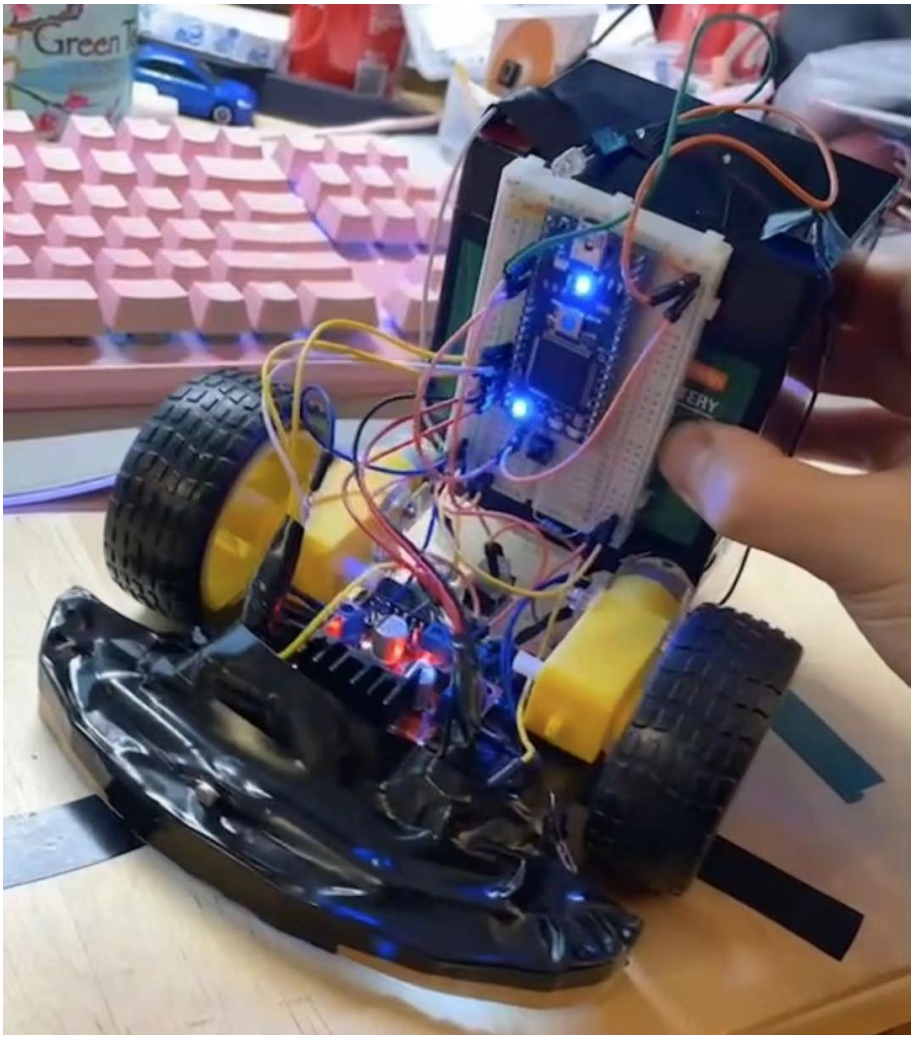
The Smart Car

Wenbo Li

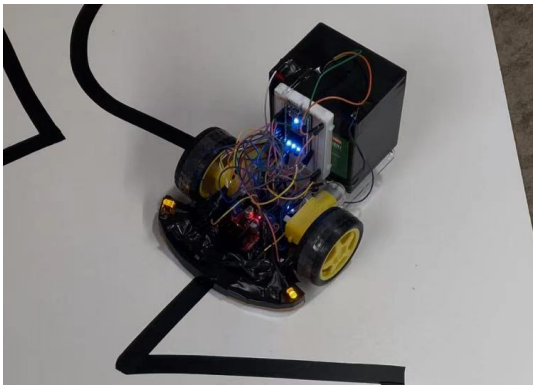
November 2020



Contents

1 Introduction

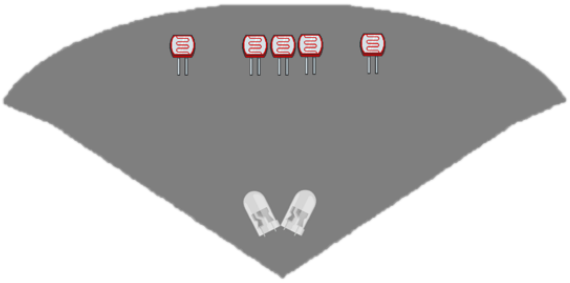
The Smart Car can trace and follow the line on ground to go straight, turns or even U-turns. It is based on Arm LPC-1768 micro-controller and photoresistors.



2 Key Components

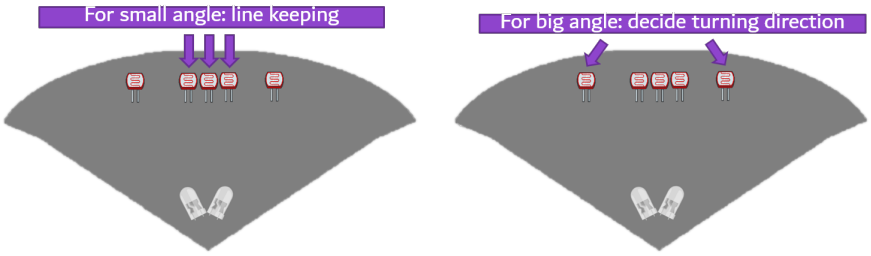
2.1 Photoresistors

Here is a diagram showing the bottom of the car. There are 5 photoresistors who play the main role in the algorithm.



The triangle area is fended by paperboard, and there are 2 LEDs inside, lighting up the area. This aims to block the light from the outside environment since the readings can be easily disturbed by the outside environment, comparing to IR sensors. This is a disadvantage for photoresistors. Photoresistors are designed to sensor the light, not to