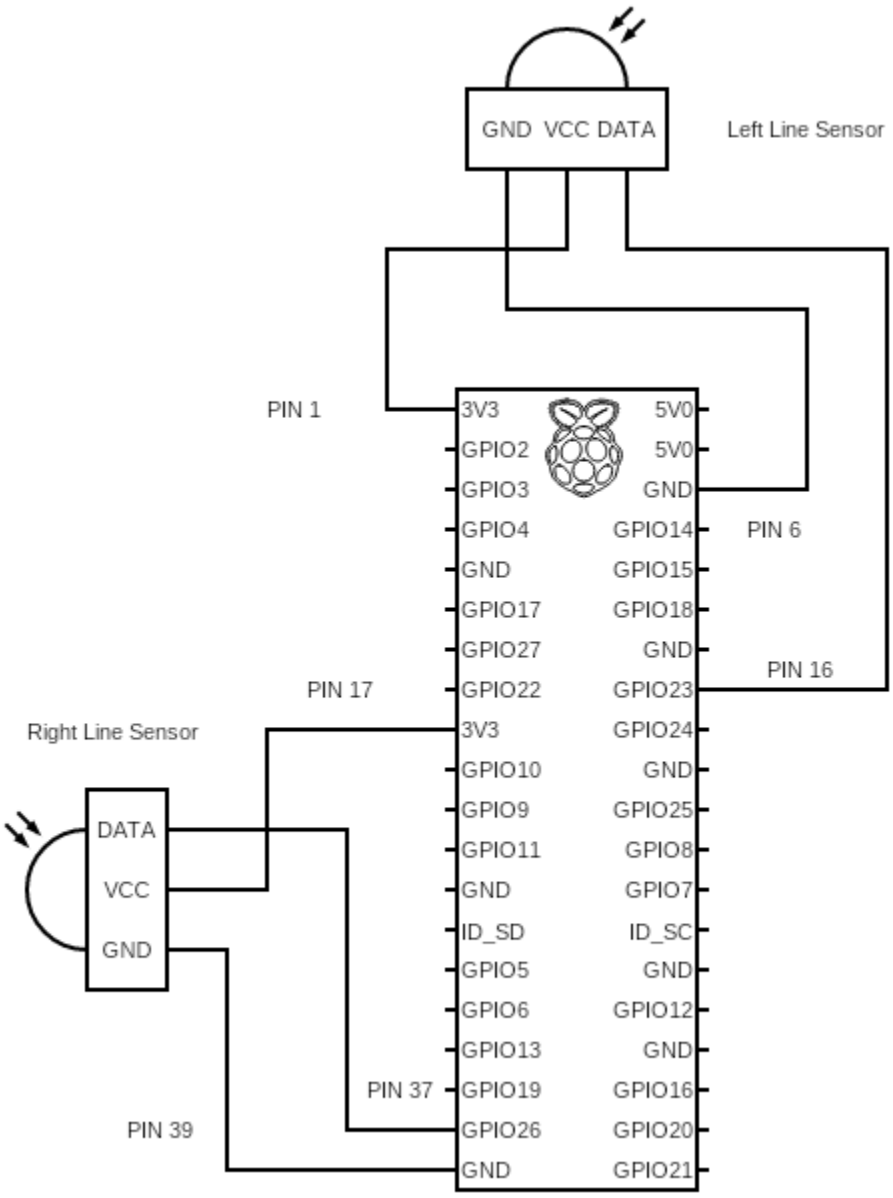
Flowchart


main
obj detected
Y
avoid
N
go straight
N
line detected
Y
Y
left line detected
Y
Turn right
N
right line detected
Y
Turn Left
right line detected
Y
keep turning

Pin assignments

Physical Pin number	GPIO Pin number	Use
1	N/A	left line sensor VCC
6	N/A	left line sensor GND
16	23	left line sensor data out
17	N/A	right line sensor VCC
39	N/A	right line sensor GND
37	26	right line sensor data out

Hardware Diagram



What worked well

- We were able to use GitHub branches for features of the car
    - we created dev branches such as “drive” and “echo” so we can work separately but in parallel to test our code
- The assembly of the car went well. The first day that the project was announced, we were able to build the car chassis and install motors and omnidirectional wheels.

What were the issues

The first issue we are facing is with controlling the omni wheels. Only two of the wheels are controllable due to a problem with the driver motor hat we have. To fix this, we replaced the current driver motor hat with one that has a different address. Moving on, we have also encountered an issue with the obstacle avoidance sensors. They are unable to detect objects in front of the robot. We decided to use the A1 RPLIDAR for object avoidance instead. The raspberry pi 4 also stopped working so one of our teammates has to get another one. We also experienced recurring problems with the Raspberry Pi freezing or rebooting unexpectedly. Later we discovered that the charging cable was the cause of these issues.