Car Warz

Place picture of project with team members here.

Complete the caption, below.

Delete this text box before pasting in your photo!

Team Members (left-to-right on picture, above)	*Class No.*	*Lab Div*
Alek Patel	3838-P	9
Ribhav Agarwal	5686-A	9
Vinay Nagarajan	9829-N	10

Report/Functionality Grading Criteria	*Points*
Originality, creativity, level of project difficulty	20
Technical content, succinctness of report	10
Writing style, professionalism, references/citations	10
Project functionality demonstration	20
Overall quality/integration of finished product	10
Effective utilization of microcontroller resources	10
Significance of individual contributions*	20
Bonus Credit Opportunities	*Bonus*
Early completion	0.5%
PCB for interface logic	2%
Poster (required for Design Showcase participation)	1%
Demo video (required for Design Showcase participation)	1%
Design Showcase participation (attendance required)*	1%

**scores assigned to individual team members may vary*

Grading Rubric for all Criteria (Including Bonus)	*Multiplier*
Excellent – among the very best projects/reports completed this semester	1.0 - 1.1
Good – all requirements were amply satisfied	0.8 - 0.9
Average – some areas for improvement, but all basic requirements were satisfied	0.6 - 0.7
Below average – some basic requirements were not satisfied	0.4 - 0.5
Poor – very few of the project requirements were satisfied	0.1 - 0.3

TABLE OF CONTENTS

1.0 Introduction

Car Warz is a game in which two cars race in a custom-made track to the finish line. One car will be controlled by a player who uses their phone to control the car. The other car is controlled by an ultrasonic sensor.

The player controlled car is connected to the player's phone via Bluetooth [3]. An already made app is what moves the car and has five controls: straight, reverse, left, right, and stop. A player can use any of these as he/she pleases to get through the race track.

The computer controlled car is controlled solely with an ultrasonic sensor [6]. If the car comes to a wall or object blocking it, it looks both ways finding a clear path and it will go in that direction.

The race track is completely made of carboard and has walls to make sure the ultrasonic sensor can see the wall and turn accordingly.

The base of the car was built by Ribhav Agarwal. This involves wiring the motors, the H bridge [2], battery pack, and the 9SC12 microcontroller. The wheels were attached to the motors and everything else was glued onto the top of the car. For the sensor [6], the servo motor [4] and the ultrasonic sensor were both wired to the microcontroller. Ribhav also made schematic for the Printed Circuit Board and the overall project schematic.

Vinay Nagarajan and Alek Patel worked on the software side of the car. For the first car, the app communicated with the Bluetooth module [3] which sent a character to the microcontroller. Each character represented a direction to move that car in which then turned the motors on accordingly. For the self-driving car, we made an Arduino nano [5] communicate with the microcontroller. The Arduino only controlled the movement of the servo motor and reading of the distances through the ultrasonic sensor. It communicated these values to the microcontroller which used the values to move the car accordingly.

Outside the packaging, the Bluetooth controlled car also contains an LED, an LCD display [1], and a switch. The ON button on the app turns the LED on, and OFF button simply turns it off. The first line of the LCD display displays which direction your car is currently going in. If it is moving straight, it will display "Moving Forward!" and so on. The second line of the LCD display contains a reaction time which is basically the time from the last button pressed on the app. If you were to press forward and then right, it would display the time elapsed between the press of forward and right. The switch simply supplies voltage to the motors when needed.

Outside the packaging of the self-driving car, there is simply a switch. This switch supplies voltage to the motors when needed.

2.0 Interface Design

Describe any external interfaces utilized (e.g., switches, LEDs, sensors). Include your Eagle or OrCAD schematic as Appendix B.

Length should be about one page.

L298N H – Bridge [2]

The L298N Dual H-Bridge motor driver has a total of six inputs and two outputs. The two outputs are connected to the four motors and a common ground. There are two enable input pins which decide if the motor is disabled or enabled. The other four input pins control the switches on the H-bridge and based on the configuration that is outputted by the active high port pins. This decides the direction of the car. There is a separate source of voltage for the H – Bridge which allows us to rate the motors at a higher voltage compared to the pins.

