

Robot Rover Remote Control Car
Software and Hardware Specification Sheet

Darrel Asare, David Asare, Deepak Singh

April 15, 2020

Declaration of Joint Authorship

We Darrel Asare, David Asare, and Deepak Singh confirm that this work is the joint work of our group, and is explained in our own words. Each of us provided functioning hardware for our sensor. I Darrel Asare, provided an LM393 Speed Sensor, David Asare provided a Distance Sensor, and Deepak Singh provided the car and the DC motors. In our project effort, I am the lead for further development of our mobile application, since I have the database set up on my firebase account. David is the lead for displaying distance and surveillance camera on the app screen, and Deepak is the lead for the movement of the car and PCB status.

Proposal

The name of our Project is called the Mini Robot Rover. We plan to make a control car that can be controlled using a cell phone app, and a camera to be implemented into the app. We want the car to display speed and distance as it moves. In our software and hardware course, we have created a mobile application, worked with databases and made PCB boards as well as the enclosure design. The PCB was designed for the sensors and the enclosure case was designed for the RPI circuit board. The enclosure design keeps both the PCB and RPI stable, so that the components do not fall off as the car is moving.

Intended project component descriptions:

Sensor/Effector 1: The LM393 Speed Sensor. This photoelectric sensor module is designed for high speed rotation counting. The module features a narrow beam IR emitter and receiver directly opposite of each other. When an object is placed in the channel to block the IR, the on-board LM393 chip registers the break and adjusts the output signal to notify the user. The max voltage for this sensor is 5 volts.

Sensor/Effector 2: The Ultrasonic Distance Sensor HC-SR04. This can be used to measure a distance of an object from about 2cm to 450cm away. It sends a soundwave to an object, and bounces back to the sensor to calculate the distance. The max voltage to be used is 5 volts.

Sensor/Effector 3: DC Servo Motors. This will be used to move the car. These motors can run from 4 to 6 volts, and must have an H-bridge such as a L293D. Motor comes with 2x screws and 2x nuts for mounting to the chassis. It is intended to connect to our wheels, and there is a screw to attach once they are plugged together.

We will continue to develop skills to configure operating systems, networks and embedded systems using Firebase and Pyre base. This will allow us to send our sensor and RPI information to our cell phone app.

Our project description can be reviewed by Samuel Martey our collaborator, who is in a position to potentially hire a graduate. He may also attend the ICT Capstone Expo to see our final project, and may be able to refer us to a job that can start off our Computer Engineering Technology experience.

The small components we are using to build are safe enough to be brought to class every week, and to be worked on at home. All of our components and parts kits are in a locker, so we can just pick whatever we need to work on anytime.

To keep our safety in mind, we will bring safety glasses every week when working on the project. We also know the RPI circuit board requires a maximum of 5 volts, so we do not burn out the project and cause any danger hazard. We will also make sure that our circuit board is powered off before we make any connections, and not left unattended when powered on.

Abstract

Accomplishments that have been described by this document, is having our PCB boards ready to power up our sensors, and ready to use for our control car. The reason my product should be purchased and I should be hired by an investor is because, other control cars that have been sold over the years have usually been basic, with minor improvements. I want to take this project of the Mini Robot Rover to another level, with more advanced ideas in technology.

Table of Contents

Declaration of Joint Authorship	2
Proposal	3
Abstract	5
List of Figures.....	9
1.0 Introduction.....	11
1.1 Scope and Requirements.....	13
2.0 Background	15
3.0 Methodology	19
3.1 Required Resources	19
3.1.2 Manufacturing	20
3.1.3 Tools and Facilities	22
3.1.4 Shipping, duty, taxes.....	24
3.1.5 Time expenditure	26
3.2 Development Platform.....	27
3.2.1 Mobile Application	27
3.2.2 Image/firmware	31
3.2.3 Breadboard/Independent PCBs	34
3.2.4 Printed Circuit Board	39
3.2.5 Enclosure	44

3.3 Integration	47
3.3.1 Enterprise Wireless Connectivity	47
3.3.2 Database Configuration	48
3.3.3 Security	49
3.3.4 Testing	50
4.0 Results and Discussions	53
5.0 Conclusions.....	55
7.0 Appendix	57
7.1 Firmware code	57
7.2 Application code.....	59
6.0 Bibliography	60

List of Figures

Figure 1. By Android Studio - https://developer.android.com/studio/ , CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=74094999	30
Figure 2. Initial schematic. This work is a derivative of "http://fritzing.org/parts/" by Fritzing, used under CC:BY-SA 3.0.....	Error! Bookmark not defined.
Figure 3. This work is a derivative of "http://fritzing.org/parts/" by Fritzing, used under CC:BY-SA 3.0.	Error! Bookmark not defined.
Figure 4. Breadboard prototype.	Error! Bookmark not defined.
Figure 5. PCB design This work is a derivative of "http://fritzing.org/parts/" by Fritzing, used under CC:BY-SA 3.0.	39
Figure 6. Humber Sense Hat Prototype PCB.	Error! Bookmark not defined.
Figure 7. Example enclosure.....	44

1.0 Introduction

Self-driving cars are very closely associated with industrial for IoT. Self-driving cars are combined by IoT with other technologies such as local computing, machine learning etc. Which is providing the required technologies for self-driving cars. The most challenging question for self-driving cars is how it is functioning. The control of the wheel is a must. There can be something working inside to make the car work without the wheel. A lot of the sensors, actuators, motors, and controllers are equipped with the car these days. Explanation of surrounding environment and possible changes to those surroundings is one of the main tasks of any machine learning in self-driving today.

The product we will be working on is called the Mini Robot Rover. The idea is to make a control car than can be controlled using an Android cellphone app, with an implemented camera to the car. Whatever the camera sees, should output the display on the phone screen. As the car moves, speed and distance should be displayed as well.

The software we are using to make the application, is Android Studio written in Java. The code to program our sensors and motors are written in Python. The 2 motors which will be used for this project to control the Mini Robot Rover is micro servo body. There will be the importance of the motor driver chip which is L293D, and why it is needed for the project to correctly function. I will talk about the wire connections, and which pins of the GPIO is needed to connect the CPU.

The motor driver chip name of the component for moving motors forward and in reverse is Motor Driver L293D. This chip will function when it is connected to the PCB and it shows the movement of the motors on either direction. The 5 volts can be used from the

raspberry pi. The pin 2 is a VCC pin which supplies 5 voltage from raspberry pi to make the motors work. We also have an option to use 4x AA batteries to make the motors work as well. L293D is a 16-pin IC which can control a set of two DC motors simultaneously in any direction. It means that you can control two DC motor with a single L293D IC. There are 4 input pins for l293d, pin 2,7 on the left and pin 15 ,10 on the right. Left input pins will regulate the rotation of the motor connected across left side and right input for motor on the right side. The data of the 2 motors will be on the raspberry pi. The firebase and pyre base are going to be used to send the data of the 2 motors to the software app. Therefore, it can be used for the app controls to move the car when it is built. At the end of the semester our project will be combined and ready with the sensors and motors to make it transform into a smart control car.

1.1 Scope and Requirements

This is an Internet of Things (IoT) capstone project that uses a distributed computing model of a smart phone application, a database access using Firebase, and a Raspberry Pi (RPI) circuit board along with the enclosure design (3D printed/laser cut). We will also merge our 3 PCB boards to be one custom PCB. One of the limits we have for this project is that the RPI circuit board cannot be powered with over 5 volts. Using a maximum of 5 volts is just what we need for our project to be successful.

Hardware Specifications:

- Raspberry Pi (5 volts max).
- PCB Design (Merged).
- Enclosure Design (Protection case for RPI).
- Control Car Parts (Built).

Android Device Requirements:

- Android 2.2 or Higher
- Average WIFI Connection

Database Specifications

- Firebase and Pyre base Completed.
- Users email and password will be stored into the database.
- User should login and be given a unique id.
- User Authentication
- Real-time database
- Hosting

2.0 Background

Humber has launched a Deep Racer League project which also has to do with a control car. The Deep Racer project uses skills in cloud computing, and our project of the Mini Robot Rover uses skills in Computer Engineering Technology. Since these projects are using a control car, it is possible add more ideas to our project from the Deep Racer, and the Deep Racer team to add ideas from our Mini Robot Rover project.

Skills in cloud computing have become most essential in the modern workforce and Humber College is taking steps it hopes will help fill the talent gap in Canada. (Buckley, S,2019, p.1)

Humber will be launching its Cloud Computing Ontario Graduate Certificate in the fall of 2020, and will be in the partnership of Amazon Web Services (AWS).

