ECE 3 FINAL PROJECT REPORT

Yunhui Ma, Huimiao Zhong

1) Introduction and Background

The purpose of the project is to design a motor-driven and line following car which utilizes a set of LEDs and phototransistors to detect the black track given on a white flat surface. The car is able to turn right, turn left and stop. It should adjust back on track automatically when it deviates. The car is capable of stopping at the final destination where there are two parallel black lines which are perpendicular to the track. The phototransistors and LEDs work in a pair in order to detect the black track on the surface. The resistance of the phototransistor changes when the brightness of light which it detects changes. The black track reflects less light than white surface so that there will be a considerable resistance difference between the value that car is running on the track and the value that car deviating to the white area.

There are several implementation schemes. In our project, we choose to build the car leveraging PID controller(proportional-integral-derivative controller). The car is consist of components including the Arduino Nano, TIP120, 6V battery, 9V battery, gear motors, LEDs(red, blue and green) and IR LEDs and IR receivers and a bunch of jumping wires.

The car can be divided into sensor section, control section, driver section. Sensor section includes IR LEDs and IR phototransistor. Control section includes batteries, Arduino Nano. Driver section includes two motors attached to wheels. TIP120 transistor helps to optimize the motors by Arduino Nano sending PWM signal. We designed and assembled the car according to the given circuit diagram. We connected the hardware. And we are able to utilizing the data to code in order to make the car has the ability to follow the track.

2) Testing Methodology:

a. Test Designs

In order to follow the black track when the car is moving, sensing devices are required to detect the track so that the mobile car can be able to turn by itself. The sensing device is designed by an IR LED and a Phototransistor which are in parallel in the circuit. **(Figure 1)**

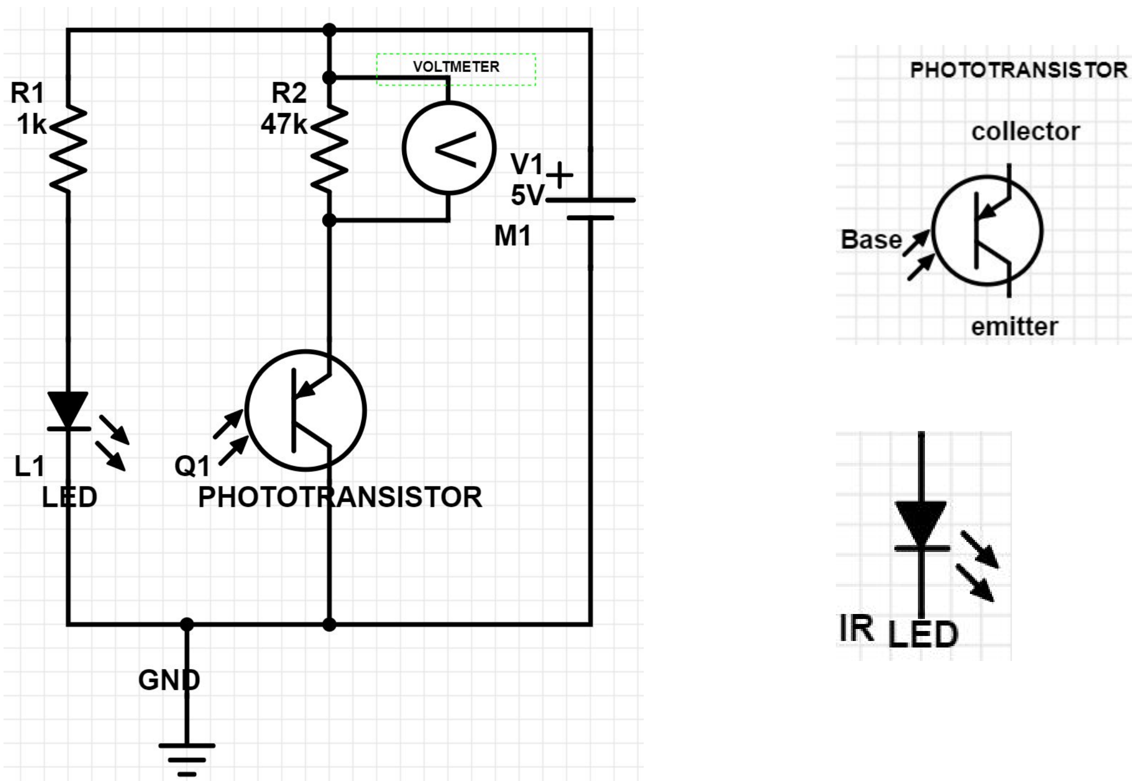


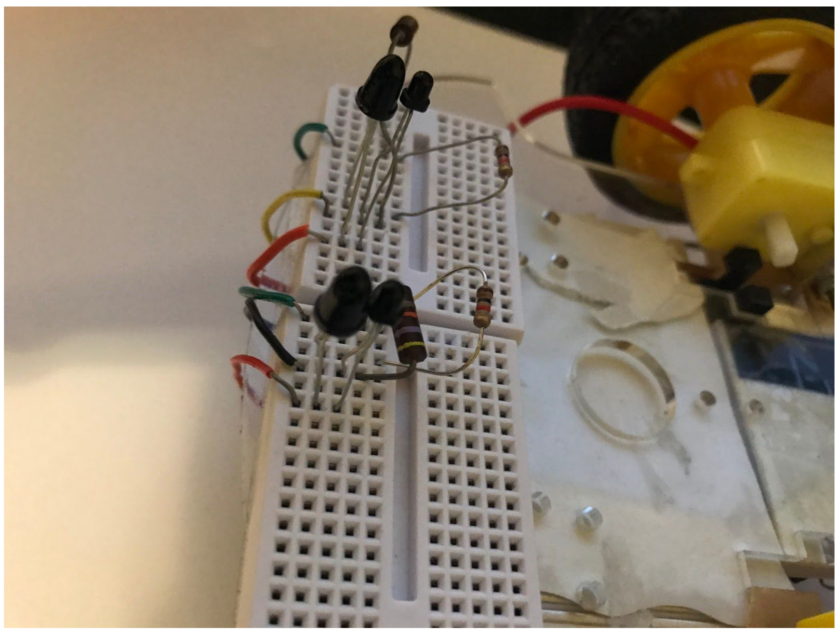
Figure 1: Path Detecting Sensor Schematic