track a line. So I have to make the readings stable by doing this easy modification. The result came out fine. See section 4 for details.

The inside 3 photoresistors do the line keeping for small angle curves. The width of these 3 photoresistors equal to the width of the black line. The outside 2 photoresistors decide the turning direction for sharp curves. When the photoresistor in the middle lost the line, outside 2 are triggered.

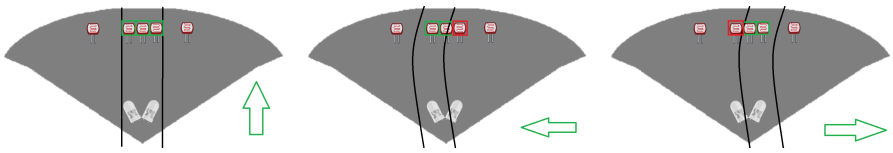


1. For the inside 3 photoresistors

   (a) If all of them detect positive, the car goes straight.

   (b) If the left two detect but the right one doesn't, the car goes left.

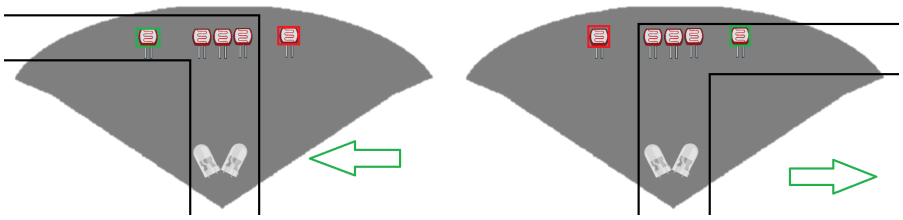
   (c) If the right two detect but the left one doesn't, the car goes right.



2. For the outside 2 photoresistors

   (a) If the left one detects but the right one doesn't, the car rotates left.