Through this partnership, Amazon will be providing Humber with its cloud infrastructure as well as many of the cloud computing tools being used in the course. Such tools include the AWS Deep Racer, a 1/18th scale race car driven by reinforcement learning, and 'go smart with Alexa', an AI training program that creates custom Alexa skills using Amazon Echo. (Buckley Smith, 2019 p.5).

Amazon Echo is a hardware lineup of Bluetooth speakers, displays and alarm clocks. They are all powered by Alexa which is Amazons voice personal assistant that listens and responds to what you ask it to do.

Thanks to Alexa, Amazons Echo devices can play music, read the news or an e-book or an audiobook, report weather and traffic conditions, recite sports scores and schedules, call a ride, shop, check your bank account, give movie schedules, check your calendar,

set a timer, tell jokes, host a game right night and answer questions. It can also connect to your smart home devices. Many of these things will work right out of the box, but you can get the Echo to do even more by enabling Skills. (Kozuch, K, 2019 p.10).

The Deep Racer team wants their control car to have many functions, which is why they are making it work with Alexa and Amazon Echo. Artificial intelligence is also advancing as the years go by. Humber College is about to become an AWS Academy Institution. It will use an AWS Certification Testing Center, which allows teachers to teach the AWS curriculum and explain the testing needed for AWS certification.

When we first came up with our project of the Mini Robot Rover, we found out that the car parts can be built with any mounted circuit board, with sensors added to it. The car parts came with wheels, DC motors, Support Wheel Metal Chassis and Top Metal Plate with mounting hardware. All that was left was for us to use our creativity on what we want the car to do, which is why we chose the Ultrasonic Distance Sensor, and Speed Sensor to work with the Raspberry Pi Circuit board. Just like the AWS Deep Racer project, many ideas can also be added to the Mini Robot Rover project.

References

Smith, B. (2019), p.1. Retrieved from:

<https://business.financialpost.com/technology/tech-news/humber-college-launching-cloud-computing-certificate-in-partnership-with-aws>

Kozuch, K. (2019) Retrieved from: <https://www.tomsguide.com/us/amazon-echo-faq,review-3377.html>

Smith, B. (2019), p.5 Retrieved from:

<https://www.itbusiness.ca/news/humber-college-launching-cloud-computing-certificate-in-partnership-with-aws/113168>

Engineered in NYC Adafruit (2019) Retrieved from:

<https://www.adafruit.com/product/2939>

3.0 Methodology

3.1 Required Resources

For this project, the Mini Robot Rover control car parts are required to be built so we can attach the Raspberry Pi to the control car. This project requires safety glasses, the Ultrasonic Distance Sensor, Speed Sensor, Motors, resistors, L293D Motor Driver, breadboard, Kuman camera and a cell phone to control the car.

The cell phone we currently have is the Motorola Moto G7. The G7 is able to cram a 6.2-inch display into a similarly sized body. It has a dewdrop notch which helps to bring the screen closer to the edges which have bezels thinner than those on the iPhone XR. The display has a nice contrast and looks good in many situations. This phone has a 12-megapixel main camera and a 5-megapixel depth camera. This offers a solid camera experience.

The Kuman camera for the Raspberry Pi circuit board. Has a 1080p OV5647 Sensor HD Video Webcam which supports night vision for all Raspberry Pi models. This is what we will use to mount on the control car. The weight of this camera object is 18.1g. This component is possibly to be implemented depending on the reliability of the project.

The Ultrasonic Distance Sensor HC-SR04 (45mm x 20mm x 15mm) component will be used to measure the distance when the Vcc, Trig, Echo, and Ground pins are connected. This component requires a 4-pin header, along with the 2 resistors which are 1k, and 2k ohms to be connected to the PCB board.

The Speed Sensor module (32mm x 11mm x 20mm) is used to measure high speed rotation counting as the car moves. The LM393 chip registers the break and adjusts the

output signal to notify the user. This sensor has 3 pins for 3 connections for ground out and Vcc and can be used with jumper wires.

The Mini Robot Rover car parts come with 2x Wheels, DC Motors in Micro Servo shape, 1x Support Wheel, Metal Chassis and Top Metal plate with mounting hardware. These parts will be used to build the control car and make space for the Raspberry Pi connection.

The Raspberry Pi 4 (RPI 4) is what we will be using for this project. The RPI 4 Model B is the latest product in the popular Raspberry Pi range of computers. It offers huge increases in processor speed, multimedia performance, memory and connectivity compared to the prior generation of the Raspberry Pi 3 Model B+. This circuit board can be used to power up the sensors you work on for a project. The recommended power for this board is 3 to 5 volts. The main computer language written for the RPI is Python.

The L293D Motor Driver is an IC component which will be important for the project, because it will help with the movement of the car, when it's connected to the 2 motors. This is a 16-pin IC and works on the concept of an H-bridge. In a single L293D chip there are two H-bridge circuits inside the IC which can rotate two dc motors independently. Due to its size, it is very much used in robotic applications for controlling DC motors.

3.1.2 Manufacturing

PCB is an acronym for printed circuit board, which mechanically supports and electrically connects electrical or digital factors the use of conductive tracks, pads and

different points etched from one or greater sheet layers of copper laminated onto and or between sheet layers of a non-conductive substrate. Components are generally soldered onto the PCB to both electrically join and robotically fasten them to it. For our final project of the capstone project we are making a smart control car. The name of our smart control is the Mini Robot Rover. The idea is that we combine our 3 components or devices which will operate as one custom PCB. The combination of the three components are: LM393 speed sensor, HC-SR04 ultrasonic distance sensor, and motor driver (L293d) with 2 motors. The PCB will be created by using the Fritzing software and will be printed out at the prototype lab. We highly recommend for everyone in the future to troubleshoot using cold testing. The cold test should be an important requirement before plugging in the PCB to the raspberry pi. The cold test meaning is that using a multimeter to test the soldering points of each connectivity to find out if it is 100% successfully connected. A multi meter is required to test the PCB to see if there can be any voltage connections running through the copper wires. When the PCB is printed and it is ready to go, we have to connect all of our sensors and effectors. Then we solder all of them, depends on where the soldering points supposed to be connected and where the points should be soldered completely. Once all of that is completed the PCB will be safe to connect on the Raspberry Pi for testing. The Multi-meter can be used to measure the resistance of the resistors, making sure they are the correct components for the PCB. This is to make sure that there is no short circuit that can affect or damage the sensor or the PI. One of the benefits of a PCB board is that all wire connections can be made on just one flat board, which can save a lot of space in an object.

We are going to design our case using the CorelDraw software. We will design our enclosure as 3D printed or as laser cut. The CorelDraw software is used create the design of the enclosure. The enclosure is a case that surrounds a Raspberry Pi CPU keeping the hardware safe and secured. We would measure the dimensions of our PCB with our Raspberry Pi and create the enclosure on CorelDraw with all correct dimensions. The design of our enclosure will vary, depending on the size of the PCB. Since, our PCB and enclosure design case has not been created as of yet, it will take hard work and dedication for our project to put everything together to get it all completed and ready for testing. Our smart control car would be sold in any market, the price will depend on the manager or customers.

3.1.3 Tools and Facilities

The tools used is the screw driver, screws, bolts, Multi-meter with lead wires to measure voltage and resistance, Soldering Iron Kit and soldering wire. The facilities used is the Humber prototype lab.

The Multi-meter is an electronic measuring device that combines multiple measurement functions in one unit. Multi-meters are capable of many different readings, depending on the model. The testers measure voltage, amperage, and resistance and can be used to check continuity, a simple test to verify a complete circuit or PCB board. Advanced Multi-meters may test for values like Alternating Current, Direct Current, Resistance, Capacity, Conductance, Decibels, Duty cycle, Frequency, Inductance, and Temperature. When testing a PCB board with the Multi-meter, we can put it on a beep

setting mode to see if current is passing around the board. If the Multi-meter beeps, it means current is passing through the board. If it does not beep, then there is no current.

The soldering iron is a tool with a metal tip that can heat up to about 800 degrees Fahrenheit, but the temperature is adjustable. This tool's job is to transfer heat to things like wires, transistor leads, and pads on PCB boards. After the important areas are heated properly, solder is applied. A good soldering kit can be brought for \$30 to \$40, which will give you a longer lasting tool that will work for a much wider variety of applications, with proper heat control to boot. There are also soldering guns available, but they are only used to repair thick cables and never on PCBs. When soldering, we must be sure to wear safety glasses, keep loose clothing and hair out of the way, and be careful with your fingers. Protective gloves can also be used when soldering.

