Final Project - Car Project

CSC-615-01 Spring 2024
Dat Vo (922171975), Charter Lin (922577939), Kotaro Iwanaga (922782507), Jimmie Wu (921549444)
Github-Name: CharterLin
Github Link: https://github.com/CSC615-2024-Spring/csc615-term-project-CharterLin

Task Description:

Dat Vo - coding, documentation

Charter Lin - Provide idea and help with wiring, documentation

Kotaro Iwanaga - Provide idea and help with wiring, documentation

Jimmie Wu - Provide idea and help with wiring, documentation

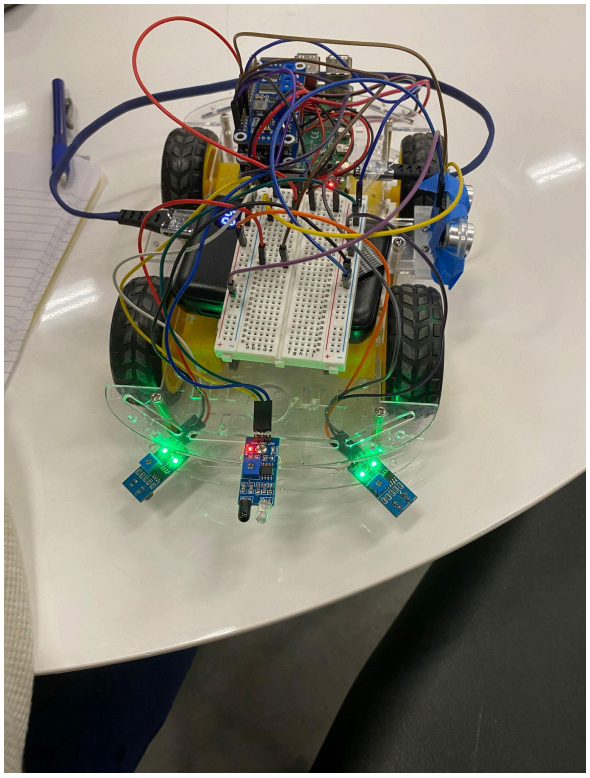
Parts / Sensors Used (include photo, and part numbers):

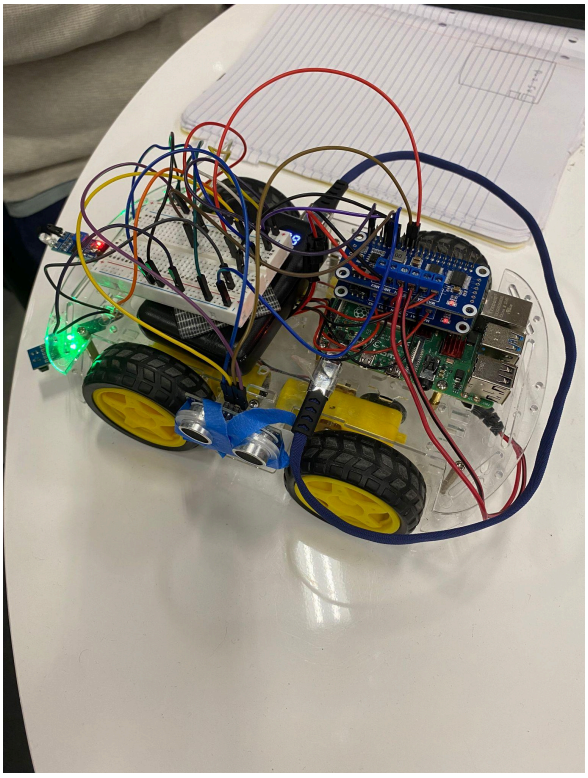
- Four motors: to drive the car.
- Two line sensors (TCRT5000 Infrared Reflective IR): to detect and make sure the car stays on the line, when either side of the sensor is triggered, the car will adjust itself to the triggered side.
- One obstacle/IR sensor (OSOYOO Obstacle Avoidance Sensor Module): to detect the obstacle in front of the car, when the IR sensor is triggered, the car should turn to the side and use the echo sensor to make its way around the obstacle.
- One echo sensor (HC-SR04): to measure the distance between a car and an obstacle.