PWM Controlled Motors

We use four motors to control all functionality: forward, reverse, right turn, and left turn. This allows a very precise control of motion. The direction control is as follows:

- To move forward, all the wheels move forward at the same speed.
- To turn right, both left wheels move forward at the same speed.
- To turn left, both right wheels move forward at the same speed.
- To reverse, all the wheels move backwards at the same speed.

This is all achieved using the H – Bridge which is controlled by the port pins PT0, PT1, PT2, and PT3.

HC-06 RS232 Bluetooth Transceiver Module [3]

The Bluetooth receiver has four pins. One pin is for Vcc, one is for ground, and the remaining two pins are output pins. The two output pins are for receiving and transmitting. The transmitting pin is not used for this car. The receiving pin receives a letter corresponding to a direction and that letter is then read by the microcontroller on the RX pin.

External Battery Pack

The two connections from the external battery pack are simply wired to the H – Bridge, one being ground and the other a voltage source.

SG90 Micro Servo [4]

There are three pins on the servo, one being a voltage source and one being grounded. The last pin is an input pin connected to the Arduino Nano.

HD44780 LCD Controller [1]

The LCD had 15 pins. The first, third and the fifteenth pin were grounded. Pins 16 and 2 were connected to 5 volts. Pin 4 on the LCD which was the register select was connected to PTT4 on the 9sC12 microcontroller. Pin 5 on the LCD which was the LCD read/write was connected to PTT5 on the 9sC12 microcontroller. Pin 6 on the LCD which was the LCD clock was connected to PTT6 on the 9sC12 microcontroller. PTT 4, 5 and 6 are all part of the TIM module of the microcontroller. Pin 7 to pin 14 were connected to the GAL 22v10 from the least significant to the most significant bit. The GAL 22v10 was connected to the microcontroller at PM5 and PM4 which are a part of the SPI module.

Arduino Nano [5]

The Arduino Nano is only used to control the servo motor. It also communicates with the ultrasonic sensor [6] in order to supply the right data to the microcontroller. The VIN and GND pins of the Arduino were connected accordingly. Another pin we connected was the TX1 pin. This pin sent information to the SCI on the microcontroller. It was used to send the proper distance taken from the ultrasonic sensor. Two other pins were used on the Arduino for the servo motor and the ultrasonic sensor. They were connected to pins A4 and A5 respectively.

3.0 Microcontroller Resource Utilization

Describe how the microcontroller's peripherals (ATD, SCI, SPI, TIM, PWM) as well as its other on-chip resources (RTI, SRAM, flash memory, etc.) were utilized, including the mode(s) in which they were programmed to operate. Provide rationale for the choices made.

The microcontroller peripherals that we used significantly are TIM, PWM, SCI, SPI and ATD. RTI was used to initialize interrupts for 2.048 ms interrupt rate. TIM Module was used to provide periodic independent interrupts every 1 millisecond to keep a count of time between two successive key presses. SPI was used to display characters on a LCD Display by reading the SPTEF bit and writing data to the SPI data register whenever necessary. PWM was used to provide a controlled duty cycles to each of the 4 motors by ensuring that a maximum 8-bit period is set. SCI was set to have a baud rate of 9600 and was configured to enable the receive data register so that data received by the Bluetooth is communicated to the HCS12 in an appropriate manner. For the self-driving car, ATD was used in the Ultrasonic ping sensor to compute the left and right distance of an obstacle from the sensor and a digital output is communicated to the HCS12 which then moves the car in desired direction. If the left distance returned is greater than the right distance, then the car moves in the left direction. If the right distance is greater than the left distance, the car moves in the right direction. It is programmed to move forward if there is no obstacle present ahead of it.

