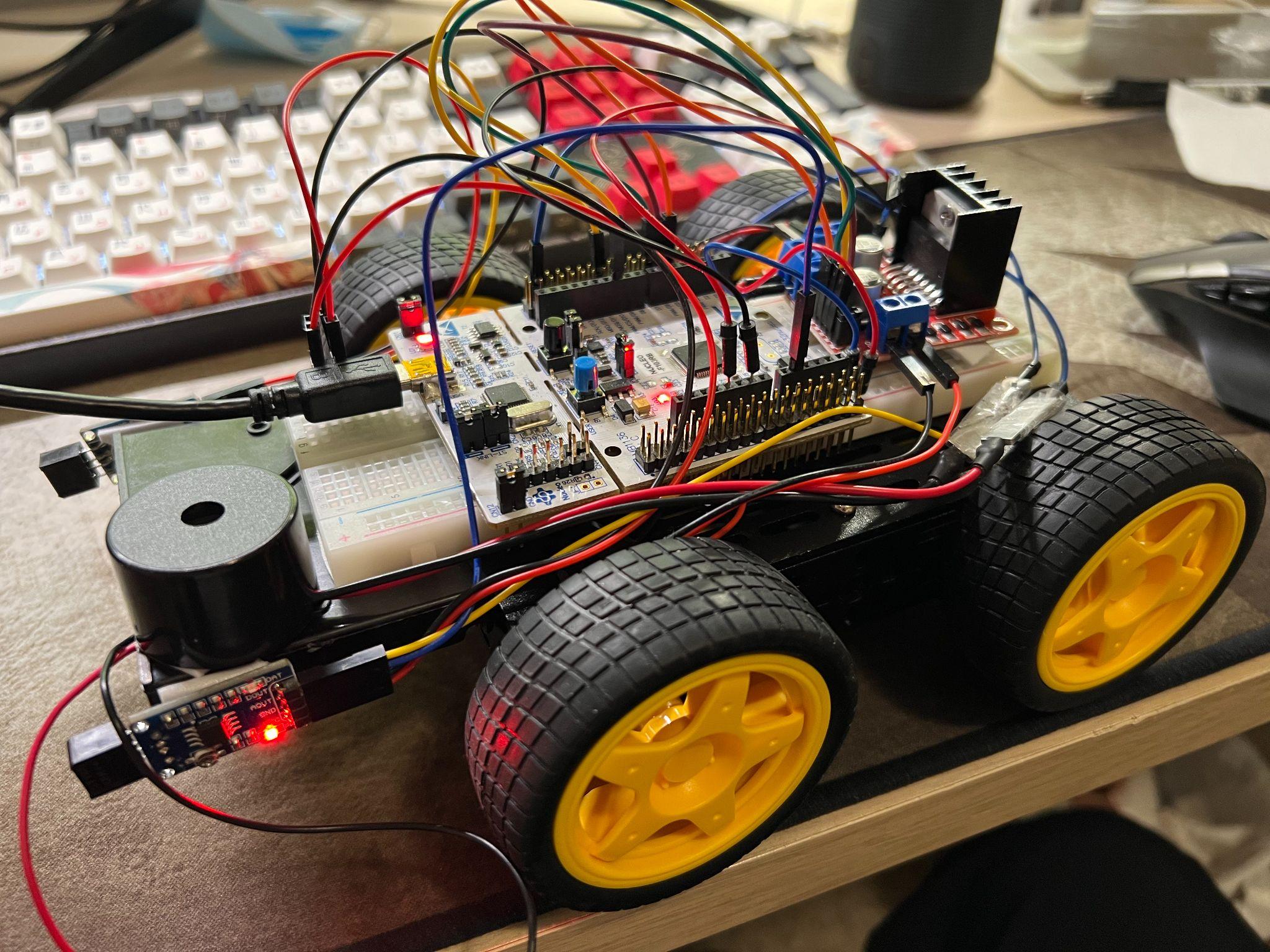
# 

# Yuting Li

# 200434800

# ENEL 351 Project Report

Cyber Truck



1. Project Introduction & Overview

1.0 Design process

I decided to build a thing that has inputs and outputs, but I have no idea what to build exactly. So, I read the last year’s requirements and decide to build something similar, which is a robot car. I used two infrared sensors, one as digital input and one as an analog input, they are the eyes of the robot car. I used the motor driver as one of my outputs so that I can manipulate the car and make some meaningful movements, I used a buzzer as another output so that when the robot car cannot handle it, it will beep for help.