The principle of the IR phototransistor has an exposed base which is sensitive to light. When light which is emitted by the IR LED strikes on the base, the phototransistor allows current to flow from the collector to the emitter. Since there is current across the resistor R2, the voltmeter can be able to read the changing of voltage from R2. If the IR LED light doesn't emit to the phototransistor, the phototransistor doesn't allow current to flow through the circuit. The voltmeter reads a zero voltage. Base on this, when a black object blocks the front of the IR LED, the light emitted by this IR LED is absorbed by the black object. The phototransistor cannot be able to receive any light from the IR LED. Therefore, no current is allowed to go through the resistor R2, and the voltmeter read 0V. On the other hand, when a white object blocks the front of the IR LED, the light reflects. The phototransistor receives the light and allows the current to go through the circuit, and the voltmeter can read the voltage from R2. The distance between the IR LED and phototransistor needs to be determined by the data because the phototransistor is very sensitive to light. If they were very close or far away to each other, the phototransistor would always receive light. It leads to inaccurate data.

This phototransistor system is installed under the mobile car, and it points down to the floor (**Figure 2**). Installing two of these systems on the right and left sides of the car is required to detect the black track coming from the right and left side.

IR LED

Phototransistor



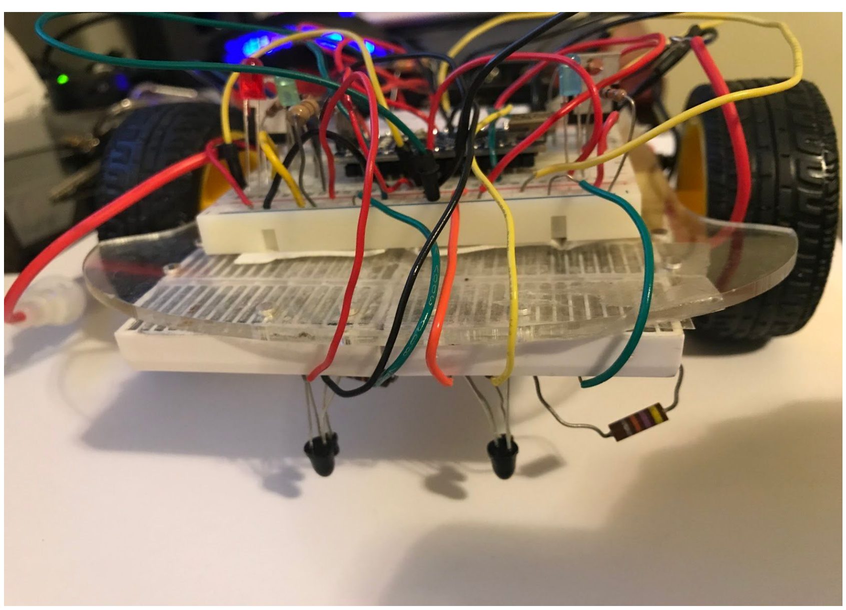


Figure 2: Path Detecting Sensor System

The motor system includes a DC motor, TIP120 NPN transistor, 6V battery and PWM signal source which is provided by the Arduino. In order to make the mobile car turn with the same voltage, the speed in either wheel should be different. By using the TIP120 NPN transistor, it allows the duty cycle of the PWM signal to control the motor speed. If the PWM signal (duty cycle) increases, the motor speed will increase. However, if the duty cycle sets to be zero, the motor would stop. The mobile car installs the TIP120 transistor for each motor which is on the left and right. When the system sends a lower duty cycle of the PWM signal to the right side of the transistor, the mobile car turns right. Conversely, sending the lower duty cycle of the PWM signal to the left side make the mobile car turn left. Sending the same PWM signal to both transistors would get the same speed for both motors. Base on the test, the motor on the left side of the mobile car is not installed straightly. The car would go to left instead of going straight when both motors run at the same speed. Therefore, to correct the error, the left transistor would have a high duty cycle PWM signal compared to the right transistor.

