**Self-Driving Car Report**

Name: Mamert Vonn G. Santelices

Student ID: 90026174

Date: 29 March 2023

Word Count:

Table of Contents

[Introduction 2](#_Toc130331160)

[Background 2](#_Toc130331161)

[Current Research 2](#_Toc130331162)

[Methodology 3](#_Toc130331163)

[Materials 3](#_Toc130331164)

[Procedure 3](#_Toc130331165)

[Schematics 4](#_Toc130331166)

[Blueprint 5](#_Toc130331167)

[Code 6](#_Toc130331168)

[Documentation 6](#_Toc130331169)

[Conclusion 7](#_Toc130331170)

[Results 7](#_Toc130331171)

[Discussion 7](#_Toc130331172)

[Recommendations 7](#_Toc130331173)

[Bibliography 8](#_Toc130331174)

# Introduction

## Background

The Unit Outline of CAT01S1 requires the enrolled student to build a self-driving robot car using an Arduino and the associated parts provided by the tutor. The included associated parts are brought from the Geekcreit DIY L298N 2WD Ultrasonic Smart Tracking Moteur Robot Car Kit**.** The Robot Car Kit includes 2 Wheels powered by DC Motors; a pivot wheel placed in the middle-back; and a distance measuring sensor intended to assist the Robot Car navigate through its surroundings. Another goal set for this project is to implement a ‘dance’ feature in the robot car. The robot car will be programmed to dance to the rhythm of a chosen song along with the ability to drive itself.

## Current Research

A quick overview of the project presents itself as constructing a simple Proto-type Roomba with a few gimmicks. Tribelhorn and Dodds (2007) asserts the value of Roombas for research and educational purposes because it is a low-cost resource. Tribelhorn and Dodds (2007) list the benefits of a Roomba such as: commercially available hardware interface; released serial API; its sensors and actuator’s accuracy along with its extensibility of its computing and sensory abilities; and its current spatial reasoning algorithms. The ideal result for this project is to produce something similar to the listed capabilities of the iRobot Roomba.

# Methodology

## Materials

|  |  |  |
| --- | --- | --- |
| Unit  Code | Item | Number of Pieces |
| U1 | Arduino Uno R3 | 1 |
| U1 | Sensor Expansion Shield V5.0 | 1 |
| U2 | L298N Motor Driver Module | 1 |
| - | HC-SR04 Ultrasonic Module Distance Measuring Sensor | 1 |
| BAT2 | 9V Battery Case Holder (w/ six 1.5V Battery Slots) | 1 |
| - | Micro Servo Motor SG90 | 1 |
| M1,M2 | DC 3V-6V Single Axis Gear Reducer Motor | 2 |
| LW,RW | Rubber Wheels | 2 |
| PW | Pivot Wheel | 1 |
| S2 | ZF SPST, On-None-Off Rocker Switch Panel Mount | 1 |
| J | DC Barrel Jack Plug (Male) | 1 |
| Miscellaneous Materials | | |
| Electrical Cables | | 8 |
| Female-to-female Jumper Wires | | 4 |
| Nut Pole Connecting Screw Rod | | 10 |
| Flat Screws | | 2 |
| Nut & Bolt | | **many** |
| 3D-Printed Placeholder Chassis | | 1 |
| 3D-Printed T-Piece | | 2 |

## Procedure

Firstly, assemble the modules at the top of the chassis. At the tail end of the car, screw four Nut Poles at the appropriate holes for the Arduino Uno. Thereafter, connect the Sensor Shield on top of the Arduino Uno. Next to the Arduino Uno and adjacent to two thin rectangular holes are two rounded holes for two Nut Poles. On top of that Nut Pole would be the Motor Driver Module. Lastly, at the bottom of the Motor Driver Module is a wide rectangular hole for the Panel Mount Switch. Moving on to the bottom of the chassis, screw four Nut Poles at the tail end of the car, then place the Pivot Wheel at the Nut Poles. Secondly, using a flat screw place the Battery Case next to the pivot wheel. Next to the Battery Case, at the left and the right are two thin rectangular holes to insert two T-Pieces. Then, screw in the two DC Motor at the left and right T-Pieces. Look at Figure 2.\* as reference.

The Battery Case is the source of energy supply for the system. Thus, the DC Barrel Jack Plug is wired from the Battery Case to the Arduino Uno. The Batteries then distributes the power to the Motor Driver by connecting the positive and negative wires to the Voltage and GND pins, respectively. In-between that connection is a Panel Mount Switch to regulate the on-and-off function of the system. The DC Motor runs using the output pins of the Motor Driver, wherein two output pins are each connected to one of the two DC Motors. The system can control the motors by using the Motor Driver’s four input pins connected to the Arduino Uno’s digital or analog pins. Further insight can be seen in Figure 1.1

## Schematics

*Figure 1.1: Setting Up Two DC Motors and The Motor Driver*

## Blueprint

|  |  |
| --- | --- |
|  |  |

## Code

|  |
| --- |
| toycar.ino |
| void setup() {  }  void loop() {  } |

## Documentation

# Conclusion

## Results

## Discussion

## Recommendations

# Bibliography

Tribelhorn, Ben and Zachary Dodds. 2007. “Evaluating the Roomba: A Low-cost, Ubiquitous Platform for Robotics Research and Education” Proceedings 2007 *IEEE International Conference on Robotics and Automation*. https://doi.org/10.1109/ROBOT.2007.363179.