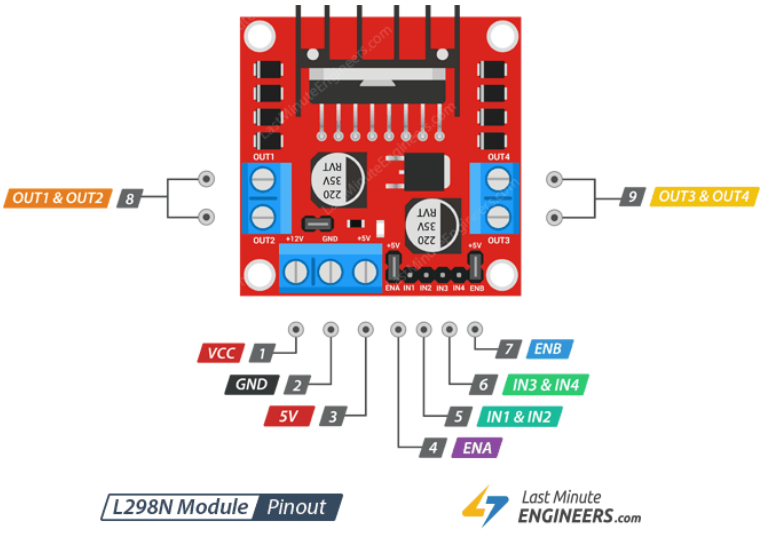
   (b) If the right one detects but the left one doesn't, the car rotates right.



2.2 Battery

A LiFePO4 Battery is chosen to provide 12v to the motor driver chip. The main reason I went for this is the low price. I would need a lot of replacement batteries if I went for AAA batteries. And that would cost a lot, comparing to this rechargeable one which only took about $20. It can store 0.1kWh energy thus is more than enough for the car to run for several miles.

2.3 Motor Driver



[1]

L298N takes 12V from the battery as input and gives a 5V output which can be used to power the LPC-1768. IN1 to IN4 pins determine the running direction of the motors, ENA and ENB pins give PWM control for speed. This chip satisfies my need perfectly.

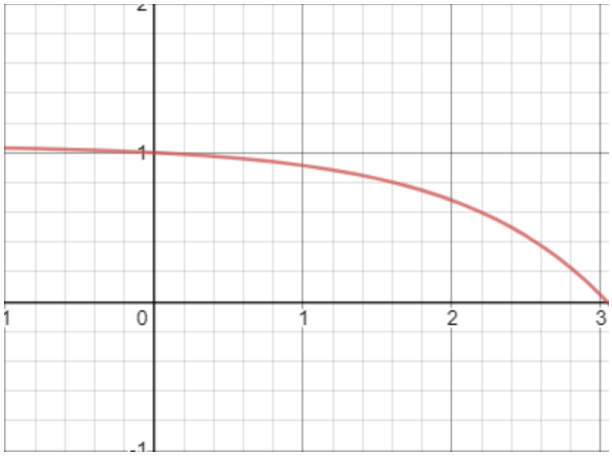
3 Difficulties

Since I want to make this project challenging, I didn't choose any trade-offs but hardcore. An easy project won't be fun and enhance my skill. Thus, I went for photoresistors but not IR sensors, ordinary DC motors but not stepper motors, etc. I met some difficulties, and here's how I solve them. Note that this is a good practicing way for academic purposes only. For industrial level designing, you might want to go as easy as you can while maintaining a cheap cost.

3.1 Acceleration Pattern

I made a design flaw by putting the heavy battery at the back while putting the driving motors in the front. This results in a bad acceleration pattern. The car is hard to accelerate from rest and is also hard to brake from rotating or sliding. It is also very likely to drift due to the heavy weight at the back. To solve this, I gave a non-linear voltage to the motor instead of a constant linear PWM signal using the following equation,