1.1.1 Descriptions of the project’s tasks and functions

The robot car is a self-driven car that is programmed by the STM32F103RB microcontroller board. It is designed to tackle different driving situations and be able to detect and avoid obstacles. This robot car is an FWD (Front-wheel-drive) four-wheel car. It is controlled and driven by the front two wheels, by adjusting the forward/reverse of each of the front wheels, the car can drive forward, turn left, turn right stop and even reverse. There are a total of two infrared detectors used by this car. Each infrared sensor can be configured as digital or analog input. They are responsible for detecting obstacles in the front-left and the front-right. Once the robot car has detected an upcoming obstacle, it will detect whether the opposite direction has any existing obstacle, if not, then the car will manoeuvre itself by turning in the other direction. However, if the way has been totally blocked and the robot car failed to provide a solution, it will buzz to announce that there is no way that it can drive through.

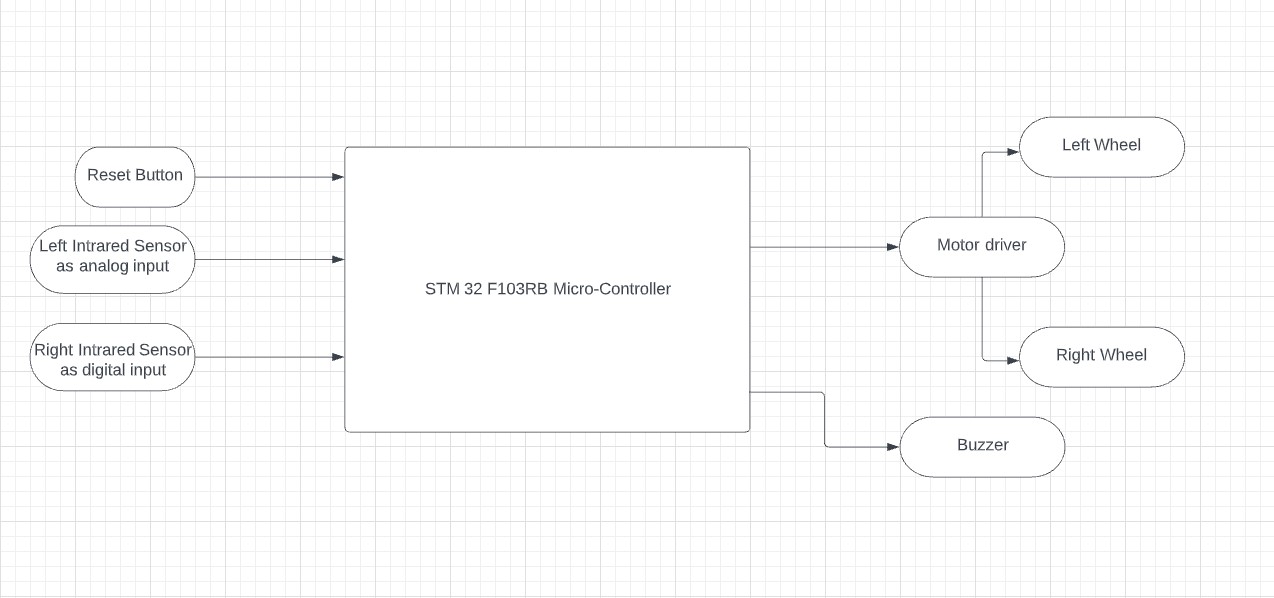
1.1.2 Other minor functions

The reset button on the microcontroller will also be used as a start button, after the code is loaded, the user has to press the reset button for the car to move. Also, if some unexpected bug happened, the user could also press this button to reset the car.

