To: Dr. K. Lopin

From: Jillian Awe
Brayden Garcia

RE: Comp Eng 2780 Final Project: Obstacle Avoidance Robo-Car

Date: 15 December 2025

Abstract

A line following Robo-Car was built for the purpose of quickly and consistently completing various tracks. The Robo-Car was designed using 7400 series AND and OR gates and multiplexers to control line detection sensors, going at various speeds and directions depending on what the line detection sensors detect. The Robo-Car was able to complete the beginner track in 14 seconds, the intermediate track in 18 seconds, and the advance track, if slowed down greatly.

Introduction

The Robo-Car was built with the purpose of line following and object detection around three separate tracks, the beginner, intermediate and advanced. The Robo-Car was constructed using a robot-chassis kit, consisting of two acrylic plates, four DC motors, four wheels, screws, and standoffs. Additionally, two Infrared Obstacle Avoidance Sensor Modules were placed on the front of the Robo-Car, as well as five Line Detection Sensor Modules. An L298N Dual H-Bridge Motor Driver was added to the Robo-Car, as well as a battery case for two 3.7v Lithium-Ion Batteries, and a breadboard. The line detection sensors are used to follow the black lines on the various tracks. The components of the Robo-Car are controlled by using 7400 series logic gates, which perform various different logic functions like AND, OR, NAND, NOR, and NOT. Additionally, a 555 timer, resistors, and capacitors were used to create a PWM to control the

speed of the Robo-Car, and a dual 4- to 1-line multiplexer was used to control the output to the motors of the Robo-Car.

Design

The Robo-Car chassis was constructed over the course of 2 hours, with red and black wires soldered to motors, with the same wire configuration on each motor. The motors were installed onto an acrylic sheet in addition to the Motor Driver. The wires from the motors were attached to Outputs A and B of the controller board. Motors 3 and 4 were attached to Output A, with the red wires from the two motors placed in output 1, pin 2, and the black wires placed in output 2, pin 3. Motors 1 and 2 were attached to output B, with the red wires placed in output 4, pin 14, and the black wires placed in output 3, pin 13. A battery pack was then attached to the controller board, with the red wire from the battery pack attached to the +5V Power output from the controller board, pin 4, and the black wire from the battery pack attached to the Power Ground output from the controller board, pin 8. 7.2V was inputted into the L298N Motor Drive Controller Board from the battery pack, into pin 4, +12 V, and grounded at pin 8. Using pin 9, +5 V, the voltage was converted and input into the breadboard.

On the front of the Robo-Car, two object detection sensors were connected to the top acrylic sheet an equal distance from the center of the Robo-Car and five line detection sensors were connected to the bottom acrylic sheet, with three centered at the width of the black line and one sensor on each side of the center three. The digital state for each sensor's output was found by measuring each sensor using a multimeter. The digital state 0 was measured at 0 V when the line detection sensor was over white, and the digital state 1 was measured at 5 V when the line detection sensor was over black. Additionally, the digital state 0 was measured at 0 V when the

object detection sensor detected an object, and the digital state 1 was measured at 5 V when the object detection sensor detected and object.

The Robo-Car was initially made to run with only two object detection sensors and two line detection sensors, following the truth table as seen in table 1. Using NAND and NOR gates, the logic for the left and right motors were found as b'cd and a'cd respectively.

Table 1: Four input truth table

LL: Left Line Detection Sensor (0 = white; 1 = black)
RL: Right Line Detection Sensor (0 = white; 1 = black)
OL: Left Obstacle Detection Sensor (1 = no obstacle; 0 = obstacle)
OR: Right Obstacle Detection Sensor (1 = no obstacle; 0 = obstacle)
L: Left Motor (0 = off, 1 = on)
R: Right Motor (0 = off, 1 = on)

LL (a)	RL (b)	OL (c)	OR (d)	L	R
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	0
0	0	1	1	1	1
0	1	0	0	0	0
0	1	0	1	0	0
0	1	1	0	0	0
0	1	1	1	0	1
1	0	0	0	0	0
1	0	0	1	0	0
1	0	1	0	0	0
1	0	1	1	1	0
1	1	0	0	0	0
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	X	X

To improve the Robo-Car, the first modification proposed was adding additional line detection sensors in the center of the Robo-Car, with the purpose of centering the Robo-Car on the black line. Initially, the proposed plan was to add two new sensors to the center of the Robo-Car to detect the black line. However, the logic for this was very complex and was unable to be implemented onto the Robo-Car. To allow the Robo-Car to have more sensors while keeping the

logic simpler, it was proposed to have a truth table for three line detection sensors, with two off to the sides of the black line and one centered on the black line and use an OR gate to OR additional sensors to the existing middle sensor, as seen in figure 1. This allows the Robo-Car to have several middle sensors detecting the black line, expanding the detection radius of the Robo-Car without having extremely complex logic.

