

# Rock 'N' Rover: RC Car

## **Group 15:**

Laila El Banna (*Computer Engineering BSCpE*)

Sanya Wadhwa (*Computer Engineering BSCpE*)

Ryan Kohel (*Electrical Engineering BSEE*)

Michael Patalano (*Electrical Engineering BSEE*)

## **Reviewers:**

Sonali Das

Di Wu

Saleem Sahawneh



# **Table Of Contents**

<b>1. Introduction.....</b>	<b>7</b>
<b>1.1 Executive Summary.....</b>	<b>7</b>
<b>2. Project Description.....</b>	<b>8</b>
2.1 Motivation and Background.....	8
2.2 Goals and Objectives.....	9
2.3 Features/Functionalities.....	9
2.3.1 Customer Input and Market Analysis.....	10
2.3.2 Advanced Mobility.....	10
2.3.3 User-Friendly Interface.....	10
2.4 Prototype Illustration.....	11
2.5 Engineering Specifications.....	12
2.6 House of Quality.....	13
2.7 Initial Project Diagrams.....	13
2.7.1 Initial Hardware Diagrams.....	14
2.7.2 Initial Software Diagrams.....	14
2.7.3 Revised Software Diagram.....	16
<b>3. Project Research.....</b>	<b>17</b>
3.1 Similar Past Projects.....	17
3.1.1 Hiwonder Smart Robot Car.....	18
3.1.2 DEERC RC Car.....	19
3.1.3 Arduino Esplora.....	19
3.1.4 PlayStation 4 Controller.....	20
3.1.5 Integration into Our Project.....	20
3.2 Automotive History.....	21
3.2.1 How Cars Work.....	24
3.3 Technology Overview and Comparison.....	33
3.3.1 Possible Technologies.....	34
3.3.2 Comparison of Technologies.....	34
3.3.3 Connecting The Controller To The Remote Control Car.....	36
3.3.3.1 Wireless vs. Wired Communication.....	36
3.3.3.2 Wifi vs. Bluetooth.....	36
3.3.3.3 Bluetooth vs. Infrared Transmission.....	36
3.3.3.4 Selecting Technology To Connect The Controller To The Remote Control Car.....	37
3.3.4 Infrared Transmission Protocols.....	37
3.3.4.1 NEC Protocol.....	38
3.3.4.2 RC5 (Philips) Protocol.....	38
3.4 Part Overview and Comparison.....	38
3.4.1 Microcontroller.....	38
3.4.1.1 ESP8266.....	39
3.4.1.2 RP2040.....	39
3.4.1.3 ATmega328P.....	39
3.4.1.4 ESP32 Series.....	40
3.4.1.5 Microcontroller Summary.....	40
3.4.2 Infrared Emitter.....	42
3.4.2.1 CSL1501R3T1.....	42
3.4.2.2 VSMB10940.....	42

3.4.2.3 IN-S126ESGHIR.....	42
3.4.2.4 Infrared Emitter Summary.....	43
3.4.3 Infrared Receiver.....	43
3.4.3.1 TSOP4838.....	44
3.4.3.2 TSOP38238.....	44
3.4.3.3 GP1UM271RKVF.....	44
3.4.3.4 IRM2638.....	45
3.4.3.5 Infrared Receiver Summary.....	45
3.4.4 External Flash Memory.....	46
3.4.4.1 IS25WP016D/IS25LP016D.....	46
3.4.4.2 W25Q16RV/W25Q16JW/W25Q16JV.....	46
3.4.4.3 LE25U40PCMC-AH.....	46
3.4.4.4 External Flash Memory Summary.....	47
3.4.5 Motor Driver IC.....	48
3.4.5.1 DRV8835DSSR.....	48
3.4.5.2 TB6612FNG.....	48
3.4.5.3 STSPIN948.....	48
3.4.5.4 MTS2916A-LGC1.....	49
3.4.5.5 Motor Driver IC Summary.....	49
3.4.6 3.3 V Output DC/DC Converter.....	49
3.4.6.1 V78E03-1000-SMT-TR.....	50
3.4.6.2 R-78AA3.3-1.0SMD.....	50
3.4.6.3 TSR 1-2433SM.....	50
3.4.6.4 EPM78V2-03R3-01R0R.....	50
3.4.6.5 3.3 V Output DC/DC Converter Conclusion.....	50
3.4.7 5 V Output DC/DC Converter.....	51
3.4.7.1 TSR 1-2450SM.....	52
3.4.7.2 VXO7805-500-M-TR.....	52
3.4.7.3 R-78AA5.0-0.5SMD-R.....	52
3.4.7.4 5 V Output DC/DC Converter Conclusion.....	52
3.4.8 Step-Up Converter.....	53
3.4.8.1 MP3120DJ-LF-Z.....	53
3.4.8.2 MP3414DJ-LF-Z.....	53
3.4.8.3 MP3221GJ-P.....	53
3.4.8.4 Step-Up Converter Conclusion.....	53
3.4.9 Kit Comparison.....	54
3.4.9.1 Choosing a Component Kit.....	55
3.4.9.2 Metal Smart Robotic RC Tank Chassis Kit.....	55
3.4.9.3 ELEGOO Conqueror Robot Tank Kit.....	57
3.4.9.4 ELEGOO UNO R3 Smart Robot Car Kit V4.....	60
3.4.9.5 Final Decision: ELEGOO UNO R3 Smart Robot Car Kit V4.....	63
3.5 Headlights.....	63
3.5.1 INJORA RC Headlights.....	64
3.6 Handheld Controller PCB parts.....	64
3.6.1 Joystick Module.....	64
3.6.1.1 THB001P.....	64
3.6.1.2 COM-09032.....	65
3.6.1.3 TW01BLK13.....	65

Decision: COM-09032.....	65
3.6.2 Sliding Potentiometer.....	65
3.6.2.1 PSM01-081A-103B2.....	66
3.6.2.2 PTA2043-2010CIB104.....	66
3.6.2.3 PS15-10MB6BR10K.....	66
Decision: PTA2043-2010CIB104.....	66
3.6.3 Power Bank.....	67
3.6.3.1 MulticoloredLED 3 AA Battery Holder.....	67
3.6.3.2 7.4V 1500mAh Li-ion Battery.....	67
Decision: 7.4V 1500mAh Li-ion Battery.....	67
3.6.4 On and Off Switch.....	68
3.6.4.1 JS202011JCQN.....	68
3.6.4.2 JS202011SCQN.....	68
Decision: JS202011SCQN.....	68
<b>4. Standards and Design Constraints.....</b>	<b>69</b>
4.1 Standards.....	69
4.1.1 Safety Standards.....	69
4.1.2 Electrical Standards.....	71
4.1.3 Mechanical Standards.....	73
4.1.4 Software Development Standards.....	74
4.1.5 Communication Standards.....	76
4.1.6 Quality Assurance Standards.....	79
4.1.7 Documentation Standards.....	81
4.2 Design Constraints.....	82
4.2.1 Wi-Fi Standards in Rock ‘N’ Rover Development.....	84
4.2.1.1 Importance of Wi-Fi Standards.....	84
4.2.1.2 IEEE 802.11 Standards.....	84
4.2.1.3 Application of Wi-Fi Standards in Rock ‘N’ Rover.....	85
4.2.1.4 Integration and Testing.....	85
4.2.2 Additional Applications of Wi-Fi Standards in Rock ‘N’ Rover.....	86
4.2.2.1 Remote Monitoring and Control.....	86
4.2.2.2 Over-the-Air (OTA) Updates.....	86
4.2.2.3 Collaborative Operations.....	86
4.2.3 Benefits of Wi-Fi Integration in Rock ‘N’ Rover.....	86
4.2.3.1 High-Speed Data Transfer.....	86
4.2.3.2 Scalability and Interoperability.....	86
4.2.3.3 Versatile Connectivity.....	86
4.2.4 Challenges and Considerations.....	87
4.2.4.1 Signal Interference and Range Limitations.....	87
4.2.4.2 Power Consumption and Battery Life.....	87
4.2.4.3 Security and Privacy.....	87
4.2.5 Future Considerations and Innovations.....	87
4.2.5.1 Wi-Fi 6 and Beyond.....	87
4.2.5.2 Edge Computing and Fog Networking.....	87
4.2.5.3 Hybrid Connectivity Solutions.....	87
4.2.6 Bluetooth Standards in Rock ‘N’ Rover Development.....	88
4.2.6.1 Importance of Bluetooth Standards.....	88
4.2.6.2 Bluetooth Special Interest Group (SIG) Standards.....	88

4.2.6.3 Application of Bluetooth Standards in Rock 'N' Rover.....	88
4.2.6.4 Integration and Testing.....	89
4.2.7 Additional Applications of Bluetooth Standards in Rock 'N' Rover.....	89
4.2.7.1 Vehicle-to-Vehicle Communication.....	89
4.2.7.2 Fleet Management and Coordination.....	89
4.2.7.3 Sensor Fusion and Data Fusion.....	89
4.2.8 Benefits of Bluetooth Integration in Rock 'N' Rover.....	90
4.2.8.1 Wireless Flexibility.....	90
4.2.8.2 User Convenience.....	90
4.2.8.3 Low Power Consumption.....	90
4.2.9 Challenges and Considerations.....	90
4.2.9.1 Interference and Signal Degradation.....	90
4.2.9.2 Security and Privacy.....	90
4.2.9.3 Range and Coverage Limitations.....	90
4.2.10 Future Considerations and Innovations.....	91
4.2.10.1 Bluetooth 5.0 and Beyond.....	91
4.2.10.2 Edge Computing and Bluetooth Mesh.....	91
4.2.10.3 Standardization and Interoperability.....	91
4.2.11 SPI Standards in Rock 'N' Rover Development.....	91
4.2.11.1 Importance of SPI Standards.....	91
4.2.11.2 SPI Specifications.....	91
4.2.11.3 Application of SPI Standards in Rock 'N' Rover.....	92
4.2.11.4 Integration and Testing.....	92
4.2.12 Additional Applications and Benefits of SPI Integration.....	93
4.2.12.1 Real-Time Control and Feedback.....	93
4.2.12.2 Reduced Wiring Complexity.....	93
4.2.12.3 Multimodal Sensor Fusion.....	93
4.2.12.4 Energy-Efficient Operation.....	93
4.2.13 Challenges and Considerations.....	93
4.2.13.1 Signal Integrity and Noise Immunity.....	93
4.2.13.2 Timing Synchronization and Latency.....	93
4.2.13.3 Scalability and Expansion.....	94
4.2.14 Future Considerations and Innovations.....	94
4.2.14.1 High-Speed SPI Variants.....	94
4.2.14.2 Fault Tolerance and Redundancy.....	94
4.2.14.3 Standardization and Interoperability.....	94
<b>5. Project Design Details.....</b>	<b>95</b>
5.1 Hardware Diagrams.....	95
5.1.1 Main Block Diagram.....	95
5.1.2 Remote Control Car Block Diagram.....	96
5.1.3 Controller Block Diagram.....	96
5.2 IR Emitter Circuit.....	97
5.3 Rock N' Rover App Development.....	99
5.3.1 Introduction to MIT App Inventor.....	99
5.3.2 MIT App Inventor Concepts.....	100
5.3.3 Innovative App Features.....	101
5.3.4 App Implementation.....	102
5.3.5 App Description.....	103

5.3.6 App Design.....	104
5.3.7 Changes in Approach: Final App Design.....	107
<b>6. Project Construction.....</b>	<b>108</b>
6.1 PCB Schematics.....	108
6.1.1 Remote Controller Schematic 1st Version.....	108
6.1.2 Remote Controller Schematic 2nd Version.....	109
6.1.3 RC Car Schematic.....	109
<b>7. Project Testing.....</b>	<b>111</b>
7.1 Kit components Testing.....	111
7.2 Headlights Testing.....	111
7.3 Hand Held PCB Testing.....	111
7.4 Infrared Transmitter/Receiver Testing.....	111
7.5 Senior Design 2 Testing.....	112
7.5.1 Car PCB Motor Driver.....	112
7.5.2 Car PCB IR Receiver.....	113
7.5.3 Remote Controller PCB IR Transmitter.....	113
7.5.4 ESP32 CAM.....	113
<b>8. Administrative Content.....</b>	<b>114</b>
8.1 Project Milestones/Schedule.....	114
8.2 Initial Budget Breakdown.....	118
8.3 Updated Budget Breakdown.....	120
Justification of Costs:.....	120
Budget Management:.....	121
Risk Identification:.....	121
Risk Assessment:.....	122
Contingency Planning:.....	122
Risk Monitoring and Review:.....	122
8.5 Work Distribution and Team Members.....	122
<b>9. ChatGPT and other AI/LLMs.....</b>	<b>124</b>
<b>10. Conclusion.....</b>	<b>126</b>
<b>11. Appendices.....</b>	<b>128</b>
11.1 Appendix A - References.....	128

# **1. Introduction**

In the world of remote-controlled cars, exciting progress is constantly being made to make them better and more fun to use. The Rock 'N' Rover is a prime example of this progress, representing a major step forward in remote-controlled car technology. This impressive vehicle is designed with a range of cool features to make it safe and enjoyable to use. For instance, it's equipped with special sensors that help it detect obstacles so it can navigate safely. Plus, it has a speaker and headlights to make interacting with it more fun and to help you see better when it's dark. With a built-in camera and Bluetooth connection, you can even see what it sees and control it from your phone. And thanks to its easy-to-use controller and app, switching between different modes of control is a breeze. Powered by strong motors and precise steering, the Rock 'N' Rover moves smoothly and can handle many different tasks. With its simple yet advanced features, the Rock 'N' Rover is leading the way in remote-controlled car technology, offering something exciting for everyone.

## **1.1 Executive Summary**

The Rock 'N' Rover is an innovative remote-controlled car designed to offer advanced features for enhanced functionality and enjoyment. Developed by a team of electrical and computer engineering students, this project integrates cutting-edge technologies to provide a versatile and interactive user experience. The vehicle is equipped with obstacle detection through strategically placed ultrasonic sensors, ensuring safe navigation, and features a camera for interactive experiences and improved visibility with front headlights for low-light conditions. With Bluetooth connectivity, users can enjoy a visual perspective through the companion app, allowing for seamless communication with external devices. The intuitive controller and app interface facilitate effortless switching between manual and remote control modes, ensuring ease of use. Powered by reliable DC motors, the Rock 'N' Rover offers responsive movement suitable for various applications.

The project utilized components from the Elegoo Uno R3 Smart robot car, incorporating them into a custom PCB with an ESP32 microcontroller. The control system included an IR transmitter on a controller PCB and an IR receiver on the car's PCB, managed by the DRV8835DSSR motor driver to control four DC motors. Despite challenges such as signal interference and power management issues, the team implemented shielding techniques and optimized the power supply system to ensure reliable operation. The Rock 'N' Rover successfully demonstrated obstacle detection, interactive camera experiences, and smooth movement control, highlighting its potential for educational and research applications in robotics and embedded systems. Future work could explore incorporating advanced sensors and machine learning algorithms for autonomous navigation, further enhancing the system's capabilities.

## **2. Project Description**

The Rock 'N' Rover, a cutting-edge remote-controlled car is designed to provide an array of features for enhanced functionality and enjoyment. This vehicle boasts advanced capabilities, including obstacle detection through strategically placed Ultrasonic Sensors, ensuring safe navigation. Additionally, it features camera for interactive experiences and front headlights for improved visibility in low-light conditions. With a built-in camera and Bluetooth connectivity, users can enjoy a visual perspective through the companion app and seamless communication with external devices. The intuitive Controller and App interface allows for effortless switching between manual and remote control modes, ensuring ease of use. Powered by reliable DC Motors, the Rock 'N' Rover offers responsive movement suitable for a variety of applications. With its user-friendly design and versatile features, the Rock 'N' Rover represents the forefront of remote-controlled car technology.

### **2.1 Motivation and Background**

The beauty of engineering is how something so small can change the world. A few electrons move in a direction, and now you have current flowing through a component. You combine these components together, and you create life-saving technology. We took inspiration from the biomedical field, and started brainstorming all types of ideas for a prosthetic arm. The only problem was the scope of the project. We have four senior engineering students, and we can do better than slapping together some servos and sensors. We kept thinking about all of the applications for robotic arms and hands, and that led us to a bomb defusal robot. These robots are able to get explosives away from people, and dispose of them in a safe environment. This takes our small idea, and adds way more challenges to it, so we focused on the idea of a remote controlled car.

Through our years of education, we have learned the skills to complete a project such as this. The thing we still need to learn is how to work as a team, all striving towards a common goal. We need to design and create a fully functioning robot that includes dozens of sensors, hundreds of parts, and thousands of lines of code, and we need to do it in a timely and cost effective manner.

Originally, our group was going to create a smart mirror that would display the weather, time, and would have other features such as an alarm clock and being able to play music through Spotify. After careful consideration, our group ultimately decided to create a remote control car with a robotic arm for our project. This idea arose because one of our team members, Michael Patalano, has an interest in robotics. He pitched the idea of creating a prosthetic arm and then we discussed making a robotic arm with a hand and fingers. We decided not to go with the prosthetic arm or the robotic arm idea for the project due to it being too complex because it would require a lot of mechanical parts and the 3-D printing needed for the project would require skills that none of us have. While discussing our project idea, someone had also mentioned a bomb defusal robot so naturally we decided to make a remote control car instead. Bomb defusal robots can either defuse the bomb or grab it and move it far away, and ours was able to grab objects



and move them away. This has many more applications, from organizing in factories to building a way stronger robot and moving much heavier objects.

## **2.2 Goals and Objectives**

The main goal was to build a remote-controlled car with a variety of features. We also designed a companion app. The most important feature was to be able to see what the car saw by using cameras on the body of the car and displaying the feed on the app. Some of the other mandatory goals were to control the arm and drive the car simultaneously, place ultrasonic sensors around the car's body to alert the driver of any objects they couldn't see from the front and rear cameras, and include a speaker that could be controlled by the app. The handheld controller, instead of being on the app, allowed the driver much more accurate control, which was very important in high-pressure situations. The ultrasonic sensors gave the driver feedback about terrain and any incoming objects. The speakers allowed us to send a message to someone without alerting others to the driver's location. An underrated goal that we needed to reach was battery life. We wanted the car to be able to drive for at least an hour. This required more powerful motors, servos, and a larger PCB to properly disperse the current to all of the components.

Some of the advanced goals we aimed to reach included being able to speak directly from the controller to the camera, having headlights with different settings, and displaying the battery life on the app. The headlights or a night vision camera were very practical additions. It got dark pretty early in Florida, so having a way to see where we were driving at night was important. We chose the headlights first because connecting them to the microcontroller was simpler, but the night vision light required programming and additional cost. A regular camera was significantly cheaper than a night vision one. The last feature involved showing the battery life on the app, but it encompassed all of the car's features. We only needed the app to show the driver where the car was going; the rest was extra. Having a screen that controlled camera angles, manual subsystem turn-offs for battery consumption, headlight brightness control, and even a driving setting that retracted the arm for optimal wind resistance in long-distance driving was an amazing feature. The fewer controls needed on the handheld controller, the better, as this allowed for more intuitive driving.

We had a few long-term stretch goals and aimed to hit some of these, but since we were in the Spring and Summer semesters, it was much harder to achieve all of them. The first one was to use the ultrasonic sensors like a radar detector and make that available in the app for the driver. This was most likely unnecessary because we were driving on streets or flat areas, so there wouldn't be terrain to stop the car from driving.

## **2.3 Features/Functionalities**

The Rock 'N' Rover was equipped with a range of features to ensure efficient and versatile functionality. An Ultrasonic Sensor facilitated obstacle detection for safe navigation. Front headlights provided visibility in low-light conditions. The camera offered a visual perspective for enhanced navigation, and Bluetooth connectivity enabled

seamless communication with external devices. The inclusion of a Controller and App interface ensured user-friendly manual and remote control. DC Motors for both the front and rear wheels. Together, these features created a sophisticated yet user-friendly robotic car suitable for diverse applications.

Features	
Ultrasonic Sensor	Controller
Bluetooth	Mobile Application
Headlights/LED lights	DC Motors For Rear Wheels
Camera	Steering Servo For Front Wheels

The design features of the Rock ‘N’ Rover robotic car align with various robotics and IoT projects that integrate sensors, actuators, and multimedia elements for enhanced functionality. Projects such as autonomous robots and IoT-controlled vehicles often employ ultrasonic sensors, cameras, Bluetooth connectivity, and motor systems for navigation, interaction, and entertainment purposes. Examples of platforms like Arduino or Raspberry Pi-based robotic projects may serve as general references for integrating similar features.

### **2.3.1 Customer Input and Market Analysis**

The project features were identified through customer input gathered from surveys, ensuring alignment with user expectations. Additionally, a detailed marketing analysis of comparable products was conducted to identify successful features and innovative strategies. This input formed the foundation for the following key features:

### **2.3.2 Advanced Mobility**

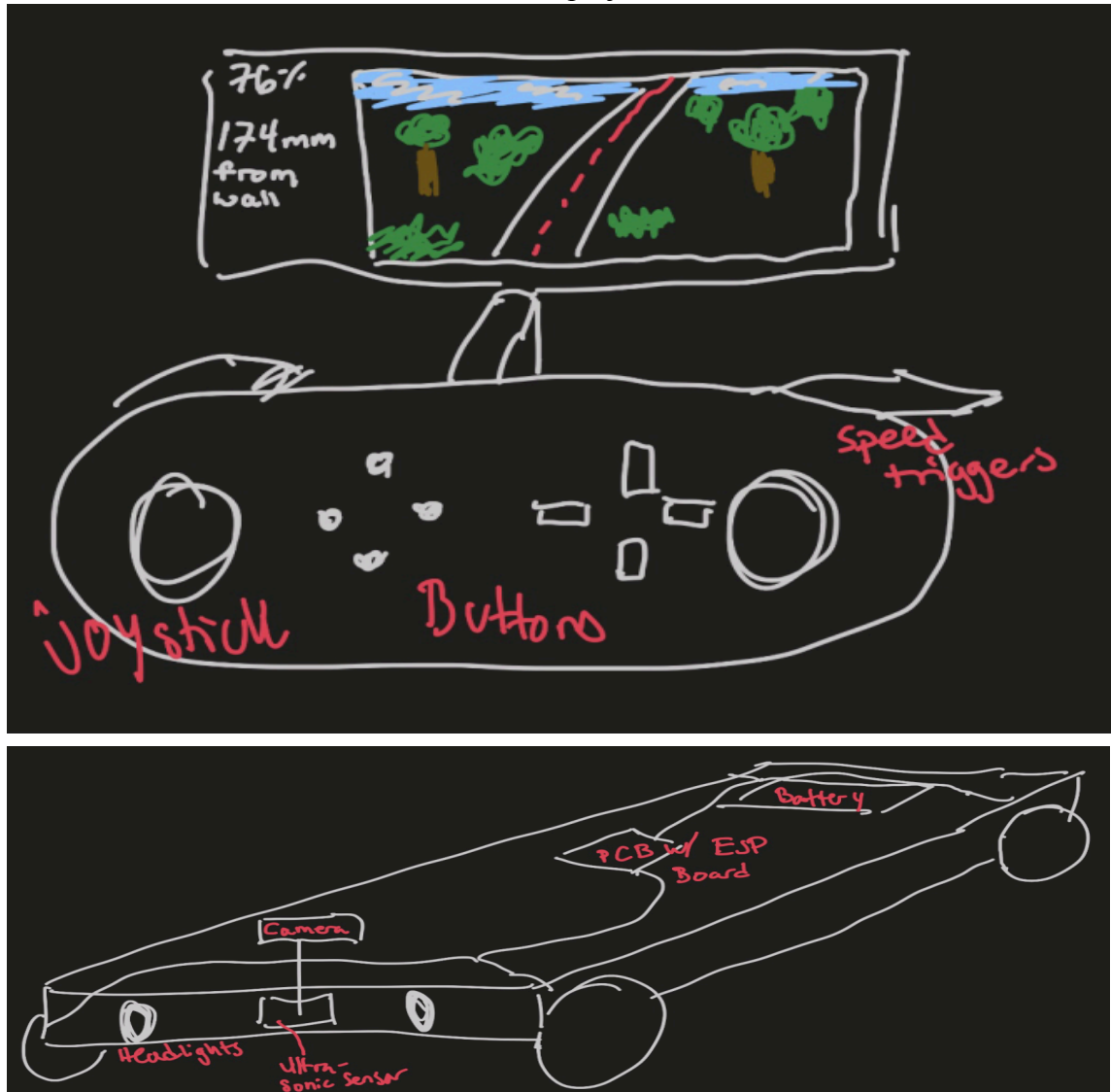
Drawing inspiration from various smart robot cars, our robotic machine incorporated high-performance motors, precision control algorithms, and obstacle avoidance sensors. This ensured smooth navigation and collision-free operation, addressing user expectations for a versatile and mobile platform.

### **2.3.3 User-Friendly Interface**

Building on the success of user-friendly interfaces seen in existing products, our project integrated intuitive controls for both remote and app-based operation. Customizable settings, inspired by customer preferences, provide an enhanced user experience. In addition, the innovative features of the 360-degree camera, headlights, and music speaker enhanced the functionality and user appeal of our robotic machine.

## 2.4 Prototype Illustration

These are sketches of our initial ideas for the project.



The drawn prototype for the Rock 'N' Rover served as a visual representation of the envisioned design. The prototype showcased a well-defined body structure equipped with four wheels to facilitate mobility. The inclusion of headlights enhanced visibility in the front, contributing to the overall practicality of the design. Additionally, a camera was strategically incorporated for navigation purposes, allowing users to view the surroundings through a visual feed. The hardware components essential for the car's operation were thoughtfully secured within the central region of the car's body, ensuring a compact and organized layout. This prototype not only captured the physical attributes of

the Rock ‘N’ Rover but also highlighted its potential capabilities, combining both mobility and utility in a single, innovative design.

## **2.5 Engineering Specifications**

<b>Key Specifications</b>
Body of the car must be at least 3 inches in width and 5 inches in length
Battery must be able to power car for at least 1 hour
Car must be able to move forwards and backwards and turn left and right
Cost of the car must be at a maximum of \$800 (\$200 per team member)
Headlights must be visible from at least 3 feet away
Must be able to drive a distance of at least 10 feet

## 2.6 House of Quality

This is our House of Quality. It aims to provide a concise and understandable description of what we are trying to achieve in this project.

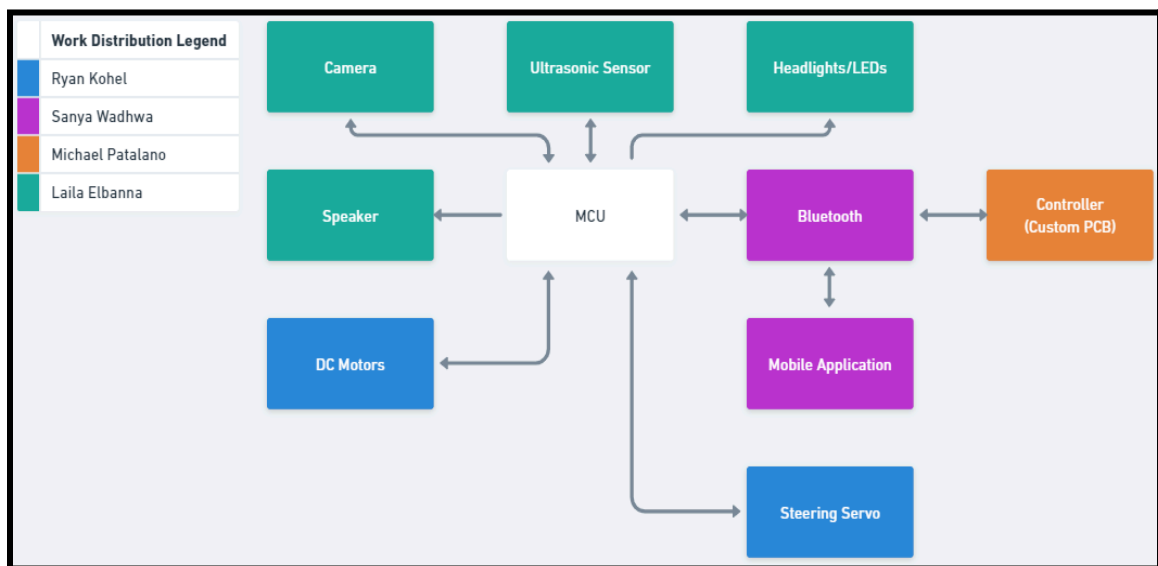
Column #	1	2	3	4	5	6	7	8	9	10
Direction of Improvement	◇	▼	▲	▲	▲	▲	▲	▲	▲	◇
Engineering Requirements	Size	Weight	Battery Life	Power Transmission Efficiency	Power Utilization Efficiency	Accuracy	Travel Range Capability	Understandability	Communication Range	Light Range
Customer Requirements (Explicit and Implicit)										
Power	●	●	●	●	●	○	○	▽	○	▽
Reliability	▽	▽	●	○	○	●	○	●	●	▽
Ease Of Use	▽	○	○	▽	▽	▽	○	●	●	▽
Compatibility	▽	▽	○	●	●	●	○	●	○	▽
Connectivity	▽	▽	○	●	●	○	○	●	●	▽
Time-Saving	▽	▽	●	●	●	●	●	●	●	▽
Ease Of Maintenance	○	○	○	▽	▽	▽	▽	▽	▽	▽
Upgradability	●	●	●	○	○	▽	○	▽	▽	●
Affordability	●	●	●	●	●	○	○	▽	○	●
Portability	●	●	▽	▽	▽	▽	▽	▽	▽	▽
Product Efficiency	○	○	●	●	●	●	●	●	●	○
Target	3" by 5"	Ability to Lift with Ease	Long Battery Life	1-2% Energy Lost during Transmission	Ideal PUE is 1.0	Can grab objects with ease without falling	Travel s large distances in a room	Respon ds to the control ler rapidly	Must respon d to comm ands from a differe nt room	A few (3-4) feet of visibilit y
Column #	1	2	3	4	5	6	7	8	9	10

## 2.7 Initial Project Diagrams

For our project, hardware diagrams and software diagrams were created initially to provide a very general, brief overview of how our project would work. Also, the initial diagrams would show the work distribution between the different members of our group. The initial hardware and software diagrams are provided in the sections below.