SPI Initializations for baud rate of 6 Mbs */

```
  SPICR1 = 0x50;
  SPICR2 = 0x0;
  SPIBR = 0x01;
```

SCI Initializations for 9600 baud rate

```
  SCIBDH =  0x00; //set baud rate to 9600
  SCIBDL =  0x9C; //24,000,000 / 16 / 156 = 9600 (approx)    //0x38
  SCICR1 =  0x00; //$9C = 156
  SCICR2 =  0x0C; //initialize SCI for program-driven operation
  DDRB   =  0x10; //set PB4 for output mode
  PORTB  =  0x10; //assert DTR pin on port
```

TIM initializations for periodic 1.000ms

```
TSCR1_TEN = 1;
  TIOS = 0x80;
  TSCR2 = 0x0C; //pre-scale factor is 16
  TC7 = 1500;
  TIE_C7I = 0;
```

/* PWM initializations */

```
  MODRR = 0x0F;    //PT3,2,1,0 used as PWM Ch 3,2,1,0 output
  PWME = 0x0F;    //enable PWM Ch 0,1,2,3
  PWMPOL = 0x01;  //negative polarity
//  PWMCTL = 0x00;  // no concatenate (8-bit)
//  PWMCAE = 0x00;  // left-aligned output mode
  PWMPER3 = 0xFF; // set maximum 8-bit period
  PWMPER2 = 0xFF; // set maximum 8-bit period
  PWMPER1 = 0xFF; // set maximum 8-bit period
  PWMPER0 = 0xFF; // set maximum 8-bit period
  PWMDTY0 = 0x7F;  // initially clear DUTY register
```

4.0 Packaging Design

Both cars have a plastic chassis with all the wiring and hardware done on top of the plastic chassis. For the wheels of the car, the tires are made of rubber and the rims are made of plastic.

For the Bluetooth controlled car, everything except the switch is inside the car. The breadboard containing our LCD display, microcontroller, Bluetooth module is at the front of the car. Behind that is the H – Bridge which controls the voltage and rotation of the motors. Finally, behind that is the battery pack which contains 4 batteries of 1.5 volts making a total of 6 volts. Underneath all of this, we have 4 motors and a 5 volt power bank which supplies voltage to the microcontroller. On top of all of this, we have a cardboard cover. This cover hides all the wiring and hardware inside the car and also protects it from debris and other damage.

For the self-driving car, the switch and the ultrasonic sensor are both out of the cover of the car. The ultrasonic sensor is at the front of the car and coming out a little bit so it can read distances properly. The servo motor is connected directly below the sensor allowing it to turn as needed. Behind these connections is our breadboard which contains the microcontroller and the Arduino connections. Behind this is our H-Bridge which controls the voltage and rotation of the motors. Finally, behind that is the battery pack which contains 4 batteries at 1.5 volts each for a total of 6 volts. Underneath all of this we have 4 motors, the servo motor, and 2 battery packs. One battery pack is for the Arduino and the other is for the microcontroller. On top of all of this is another cardboard cutout which is our cover for the car. Again, this prevents from damage and hides most of the hardware and wiring.

5.0 Summary and Conclusions

Describe what you learned from completing the project and what you might do to improve your design if you had more time.

Length should be about one page.

Learned Objectives

- The most important thing we learned is how to use the microcontroller to make it perform multiple tasks at the same time. We also learned how to code necessary logic more efficiently to consume less power from our batteries and make the cars run smoothly.
- We learned more thoroughly how to utilize the modules, especially SCI.
- Learned how to use numerous software such as ORCAD.
- We also learned how to use many external peripheral devices such as the H-Bridge, servo motors, ultrasonic sensors, and Bluetooth modules.

Time Constraint

- If we had a little more time, we could implement a voice controlled car or a motion detecting car. A car that can view your hand gestures and move accordingly. We could also improve on the body of the car if we were given a little more time.

Conclusion

- This project was truly a learning experience for all of us as it combined everything we learned in class and labs in one project. This experience has thought us many things such as working with a team on order to achieve a goal within a time constraint. This project instills in us the capability to build more advanced designs in future classes as well as industry.