Solder is a thin tube, usually rolled and made of various metal. It is used to hold the individual components together. For computer electronics, we are usually looking at 60% tin and 40% lead. Lead free solder is also available, but it has higher melting temperatures, meaning a better soldering iron will be needed in order to remove it. Lead-free solder is better for the environment and has more benefits. A wet sponge also comes in handy to wipe off the solder after the iron is heated up. This will help with cleaner and faster soldering on a PCB board.

The Humber prototype lab is a room with many tools and equipment that can be used for Computer Engineering and Electronics Engineering students to complete capstone projects. The prototype lab also has 3D printers, and laser cut printers. This can help us to adjust any specific measurements we need to properly fix the project. Teachers in the prototype room can also show you the best software to use, depending on what kind of

project you are working on and may have ideas on how you can improve on your project. They can teach you how to use software like Corel Draw to print a laser cut case and Fritzing to make an organized and connected PCB board.

3.1.4 Shipping, duty, taxes

The Ultrasonic Sensor was purchased for a pack on Amazon for \$11.59 with free shipping. It can be shipped to your location the next day after purchase. There are a few shipping options for Amazon which are, Standard Shipping for 3 to 6 business days, Express Shipping for 2 to 4 business days and priority shipping for 1 to 3 business days. 1-day shipping is what we used. The Raspberry Pi 4 was purchased for \$90 and can only be returned within the first 30 days of purchase. The Speed Sensor has a price of \$2.50 with a shipping time of 30 days, but an option to purchase it for \$30 for a shipping time of 3 days. It was brought from the website pihut.com. If within 14 days of your purchase you decide you aren't happy with the product you can ship it back to them for a refund. The Mini Robot Rover Chassis kit – 2WD with DC Motors comes with a price of \$24.95. It is sold from the site adafruit.com and takes 2 to 3 weeks to arrive after purchase. This site has many distributors like EXP-Tech, Chicago Electronic Distributors, The Pi Hut, Core Electronics, S.O.S Solutions, Kiwi Electronics and Makersify. These are the sites and companies Adafruit works with in order to sell electronic parts that can be used for a project. Adafruit has 3 shipping options such as, USPS (United States Postal Service), UPS (United Parcel Service) and DHL (Dalsey, Hillblom and Lynn). USPS is usually cheaper and does not provide detailed tracking. For domestic orders, UPS is sometimes faster and a little more expensive, but has a

high-quality tracking system. The tracking information is emailed when the package is shipped. When in doubt of choosing a shipping method on Adafruit, DHL or UPS are recommended. DHL is available to many countries worldwide across the globe. If DHL shipping is available in your location, the option will show at checkout.

3.1.5 Time expenditure

We will have work time and lead time for this project. We have our Computer Systems Project course 3 hours every week and will come out with a total of 56 hours by the end of the semester when the project is completed. During our 3-hour course, we will use that time to work on our project by building the control car and putting all of are purchased parts, components and materials to use. Outside of our Computer Systems Project course, we will have the rest of the week to continue to develop our user interface design in our mobile application and making the app easier to understand for anyone using it. This time can also be used to work on our reports and come up with ideas on how to improve on our project. With our lead time, our total will be more than 56 hours when we include the time working on the project outside of the class. Some of the components we purchased like the Speed Sensor, is also sold on Amazon and can take 15 to 20 days to be shipped from China to Canada since the speed sensor module takes a few days to manufacture and build. The speed sensor comes from a Chinese company called All Dazzling. The Distance sensor is sold by a company called SunFounderCA. They have sold products in many countries like the United States, Germany, Canada, the United Kingdom, France, Italy, Spain, Japan, etc. They have a pack of products that are ready and made to be sent to your location after 1 day of purchase.

3.2 Development Platform

In this section I will talk about the development platform of our mobile application and the details of how we made our android app.

3.2.1 Mobile Application

I will explain how we made our Mobile application, along with login activity, data visualization activity and action control activity. I will also talk about what software we used to plan how we wanted our Mobile app to look and what language we used to make the actual Android application.

Before we made the Mobile Application, we used the mockup software to plan how we want our android app to look. Mockups are one of the most effective ways of communicating visual requirements clearly. Making decisions about fonts, color schemes, brand assets, content layout and navigation pattern styles will be worked out on a mockup screen. That way, we can have a good idea of what we want our Mobile app to look like when we create it. With the mockup design, you may also be able to realize what User Interfaces you may want, or don't want when you create the android app. This will help us make our Mobile app in a more organized fashion.

After our mockup design was complete, we used Android Studio to make our Mini Robot Rover Mobile application. Android studio is the official integrated development environment for Googles Android operating system. This software is specifically designed for Android development and is available for download on Windows, macOS and Linux based operating systems. It is written using Java and XML. We used Java for the Android functions and XML for the User Interface design.

We created the app in a way that the user should be able to successfully register an account and login, to be directed to the control buttons page. When the user is logged in, they should see a welcome page that gives some information on how the app works. They will then be directed to a short instructions page which will show them how to move the car. The user should have access to all the product functions when they are connected on a WIFI network with access to the internet.

We made the welcome page using XML. We added a picture to that page by downloading a picture from google images and putting it in the drawable folder. That way we were able to drag the picture setting to the android page and add the picture we downloaded. In XML it's called Image View. The 2 buttons on that page are the next button and the skip button. The next button turns to the next page of the app and the skip button skips to the last page of the app. This is done using intent functions in Java, which allows you to click and load from one page to another. We also have the color green for most of our pages on the app.

After the welcome page, we have the instructions page. The paragraph of instructions was made using the XML Text View, where the written paragraph was saved as a string. This helps to save space in the code since the paragraphs will be stored in the strings.xml file. To get an image icon for the app, we downloaded an image and opened it in the app as an image asset.

For the login activity, we used Firebase SDK Authentication for logging users into the application. This is used to authenticate users with their email address and passwords. The Firebase Authentication SDK provides methods to create and manage users that use their emails and passwords to sign in. It also handles sending password reset

emails. After the user signs in for the first time, a new user account is created and linked to the credentials the user signed in with. This new account is stored as part of the Firebase project, and can be used to identify a user across every app in our project. In the Firebase Realtime Database and Cloud Storage Security rules, we can get the signed in users unique id from the auth variable and use it to control what data the user can access.

For the Data visualization activity (Figure 1), we have our file Controls.xml to show the control buttons for the car as arrows with speed and distance showing at the top. The arrows will be used for forward, backward, left, right and stop. This is the main page for our android application, where the Mini Robot Rover will be controlled from. So far, we can click the arrows and display speed and distance at the top so it shows information like a virtual control car.

For the Action control activity, we have Java functions for direction, motion and rover Name set in the firebase. The app will separate forward and back motion (throttle) from the left and right motion (direction). When forward is pushed, the app will notify firebase that forward is pushed and this will remain until a stop action or another throttle or direction is pushed by the user. Updating rpm and direction from firebase is loaded by the app. The simulator can read the user actions and fake the position of the rover. In our Controls.java file, we also have an onClickListener to process which button was pressed and an Event Listener to process when the button was pressed by the user. Data visualization and Action control activity both work together. This is how we made our Mobile application.



Figure 1. Mini Robot Rover Mobile Application

3.2.2 Image/firmware

Hardware

The hardware we are currently using is the Raspberry Pi 4 model B CPU, and the Mini Robot Rover control car. We will be using a 4 to 6 volts battery back to power up the motors, and a portable charger to power up the RPI CPU. The portable charger must have a type C port or cable to be usable.

Firmware

The Firmware is a software program that receives a set of instructions for the hardware device communication, which will then perform an expected operation. The 64 GB SD card of the Raspberry Pi has been formatted. The RPI 4 CPU uses the NOOBS operating system installer containing specifically Raspbian operating system to carry the code and connections of the hardware. All of the files for Noobs have been copied and pasted into the SD card for the setup.

We then inserted the SD card into the Pi after the image has been put on it. For this project, the Firmware is the part of the RPI 4 that is responsible for reading the information from the Firebase to the CPU. Pyre base is the part of the Firmware we will be using to read the values from the Firebase when the buttons on the screen are pushed, and will perform the movement operation of the Robot Rover control car. The Pyre base Firmware is written with the Python 3 programming language to carry out the task of our project, and is capable of using several Firebase services. Pyre base is a good asset to our project because of the capability.

Our Firebase has been setup to carry out the task on the android phone when the button is pushed, showing the speed in rpm, and the distance in cm. The Pyre base will be able to connect to the motor for when the movement task is being executed. To install the Pyre base, the command “pip install pyre base” will need to be entered in the terminal.