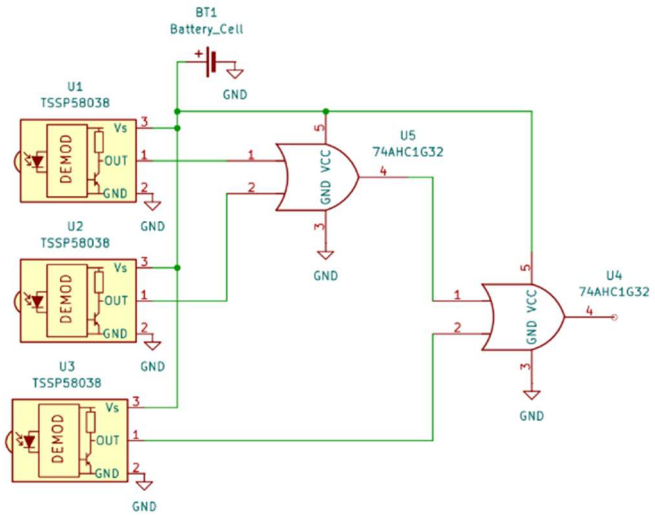


Figure 1: Middle sensor schematic

The second modification proposed to improve the Robo-Car was adding speed controllers to the Robo-Car to adjust the speed of the car as it goes around corners, made using a pulse-width modulation (PWM). Initially, a multiplexer was going to be used to create speeds of 0%, 33%, 66%, and 100% to adjust to different turns, depending on the sharpness of the turn. Given the initial configuration of four-line detection sensors, a truth table was made, as seen in table 2.

Table 2: Proposed second modification truth table

LL: Left Line Detection Sensor (0 = white; 1 = black) LM: Left Middle Line Detection Sensor (0 = white; 1 = black) RM: Right Middle Line Detection Sensor (0 = white; 1 = black) RL: Right Line Detection Sensor (0 = white; 1 = black) L: Left Motor (0 = off, 1 = on) R: Right Motor (0 = off, 1 = on)									
LM (a)	RM (b)	LL (c)	RL (d)	S_{L0}	S_{L1}	S_{R0}	S_{R1}	L	R
0	0	0	0	0	0	0	0	0%	0%

0	0	0	1	1	1	0	0	100%	0%
0	0	1	0	1	0	0	1	66%	33%
0	0	1	1	1	1	0	1	100%	33%
0	1	0	0	0	1	1	0	33%	66%
0	1	0	1	0	1	1	0	33%	66%
0	1	1	0	1	1	1	1	100%	100%
0	1	1	1	1	1	1	1	100%	100%
1	0	0	0	0	0	1	1	0%	100%
1	0	0	1	X	X	X	X	X	X
1	0	1	0	1	0	0	1	66%	33%
1	0	1	1	1	1	0	0	100%	0%
1	1	0	0	0	0	1	1	0%	100%
1	1	0	1	0	0	1	1	0%	100%
1	1	1	0	1	1	1	1	100%	100%
1	1	1	1	1	1	1	1	100%	100%

A PWM was created to decrease the speed of the Robo-Car using a 555-timer, two resistors and a capacitor, as seen in figure 2. The resistor and capacitor values were found to create the desired output speed. Initially, two PWMs were created to output 33% and 66%, with resistors R1 and R2, and capacitor values of 47kΩ, 100kΩ and 10nF and 100kΩ, 47kΩ, and 10nF respectively, found using equation 1.

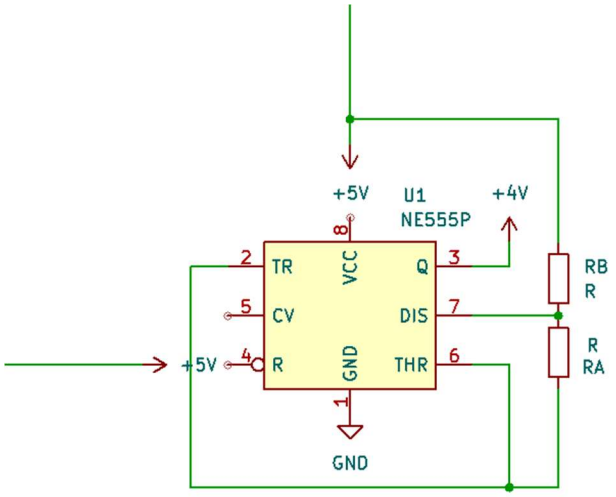


Figure 2: PWM schematic

$$Duty\ Cycle = \frac{R1 + R2}{R1 + 2R2}$$

Equation 1: PWM duty cycle

Using a multimeter, the actual output of each PWM was found to be 51% and 67%, respectively. It was found that to achieve a duty cycle of less than 50%, such as 33%, the PWM must be made with diodes, or the output from the 66% PWM could be passed through a NOT gate, which would invert the signal and output 33% of the input voltage. This allows the Robo-Car to have both 33% and 66% of the input voltage output using only one PWM.