6.0 References

Data Sheet
[1] Hitachi, “Dot Matrix Liquid Crystal Display Controller/Driver”, HD44780 data sheet, 1999 [Revised 2000].

Data Sheet
[2] STMicroelectronics, “Dual Full-Bridge Driver”, L298 data sheet, 2000

Data Sheet
[3] Guangzhou HC Information Technology Co., Ltd., “Product Data Sheet”, HC-06 data sheet, 2000

Data Sheet
[4] Tower Pro, “Data Sheet”, Servo Motor SG90 data sheet, 2001

Online Source
[5] Arduino. (2016). Arduino Nano [Information Page]. Available Online: https://www.arduino.cc/en/Main/ArduinoBoardNano

Data Sheet
[6] Elec Freaks, “Ultrasonic Ranging Module HC-SR04”, “Product Features”, HC-SR04 data sheet, 2002

Appendix A:

Individual Contributions and Activity Logs

Activity Log for: Alek Patel **Role:** Interfacing Leader

Activity	*Date*	*Start Time*	*End Time*	*Time Spent*
Brainstorming Ideas	11/16	7:00 pm	8:30 pm	1.5 hours
Materials Ordering	11/20	3:00 pm	5:00 pm	2.0 hours
Materials Shopping	11/22	1:00 pm	2:00 pm	1.0 hour
Wiring and Building Bluetooth Car	11/25	3:00 pm	8:00 pm	5.0 hours
Coding the Bluetooth to Simply Move	11/26	7:00 pm	11:00 pm	4.0 hours
Working on the Bluetooth Car	11/27	3:00 pm	12:00 am	9.0 hours
Wiring and Building Self-Driving Car	11/28	3:00 pm	9:00 pm	6.0 hours
Coding the Self-Driving Car	12/1	4:00 pm	7:00 am	15.0 hours
Coding the Self-Driving Car	12/2	1:00 pm	5:00 pm	4.0 hours
Adding the LCD Display	12/2	10:00 pm	12:00 am	2.0 hours
Final Report	12/3	12:00 am	6:30 am	6.5 hours
YouTube Video	12/3	11:00 am	12:00 pm	1.0 hour
PowerPoint Presentation	12/3	3:00 pm	6:00 pm	3.0 hours
Finalizing	12/4	4:00 pm	8:30 am	16.5 hours
Finalizing	12/5	12:00 pm	12:00 am	12.0 hours
TOTAL				**88.5 hours**

Written Summary of Technical Contributions: Alek Patel

As the TDP and peripheral leader of this project, I worked on packaging and coding. I made sure that all the peripherals were being used properly and significantly as per the guidelines of this project. I assisted the software leader, Vinay, greatly in logic of the code and debugging the code. For this project, we used TIM, SCI, SPI, and PWM. With all of these peripheral modules, we were able to make both cars fully functional.

My most major contribution to this project was working with the SCI peripheral in order to receive inputs from both the Bluetooth module and the Arduino Nano. The Bluetooth module sends information from the app and then sends this information to our microcontroller. The inchar function reads this information from SCI and we use it to move the car in whichever direction the user pleases. For the self-driving car, the Arduino communicates with the microcontroller in order to drive the car. The microcontroller controls most of the functionality of the car while the Arduino is only controlling the servo motor and the ultrasonic sensor.

I also made the entire car and the packaging for the self-driving car. It consists of a cardboard cutout that fits the car. There is an opening at the front of the car allowing the ultrasonic sensor to detect obstructions to the left and right of the car. There is a switch on top of the car which supplies voltage to the H-Bridge when it is turned on. Ribhav was in charge of the packaging of the Bluetooth controlled car while I was debugging some issues with the other car.

I am also responsible for editing of this final report along with working on presentation and coordinating the YouTube video.

I also made the entire race track which consisted of simply 2 right turns. The race track is big enough for two cars to occupy it with a 6-inch separation between the cars. Basically, the track was 20 inches wide during straight paths and turns. It consists of cardboard walls so the ultrasonic sensor can see the walls and turn accordingly.