1.1.3 User Controls and Indicators

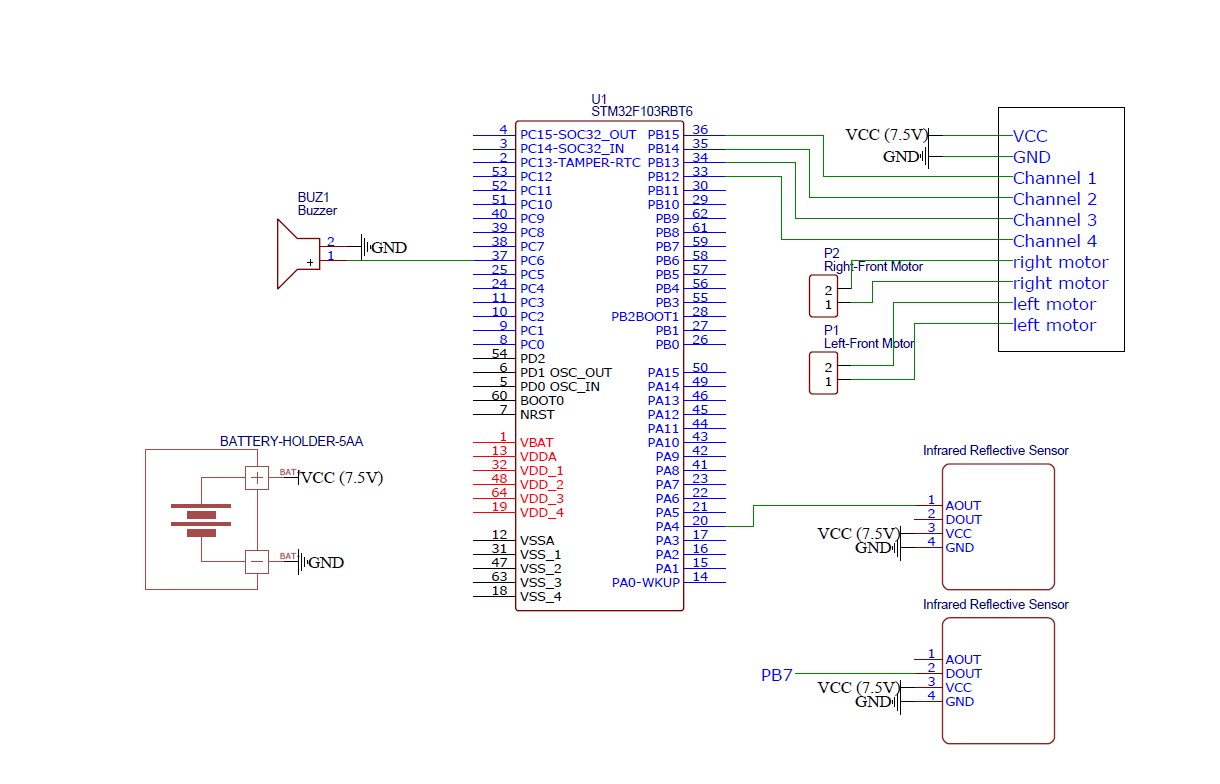
All behaviours will be directly programmed into the microcontroller, and the robot will be self-driven. The user will not be able to directly control this car at the runtime. However, the behaviours of the car will be able to be modified by the codes. For example, its driving speed.

1.2 Block Diagram



1.4.1 Connection details for all I/O components

The detailed I/O connections are shown in the diagram below:



1.4.2 All Electrical Components Properties

|  | Input Voltage | Working Frequency | Supply/Output Voltage | Working Current |
| --- | --- | --- | --- | --- |
| LEDs | 3.3V |  |  |  |
| LCD | 2.5~6V |  |  |  |
| Battery Holder |  |  | 1.5~7.5V (Depending on the # of battery cells) | About 1mA |
| Motor | 3~6V |  |  | 200~400mA |
| Buzzer | No datasheet found, need to be tested | | | |
| Ultrasonic Sensors | 5V | 40Hz | Digital/Analog Output | 15mA |
| Infrared Range  Sensors | 3.0~5.3V |  | Digital/Analog Output |  |
| STM32F103RB | 5V |  | 3.3V/5V | 300mA |

1. Testing Strategies

2.1 Battery holder Testing (5 batteries, 7.5V)

I tested the battery by connecting it to my motor driver which can handle voltage between 5 ~ 12V and it worked.