Depending on what the line detection sensors detect, a different speed for each motor can be selected in order to allow the Robo-Car to go slower around tighter corners. This was achieved using a multiplexer. Depending on what the line detection sensors detect, a different select line of the multiplexer will be chosen, as seen in figure 3. This will then output the desired speed for the Robo-Car. It was found that for the multiplexer to output the correct input, the enable pin of the multiplexer must be wired directly to ground, or low, to output the correct voltage, or high.

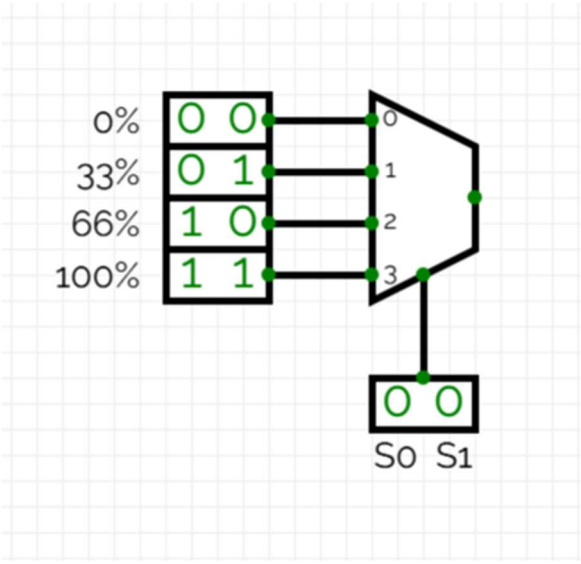


Figure 3: Proposed Multiplexer Layout

After some testing, it was found that only decreasing the forward drive was not enough to go around sharp corners, even when one side of the Robo-Car was at 100% and the other side

was at 0%. To go around extremely sharp corners, the reverse drive of the Robo-Car must be used, making one side of the car go forward while the other side goes backwards.

The reverse drive was implemented in a similar way to the forward drive using the PWM, using a multiplexer. The multiplexer used is a CD74HC153 Dual 4- to 1-Line Multiplexer, which has one select line that controls two different outputs. One multiplexer was used for each side of the drive, set up as seen in Figure 4. The forward drive of each side is controlled by one side of the multiplexer, and the reverse drive of each side is controlled by the other side of the same multiplexer, sharing the same select input. When the select line is 00, both the forward and reverse drive are off, turning that side of the Robo-Car's drive off. When the select line is 10, the forward drive is off and the reverse drive is on, making that side of the Robo-Car go in reverse. When the select line is 11, the forward drive is on and the reverse drive is off, making that side of the Robo-Car go forward. The select line 01 doesn't lead to anything and is not used in the logic. A schematic of the left and right multiplexers connected to the forward and reverse drive pins of the motor driver can be seen in figure 5.