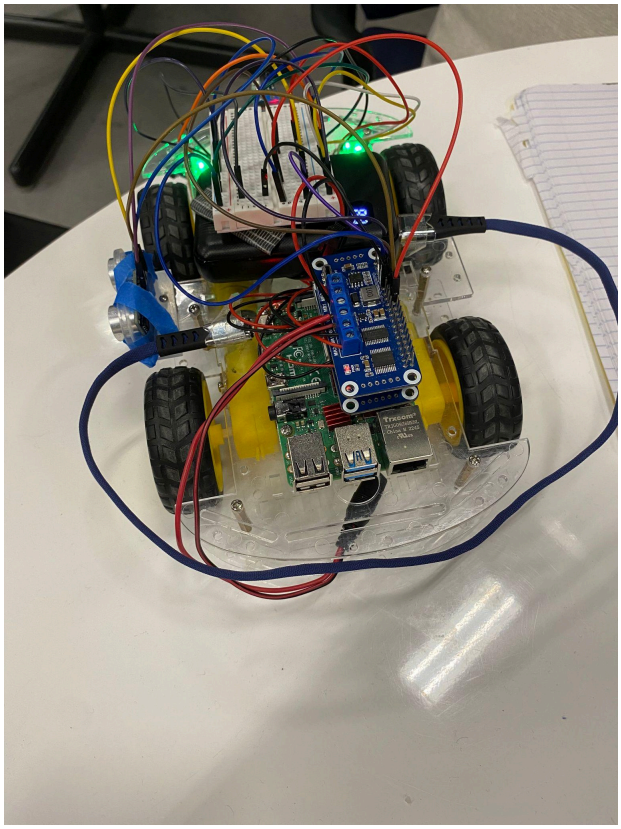
How was bot built (good to include photo):

The bot is built with four motors to control the motion of the car, two line sensors in front of the two front wheels, along with one IR sensor that can look for obstacles in front. On the side, there is echo sensor that can detect the distance of the obstacle, when the echo sensor is able to detect the obstacle, move the car forward until the sensor detects nothing or far distance, then turn back onto the line.

Building the Robot (include photos):