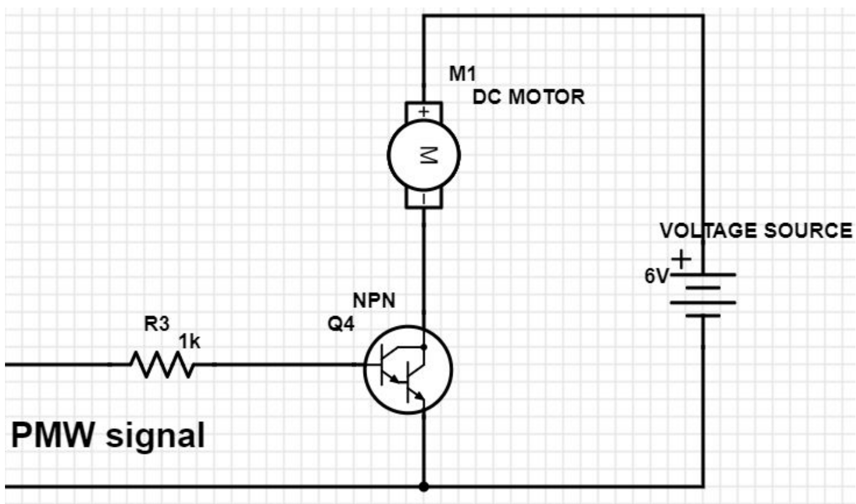


Figure 3: Gear Motor Schematic

The Red, Green, Blue LED system is used to show the behavior of the mobile car. It is controlled by the Arduino pin output mode individually. When the car goes straight, the green LED will light. The red and blue LED lights correspond to the right and left turn of the car respectively.

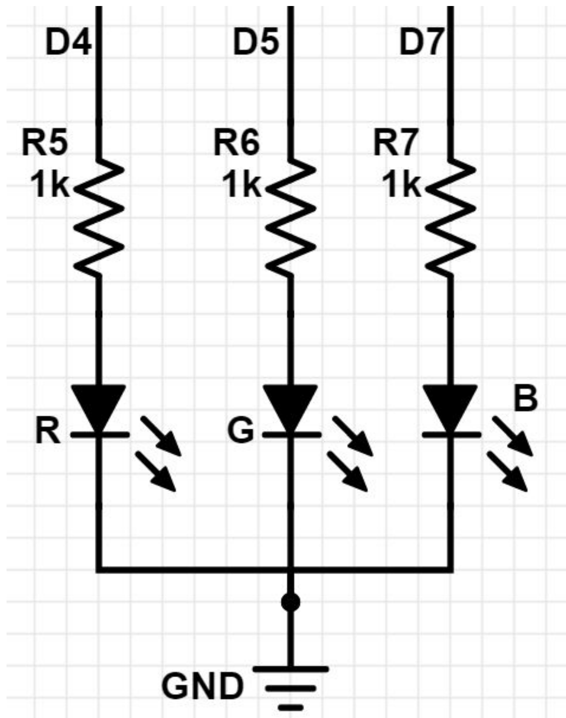


Figure 4: RGB LED Schematic

The Arduino NANO board is used to control all of the systems in the mobile car. The pin A1 and A3 can be used to get the voltage reading from resistors R2 and R9 which are in series with the phototransistor. Pin D3 and D10 can be able to send the PWM signal to the TIP 120 transistors in order to control the motor speed. Pin D4, D5, D7 are used to control the RGB LED light. In addition, the mobile car needs two sets of batteries. The 6V battery provides the power source to the motors, and the 9V battery provides the power source to the Arduino NANO. By using two separated batteries, it can allow the Arduino NANO, motors to have enough power and work effectively.