Activity Log for: Vinay Nagarajan **Role:** Software Leader

Activity	*Date*	*Start Time*	*End Time*	*Time Spent*
Brainstorming Ideas	11/16	7:00 pm	8:30 pm	1.5 hours
Materials Ordering	11/20	3:00 pm	5:00 pm	2.0 hours
Materials Shopping	11/22	1:00 pm	2:00 pm	1.0 hour
Wiring and Building Bluetooth Car	11/25	3:00 pm	8:00 pm	5.0 hours
Coding the Bluetooth to Simply Move	11/26	7:00 pm	11:00 pm	4.0 hours
Working on the Bluetooth Car	11/27	3:00 pm	12:00 am	9.0 hours
Wiring and Building Self-Driving Car	11/28	3:00 pm	9:00 pm	6.0 hours
Coding the Self-Driving Car	12/1	4:00 pm	7:00 am	15.0 hours
Coding the Self-Driving Car	12/2	1:00 pm	5:00 pm	4.0 hours
Adding the LCD Display	12/2	10:00 pm	12:00 am	2.0 hours
Final Report	12/3	12:00 am	6:30 am	6.5 hours
YouTube Video	12/3	11:00 am	12:00 pm	1.0 hour
PowerPoint Presentation	12/3	3:00 pm	6:00 pm	3.0 hours
Finalizing	12/4	4:00 pm	8:30 am	16.5 hours
Finalizing	12/5	12:00 pm	12:00 am	12.0 hours
TOTAL				**88.5 hours**

Written Summary of Technical Contributions: Vinay Nagarajan

Provide a concise but sufficiently detailed description of your technical contributions to the project.

Length should be about one page.

As the software and program leader, I was involved in coding, designing and programming the HCS12 microcontroller to deliver the tasks. Initially, I started with programming the PWMDTY channels 0,1,2 and 3 to ensure that a command given by the microcontroller can control the motion and speed of a car. Then I proceeded to interface the Bluetooth to the HCS12 by enabling the corresponding SCI transmit bit and receive bit. Upon writing test code to ensure that Bluetooth interfacing works appropriately, the corresponding characters pertaining to left, forward, right and reverse were programmed into the microcontrollers. Then I proceeded to program the self-driving car to move forward, left, right and backward in the same manner.

Once this was completed I proceeded to work with the servo motor and ultrasonic ping sensor so that a self-driving car can function correctly. This task was challenging as I had to program the servo motor to move the sensor in the appropriate direction as necessary. I programmed the servo motor to turn right if it encountered an obstacle ahead of it.

Then it had to compare the left and right distance parameters and ensure which one of them was lesser and move the car in that direction. I constantly worked on this feature for a whole night and got it to work the next day morning. Moving on, a LCD interface was added to the Bluetooth car so that one can find out the total time taken between a start and stop till end of the race. Overall software designing was my primary role and I enjoyed every bit of it. The parts worked individually and I was overall satisfied when the integration of the peripherals worked to make a ‘smart’ self-driving car. Further, I made sure that the Bluetooth driven car was programmed in such a way that one finds it hard to control the race. This makes sure that we will have a tough fight to prove who wins – Bluetooth car VS Self Driving Car!