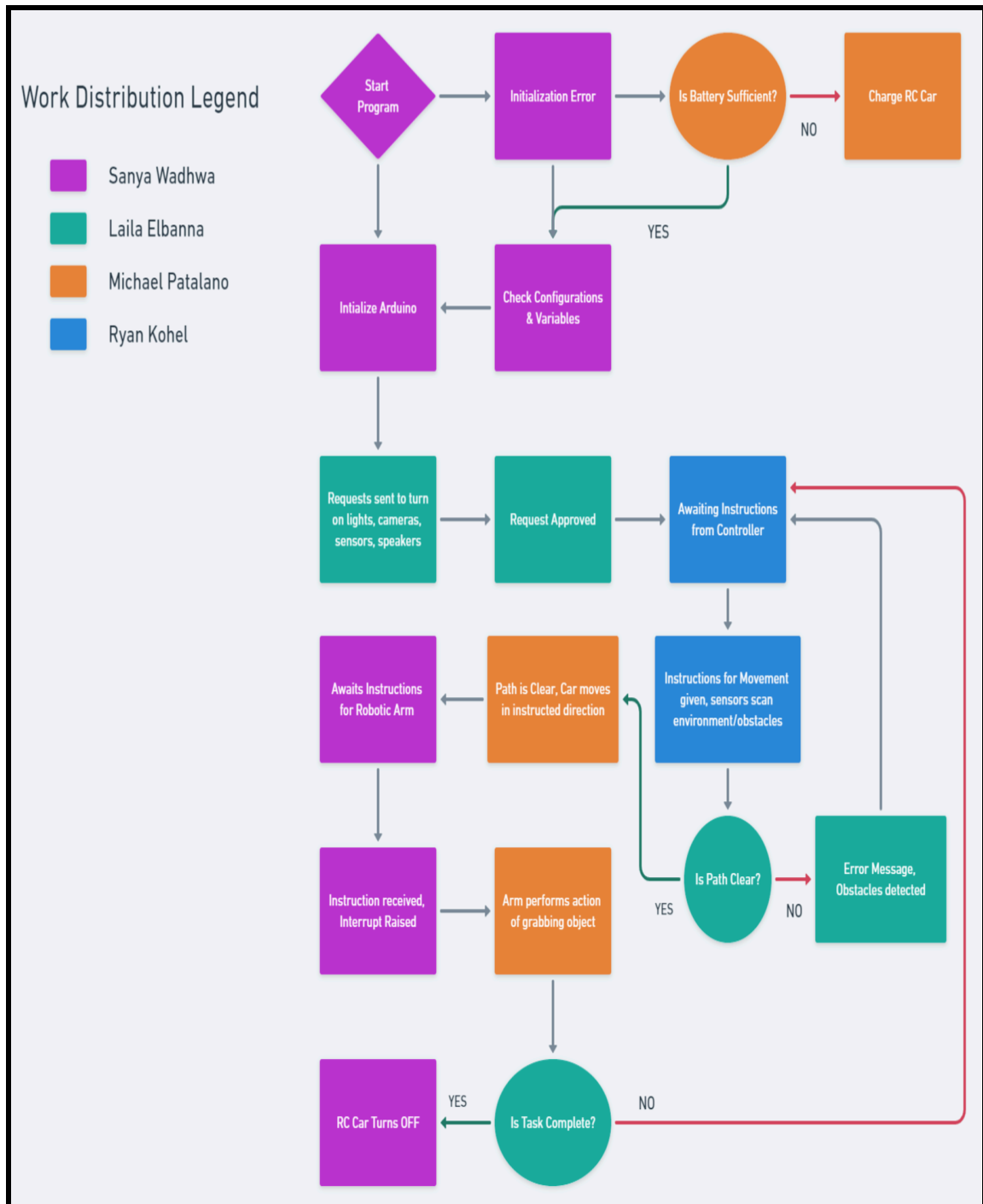
### **2.7.1 Initial Hardware Diagrams**

This initial hardware diagram below provides a general overview of the different components of the project and how they are connected to one another. The microcontroller is connected physically to 6 different components: a camera, an ultrasonic sensor, headlights, a speaker, DC motors, and a steering servo. The camera was used to have a POV in front of the remote control car. The ultrasonic sensor was used to detect if the remote control car comes close to an object. The speaker was used to produce a beep if the remote control car comes too close to an object. The DC motor will move the remote control car forwards and backwards and the steering servo was used to turn the remote control car left and right. Initially, we were going to connect the mobile application and the controller via Bluetooth, however, we decided to connect the application via Wifi and connect the controller via an infrared transmitter and receiver. So, we will have an application that was connected to the microcontroller via Wifi, and the application can be used to drive the remote control car and view the video from the camera. We will also be creating a custom PCB for our controller to control the remote control car. The controller will include an infrared transmitter that will send information to an infrared receiver on the car to control the car remotely.



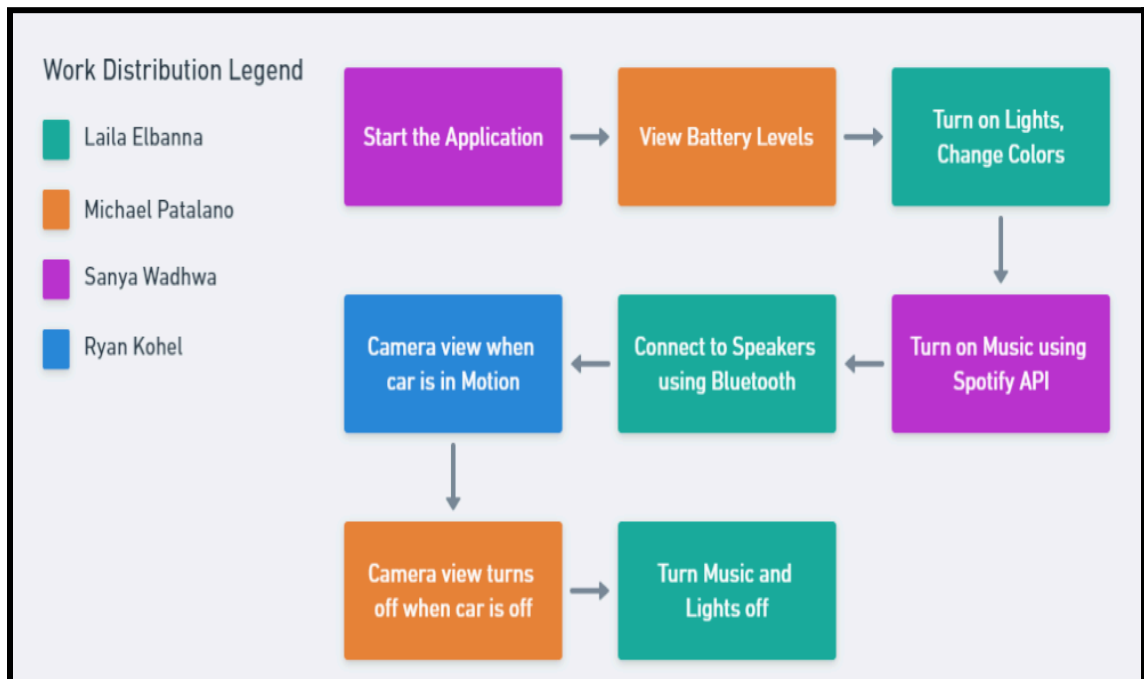
### **2.7.2 Initial Software Diagrams**

The software diagram below illustrated the communication between the Arduino and the controller, detailing the program's execution and functionality. It encompassed the initialization of the Arduino, reading sensor data for environmental awareness, configuring motors, and receiving user inputs from the controller. The program then assesses sensor data for potential obstacles or targets, determines desired rover movements, and executes corresponding commands. Following this, it halts the motors and the arm, culminating in the termination of the program.

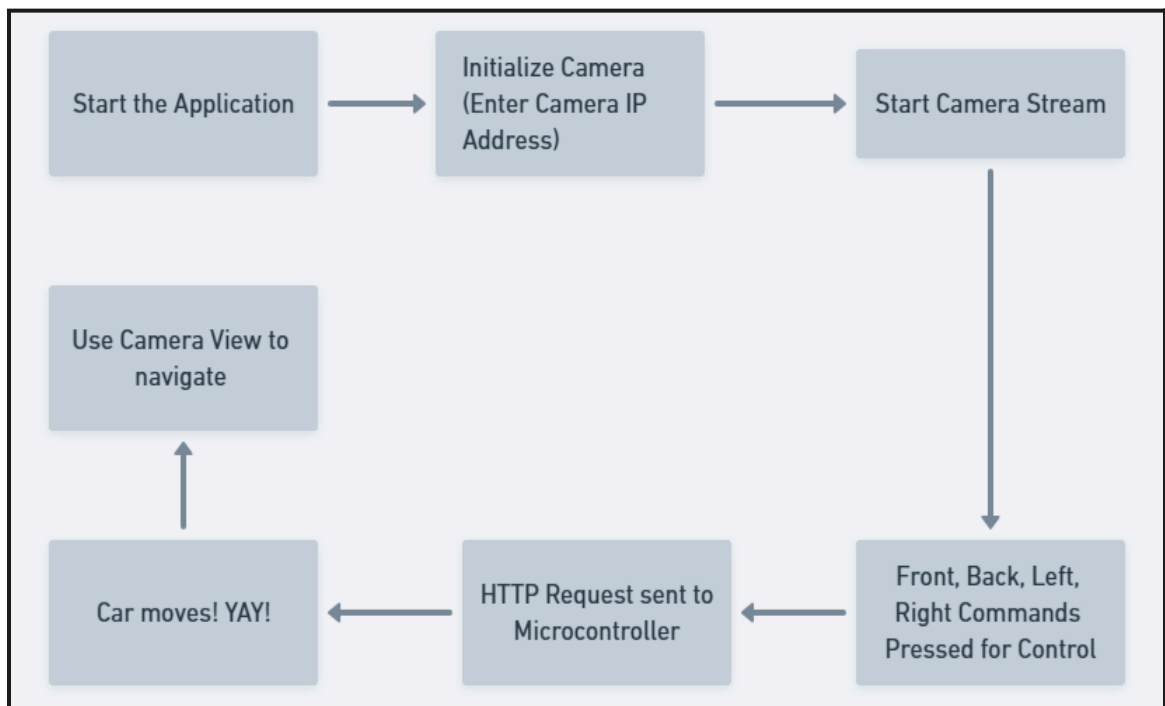


The application software diagram below provides a comprehensive overview for various functions of the RC car. It intricately outlines the process of turning lights on and off, allowing for dynamic color changes, and even incorporating music playback through Spotify, enhancing the user experience for entertainment purposes. Additionally, the diagram showcases the functionality of navigation, utilizing the camera view to provide a visual perspective during operation. Beyond that, it introduces a monitoring system for tracking the battery levels of the car, ensuring users are informed about the power status

for optimal control and efficient operation of the robotic car. In essence, this application software diagram serves as a visual guide, offering a clear depiction of the diverse and multifunctional capabilities.



### 2.7.3 Revised Software Diagram





### **3. Project Research**

In this section, we will discuss the various different parts of research involved in creating our project. We will discuss similar projects that inspired us, different technologies that was used in our project, and we will discuss how different parts and technologies was selected and used in our project.

#### **3.1 Similar Past Projects**

As we enter the journey of our senior design project, it's important to acknowledge the history and evolution of robotics, particularly within the realm of educational robotics. The field of robotics has seen remarkable advancements over the decades, driven by a combination of technological innovation, scientific discovery, and educational initiatives.

The concept of remote-controlled (RC) cars, in particular, holds a special place in the records of robotics history. RC cars have served as both recreational toys and educational tools, captivating enthusiasts of all ages with their combination of technology, engineering, and entertainment.

The origins of RC cars can be traced back to the early 20th century, with the invention of the first remote control systems. Early prototypes were basic and limited in functionality, relying on simple radio frequency transmitters and receivers to transmit commands to the vehicle. Despite their limitations, these early RC cars captured the imaginations of enthusiasts, laying the groundwork for future advancements.

Throughout the mid-20th century, RC cars underwent significant advancements in design, technology, and performance. The introduction of more sophisticated control systems, such as proportional radio control and digital signal processing, revolutionized the capabilities of RC vehicles. This era saw the emergence of specialized RC car kits and racing events, creating a lively community of enthusiasts and innovators.

In recent decades, the convergence of robotics and RC technology has given rise to a new generation of educational robotics platforms. These platforms, represented by products like the Hiwonder Smart Robot Car and the DEERC RC Car, offer a unique blend of entertainment, education, and hands-on learning.

The Hiwonder Smart Robot Car, for example, represents an ideal shift in educational robotics, providing users with a versatile platform for learning electronics and programming. Its modular design, customizable features, and user-friendly interface make it an ideal tool for educators and students both.

Similarly, the DEERC RC Car embodies the spirit of innovation and exploration, pushing the boundaries of remote control technology with its advanced features and immersive experience. By combining performance, entertainment, and educational value, the DEERC RC Car inspires curiosity and creativity in users of all ages.

As we indulge into our senior design project, we get inspiration from the history and legacy of RC cars, recognizing their role as pioneers in the field of robotics education. By immersing ourselves in the study of similar past projects like the Hiwonder Smart Robot Car and the DEERC RC Car, we position ourselves to obtain valuable insights, strategies, and innovations that will shape the trajectory of our own project.

### **3.1.1 Hiwonder Smart Robot Car**

While researching past projects, we were heavily inspired by the remarkable steps made in educational robotics, particularly by revolutionary products such as the Hiwonder Smart Robot Car. This project marks a significant moment where we aim to draw from the layers of robotics history, focusing particularly on the transformative impact of the Hiwonder Smart Robot Car.

The Hiwonder Smart Robot Car stands as a representation of innovation, reshaping the world of educational robotics with its versatile platform and accessible learning experience. Its influence extends far beyond just functionality. It represents exploration and discovery that resonates deeply with our project objectives.

At its core, the Hiwonder Smart Robot Car embodies a commitment to making robotics education accessible to all. Its user-friendly assembly process transcends basic construction, creating an environment where exploration and experimentation thrive. By analyzing its assembly instructions, component layout, and interface design, we gain valuable insights into simplifying complexity without sacrificing depth. Moreover, the car's programmable nature serves as a gateway to boundless educational opportunities. Through its codebase and software interface, we explore the interplay between theory and practice, where concepts turn into tangible outcomes.

One of the most striking features of the Hiwonder Smart Robot Car is its modular design, which allows for endless customization and expansion. This modularity not only enhances the educational experience by encouraging experimentation but also ensures adaptability to a wide range of learning objectives and skill levels. Whether students are exploring basic concepts in robotics or delving into advanced programming techniques, the Hiwonder Smart Robot Car offers a platform that can evolve with their needs.

Additionally, the Hiwonder Smart Robot Car is equipped with a comprehensive set of sensors and peripherals, further enriching the learning experience. From ultrasonic sensors for obstacle detection to motor encoders for precise control, these components provide students with hands-on experience in working with real-world sensors and actuators. Furthermore, the integration of wireless connectivity allows for seamless communication with external devices, opening up possibilities for collaborative projects and remote control applications.

As we get into our senior design project, we draw inspiration from the foundational principles embodied by the Hiwonder Smart Robot Car. Our goal is not merely to replicate its features but to embrace its versatility, accessibility, and hands-on learning. By infusing our project with these guiding principles and leveraging the innovative

design of the Hiwonder Smart Robot Car, we aim to create an educational experience that empowers and inspires, fostering creativity and innovation in students of all backgrounds.

### **3.1.2 DEERC RC Car**

The DEERC RC Car stands as a fascinating embodiment of the relationship between entertainment, performance, and education, pushing the boundaries of remote control technology with its immersive experience. Beyond its exterior, which instantly captivates the eye, lies a testament to engineering prowess and technological innovation, positioning it as a frontrunner in the realm of educational robotics.

Some of the captivating aspects of the DEERC RC Car is its seamless integration of advanced control mechanisms and robust construction. Delving into its mechanical design, is carefully engineered components working in harmony to deliver optimal performance and user engagement. Each aspect, from the precision-engineered structure to the complexly designed electronic systems, reflects a dedication to quality and excellence that elevates the DEERC RC Car beyond being a basic toy. It is a channel for exploration, where every movement serves to spark curiosity and stimulate the imagination.

However, what truly sets the DEERC RC Car apart is its multifaceted nature, which smoothly blends entertainment with education. Beyond its role as a source of amusement, the car serves as a powerful educational tool, offering a rich array of learning opportunities for users of all ages. Through interactive features, such as programmable control interfaces and immersive simulation environments, users are encouraged to explore fundamental concepts in robotics, programming, and engineering in an engaging and hands-on manner. Moreover, the DEERC RC Car creates a sense of community through its online platforms and community engagement initiatives, providing a space for users to share knowledge, collaborate on projects, and inspire one another.

Our integration of the principles embodied by the DEERC RC Car extends far beyond surface-level replication. It represents a deep rooted commitment to excellence and innovation. By infusing our project with the aspects of performance, entertainment, and educational enrichment represented by the DEERC RC Car, we aspire to craft an experience that transcends the ordinary.

Our vision is to create an educational robotics project that not only captivates the minds of users but also inspires curiosity, fosters creativity, and cultivates a lifelong passion for learning. Through careful planning, thoughtful design, and iterative development, we aim to build upon the legacy of the DEERC RC Car, creating an experience that leaves a lasting impact on the field of educational robotics and the individuals who engage with it.

### **3.1.3 Arduino Esplora**

One of the biggest inspirations to have a handheld PCB controller was the Arduino Explorer. We had used it in the past, so once we decided we wanted to have a RC car, this was something we wanted to implement. It is a microcontroller board derived from the Arduino Leonardo. The Explorer is different because most Arduino boards come with all

the GPIO pins with female header pins, so you as a consumer can connect to those pins using all sorts of components and create your own system. This design is an oval shape and it is meant to be an all in one board testing kit, that includes all of the common components to use if you were due if you were to begin Arduino programming. The Arduino Explorer includes an ATmega-32 E4 microcontroller, an analog joystick, four push buttons, a light sensor, a temperature sensor, a RGB LED, and a linear potentiometer. The linear potentiometer is a sliding one. It also includes an accelerometer, which will allow you to measure acceleration along the XYZ-axis in terms of detecting changes in orientation and movement. It even has a built-in speaker and a microphone. It has pins allowing you to connect it to other devices and send information that way. There are 4 3-pin headers, each with a 5V, GND, and digital pin. In the middle of those four pins is a USB Micro Type B port. This is for writing code on the Arduino IDE, and uploading it onto the Esplora. Arduinos are meant to be all in one platform where you can connect multiple components together in the form of inputs and outputs, where this board allows all of the common input outputs that beginners would use to be in one place. The code is slightly different than it would be for the Arduino UNO, and a couple of tweaks are required. This is a massive inspiration for our handheld controller PCB, as we will include the sliding potentiometer, the joysticks, one of the buttons, and a way to upload code from the Arduino IDE to our microcontroller.

### **3.1.4 PlayStation 4 Controller**

Another project that is similar to our handheld PCB is a PlayStation 4 controller. There is one important distinction between this controller and our controller. This one is meant to be applicable to an infinite amount of video games, so it has a ton of additional buttons and joysticks and triggers that you can use, and that designers can use to control how their game works. Our design is much more simple. Only a few buttons are needed, We do not have the handlebar design that the PlayStation 4 controller has. It has two joysticks where your thumbs go, a touchpad on the left side, and four buttons on the right side. It has two buttons on the top of the controller, along with a start button in the middle of the controller on the top of the controller it has, two push buttons, a light and USB port for charging and two trigger components it also has motion sensors in forms of gyroscope and accelerometer so this all allows for motion controlled gameplay and enhanced interactive experiences.

### **3.1.5 Integration into Our Project**

The insights gathered from an in-depth examination of the Hiwonder Smart Robot Car and the DEERC RC Car served as the foundation of our project's development journey, laying the groundwork for a path guided by innovation. Through a process of integration and adaptation, we aimed to create a combination of features and functionalities that resonated deeply with the aspirations and expectations of our target audience, setting a model for transformative advancements in the realm of educational robotics.

With the Arduino Esplora and the PlayStation 4 controller in mind, those two projects led us to design an oval-shaped handheld PCB controller that looked similar to the Arduino Esplora but included many similar components and the layout of a PlayStation controller.

It was a beautiful combination because the Arduino Esplora is meant to lay on the table, and you can interact with it as if it were a bunch of components connected on a breadboard. While that is a great quality, it is not meant to be held for long periods. This is where the PlayStation 4 controller came into play, as it is meant for exactly that. We did not copy its handlebar shape, which would have been better for the driver but would have cost much more in manufacturing. Instead, we adopted their layout of components for maximum ease of use. The main idea was to have the joysticks right next to the thumbs, so the driver did not have to stretch and extend their hands a lot.

Our approach to integration was more than simply replicating; it was a journey characterized by exploration, creativity, and a commitment to user-centric design principles. By infusing our project with curiosity-driven exploration and innovative problem-solving, we strove not only to meet but to exceed the expectations of our users, pushing educational robotics into a new era of immersive and engaging learning experiences.

Every feature and functionality identified through our research process underwent refinement to ensure alignment with the preferences and aspirations of our target demographic. Analyzing a diverse array of inputs, including comprehensive customer feedback, extensive market analysis, and deep insights gathered from the study of existing products, we carefully crafted a roadmap toward innovation and excellence in educational robotics.

Embracing a divide-and-conquer approach, our project was structured to facilitate a comprehensive and cohesive development process. Each aspect of the design was carefully crafted to deliver maximum impact, generating a smooth integration of components and functionalities that enhanced the overall user experience. By aligning our vision with the aspirations and needs of our users, we embarked on a transformative journey powered by a relentless pursuit of excellence, where curiosity was nurtured, and creativity flourished.

Through dedication, perseverance, and a persistent pursuit of innovation, we aspired to redefine the world of educational robotics, leaving an undeniable mark on the field and inspiring future generations of innovators and educators. In our journey to push the boundaries of what is possible, we remained committed to creating educational experiences that not only empowered learners but also fostered a deep and enduring love for exploration, discovery, and lifelong learning.

### **3.2 Automotive History**

The first step in building an RC car was to learn what it is we were trying to replicate. The automobile was invented by Carl Benz in 1879, in Mühlburg, Germany . It was not fully completed until 1885. It had a single-cylinder two-stroke engine, and Benz was able to focus on improving it to eventually having a lightweight car with a gasoline engine. The 1885 version included a single-cylinder four-stroke engine, a steel tube frame, and three wire-spoked wheels. If you are just as automotively inclined as us, you are probably wondering what that means. There have been hundreds of different engines designed over

the years, but this is where it all started. A four-stroke engine refers to the steps it takes to work. It uses a piston, which is the moving component in engines, pumps, and compressors. Its job is to transfer the force of the gas being pushed from the cylinder into the intake. Once this is achieved, our four-stroke steps begin. The steps are as followed: Intake, Compression, Combustion, and Exhaust. Intake is when gas is being transferred into the engine. This is caused by the piston. The intake is open and when the air gets forced into, it creates a negative space in the engine, which causes a build up of pressure. Compression is when the fuel mixture is being added to the gas that is sitting there. Combustion is where the power is created, as the crankshaft goes around in 360 degrees, and causes even more of a buildup in the engine of pressure, which leads to the final step. Exhaust is when the pressure is so forceful, that the valve holding the air-fuel mixture is opened, releasing the pressure and making the engine work. This process runs indefinitely. The single-cylinder is where that whole process is held in. Being only one cylinder, it is more compact, and it is used for smaller machines. Some modern examples are motorcycles, scooters, and go-karts. The cylinder allows for much more airflow than multi-cylinder designs, so you do not have to worry about cooling. The air takes care of all the heat from the mini-explosions caused by the engine. Because all of this is only happening in one tube basically, it leads to a ton of vibrations. There is only one surrounding it, so that is what takes the outer force. Using this engine, Benz was able to output 0.75 horsepower, or 0.55 kiloWatts. He filed his patent for a “vehicle powered by a gas engine” on January 29th, 1886.

The history of automobiles does not just start in Germany, as the whole invention of an internal combustion engine was in France, by Jean Joseph Étienne Lenoir in 1859. Benz has the claim for the first official automobile, but in 1901, Wilhelm Maybach created the first motorcar, which shares real similarities in the cars we drive today. It had a 35 horsepower engine, and topped out at just over 50 miles per hour. Even though Germany and France were the first to this industry, the United States would quickly take the mantle as the premier country for the automotive industry. Innovation was coming soon, but the first Mercedes was made in 1901, and had all of those features. The automobile offered in America was made by Old Motor Works, and it was a one-cylinder, 3 horsepower, horse buggy. This might still seem insane, but the crazy part is not even the vast difference in the horsepower, it is the fact that Mercedes was able to do this for 5 more years. The Old Motor Works company sold those cars for 5 years in 1901-1906, and they sold more than any automobile before, all while being so behind in technology. This was largely due to the price point being so more affordable to middle class Americans. Decent, but affordable productions will always sell much more than high quality, expensive products.

The United States being far behind would quickly end, as Henry Ford came out with the Model T in 1908. America and Europe are extremely different areas. Europe had been settled for hundreds of years at that point, and the United States was still this massive country of land. If you ever wanted to explore and expand, you needed an automobile to get around. The American economy was also doing better than many in Europe, so people had more means to purchase these cars. Henry Ford not only revolutionized the automotive industry, but he changed the entire world, when he introduced mass production. This is where the assembly line was created, and it was discovered that it was

much more efficient to have a worker specialized in a specific part of the process, hand it off to the next person in line, and then restart the process. By specializing, it made someone extremely talented at their job, and allowed for production to increase exponentially. This industrial revolution led to the United States making just over 80% of all cars in 1913. Other American companies started to use these mass production techniques as well, so the massive volume of sales they were having led to the elimination of all small American producers. Only the big guys could stay in business.

The industry kept getting larger and larger, because everyone wanted an automobile. This eventually stopped when everyone already had one, so the companies had to turn their focus on making better cars that people could trade their old ones in for. The Model T was the first mainstream car in America, and it did its job. The whole goal was to make something that works and gets the job done. Since there were not many new buyers, and there were only so many advancements to make at the time, they started to focus on aesthetics. Companies thought that if you did not need a new car, because your car already works, then the only way to increase sales is to make you want a new one. By focusing on design and details, the industry was able to make another turn up. Companies were making more money, but the product itself took a dip, because they were more focused on styling and cost-cutting measures, than making the best automobile possible.

World War II had a massive impact on the automotive industry. They were responsible for producing vehicles and other various materials during World War I, but now they were asked to do much more. The American Automotive industry produced millions of military vehicles, and the majority of them were not even a domestic motorcar. They produced almost \$29 billion worth of materials, and that was one-fifth of all of the United States production. Not only was the industry making a bunch of money during the war, but post WWII was going to be prosperous, as there was a massive demand for new cars. So many of the materials required to build them were going to the military, so people had to keep their cars way longer than they wanted.

The United States was not the only country to fund their automotive industry during the war. This led to Europe and Japan also becoming big automotive producers, and that led to a massive shift later on. The United States did not go away from the idea of “reskinning” their cars. They made a bunch of similar cars, but changed to look and feel, in hopes of people buying new cars. This mass production led to cars that were big, bulky, not fuel efficient, and oozed air pollution. The United States passed multiple laws from 1965-1975 to help combat these issues. Another issue arose as there were massive increases in oil prices during 1973 and 1979. Because of these issues, the United States companies started producing less and less, and were importing a lot of more fuel-efficient and functional cars from Japan. The Volkswagen Bug was also incredibly popular from Germany as it was deemed “a modern Model T.” The sale of American cars went from 12.87 million in 1978, to just 6.95 million in 1982. During that time, imports rose 10%, going from only making up 17% of the United States market, to 27%. Japan also became the leading producer of automobiles, and it is still currently the leading producer.