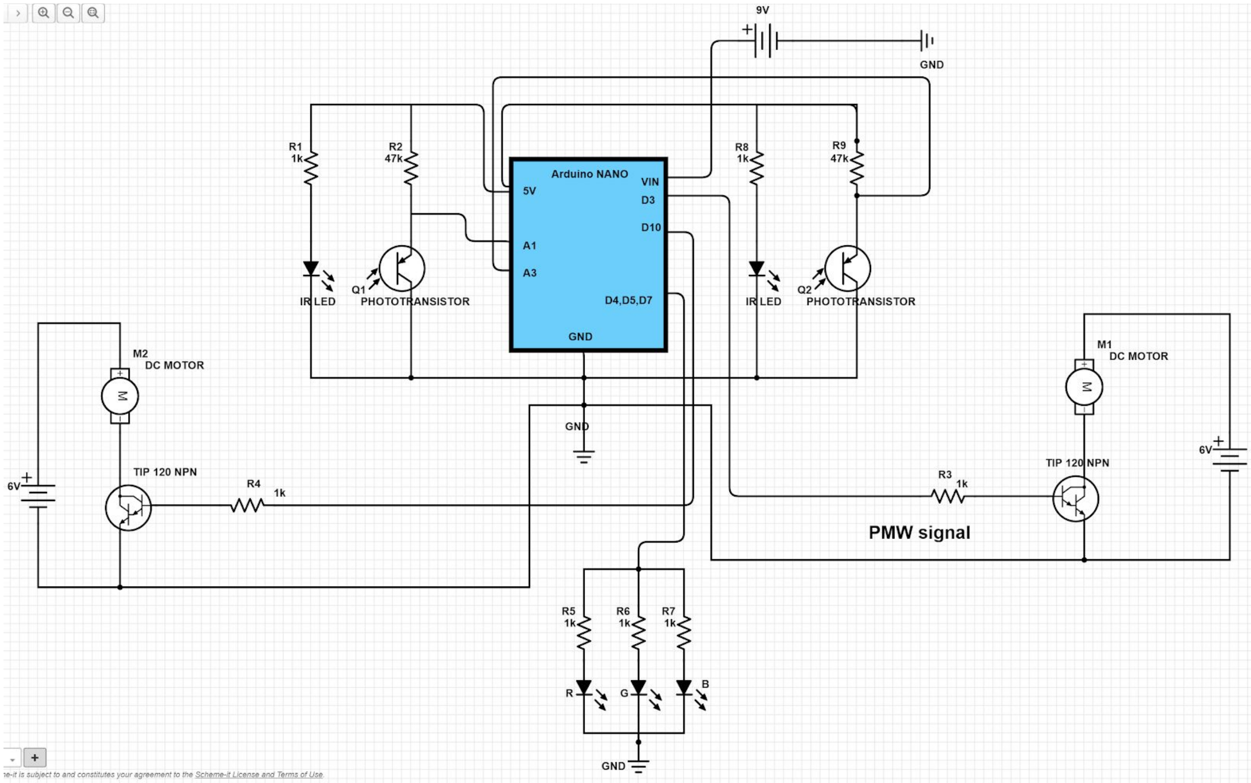


Figure 5: Main Circuit Schematic

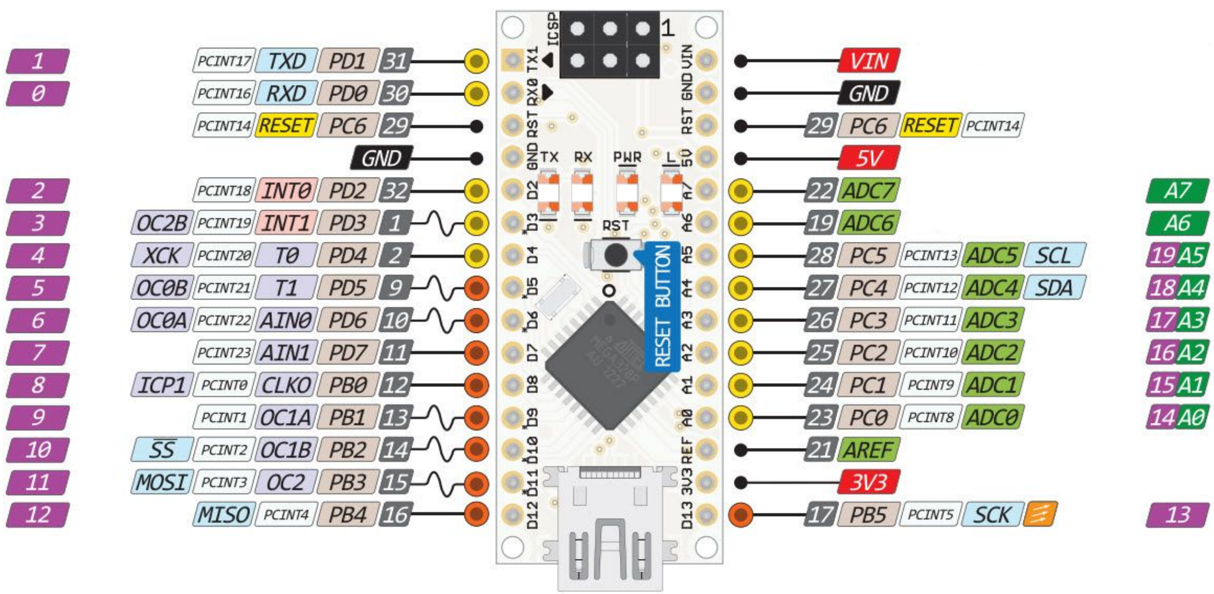


Figure 6: Arduino Nano Pinout Schematic

b. How the tests were conducted

In order to test each Path Detecting Sensor system, we place the sensor system above the black track and white paper. By using **analogRead()** method, the NANO can read and print out the voltage data to the computer screen. [1]

First, we test the both sensors above the white paper. Second, we test the left sensor above the black track while the right sensor is above the white. Third, we test the right sensor above the black track while the left sensor is above the black track.