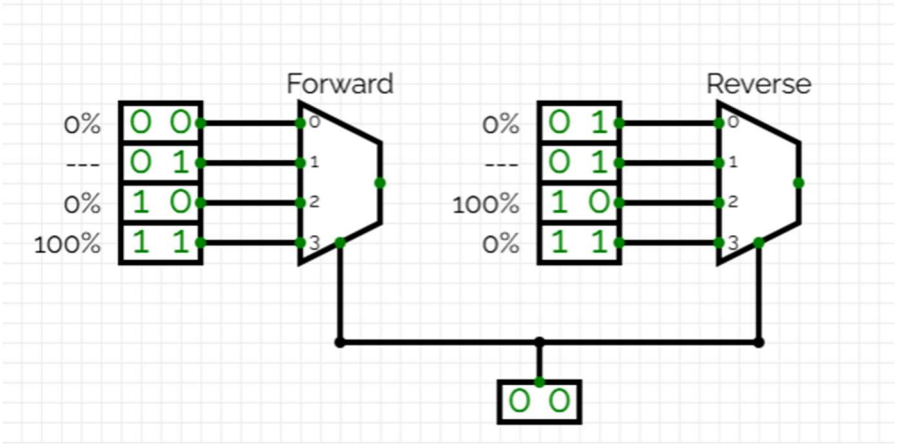


Figure 4: Multiplexer layout

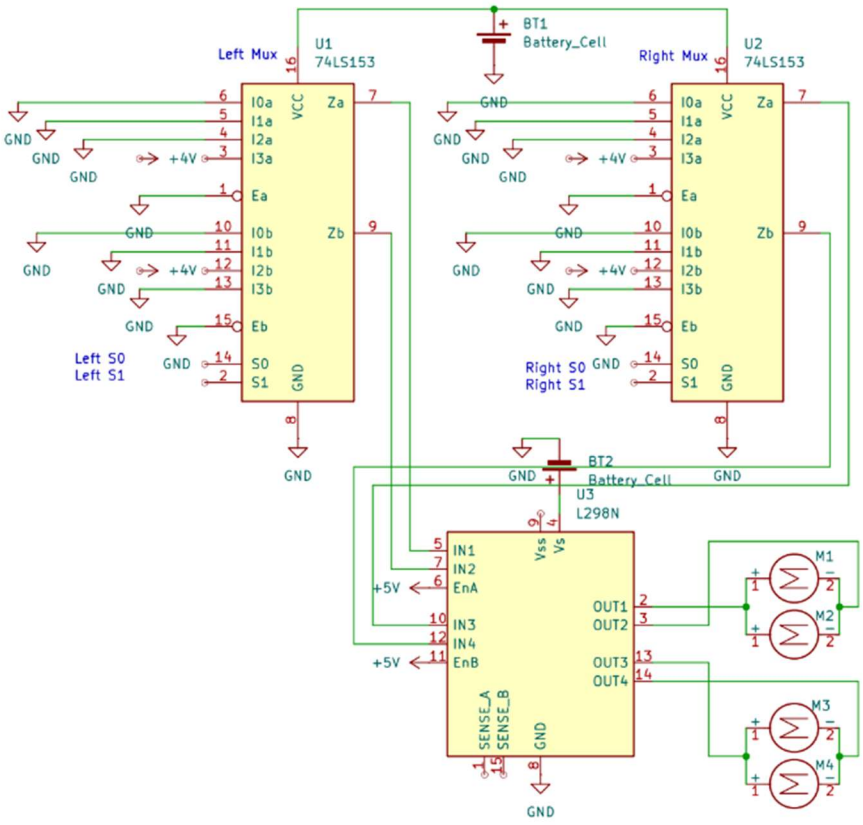


Figure 5: Schematic of multiplexer layout and motor driver

Using five line detection sensors, as explained previously, and the forward and reverse drives on the Robo-Car, a truth table and logic was created to output the correct speeds and direction depending on what the line detection sensors see, as seen in tables 3 and 4, the truth table and state table respectively. The logic equations for the left multiplexer were found to be (ac') + (cb') for S_1 and (ab') + (cb') for S_2 and (ab') + (bc') for S_1 and (ac') + (bc') for S_2 for the right multiplexer. These equations were found using a Python program that calculates the equations using the minTerms and maxTerms. A schematic of the line detection sensors and the logic for the correct select line outputs can be seen in figure 6, with schematic placed on the Robo-Car as seen in figure 7.

Table 3: Final truth table

L: Left Line Detection Sensor (0 = white; 1 = black) M: Middle Line Detection Sensor(s) (0 = white; 1 = black) R: Right Line Detection Sensor (0 = white; 1 = black) L: Left Motor (0 = off, 1 = on) R: Right Motor (0 = off, 1 = on)								
M (a)	L (b)	R (c)	S_{L0}	S_{L1}	S_{R0}	S_{R1}	L	R
0	0	0	0	0	0	0	0%	0%
0	0	1	1	1	0	0	100%	0%
0	1	0	0	0	1	1	0%	100%
0	1	1	0	0	0	0	0%	0%
1	0	0	1	1	1	1	100%	100%
1	0	1	1	1	1	0	100%	-100%
1	1	0	1	0	1	1	-100%	100%
1	1	1	0	0	0	0	0%	0%

Table 4: Final State Table

L: Left Line Detection Sensor (0 = white; 1 = black) M: Middle Line Detection Sensor(s) (0 = white; 1 = black) R: Right Line Detection Sensor (0 = white; 1 = black) L: Left Motor (0 = off, 1 = on) R: Right Motor (0 = off, 1 = on)						
M (a)	L (b)	R (c)	Left Multiplexer	Right Multiplexer	Left State	Right State
White	White	White	0 0	0 0	Off	Off
White	White	Black	1 1	0 0	Forward	Off
White	Black	White	0 0	1 1	Off	Off
White	Black	Black	0 0	0 0	Off	Off
Black	White	White	1 1	1 1	Forward	Forward
Black	White	Black	1 1	1 0	Forward	Reverse
Black	Black	White	1 0	1 1	Reverse	Forward
Black	Black	Black	0 0	0 0	Off	Off