In order to combat these drastic shifts in the industry, America started a massive overhaul. They invested \$80 billion to remake the factories, and they aimed at a smaller production, but good product model. If they did not make the change to smaller, more efficient cars, the United States would have lost all market share. They even started implementing “computerized” designs and manufacturing. This allowed for companies to still turn a profit, even with lower production, and still take care of their employees.

The automotive industry has a massive impact on not only American history, but also world history. It plays a massive factor in multiple industries alive like petroleum, steel, and industrial products. It also allowed for travel and tourism to flourish. It led to service stations, restaurants, and motels. It also led to the Interstate Highway Act of 1956, which kick started the hundreds of highways we have across the nation. It was also one of the largest public work programs in history.

### **3.2.1 How Cars Work**

The first step in building an RC car was to learn what it is we were trying to replicate. Modern day cars can be broken down into 6 subsystems: body and frame; engine; transmission/drivetrain, suspension system, wheels, and tires; control systems; and electrical systems; engine subsystems, and climatic equipment. These subsystems are all extremely critical to making a car run properly. Before we break down specifics, it should be noted that every car is different, so every car will have an engine, but there are a bunch of different types of them. It all depends on what the purpose of that car is to the market.

**3.2.1.1 Body and Frame:** The body of the car is an empty structure. Its goal is to provide a place for the driver, passengers, and cargo. It also safely stores all of the systems inside of it. The body is also what the windows, doors, handles, trunk, hood, and the frame are attached to. The frame is what the body is on top of. If you were to look down at it, it would look like a rectangle. It is meant to be a solid and square base for the body of the car to rest upon. There are two different uses for a frame. The most common one is when the body and frame are built together. It is called a unibody. This saves time and money in the process, but there are many benefits to having them as separate pieces. The main one being that you can have similar shaped cars, but change the body based on certain criteria. It is primarily used for off-road SUVs, which are all about customizing as many parts as possible. This flexibility could save you hundreds of thousands of dollars if you use it for multiple designs. This option is called body-on-frame. Again, the flexibility is a positive, but having to pay for two separate molds is extremely expensive. Manufacturers already have to pay for a mold to make the bodies, so you might as well include the frame in that mold, and skip that step entirely.

**3.2.1.2 Engine:** The main goal of an engine is to take one form of energy and convert it into one that allows the car to move. They are typically located in the front of cars, right under the hood. This lets the manufacturer put a trunk in the back, allowing for plenty of storage for your items. Some cars do put the engines in the back, and that is for performance. Some even have it in the middle of the car. The engine is the heaviest part of a car, so wherever it is, will affect how you drive. There are multiple different types of



“X-wheel drive.” We will discuss those later in the transmission section, but the short summary is that the engine turns the wheels. Wherever the engine is, it will put the weight on the wheels, and give it more traction. This weight helps you choose which wheel drive you want, and that depends on the cars’ application. For example, for people that live in ice places, you should have a 4-wheel drive car, because they work better to get traction everywhere. That does not matter in a place like Florida.

There are dozens of different types of engines on cars in 2024. The 3 main types are Internal Combustion Engines (ICE), Electric Motors, or Hybrid motors (a combination of the two).

**3.2.1.3 Internal Combustion Engines:** Before discussing all of the different kinds of ICEs there are, it is important to know how they work. ICEs are run using gasoline or diesel. Some of them can even run on natural gas, propane, biodiesel, or ethanol. There are two options: Spark ignition gasoline and compression ignition diesel engines. The spark ignition ones are the most common, but we will discuss all the types of compression ignition ones as well. ICEs work by combustion, or burning, which is what happens when we release the energy from air-fuel mixture. The fuel can be any of the types listed above.

There are four steps that an Internal combustion engine takes: Intake, Compression, Combustions, and Exhaust. Intake is when the engine sucks in a mixture of fuel and air, creating air-fuel. A piston starts this process by moving downwards. This creates a vacuum and draws air into the cylinder using the intake valve. The second step is compression. The cylinder that currently has a piston in a downward position and is filled with air-fuel, is extremely durable and is able to handle the force that is about to be put on it. The intake valve closes allowing the piston to move back up, and once the piston moves back up the air and fuel becomes compressed and causes pressure in the cylinder. The third step is combustion. Combustion is when the compressed fuel and gas explode using a catalyst. The most commonly used item to do this is a spark plug. A spark plug is ignited inside the cylinder and that creates a small explosion. This process happens over, and over, and over again as the engine is running. Each time there is an explosion, it leads us to the fourth step, which is exhaust. Both the intake and exhaust valves can open and close. The intake only opens when the engine is sucking in all of the air-fuel. During the other three steps, the intake is closed. The exhaust remains closed during steps one and two, but due to the forces from the explosion in the combustion step, it is forced open. The exhaust system leads this energy into the transmission, and that is what is connected onto the vehicle's axles. The transmission puts torque on the axles, causing the vehicle to move.

Once you find out that your engine is an ICE, there are many different types of those. Those types are the number and arrangement of the cylinders inside the engine. Since the cylinders are where all the micro explosions are happening, all the air and fuel gets compressed, the explosions sparks it, and that just keeps on pushing more and more energy to the exhaust. That energy goes to the transmission, which is putting all that

energy right into the wheels and moving the car. The more cylinders you have, the more output you will generate.

Internal Combustion Engines come in three different layouts: Straight/Inline, Flat, or V-shaped. The number of cylinders is a direct way to calculate how powerful your engine will be, but then you have to figure out how you want to put the engine itself in the body of the car. An Inline layout is the most common, because it is one most frequently used in everyday cars. This is when all the cylinders are perfectly vertical in a row, straight up and down. If this row is placed parallel to the car, the engine layout is called straight and if the engine is placed perpendicular to the car, this is an in-line layout. These engines are widely used because they are extremely inexpensive. You can put it all in a row, make sure everything is perfectly up, and then install it right into the body of the car. This vertical layout of the cylinders leads to a lot of vibrations. No matter the cylinder count, all of them have the intake and exhaust at the top, and a piston at the bottom. The piston goes down to intake all the air-fuel, and that causes force downwards. The piston then shoots back up, and compresses the air-fuel. This is an extremely strong upward force. Both forces happen at different times, so the body of the car will feel the full effects of them. The four-step process is extremely fast and happens repeatedly, so these forces show themselves in vibrations. An Inline engine can have both an even or odd number of cylinders. The odd number engines have a much more noticeable vibration, and that allows the three, five, and seven cylinders to be told apart easily.

The second type is a Flat engine. This is the complete opposite of the Inline engine, as instead of the cylinders being up and down, they are laid horizontally. A flat engine can only have an even number of cylinders, unlike the Inline engine. This is because they put the engine in like a horizontal "I" shape. The middle of the "I" holds the intake and exhaust, the lines leading to the two branches are where the cylinders are. The pistons are on the branches, so the movements of the engine mirror each other. When an inline engine fires, the forces go down and then up. For a flat engine, the piston retracts towards the exterior of the car, and then moves towards the middle, when compressing the air-fuel. This happens on both sides, so while one piston is moving to the right, the other is moving to the left, and vice versa. This causes an equal reaction of forces on the car, and it does not move. The flat engine also has a low center of mass so this is because it's much wider than something vertically. Instead of having the cylinders straight up and down, you are putting them sideways, and have them mirroring each other, so it doubles the length of a cylinder. The cars need a wider frame and they're extremely expensive because cars that size and shape are not desired anymore. Most people do not want them, and the manufacturers do not make them anymore. This started due to what happened in the 60s, when American companies were losing market share to Japanese companies, because Japan was making smaller, more efficient cars. The bigger you make the body to fit these engines, the less efficient the car will be, and the more expensive it was in the long run for the consumer.

The final ICE type is a V-shaped engine. It is the most common type of engine in performance vehicles. Instead of all of the cylinders being straight up and down like the inline engine, or completely horizontal like the flat engine, these are at a diagonal. When

they are at an angle, that allows the engine to have more cylinders fit inside of the car body. V-shaped offers the most performance compared to its size. It can pack in a ton of cylinder, and the cylinder allows for a ton of explosions which means even higher power performance. Most people also think that they look way cooler than inline and flat engines. Even though it does have a lot of positives, it does have its flaws. It is by far the most expensive to repair, because there are so many things packed in one space, so even though it has great performance, you will have to pay for it. One of the other big issues is the vibrations. It has the most vibrations out of the three options, due to the cylinder layout. This is because it's going left and right at an angle so those angles are not that force is not gonna be offset by the other one like it would for the flat engine, but the good thing is that they are smaller.

- Single Cylinder Engine: As discussed earlier, when breaking down the history of the automobile industry, there are multiple types of engines. The most basic one is a single cylinder engine. This is where all of the fuel is put inside of it, then the cylinder gets compressed via the piston, and then there's a spark, which goes right out the exhaust. Since there's only one cylinder and one explosion and one exhaust release, it is the least powerful, but it is the simplest and the cheapest most basic option.
- Twin Cylinder Engine: This is rarely found on cars now, because it is still super slow. Just like the single cylinder, it can be found on motorcycles or smaller things such as pressure washers and go karts. It is similar to the V-shape design, where both cylinders are at an angle, but they also can be vertical. Most the time, the vertical ones have more than two cylinders.
- Three Cylinder Engine: Three cylinders are very, very popular and they're known for the humming type of sound that they produce. They have an odd firing sequence, due to having an odd number of cylinders. This means that instead of just going one after another, they alternate and this causes a weird sound that we can immediately detect. They are more powerful than the two cylinder and the one cylinder, and sometimes these have inclusions of turbo chargers, which increase the power and maintain the efficiency of the engine. These are used on compact SUVs mainly.
- Four Cylinder Engine: A four-cylinder engine is most commonly used now because it is powerful and it's efficient. These are almost exclusively seen in the inline layout, because it's extremely easy to manufacture that way. There is no reason to put them in a V-shape because it is just four cylinders. You don't need to have them in diagonal because there's plenty of space inside the car because it is compact. This is the best option when looking for the balance of performance, power, and cost effectiveness. These are very popular in Honda and Hyundai cars that are used today.
- Five Cylinder Engine: An extremely rare configuration is the five cylinder. These are only found in luxury cars like Audi and Volvo. This is again because it has an odd number of cylinders just like the three cylinder configuration. Even though the three cylinder ones are odd, they are easier to repair and work on. A five cylinder is much harder to work on, hence it only being used in very nice cars. The five cylinder one also has a weird sound, called warbling.

- Six Cylinder Engine: A six cylinder engine is most commonly found on sports cars and they have a very high-pitched sound. This configuration is either built in the inline or the V-shaped engine. This is all depending on the car. If the manufacturing company is going for a slightly larger build, then it should be the V-shape. The V-shape is compact compared to how much power it outputs, but compared to an Inline engine, it is slightly larger horizontally (because the cylinders are at an angle). If it has a little bit of space to operate, it'll go with the straight line. The inline is much easier to work on and prepare, versus a V-shape being way more difficult and expensive, but you can fit more stuff in the V shape and get more performance out of it, compared to a similar size inline. With that being said, the inline saves you way more time and money during repairs. It can also be paired with a turbocharger/supercharger, and it gives it an immense amount of power. They are typically used in luxury cars like the BMW or Mercedes-Benz.
- Seven Cylinder Engine: Even though three cylinder engines are extremely common, they are basically the only odd number of chambers that you see. Luxury cars have 5, but those are obviously exclusive. You almost never see a seven cylinder engine. They are almost never manufactured, and they do have the humming noises that other odd number engines have.
- Eight (or more) Cylinder Engines: Eight cylinder configurations are used in supercars. These are almost always in a V-shape engine layout, because again you don't have unlimited space. You want to have it in a compressed area, all while having the most amount of power possible, and the V-shape is the best option for that. These are found in Bugattis, Ferraris, and even Lamborghinis. The base is eight cylinders, but then it goes ten, twelve, skips fourteen, and ends at sixteen. Manufacturers noticed that the difference between twelve and fourteen was not a good enough reason to justify building it, so they went straight up to sixteen cylinders. Sixteen cylinders is a very clever design, because they are placed in a W-shaped layout, rather than a V-shape. Manufacturers achieve this by taking the eight cylinders and slapping another one to the side of it. Together, the two V-shapes make a W. The sound of these engines are very similar to a lion, or a large cat's roar.

**3.2.1.4 Electric Motors:** The next major category of engines is electric motors. This is extremely important, because they are just larger versions of what we will use in our RC car build. Before getting into the five different types of electric motors, it is important to discuss how they all generally work. The most basic electric motor we have is a DC motor. It can be connected in series to a battery and a switch, and will run when the switch is closed. Electricity is the movement of electrons when there is a potential difference between two points. This is also what we refer to as current in Electrical Engineering. Current flows from the positive end to the negative end, so the orientation of a battery will determine what way the current flows.

Current can be turned on and off, so this directional flow can be stopped. A magnet is a metal object with two poles: north and south. A magnet will attract metals to it, but it has a different reaction to other magnets. Since magnets, no matter the shape or size, always

have two poles, this property will always apply: opposites attract, and the same side repels. This means that if the N side of a magnet gets close enough to the S side of another magnet, they will attract. If the N side of a magnet gets close enough to the N side of another magnet, they will repel. A magnet never turns off, so you have to create one that is similar. An electromagnet is when you wrap wire around metal and run a current through it. It does nothing when powered off, but when there is current flow, it will become a magnet with two sides. Electric motors work by placing an electromagnet on an axis, known as a rotor, and surrounding it with two other magnets, like two letter "C"s cupping a circle. One of the stators was a S pole, and the other was a N pole. It is important to note that they have a gap between them. Those stationary magnets that surround the rotor are known as the stator, because they do not move. The rotor having a N and S side, will spin downwards depending on whether or not it is in front of a N or S stator. If there is nothing else to this apparatus, the rotor will move down because it is attracted to one side, but then repel from the other side, and end up stationary in the gap. In order to change the force from repelling and keeping the rotor in the gap, you can flip the current by switching the orientation of the battery. When you switch the flow of electricity, you cause whatever side of the electromagnet was N, to be S, and vice versa. If you switch the flow of current right as the rotor is about to reach the gap, this causes it to attract again, and pull up the electromagnet. If you constantly switch the flow of current, the rotor will spin indefinitely. You could do this manually, but it is by no means realistic, so you use a component known as a commutator and a brush. A commutator is connected to the brushes like a wheel on an axle. As current flows through the brush, it causes the commutator to start the flow of electricity in the electromagnet. This allows the process of attraction to begin, and the rotor moves to the other side. Just as it reaches the point where repulsion would start, the commutator hits the brush on the other side, and this causes the current to switch. Once it switches, the poles on the rotor change again, and it attracts again. Electric motors use wire coils as the rotor for a much smoother transition of attraction and repulsion. There is an increase of torque, as the wire coils increase, and that causes the motor to spin even faster. In smaller applications, there is an extension of the rotor, and that axis is connected to an object, which spins/turns something else. In larger applications like a car, the energy gets output into the transmission and that turns the axle of a car, just like the ICEs do.

The transmission is connected directly to an engine, so that takes all of the output from the engine, and transmits a torque into the wheels. For an ICE, that output is all that exhaust from the mini explosions. The output in an electric motor is very different. Electric motors are always found in electric vehicles EV's and they're being extremely prevalent on our roads today. This means way more care is being taken into making sure those people have power stations to charge their vehicles. The way an electric motor operates is on electromagnetic fields: an electric current flows through a coil of wire, and in the presence of a magnetic field, it generates force, and that force causes the rotor to turn. These rotors are in direct drive with an electric motor that can control a pair of wheels so once these motors are getting turned then, these wheels are also being turned. One benefit compared to ICE is its instant torque. It is also extremely quiet and it has zero emissions. Electric Vehicles are going to continue becoming more and more improved, whereas we're almost at the absolute cap for ICEs, because there's only so

much you can possibly do without turning it into a hybrid. There are five types of electric motors: DC Series Motor, Brushless DC Motors, Permanent Magnet Synchronous Motor (PMSM), Three Phase AC Induction Motors, and Switched Reluctance Motors (SRM).

- DC Series Motor: The first type is the DC series motor. These are the most popular because they are powerful and efficient. These motors all vary in size, whether you're putting it in a truck or a SUV, or even a scooter, they all vary in size depending on the body of the car. DC Series Motors work by having a DC current coil, and that generates a magnetic field. Once the magnetic field is created, that creates rotational motion. Rotational motion is then put into the wheels, and that drives them. Some of the advantages are electricity consumption. Their efficiency increases as the speed decreases, which means instead of always having a minimum amount of consumption if you are going slower, you are going to consume less. This is great when you are in cities because electric vehicles do not have the longest battery life, so having a motor that consumes significantly less when you are just driving in a city, or when you are parking, or even when you are waiting in traffic. Also, when you just have to go slow because you are in a school zone. All these situations make it much better for electric vehicles over traditional ICE powered automobiles. The last thing they do is provide excellent speed control. This is because the motor is specifically working with exactly how much you put into the pedal. When you ease up on that, the machine immediately reacts to it and puts in less torque, so you can make it go faster and slower and control your speed just by moving your foot. Obviously the same thing goes for ICEs, because they are driven using gas pedals, but EVs are much more responsive than an ICE. Electric motors produce an immense amount of heat during operation so you need to have a bunch of temperature monitoring sensors, fans and a large cooling system installed around the motor. This requires maintenance and sometimes can be replaced, and the more you replace the cooler it will be, and the better the car was in the long run.
- Brushless DC Motors: The brushless motors are very similar to the DC series motor. The difference is it doesn't have any brushes, which means the motor itself can be more compact and have a simpler design, so heat management and efficiency are both increased. This motor does not need any mechanical communication. It does not need bearings, which saves on parts right there, and the less parts that it has it means less initial costs in buying the materials to build a motor. It also means there are less things that can cause issues, so there are less manufacturing issues and upkeep costs with maintenance. Again, because this system is so simple, brushless DC Motors have even lower energy losses compared to a DC series motor. They have a higher energy saving, and the whole purpose of getting an electric vehicle is to save gas mileage, and have the most effective and efficient ride possible. This is an incredibly good option, as it has even more benefits including having lower friction so longer life expectancy is longer than a DC series motor. It also has higher torque capacity and it's quieter than the DC motor.

- Permanent Magnet Synchronous Motor (PMSM): The third option is the permanent magnet synchronous motor (PMSM). These are way more efficient than the previous options. It is more energy efficient due to its relationship between the stator and rotor. It requires less current compared to other models, while still receiving the same exact output of torque and speed. The rotor design also has advantages for torque because this permanent magnet can be adjusted to accommodate all needs. You can also have different configurations. A brushless DC motor just has a rotor and stator, whereas a PMSM has different types, all depending on how many poles are used. You can either have a single phase or three phase version, and those are the two that are most commonly used in electric vehicles. You can also have different winding options.
- Three Phase AC Induction Motors: Three phase AC induction motors are a really great option for multiple reasons. They have a much higher torque control compared to other electric vehicles, and that means that they get an extremely precise power output. These also are extremely power efficient so they are used in situations where energy conservation is important, such as areas where there is not a lot of sunshine, so you can not charge them using solar panels, or if you are not close to outlets. In those cases, you need to get as much charge as you possibly can out of it. These motors also have an amazing cooling system. The motors have a much lower risk of overheating, but the downside is that it only has a specific power output. It does not vary for many applications, so if you were looking for a car that just goes extremely fast and in a straight line, or even if it is a truck, you need to really get some power out of it when you are putting a bunch of stuff in it. Whereas, if you are looking for an everyday vehicle, not being able to control the amount of power you are getting out of the car, when you need to be able to go fast on the highway or slow down a random street. This is not gonna be a great everyday car, but a positive about that extremely precise power control is how power efficient it is. This extreme power efficiency will save you money in the long run with charging, and this will allow you to go further if you're gonna be driving out and about with your electric vehicle.
- Switched Reluctance Motors (SRM): The last option is switched reluctance motors (SRM). These are an extremely great option for torque, but they are way more cost-effective, compared to three phase induction motors. These are both very good comparisons because they both give precise power output. SRMs are made with an arrangement of electromagnets, and these produce magnetic fields when they're turned on. This creates mechanical force required to drive the rotor, and makes a low variation as it rotates. This leads to a far more accurate torque, and is way easier to operate than other motors. They are also extremely efficient and have very low heat dissipation, which further their cost efficiency. The main benefit of a switched reluctance motor is how cost-effective it is while providing you a precise torque and speed ratings than all the other alternatives out there. It also has superior performance qualities like reduced levels of noise, pollution, and extreme reliability over time.

**3.2.1.5 Hybrid Engines:** The very last type of engine is a hybrid engine. These are a combination of internal combustion engines and electric motors. They operate on electric power at low speeds, and then switch to a combustion engine at a higher speed. The reason for this is because it does not need as much power at the lower speed, so it can use electricity, but it needs way more power at higher speeds, so it can use a combustion engine to get the most efficiency. The main advantages here are going to be fuel efficiency, because you're using electric power for city driving and then switch over onto a highway and you're able to go away faster when you need to. There is also reduced emissions, because Hybrids typically produce zero emissions. This means a clearer environment, and that was one of the issues that the United States was having back in the 60s, which led to a lot of the laws being passed to make manufacturers cut back on production. The best part about these cars is the regenerative braking system. They have electric motors, so when you hit the breaks, the torque is stronger, and stops the car quicker. When they stop on a dime, the kinetic energy is being converted back into electrical energy, which is stored into the battery. Every time it stops, the energy is going back into the battery, and batteries are then able to reuse it later on. It's only a small amount because the system loses energy to heating and everything surrounding it. You are still able to save some of that energy that otherwise would've been completely wasted in a regular engine.

**3.2.1.6 Transmission/Drivetrain:** The transmission is incredibly important, as it is what takes the energy from the engine, and puts it into the wheels. The wheels are connected to an axle, so the transmission puts a torque on it, and causes it to spin. All cars control the transmission using one of two ways: automatic or manual. Automatic makes driving much simpler, as the transmission goes on when you put the car into drive (D). For manual cars, you need to constantly change the gear number, as these cars work up to specific speeds. The number gear you are on will determine your max speed, so the higher you go up, the faster you can drive.

The other main component is what type of wheel drive the car is. Wheel drive refers to what the transmission is applying torque to. The options are front two-wheel drive, rear two-wheel drive, and four-wheel drive. The first two are simple to explain. Depending on whether you want the drive to be in the front or rear, that is where you put the engine. The engine being on top of those wheels is important, as it gives it additional traction. The transmission connects to the axle of the specific wheels, and then turns it. Because the other axle is not connected anywhere, the axle moving will cause all four tires to move, which moves the car. Four wheel different, as the transmission is connected to all four tires. This means you do not need the engine in a specific place, but there are plenty of reasons to pick where you want the engine. If the engine is in the front of a four-wheel drive car, there was another axle running down the middle of the car, connecting the front and back axles. The front wheels was moving from the transmission, but since the transmission is also connected to the rear view ones, it will move as well. This is great for areas that really need traction, because it gives the car more area trying to grip on the ground.



**3.2.1.7 Suspension System, Wheels, and Tires:** This is what is known as the “base” of the car. You can have the greatest body and engine, but if the base is not up to par, then the car will not perform. A suspension is to absorb vertical plane impact. Going over a speed bump or into a pothole are times when this is necessary. The suspension is built up of the wheel hub, springs, shock absorbers, links/arms/beams, anti-roll bar, and the subframe. The wheel hub is where tires are connected to the car, and as the wheel hub spins, so does the wheel. Springs are used to hold the car in place. If there is a bumpy road, this allows the car freedom to move in all directions, and still maintain its shape. Shock absorbers are pads that ease all of the pressure on surrounding components. Links/Arms/Beams are what help connect the axle to the frame of the car. By having them at right angles, it gives even more support than the absorbers. An anti-roll is just a way to keep the car in place. Normally a car will keep moving, even if it is parked on a hill. This ball allows all movement to stop immediately. The last piece is the subframe. This is what the suspension connects to the body of the car, and keeps everything in place.

**3.2.1.8 Control Systems:** The steering and braking systems is what makes up this section. We discussed how fast a car can go, but it needs to be able to stop, and turn the wheels.

**Electrical, Engine, and Climatic Subsystems and equipment:** With how advanced cars have gotten, they are absolutely covered in wires. These all serve a purpose, and the basics found in all cars include the power sources, consumers of said power, electrical wiring, and random elements. The two sources are the battery and generator. The generator builds up energy from the engine, and then distributes it to the car battery. The car battery then distributes this energy to the entire car. The consumers of power are things like headlights, GPS systems, and charging devices. Electrical wiring is what connects all of these components, and ensures that there is a constant flow of electricity to them. The random elements include buttons on the dashboard and switches like the ones to roll down windows.

### **3.3 Technology Overview and Comparison**

The RC Car was revolutionized with cutting-edge technology, as it was controlled seamlessly through a dedicated mobile application. This innovative approach not only enhanced user experience but also opened up a realm of possibilities for customization and advanced functionalities. Leveraging Arduino technology, the app empowered users to control the car effortlessly, with features such as directional control, speed adjustment, and even programmable routes. Technologies like Arduino IDE, which offers a user-friendly platform for programming Arduino boards, and libraries such as Bluetooth Low Energy (BLE) communication protocols, facilitate the development of the app. Additionally, frameworks like MIT App Inventor and platforms such as Blynk enable the creation of intuitive user interfaces and seamless connectivity between the mobile device and the Arduino-powered RC Car, ensuring a seamless and immersive control experience. With these tools at hand, the RC Car becomes not just a toy, but a gateway to explore the fascinating intersection of mobile app development and robotics.

### **3.3.1 Possible Technologies**

#### **1. Arduino IDE:**

The Arduino Integrated Development Environment (IDE) is a software platform used to write and upload code to Arduino boards. It provides a streamlined interface for writing, compiling, and uploading code to Arduino microcontrollers. Arduino IDE supports a variant of the C++ programming language, making it accessible to beginners while still offering advanced features for experienced developers. Its simplicity and compatibility with a wide range of Arduino boards make it a popular choice for programming Arduino-based projects like the RC Car.

#### **2. Bluetooth Low Energy (BLE):**

Bluetooth Low Energy is a wireless communication protocol designed for short-range communication with low power consumption. It enables devices like smartphones and tablets to communicate with peripherals such as Arduino-based projects over short distances. BLE is particularly suitable for applications like controlling an RC Car from a mobile app, as it allows for reliable communication while conserving battery life. Arduino boards equipped with BLE modules or shields can establish a connection with a mobile device, enabling bidirectional communication for controlling the RC Car.

#### **3. MIT App Inventor:**

MIT App Inventor is a web-based platform developed by the Massachusetts Institute of Technology (MIT) for creating Android applications without requiring extensive programming knowledge. It features a visual drag-and-drop interface that allows users to design the user interface and define the functionality of their apps using blocks of code. MIT App Inventor provides a beginner-friendly environment for developing mobile apps to control Arduino-based projects like the RC Car. Users can easily integrate features such as buttons, sliders, and sensors into their apps to control the car's movements and behavior.

#### **4. Blynk:**

Blynk is a platform that simplifies the process of building IoT (Internet of Things) projects by providing a comprehensive suite of tools for developing mobile apps and connecting them to hardware devices. It offers a range of widgets and libraries that enable users to create customized interfaces for controlling and monitoring Arduino-based projects remotely. Blynk supports various communication protocols, including Wi-Fi, Bluetooth, and Ethernet, allowing users to choose the most suitable connectivity option for their projects. With Blynk, developers can quickly prototype and deploy mobile apps to control the RC Car, leveraging its intuitive interface and robust backend infrastructure.

### **3.3.2 Comparison of Technologies**

Before diving into the comparisons of various technologies for controlling Arduino-based RC Cars, it's essential to understand the landscape of tools available for such projects. These technologies play a crucial role in shaping the development process, influencing factors such as ease of use, popularity, and complexity. By evaluating each technology against these criteria, developers can make informed decisions about which tools best suit their needs. Let's explore the ratings of Arduino IDE, Bluetooth Low Energy (BLE), MIT App Inventor, and Blynk, considering factors such as ease of use, popularity, and complexity, to gain insights into their suitability for creating apps to control Arduino-powered RC Cars.

#### **1. Arduino IDE:**

- Ease of Use: 4/5 - Arduino IDE is generally considered user-friendly, especially for beginners due to its simple interface and extensive documentation.
- Popularity: 5/5 - Arduino IDE is highly popular within the maker and hobbyist community, with a large user base and active online forums for support.
- Complexity: 3/5 - While Arduino IDE is relatively straightforward for basic projects, more complex functionalities may require a deeper understanding of programming concepts.