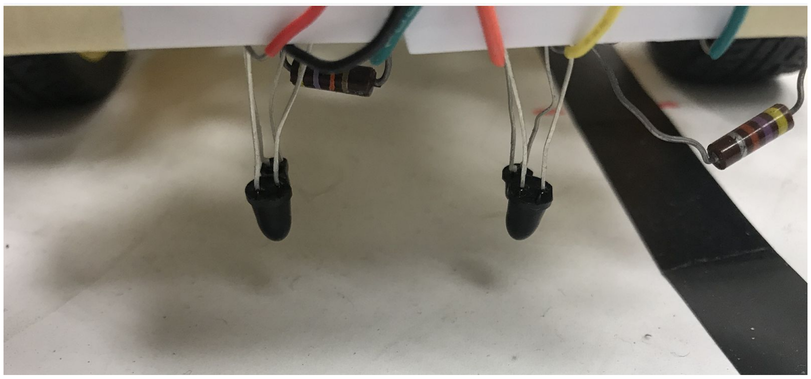


Figure 7.1: Testing Sensor above white

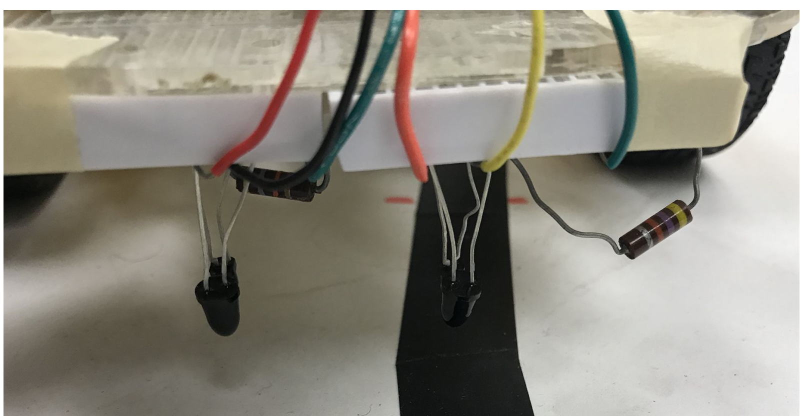


Figure 7.2: Testing Left Sensor above black

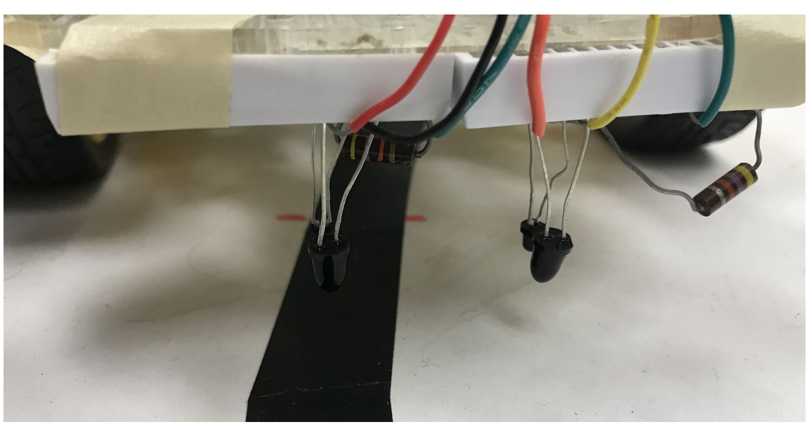


Figure 7.3: Testing Right Sensor above black

In the beginning, the data received from both sensors is always small no matter how white the paper color is. We check the wires connection, but it is connected correctly. Although the data value readings are still different between black and white, their difference is not obvious, which is about 100. It's difficult for the sensor to distinguish whether the sensor is above the white paper or black track. However, we change the resistor's value from 1k to 47k. The resistors that we change are in series with the phototransistors. Therefore, although a small current goes through the circuit, the Arduino can read a large voltage from the 47k resistors. With a bigger resistor, the data value readings become more obvious. When the sensor system is above the white paper, the reading is around 950. While the sensor system points to the black track, the reading decreases to about 500. This is good enough to use to detect the black track.

After many tests, we determine the distance between the IR LED and the phototransistor which is about 0.8 cm, which would give obvious data value readings between white and black.

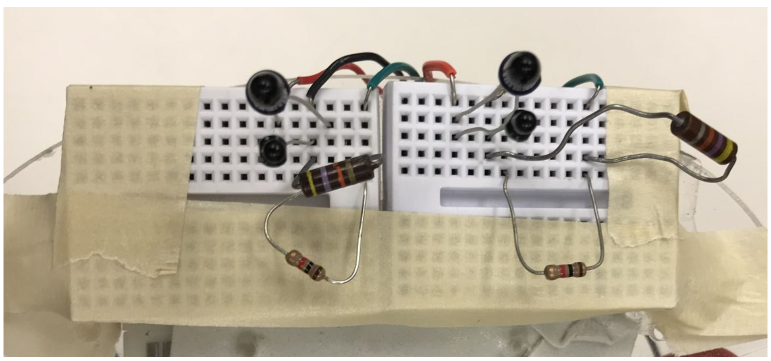


Figure 8: Distance between IR LED & Phototransistor

c. Data Analysis

First, we test both sensors on the white paper, and the sensors readings are shown in **Table 1**. Second, we move the left sensor slowly and approach to the black track. We stop until the left sensor is above the middle of the black track (**Table 2**). Then, we do the same thing for the right sensor, which also gives a normal reading. The data readings decrease as the sensors moving deeper into the black track.

time	Paper color	Left Sensor	Paper color	Right Sensor
1	white	953	white	947
2	white	953	white	947
3	white	955	white	948

Table 1: Data Readings from both IR sensors above white

time	Paper color	Left Sensor	Paper color	Right Sensor
1	black	849	white	960
2	black	775	white	960
3	black	726	white	960
4	black	621	white	960
5	black	602	white	960
6	blakc	571	white	960

Table 2: Data readings for left sensor begins and enters to the black track

time	Paper color	Left Sensor	Paper color	Right Sensor
1	white	948	black	852
2	white	948	black	718
3	white	948	black	669
4	white	949	black	596
5	white	949	black	578

6	white	949	black	561

Table 3: Data readings for right sensor begins and enters to the black

Figure 9 shows a graph (data readings vs time) of IR Sensor. When the left sensor first moves into the black track, the readings begin to decrease to minimum as the sensor approaching the middle of black track. As the sensor getting away the middle of the black, the readings begin to increase. It goes back to a maximum which is above the white paper.