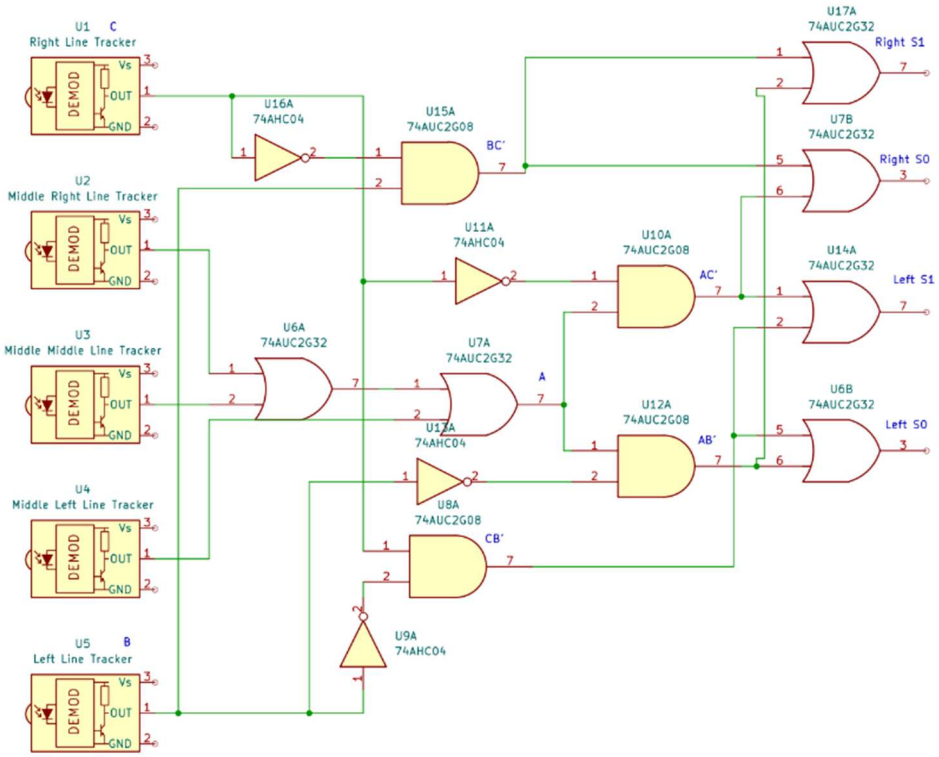


Figure 6: Line following logic schematic

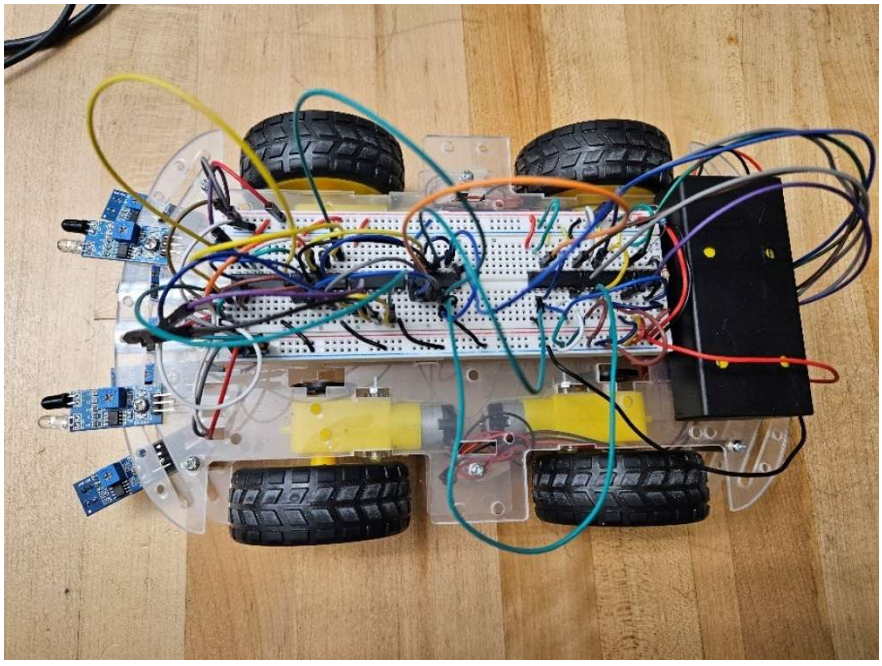


Figure 7: Logic on Robo-Car

The third modification proposed was adding LED headlights to the Robo-Car to illuminate the black track and allow the line detection sensors to detect the line better. Red LEDs

were placed on the front of the Robo-Car, as seen in figure 8. A schematic of the LED layout can be seen in figure 9, with the LEDs placed in series with 220 Ω resistors and connected to 5V.