For the Python code to work on our sensors we had to make sure the GPIO(General-Purpose-Input/Output) had been installed on the RPI 4. The GPIO is responsible for the communication of the sensors to perform their required task. It will notice the location of the GPIO pins that are on the Pi, and execute with the use of the Python programming language. The command to install the GPIO is “sudo pip install rpi.gpio”. This command is to upgrade the packages of the GPIO.

The I2C tools interface of the RPI has also been enabled. I2C is responsible for supporting the interface of the GPIO header. This is to allow the connection of the sensors when running their code on the RPI. Without the I2C being enabled, the hardware components will not function the way it is supposed to when running the output command of the sensors. The command “sudo raspi-config” must be entered in the terminal to bring up the software configuration tools. We used the arrow keys to select the interface options clicking yes to enable the I2C. When the Firmware of the project is assembled correctly, we should be able to move the Mini Robot Rover by using our physical android phone as a remote controller. The user should be able to use the phone in a way that people can use regular controllers to move control cars. In this case, it will be touch screen control buttons sending the signal.

Remote Desktop

The code for this project can run on a Remote Desktop connection. A Remote Desktop service allows users to access a host of a computer over a network, giving the authentication of resources being controlled. This will allow us to work on the RPI from any location, whether at school or at home.

Wireless Connectivity

For this project, we will be using WIFI connectivity for the control car and the RPI to function. WIFI operates at a faster rate than Bluetooth, so it will be more reliable. It will also avoid interfering with other connected devices and have a stronger range.

Sensor Code on Repository

The code for our sensors for speed distance and the movement can be found on the repository. The filenames are ds.py, motors.py and rpm.py. The code has the connections of the GPIO pins and the functionality.

<https://github.com/Darrelasare/Computer-Systems-Project>

3.2.3 Breadboard/Independent PCBs

I will explain how we made our hardware, along with of our connectivity of our sensors/actuators. I will also talk about what software we used to plan how we wanted our hardware of our PCB to be designed as.

Before we made the hardware, we used the Fritzing software to plan how we want our design to look like. Fritzing is an open-source initiative to develop amateur or hobby CAD software for the design of electronics hardware, to support designers and artists ready to move from experimenting with a prototype to building a more permanent circuit. That way, we can have a good idea of what we want our hardware to look like when we create it.

PCB is the hardware we are currently using for our project. PCB stands for printed circuit board mechanically supports and electrically connects electrical or electronic components using conductive tracks, pads and other features etched from one or more sheet layers of copper laminated onto and/or between sheet layers of a non-conductive substrate. For our final project of the capstone project we are making a smart control car.

The name of our smart control is the mini robot rover. The idea is that we combine our sensor/actuators which will operate as one custom PCB. The sensor/actuators we are using are: LM393 speed sensor, HC-SR04 ultrasonic distance sensor, and motor driver (L293D) with 2 motors.

Sensor/Actuator 1: The Motor Driver. For our connectivity of the L293D, we added data wires from the GPIO pins to the L293D. GPIO 25 of Pin 22 connects to L293D Pin

1 and GPIO 24 of Pin 18 connects to L293D Pin 2. GPIO 23 of Pin connects L293D of Pin 7. For the first motor connectivity, wire 1 is connected to L293D pin3. The motor wire 2 is connected to L293D pin6. For the second motor connectivity, GPIO 11 pin 23 connects to L293D pin 10. GPIO 9 pin 21 connects to L293D pin 10, and GPIO 10 pin 19 connects to L293D pin 15. The new merged PCB will be created using the Fritzing software and will be printed out at the prototype lab.

Sensor/Effector 2: Ultrasonic Distance Sensor HC-SR04. This can be used to measure a distance of an object from about 2cm to 450cm away. It sends a soundwave to an object, and bounces back to the sensor to calculate the distance. The max voltage to be used is 5 volts, but we used resistors to bring it down to 3 volts for safety purposes.

Sensor/Effector 3: The LM393 Speed Sensor. This photoelectric sensor module is designed for high speed rotation counting. The module features a narrow beam IR emitter and receiver directly opposite of each other. When an object is placed in the channel to block the IR, the on-board LM393 chip registers the break and adjusts the output signal to notify the user. We used 3 wire connections for ground, out, and voltage. The max voltage for this sensor is 5 volts, but on our merged PCB we brought it down to 3 volts.

With the Fritzing software, we may also be able to realize, what size of the PCB we would require, so it would fit perfectly in our Raspberry pi. The design of the PCB may come in a smaller size, medium, or large size. It would only depend up on by how many sensor/effectors connectivity it would have. Before, we submit our PCB to prototype lab, we would have to make sure, and it is not overlapping. The meaning of this is that Fritzing has a tendency to go through errors when traces and pads are close

together, the upper wire pad which is yellow and the orange wire pad which is the lower one.

Before sending it to the prototype lab, we would make sure that the wires are not overlapping. The design rules check tells you where the problem is of the wire padding and how to fix them. In order to get this error out of the way, is make sure that upper wire is not touching the other upper wires and not close together. The upper wire should be touching the lower wire. And the lower wire should be touching the upper wire.

Once it is printed out from the prototype lab and it's successfully fixed, we would check the connectivity at where the points have to be soldered, including the sensor/actuators as well. When that's all completed, we would do the cold testing by using the multi meter which tells us if all the connectivity is all connected and soldered successfully. When that's all done it will be ready for testing. This is how we made our hardware for our final project.