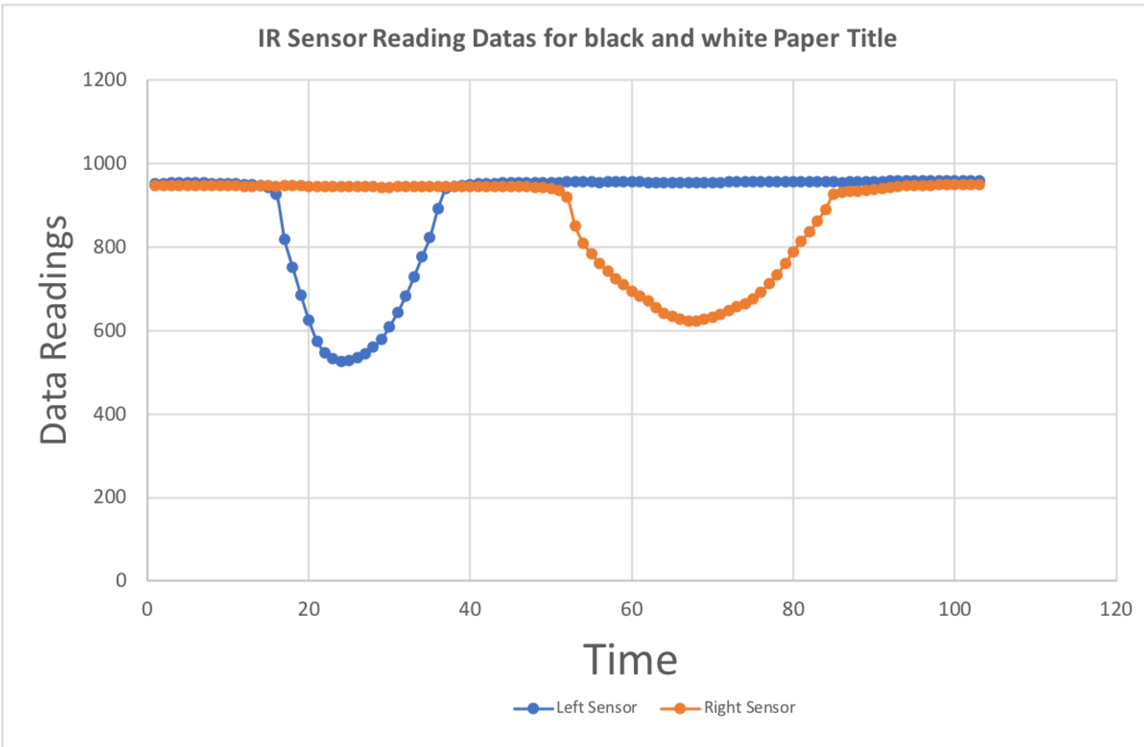


Figure 9: Data for both IR Sensors across the black track

Both IR sensors work perfectly as they are moving from the white paper to black track. Errors would be produced by the difference between their readings from black and white paper. By using those errors, they can determine the speed of both motors to make the mobile car turn.

d. Test Data Interpretation

Base on the reading data, we know that the reading values are different between white and black. We call the difference is an error. By using this error, we control the speed of both motors in order to go straight and turn. [2]

Introduction of PID method:

First, we initialize a **pErrorLeft** which is the subtraction between **set_leftSensor** and **sensorLeft.** The **set_leftSensor** is the data value when the left sensor detects the white paper. The **sensorLeft** is the actual value that the left sensor detects. In the beginning, we set the **set_leftSensor** to be a constant value which is 980. However, we find out that sometimes the sensor cannot be able to get 980 when it detects the white color. The sensor reads 890 on the racing track, while it reads 950 on the lab table. Therefore, we store a maximum value in the **set_leftSensor**. The maximum value is from the actual value when the sensor detects the white color. (**Figure 10**)

```
if(set_leftSensor > sensorLeft)
set_leftSensor = set_leftSensor;
else if(set_leftSensor < sensorLeft)
set_leftSensor = sensorLeft;
```

Figure 10: Set the maximum to set_leftSensor

Second, we subtract the present **pErrorLeft** from the previous **pErrorLeft**, and get the **dErrorLeft**. Then, we add **kp*pErrorLeft** and **kd*dErrorLeft** to get a **totalErrorLeft**. **Kp** and **kd** are constant. On the right sensor, it also does the same thing.

```
int initialError = pErrorLeft;
pErrorLeft = set_leftSensor - sensorLeft;
dErrorLeft = pErrorLeft - initialError;
totalErrorLeft = kp*pErrorLeft + kd*dErrorLeft;

int initialError_right = pErrorRight;
pErrorRight = set_rightSensor - sensorRight;
dErrorRight = pErrorRight - initialError_right;
totalErrorRight = kp*pErrorRight + kd*dErrorRight;
```

Figure 11: PID method on both sensor

When both sensors detect the white color, meaning that the car is going straight. The **totalError** is small. We set both motors at high speed. In order to send the PWM signal to control the motor speed, the **analogWrite(pin, value)** method can be able to send the PWM signal to the transistor by setting different value. The maximum value is 255, and the minimum value is 0. Sending 255 allows the motor to have a high speed. On our project, the car cannot be able to turn if the motor speed goes too fast. Therefore, when the car goes straight, we set the left motor speed to be 230, and right motor speed to be 200.

Left Sensor	Right Sensor	totalErrorLeft	totalErrorRight	Left speed	Right speed
949	928	0	-1	230	200
949	928	0	0	230	200
949	928	0	0	230	200
949	928	0	0	230	200

Table 4: Car going straight (both sensors detect white paper)

When the left sensor detects the black track, the difference would be produced, and the **totalError** becomes bigger and bigger. Then, the left motor's speed begins to decrease from 150 to 0 as the left sensor gets into the middle of the black track. The car turns left.

Left Sensor	Right Sensor	totalErrorLeft	totalErrorRight	Left speed	Right speed
955	960	77	15	150	200
921	960	274	15	150	200
849	960	672	15	150	200
775	960	1044	15	0	200
726	960	1264	15	0	200

Table 5: Car turning Left (Left sensor detects black track)

We also decrease the right motor speed from 150 to 0 as the **totoalErrorRight** is getting bigger. The car turns right.