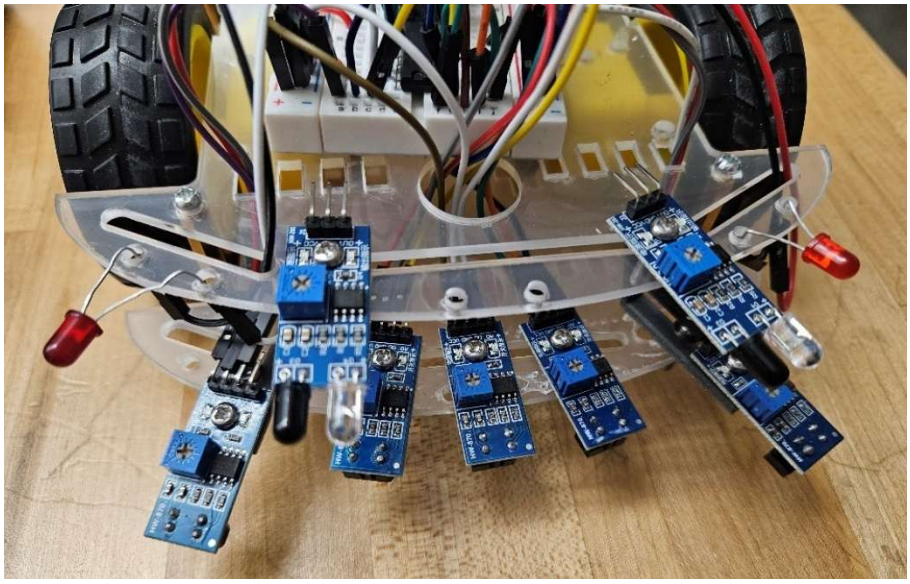


Figure 8: Robo-Car with LEDs

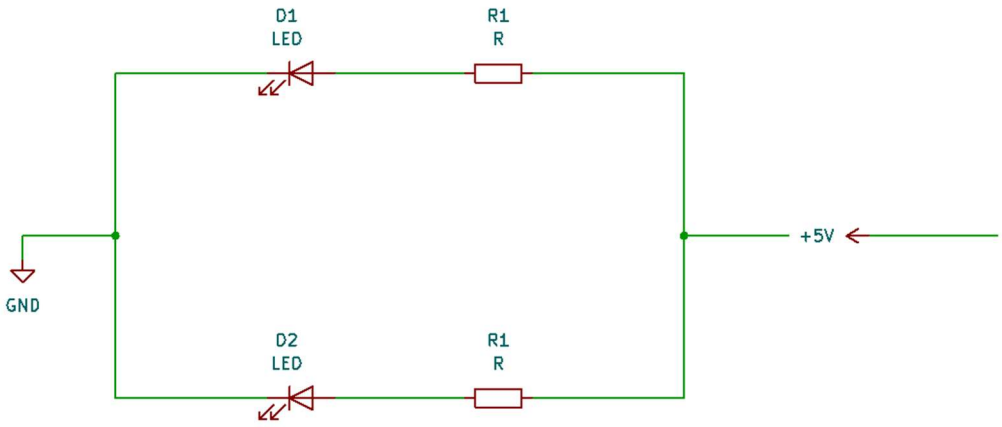


Figure 9: LED schematic

A schematic of the overall Robo-Car design, including all of the elements, can be seen in figure 10. This includes all the modifications added to the Robo-Car.