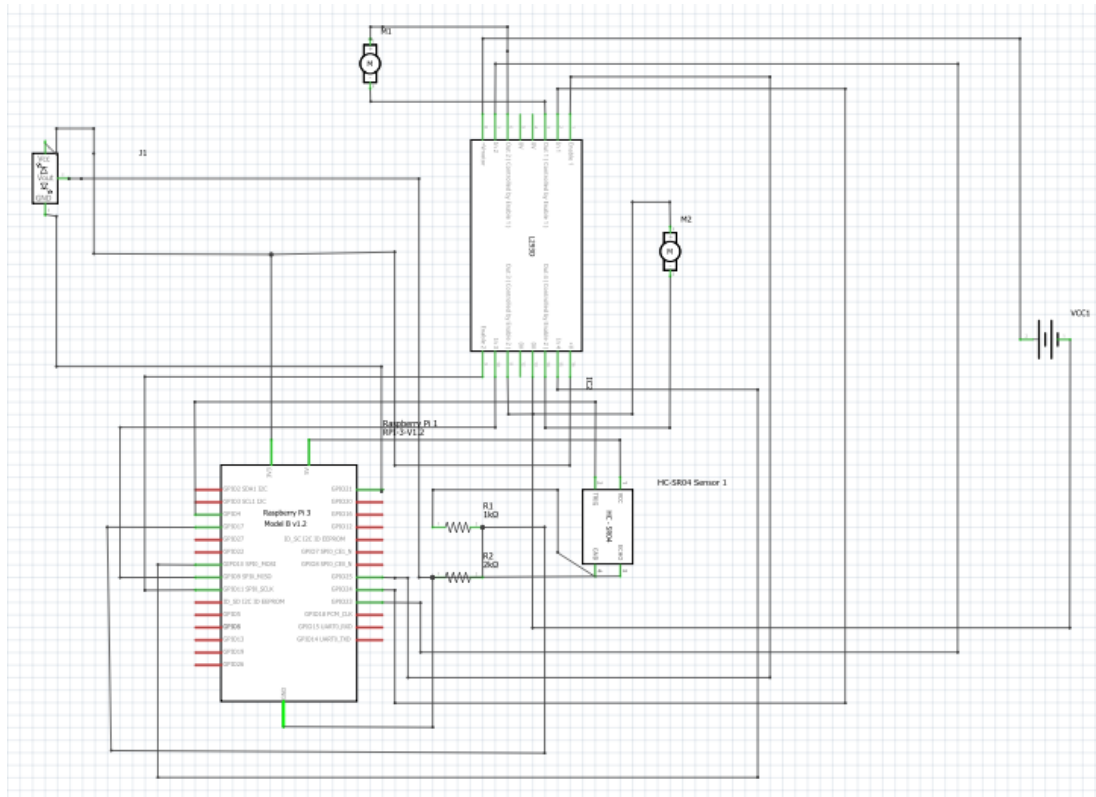


Figure 2. Initial schematic. For 2 motors with L293D motor driver

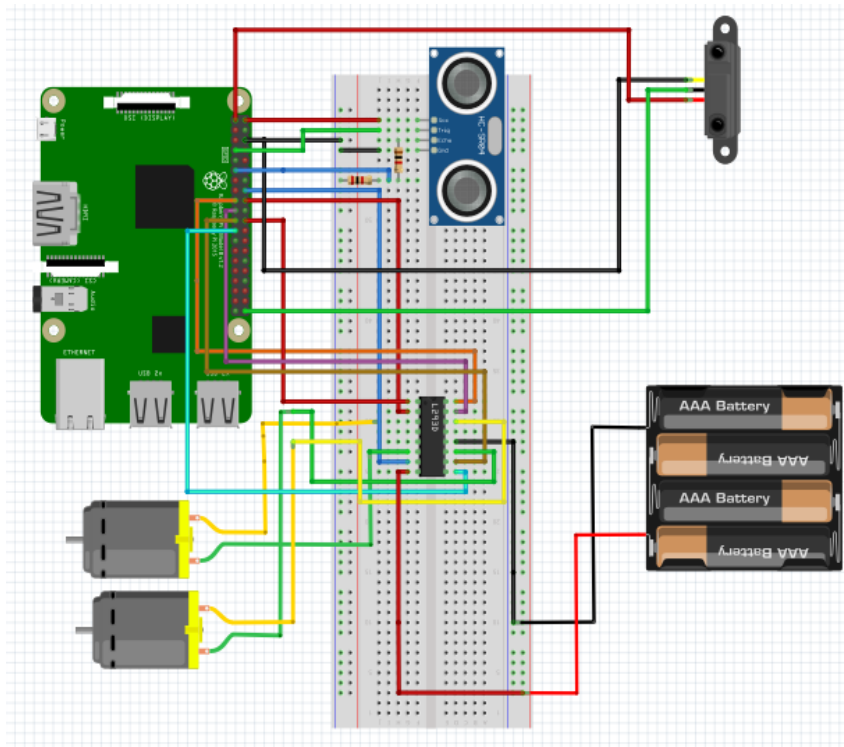


Figure 3. Merged Breadboard for all sensors

3.2.4 Printed Circuit Board

We made our PCB board by merging the distance sensor, speed sensor and motors on one pcb. We connected to all the gpio pins needed on the Raspberry Pi to power up our sensors. To use a safe amount of voltage, we had the speed sensor and the motors connecting to 3 volts, and the distance sensor connecting to 5 volts with the resistors bringing it down to 3 volts. Everything on this PCB board is connected and passed the multimeter test. All our sensors and one motor are working on this PCB, while the other one is not. There could be one minor mistake on the L293d chip where one connection is shifted down. Below is a screenshot of the PCB board we worked on.

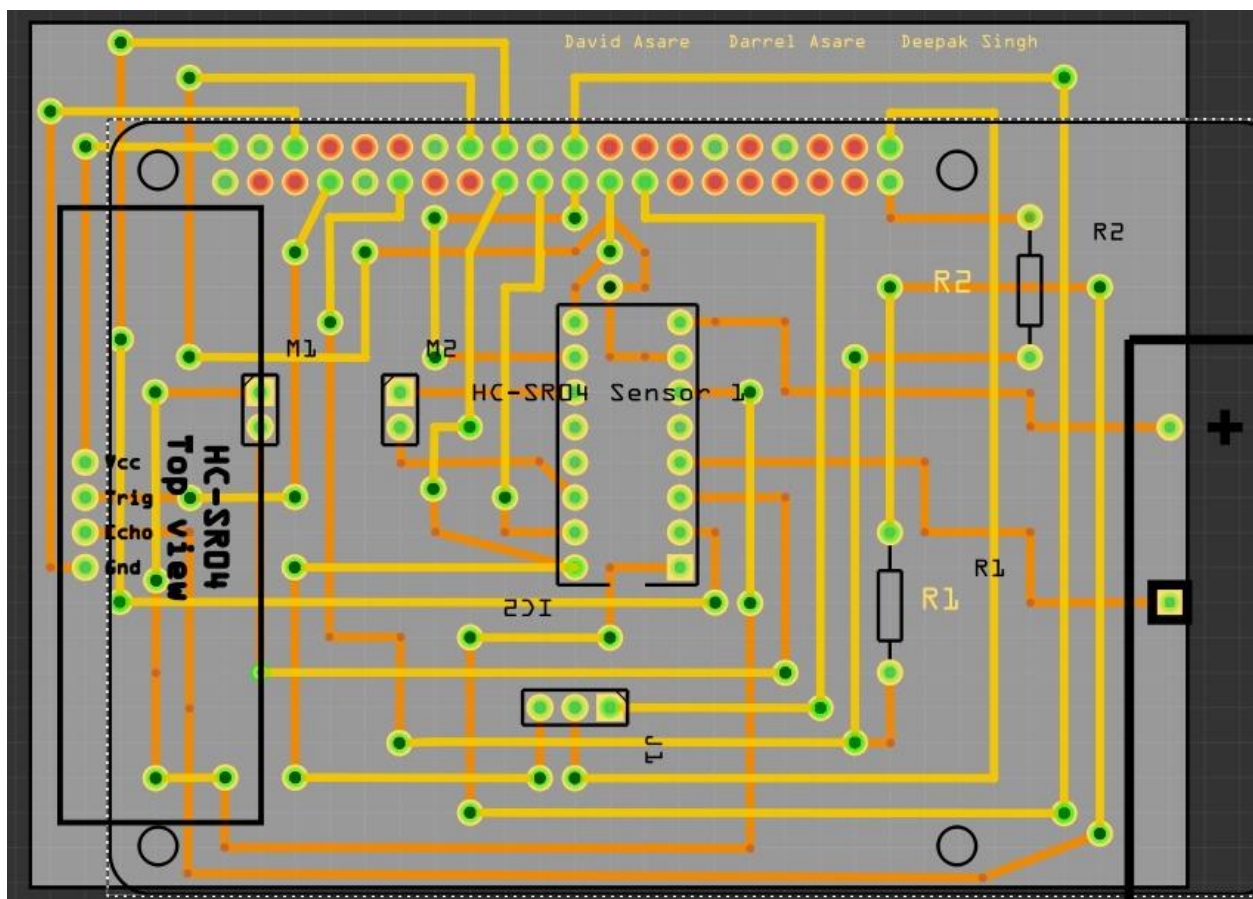


Figure 4. Merged PCB derivative of "<http://fritzing.org/parts/>" by Fritzing, used under CC:BY-SA 3.0.

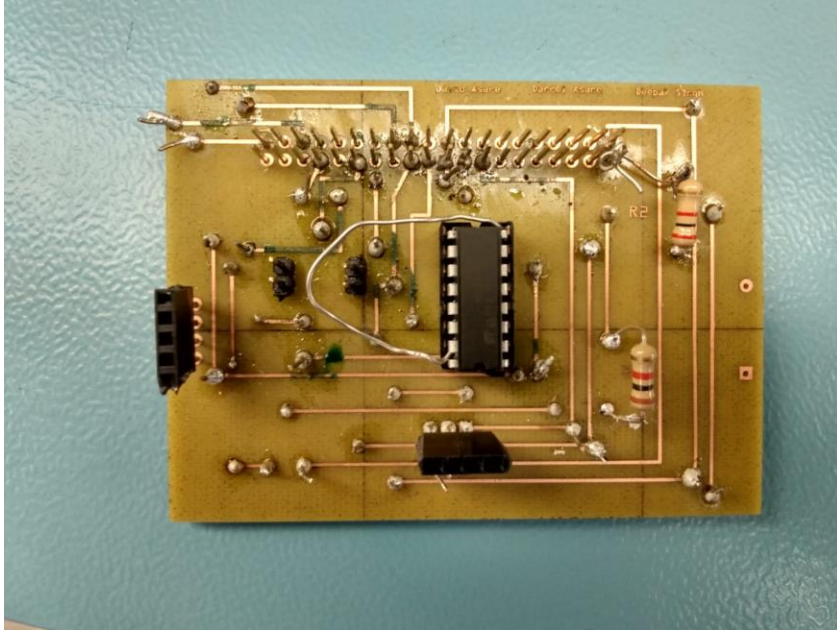


Figure 5. Humber Printed Circuit Board.