2.2 Portable Charger Testing (Very steady 5V)

I loaded my code to my board from my PC, and then disconnected it from the PC and plugged the cable into my portable charger instead, and the board is still being able to run without any problem.

2.1 Motor Testing

The motors used for this project each have two lines, positive and negative. I connected the positive to the VCC and the negative to the ground, the motor driving forward. On the contrary, if I connect the negative to the VCC and the positive to the ground, the motor will reverse. This tells me that it is very difficult to make one wheel drive forward and another reverse at the same time, and therefore I would need a motor driving for precise control.

2.2 Motor Driver Testing

For the motor driver, I used two pins for the power supply, the negative to the ground and the positive to my 7.5V power source.

2.3 IR Sensor Testing - Digital

I tested the digital output by setting up the appropriate subsystems and GPIOs, and then I moved an object close to the sensor, the red LED on it is on and the digital input becomes 1, when I removed the object, the red LED is off and the digital input becomes 0.

2.4 IR Sensor Testing - Analog

I tested the analog output by first modifying my lab codes from the ADC lab and making the input pin PA1. Then I compared this input with multiple values and confirmed that its triggering voltage is (3.8 \* 0.8)V.

2.5 Buzzer Testing

I tested the buzzer by simply connecting the positive to the VCC and negative to the group and it will beep. When it is disconnected, it will not beep.

1. Modifications throughout project building

3.0 Original Function Specifications

1. Using batteries to supply the board
2. Not using motor driver
3. Using ultrasonic sensors
4. Using LCD to display driving information
5. Using LEDs as robot car’s turning signal

3.1 Using a portable charger instead of using batteries to supply the board

* Firstly of all, each AA battery supplies 1.5V therefore if I use 3, the total voltage will be 4.5V and if I use 4 the total voltage will be 6V. I want to supply my board with a steady 5V and using AA batteries without using a regulator is impossible.
* Secondly, it is very convenient to use a portable charger, so that I can simply connect the cable to the port instead of connecting VCC and ground by using pins

3.2 Using a motor driver instead of directly manipulating motors

I decided to use a motor driver because if I want to make a wheel reverse, I must supply 0V to its positive and a positive voltage to its negative, this is almost impossible let alone making one wheel forward and the other one reverse at the same time. But a motor driver can do this job perfectly and all I have to do is just load the bits onto the corresponding driver manipulation pins.

3.3 Using the IR sensor’s analog input instead of using an ultrasonic sensor

* I decided to use my left IR sensor as an analog input because an ultrasonic sensor is a bit too complex for me, it might take a much longer time and it might fail. I want to deliver my functioning project on time. Therefore I decided to go with the easiest option.
* Also, an IR sensor’s analog input can do just as good as an ultrasonic sensor in my project.

3.4 Not using an LCD display

* I decided not to use an LCD display because I am not good at USART/UART and I already have two outputs so I abandoned this feature.

3.5 Not using LEDs

* Using LEDs as turning signal lights is quite unnecessary because it is too easy to implement and cannot demonstrate anything that I have learned from this course.

4. Suggestions for Improvements

1. The motors are driving too fast, it would be better to control the wheels by setting duty cycles. I wrote some functions of driving the car using PWM but I failed, those functions are kept in PWM.c
2. The ideal way of handling obstacles in both directions could be 1. Reverse enough distance 2. Turning to either left/right 3. Repeat the two steps. I think this might be feasible for some small obstacles but might not work well for big obstacles but it is definitely something worth trying.