#### **2. Bluetooth Low Energy (BLE):**

- Ease of Use: 3/5 - BLE can be moderately challenging for beginners due to its technical nature, but there are libraries and resources available to simplify implementation.
- Popularity: 4/5 - BLE is widely used in IoT and mobile applications, making it a popular choice for connecting Arduino projects to mobile devices.
- Complexity: 4/5 - Implementing BLE communication can be complex due to factors such as pairing, security, and compatibility with different devices.

#### **3. MIT App Inventor:**

- Ease of Use: 5/5 - MIT App Inventor is exceptionally user-friendly, with a visual drag-and-drop interface that simplifies app development, particularly for beginners.
- Popularity: 4/5 - MIT App Inventor is popular among educators, students, and hobbyists for its simplicity and accessibility.
- Complexity: 2/5 - While MIT App Inventor is easy to use, it may lack the flexibility and advanced features required for more complex app development projects.

#### **4. Blynk:**

- Ease of Use: 4/5 - Blynk provides a user-friendly platform for building IoT projects, with intuitive drag-and-drop widgets and straightforward configuration options.
- Popularity: 4/5 - Blynk has gained popularity in the IoT community for its versatility and ease of use, with a growing user base and active community support.
- Complexity: 3/5 - While Blynk simplifies the process of building IoT applications, setting up and configuring the platform may require some technical expertise, especially for advanced features.

### **3.3.3 Connecting The Controller To The Remote Control Car**

A controller was designed to control our car remotely. Three different wireless technologies were discussed for controlling the car remotely: Wifi, Bluetooth, and infrared transmission.

#### **3.3.3.1 Wireless vs. Wired Communication**

The controller needed to be able to communicate to the remote control car. To do this, there were two types of communication that needed to be considered: wired and wireless communication. The advantages of wired communication are that it would be faster and more stable because there would be less interference from other sources. However, there was no need for a higher data transfer rate because the communication between the controller and the remote control car was simple. Also, having a wired connection between the controller and the car meant there was less range that the car can travel based on how long the physical connection is. Therefore, we decided that some form of wireless connection would be the best as it would extend the range that the car can travel. Typically, toy cars are remote controlled and use a wireless connection due to this reason.

#### **3.3.3.2 Wifi vs. Bluetooth**

Two common forms of wireless communication are Wifi and Bluetooth. For our project, we considered both for connecting the controller to the car. Both Wifi and Bluetooth operate at 2.4 GHz, so they both had similar interferences. Wifi can also operate at 5 GHz, but the higher speed that comes with the higher frequency was not necessary as there was not much data transfer between the controller and the car. Wifi offered higher data transfer speeds than Bluetooth. Bluetooth had higher latency than Wifi but the difference was negligible for our purpose. Wifi typically offered a higher range than Bluetooth, however, we did not need as much range as Wifi provided. Bluetooth was more ideal because it used less power due to having a lesser speed and Bluetooth is typically used for connecting two devices. Also, we did not need the higher range and the higher speed that Wifi provides. The mobile application did also use the Wifi interface, so it may not have been ideal to be having both the controller and the mobile application using the same interface if for some reason we wanted to use both the application and the controller simultaneously. However, we considered another option: infrared.

### **3.3.3.3 Bluetooth vs. Infrared Transmission**

Another option of wireless communication was infrared. Both Bluetooth and infrared transmission are used to transfer data over short distances. However, Bluetooth devices did not have to be within line-of-sight of each other. Infrared transmission requires a line-of-sight connection. You will probably not be driving the remote control car unless you could see where you are driving it so this does not matter. Infrared transmission also has less range than Bluetooth, however, the person using the controller would also not be very far from the car. Bluetooth can connect up to multiple devices at a time, while infrared transmission works on a one-to-one basis where one device will communicate to the other one. Compared to Bluetooth, infrared transmission is also much simpler to use and there is less interference. In addition, infrared transmission is typically used for remote control cars.

### **3.3.3.4 Selecting Technology To Connect The Controller To The Remote Control Car**

The comparisons above can be summarized in the table below. Each plus indicates a higher value.

	Wifi	Bluetooth	Infrared
Speed	+++	++	+
Range	+++	++	+
Interference	++	++	+
Cost	++	++	+

To summarize, three different wireless technology options were discussed and compared: Wifi, Bluetooth, and infrared transmission. It was ultimately decided to use infrared transmission to connect the controller to the remote control car because it is simple, there is less interference from other devices, and both Bluetooth and Wifi offer higher ranges that are not needed. Also, Wifi offers higher speeds that are not necessary for simple communication between the controller and the remote control car. The Wifi interface will also be used to connect the mobile application to the remote control car, so it would not be ideal if we wanted to use the controller and the mobile application simultaneously. Also, it is typical for remote control cars to communicate with the controller via infrared transmission, so it is clearly a good option. Infrared transmission is also much simpler compared to Bluetooth and Wifi. Wifi and Bluetooth devices cost more than an infrared transmitter and an infrared receiver, however, this may not matter if the microcontroller we used comes with in-built Wifi and/or Bluetooth. In conclusion, we chose infrared transmission because there is less interference and it is simple compared to Bluetooth or Wifi.

### **3.3.4 Infrared Transmission Protocols**

In this section, we will discuss two of the more common protocols used in infrared transmission. The two protocols we discussed are: the NEC protocol and the RC5 protocol.

#### **3.3.4.1 NEC Protocol**

The NEC protocol is an infrared transmission protocol based on a technique called pulse distance encoding. The NEC protocol uses a carrier frequency of 38 kHz. The protocol distinguishes between 0's and 1's by using pulses of different lengths. In the NEC protocol, a logical 1 is encoded in a 562.2 microsecond burst followed by a 1.687 ms low period. A logical 0 is represented by a 562.2 microsecond burst followed by a 562.2 microsecond low period. Therefore, in the NEC protocol, the difference between a logical 1 and a logical 0 is in the width of the low period. The message frame for the NEC protocol consists of a 9 ms leading burst to start the transmission, a 4.5 ms low period, the 8-bit address of the device, the logical inverse of the first 8 bits, the 8 bit command, the logical inverse of the last 8 bits, and a 562.2 microsecond burst to end the transmission. To summarize, the first 8 bits of the message is the device address and the second 8 bits is the command. The inverse of the first and the second 8 bits is used to check that the address and the command were sent and received correctly. The NEC protocol can support repeat commands to send repeat messages to the receiver.

#### **3.3.4.2 RC5 (Philips) Protocol**

The RC5 protocol, also called the Philips protocol, is based on an encoding technique called Manchester encoding. The RC5 protocol uses the order of the pulses and breaks in the message to represent the different bits. The RC5 protocol uses a 36 kHz carrier frequency, whereas the NEC protocol uses a 38 kHz carrier frequency. In the RC5 protocol, a logical 1 is represented by a 889 microsecond low period followed by a 889 microsecond burst. A logical 0 is represented by a 889 microsecond burst followed by a 889 microsecond low period. Therefore, the width of the pulse does not vary between bits and the value of the bit is determined by whether the burst is first or if the low period is first. The message frame contains two start bits, which are both high, to determine the start of the message, a toggle bit, five address bits, and six command bits. The toggle bit is used to determine whether a repeat message is being sent. For example, if a button is being pressed once, the infrared emitter will just send one message and each time the button gets pressed, the toggle bit was toggled so the device can know if the same message is repeated by a button press or by the button being held down. If a button is held down, the infrared emitter may send multiple repeat messages, so each time the toggle bit will remain the same unless it is a new copy of that message, so that the receiving device can detect repeat messages. In the RC5 protocol, a message takes approximately 24.892 ms to send.

### **3.4 Part Overview and Comparison**

In this section, we will discuss and compare various components that were considered for use in our project. We will compare remote control car kits, microcontrollers, infrared emitters, infrared receivers, motor driver ICs, DC/DC converters, and step-up converters for use in our project.

### **3.4.1 Microcontroller**

In this section, we will discuss and compare several different microcontrollers to be considered for our custom PCB for the controller. The microcontroller was reading from inputs such as buttons and an analog stick, and was sending information over an infrared transmitter to the remote control car. The four microcontrollers that we were considering are: the ESP8266, the RP2040, the ATmega328P, and the ESP32.

#### **3.4.1.1 ESP8266**

The ESP8266 microcontroller is designed by Espressif Systems. The ESP8266 features integrated WiFi, a 32-bit RISC processor, a GPIO interface, an SDIO interface, an SPI interface, an I2C interface, an I2S interface, a UART interface, a PWM interface, an IR remote control interface, and an ADC. The integrated WiFi has 802.11 b/g/n support with a maximum speed of 72.2 Mbps at 2.4 GHz. WiFi channels 1 through 14 are also supported. The 32-bit processor is a Tensilica L106 32-bit RISC processor, which has a maximum clock frequency of 160 MHz. The ESP8266 also has 50 KB of SRAM. The microcontroller has no internal memory, so the code must be stored in an external flash memory module. The microcontroller requires a minimum of 512 kB of external flash memory and can support up to 16 MB of external flash memory. The ESP8266 has a total of 32 pins. 17 of those pins are GPIO pins which can be multiplexed with the I2C, I2S, UART, PWM, IR remote control, and the ADC functions. The ESP8266 requires a minimum operating voltage of 2.5V, and a maximum operating voltage of 3.6V. So, the ESP8266 has an operating voltage of 3.3V. The maximum current the ESP8266 can draw is 12 mA. The ESP8266 is currently available at a cost starting from \$1.60 on the Mouser Electronics store.

#### **3.4.1.2 RP2040**

The RP2040 microcontroller is designed by Raspberry Pi Ltd. It is used by the Raspberry Pi Pico microcontroller board and other microcontroller boards. The RP2040 contains a 32-bit dual-core ARM Cortex-M0+ processor, 264 kB of SRAM, 30 GPIO pins. Like the ESP8266, the RP2040 does not have internal memory, so the code must be stored in an external memory module. Like the ESP8266, the RP2040 can support up to 16 MB of external flash memory. Of the 30 GPIO pins, 4 of those pins can be used as analog inputs and 2 pins support UART. There are 2 SPI controllers and 2 I2C controllers, there are 16 PWM channels, there is a USB 1.1 controller, and 8 PIO state machines. Unlike the ESP8266, The RP2040 does not have integrated WiFi. Like the ESP8266, The RP2040 has an operating voltage of 3.3 V. The RP2040 has a maximum current of 50 mA. The RP2040 also has a maximum clock frequency of 133 MHz. The RP2040 is currently available at a cost starting from \$0.70 on the Mouser Electronics store.

#### **3.4.1.3 ATmega328P**

The ATmega328P is a 8-bit microcontroller designed by Atmel. The ATmega328P is used by the Arduino Uno R3 microcontroller board and other microcontroller boards and is also used in the automotive industry. The ATmega328P includes 2 KB of SRAM, 6 PWM channels, an SPI interface, an I2C interface, and an 8-channel 10-bit ADC. Unlike the ESP8266 and the RP2040, the ATmega328P has 32 KB of internal flash memory. It

does not have external flash memory support nor does it have integrated WiFi. The microcontroller also has a maximum clock frequency of 16 MHz. The ATmega328P has a minimum operating voltage of 2.7 V and a maximum operating voltage of 5.5 V. At a clock frequency of 16 MHz, and an operating voltage of 5 V, the ATmega328P has a maximum current draw of 14 mA. The ATmega328P is available at a cost starting from \$1.56 on the Mouser Electronics store.

#### **3.4.1.4 ESP32 Series**

The ESP32 microcontroller is designed by Espressif Systems. The ESP32 features integrated WiFi, integrated Bluetooth, a 32-bit Xtensa LX6 processor, 520 KB of SRAM, 34 GPIOs, a 12-bit ADC, two 8-bit DACs, 10 touch sensors, 4 SPI interfaces, 2 I2S interfaces, 2 I2C interfaces, 3 UART interfaces, an ethernet interface, a motor PWM, and an LED PWM with up to 16 channels. The ESP32's integrated WiFi has 802.11 b/g/n support and can transfer up to 150 Mbps. Depending on the specific model number, the ESP32 can come with internal flash storage or no internal flash storage so it may or may not require external flash memory depending on the model number purchased. The ESP32 can support multiple flash memory modules and RAM modules via a QSPI interface. The minimum operating voltage is 1.8 V, and the maximum operating voltage is 3.6 V. So, the ESP32 can operate at a voltage of 3.3V. The maximum current that can be supplied is 500 mA.

There are several different microcontroller models for the ESP32 series: the ESP32-D0WD-V3, the ESP32-D0WDR2-V3, the ESP32-U4WDH, the ESP32-D0WDQ6-V3, the ESP32-D0WD, the ESP32-D0WDQ6, and the ESP32-S0WD. All ESP32 models can operate at a supply voltage of 3.3 V, with the ESP32-D0WDR2-V3 and the ESP32-U4WDH exclusively requiring 3.3 V. The other models can also operate at a supply voltage of 1.8 V. The ESP32-S0WD is the only single-core model for the ESP32 series. The rest are dual-core microcontrollers. The ESP32-U4WDH is the only model in the ESP32 series that contains in-package flash memory. Therefore, the ESP32-U4WDH is the model that we used out of all of the models in the ESP32 series. The ESP32-U4WDH contains 4 MB of in-package flash memory. The ESP32 is currently available at a cost of \$1.88 on the Mouser Electronics store.

#### **3.4.1.5 Microcontroller Summary**

The comparisons above can be summarized in the table below:



	ESP8266	RP2040	ATmega328P	ESP32 Series
Manufacturer	Espressif Systems	Raspberry Pi Ltd.	Atmel	Espressif Systems
WiFi?	Yes	No	Yes	Yes
Bluetooth?	No	No	No	Yes
Internal Flash Storage?	No	No	Yes	Depends on model number
SRAM Capacity	50 KB	264 KB	2 KB	520 KB
# of GPIO Pins	17	29	23	34
# Of I2C Interfaces	1	2	1	2
# Of SPI Interfaces	2	2	1	4
# Of UART Interfaces	2	2	1	3
Operating Voltage	3.3 V	3.3 V	3.3 V/5 V	3.3 V
Maximum Current Draw	12 mA	50 mA	14 mA	500 mA
Price	\$1.60	\$0.70	\$1.56	\$1.85

To summarize, the RP2040 is the only microcontroller of the four considered that does not have integrated WiFi. The ESP32 is the only microcontroller that has integrated Bluetooth. The controller does not require Wifi or Bluetooth capability, so that was not a factor in determining which microcontroller to use. The ESP32 has the most SRAM capacity at 520 KB and the ATmega328P has the least SRAM capacity at 2 KB. The RP2040 has 264 KB of SRAM capacity and the ESP8266 has 50 KB of SRAM capacity. The ATmega328P may not have enough SRAM for our project, so that microcontroller will not be considered because we do not want that to constrain the performance. Also, since the controller will have a limited number of peripherals, the number of I/O interfaces and the number of GPIO pins does not matter too much. All of the microcontrollers are also priced similarly so that is not a factor in choosing a microcontroller to use. The ESP32 has more features than is needed so it is not necessary. Initially, the ESP8266 microcontroller was chosen for our custom PCBs because it is a simplistic microcontroller with minimal features and has sufficient SRAM capacity. The ESP8266 also draws the least power, which was advantageous to extend the battery life of the controller. The number of GPIO pins and I/O interfaces on the ESP8266 are also

sufficient. However, we finally decided to use the ESP32 because there is an ESP32 model that has internal flash memory and it would have been difficult to upload our code to an external flash memory module. The ESP32 also has a higher SRAM capacity and a sufficient number of GPIO and I/O interfaces. The ESP32 also has higher performance than the ESP8266. Therefore, we can ensure we will not run out of SRAM or lack the processor performance to run our code. So, in order to simplify the PCB and make it easier to upload the code to the microcontroller, the ESP32 was chosen over the ESP8266.

### **3.4.2 Infrared Emitter**

The custom PCB for our controller required an infrared emitter to send data to the remote control car. The ESP8266 has an in-built infrared remote control interface, however, this was not sufficient for our project because the range is limited to about 1 meter. Ideally, we wanted an IR emitter that emits a wavelength of 940 nm, as the maximum spectral sensitivity is at that wavelength for an IR emitter, so the IR emitter will have the maximum range at that wavelength. Also, we wanted an IR emitter with a side view because the controller will need line of sight to the remote control car and will need to be pointed at the remote control car. In this section, we discussed and compared three different options to use for our infrared emitter.

#### **3.4.2.1 CSL1501R3T1**

The CSL1501R3T1 is an IR LED created by ROHM Semiconductor. The CSL1501R3T1 is a side view, surface mount IR LED. This model has a maximum power dissipation of 100 mW, a maximum forward current of 50 mA, and a maximum peak forward current of 200 mA. The forward voltage across the LED is 1.5 V. The radiant intensity of the LED is 2.5 mW/sr. The wavelength of this LED is 940 nm. The CSL1501R3T1 also has a viewing angle of 70 degrees. The CSL150R3T1 is currently available at a cost of \$0.71 on the Mouser Electronics store.

#### **3.4.2.2 VSMB10940**

The VSMB10940 is an IR LED created by Vishay Semiconductors. It is a side view, surface mount IR LED. This IR LED has a maximum forward current of 65 mA, a maximum peak forward current of 130 mA, and a maximum power dissipation of 104 mW. The forward voltage across the LED is 1.3 V. The radiant intensity of the LED is 3.05 mW/sr at a forward current of 65 mA. The wavelength of this LED is 940 nm. The viewing angle of the LED is 75 degrees. Compared to the previous model, this is a very similar model. The viewing angle is slightly higher, the power dissipation is slightly higher, the maximum forward current is slightly higher, and the radiant intensity is slightly higher. However, this is a cheaper model than the CSL1501R3T1. The VSMB10940 is currently available at a cost of \$0.39 on the Mouser Electronics store.

#### **3.4.2.3 IN-S126ESGHIR**

The IN-S126ESGHIR is an IR LED created by Inolux. It is a side view, surface mount IR LED. This IR LED has a maximum power dissipation of 180 mW, a maximum forward current of 100 mA, and a maximum peak forward current of 1000 mA. The forward voltage across the LED is 1.5 V. The radiant intensity of the LED is 92 mW/sr. The

wavelength of this LED is 940 nm and the viewing angle is 30 degrees. Compared to the other two models, this IR LED is a higher power LED with a higher radiant intensity. The viewing angle is also more narrow, which means this IR LED will have better range and connectivity than the other two. The IN-S126ESGHIR is currently available at a cost of \$0.55 on the Mouser Electronics store.

#### **3.4.2.4 Infrared Emitter Summary**

The comparisons above can be summarized in the following table:

	CSL1501R3T1	VSMB10940	IN-S126ESGHIR
Manufacturer	ROHM Semiconductor	Vishay Semiconductors	Inolux
Max. Power Dissipation	100 mW	104 mW	180 mW
Max. Forward Current	50 mA	65 mA	100 mA
Max. Peak Forward Current	200 mA	130 mA	1000 mA
Forward Voltage	1.5 V	1.3 V	1.5 V
Radiant Intensity	2.5 mW/sr	3.05 mW/sr	92 mW/sr
Viewing Angle	70 degrees	75 degrees	30 degrees
Cost	\$0.71	\$0.39	\$0.55

In summary, all three models compared were side view IR LEDs with a wavelength of 940 nm. The CSL1501R3T1 and the VSMB10940 both had similar power dissipation, similar forward current, similar forward voltage, similar viewing angle, and similar radiant intensity. The VSMB10940 was cheaper than the CSL1501R3T1 and had slightly higher maximum power dissipation, maximum forward current, radiant intensity, and viewing angle. This means that this LED was slightly more powerful than the CSL1501R3T1 and would have slightly better transmissivity and slightly better range. However, the IN-S126ESGHIR was the most powerful LED out of the three and was in the middle in the price range. The maximum power dissipation for this model was 180 mW, which is substantially higher than the other two models. The maximum forward current was 100 mA. The forward voltage was 1.5 V, which was similar to the other two models. The IN-S126ESGHIR also had a radiant intensity of 92 mW/sr, which was substantially higher than the other two models and had a narrower viewing angle of 30 degrees. This means the infrared beam would be more concentrated and higher power, so this LED would have the best range and the best transmissivity out of the three models. In conclusion, the IN-S126ESGHIR would be the best IR emitter out of the three and that was what was chosen for the project.

### **3.4.3 Infrared Receiver**

For the PCB that will control the components of the car, we needed an infrared receiver to receive the commands from the infrared LED on the remote controller. The infrared receiver will need to have a demodulating circuit and a band-pass filter and will need to be able to receive a signal with a carrier frequency of 38 kHz because we used the NEC infrared transmission protocol, which uses a carrier frequency of 38 kHz to transmit the signal. The internal demodulator and the internal band-pass filter will also allow us to avoid creating a demodulator and a band-pass filter on the PCB, which will simplify the design process. The infrared receiver should filter out frequencies outside of the 38 kHz range, so we do not have to be worried about interference at other frequencies, and the infrared receiver should demodulate the signal so that the microcontroller can process it. We were comparing and discussing several different infrared receivers: the TSOP4838, the TSOP38238, the GP1UM271RKVF, and the IRM2638.

#### **3.4.3.1 TSOP4838**

The TSOP4838 is a through-hole infrared receiver created by Vishay Semiconductors. It can receive an infrared signal with a carrier frequency of 38 kHz. This infrared receiver is recommended to be used with the RC5, RC6, Panasonic, NEC, Sharp, r-step, Thomson RCA, Sony, Mitsubishi, Sejin 4PPM, or the Cisco IR transmission protocols. This infrared receiver requires a minimum supply voltage of 2V and a maximum supply voltage of 5.5V. The TSOP4838 is rated for a maximum supply current of 5 mA and can consume up to 10 mW of power. However, the typical current usage is between 0.25 mA and 0.45 mA. The transmission distance using a TSAL6200 IR LED and a forward current of 50 mA was approximately 24 meters. The TSOP4838 infrared receiver is currently available at a cost of \$1.17 on the Mouser Electronics store.

#### **3.4.3.2 TSOP38238**

The TSOP38238 is a through-hole infrared receiver created by Vishay Semiconductors. It is very similar to the TSOP4838 through-hole infrared receiver, which is also created by Vishay Semiconductors. It can receive an infrared signal with a carrier frequency of 38 kHz. The TSOP38238 is recommended to be used with the RC5, RC6, Panasonic, NEC, Sharp, Mitsubishi, or the Thomson RCA IR transmission protocols. Like the TSOP4838 IR receiver, the TSOP38238 requires a minimum supply voltage of 2V and a maximum supply voltage of 5.5V. It is also rated for a maximum supply current of 5 mA and can consume up to 10 mW of power. However, the typical current usage is between 0.25 mA and 0.45 mA. The transmission distance using a TSAL6200 IR LED and a forward current of 50 mA was approximately 30 meters, which is slightly higher than the 24 meters offered by the TSOP4838. The shape of the TSOP38238 is similar to the shape of an LED, whereas the TSOP4838 is in a different form similar to a sphere. The TSOP38238 is currently available at a cost of \$0.99 on the Mouser Electronics store.

#### **3.4.3.3 GP1UM271RKVF**

The GP1UM271RKVF is a through-hole infrared receiver created by Sharp Microelectronics. It can receive an infrared signal with a carrier frequency of 38 kHz. This infrared receiver requires a minimum supply voltage of 4.5 V and a maximum supply voltage of 5.5 V. The typical current usage of the GP1UM271RKVF is 0.5 mA.

The transmission distance tested was 4 meters to 8.5 meters, which is considerably less than the transmission distance offered by the TSOP4838 and the TSOP38238. The GP1UM271RKVF is currently available at a cost of \$0.604 on the Mouser Electronics store. However, a minimum order of 2,000 components is required to purchase the GP1UM271RKVF and the stock is very limited on other electronics components stores. So, this infrared receiver will not be ideal.

#### **3.4.3.4 IRM2638**

The IRM2638 is a through-hole infrared receiver created by Everlight Electronics Company. It can receive an infrared signal with a carrier frequency of 38 kHz. It requires a minimum supply voltage of 4.5 V and a maximum supply voltage of 5.5 V. The typical current usage of the IRM2638 is 1.1 mA. The transmission distance tested with this infrared receiver was 6 to 12 meters, which is more than the transmission distance offered by the GP1UM271RKVF, however, it is much less than the transmission distance offered by the TSOP4838 and the TSOP38238 infrared receivers. The IRM2638 is currently available at a cost of \$0.239 on the Mouser Electronics store, however, a minimum order of 300,000 is required. Like the GP1UM271RKVF, the stock is very limited on other electronics components stores. So, this infrared receiver will also not be ideal.

#### **3.4.3.5 Infrared Receiver Summary**

The comparisons above can be summarized in the table below:

	TSOP4838	TSOP38238	GP1UM271RKVF	IRM2638
Manufacturer	Vishay Electronics	Vishay Electronics	Sharp Microelectronics	IRM2638
Supply Voltage	3.3 V/5 V	3.3 V/5 V	5 V	5 V
Current Usage	0.45 mA	0.45 mA	0.5 mA	1.1 mA
Maximum Transmission Distance	24 meters	30 meters	8.5 meters	12 meters
Cost	\$1.17	\$0.99	\$0.604	\$0.239

In conclusion, we have chosen the TSOP38238 infrared receiver because it has the least current usage, it can operate at supply voltages 3.3 V and 5 V, and can receive infrared signals at the highest distance out of the four infrared receivers discussed. The IRM2638 uses the most current and the TSOP4838, the TSOP38238, and the GP1UM271RKVF use less current. The TSOP4838, the TSOP38238, and the GP1UM271RKVF use a similar amount of current. The IRM2638 and the GP1UM271RKVF offer a smaller transmission distance than the two infrared receivers from Vishay Electronics. Also, the GP1UM271RKVF and the IRM2638 both have limited availability online, so to ensure that we are able to purchase an infrared receiver, we must purchase one of the IR receivers from Vishay Electronics. Therefore, we chose the TSOP38238 over the

TSOP4838 because although they are both very similar, the TSOP38238 offers a slightly higher transmission distance than the TSOP4838.

### **3.4.4 External Flash Memory**

Initially, we decided to use the ESP8266 microcontroller on the custom PCB, which would require external flash memory for storing our code. The ESP8266 microcontroller will need to be configured to read our code from the external flash storage module via an SPI interface. However, we ultimately decided to use the ESP32 microcontroller, which has a model which contains internal flash memory storage. In this section, we will discuss and compare different options to use for the external flash storage module if we had decided to keep using the ESP8266 microcontroller. The different options we discussed are two similar models from ISSI, three similar models from Winbond Electronics, and a model from onsemi.

#### **3.4.4.1 IS25WP016D/IS25LP016D**

The IS25WP016D model and the IS25LP016D model are flash memory storage chips created by ISSI. This model has a memory capacity of 16 Mb. This model has a maximum clock frequency of 133 MHz in fast read mode and has a maximum clock frequency of 50 MHz in normal mode and supports up to 66 Mbytes/s of data throughput. Data is stored in 256 byte pages. This model offers 100,000 erase cycles and more than 20 years of data retention. The IS25WP016D has a supply voltage between 1.65 V to 1.95 V and the IS25LP016D has a supply voltage between 2.3 V to 3.75 V, so the IS25LP016D would be more ideal for our project because our microcontroller requires a supply voltage of 3.3 V, so both could use the same supply voltage. This model has a maximum current usage of 25 mA. The IS25WP016D model is currently available for \$0.65 at the Mouser Electronics store and the IS25LP016D model is currently available for \$0.57 at the Mouser Electronics store.

#### **3.4.4.2 W25Q16RV/W25Q16JW/W25Q16JV**

The W25Q16RV, the W25Q16JW, and the W25Q16JV models are flash memory storage chips created by Winbond Electronics. Similar to the ISSI models, all of these Winbond Electronics models have a memory capacity of 16 Mb and a maximum clock frequency of 133 MHz. Also, the Winbond Electronics models have a data transfer rate of up to 66 Mbytes/s. The Winbond Electronics models have 256 byte programmable pages. These models can withstand a minimum of 100,000 erase cycles and have more than 20 years of data retention. The W25Q16JW model requires a supply voltage between 1.65 V and 1.95 V. The W25Q16RV model and the W25Q16JV model require a supply voltage of 2.7 V to 3.6 V. Either the W25Q16RV model or the W25Q16JV model is more ideal because it could use the same supply voltage as the microcontroller. The W25Q16JV model uses a maximum current of 25 mA, the W25Q16JW model uses a maximum current of 20 mA and the W25Q16RV model uses a maximum current of 15 mA. Therefore, the W25Q16JV model would be more ideal because it uses slightly less power. The Winbond Electronics models are almost identical to the ISSI models except the Winbond Electronics W25Q16JV model uses slightly less current. The W25Q16JV model is currently available starting from \$0.50 at the Mouser Electronics store.