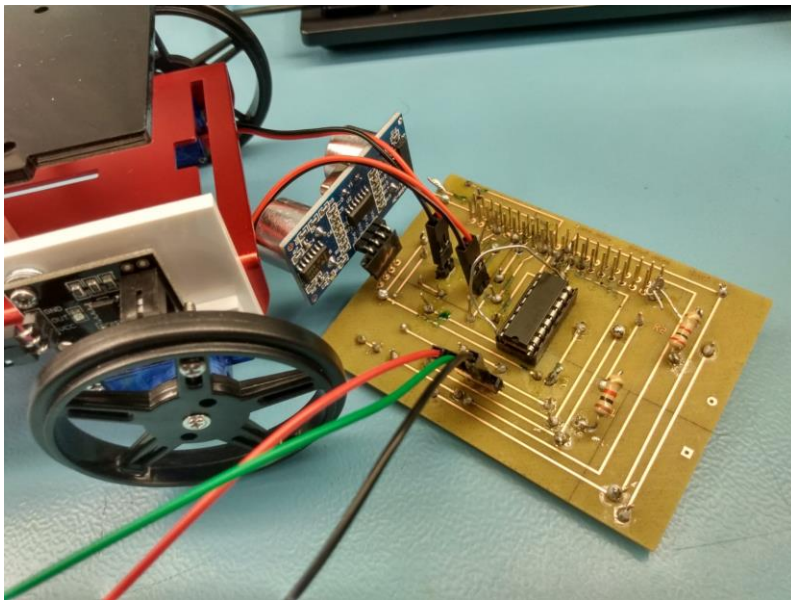


Figure 6. PCB with car connection.

We also made another PCB that is simpler, takes less space, and uses less via holes. First, we used a 5-pin hole header on fritzing, so we can connect a Micro USB header to the board when it is printed. That way, the Raspberry Pi will need only the USB Phone Charger, and not the battery pack. When we printed the new PCB board, the 5-pin

holes were too small for the Micro USB header to fit, so we corrected the size by adding a 6-pin header on Fritzing. Only 5 holes will be needed. With this new PCB, we may have to update the code to get the motors working. This PCB passed the Multimeter test. Below is a screenshot of the new PCB we made on Fritzing, which now works with the Speed Sensor, Distance Sensor and Motors.

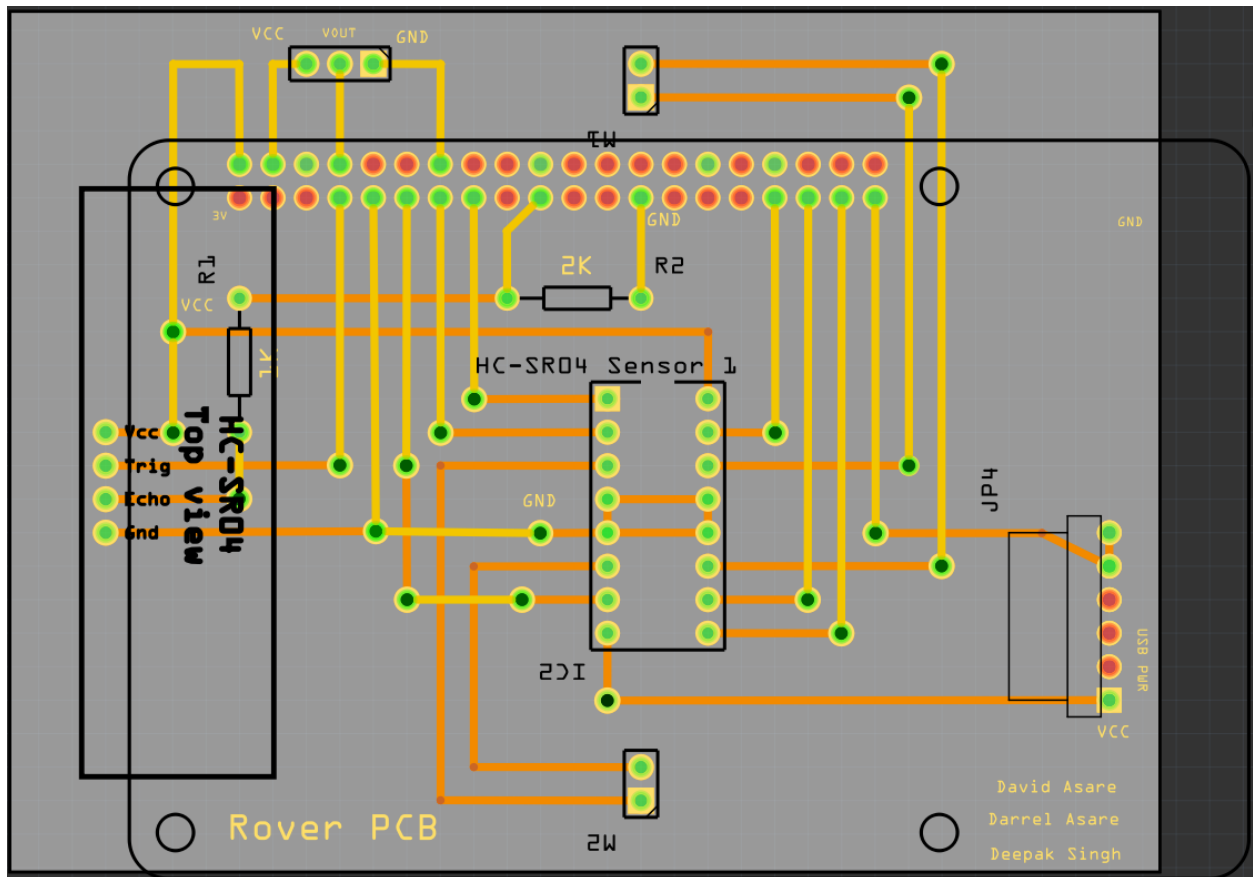
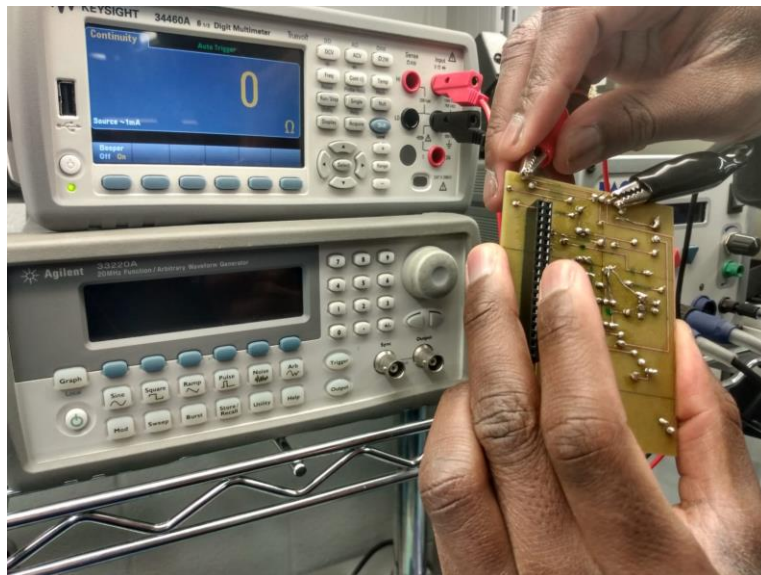
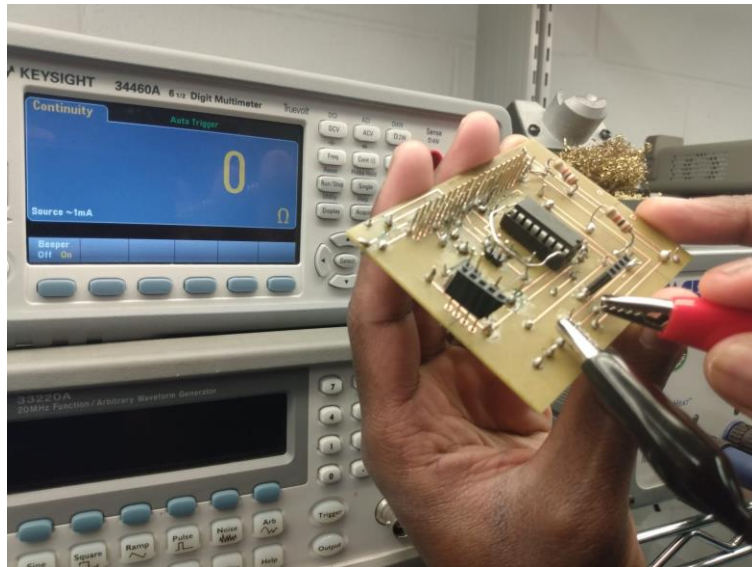


Figure 7. PCB new design.

Multimeter test pictures for Figure 4 design.





3.2.5 Enclosure

I made the prototype case by using the CorelDraw software, where I increased the measurements of the Raspberry Pi 4 case. This can help keep the RPI and PCB stable as the car is moving with speed and allows for more protection if the car hits objects. Below are pictures of the enclosure mounted on the RC car with the working PCB attached.