Activity Log for: Ribhav Agarwal **Role:** Hardware and Packaging Leader

Activity	*Date*	*Start Time*	*End Time*	*Time Spent*
Brainstorming Ideas	11/16	7:00 pm	8:30 pm	1.5 hours
Materials Ordering	11/20	3:00 pm	5:00 pm	2.0 hours
Materials Shopping	11/22	1:00 pm	2:00 pm	1.0 hour
Wiring and Building Bluetooth Car	11/25	3:00 pm	8:00 pm	5.0 hours
Coding the Bluetooth to Simply Move	11/26	7:00 pm	11:00 pm	4.0 hours
Working on the Bluetooth Car	11/27	3:00 pm	12:00 am	9.0 hours
Wiring and Building Self-Driving Car	11/28	3:00 pm	9:00 pm	6.0 hours
Coding the Self-Driving Car	12/1	4:00 pm	7:00 am	15.0 hours
Coding the Self-Driving Car	12/2	1:00 pm	5:00 pm	4.0 hours
Adding the LCD Display	12/2	10:00 pm	12:00 am	2.0 hours
Final Report	12/3	12:00 am	6:30 am	6.5 hours
YouTube Video	12/3	11:00 am	12:00 pm	1.0 hour
PowerPoint Presentation	12/3	3:00 pm	6:00 pm	3.0 hours
Finalizing	12/4	4:00 pm	8:30 am	16.5 hours
Finalizing	12/5	12:00 pm	12:00 am	12.0 hours
TOTAL				**88.5 hours**

Written Summary of Technical Contributions: Ribhav Agarwal

Provide a concise but sufficiently detailed description of your technical contributions to the project.

Length should be about one page.

As the Hardware and Packaging leader for the project I worked on the task of interfacing all the connections from the 9sc12 microcontroller to the external peripherals used which included devices like a car chassis with four motors, external power bank, an H-bridge dual motor stepper, Bluetooth module device, ultrasonic sensor and a rotation motor.

My contributions involved and started from first constructing the basic model of the car to be used from the parts purchased. After the construction of a tested chassis the next task was to set it up with a battery pack and a dual motor h-bridge that would make the motors run from the 9sc12.

With the setup for a working car, the microcontroller was interfaced with the H bridge using its PWM channel, with the Bluetooth module using SCI channel and with the LCD display using the SPI channel. Also an Arduino was interfaced to control the servo motor for the ultrasonic sensor that made the sensor look in all the directions. All connections were done using jumper wires and required soldering the motor wires in the chassis.

For the packaging of the constructed cars, cardboard boxes were used to provide covers for the car while maintaining the functionality of the cars and all the peripherals.

The hardware construction also required to allow continuous testing of all the peripherals therefore most of the pins were wired instead of using a PCB until the surety of all the possible connections on the bread board.

Activity Log for: <name-4> **Role:** <role on team>

Activity	***Date***	***Start Time***	***End Time***	***Time Spent***

Written Summary of Technical Contributions: <name-4>

Provide a concise but sufficiently detailed description of your technical contributions to the project.

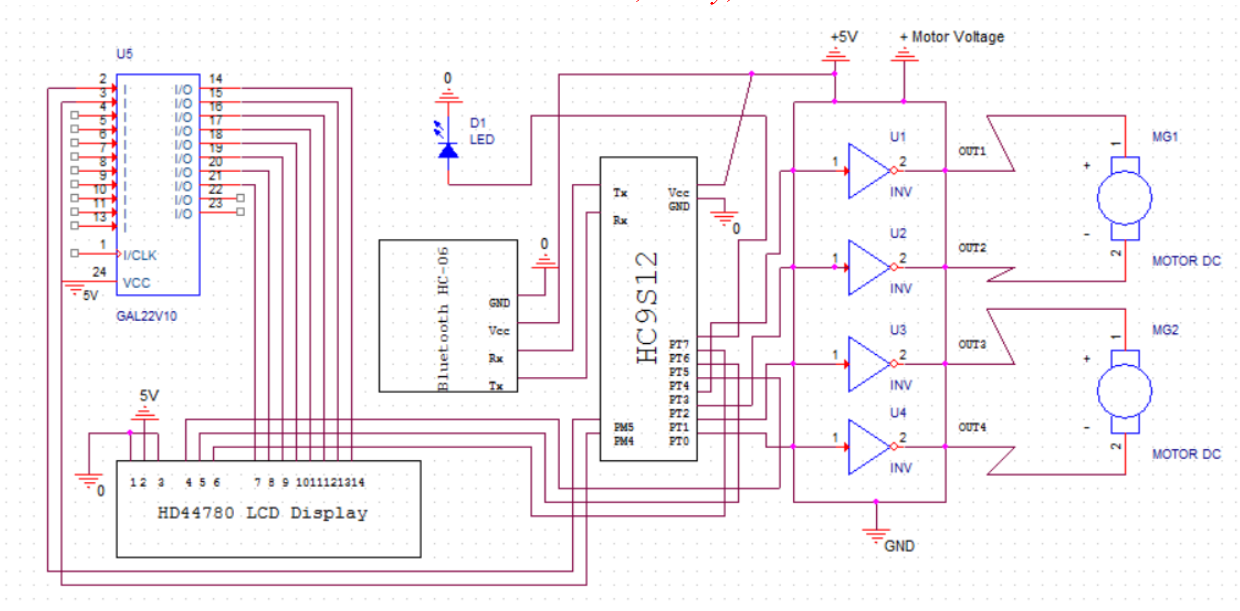
Length should be about one page.