#### **3.4.4.3 LE25U40PCMC-AH**

The LE25U40PCMC-AH model is a flash memory storage chip created by onsemi. This model has a memory storage capacity of 4 Mb, which is less than the 16 Mb of capacity offered by the ISSI and the Winbond Electronics models. This model has a clock frequency of 30 MHz and has a supply voltage between 2.3 V and 3.6 V, so it can use the same supply voltage as the microcontroller. Each programmable page is 256 bytes. Similar to the ISSI and Winbond Electronics models, this model offers 100,000 erase cycles and 20 years of data retention. This model does not have QSPI, whereas the Winbond Electronics models and ISSI models support QSPI. This model only supports dual read. This model has a maximum current usage of 15 mA. The LE25U40PCMC-AH is currently not available.

#### **3.4.4.4 External Flash Memory Summary**

The comparisons above can be summarized in the table below:

	IS25LP016D	W25Q16JV	LE25U40PCMC-AH
Manufacturer	ISSI	Winbond Electronics	onsemi
Supply Voltage	3.3 V	3.3 V	3.3 V
Memory Capacity	16 Mb	16 Mb	4 Kb
Clock Frequency	133 MHz	133 MHz	33 MHz
Maximum Current	25 mA	15 mA	15 mA
QSPI Support?	Yes	Yes	No
Price	\$0.57	\$0.50	N/A

To summarize, the three models that were considered were the IS25LP016D from ISSI, the W25Q16JV from Winbond Electronics, and the LE25U40PCMC-AH from onsemi. The W25Q16JV model was chosen to be used as the external flash memory storage chip to be used in our project. All three models are NOR flash memory chips, and all three models require a supply voltage of 3.3 V. So, those two things were not a determining factor in which model we used. The IS25LP016D and the W25Q16JV both have a memory capacity of 16 Mb, which is higher than the memory capacity of the LE25U40PCMC-AH. The LE25U40PCMC-AH also has a lower clock frequency compared to the IS25LP016D and the W25Q16JV, which means there was slower data transfer. The LE25U40PCMC-AH also does not have QSPI support; it only supports up to dual read. Therefore, the options were narrowed down to either the IS25LP016D and the W25Q16JV. The W25Q16JV was chosen over the IS25LP016D because although both options were nearly identical, the W25Q16JV uses slightly less power and the W25Q16JV has greater availability online currently than the IS26LP016D. The LE25U40PCMC-AH flash memory storage did not have any availability online currently.

### **3.4.5 Motor Driver IC**

In order to control the motors on the remote control car, we needed a motor driver IC to control the motor direction and to supply current to the motors. Ideally, we will want a dual motor driver IC to control the left motors and the right motors separately, so that we will have that option if needed. The four motor driver ICs that we discussed and comparing are: the DRV8835DSSR, the TB6612FNG, the STSPIN948, and the MTS2916A-LGC1.

#### **3.4.5.1 DRV8835DSSR**

The DRV8835DSSR IC is a half-bridge dual motor, low-voltage motor driver created by Texas Instruments. The DRV8835DSSR can use a supply voltage between 0 V and 11 V to power the motors and can use a supply voltage between 2 V and 7 V to power the logic. The DRV8835DSSR has two different modes to control the motors: IN/IN and PHASE/ENABLE. In IN/IN mode, there are four different functions: coast, reverse, forward, and brake. In PHASE/ENABLE mode, there are only three different functions: reverse, forward, and brake. In IN/IN mode, you send bits over the IN1 and IN2 pins to select a function. Each function has a corresponding 2-bit value. In PHASE/ENABLE mode, you use IN1 as the PHASE pin and you use IN2 as the ENABLE pin. The ENABLE pin determines whether the motor is moving or not and the PHASE pin sets the direction of the motor. So, if the ENABLE pin is set to LOW, the motor brakes. If the ENABLE pin is set to HIGH, then the motor either moves forward or in reverse. A value of HIGH for the PHASE pin will set the motor to reverse and a value of LOW for the PHASE pin will set the motor to forward. The DRV8835DSSR can provide an output current of up to 1.5 A to the DC motors. The DRV8835DSSR is currently available at a cost of \$1.58 on the Mouser Electronics store.

#### **3.4.5.2 TB6612FNG**

The TB6612FNG IC is a dual motor driver created by Toshiba. The TB6612FNG can use a supply voltage up to 15 V to power the motors and can use up to 6 V to power the logic. The TB6612FNG uses two input signals, IN1 and IN2, to choose four different modes: clockwise, counter-clockwise, short brake, and stop mode. There are also two other input signals: PWM and STBY. The PWM signal determines whether the DC motors move in the set direction by IN1 and IN2 or if the DC motors brake. The STBY signal determines whether the DC motors enter standby mode. The TB6612FNG can provide an output current of up to 1.2 A to the DC motors. The TB6612FNG is currently available at a cost of \$1.97 on the Mouser Electronics store.

#### **3.4.5.3 STSPIN948**

The STSPIN948 is a full-bridge dual motor driver created by STMicroelectronics. The STSPIN948 can use a supply voltage up to 58 V to power the motors and can use up to 4 V to power the logic. The STSPIN948 can provide an output current of up to 4.5 A to the DC motors, which is much more current than the previous two motor drivers could provide to the DC motors. The STSPIN948 is currently available at a cost of \$5.02 on the Mouser Electronics store.



#### **3.4.5.4 MTS2916A-LGC1**

The MTS2916A is a full-bridge dual motor driver created by Microchip Technology. The MTS2916A can use a supply voltage of up to 40 V to power the motors and can use 5 V to power the logic. The MTS2916A can provide an output current of up to 750 mA to the DC motors, which is closer in range to the output currents provided by the DRV8835DSSR and the TB6612FNG. The MTS2916A is currently available at a cost of \$1.29 on the Mouser Electronics store.

#### **3.4.5.5 Motor Driver IC Summary**

The comparisons of the motor driver ICs can be summarized in the table below:

	DRV8835DSSR	TB6612FNG	STSPIN948	MTS2916A
Manufacturer	Texas Instruments	Toshiba	STMicroelectronics	Microchip Technology
Max. Motor Voltage Input	11 V	15 V	58 V	40 V
Logic Voltage	3.3 V/5 V	3.3 V/5 V	3.3 V	5 V
Max. Output Current	1.5 A	1.2 A	4.5 A	0.75 A
Cost	\$1.58	\$1.97	\$5.02	\$1.29

The STSPIN948 is too powerful for our purpose. The maximum motor voltage input is much higher than the maximum motor voltage inputs for the other three motor driver ICs and the maximum output current is also much higher than the other three motor driver ICs. In addition, the cost of the STSPIN948 is higher than the other three motor driver ICs. The DRV8835DSSR, the TB6612FNG, and the MTS2916A have similar prices. They also have similar maximum output currents. However, the MTS2916A has a much higher maximum motor voltage input of 40 V, which is not necessary for our project. So, the decision comes down to either the DRV8835DSSR or the TB6612FNG. Both of these motor driver ICs can operate at 3.3 V or 5 V, so that is not a deciding factor in choosing between the two. The battery that we used to power the remote control car also supplies a voltage of 7.4 V, which both of these motor driver ICs can use because their maximum motor voltage inputs are above that amount. The DRV8835DSSR has a slightly higher maximum output current than the TB6612FNG. The DRV8835DSSR also has a much simpler interface for controlling the motors. Therefore, we will use the DRV8835DSSR because its control interface is simple and easy to work with, it can use a voltage of 7.4 V from our battery, and is not too powerful for our DC motors.

#### **3.4.6 3.3 V Output DC/DC Converter**

For our remote control car PCB, we needed a DC/DC converter to take our 7.4 V input from the battery and output 3.3 V to power the microcontroller, the motor driver IC, and

the IR receiver. The recommended power supply current for the ESP32 is 500 mA, so in order to ensure all of our peripherals are receiving enough power, we must use a DC/DC converter that can provide at least 1 A of current. We will also be using a non-isolated DC/DC converter because isolated converters tend to be more costly and it is not necessary for the battery to be isolated from the rest of the circuit for our project. Also, non-isolated DC/DC converters tend to have a higher efficiency. The four 3.3 V output DC/DC converters that was discussed are: the V78E03-1000-SMT-TR, the R-78AA3.3-1.0SMD, the TSR 1-2433SM, and the EPM78V2-03R3-01R0R.

#### **3.4.6.1 V78E03-1000-SMT-TR**

The V78E03-1000-SMT-TR is a non-isolated DC switching regulator created by CUI Inc. It takes an input voltage between 6.5 V and 36 V and it has an output voltage of 3.3 V. It has a maximum output current of 1 A and can output a maximum of 3.3 W of power. It has a typical efficiency of 90%. The V78E03-1000-SMT-TR also has an adjustable output voltage and is available in a 10-pin package. The V78E03-1000-SMT-TR is currently available starting from \$4.92 on the Mouser Electronics store.

#### **3.4.6.2 R-78AA3.3-1.0SMD**

The R-78AA3.3-1.0SMD is an adjustable output non-isolated buck converter created by RECOM Power. It takes an input voltage between 4.75 V and 18 V and it has an output voltage of 3.3 V. The output voltage can be adjusted to between 3 V and 5.5 V. It has a maximum output current of 1 A and can output a maximum of 3.3 W of power. It has an efficiency between 84% and 90%. Like the V78E03-1000-SMT-TR, the R-78AA3.3-1.0SMD also has an adjustable output voltage and is also available in a 10-pin package. The R-78AA3.3-1.0SMD is currently available starting from \$10.13 on the Mouser Electronics store.

#### **3.4.6.3 TSR 1-2433SM**

The TSR 1-2433SM is a non-isolated step-down switching regulator created by TRACO Power. It takes an input voltage between 4.75 V and 36 V and it has an output voltage of 3.3 V. It has a maximum output current of 1 A and can output a maximum of 3.3 W of power. It has a typical efficiency of 91%. The TSR 1-2433SM is available in a 6-pin package. Also, the TSR 1-2433SM has internal bypass capacitors, so it is not required to have external bypass capacitors. The TSR 1-2433SM is currently available starting from \$9.96 on the Mouser Electronics store.

#### **3.4.6.4 EPM78V2-03R3-01R0R**

The EPM78V2-03R3-01R0R is a non-isolated switching regulator created by Eaton. It takes an input voltage between 4.75 V and 32 V and it has an output voltage of 3.3 V. It has a maximum output current of 1 A and can output a maximum of 3.3 W of power. It has a typical efficiency of 90%. The EPM78V2-03R3-01R0R is available in a 3-pin package. The EPM78V2-03R3-01R0R is currently available starting from \$5.24 on the Mouser Electronics store.

#### **3.4.6.5 3.3 V Output DC/DC Converter Conclusion**

The comparisons above can be summarized in the table below:

	V78E03-1000-SMT-TR	R-78AA3.3-1.0SMD	TSR 1-2433SM	EPM78V2-03R3-01R0R
Manufacturer	CUI Inc.	RECOM Power	TRACO Power	Eaton
Min. Input Voltage	6.5 V	4.75 V	4.75 V	4.75 V
Max. Input Voltage	36 V	18 V	36 V	32 V
Efficiency	90%	84%	91%	90%
Pins	10	10	6	3
Cost	\$4.92	\$10.13	\$9.96	\$5.24

All four of the 3V output DC/DC converters considered have similar minimum input voltages. The V78E03-1000-SMT-TR has a minimum input voltage of 6.5 V and the rest have a minimum input voltage of 4.75 V. The R-78AA3.3-1.0SMD has a lower maximum input voltage of 18 V and the other three DC/DC converters have similar maximum input voltages. The input voltage range on all of the DC/DC converters considered can all take an input of 7.4 V, which is the voltage of the battery that we used, so that was not a deciding factor. All four DC/DC converters considered also have a similar efficiency. The V78E03-1000-SMT-TR and the R-78AA3.3-1.0SMD both have adjustable voltage outputs and are both in a 10-pin package. The TSR 1-2433SM and the EPM78V2-03R3-01R0R do not have an adjustable output voltage and are in a more simplistic package with fewer pins. Ultimately, the decision is between the TSR 1-2433SM and the EPM78V2-03R3-01R0R because they do not have an adjustable output voltage and that is not required for our project. Also, they are in a more simplistic package with less pins. Therefore, we used the TSR 1-2433SM because it is a more simplistic package, it does not require external bypass capacitors, and it has the highest efficiency.

### **3.4.7 5 V Output DC/DC Converter**

For our remote control car PCB, we needed a DC/DC converter to take our 7.4 V input from the battery and output 5 V to power the ultrasonic sensor. Since the ultrasonic sensor was the only component that operates at 5 V, the output current required will not need to be as high as the output current required for the 3.3 V output DC/DC converter. We will also be using a non-isolated DC/DC converter because isolated converters tend to be more costly and it is not necessary for the battery to be isolated from the rest of the circuit for our project. Also, non-isolated DC/DC converters tend to have a higher efficiency. The three 5 V output DC/DC converters that we discussed are: the TSR 1-2450SM, the VXO7805-500-M-TR, and the R-78AA5.0-0.5SMD-R.

#### **3.4.7.1 TSR 1-2450SM**

The TSR 1-2450SM is a non-isolated step-down switching regulator created by TRACO Power. It is the same series as the TSR 1-2433SM discussed in the previous section, however, it provides an output voltage of 5V. It takes an input voltage between 6.5 V and 36 V and provides a maximum output current of 1 A. It also supplies a maximum power of 5 W. It has a typical efficiency of 94%. Like the TSR 1-2433SM, it has internal bypass capacitors, so no external bypass capacitors are required. The TSR 1-2450SM is currently available on the Mouser Electronics store at a cost of \$9.96.

#### **3.4.7.2 VXO7805-500-M-TR**

The VXO7805-500-M-TR is a non-isolated switching regulator created by CUI Inc. It takes an input voltage between 6.5 V and 36 V and provides an output voltage of 5 V. It provides a maximum output current of 0.5 A and supplies a maximum power of 2.5 W. It supplies less current than the TSR 1-2450SM. It has a typical efficiency of 90%. The VXO7805-500-M-TR is currently available at a cost of \$2.60 on the Mouser Electronics store.

#### **3.4.7.3 R-78AA5.0-0.5SMD-R**

The R-78AA5.0-0.5SMD-R is an adjustable output non-isolated buck converter created by RECOM Power. It is similar to the R-78AA3.3-1.0SMD discussed in the previous section, however, the R-78AA5.0-0.5SMD-R provides an output voltage of 5 V and a maximum output current of 0.5 A. It can take an input voltage between 6.5 V and 32 V. The R-78AA5.0-0.5SMD-R provides the same current as the VXO7805-500-M-TR. Like the VXO7805-500-M-TR, the R-78AA5.0-0.5SMD-R also supplies a maximum power of 2.5 W. The R-78AA5.0-0.5SMD-R has an efficiency between 86% and 94%. The R-78AA5.0-0.5SMD-R is currently available at a cost of \$9.35 on the Mouser Electronics store.

#### **3.4.7.4 5 V Output DC/DC Converter Conclusion**

The comparisons discussed above can be summarized in the table below:

	TSR 1-2450SM	VXO7805-500-M-TR	R-78AA5.0-0.5SMD-R
Manufacturer	TRACO Power	CUI Inc.	RECOM Power
Min. Input Voltage	6.5 V	6.5 V	6.5 V
Max. Input Voltage	36 V	36 V	32 V
Efficiency	94%	90%	86%
Cost	\$9.96	\$2.60	\$9.35

The three 5 V output DC/DC converters discussed all have minimum input voltages of 6.5 V and have similar maximum input voltages. All three of these DC/DC converters

can take an input of 7.4 V, which is the voltage provided by the battery. All three of these DC/DC converters also have similar efficiencies. The VXO7805-500-M-TR is the cheaper option out of the three, however, the SMD package is large. The R-78AA5.0-0.5SMD-R also has an adjustable voltage, which is not needed for this project. Therefore, the TSR 1-2450SM is the most ideal option out of the three because it does not have an adjustable voltage, it is the most efficient out of the three options, and it does not require external bypass capacitors.

### **3.4.8 Step-Up Converter**

For the remote controller PCB, the power will come from two AA batteries for a total voltage of 1.5 V. However, that voltage will decrease to about 1.2 V as the batteries deplete charge. Therefore, the voltage supplied by the two AA batteries can range anywhere from 2.4 V to 3 V. All components on the PCB require a supply voltage of 3.3 V. So, in order to ensure that there is a constant supply voltage of 3.3V, a step-up converter must be used to step-up the voltage to 3.3 V. In order to ensure there was enough current supplied to all of the components on the PCB, the step-up converter should be able to supply at least 500 mA of current. The three step-up converters that was discussed are all created by Monolithic Power Systems. The three step-up converters that was discussed are: the MP3120DJ-LF-Z, the MP3414DJ-LF-Z, and the MP3221GJ-P.

#### **3.4.8.1 MP3120DJ-LF-Z**

The MP3120DJ-LF-Z is a high efficiency, synchronous step-up converter created by Monolithic Power Systems. It has a 1.1 MHz switching frequency and has up to 96% efficiency. It can take an input voltage between 0.8 V and 5 V and can output a voltage between 2.5 V and 5 V. It can provide up to 1.2 A of current. The MP3120DJ-LF-Z is currently available on the Mouser Electronics store at a cost of \$1.91.

#### **3.4.8.2 MP3414DJ-LF-Z**

The MP3120DJ-LF-Z is a high efficiency, synchronous step-up converter created by Monolithic Power Systems. It has a 1 MHz switching frequency and has up to 96% efficiency. It can take an input voltage between 0.6 V and 4V and can output a voltage between 1.8 V to 4 V. It can provide up to 1.8 A of current. The MP3414DJ-LF-Z is available on the Mouser Electronics store at a cost of \$1.87.

#### **3.4.8.3 MP3221GJ-P**

The MP3221GJ-P is a high-efficiency step-up converter created by Monolithic Power Systems. It has a 1.2 MHz switching frequency and has up to 92% efficiency. It can take an input voltage between 0.8 V and 5 V and can output a voltage between 2.5 V and 5 V. It can provide up to 2.7 A of current. The MP3221GJ-P is currently available on the Mouser Electronics store at a cost of \$2.33.

#### **3.4.8.4 Step-Up Converter Conclusion**

The comparisons discussed above can be summarized in the table shown below:

	MP3120DJ-LF-Z	MP3414DJ-LF-Z	MP3221GJ-P
--	---------------	---------------	------------

Max. Current	1.2 A	1.8 A	2.7 A
Max. Efficiency	96%	96%	92%
Cost	\$1.91	\$1.87	\$2.33

So, the MP3221GJ-P has a slightly higher cost and a slightly lower efficiency than the other two step-up converters. It also has a maximum output current of 2.7 A, which is a lot more than what is needed for our PCB. Therefore, the decision is between the MP3120DJ-LF-Z and the MP3414DJ-LF-Z. Both have the same efficiency and a similar price, however, the MP3414DJ-LF-Z has a maximum output current of 1.8 A, which is higher than the 1.2 A provided by the MP3120DJ-LF-Z. Both of these currents are sufficient for powering our circuit, so that is not a deciding factor. The MP3120DJ-LF-Z requires less resistors and less capacitors connected to it externally, therefore, reducing the cost slightly. In conclusion, the MP3120DJ-LF-Z was chosen because it has the highest efficiency, is slightly less costly than the MP3221GJ-P, and it requires less external resistors and external capacitors to function.

### **3.4.9 Kit Comparison**

It was extremely important to go into detail about how a car works and the key components/systems, because that is how we are going to replicate that in an RC car. First and foremost, the most obvious thing is where do you put all of these components to make a functioning RC car. We need a frame and in order to replicate a real car, all we need is just a flat platform that things could be screwed onto. It could be a thick piece like a plate, or a rectangular prism shape where you house components inside that container, and then have the wheels or other accessories on top or the bottom.

The engine component will not be the same as it would be for a regular car, because the regular car has to take all of the energy and put it into the transmission to make the tires spin. We do not need any of that. The RC car is much simpler and in order to replicate it, we just need a DC motor. That will need to be connected to the frame of the car and then the part of the motor that spins was where we connect a wheel. These wheels could be any type: two tires, four tires, or treads.

Normal cars have electronics inside like powertrain electronics, safety, chassis electronics, and infotainment systems. Our car is much more bare bones. We are going to have an ultrasonic sensor to measure how far we are from the things in front of us, two headlights to illuminate what is in front, and a camera to allow the user to see what the car is seeing. This is important, because you can be away from the car and still actually know what is happening in front of it. The very last feature we need to replicate is the fuel system. In a regular car you would either use gasoline or electricity. Gasoline is easy because you go to a gas station, fill up, and you are on your way. Electric cars need to use a charging station that takes much longer than gas. For our system, we used an external battery pack and this can be charged whenever the car is not in use. You plug into it using a regular 12V house outlet. That powerbank plugs in right into the PCB, and it powers on the entire system so that it'll be able to drive and be controlled, along with being able to connect to all of our other systems, like the app and controller PCB.

The very last thing is how we drive this car. We have two options: replicate self-driving cars, or allow for complete user control. Replicating self-driving cars would take away from the heart and soul of the project, since all we needed to do is write a list of commands, turn on the car, and it will execute them. We want to have it replicate a normal driving experience, we need to build handheld controls, and have those controls connected to our system as an input, and the motor, lights, and servos act as the outputs. You turn the steering wheel and then that will rotate the wheels on the axis and it'll turn whereas for our car we need to have it connect to a PCB that was a handheld controller while also connecting to a Companion app and we used a microcontroller that was able to control everything.

#### **3.4.9.1 Choosing a Component Kit**

There are many factors that need to go into choosing the kit that we will use for our car. We could go online and purchase every single part we need individually, and then put it all together as one unit. and then design our car and it should work perfectly. We could also go online and find a kit that already comes with a bunch of components. They might not be exactly what we needed, so we still will have to go out and purchase additional pieces.

The main benefit to buying the parts individually is that we can have complete control over the design. This includes the size, shape, power, and additional features. We could orient our budget to focus on the most important things. We want to have an amazing driving experience, so we could spend most of the budget on the motors, joysticks, and the camera module, while worrying less about the frame or power of the headlights. A few drawbacks are the time it would take to find and purchase each component, having to buy multiple of the same parts, and ordering the wrong things. If we need a specific voltage regulator to make our power bank work with our PCB, that component might only be \$0.15, but we have to buy it in a pack of 10 and pay for shipping. That seems small, but ordering dozens of different components can make those numbers skyrocket. Buying the wrong this is also a huge factor. If we misread a datasheet, click on the wrong part, or not realize that these expensive motors need a specific type of motor driver. All of that adds time and money to the process, and if we are missing an add-on component and that is causing a whole system to not work, it could be hard to diagnose the problem.

A kit with a bunch of components might be a little bit more expensive upfront, but it's value is in the significantly more time you save. You can also guarantee that the systems will work together as one until, because that is the product you are buying. We would know that the power is safely regulated, and that we will not be wasting our money on duplicates of pieces.

#### **3.4.9.2 Metal Smart Robotic RC Tank Chassis Kit**

The very first kit that we looked at is the metal smart robotic RC tank chassis kit. It is the most basic kit that we could find, while still giving us a great frame to work off. The massive benefit to this kit is that the price is only \$25.99. It is the cheapest kit that we will discuss and it allows for complete creativity and design choices. Although it is subjective, tanks look really awesome. We are trying to build an RC car, and this is technically a tank, but it has all the same features needed.

This is the most basic shit that we can find, as it gives us a frame to build everything on top of. It includes four DC motors and two large tank-style tire treads. The tank treads are made up of engineering plastic, and are turned with four plastic wheels that are connected to the DC motors. When everything is assembled, it is a platform that we can add on all the other components that we needed to purchase, since there is not anything else included. The benefits and drawbacks to that were discussed in the previous section. The TTC geared motor has an input voltage of 3V to 6V, a reduction ratio of 1-48, and a torsion of 0.8 kg. They are each 2 cm x 2.5 cm x 7 cm large, and at 6V, it has no load of less than or equal to 250 mA, and its load speed is 240 +/- 10% RPM. At 3 V, it has no load of less than or equal to 170 mA, and its load speed is 110 +/- 10% RPM.

The tank-style tire treads are made up of engineering plastic. They are both 94.5 cm long, and adjusted length to width is only 4.5 cm, so it is really thin, but long enough that it will wrap around the plastic “wheels” and drive. Plastic is not the strongest material, but they are calling it engineering plastic, so you have to assume it will hold up better than normal. With that being said, tank treads work by having a bunch of individual pieces get locked into a chain, and they grip to the ground separately. The treads will eventually fade and wear down, and they will lose traction. In order to replace them, we would have to get the exact same size tread, but this product is from a small seller, so finding just the right tread replacement was difficult. It does not allow for long term ease of use, whereas tires are all really replaceable. Unless the height of the car off the ground is important, we would just need to find something with the same inner size as the end of the motors, and place them. An underrated positive is that we will not have to worry about things getting underneath the tires. It'll just be able to go over certain objects where in a normal car having four wheels just means more.

It also includes M4 screws to connect the treads to the plastic tires on the car. They are 54.5 mm in diameter, height 35 mm and have a coupling of 45 and 6 mm each, depending on the type of motor that we use in the car. We would use the 4 mm for a 25 mm motor, the 5 mm for a 33 mm motor, and we use the 6 mm for a 37 mm motor. The actual chassis that the entire kit will sit on is made of metal and it is 13.49cm x 7.49cm x 3.5cm in measurements. It is sturdy and robust, since it is made out of lightweight metal. The full product is 6.42 x 2.36 x 7.6“ and it weighs 1.14 pounds.

The main benefit to buying the parts individually is that we can have complete control over the design. We could orient our budget to focus on having an amazing driving experience, while worrying less about the frame or power of the headlights. A few drawbacks are the time it would take to find and purchase each component, having to buy multiple of the same parts, and ordering the wrong things. All of that adds time and money to the process, and if we are missing an add-on component, and that is causing a whole system to not work, it could be hard to diagnose the problem.



### **3.4.9.3 ELEGOO Conqueror Robot Tank Kit**

The second kit we looked at is the ELEGOO Conqueror Robot Tank Kit. It is completely on the opposite end of the spectrum for starters. It is \$120.99 compared to our previous kit which was only \$25.99, and this one is also a tank base design as well.

Due to a global chip shortage, they are no longer using the DRV8835 motor driver chip or the HT42B534 USB to serial chip. It includes the TB6612 motor driver chip and CH340 USB to serial chip. The motor driver chip is already pre-soldered onto the shield that plugs perfectly the Arduino Uno that comes with the kit. Once it's in place, you don't have to worry about anything other than plugging in the components into their respected female header pins. The USB to serial chip is already pre-soldered onto the Arduino copy that they give you.

The Conquer Robot Tank Kit is a tank and it comes with an already programmed Arduino copycat. The dimensions are 8.58 x 8.78 x 7.4" and it weighs 3.4 pounds. This product comes with four metal plates that come together to make a rectangular prism design, and that will act as the frame of the car. You can either put all of the components inside of this frame or attach everything onto the top. It depends on whether or not you would prefer additional access to some of the components or whether you would rather just put it all together at once and it be protected. Attaching all of the components to the top will allow for access all the time, so you can make adjustments and improvements. The downside to this decision is that the entire system is exposed to the elements. Although this is a Senior Design project, we were careful not to break anything, having all of the electronics out and about leaves the possibility out there. If we wanted to assemble the components and mount them inside the frame, it would make it much harder to access and upgrade, as we would have to take apart the tank every single time, but it would give the components a decent level of security. The frame is just like a metal box with tank-treads on the side, so if there is rain, rocks flying, or if someone were to step on it, it should protect the brain of the RC car.

The Arduino Uno is this company's specific brand, called the Elegoo Uno. Arduino is an open source platform, so they copied the layout, and used the parts with the best cost to performance ratio. The replica uses a MPU6050 shield that goes into the Uno is called "SmartCar-Shield-V1.1". It is their own shield that they designed to control some of the other components in this kit. The most important thing is that it comes with an infrared LED so the remote also comes with it and can send direct signals to the car. The Uno is already pre programmed and ready to drive using that hand held controller's infrared LED transmitter, or the companion app and a wifi connection. It also has an on and off switch that will completely turn the system on and off. It has an LED to show whether or not the battery pack is connected and if the power is being supplied to the circuit. It has a reset button that will help in case of any bugs or glitches. Trying to turn it on and off is always the first step in troubleshooting. The battery pack that comes with the system was plugged into female header pins on the shield. It also has a pin for the ultrasonic sensor that is four pins. It has a four-pin header for the line tracker component to be plugged in, and it has four-pin female headers for both the UART and the ultrasonic sensor. It has four two-pin female headers for the servos, with them being labeled M1 for the front left tire, M2 for the front right tire, M3 for the back left tire, and M4 for the back right tire.