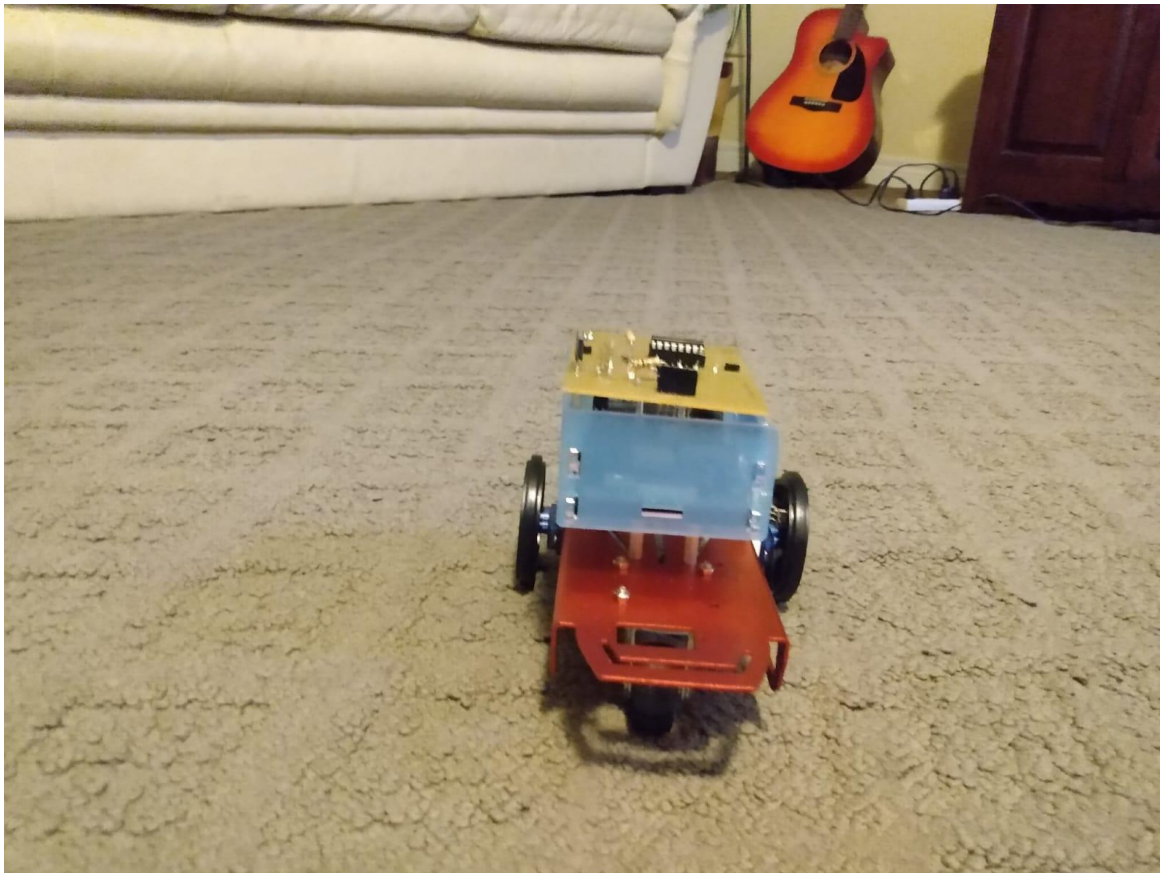


Figure 8. Enclosure Front Side.

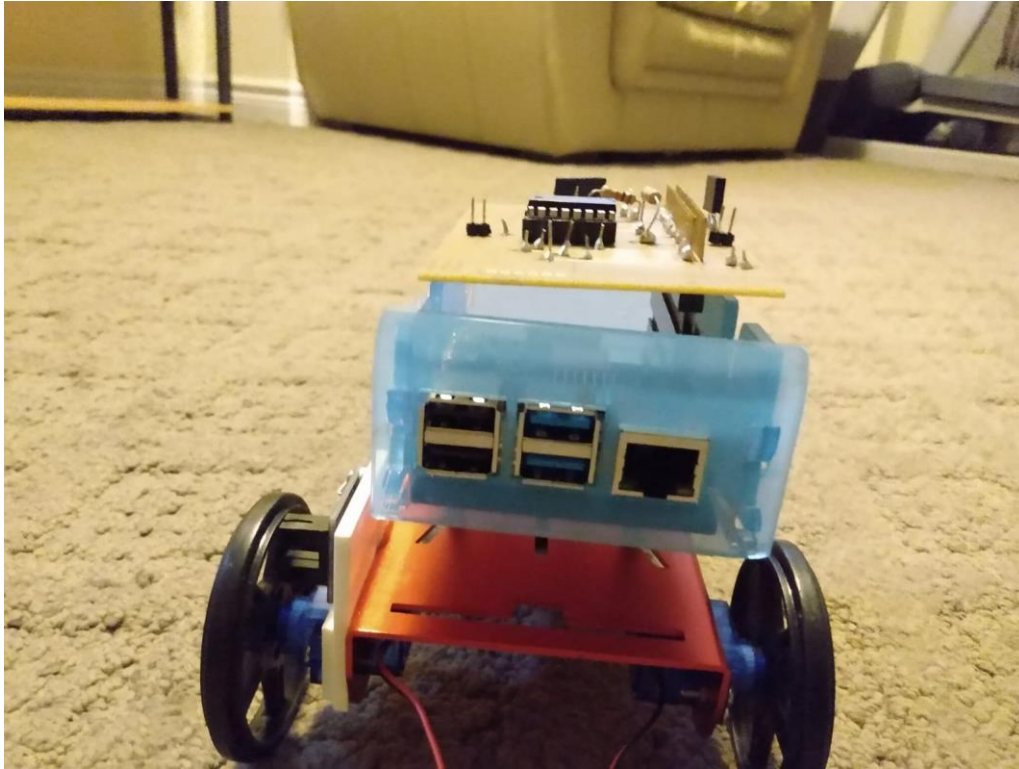


Figure 9. Enclosure Back Side.

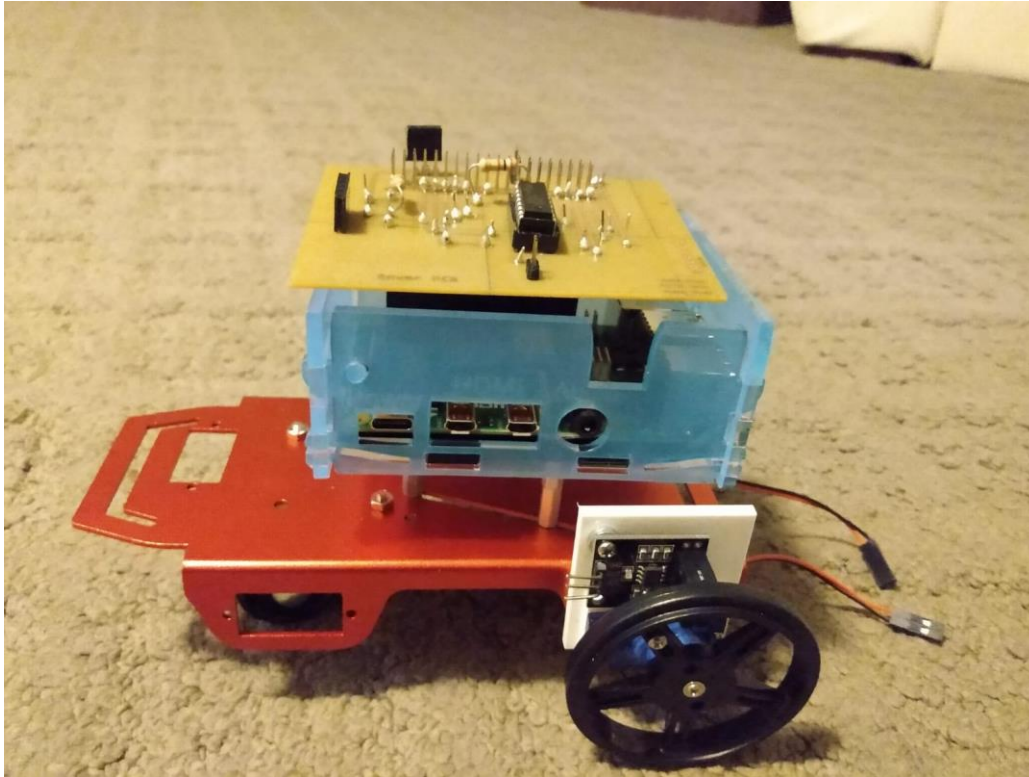


Figure 10. Enclosure Left Side.

3.3 Integration

3.3.1 Enterprise Wireless Connectivity

How did you make a Database accessible by both your Prototype and Mobile Application?