Left Sensor	Right Sensor	totalErrorLeft	totalErrorRight	Left speed	Right speed
948	916	5	63	230	150
948	907	5	114	230	150
948	852	5	435	230	150

948	780	5	812	170	0
948	718	5	1112	170	0

Table 6: Car turning Right (Right sensor detects black track)

Third, in order to stop at the finish line, we set a while loop for both motors' speed at 0 when both sensors detect the black track which is the finish line.

3) Results and Discussion:

a. Test Discussion

Since we have the adjustable maximum value for the **set_leftSensor** and **set_rightSensor**, the PID method works well on different track. We test it on two practice tracks, and the car can turn and go straight well. However, because we write a while loop for both motors stopping at the finish line, sometimes both IR sensors detect the black track at the same time. The car stops on the way before approaching the finish line. Therefore, we separate both IR sensors a little farther, and the car works well again on the two practice tracks. A video for the car running on the practice is uploaded on https://www.youtube.com/watch?v=jR7kexYQvhA.

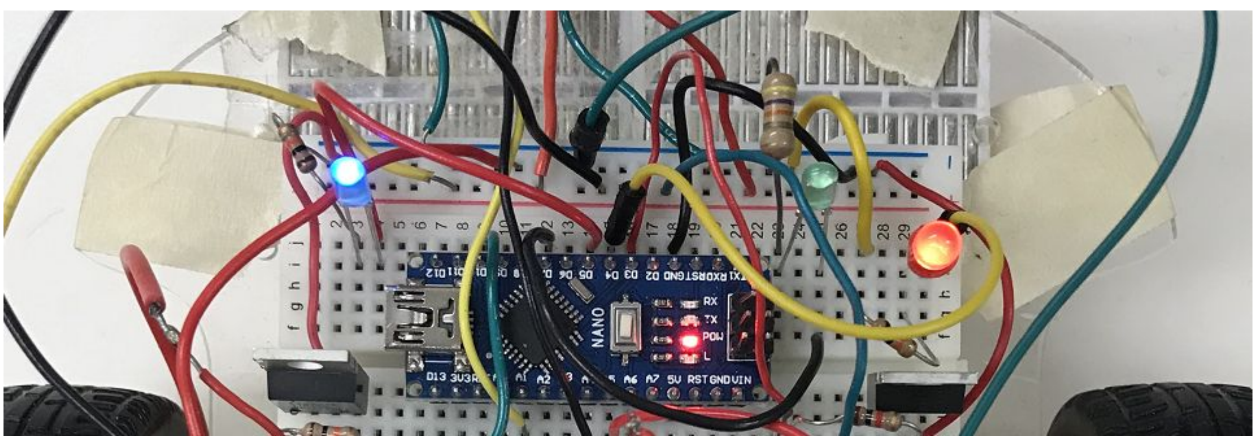


Figure 12: LED System. The system includes blue, green and red LED. Green LED lights for the car going straight. Red LED lights for the car turning right, and Blue LED lights for car turn left. Solid Blue and Red for the car stopping at the finish line.