Once you plug in the Arduino and plug the shield into that, what actually goes into the shield are two servos, two high-powered DC motors with an entire suspension system to make the tank tires operate, ton of screws and bolts, along with an ultrasonic sensor, a photoresistor for the line tracking module, a camera module, and an infrared remote controller. If you assembled everything to this car's capabilities, it would have an ultrasonic sensor in the front to measure how far away it is from incoming objects and a camera module to record what is happening in front.

The awesome thing about this kit is the companion app comes with a way to see what the camera module is seeing, and it has a “joystick” to steer the car. All you have to do is turn on the car, go to your phone's WiFi services, and connect to the ELEGOO network. Once there, you can open the app and it will connect to the car. This will work extremely similar to ours, but we designed ours from scratch, so it was different. The main thing that their app has helped us discover is how important the joystick to DC motor connection is. There will need to be a formula that will change the speed of some or all of the motors, based on the position of a joystick. Their app has a joystick that covers the X and Y axis. That is great for saving screen space, but due to its small size, it causes a lack of control and the car will move sporadically. This is shown in our testing phase of the components. Again, we will not be using the app, but it is an amazing frame of reference, as we are trying to accomplish a similar goal.

It has two servos, one servo is for controlling that camera going up and down, and the other for the camera going left to right. You do not want to just have the view directly in front of you, so you want to be able to see all around you. The servo will rotate 180°, giving you a nice range of motion anywhere you will go. The car also has two DC motors. These are significantly more powerful than the previous kit's. For starters, this is because the kit is much larger. It needs to be much more powerful because the kid is larger, and it has to generate enough torque to actually move the wheels. Not only would the motors need to be stronger in general, but the kit only includes two, so one of these motors is significantly stronger than one of the motors in the Metal Smart Robotic RC Tank Chassis Kit. The system also includes suspension, along with actual tires. This is a very heavy duty tank design, and in order to rotate everything properly, it needs a ton of torque.

It also comes with a photoresistor line tracking module, which will go on the bottom to maximize its line tracking capabilities. If you were to go into a room with a white flooring and then tape a black line on the ground, it would follow that black line. This is a really cool feature, but we are not going to include that in our final design. It would be a “wasted” part to this kit, and while that is a negative, it does free up additional space on their shield for testing. We could plug our headlights right into the same place, and make sure those work as well. It is an additional component that we don't need, but this kit will have a ton of features that we do want. We want everything except for the photoresistor and line tracking system.

The materials list are six shock absorbers, eight wheel cap pieces, two coupling pieces, eight driven wheels, two wheels (the tank treads), one ultrasonic sensor module, one ultrasonic bracket, and two SG 90s (servos). The suspension is made up of eight different rockers added together on the outside of the car where the tank trends are, and this is the rigid end support for the entire vehicle. It also comes with two larger DC motors modules, one bracket extension shield, GY521 module (accelerator), one line tracking module, and two arms for the servos. It comes with a bracket base, a bracket top, and a left and right side of the bracket. It also includes a micro USB cable and a type B USB cable. Arduino Uno copycat a cell box that includes lithium batteries inside and includes. It includes the base plate, top plate, left side, and right side, along with a remote control that you can use to drive the car if you do not have access to the Companion app. The camera module uses an ESP 32S3WROOM1. This has a ESP32 S3 chip as the microcontroller. It also has Wi-Fi connectivity. It has integrated flash memory, external flash, antenna, powerful management, circuitry, and peripherals such as SPI, I2C, UART, ADC, DAC, PWM, and more. This component is then turned into the ESP 32S3 camera version 1.0. This allows you to record with a camera that is on one face of the PCB. It has two options, one being upload and the other being camera. When it is on camera, the module is recording what is in front of it, and then when you click upload it, then takes the recorded data and saves it onto the Arduino. You can then access that information by plugging the Arduino back into your laptop. That is why there needs to be an ESP, because it has Wi-Fi connected into it so you go onto your smartphone.

One of the more valuable modules that is included with this kit is the GY – 521 module. Its main purpose is to record motion sensing and control. This includes the MPU – 6050 sensor. It is a motion tracking device that combines a three-axis gyroscope and a three-axis accelerometer. The gyroscope measures the rate of rotation or angular velocity around the three axes. The accelerometer measures acceleration along the three axes. It will also provide the module's linear acceleration, which takes into consideration the force of gravity. It even includes a voltage regulator to ensure stable operation voltages along with pull up resistors interface pins, and an onboard logic shifter in an RC car. It will tell you how fast the car is accelerating, which can give information about the terrain. A common use for RC cars is to monitor the traction and adjust the power sent to those specific tires. That feature is of no use in this design, because it has tank treads, instead of a traditional four wheels. It also has jester control. This would be for a handheld controller, and by tilting the controller, the gyroscope will sensor the motion and steer the car in that direction.

Although the ELEGOO Conqueror Robot Tank Kit is an amazing kit with a ton of awesome features, it does have some flaws. The first thing is the price. It is the most expensive by a wide margin, and we are not even using all of the components in it. On top of that, we cannot forget that we are designing a hand held PCB controller, so we cannot spend our budget on the RC car, and have nothing left over for that. This does not include all the things that we needed for our handheld PCB, which requires two joysticks, a potentiometer, an infrared LED transmitter, a UART header pins, along with an on and off switch, and a button to control the headlights. Additionally, it does not come with the ESP module that was used to program the car either. There are no headlights in this kit, so although it comes with a ton of features, it is a pretty expensive kit. It does look really

awesome because tanks look cool, but at that price, it would be nice to not have to waste any materials such as the line tracking system and to have everything we need and we're missing out on all the components for our handheld PCB along with our headlights. A positive to this kit is how large the frame is, as it allows plenty of room for expansion, and installation of components. We could just add a piece of command strip and connect the headlights on both sides.

#### **3.4.9.4 ELEGOO UNO R3 Smart Robot Car Kit V4**

The last kit we looked at is the ELEGOO UNO R3 Smart Robot Car Kit V4. It is a perfect blend of the previous kits, coming in at \$79.99. The overall design is different, as it is a car shape instead of a tank design, so it is what we were aiming for in the brainstorming process.

Due to a global chip shortage, they are no longer using the DRV8835 motor driver chip or the HT42B534 USB to serial chip. It includes the TB6612 motor driver chip and CH340 USB to serial chip. The motor driver chip is already pre-soldered onto the shield that plugs perfectly the Arduino Uno that comes with the kit. Once it's in place, you don't have to worry about anything other than plugging in the components into their respected female header pins. The USB to serial chip is already pre-soldered onto the Arduino copy that they give you.

The R3 Smart Robot Car Kit V4 comes with an already programmed Arduino copycat. The dimensions are only 8 x 7 x 6" so it is much more compact than the previous design and it only weighs 2.3 pounds. This product comes with two plastic plates that come together to make a rectangular prism design, and that will act as the frame of the car. The plates are held together using copper braces, and the outside faces of the prism are open, unlike the Conquer Robot Tank Kit. You can either put all of the components inside of this frame or attach everything onto the top. It depends on whether or not you would prefer additional access to some of the components or whether you would rather just put it all together at once and it be protected. Attaching all of the components to the top will allow for access all the time, so you can make adjustments and improvements. The downside to this decision is that the entire system is exposed to the elements. Although this is a Senior Design project, we were careful not to break anything, having all of the electronics out and about leaves the possibility out there. If we wanted to assemble the components and mount them inside the frame, it would make it much harder to access and upgrade, as we would have to take apart the tank every single time, but it would give the components a decent level of security.

The Arduino Uno is this company's specific brand, called the Elegoo Uno. Arduino is an open source platform, so they copied the layout, and used the parts with the best cost to performance ratio. The replica uses a MPU6050 shield that goes into the Uno is called "SmartCar-Shield-V1.1". It is their own shield that they designed to control some of the other components in this kit. The most important thing is that it comes with an infrared LED so the remote also comes with it and can send direct signals to the car. The Uno is already pre programmed and ready to drive using that hand held controller's infrared LED transmitter, or the companion app and a wifi connection. It also has an on and off

switch that will completely turn the system on and off. It has an LED to show whether or not the battery pack is connected and if the power is being supplied to the circuit. It has a reset button that will help in case of any bugs or glitches. Trying to turn it on and off is always the first step in troubleshooting. The battery pack that comes with the system was plugged into female header pins on the shield. It also has a pin for the ultrasonic sensor that is four pins. It has a four-pin header for the line tracker component to be plugged in, and it has four-pin female headers for both the UART and the ultrasonic sensor. It has four two-pin female headers for the servos, with them being labeled M1 for the front left tire, M2 for the front right tire, M3 for the back left tire, and M4 for the back right tire.

Once you plug in the Arduino and plug the shield into that, what actually goes into the shield are one servo, four DC motors, along with an ultrasonic sensor, a photoresistor for the line tracking module, a camera module, and an infrared remote controller. If you assembled everything to this car's capabilities, it would have an ultrasonic sensor in the front to measure how far away it is from incoming objects and a camera module to record what is happening in front.

The awesome thing about this kit is the companion app comes with a way to see what the camera module is seeing, and it has a “joystick” to steer the car. All you have to do is turn on the car, go to your phone's WiFi services, and connect to the ELEGOO network. Once there, you can open the app and it will connect to the car. This will work extremely similar to ours, but we designed ours from scratch, so it was different. The main thing that their app has helped us discover is how important the joystick to DC motor connection is. There will need to be a formula that will change the speed of some or all of the motors, based on the position of a joystick. Their app has a joystick that covers the X and Y axis. That is great for saving screen space, but due to its small size, it causes a lack of control and the car will move sporadically. This is shown in our testing phase of the components. Again, we will not be using the app, but it is an amazing frame of reference, as we are trying to accomplish a similar goal.

It has one servo for the camera going left to right. You do not want to just have the view directly in front of you, so you want to be able to see all around you. The servo will rotate 180°, giving you a nice range of motion anywhere you will go. The car also has four DC motors. These are less powerful than the Conquer Robot Tank, and about the same as the Robotic RC Tank Chassis Kit. It has a similar shape to the Tank Chassis Kit, so they both use four small DC motors.

It also comes with a photoresistor line tracking module, which will go on the bottom to maximize its line tracking capabilities. If you were to go into a room with a white flooring and then tape a black line on the ground, it would follow that black line. This is a really cool feature, but we are not going to include that in our final design. It would be a “wasted” part to this kit, and while that is a negative, it does free up additional space on their shield for testing. We could plug our headlights right into the same place, and make sure those work as well. It is an additional component that we don't need, but this kit will have a ton of features that we do want. We want everything except for the photoresistor and line tracking system.

The materials list are four wheels, one ultrasonic sensor module, one ultrasonic bracket, and one SG 90 (servo). It also comes with four DC motors modules, one bracket extension shield, a GY521 module (accelerator), one line tracking module, and two arms for the servos. It comes with a top and bottom side of the frame. It also includes a micro USB cable and a type B USB cable. Arduino Uno copycat a cell box that includes lithium batteries inside and includes. It includes a remote control that you can use to drive the car if you do not have access to the Companion app. The camera module uses an ESP 32S3WROOM1. This has a ESP32 S3 chip as the microcontroller. It also has Wi-Fi connectivity. It has integrated flash memory, external flash, antenna, powerful management, circuitry, and peripherals such as SPI, I2C, UART, ADC, DAC, PWM, and more. This component is then turned into the ESP 32S3 camera version 1.0. This allows you to record with a camera that is on one face of the PCB. It has two options, one being upload and the other being camera. When it is on camera, the module is recording what is in front of it, and then when you click upload it, then takes the recorded data and saves it onto the Arduino. You can then access that information by plugging the Arduino back into your laptop. That is why there needs to be an ESP, because it has Wi-Fi connected into it so you go onto your smartphone.

One of the more valuable modules that is included with this kit is the GY – 521 module. Its main purpose is to record motion sensing and control. This includes the MPU – 6050 sensor. It is a motion tracking device that combines a three-axis gyroscope and a three-axis accelerometer. The gyroscope measures the rate of rotation or angular velocity around the three axes. The accelerometer measures acceleration along the three axes. It will also provide the module's linear acceleration, which takes into consideration the force of gravity. It even includes a voltage regulator to ensure stable operation voltages along with pull up resistors interface pins, and an onboard logic shifter in an RC car. It will tell you how fast the car is accelerating, which can give information about the terrain. It can monitor each wheel of the RC car, and adjust the power sent to those specific tires and maximize traction. It also has jester control. This would be for a handheld controller, and by tilting the controller, the gyroscope will sensor the motion and steer the car in that direction.

Although the ELEGOO UNO R3 Smart Robot Car Kit V4 is an amazing kit with a ton of awesome features, it does have some flaws. The first thing is that we cannot forget that we are designing a hand held PCB controller, so we cannot spend our budget on the RC car, and have nothing left over for that. This does not include all the things that we needed for our handheld PCB, which requires two joysticks, a potentiometer, an infrared LED transmitter, a UART header pins, along with an on and off switch, and a button to control the headlights. Additionally, it does not come with the ESP module that was used to program the car either. There are no headlights in this kit, so although it comes with a ton of features, and is cheaper than the ELEGOO Tank, it is a pretty expensive kit. It would be nice to not have to waste any materials such as the line tracking system and to have everything we need and we're missing out on all the components for our handheld PCB along with our headlights.

#### **3.4.9.5 Final Decision: ELEGOO UNO R3 Smart Robot Car Kit V4.**

The main benefit to buying the parts individually was that we can have complete control over the design. This included the size, shape, power, and additional features. We oriented our budget to focus on the most important things. We wanted to have an amazing driving experience, so we could spend most of the budget on the motors, joysticks, and the camera module, while worrying less about the frame or power of the headlights. A few drawbacks are the time it would take to find and purchase each component, having to buy multiple of the same parts, and ordering the wrong things. If we need a specific voltage regulator to make our power bank work with our PCB, that component might only be \$0.15, but we have to buy it in a pack of 10 and pay for shipping. That seems small, but ordering dozens of different components can make those numbers skyrocket. Buying the wrong this is also a huge factor. If we misread a datasheet, click on the wrong part, or not realize that these expensive motors need a specific type of motor driver. All of that adds time and money to the process, and if we are missing an add-on component and that is causing a whole system to not work, it could be hard to diagnose the problem.

A kit with a bunch of components might be a little bit more expensive upfront, but it's value is in the significantly more time you save. You can also guarantee that the systems will work together as one until, because that is the product you are buying. We would know that the power is safely regulated, and that we will not be wasting our money on duplicates of pieces.

We decided to go with the ELEGOO UNO R3 Smart Robot Car Kit V4. This kit includes the exact things the ELEGOO Tank has, except it comes in at \$40 cheaper than it, but more expensive than the more bare bones design that we get from the Tank Chassis Kit. Since they are the same kit, just a different shape and wheel design, we are still getting all of the desired parts, while having the \$40 to purchase additional components, like investing in quality parts for our handheld PCB. Along with that, we have the headlights and potentially, any other add-ons we might include from our long term stretch goals.

### **3.5 Headlights**

Because we chose the ELEGOO UNO R3 Smart Robot Car Kit V4, we needed to find headlights that would be able to attach to it. Headlights allow a current to flow through a wire called filament, which will then illuminate. That gets focused on headlights, and it aimed forwards, so you can see what is in front of you while driving. We are going to be inside most of the time, during testing and demonstration, but headlights are very useful. The most important thing is illuminating the path in the front of the car, which will give us a much clearer picture when looking through the camera module. Another important feature is the ability to communicate. Due to a speaker and mic system being a stretch goal, this still gives us a way to relay messages to people. We could turn on and off the lights in morse code, and someone on the other end of the device could understand what we were saying.

#### **3.5.1 INJORA RC Headlights**

The INJORA RC Headlights are a pair of headlights, connected to two pins: 5V and GND. The 5V was turned on anytime there is a current, so we connected it to a digital GPIO, so when we send a high signal from the ESP, the headlights will turn on. It has an

operating voltage of 5V-7.4V, and produces light from a light emitting diode. It is also 1:10 size ratio to real car headlights. The best part about this product is the plastic hole on the bottom. All we had to do was drill small holes through the plastic frame of the car, and we were able to put a screw into that, and the plastic hole on the other side. Once tightened, it was completely secured to the car. The price is also extremely affordable coming in at only \$9.98.

### **3.6 Handheld Controller PCB parts**

A massive part of our project is the handheld PCB. This is what we are going to use to control the robot, as the app will just be used for reading diagnostics, along with seeing from the camera module. If we have enough time during the summer semester, a redesign could include controls in the app. The controller was oval shaped and it had two different joysticks, an on and off switch, a potentiometer, a IR LED transmitter, and a button used for the headlights. It will also be controlled with another ESP board and have a USB to UART header pins.

#### **3.6.1 Joystick Module**

Joysticks work by having a 180 by 180, X and Y axis grid, and it starts at the point (0, 0). As it goes more in the left direction, the joystick will see the value is now at a different point, and will move the component it is controlling accordingly. It works by having two potentiometers, one on each axis, and the more it moves on one of those potentiometers, the more the values will change accordingly. The most obvious thought was to use one joystick for our entire car. If it moves in the Y direction, the car will drive forwards or backwards, and if it moves in the X direction, it will drive left or right. The problem with this is that we want extremely precise measurements. The joystick was able to read (-44, 83), but it was very hard for a user to get exact measurements with just one movement. Giving the user separate joysticks, one for going forwards and back, and the other for going left and right, will give us a much better driving experience.

##### **3.6.1.1 THB001P**

The THB001P is a joystick module created by C&K, and available on DigiKey. It is a surface mount joystick, with a solid plastic piece sticking out. It is a through hole mounting type, and a PC pin termination style. By pushing this stick in the desired direction, it will move our car. The voltage rating in AC is 50V, and its DC rating is 5V. It has a resistance of 10KOhms for the potentiometers inside. It also has an operating temperature of -10°C to 70°C. A great benefit is the price at only \$2.73 per unit, but the main negative is that there is not a rubber piece to control the movement. This is just a plastic nub, whereas traditional joysticks have rubber circles that allow for more comfort, and more surface area for better control of movements.

##### **3.6.1.2 COM-09032**

The COM-09032 is a joystick module created by SparkFun Electronics, and available on DigiKey. It is a surface mount joystick, with a solid rubber piece sticking out. It is a through hole mounting type, and a PC pin termination style. By pushing this stick in the desired direction, it will move our car. The operating voltage is 3.3V or 5V. It has a resistance of 10KOhms for the potentiometers inside. The product description describes it



as a copycat to the PlayStation 2 controller's joystick, so although it is slightly older, that is the design we were going for. A great benefit is the price at only \$4.50 per unit, and there is a rubber piece to control the movement. This allows for more comfort, and more surface area for better control of movements. It is a little more expensive than the THB001P, but with it being closer to our original designs, it is a better deal.

### **3.6.1.3 TW01BLK13**

The TW01BLK13 is a joystick module created by APEM Inc., and available on DigiKey. It is a panel mount thumbwheel, with a solid rubber piece sticking out. It is a more efficient design for us, since it is just using one potentiometer. We could mount one in the Y axis, and the other in the X axis, so it improves the user experience. It is a panel mounting type, and wire leads termination style. The operating voltage is 5V DC. It has a resistance of 10K Ohms for the potentiometers inside. It also has an operating temperature of -40°C to 85°C. All the features of this design are so much better than the other two options, but the price is the problem. It is priced at \$159.61 per unit.

### **Decision: COM-09032**

Even though the quality of the TW01BLK13 is the highest, and the design is super user friendly, the cost to performance ratio is too high. Getting 90% of the functionality in the COM-09032, for a tenth of the price, is the best decision by far. Although the THB001P does the same job as the COM-09032, for a slightly cheaper price, the decision to not include a rubber nub to push on, will lead to a poor user experience, and makes the COM-09032 the clear choice for us.

## **3.6.2 Sliding Potentiometer**

Potentiometers are a simple voltage divider circuit. The more you move the center position, the higher you are changing both of the resistors in the component. This potential difference in voltage is given in a number 0-255, and we have to convert that value into a number we can use in electronics. By taking the 0-255 input, and multiplying it by (5/1023), we get a number that is between 0 and 5. This normally signifies on and off in electronics, but we can get a number in between, so the voltage sent out to a component will not be the maximum it can receive. This allows us to get precise measurements/movements in our outputs. In our case, this sliding potentiometer will control the servo that the camera module rests on. When driving the RC Car, we need to be able to see what is in front of us, so by being able to rotate the camera, we now have a 180 degree range. If the potentiometer is all the way to the left, then it was a value 0 on the servo, making the camera look to the left. If the potentiometer is all the way to the right, it will give the servo a value of 180, putting the camera in the right direction.

### **3.6.2.1 PSM01-081A-103B2**

The PSM01-081A-103B2 is a sliding potentiometer created by Bourns Inc., and available on DigiKey. It is a surface mount potentiometer, with a solid plastic piece sticking out. It is only 100MM in total travel length, so it was a short enough distance for the driver to quickly turn it all the way, while still being long enough to dial in a "precise" value. It is 10K Ohm resistance, has a resistance tolerance of +/- 20%. It is also a through hole mounting type. By pushing this stick in the desired direction, it will move the servo. The operating voltage is 5V - 11V, so we needed to add a step-up inverter, as the ESP only

outputs 3.3V and 5V. It also has an operating temperature of -10°C to 55°C. Although this is a high quality piece, it is \$23.35, so it is very expensive for just one part of our design.

#### **3.6.2.2 PTA2043-2010CIB104**

The PTA2043-2010CIB104 is a sliding potentiometer created by Bourns Inc., and available on DigiKey. It is a surface mount potentiometer, with a solid plastic piece sticking out. It is only 20MM in total travel length, so it was a short enough distance for the driver to quickly turn it all the way, but it was a small space to dial in a “precise” value. It is 100K Ohm resistance, has a resistance tolerance of +/- 20%. It is also a through hole mounting type. By pushing this stick in the desired direction, it will move the servo. The operating voltage is 5V - 10V, so we needed to add a step-up inverter, as the ESP only outputs 3.3V and 5V. It also has an operating temperature of -10°C to 50°C. It is also just \$1.71, so it is very inexpensive.

#### **3.6.2.3 PS15-10MB6BR10K**

The PS15-10MB6BR10K is a sliding potentiometer created by TT Electronics/BI, and available on DigiKey. It is a surface mount potentiometer, with a solid plastic piece sticking out. It is only 15MM in total travel length, so it was a short enough distance for the driver to quickly turn it all the way, but it was an extremely small space to dial in a “precise” value. It is 10K Ohm resistance, has a resistance tolerance of +/- 20%. It is also a through hole mounting type. By pushing this stick in the desired direction, it will move the servo. The operating voltage is 3V - 6V, so we will not need to add a step-up inverter, like the others. Although it is the worst in quality and ease of use, it is the cheapest, coming in at only \$0.72.

#### **Decision: PTA2043-2010CIB104**

Even though the quality of the PSM01-081A-103B2 is the highest, and the design is super user friendly, the cost to performance ratio is too high. Getting a close approximation of the functionality in the PTA2043-2010CIB104, for a thirteenth of the price, is the best decision by far. Although the PS15-10MB6BR10K does the same job as the PTA2043-2010CIB104, for a slightly cheaper price, the sliding length being smaller than an already small design, will lead to a horrible user experience. Choosing parts is all about cost to performance, and the camera module is a super important feature in this car. If we cannot dial in the placement, it is a problem. That problem is not so big that we need to spend \$22 more for one component, so that makes the PTA2043-2010CIB104 the clear choice for us.

### **3.6.3 Power Bank**

This component is very important. We wanted to be able to drive the car for as long as possible, so having a battery pack is crucial. A battery stores energy in chemical form, and releases it into the desired output. In our case, the battery was sending power to the controller, which only consists of a few components. This means we had a couple options, because the controller requires very little current. We could go the super cheap route of putting 2-3 AA batteries on the back, and change them out when the controller

dies, or purchase a rechargeable bank that is more expensive upfront, but could be the better deal in the long run.

#### **3.6.3.1 MulticoloredLED 3 AA Battery Holder**

The MulticoloredLED 3 AA Battery Holder is a AA Battery storage case. It has three inner leads, which allows the user to orient the batteries in positive, negative, and positive. It only costs \$7.99, and comes in a four pack. A couple great features is the screw on lid that completely protects the batteries from the elements. Additionally, there is a switch on the bottom of the case, which will or will not allow the current to flow. This is critical, because over time, there was some loss of charge in the batteries, but this can keep that to the minimum. That will maximize the use while operating the controller. This option means we needed to use 3 AA batteries which makes this a 4.5V battery bank. That is plenty of power to the microcontroller, and that in turn will power most of the other components. The issues lie in the components that require at least 5V. They needed a step-up inverter attached. This design had a couple of flaws: there were no holes in the case to screw it onto something, and there are no header pins at the end of the red and black wires. Normally no mounting would be okay, as we could just use command strips to the back, and stick it onto a surface. Since this is for a handheld PCB, there was not a flat surface to connect to, so we needed to drill a hole into the plastic of the case, and design additional through holes for mounting. The wire issue is not as big, but we needed to solder on header pins to make it much easier to connect to our PCB.

#### **3.6.3.2 7.4V 1500mAh Li-ion Battery**

The 7.4V 1500mAh Li-ion Battery is very different from the previous design. For starters, it costs \$13.99. With that price comes a lot of great features. The most important thing is that it is 7.4V. This means that all of our components were receiving ample power, and the very few that cannot take that much, were getting power from the 3.3V on the ESP. It also has two sets of wires: input and output. The input connects to the provided USB cable, and it can recharge the batteries. The output is a two-pin female header, that has a positive and negative wire. It makes it extremely easy to plug into our PCB and power the whole system. It does also come with an on/off switch, but still does not come with a mounting method. This being smaller does allow for more customization, so we could encase it in a 3D printer holder, and screw that into the back of the PCB.

#### **Decision: 7.4V 1500mAh Li-ion Battery**

Overall, this decision is an easy one. The 7.4V 1500mAh Li-ion Battery is the same price it would be to purchase the MulticoloredLED 3 AA Battery Holder and fill in all 4 cases with AA batteries, but it is significantly easier to use, is easier to mount, and will pay for itself. Not only will it be cheaper to recharge it than constantly replacing it with AA batteries, but it will also save us money on the step-up inverters we needed to purchase if we got the MulticoloredLED 3 AA Battery Holder. One solution to this would be to just buy a case large enough to house 4 or 5 AA batteries. This would give us the desired voltage, while remaining affordable. The issue then becomes size and weight. The whole point of this is to have a compact PCB that allows us to drive the car. If we have a battery

pack that is larger than the controller, no one would ever want to use it. For those reasons, it is the best option to go with the 7.4V 1500mAh Li-ion Battery.

### **3.6.4 On and Off Switch**

This component is extremely simple. It is a switch that blocks the current from flowing into the circuit when it is in the open position, but allows current to flow when it is in the closed position. We do not want really bad components on our project, but this is an extremely basic task, so we will find great options at super affordable prices.

#### **3.6.4.1 JS202011JCQN**

The JS202011JCQN is an on and off switch created by Bourns Inc., and available on DigiKey. It is a surface mounting style, and a J Lead termination style. It has a voltage rating of 6V DC and a current rating of 300mA DC. It is only 2mm in size, so it was very compact and out of the way. The operating temperature of -40°C to 85°C. The most important factor here is the price, and it comes in at only \$0.80.

#### **3.6.4.2 JS202011SCQN**

The JS202011SCQN is an on and off switch created by Bourns Inc., and available on DigiKey. It is a surface mounting style, and a S Lead termination style. It has a voltage rating of 6V DC and a current rating of 300mA DC. It is only 2mm in size, so it was very compact and out of the way. The operating temperature of -40°C to 85°C. The most important factor here is the price, and it comes in at only \$0.73.

#### **Decision: JS202011SCQN**

The quality of both products are identical, as they are produced by the same manufacturer. The only difference is their lead termination style, and the JS202011SCQN being \$0.07 cheaper. We will not be worrying about the termination style, because we are going to pay extra for a manufacturer to solder everything. With all that being said, it makes sense to go with the cheaper option, in the JS202011SCQN.

## **4. Standards and Design Constraints**

In this section, we will discuss the different standards involved in creating the Rock ‘N’ Rover. We will also discuss some design constraints that will affect the development of the project.

### **4.1 Standards**

Standards form the backbone of any engineering project, serving as a blueprint for safety, reliability, and efficiency across various domains. In the realm of safety, adherence to established standards ensures that equipment, processes, and structures meet rigorous criteria, minimizing risks to personnel and the environment. In software development, standards guarantee interoperability, compatibility, and security, facilitating seamless integration and reducing vulnerabilities. Electrical standards ensure that systems operate reliably, preventing hazards such as short circuits or overloads. Mechanical standards dictate design specifications and manufacturing processes, optimizing performance and longevity while mitigating failure risks. Communication standards foster interoperability among disparate systems, enabling seamless data exchange and collaboration. Documentation standards establish clear guidelines for recording project details, facilitating comprehension, maintenance, and future modifications. Finally, quality assurance standards provide a framework for consistent evaluation, verification, and validation, ensuring that products and processes meet predefined criteria for functionality and performance. Ultimately, adherence to these standards is paramount, as they collectively safeguard against potential hazards, ensure operational integrity, and uphold the overall quality of engineering endeavors.