To make the database accessible by both Prototype and Mobile Application, we had to get Firebase and Pyre base to work together. Firebase was connected to the Mobile App using the google-services.json file. Pyre base is a python wrapper for the Firebase API. It allows you to use python to manipulate your Firebase database, authentication and storage. In order to get Pyre base to connect to the Firebase, we first needed to install Pyre base and its dependencies using the `sudo pip install Pyre base` command. After, we imported the Pyre base Module into the python file to make the connection to our Firebase database, storage and authentication. After that, we were ready to implement our Python code for the speed sensor, distance sensor and motors to the Pyre base, so the signals can be sent to the Firebase real-time database, and the car can be controlled from the Mobile Application.

3.3.2 Database Configuration

For the Database configuration, we used the CRUD operations for Create, Read, Update and Delete. Bellow shows how we did the operations.

Create: get a reference to the auth service, log in a user, get a reference to the database service, save data and pass the users idToken to the push method.

Read: Get a reference the auth service, login the user, get a reference to the database service, retrieve value and key.

Update: Get reference to the auth service, login the user, get a reference to the database service, and update preexisting data.

Delete: Get a reference to the auth service, login the user, get a reference to the database service, and remove preexisting data.

This was used to configure the database.

3.3.3 Security

To login to our RPI desktop, we use VNC Viewer. VNC has a few security principles. They do not record your sessions, and data cannot be decrypted now or in the future. Every connection is treated like a hostile environment, and the owner of the remote computer can decide who is able to connect. When you sign up for a RealVNC account, the email and password are important and should not be shared with anyone. You can sign in to VNC Viewer to remotely access computers, by entering the ip address. The account password should be at least 8 characters long and should be different from the VNC server password. To maximize the security of VNC, you will need to enable the 2-step verification on the RealVNC account on the security page online. In order for the VNC Viewer to connect remotely to the RPI desktop, the `wpa_supplicant.conf` file also needs to be configured in the RPI. This is how we can connect to the internet and get the address of the RPI we can use to connect anytime the RPI is powered on.

3.3.4 Testing

Unit testing and Production testing.

To do our project, we first had to test the all of our sensors on our individual PCB boards to see if they were all working correctly. The sensors we tested was the LM393 Speed Sensor, the HC-SR04 Distance Sensor, and the DC Servo Motors. All of the test was successful on our own PCB boards, so we were ready to plan on how we can merge our PCB connection in to one PCB board to use for the Mini Robot Rover.

We used the Fritzing software to design our first merged PCB. We started with the schematic diagram where we brought out all of our sensors, and connected our pin connections the same way we did on our individual PCB boards. The only difference is that the Speed Sensor and the Motors were connected to 3 volts instead of 5 volts so we avoid burning out the RPI circuit board. We then organized all the wires on the PCB to make sure there was no overlaps, and the PCB was ready for production.

After we printed our merged PCB board, the Distance Sensor worked perfectly, but the Speed Sensor and Motors were not working that well. We tested the PCB with the multimeter, everything beeped and said it was connected, but only one Motor was working. For the Speed Sensor, the PCB was not detecting a connection, so we had to hard wire it to get it to fully work. We still could not figure out the issue with the Motor connections, so we decided to print out a new merged PCB board that is simpler, uses less wiring, and less via holes to solder.

When we printed our second merged PCB, soldered the board, did the multimeter test and everything was connected. We then tested it on the RPI circuit board and the Speed Sensor, Distance Sensor, and the 2 DC Servo Motors were all working correctly.

On this new PCB, we also added a 6-pin hole header which can be optionally be used for the USB Power Bank instead of the battery pack. That way the Power bank can power up both the RPI and the DC Servo Motors. The new working PCB board is shown on Figure 7.

4.0 Results and Discussions

Is your prototype perfect? What did you learn?

Our prototype is now perfect because we got our project to work, and it does everything we planned. The RC car and our Mobile Application is now stable and easy to use.

During this project, we learned how to program in python to get are sensors working for the Raspberry Pi circuit board. We also learned how to make Mobile Applications to connect with Firebase, Pyre base and the RPI. We learned how to program the app using java, and how-to setup a real time database for the Mini Robot Rover to connect to. We learned more about data structures, databases, constructors, authentication, user interface design and their important role for projects. The troubleshooting experience taught us to be more aware of the project designs, and connections.

Troubleshooting experiences improved our skills by implementing our ideas like re-soldering the PCB. We also learned more about the importance of electronics like voltage and current, and how we can manipulate it to work with all of our sensors together and correctly.

5.0 Conclusions

We will explain the results and discussion of our project. When we finished the Mini Robot Rover, we got our expected results which is having our Merged PCB board to connect the Speed Sensor, Distance Sensor and Motors on one PCB. When we connect our PCB to the RPI, we can turn on the Power Bank switch to power on the RPI. After that, we can start our Mobile App to control the car.

In conclusion, it was a good experience learning more about how to make IOT (Internet of Things) projects since they are widely used in the workforce today. We now have a better understanding on how software and hardware can work together to make important projects. We can now use Firebase and Pyrebase to enhance our learning in the computer engineer field. We have experienced both the hardware and software side of how objects can work, based on the standards of technology.

7.0 Appendix

7.1 Firmware code

Problem with PCB

The progress of this Mini Robot Rover Project is going well despite the one error we have, needing both motors to function at the same time. The distance sensor and the speed sensor function correctly on the PCB, with one motor getting a signal to turn the wheel.

Solution to PCB

When we made another Merged PCB board, we connected all of our sensors together, but with smaller and simpler connections. We connected the wires with the shortest distance possible, and the 2 motors started working correctly. We also made sure the code for the motors were correct and where connecting to the right pins on the RPI circuit board.

Problem with Enclosure Case

For the Mini Robot Rover, we made blue enclosure design case, that was slightly off in the measurements. The height of the case was tall, but made it difficult to plug in the PCB board to the RPI circuit board. It also caused more weight to the car, which can cause it to move slowly.

Solution to Enclosure Case

The solution to the enclosure case is the black case, which is shown on the Mini Robot Rover Poster. The case is shorter, which allows us to plug in the PCB board easily. The case will also be lighter on the car allowing us to move the car faster. The type of RPI

we are using is the RPI4, but we made this enclosure in a way that can cover both the RPI3 and RPI4 circuit board.

Financial Update

With our financial update, we brought the battery packs which were \$10 and comes with 2 packs. We also brought a USB phone charger which was \$29.99 on Amazon. This is what we brought for our project this semester. The battery pack will be used to power up the motor driver chip, while the USB phone charger will power up the Raspberry Pi.

These are the links to our modified code files.

Link to sensor code: <https://github.com/Darrelasare/Computer-Systems-Project/tree/master/Firmware>

7.2 Application code

We used Firebase Authentication for our login activity. Firebase Authentication provides backend services, easy to use SDKs, and ready to go user interface libraries to authenticate users to the app. It supports authentication using passwords, phone numbers, and identity providers like Google, Facebook, and twitter. For this app, all we need is the email address and password to login. When the email and password is registered, it is stored in the firebase. The password needs to be at least 6 characters long to login to the app.

In our Mini Robot Rover app, the Data visualization and Action control are in the same activity. The Data visualization allows for us to have the rover information stored on one mode. We get rotations per minute and distance when the car moves, and it is displayed at the top of the app screen, shown in a black rectangle. This was made from XML, where we used the Linear Layout class to draw the rectangle, where the speed and distance will be displayed. In Controls.java, we used our SetUpDataListeners class to calculate the speed and distance. We use the Firebase real-time database to stream the values to and from the rover. For the Action control, we have 4 arrows used to go forward, backward, left and right. This was drawn in XML using the Image Button class. In Controls.java, we have functions that can process the button presses on the view, and an event listener to all the buttons so that we can process when the user presses a button. All of these actions are connected to our Firebase database values. Below is the link to the complete code in the repository.

https://github.com/Darrelasare/Mini_Robot_Rover

6.0 Bibliography

Bosch Sensortec. (2019, July). *BME680 - Datasheet*. Retrieved from Robert Bosch GmbH: https://ae-bst.resource.bosch.com/media/_tech/media/datasheets/BST-BME680-DS001.pdf

Kinsella, J. (2019). Five trends predicted for the cloud industry in 2019. *Software World*, 50(1), 11.

Media, O. (2019). *O'Reilly artificial intelligence conference 2019 - San Jose, California*. California: O'Reilly Media, Inc.

OACETT. (2017, March). *I need to Complete a Technology Report*. Retrieved from The Ontario Association of Certified Engineering Technicians and Technologists: <https://www.oacett.org/Membership/Technology-Report-and-Seminar>

Robuck, M. (2018, 11). AWS goes deep and wide with machine learning services and capabilities. *Fierceinstaller*.