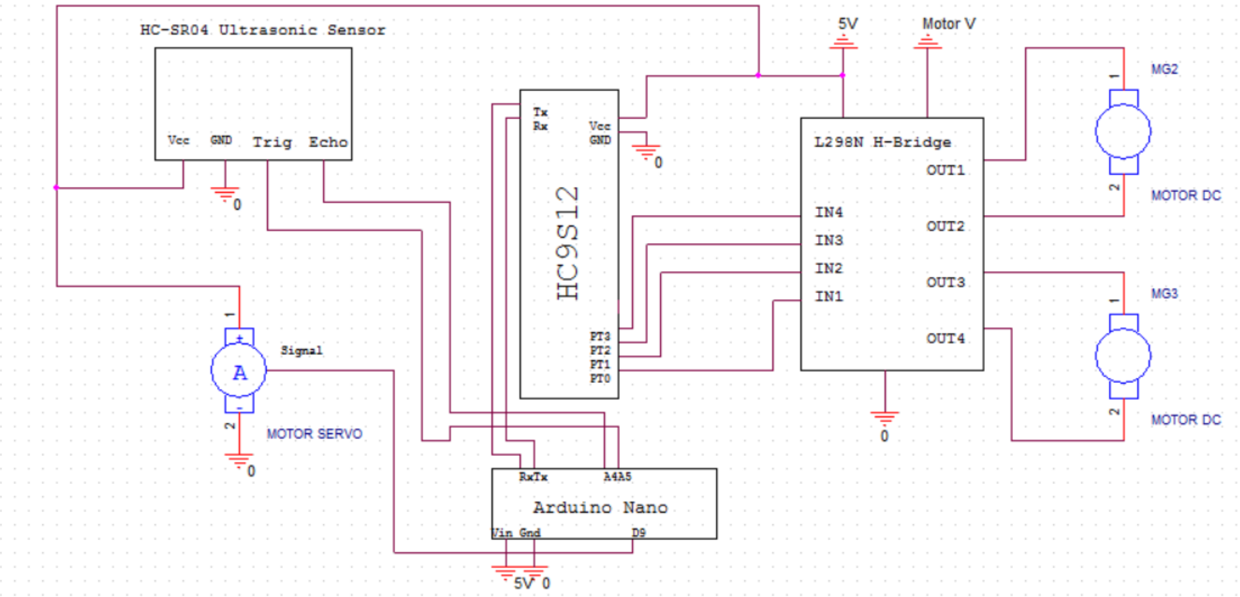
Appendix B:

Interface Schematic
and
PCB Layout Design

Bluetooth Car

- *Ribhav, Vinay, and Alek*





Self-Driving Car

- *Ribhav, Vinay, and Alek*

Appendix C:

Software Flowcharts

Include software flow diagrams and/or pseudo code here.

Be sure to clearly identify the team member(s) responsible for producing this documentation.

NOTE: Software source listing file must be submitted on-line and should NOT be included here.

Appendix D:

Packaging Design

Paste illustrations/pictures of your project packaging here.

Be sure to clearly identify the team member(s) responsible for producing this documentation.