The Rock ‘N’ Rover project underscores the critical importance of meeting all necessary standards for optimal success. In every aspect of its design and implementation, from safety protocols to software integration, electrical systems, mechanical components, communication interfaces, documentation practices, and quality assurance measures, adherence to established standards is paramount. Ensuring that the Rock N Rover project complies with these standards not only mitigates risks but also enhances performance, reliability, and efficiency. By meticulously following industry best practices and regulatory guidelines, the Rock N Rover team fosters confidence in the project's outcomes, bolstering its potential for groundbreaking achievements in exploration and discovery.

#### **4.1.1 Safety Standards**

Safety standards are an indispensable aspect of product development, crucial for ensuring occupational health and safety in various industries. These standards establish a comprehensive regulatory framework aimed at identifying, assessing, and mitigating risks and hazards associated with product development and utilization across different sectors. They entail meticulous evaluation of potential threats, strict adherence to statutory directives, and the integration of preventive measures throughout the design, manufacturing, and operational phases. Adhering to robust safety protocols not only minimizes the risk of accidents and injuries but also enhances overall workplace efficiency and productivity. By prioritizing safety standards, organizations demonstrate

their commitment to safeguarding employee well-being, fostering a culture of safety, and complying with regulatory requirements. This ensures a conducive working environment and contributes to the promotion of occupational health and safety practices across industries. Here are several key points highlighting the importance of safety standards in a project:

1. **Protection of Personnel:** Safety standards are primarily in place to protect the lives and health of all personnel involved in a project. Whether it's construction workers, engineers, technicians, or any other professionals, adherence to safety standards minimizes the risk of accidents, injuries, and fatalities in the workplace.
2. **Legal Compliance:** Most jurisdictions have stringent laws and regulations governing workplace safety. Compliance with these standards is not only a moral imperative but also a legal requirement. Failure to adhere to safety standards can lead to legal repercussions, fines, lawsuits, and even project shutdowns.
3. **Risk Mitigation:** Projects often involve various hazards, whether it's working at heights, handling heavy machinery, or dealing with hazardous materials. Safety standards help identify, assess, and mitigate these risks through measures such as personal protective equipment (PPE), safety procedures, and hazard controls.
4. **Enhanced Productivity:** Ensuring a safe working environment can contribute to increased productivity. When workers feel safe and secure, they are more focused, motivated, and efficient in their tasks. Conversely, accidents and injuries can disrupt workflow, lead to downtime, and hamper project progress.
5. **Cost Savings:** Implementing safety standards may require an initial investment in training, equipment, and safety measures. However, the long-term benefits often outweigh the costs. Fewer accidents mean lower medical expenses, reduced insurance premiums, and avoidance of costly legal battles. Moreover, avoiding project delays due to accidents can help maintain budgetary targets.
6. **Reputation and Stakeholder Confidence:** A commitment to safety reflects positively on the reputation of the project stakeholders, whether it's the project owner, contractor, or subcontractors. Demonstrating a proactive approach to safety not only attracts skilled workers but also instills confidence in clients, investors, and the public.
7. **Sustainability:** Safety standards often incorporate environmental considerations, promoting sustainable practices and minimizing the project's ecological footprint. By adhering to environmentally friendly practices, projects can mitigate negative impacts on ecosystems and communities, contributing to long-term sustainability.
8. **Continuous Improvement:** Safety standards are not static; they evolve over time based on advancements in technology, changes in regulations, and lessons learned from past incidents. Incorporating feedback, conducting safety audits, and

fostering a culture of continuous improvement ensure that safety remains a top priority throughout the project lifecycle.

Safety is of utmost importance in every phase of the Rock 'N' Rover project's development. Before delving into the intricate technical aspects, a comprehensive foundation of safety standards is conscientiously established to ensure the well-being of users, bystanders, and the integrity of the project. Rigorous adherence to industry-specific safety regulations is paramount, supplemented by the implementation of fail-safe mechanisms tailored to the project's unique requirements. Among these measures are strategically positioned emergency stop buttons, equipped to halt operations swiftly in unforeseen circumstances, and cutting-edge obstacle avoidance systems designed to navigate complex terrains with precision and reliability. Furthermore, comprehensive risk assessments are conducted, addressing potential hazards and identifying mitigation strategies to proactively manage risks throughout the project lifecycle. These meticulous safety protocols not only prioritize the protection of individuals and property but also exemplify a commitment to excellence in project execution. By integrating safety measures seamlessly into the project framework, we are fostering a culture of responsibility and accountability, elevating the project's reputation and setting a standard for excellence in safety practices within the industry.

#### **4.1.2 Electrical Standards**

Electrical standards are crucial guidelines and specifications that govern the design, installation, operation, and maintenance of electrical systems and equipment. They are established by national and international organizations such as the International Electrotechnical Commission (IEC), the National Electrical Manufacturers Association (NEMA), the Institute of Electrical and Electronics Engineers (IEEE), and others. These standards serve as benchmarks for ensuring safety, reliability, interoperability, and efficiency in electrical systems across various industries and applications. Here are some key reasons why electrical standards are important:

1. **Safety:** Perhaps the most critical aspect of electrical standards is ensuring the safety of individuals, property, and the environment. Standards provide guidelines for the proper handling of electrical equipment, wiring, and installations to prevent hazards such as electric shock, fire, and explosions. Compliance with these standards minimizes the risk of accidents and injuries.
2. **Reliability:** Electrical standards help to ensure the reliability and performance of electrical systems and equipment. By specifying design criteria, testing methods, and performance requirements, standards facilitate the development of products that meet quality benchmarks and perform consistently under various operating conditions. This reliability is particularly important in critical sectors such as healthcare, telecommunications, and manufacturing.
3. **Interoperability:** In today's interconnected world, interoperability—the ability of different systems and devices to work together seamlessly—is crucial. Electrical standards define interfaces, protocols, and communication protocols that enable

interoperability between diverse electrical components and systems. This is essential for the integration of technologies in areas such as smart grids, automation, and the Internet of Things (IoT).

4. **Efficiency:** Standards play a vital role in promoting energy efficiency and sustainability in electrical systems. They establish requirements for energy-efficient equipment, design practices, and performance metrics that help minimize energy consumption, reduce greenhouse gas emissions, and optimize resource utilization. Adhering to these standards not only lowers operational costs but also mitigates the environmental impact of electrical infrastructure.
5. **Compliance and Regulation:** Electrical standards form the basis for regulatory requirements and codes that govern electrical installations and products. Compliance with these standards is often mandated by government agencies and regulatory bodies to ensure public safety and industry best practices. Failure to meet these standards can result in legal liabilities, fines, or penalties.
6. **Global Trade and Market Access:** Harmonized electrical standards facilitate international trade by eliminating technical barriers and promoting product interoperability. Manufacturers and exporters can demonstrate compliance with recognized standards to gain access to global markets and compete on a level playing field. Standardization also simplifies procurement processes and reduces costs associated with product testing and certification.

Electrical integrity stands as a fundamental pillar in ensuring the reliable operation of the Rock 'N' Rover, particularly given its dynamic and technologically sophisticated nature. As the vehicle navigates through various terrains and executes intricate maneuvers, adherence to stringent electrical standards becomes paramount. To this end, the implementation of industry-approved wiring practices, such as those outlined by organizations like the International Electrotechnical Commission (IEC) and the National Electrical Manufacturers Association (NEMA), ensures that the wiring is capable of withstanding the rigors of off-road exploration. Meticulous selection of components boasting appropriate voltage, current, and temperature ratings is imperative to withstand the harsh environmental conditions encountered during missions. For instance, components with high ingress protection (IP) ratings are chosen to shield against dust and water ingress, while those with wide temperature ranges ensure reliability in extreme climates. Through these measures, we not only safeguard against potential electrical failures but also ensure the stability and longevity of the vehicle's electrical system.

Moreover, rigorous testing protocols are employed to validate the integrity and resilience of the electrical infrastructure under diverse operational conditions. This includes environmental testing to simulate the vehicle's exposure to temperature variations, humidity, vibration, and mechanical shock. Electrical stress testing evaluates the performance of components under high voltage and current conditions, ensuring they can handle peak loads without degradation. Furthermore, functional testing verifies the proper



functioning of sensors, actuators, communication systems, and power distribution modules under real-world scenarios.

This proactive approach not only enhances the safety and reliability of the Rock ‘N’ Rover but also underscores our unwavering commitment to excellence in engineering and adherence to industry best practices. By adhering to established electrical standards and conducting thorough testing, we mitigate risks associated with electrical failures, thereby maximizing the vehicle's operational uptime and mission success. Additionally, compliance with recognized standards facilitates interoperability with other systems and promotes seamless integration of future upgrades and enhancements. As a result, the Rock ‘N’ Rover stands poised to fulfill its mission objectives with efficiency, resilience, and unparalleled performance in the most challenging environments.

#### **4.1.3 Mechanical Standards**

Mechanical standards play a crucial role in ensuring safety, interoperability, and efficiency across various industries that involve mechanical engineering. These standards are established guidelines or specifications that define the design, construction, testing, and performance requirements for mechanical components, systems, and processes. They are developed and maintained by organizations such as the American Society of Mechanical Engineers (ASME), International Organization for Standardization (ISO), and others, often in collaboration with industry experts, engineers, researchers, and regulatory bodies. The importance of mechanical standards cannot be overstated, and here are several reasons why they are essential:

1. **Safety:** One of the primary objectives of mechanical standards is to ensure the safety of products, equipment, and processes. By adhering to established standards, manufacturers can design and produce mechanical components and systems that meet minimum safety requirements. This helps prevent accidents, injuries, and fatalities in various industries, including manufacturing, construction, transportation, and energy.
2. **Interoperability:** Mechanical standards facilitate interoperability by ensuring compatibility and interchangeability between different components, equipment, and systems. This is particularly critical in industries where various parts need to work together seamlessly, such as in automotive manufacturing or machinery production. Standardized dimensions, tolerances, and interfaces enable easier integration and assembly of mechanical parts from different manufacturers.
3. **Quality Assurance:** Compliance with mechanical standards helps ensure the quality and reliability of products and processes. Standards outline specific criteria for materials, manufacturing processes, testing procedures, and performance characteristics, providing manufacturers with clear guidelines to follow. Adhering to these standards can lead to improved product quality, reduced defects, and enhanced customer satisfaction.

4. **Regulatory Compliance:** Many industries are subject to regulatory requirements and legal obligations related to product safety, environmental protection, and public health. Mechanical standards often serve as a basis for regulatory frameworks and compliance assessments. Companies that fail to meet applicable standards may face penalties, fines, or legal liabilities, highlighting the importance of adherence to established norms.
5. **Innovation and Efficiency:** While standards provide a baseline for safety and quality, they also drive innovation and efficiency by promoting best practices and technological advancements. Engineers and designers use standards as reference points for developing new solutions, optimizing processes, and improving performance. Standardization can streamline design and manufacturing processes, reduce costs, and accelerate time-to-market for innovative products and technologies.
6. **Global Trade and Market Access:** In an increasingly interconnected world, adherence to international mechanical standards is essential for global trade and market access. Harmonized standards facilitate cross-border commerce by ensuring that products meet consistent requirements and regulations across different countries and regions. This simplifies regulatory compliance, reduces trade barriers, and fosters economic growth and competitiveness.

Ensuring the Rock 'N' Rover's toughness, reliability, and effectiveness relies heavily on strict mechanical standards. Before diving into the nitty-gritty of mechanical engineering, it's crucial to establish clear rules for choosing parts, making components, and putting them together. This careful approach guarantees that every part of the vehicle can handle its job without breaking down. By using tough, long-lasting materials known for their strength and resistance to rust, we strengthen the vehicle's structure, making it less likely to wear out too soon. We also pay close attention to how moving parts fit together, making sure they run smoothly and don't wear down quickly due to rubbing against each other. Following these precise standards not only improves the Rock 'N' Rover's performance and lifespan but also shows our commitment to creating a top-notch product in mechanical engineering.

#### **4.1.4 Software Development Standards**

Software development standards are paramount in product development, serving as the backbone of quality, reliability, and security in the realm of software engineering. These standards encompass a comprehensive set of guidelines, best practices, and methodologies that govern the design, coding, testing, deployment, and maintenance of software systems. Adhering to established software development standards ensures consistency, efficiency, and transparency throughout the development lifecycle. By following standardized processes and practices, developers can mitigate the risk of errors, bugs, and vulnerabilities, thereby enhancing the overall quality and performance of the software product. Moreover, compliance with software development standards fosters interoperability and compatibility, facilitating seamless integration with other software systems and platforms. Beyond technical considerations, adherence to software

development standards also contributes to regulatory compliance, risk management, and customer satisfaction. In essence, software development standards not only ensure the reliability and functionality of software products but also uphold industry best practices, bolster market competitiveness, and instill confidence among users and stakeholders.

1. **Interoperability:** One of the most significant advantages of adhering to software standards is the promotion of interoperability. By following established standards, different software components can communicate and work seamlessly with each other, regardless of their origin or the platforms they run on. This interoperability fosters integration, allowing disparate systems to function together harmoniously. For instance, adherence to web standards ensures that applications are accessible across different browsers and devices, enhancing the user experience.
2. **Compatibility:** Software standards ensure compatibility across different versions of software and hardware. This is particularly crucial in environments where various vendors provide products or services. For example, adherence to standards such as HTML, CSS, and JavaScript ensures that web applications function correctly across different web browsers and operating systems. Similarly, adherence to file format standards ensures that documents can be exchanged and opened across various software applications without loss of fidelity or functionality.
3. **Ease of Development:** Standards provide developers with clear guidelines and best practices, streamlining the development process. Developers can leverage existing standards and frameworks to build upon established solutions rather than reinventing the wheel. This not only accelerates development but also improves the quality and reliability of software products. Moreover, adherence to standards facilitates collaboration among developers by establishing a common language and set of conventions.
4. **Quality Assurance:** Software standards often include guidelines for testing and quality assurance. By adhering to these standards, developers can ensure that their software undergoes rigorous testing to identify and rectify defects early in the development lifecycle. This results in higher quality software products that are more reliable and resilient to failures, ultimately leading to increased customer satisfaction and trust.
5. **Future-proofing:** Standards provide a level of future-proofing by ensuring that software remains relevant and compatible with evolving technologies and industry trends. As technology advances, standards evolve to accommodate new requirements and innovations, thereby safeguarding investments in software development. Additionally, adherence to standards reduces the risk of vendor lock-in, allowing organizations to switch between different software solutions without encountering significant compatibility issues.

6. **Regulatory Compliance:** In many industries, compliance with regulatory requirements is mandatory. Software standards often align with regulatory frameworks and industry-specific guidelines, ensuring that software products meet legal and security requirements. Adhering to these standards not only mitigates the risk of non-compliance but also instills confidence among stakeholders regarding the security and integrity of the software.

Software reliability serves as a linchpin for the seamless operation and mission success of the Rock 'N' Rover project, where precision and dependability are non-negotiable. Prior to initiating software development endeavors, it is imperative to establish a robust framework of stringent standards to govern every facet of the process. This comprehensive approach encompasses not only code quality and maintainability but also extends to encompassing scalability, performance optimization, and security considerations. By meticulously adhering to industry best practices, such as adhering to coding standards and utilizing design patterns, developers can ensure the production of clean, efficient, and maintainable code. Additionally, the implementation of robust error handling and logging mechanisms serves as a proactive measure to identify and address issues promptly, minimizing disruptions to operations and facilitating rapid troubleshooting. Regular updates and maintenance activities further bolster the reliability and security of the vehicle's software systems, ensuring that it remains resilient in the face of evolving threats and operational demands. Through these concerted efforts, we affirm our commitment to delivering a software ecosystem characterized by stability, security, and unwavering performance, thereby advancing the objectives of the Rock 'N' Rover project with confidence and reliability.

#### **4.1.5 Communication Standards**

Communication standards play a crucial role in product development across various industries, ensuring compatibility, interoperability, and seamless integration between different components, systems, and devices. These standards serve as guidelines or protocols that dictate how data is transmitted, interpreted, and received between different entities, enabling effective communication and collaboration within complex ecosystems. Here's why communication standards are essential in product development:

1. **Interoperability:** In today's interconnected world, products often need to communicate with each other, regardless of their origin or manufacturer. Communication standards ensure that devices and systems from different vendors can work together seamlessly. For example, USB, HDMI, and Bluetooth are widely adopted standards that allow devices from different manufacturers to connect and communicate without compatibility issues.
2. **Efficiency:** Standardized communication protocols streamline the development process by providing clear guidelines and specifications. Developers don't need to reinvent the wheel every time they want two devices to communicate; they can simply follow existing standards. This saves time and resources, accelerating product development cycles and reducing time to market.

3. **Cost Reduction:** Developing proprietary communication protocols can be costly and time-consuming. By leveraging existing standards, companies can significantly reduce development costs and focus their resources on creating unique features and functionalities that add value to their products.
4. **Scalability:** As products evolve and new features are added, communication standards provide a scalable framework that accommodates growth and expansion. Standards like TCP/IP for networking or MQTT for IoT (Internet of Things) enable products to scale efficiently without requiring major architectural changes.
5. **Compatibility:** Consumers expect products to work together seamlessly, regardless of the brand or model. Communication standards ensure compatibility between different generations of products and facilitate backward and forward compatibility. For example, smartphones with different operating systems can communicate with each other via standard protocols like SMS, MMS, or Bluetooth.
6. **Reliability:** Established communication standards have undergone rigorous testing and validation, ensuring reliability and robustness in various scenarios. This reliability is crucial, especially in mission-critical applications such as industrial automation, healthcare, and transportation systems.
7. **Global Adoption:** Communication standards that are widely adopted across industries and regions provide a common language for developers, manufacturers, and users worldwide. This global adoption fosters innovation, interoperability, and collaboration on a large scale.
8. **Regulatory Compliance:** In many industries, adherence to specific communication standards is mandated by regulatory bodies to ensure safety, security, and compliance with industry regulations. For example, medical devices must adhere to standards such as DICOM (Digital Imaging and Communications in Medicine) to ensure interoperability and data security in healthcare environments.

In addition to facilitating communication between products, communication standards also play a vital role in enabling collaboration and coordination among engineers during the product development process. Here's how communication standards support developer collaboration:

1. **Shared Protocols and Interfaces:** Communication standards provide developers with common protocols and interfaces for exchanging information, sharing data, and collaborating on projects. Whether it's through APIs (Application Programming Interfaces), messaging formats like JSON or XML, or version control systems like Git, standardized communication methods ensure that

developers can work together effectively regardless of their geographic location or organizational boundaries.

2. **Consistent Communication Patterns:** Standardized communication patterns help establish common ground among developers, ensuring clarity, consistency, and understanding throughout the development lifecycle. By following established conventions and protocols, developers can communicate more efficiently, reducing misunderstandings and enhancing productivity.
3. **Cross-Platform Compatibility:** In a diverse development environment where teams may use different tools, programming languages, and technologies, communication standards provide a common framework for interoperability. For instance, standards like RESTful APIs enable developers to build applications that can communicate seamlessly across different platforms, devices, and programming languages.
4. **Integration with Development Tools:** Many development tools and platforms support communication standards, allowing developers to integrate various software components and streamline collaboration workflows. For example, project management tools like Jira or communication platforms like Slack often provide integrations with version control systems and issue tracking systems, enabling seamless communication and collaboration among development teams.
5. **Real-Time Collaboration:** Communication standards facilitate real-time collaboration among developers, enabling instant messaging, video conferencing, screen sharing, and collaborative editing of code and documents. Platforms like Microsoft Teams, Google Workspace, and GitHub provide features that support real-time communication and collaboration, fostering teamwork and innovation.
6. **Documentation and Knowledge Sharing:** Communication standards often come with documentation and best practices that help developers understand how to interact with each other and with external systems effectively. This documentation serves as a valuable resource for knowledge sharing, onboarding new team members, and maintaining consistency in development processes.
7. **Community Engagement and Support:** Many communication standards have vibrant communities of developers who actively contribute to their development, support, and improvement. Engaging with these communities provides developers with opportunities to learn, share experiences, and collaborate on enhancing communication standards to meet evolving needs and challenges.

Effective communication standards are vital for ensuring the smooth integration of the Rock 'N' Rover with external devices and systems, guaranteeing its proper functioning and compatibility. To achieve this, thorough attention must be given to establishing comprehensive guidelines before proceeding with the implementation of communication systems. These standards encompass various aspects beyond just selecting protocols and

transmission methods. By adopting industry-standard wireless communication protocols like Bluetooth and Wi-Fi, we establish a solid foundation for connectivity. Moreover, these communication standards extend beyond device-to-device communication and also encompass communication between engineers involved in the Rock 'N' Rover project. Through shared protocols and interfaces, developers can collaborate seamlessly, regardless of their geographic location or organizational boundaries. Consistent communication patterns ensure clarity and understanding throughout the development process, reducing misunderstandings and enhancing productivity. Additionally, incorporating encryption and authentication mechanisms helps protect the integrity and confidentiality of data exchanged among team members, creating a secure environment for collaboration. Real-time collaboration tools such as instant messaging and video conferencing further facilitate communication and coordination among engineers, enabling efficient problem-solving and decision-making. Finally, ensuring compatibility with a wide range of development tools and platforms enhances workflow efficiency and promotes integration among different software components. By adhering to these meticulously crafted communication standards, the Rock 'N' Rover project not only facilitates seamless integration with external devices but also upholds reliability, security, and adaptability, essential for its success.

#### **4.1.6 Quality Assurance Standards**

Quality assurance (QA) standards are crucial in product development to ensure that products meet predefined quality criteria and standards throughout their lifecycle. These standards encompass various processes, procedures, and guidelines aimed at maintaining consistency, reliability, and customer satisfaction. Here's an overview of quality assurance standards in product development and their importance:

##### **1. Establishing Quality Metrics:**

- Define Standards: QA standards establish benchmarks and metrics for product quality. These standards specify criteria such as performance, reliability, usability, and safety.
- Setting Objectives: They help set clear objectives for product quality, allowing teams to focus efforts on meeting or exceeding these objectives.

##### **2. Process Improvement:**

- Continuous Improvement: QA standards promote a culture of continuous improvement by identifying areas for enhancement in processes, methodologies, and tools.
- Root Cause Analysis: When defects or issues arise, QA standards facilitate thorough root cause analysis, enabling teams to address underlying problems and prevent recurrence.

##### **3. Risk Management:**

- Identifying Risks: QA standards help identify potential risks early in the product development lifecycle, allowing proactive mitigation strategies to be implemented.

- Mitigation Strategies: By implementing robust QA processes, organizations can minimize the likelihood of defects, errors, or failures, thus reducing risks associated with product development.

#### **4. Compliance and Regulation:**

- Ensuring Compliance: QA standards ensure that products adhere to relevant industry regulations, standards, and compliance requirements, mitigating legal and regulatory risks.
- Certification: Compliance with QA standards may be necessary for obtaining certifications or approvals essential for market acceptance or regulatory compliance.

#### **5. Customer Satisfaction:**

- Meeting Expectations: QA standards help deliver products that meet or exceed customer expectations in terms of quality, functionality, and reliability.
- Building Trust: Consistently high-quality products enhance customer trust and loyalty, leading to positive brand reputation and increased competitiveness.

#### **6. Cost Efficiency:**

- Preventing Rework: Implementing QA standards helps detect and rectify defects early in the development process, reducing the need for costly rework or corrective actions later.
- Optimizing Resources: By streamlining processes and focusing efforts on quality, organizations can optimize resource utilization and minimize wastage.

#### **7. Stakeholder Confidence:**

- Assurance: QA standards provide stakeholders, including customers, investors, and partners, with assurance that products are developed and delivered with a high level of quality and reliability.
- Transparency: Transparent adherence to QA standards fosters trust and confidence among stakeholders, demonstrating a commitment to delivering value and meeting expectations.

Quality assurance is fundamental to the development process of the Rock ‘N’ Rover, integrated seamlessly at every step of its creation. Before testing even begins, our team invests significant effort into establishing thorough standards, drawing from industry best practices and innovative approaches. These standards serve as precise blueprints, ensuring comprehensive validation of every aspect of the vehicle's functionality and performance. Our testing procedures are designed to scrutinize each component carefully, from the strength of its chassis to the efficiency of its control systems, leaving no aspect unchecked in our pursuit of excellence. Additionally, detailed documentation of test results and the maintenance of a comprehensive quality assurance log demonstrate our unwavering commitment to accountability and transparency. Through this attention to detail and our relentless pursuit of perfection, we continuously enhance the Rock ‘N’ Rover, pushing the boundaries of innovation and setting new benchmarks in automotive engineering.



#### **4.1.7 Documentation Standards**

Documentation standards for product development are essential guidelines and protocols that dictate how information about a product is recorded, organized, and maintained throughout its lifecycle. These standards encompass various types of documents, including requirements specifications, design documents, test plans, user manuals, and release notes. They ensure consistency, clarity, and completeness in the documentation, enabling effective communication among stakeholders and facilitating efficient product development processes. The importance of documentation standards in product development cannot be overstated, and here's why:

1. **Clarity and Understanding:** Clear and well-organized documentation helps teams understand the product's requirements, design, and functionality. It serves as a reference point for developers, designers, testers, and other stakeholders, reducing misunderstandings and ambiguities.
2. **Consistency and Standardization:** Documentation standards establish uniform formats, templates, and terminology for all project documentation. This consistency fosters better collaboration, streamlines processes, and makes it easier for team members to locate and use relevant information.
3. **Traceability and Accountability:** Proper documentation allows for tracing requirements throughout the development process. By maintaining a clear trail of decisions, changes, and dependencies, documentation standards help ensure accountability and enable effective change management.
4. **Risk Management:** Comprehensive documentation mitigates risks associated with product development by providing a comprehensive overview of project scope, goals, constraints, and dependencies. It helps identify potential issues early in the development cycle, allowing teams to address them proactively.
5. **Onboarding and Knowledge Transfer:** Well-documented projects facilitate the onboarding of new team members by providing them with a structured overview of the product and its development history. Additionally, thorough documentation aids in knowledge transfer between team members, ensuring that institutional knowledge is preserved and shared effectively.
6. **Regulatory Compliance and Audits:** In regulated industries, adherence to documentation standards is often a legal requirement. Documentation serves as evidence of compliance with regulatory standards and provides the necessary documentation for audits and inspections.
7. **Customer Support and Training:** User manuals, guides, and other customer-facing documentation are crucial for helping users understand how to use the product effectively. Clear and comprehensive documentation enhances the customer experience, reduces support requests, and minimizes user frustration.

8. **Continuity and Maintenance:** As products evolve and undergo updates, well-maintained documentation ensures continuity by documenting changes, bug fixes, and new features. This continuity is vital for future maintenance, troubleshooting, and further development efforts.

Documentation plays a crucial role in fostering transparency and collaboration throughout the development of the Rock 'N' Rover. To effectively leverage documentation as a tool for enhancing project outcomes, it is imperative to define clear standards before initiating the documentation process. These standards serve as guiding principles for creating and maintaining various types of documentation, such as design specifications, schematics, software code, and test reports. By adhering to these standards and regularly updating documentation to reflect the project's progress and any modifications, we ensure that every team member remains well-informed and capable of making meaningful contributions to the project's success. Thorough and up-to-date documentation serves as a comprehensive reference point, facilitating efficient communication, decision-making, and problem-solving throughout the development lifecycle.

## **4.2 Design Constraints**

Design constraints are the boundaries within which designers operate when creating a product, system, or solution. These constraints encompass a wide array of factors that influence the design process, ranging from technical limitations to user needs and market requirements. Understanding and effectively navigating these constraints are crucial for producing successful designs that meet the intended objectives and satisfy stakeholders.

Technical constraints are often among the most tangible and immediately apparent limitations designers face. These constraints may include factors such as available materials, manufacturing processes, technological capabilities, and performance requirements. For example, a product designed for mass production may need to adhere to specific dimensions or material properties to ensure cost-effectiveness and feasibility within the manufacturing process. Similarly, technological limitations may dictate the functionality or compatibility of a digital product within existing hardware or software ecosystems.

Beyond technical considerations, designers must also contend with constraints related to user needs and preferences. User-centric design requires a deep understanding of the target audience, including their demographics, behaviors, and pain points. Designers must balance competing user requirements and preferences, ensuring that the final product is intuitive, accessible, and enjoyable to use. Accessibility considerations, for instance, may impose constraints related to font sizes, color contrast, or input methods to accommodate users with disabilities or diverse needs.