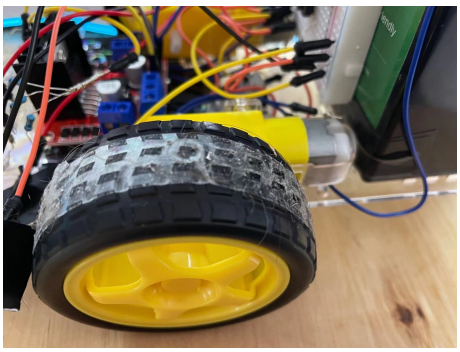
$$\texttt{PWM\%} = \frac{A}{-e^{-B \times t}} + C$$



After tweaking parameters, I got an instant acceleration and smooth brake. I could have used the micro:Bit chip as an accelerometer to deliver an accurate voltage. However, using this equation makes it smooth enough to complete the track. So I didn't put the micro:Bit on.

3.2 Traction Control

However, the tires that came with the car chassis are so cheap which can deliver almost zero traction while moving. To solve this, I wrapped them with double-sided tape, and they stick to the ground and deliver great traction like a drag race car. This tape is originally designed for sticking the window isolation film. I found it pretty durable since it will remain sticky even with dust on it.



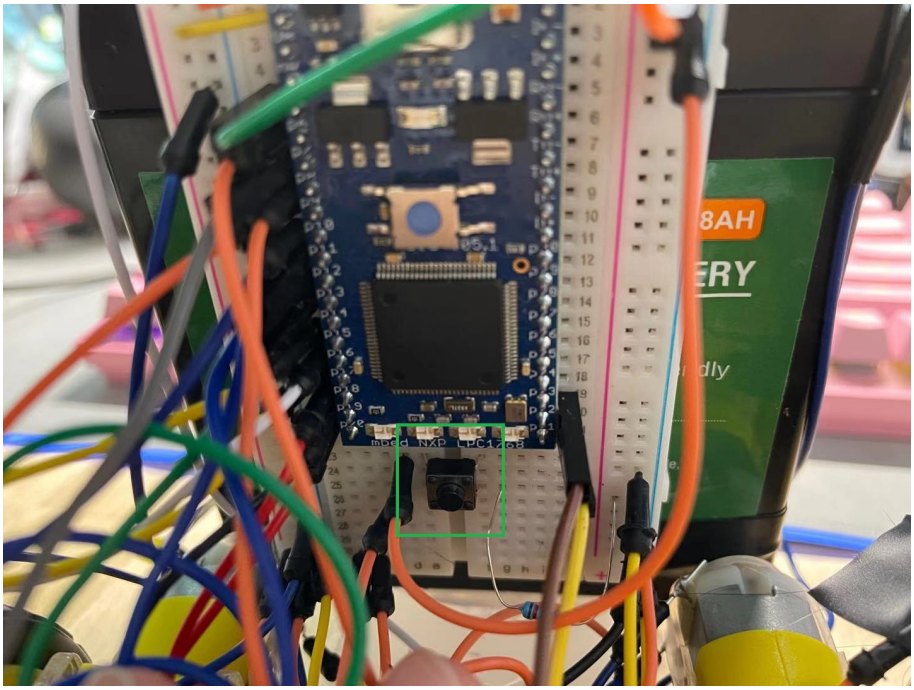
3.3 Dynamic Calibration

The readings from photoresistors are not 1 and 0, but are float numbers representing lumen. I have to manually convert them into digital output. I did this by recording the minimum and maximum reading of each sensor and then give digital result by checking

$$\texttt{current percent} = \frac{\texttt{current lumen}}{\texttt{maximum lumen} - \texttt{minimum lumen}}$$