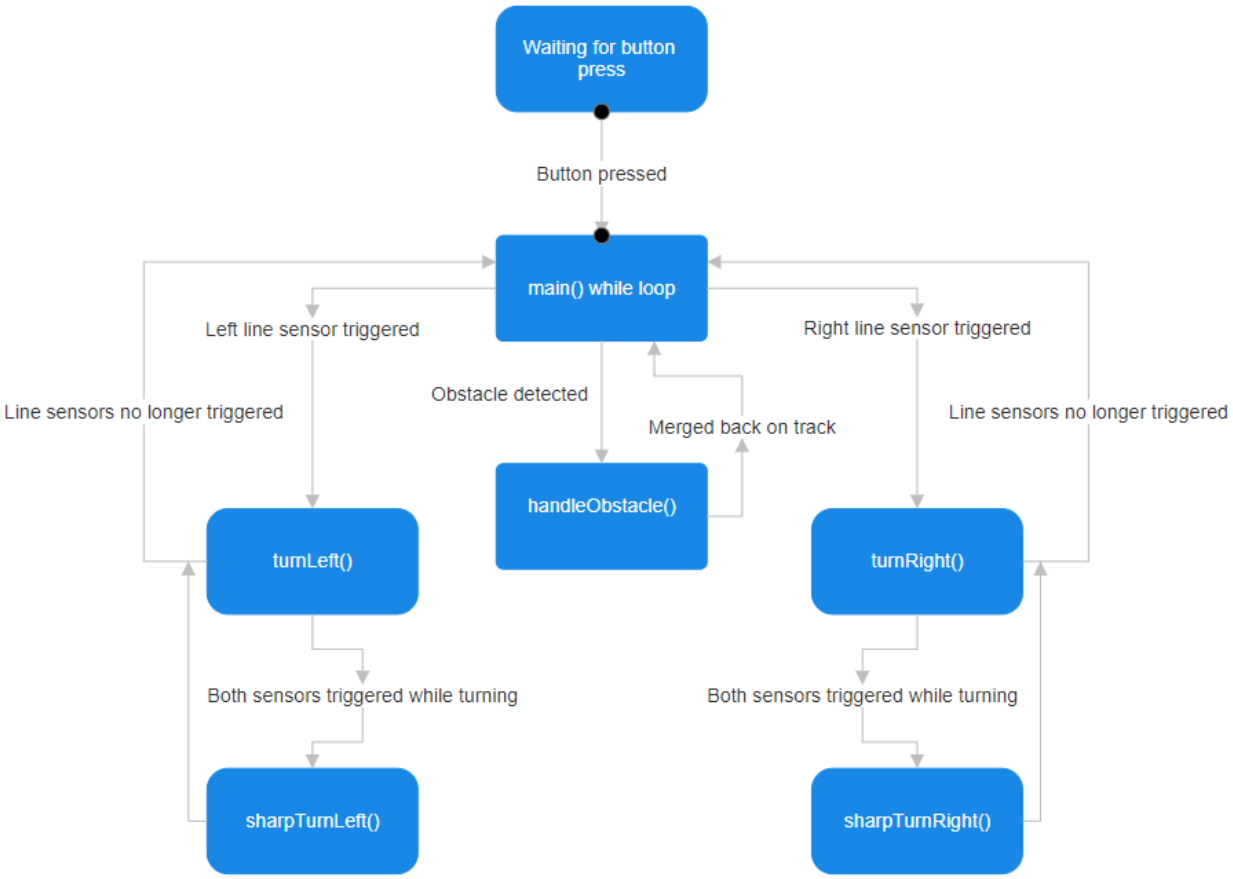




libraries/software (include full reference):

- VS Code
- GPIO library (Reference: https://abyz.me.uk/rpi/pigpio/cif.html)
- PCA9685 (Reference: https://www.waveshare.com/wiki/Motor_Driver_HAT)
- DEV_Config (Reference: https://www.waveshare.com/wiki/Motor_Driver_HAT)
- Pthread
- Signal
- sys/time
- C POSIX (unistd)

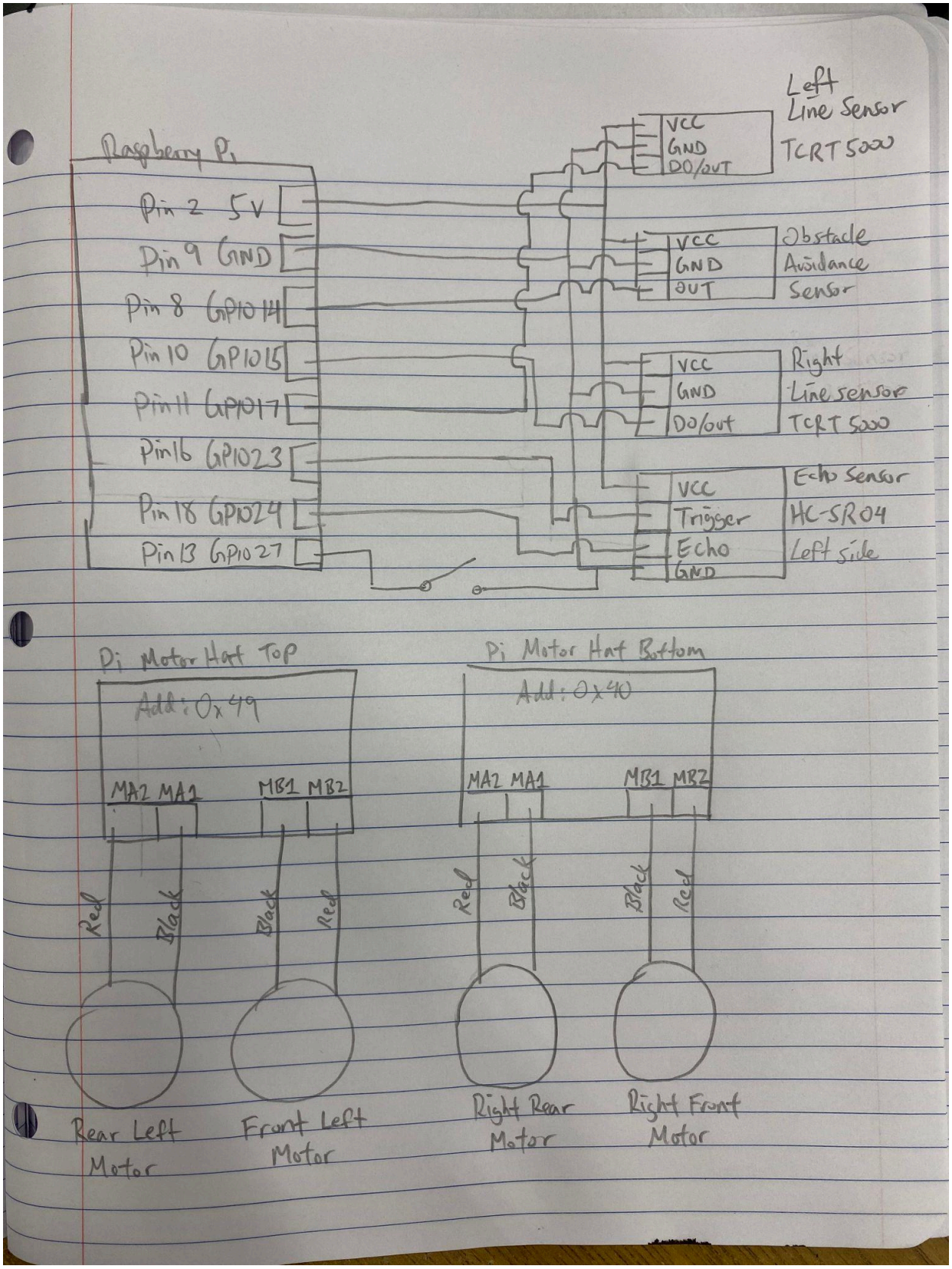
Flowchart of the code:



Pin Assignments used:

Pin 5: 5V
Pin 9: GND
Pin 8: GPIO 14 reads output from obstacle sensor
Pin 10: GPIO 15 reads output from right line sensor
Pin 11: GPIO 17 reads output from left line sensor
Pin 16: GPIO 23 is the trigger for the echo sensor
Pin 18: GPIO 24 reads the output/echo from the echo sensor
Pin 13: GPIO 27 is connected to a button switch

Hardware Diagram:


Raspberry Pi
Pin 2 5V
Pin 9 GND
Pin 8 GPIO14
Pin 10 GPIO15
Pin 11 GPIO17
Pin 16 GPIO23
Pin 18 GPIO24
Pin 13 GPIO27
Left Line Sensor
VCC
GND
DO/OUT
TCRT 5000
Obstacle Avoidance Sensor
VCC
GND
OUT
Right Line sensor
VCC
GND
DO/out
TCRT 5000
Echo Sensor
VCC
Trigger
Echo
GND
HC-SR04
Left side
Pi Motor Hat Top
Add: 0x49
MA2 MA1
MB1 MB2
Red
Black
Black
Red
Rear Left Motor
Front Left Motor
Pi Motor Hat Bottom
Add: 0x40
MA2 MA1
MB1 MB2
Red
Black
Black
Red
Right Rear Motor
Right Front Motor

What worked well:

- The overall team collaboration is great, everyone contributed a fair amount of ideas and work. When we run into issues we would talk and brain storm ideas. We would then consider each idea and try them all out. We were open to other ideas and not dead set on one specific way to do anything. We met usually twice a week and had very responsive communication on discord so everyone was up to date and involved.
- With our tank tread method, we were able to get the robot to follow the line very easily. Once a line sensor detects the line, the robot does miniscule tank turns until both sensors are off the line.

What were issues:

- The first issue we had was when two line sensors read the line at the same time.
- We solved this issued by using a “sharp turn” rather than a tank turn on sharp angles. The sharp turn involves moving the wheels on only one side of the robot, either left or right, and leaving the other side off.

- An issue we had was that the robot was not driving consistently. It would sometimes turn at a good angle and other times at a bad angle. The distances from the object and when we detect it would also vary.
- We solved this issue by adding a 1k resistor to the echo sensor. We added it between the echo pin and the GPIO read pin. This resulted in a more consistent turn and reading which we were able to use to successfully navigate an obstacle.

- Another issue we had was that the robot was not moving forward in a straight line. It would slightly go forward leaning towards the right.
- We solved this issue by lowering the power level on the right motors by 10%. This allowed the robot to go in a near straight line.

- Another issue we had was when the robot would make a sharp turn. This would sometimes cause both line sensors to read which made the robot turn left and right. This would sometimes lead the robot to go off course.
- We resolved this by telling the robot to do a “sharp” turn when both line sensors are triggered. The “sharp” left turn would tell the robot to continue the turn in the direction before both sensors were activated.

- Another issue was that the robot was making an inconsistent first turn when an obstacle was detected. We tried to use an object IR sensor on the side but it did not give us a good consistent turn. We then tried 2 IR sensors and 3 IR sensors on the side but the turn was still no good.

- We resolved this by adding an echo sensor to the side and replaced all the object sensors. This gave us a better turn that we felt confident in.