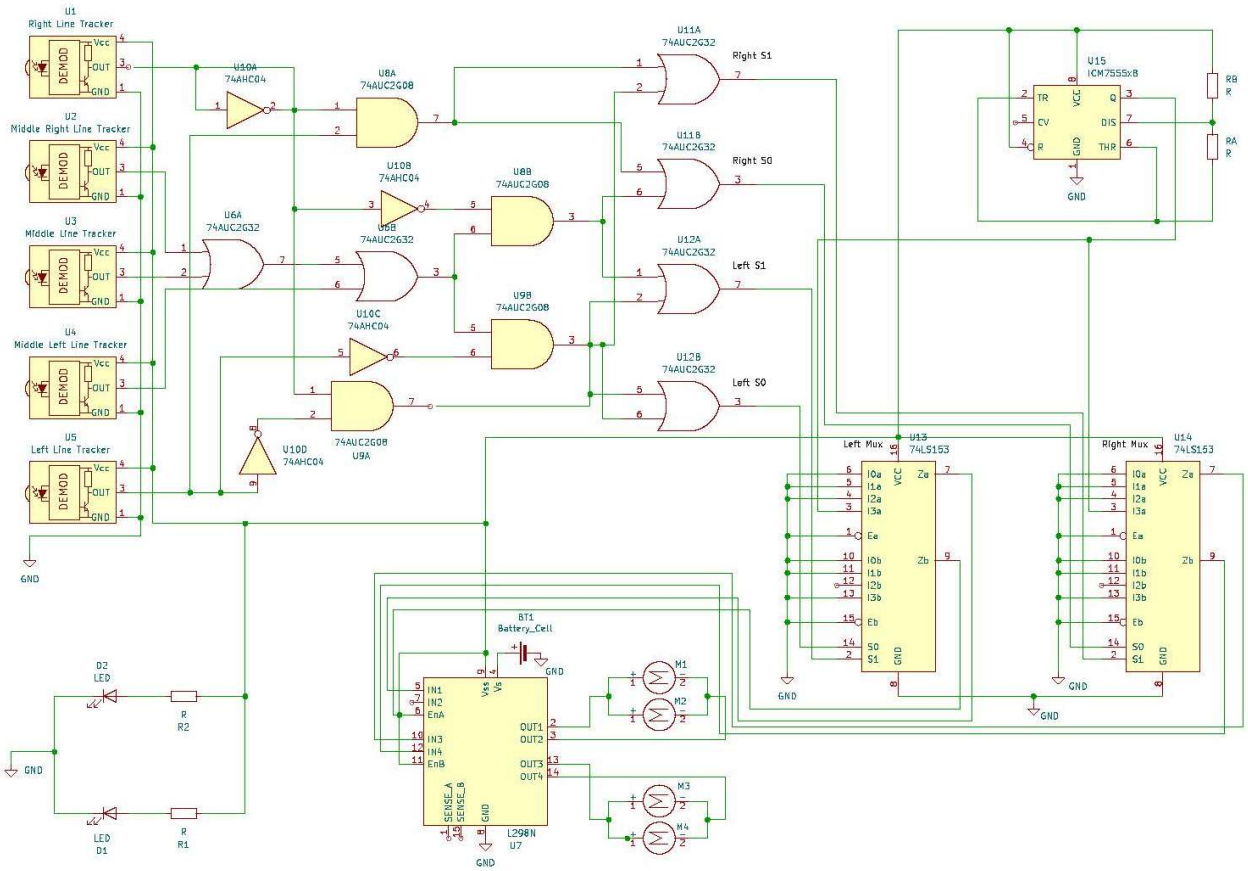


Figure 10: Robo-Car Schematic

Testing

The logic was designed to, depending on which line detection sensor detected black, turn at different angles depending on the sharpness of the corner, as based on the truth table seen in table 3. When tested on the track, the Robo-Car performed as expected, turning when the left or right sensors detected black and going straight when only the middle sensor detects black.

Adding additional sensors to the Robo-Car, the first modification made, allows the Robo-Car to remain centered on the black line and helps the Robo-Car to complete sharp corners. Additionally, the sensors allow the Robo-Car to drive more accurately around the track, allowing the Robo-Car to go at faster speeds while still staying accurate on the track. Changes that would

be made to this proposal after implementation would be adding only one middle sensor to the logic instead of two originally. Although the two line sensors theoretically provided better tracking on the black line, the purpose of the middle line sensor was to ensure that the Robo-Car remained centered on the black line. Therefore, only one sensor completes the task while also simplifying the logic required.

Adding a multiplexer, PWM and using the reverse drive pins, the second modification, allows the Robo-Car to complete sharp corners. The multiplexer allows the Robo-Car to have different outputs to the motors depending on what the line detection sensors detect, using the PWM and reverse drive pins. The PWM allows the Robo-Car to drive at slower speeds, which can also be changed to different speeds, to help complete the intermediate and advanced tracks. Additionally, using the reverse drive pins in addition to the forward drive pins, allows the Robo-Car to complete very sharp corners, which is good for the intermediate and advanced tracks. Changes that would be made to this proposal after implementation would be initially starting using the reverse drive pins instead of starting only using forward drive pins. Using the forward drive pins and slowing down the Robo-Car around corners was not sharp enough to complete the corners.

Adding LEDs as headlights on the Robo-Car did not change the car's performance at all. It was found that they did not provide enough light to illuminate the black line on the track, causing them not to affect the line detection sensor's detection of the line. It was also found that the LEDs cause the Robo-Car to overdraw current. When everything, including the LEDs, are plugged into the Robo-Car, several of the line detection sensors begin to blink on and off, as there is not enough current provided to run all of the elements. Changes that would be made to this proposal after implementation are mounting LEDs on the front of the Robo-Car farther from

the center. Originally, the LEDs were placed near the center of the Robo-Car, however they were moved farther out to accommodate the object detection sensors, which need to be closer to the center of the Robo-Car.

Various resistor values were tested with the PWM to test how different speeds affected the Robo-Car's movement. The PWM was connected as the "100%" voltage on each of the multiplexers to test slower speeds for the Robo-Car, with tested values at 72%, 80%, 85%, and 90%. It was also found that the speed of the Robo-Car depends on how charged the battery is. If the battery is fully charged, the output from the Motor Driver is 5V. However, if the battery has been used for a while, the output decreases to as low as 4V before the Robo-Car stops moving.

The line detection sensors were moved around in various configurations to find the configuration that works best for corners and different tracks. It was found that the distance between each of the line detection sensors could not be wider than the black line, because then two sensors could detect white at the same time and the Robo-Car would stop driving. When on the intermediate track, it was found that the closer together the sensors, the better, with the three middle sensors as close together as possible, and the outer two sensors pointed inward. However, on the advanced track, it was found that the sensors need to be a lot more spread out, as far apart as possible, to detect the sharp corners in time to turn.

With this logic and design, the Robo-Car was found to be inconsistent on sharper and tighter turns, especially when going faster. On the extremely sharp turns of the intermediate and advanced track, the Robo-Car had to be going in at an exact angle to correctly pass the corner at faster speeds. There is an obvious trade off with speed and accuracy, as the faster the Robo-Car goes, the less accurate it is around corners, and the more it runs off the track.

Conclusion

The final design of the Robo-Car was built to be line-detecting and complete sharper corners. Five line detection sensors were added to the front of the Robo-Car, with three in the center of the Robo-Car focused on the black line and one on each side detecting white and turning on black. Additionally, a multiplexer was added to the Robo-Car to provide different outputs depending on what the line detection sensors detect, with three different states, forward, backwards and off. This also utilizes the forward and reverse drive pins on the motor driver. A PWM was also added to allow the Robo-Car to drive at various speeds. Additionally, LEDs were added to the front of the Robo-Car as headlights to allow the Robo-Car to see the line better. Overall, many things were learned, including how to utilize logic gates to control various outputs, work with multiplexers, and select lines to choose different outputs and create a PWM to output different voltages.