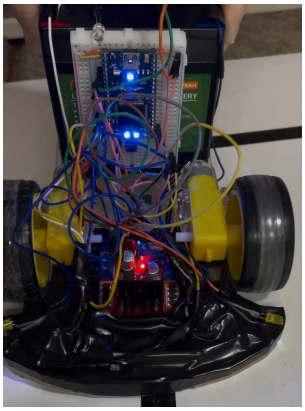
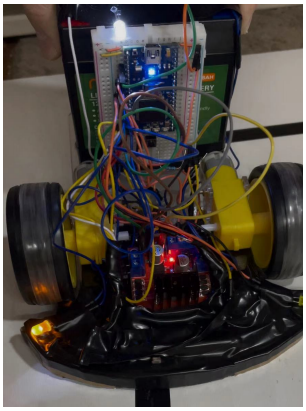
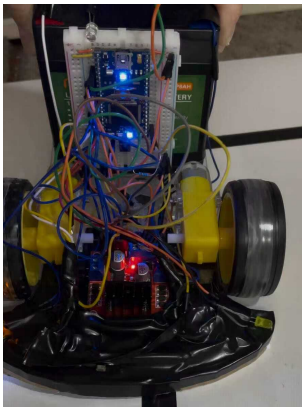
if the current percent is over 50%, it is considered a 0, which means the sensor is not at the above of the black line. Otherwise, it is 1, which means a black line is detected. However, even with the fender around the photoresistors. The readings are still significantly different from diffident backgrounds. Different material leads to different refractive index. I have to recalibrate the sensors when testing on different surfaces. To avoid manually typing in the data, I make a calibration mode. To activate calibration mode, just click on the button, and the motors will stop. Now put the car on the surface you are testing with and wait for 2 seconds. Then the new minimum and maximum reading of each

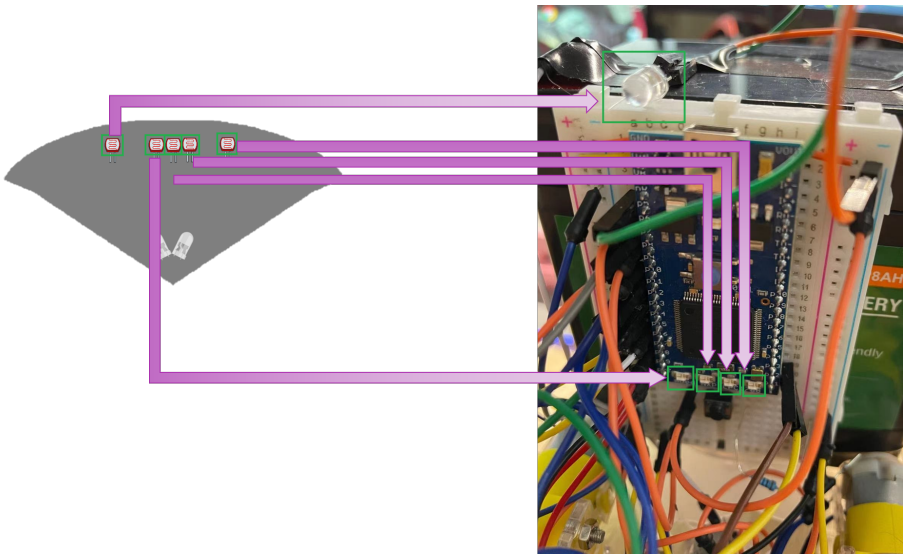
sensor are stored as a text file and can be retrieved by the code. And the car is good to go.



3.4 Debugging

Detecting lines is the most important part of this project so I want to monitor it all the time. I use 4 LEDs on board and one extra LED to representing if those 5 sensors detect anything.



4 Source Code and Demo

See the effect of photoresistors detecting lines at *Photoresistors detect lines*[2].
See the demo run at *The Smart Car Demo Run*[3].
Review the presentation on Dec. 1st at *The Smart Car Presentation & Demo*[4].
See the source code at *The-Smart-Car*[5].

References

[1] Credit to LastMinuteEngineers.com at,
https://lastminuteengineers.com/l298n-dc-stepper-driver-arduino-tutorial/

[2] See the effect of photoresistors detecting lines at
https://youtu.be/2OX1IBFHP-k

[3] See the demo run at,
https://youtu.be/IT1uIOLXRqQ

[4] Review the presentation on Dec. 1st at,
https://youtu.be/E2LsvKXMEi8

[5] See the source code at,
https://github.com/lwbhahahaha/The-Smart-Car