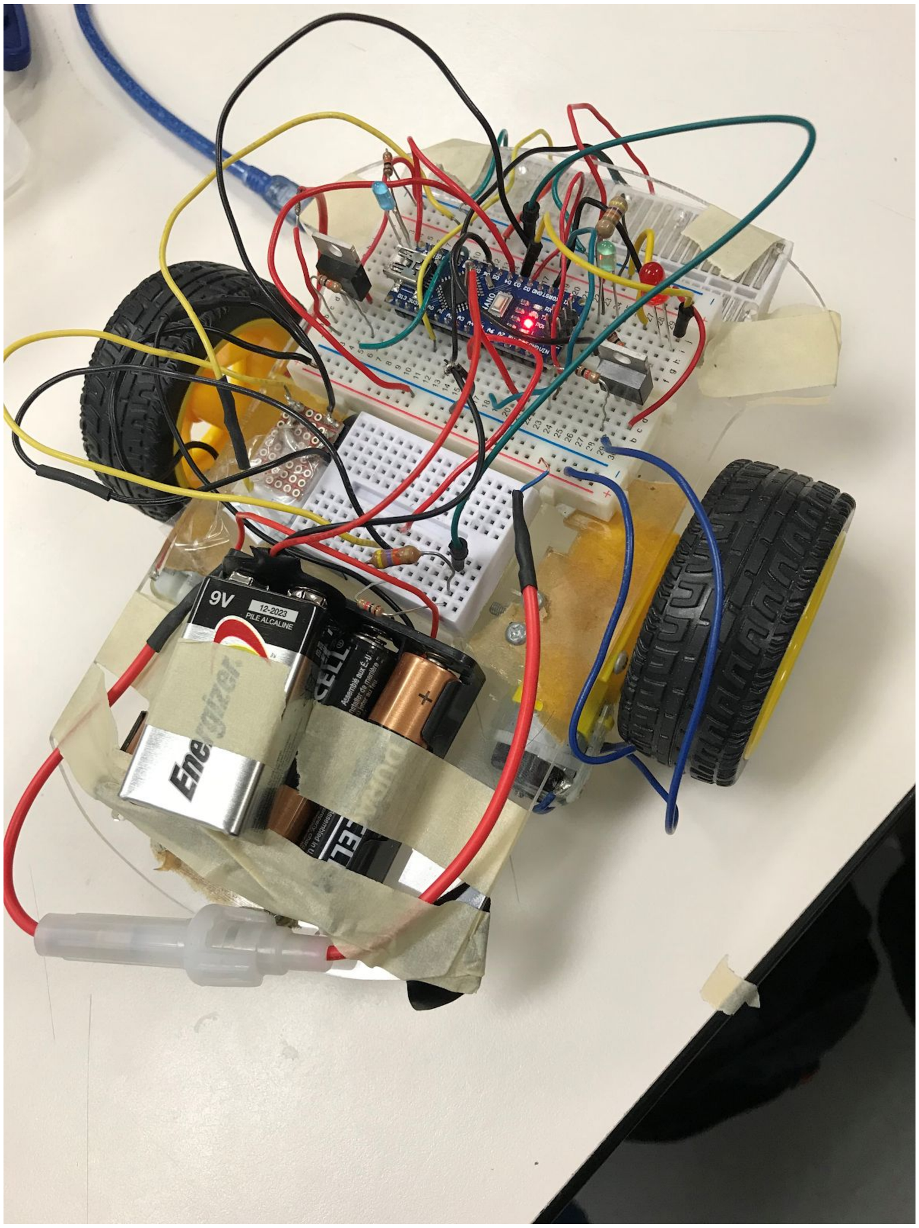


Figure 12: Final product of the Car. Final testing the code and recording data.

b. Race Day Discussion

On Race Day, an accident happens. The car cannot make a right turn at the second corner. The right turning is failed. We test it again on the practice track, but it works well. However, due to the color of the race track which is brighter than the practice track, we adjust the left and right IR sensors farther away from each other. In addition, we slow down the speed of both motors to 210 and 190. Finally, even though the car goes left and right on the straight track, it successfully runs to the finish line in 19 seconds.

For the extra credit, we use a counter to determine whether the car's wheel is running. We calculate the difference between every two values. If the difference is the same, it means that the wheel stops. It can either block or pass the light to the phototransistor. Then, the counter begins to count. As the counter counts to 10, we increase the speed of both motors to 255. After that, we use **delay(1000)** to allow the high speed to run for one second before it goes back to the normal speed.

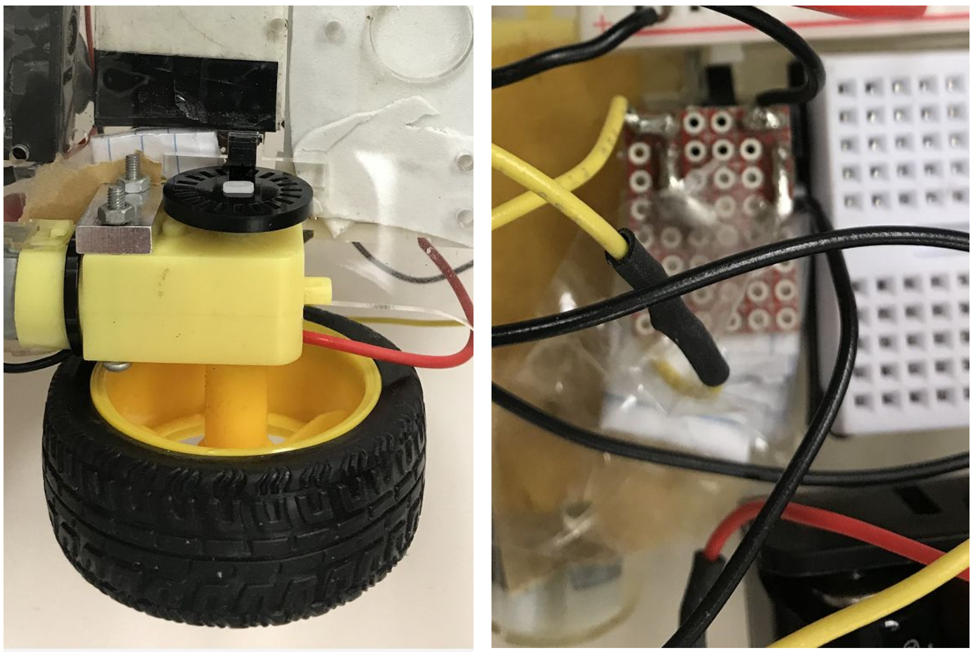


Figure 13: Wheel Speed Sensor. Its principle is the same as the IR Sensor. One of the legs is the IR LED, and another leg is the phototransistor. It gets different value when the wheel is running, while it gets the same value when the wheel stops. The little black wheel is able to block or pass the light to the other side which is the phototransistor.

4) Conclusions and Future Work

This project is designed to build a car which can follow a black line. We learned how to collect data from the controller and interpret the data in order to detect the track. We learned how to assemble the car structure. After finishing assembling the car, we started to write code to control the car. It was difficult at first because we didn't have sufficient knowledge of the function from Arduino. First, we consult mentor and TA, and with the help of Arduino documentation, we finished our first draft code. But the car had trouble when turning. During the stage of adjusting our input and understanding the output data, we learnt how to debug. We had our car finished the track in week 7. Later, we handed on optimized the code in order to make our car ran more smoothly. Generally, we learned the engineering design process(Research, Imagine,Plan, Create, Test, Improve) [4].

For future work, we can ameliorate our car that to make it run on track with different colors, since the amount of light reflect would be different if the color changes. What's more, the knowledge we learned from this project could set the future foundation to build more complicated robots such as cleaning robots and delivery robots.

5) References

[1] Arduino Nano Pin Diagram. https://components101.com/microcontrollers/arduino-nano.

[2] A Conceptual Description of the PID Controller by Dr. Briggs, Dennis

[3] PID Controller Wikipedia https://en.wikipedia.org/wiki/PID_controller

[4] https://www.teachengineering.org/k12engineering/designprocess

Contributor:

Li, Xin

Hayato, Kato