Future proposals that would add to the performance of the Robo-Car would be implementing the PWM to slow down the Robo-Car around corners in addition to the reverse drive pins. This would help around very sharp corners, as seen on the advanced track. Additionally, more line detection sensors could be added to the front of the Robo-Car, allowing the Robo-Car to detect sharper corners before they happen and complete corners better

Contribution Table

Team Member Name	Contributions	Hours Spent
Jillian Awe	Created PWM and multiplexer layout, helped with logic and schematic creation. Tested various components. Wrote lab report and presentation	30
Colin Essary	Aided in lab report and presentation	6
Brayden Garcia	Created and simulated logic for line following, made schematics, aided in testing with the PWM and mux, proofread the lab report and presentation.	30

References

[1] Handson Technology, *L298N Dual H-Bridge Motor Driver User Guide,* *L298N Motor Driver.pdf*

[2] Texas Instruments, *CDx4HC08 Quadruple 2-Input AND Gates,* *CDx4HC08 Quadruple 2-Input AND Gates datasheet (Rev. D)*

[3] Texas Instruments, *CDx4HC32 Quadruple 2-Input OR Gates,* *CDx4HC32 Quadruple 2-Input OR Gates datasheet (Rev. D)*

[4] Texas Instruments, *CDx4HC00 Quadruple 2-Input NAND Gates,* *CDx4HC00 Quadruple 2-Input NAND Gates datasheet (Rev. D)*

[5] Texas Instruments, *CDx4HC02 Quadruple 2-Input NOR Gates,* *CDx4HC02 Quadruple 2-Input NOR Gates datasheet (Rev. D)*

[6] Texas Instruments, *CDx4HCT04 Hex Inverters* , *CD54HCT04, CD74HCT04 High-Speed CMOS Logic Hex Inverter datasheet*

[7] Texas Instruments, *CD54HC153, CD74HC153 Dual 4- to 1-Line Selector/Multiplexer, CD54HC153, CD74HC153, CD54HCT153, CD74HCT153 datasheet (Rev. C)*

[8] Texas Instruments, *xx555 Precision Timers, xx555 Precision Timers datasheet (Rev. J)*

[9] Texas Instruments, *CDx4HC151 High-Speed CMOS Logic 8-Input Multiplexer, CDx4HC151, CDx4HCT151 High-Speed CMOS Logic 8-Input Multiplexer datasheet (Rev. D)*

Attachments

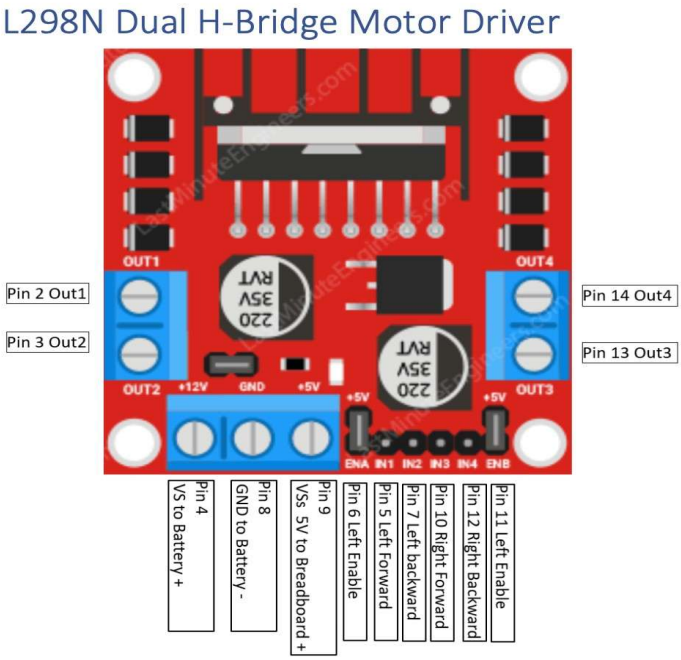
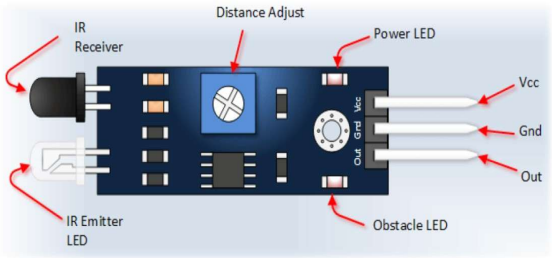


Figure 11: L298N Dual H-Bridge Motor Driver Pin Configuration

IR Obstacle Detection Module Pin Outs

The drawing and table below identify the function of module pin outs, controls and indicators.



Pin, Control Indicator	Description
Vcc	3.3 to 5 Vdc Supply Input
Gnd	Ground Input
Out	Output that goes low when obstacle is in range
Power LED	Illuminates when power is applied
Obstacle LED	Illuminates when obstacle is detected
Distance Adjust	Adjust detection distance. CCW decreases distance. CW increases distance.
IR Emitter	Infrared emitter LED
IR Receiver	Infrared receiver that receives signal transmitted by Infrared emitter.

Figure 12: IR Obstacle Detection Module Pin Outputs