Market constraints further shape the design process by reflecting broader economic, cultural, and competitive dynamics. Designers must account for market trends, consumer expectations, and regulatory requirements to ensure their designs remain relevant and

compliant. Competitive analysis can reveal opportunities and constraints inherent in the competitive landscape, guiding designers in differentiating their offerings and delivering unique value propositions.

Additionally, economic constraints such as budgetary limitations or resource availability may influence design decisions, requiring designers to innovate within constrained parameters.

In navigating these multifaceted constraints, designers must adopt a creative and adaptable mindset, viewing limitations not as impediments but as opportunities for innovation and problem-solving. Iterative design processes, prototyping, and interdisciplinary collaboration can help identify and address constraints early in the design cycle, reducing the risk of costly revisions or project delays later on. By embracing constraints as integral components of the design process, designers can leverage them to drive creativity, foster empathy, and ultimately deliver solutions that resonate with users and stakeholders alike.

Design constraints for the Rock ‘N’ Rover:

**1. Technical Constraints:**

- Size and Dimension: The body of the car must adhere to specific dimensions, with a minimum width of 3 inches and a minimum length of 5 inches to accommodate all integrated components effectively.
- Battery Life: The car must be powered by a battery capable of sustaining operation for at least 1 hour to ensure prolonged use and enjoyment.
- Mobility: The design must allow the car to move forwards, backwards, and turn left and right smoothly and efficiently to enable versatile navigation.
- Cost: The total cost of the car must not exceed \$800 to align with budgetary constraints, equivalent to \$200 per team member, ensuring affordability and feasibility.

**2. User Needs and Preferences:**

- Visibility: The headlights must be visible from a distance of at least 3 feet to enhance visibility in low-light conditions, addressing user safety concerns.
- Ease of Use: The controller and app interface must be intuitive and user-friendly, allowing seamless switching between manual and remote control modes to cater to users of varying technical proficiencies.

**3. Market Constraints:**

- Competitive Pricing: The cost of the Rock ‘N’ Rover must be competitive within the market to attract potential buyers and ensure commercial viability.
- Performance: The car must meet or exceed industry standards for remote-controlled cars in terms of performance, functionality, and features to remain competitive and appealing to consumers.

#### **4. Advanced Features:**

- Obstacle Detection: Integration of ultrasonic sensors for obstacle detection to ensure safe navigation and collision avoidance, enhancing user experience and safety.
- Entertainment Features: Incorporation of a speaker for interactive experiences and music playback controlled via the companion app, providing entertainment value to users.
- Camera System: Integration of a camera system with Bluetooth connectivity to provide users with a visual perspective through the app, enabling immersive control and navigation experiences.

#### **5. Manufacturing Constraints:**

- Assembly and Manufacturing Processes: The design must consider ease of assembly and manufacturing processes to streamline production and minimize costs, ensuring scalability and efficiency in manufacturing.

By adhering to these design constraints, the Rock ‘N’ Rover can effectively meet user needs, technological requirements, and market demands, ensuring its success as a cutting-edge remote-controlled car in the competitive landscape of RC vehicles.

### **4.2.1 Wi-Fi Standards in Rock ‘N’ Rover Development**

#### **4.2.1.1 Importance of Wi-Fi Standards**

Wi-Fi, a technology that enables wireless networking, plays a crucial role in modern product development, especially in projects like Rock ‘N’ Rover, where connectivity and data transmission are vital. Adhering to Wi-Fi standards ensures compatibility, reliability, and security in wireless communication, facilitating seamless integration with external devices and systems.

#### **4.2.1.2 IEEE 802.11 Standards**

Wi-Fi standards are defined by the Institute of Electrical and Electronics Engineers (IEEE) under the 802.11 family of standards. The following are some of the key IEEE 802.11 standards relevant to the Rock ‘N’ Rover project:

IEEE 802.11a/b/g/n/ac/ax: These standards define the physical and media access control layers for wireless communication in the 2.4 GHz and 5 GHz frequency bands. Each standard offers improvements in data rates, range, and interference mitigation techniques, allowing for more efficient and reliable wireless connectivity.

IEEE 802.11i (WPA2): Security is paramount in wireless communication, especially in applications like Rock ‘N’ Rover, which may transmit sensitive data. IEEE 802.11i, also known as Wi-Fi Protected Access 2, provides robust

encryption and authentication mechanisms to secure Wi-Fi networks against unauthorized access and data breaches.

IEEE 802.11e (WMM): Quality of Service is essential for applications that require real-time data transmission, such as video streaming or remote control systems in vehicles like Rock ‘N’ Rover. IEEE 802.11e, or Wi-Fi Multimedia, introduces QoS enhancements to prioritize different types of traffic, ensuring a smooth and reliable user experience.

#### **4.2.1.3 Application of Wi-Fi Standards in Rock ‘N’ Rover**

In the context of the Rock ‘N’ Rover project, adherence to Wi-Fi standards is critical for several reasons:

Interoperability: By following IEEE 802.11 standards, Rock ‘N’ Rover can communicate with other Wi-Fi-enabled devices and access points, ensuring interoperability and compatibility in diverse environments.

Reliability: Wi-Fi standards define protocols for error detection, retransmission, and congestion control, enhancing the reliability of wireless communication in Rock ‘N’ Rover. This reliability is essential for maintaining connectivity in challenging conditions, such as remote exploration environments.

Security: Implementing IEEE 802.11i/WPA2 ensures that Rock ‘N’ Rover's Wi-Fi network is protected from unauthorized access and data interception. Robust encryption and authentication mechanisms safeguard sensitive information transmitted between the vehicle and control systems.

Performance: The evolution of Wi-Fi standards, from IEEE 802.11a to the latest IEEE 802.11ax (Wi-Fi 6), offers improvements in data rates, latency, and spectral efficiency. By leveraging advanced Wi-Fi technologies, Rock ‘N’ Rover can achieve higher performance and throughput, supporting demanding applications and data-intensive tasks.

#### **4.2.1.4 Integration and Testing**

Integration of Wi-Fi capabilities into Rock ‘N’ Rover involves hardware implementation, software development, and rigorous testing to ensure functionality and performance. During the integration phase, compatibility with existing Wi-Fi networks and protocols must be verified, and any potential conflicts or interoperability issues addressed.

Testing of Wi-Fi functionality encompasses various scenarios, including range testing, throughput testing, security testing, and interference testing. By subjecting Rock ‘N’ Rover's Wi-Fi system to comprehensive testing procedures, developers can identify and rectify any issues before deployment, ensuring optimal performance in real-world conditions.

## **4.2.2 Additional Applications of Wi-Fi Standards in Rock ‘N’ Rover**

### **4.2.2.1 Remote Monitoring and Control**

Wi-Fi connectivity enables remote monitoring and control of Rock ‘N’ Rover systems from command centers or control stations located at a distance. Operators can access vehicle telemetry data, video feeds, and sensor readings in real-time, allowing for proactive decision-making and intervention during missions.

### **4.2.2.2 Over-the-Air (OTA) Updates**

Wi-Fi standards facilitate over-the-air (OTA) software updates and firmware upgrades for Rock ‘N’ Rover's onboard systems and components. OTA updates streamline the maintenance process, allowing for timely deployment of patches, bug fixes, and feature enhancements without requiring physical access to the vehicle.

### **4.2.2.3 Collaborative Operations**

Wi-Fi connectivity enables collaborative operations between multiple Rock ‘N’ Rover vehicles or between Rock ‘N’ Rover and other autonomous systems. Vehicles can exchange mission-critical data, coordinate movements, and share situational awareness information, enhancing teamwork and coordination in complex missions.

## **4.2.3 Benefits of Wi-Fi Integration in Rock ‘N’ Rover**

### **4.2.3.1 High-Speed Data Transfer**

Wi-Fi standards offer high-speed data transfer capabilities, facilitating rapid exchange of large datasets, multimedia streams, and telemetry data between Rock ‘N’ Rover and external devices or networks. High-speed connectivity enhances operational efficiency and real-time decision-making in mission-critical scenarios.

### **4.2.3.2 Scalability and Interoperability**

Wi-Fi technology supports scalable network architectures and interoperability with a wide range of devices and systems. Rock ‘N’ Rover can seamlessly integrate into existing Wi-Fi networks, infrastructure, and ecosystems, enabling interoperability with ground stations, base camps, or remote facilities.

### **4.2.3.3 Versatile Connectivity**

Wi-Fi connectivity provides versatility in communication options for Rock ‘N’ Rover, including point-to-point, point-to-multipoint, or mesh network configurations. This flexibility allows for adaptive communication strategies based on mission requirements, terrain characteristics, and operational constraints.

## **4.2.4 Challenges and Considerations**

### **4.2.4.1 Signal Interference and Range Limitations**

Wi-Fi signals are susceptible to interference from obstacles, electromagnetic noise, and environmental factors, which can degrade signal quality and reduce coverage range. Mitigating interference and optimizing antenna placement are essential considerations in designing robust Wi-Fi communication systems for Rock ‘N’ Rover.

### **4.2.4.2 Power Consumption and Battery Life**

Wi-Fi connectivity consumes significant power, which can impact battery life and operational endurance in Rock ‘N’ Rover vehicles. Implementing power-efficient Wi-Fi protocols, optimizing transmission parameters, and employing energy-saving techniques are essential for maximizing battery life and mission duration.

### **4.2.4.3 Security and Privacy**

Securing Wi-Fi communication is paramount to protecting Rock ‘N’ Rover's data, systems, and operations from unauthorized access, interception, or tampering. Implementing encryption, authentication, and access control mechanisms ensures confidentiality, integrity, and availability of transmitted data in Wi-Fi networks.

## **4.2.5 Future Considerations and Innovations**

### **4.2.5.1 Wi-Fi 6 and Beyond**

Next-generation Wi-Fi standards, such as Wi-Fi 6 (802.11ax), offer advancements in speed, capacity, and efficiency, making them well-suited for demanding applications like Rock ‘N’ Rover. Adopting Wi-Fi 6 technology enables Rock ‘N’ Rover to leverage enhanced performance, reliability, and connectivity in future deployments.

### **4.2.5.2 Edge Computing and Fog Networking**

Integration of edge computing and fog networking capabilities with Wi-Fi technology enables distributed processing and decision-making at the network edge. Rock ‘N’ Rover vehicles can perform real-time data analysis, sensor fusion, and decision-making locally, reducing latency and bandwidth requirements in Wi-Fi communication.

### **4.2.5.3 Hybrid Connectivity Solutions**

Hybrid connectivity solutions combine Wi-Fi with other wireless technologies, such as cellular networks or satellite communication, to provide redundant and resilient connectivity for Rock ‘N’ Rover in remote or challenging environments. Hybrid solutions ensure continuous communication and mission continuity, even in the absence of Wi-Fi coverage.

## **4.2.6 Bluetooth Standards in Rock ‘N’ Rover Development**

### **4.2.6.1 Importance of Bluetooth Standards**

Bluetooth technology enables short-range wireless communication between devices, making it indispensable for various applications, including automotive systems like Rock ‘N’ Rover. Adherence to Bluetooth standards ensures interoperability, reliability, and security in wireless connections, facilitating seamless integration with external devices and systems.

#### **4.2.6.2 Bluetooth Special Interest Group (SIG) Standards:**

Bluetooth standards are developed and maintained by the Bluetooth Special Interest Group, which defines specifications for Bluetooth communication. The following are some key Bluetooth standards relevant to the Rock ‘N’ Rover project:

**Bluetooth Classic (BR/EDR):** Bluetooth Classic is the original Bluetooth technology primarily used for audio streaming, file transfer, and device pairing. It operates in the 2.4 GHz frequency band and supports data rates up to several megabits per second.

**Bluetooth Low Energy (BLE):** Bluetooth Low Energy is a power-efficient variant of Bluetooth designed for applications requiring low power consumption and long battery life. It is suitable for transmitting small packets of data intermittently, making it ideal for sensors, wearables, and IoT devices.

**Bluetooth Mesh:** Bluetooth Mesh extends Bluetooth connectivity to large-scale networks, allowing devices to communicate with each other in a mesh topology. This standard is relevant for applications where multiple devices need to work together, such as smart home automation or industrial IoT deployments.

### **4.2.6.3 Application of Bluetooth Standards in Rock ‘N’ Rover**

Bluetooth standards offer several benefits when integrated into the Rock ‘N’ Rover project:

**Wireless Connectivity:** Bluetooth enables wireless communication between Rock ‘N’ Rover and external devices, such as smartphones, tablets, or wearable devices worn by operators or researchers. This wireless connectivity facilitates remote control, data exchange, and telemetry monitoring without the need for physical connections.

**Sensor Integration:** Bluetooth Low Energy (BLE) can be utilized to interface with sensors and peripherals onboard Rock ‘N’ Rover, such as environmental sensors, GPS modules, or inertial measurement units (IMUs). By leveraging BLE, Rock ‘N’ Rover can collect sensor data wirelessly and transmit it to the control system or external devices for analysis.

**Human-Machine Interface (HMI):** Bluetooth Classic enables the establishment of a wireless link between Rock ‘N’ Rover and user interfaces, such as mobile apps or dashboard displays. This HMI integration allows operators to interact with



Rock ‘N’ Rover remotely, sending commands, receiving telemetry data, and monitoring vehicle status in real-time.

Mesh Networking (Future Consideration): As the Rock ‘N’ Rover project evolves and expands, Bluetooth Mesh could be explored as a means of creating a self-healing and resilient network architecture. By forming a mesh network, multiple vehicles or nodes could communicate with each other and coordinate their actions, enhancing collaboration and scalability in complex environments.

#### **4.2.6.4 Integration and Testing**

Integration of Bluetooth functionality into Rock ‘N’ Rover involves hardware integration, firmware development, and testing to ensure compatibility and performance. Hardware components such as Bluetooth modules or chips must be selected based on compatibility with Bluetooth standards and Rock ‘N’ Rover’s requirements.

Firmware development entails implementing Bluetooth protocols, profiles, and services to enable desired functionalities, such as data transmission, device discovery, and secure pairing. Thorough testing is conducted to verify Bluetooth interoperability, connectivity range, power consumption, and adherence to security best practices.

### **4.2.7 Additional Applications of Bluetooth Standards in Rock ‘N’ Rover**

#### **4.2.7.1 Vehicle-to-Vehicle Communication**

Bluetooth standards can facilitate vehicle-to-vehicle (V2V) communication, allowing multiple Rock ‘N’ Rover vehicles to exchange data and coordinate their movements. This capability enables collaborative tasks, such as formation driving, convoy operations, or distributed sensing, enhancing the efficiency and safety of missions in rugged or hazardous terrain.

#### **4.2.7.2 Fleet Management and Coordination**

By integrating Bluetooth connectivity, Rock ‘N’ Rover vehicles can communicate with a central control system or fleet management platform. This communication enables real-time monitoring of vehicle status, location tracking, route optimization, and task allocation, streamlining operations and maximizing resource utilization in large-scale deployments.

#### **4.2.7.3 Sensor Fusion and Data Fusion**

Bluetooth standards complement other sensor technologies onboard Rock ‘N’ Rover, such as GPS, LiDAR, or cameras, by providing additional contextual information or redundancy. Sensor fusion techniques can be employed to integrate data from Bluetooth-enabled sensors with other sensor modalities, enhancing situational awareness, obstacle detection, and navigation accuracy.

## **4.2.8 Benefits of Bluetooth Integration in Rock ‘N’ Rover**

### **4.2.8.1 Wireless Flexibility**

Bluetooth integration eliminates the need for cumbersome wired connections, reducing cable clutter and simplifying vehicle design. Wireless connectivity enhances flexibility in deploying and reconfiguring Rock ‘N’ Rover systems, accommodating dynamic mission requirements and operational scenarios.

### **4.2.8.2 User Convenience**

Bluetooth connectivity enables seamless interaction between Rock ‘N’ Rover and user interfaces, such as smartphones or tablets. Operators can control the vehicle, monitor telemetry data, and receive alerts or notifications remotely, enhancing convenience and accessibility in field deployments or remote exploration missions.

### **4.2.8.3 Low Power Consumption**

Bluetooth Low Energy (BLE) technology offers power-efficient communication, extending battery life in Rock ‘N’ Rover's onboard devices and sensors. Low-power operation is particularly advantageous for applications requiring prolonged operation in remote or off-grid environments, where power constraints are significant.

## **4.2.9 Challenges and Considerations**

### **4.2.9.1 Interference and Signal Degradation**

Wireless communication, including Bluetooth, is susceptible to interference from environmental factors, electromagnetic noise, or other wireless devices operating in the vicinity. Mitigating interference and ensuring reliable signal transmission are critical considerations in designing robust Bluetooth-enabled systems for Rock ‘N’ Rover.

### **4.2.9.2 Security and Privacy**

While Bluetooth standards incorporate security features such as encryption and authentication, vulnerabilities and privacy concerns may still arise, especially in mission-critical applications. Implementing robust security measures and adhering to best practices in secure communication protocols are essential to safeguarding Rock ‘N’ Rover's data and operations.

### **4.2.9.3 Range and Coverage Limitations**

Bluetooth's range and coverage are inherently limited compared to other wireless technologies like Wi-Fi or cellular networks. Ensuring adequate coverage and range for communication between Rock ‘N’ Rover and external devices or control systems requires careful consideration of antenna placement, power levels, and environmental factors.

## **4.2.10 Future Considerations and Innovations**

### **4.2.10.1 Bluetooth 5.0 and Beyond**

Future iterations of Bluetooth technology, such as Bluetooth 5.0 and upcoming releases, offer advancements in range, data rate, and reliability. Rock ‘N’ Rover can leverage these enhancements to achieve higher performance, extended range, and improved connectivity in challenging environments.

### **4.2.10.2 Edge Computing and Bluetooth Mesh**

Integration of Bluetooth Mesh networking and edge computing capabilities enables distributed processing and decision-making at the network edge. Rock ‘N’ Rover vehicles can form self-organizing mesh networks, exchanging data and collaborating autonomously without relying on centralized control, enhancing resilience and adaptability in dynamic environments.

### **4.2.10.3 Standardization and Interoperability**

Continued efforts in standardization and interoperability initiatives promote compatibility and seamless integration between Bluetooth-enabled devices and ecosystems. Rock ‘N’ Rover can benefit from standardized protocols, profiles, and interoperability guidelines, ensuring compatibility with a wide range of Bluetooth-enabled devices and systems.

## **4.2.11 SPI Standards in Rock ‘N’ Rover Development**

### **4.2.11.1 Importance of SPI Standards**

SPI (Serial Peripheral Interface) is a synchronous serial communication interface commonly used in embedded systems, microcontrollers, and integrated circuits. Adherence to SPI standards ensures efficient data transfer, device interoperability, and compatibility in interconnected systems, making it essential for the development of Rock ‘N’ Rover's electronics and control systems.

### **4.2.11.2 SPI Specifications**

SPI standards define the electrical characteristics, communication protocols, and timing requirements for data exchange between master and slave devices. The following are key aspects of SPI standards relevant to the Rock ‘N’ Rover project:

Hardware Interface: SPI uses a master-slave architecture, where one device acts as the master controller initiating data transfers, and multiple devices serve as slaves responding to commands from the master. The hardware interface typically consists of four signal lines: MOSI (Master Out Slave In), MISO (Master In Slave Out), SCK (Serial Clock), and SS (Slave Select).

Communication Protocol: SPI employs full-duplex communication, allowing simultaneous transmission and reception of data between master and slave devices. Data is transmitted in a synchronous manner, with the master generating

clock pulses to synchronize data transfer. SPI supports various data formats, including 8-bit, 16-bit, or custom formats based on application requirements. Clock Speed and Timing: SPI standards specify clock speed limits and timing constraints for reliable data transfer. Clock speed is configurable and can range from a few kHz to several MHz, depending on the capabilities of the devices involved. Timing parameters such as setup time, hold time, and clock phase (CPOL) and polarity (CPHA) are critical for maintaining signal integrity and synchronization.

#### **4.2.11.3 Application of SPI Standards in Rock ‘N’ Rover**

SPI standards offer several benefits when integrated into the Rock ‘N’ Rover project:

Sensor and Peripheral Integration: SPI facilitates communication with a wide range of sensors, actuators, and peripheral devices onboard Rock ‘N’ Rover, such as inertial measurement units (IMUs), GPS modules, motor controllers, and display screens. By adhering to SPI standards, Rock ‘N’ Rover can interface seamlessly with SPI-compatible devices, enabling data exchange and control functionalities.

Embedded System Communication: SPI communication is widely used for inter-chip communication within embedded systems, allowing microcontrollers, sensors, and peripheral ICs to exchange data efficiently. Rock ‘N’ Rover's onboard electronics, including microcontrollers, sensor arrays, and motor drivers, can communicate via SPI, enabling coordinated operation and control.

High-Speed Data Transfer: SPI's full-duplex communication and configurable clock speed enable high-speed data transfer between Rock ‘N’ Rover's control system and external devices. This capability is advantageous for real-time data acquisition, command execution, and telemetry monitoring, enhancing responsiveness and performance in mission-critical applications.

#### **4.2.11.4 Integration and Testing**

Integration of SPI functionality into Rock ‘N’ Rover involves hardware design, firmware development, and testing to ensure compatibility and reliability. Hardware components such as microcontrollers, sensors, and peripheral ICs must support SPI communication and adhere to SPI standards.

Firmware development entails implementing SPI communication protocols, data formatting, and error handling routines in the embedded software running on Rock ‘N’ Rover's microcontrollers. Thorough testing is conducted to verify SPI functionality, data integrity, and timing accuracy under various operating conditions.

## **4.2.12 Additional Applications and Benefits of SPI Integration**

### **4.2.12.1 Real-Time Control and Feedback**

SPI facilitates real-time control and feedback loops in Rock ‘N’ Rover's systems, enabling rapid response to changing environmental conditions or mission requirements. By interfacing with motor controllers, sensor arrays, and feedback devices via SPI, Rock ‘N’ Rover can adjust its trajectory, speed, and behavior dynamically during operation.

### **4.2.12.2 Reduced Wiring Complexity**

SPI communication minimizes wiring complexity in Rock ‘N’ Rover's electronics and control systems, as multiple devices can share the same SPI bus using dedicated chip select lines. This simplifies the hardware layout, reduces signal routing complexity, and conserves board space, contributing to a more compact and streamlined design.

### **4.2.12.3 Multimodal Sensor Fusion**

SPI facilitates multimodal sensor fusion in Rock ‘N’ Rover, allowing data from different sensors to be synchronized and combined for comprehensive environmental perception. By interfacing with IMUs, cameras, LiDAR sensors, and environmental sensors via SPI, Rock ‘N’ Rover can enhance its situational awareness, obstacle detection, and navigation capabilities.

### **4.2.12.4 Energy-Efficient Operation**

SPI communication consumes minimal power compared to other communication interfaces, making it suitable for energy-constrained applications in Rock ‘N’ Rover. Low-power operation is advantageous for extending battery life, optimizing energy efficiency, and enhancing mission endurance, particularly in remote or off-grid deployments.

## **4.2.13 Challenges and Considerations**

### **4.2.13.1 Signal Integrity and Noise Immunity**

SPI signals are susceptible to noise, crosstalk, and signal integrity issues, especially in electrically noisy environments or high-speed communication scenarios. Implementing proper signal conditioning, shielding techniques, and noise suppression mechanisms are essential for maintaining signal integrity and reliability in Rock ‘N’ Rover's SPI communication.

### **4.2.13.2 Timing Synchronization and Latency**

Achieving precise timing synchronization and minimizing latency in SPI communication is critical for coordinated operation and control in Rock ‘N’ Rover's systems. Variations in clock skew, propagation delays, and data transmission rates can affect system performance and responsiveness, requiring careful timing calibration and synchronization strategies.

#### **4.2.13.3 Scalability and Expansion**

As Rock ‘N’ Rover evolves and additional functionalities are integrated, scalability and expansion of SPI communication interfaces become important considerations. Designing modular, extensible SPI architectures allows for seamless integration of new sensors, actuators, or control modules, enabling scalability and future-proofing of Rock ‘N’ Rover's systems.

### **4.2.14 Future Considerations and Innovations**

#### **4.2.14.1 High-Speed SPI Variants**

Exploration of high-speed SPI variants, such as Quad SPI (QSPI) or Octal SPI, offers potential enhancements in data transfer rates, bandwidth, and throughput for Rock ‘N’ Rover's applications. Adopting advanced SPI technologies enables faster communication, reduced latency, and increased performance in data-intensive tasks.

#### **4.2.14.2 Fault Tolerance and Redundancy**

Incorporating fault-tolerant and redundant SPI communication mechanisms enhances reliability and fault resilience in Rock ‘N’ Rover's systems. Implementing error detection, correction, and retry mechanisms mitigates the impact of communication errors, transient faults, or device failures, ensuring continuous operation and mission success.

#### **4.2.14.3 Standardization and Interoperability**

Adherence to standardized SPI protocols, pinouts, and communication conventions promotes interoperability and compatibility between Rock ‘N’ Rover's components and subsystems. Following established SPI standards ensures seamless integration with off-the-shelf sensors, actuators, and embedded modules, simplifying procurement, development, and maintenance processes.

## **5. Project Design Details**

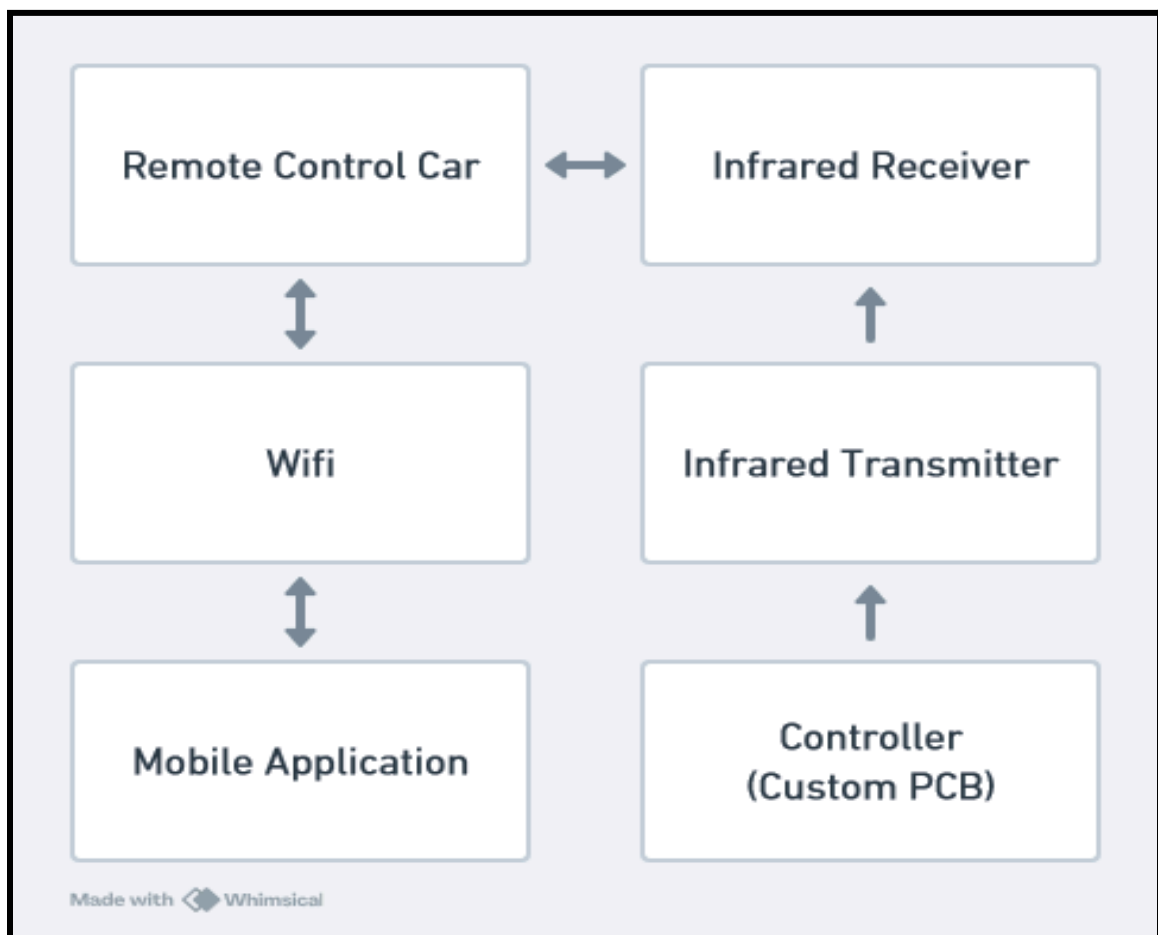
In this section, we will discuss the design of the project. We will discuss various hardware and software diagrams and discuss the design involved in various portions of the project.

### **5.1 Hardware Diagrams**

In this section, we will show the hardware diagrams that were created after the initial hardware diagrams were created. The hardware diagrams shown are the main block diagram and the block diagrams for the two subsystems: the remote control car and the controller.

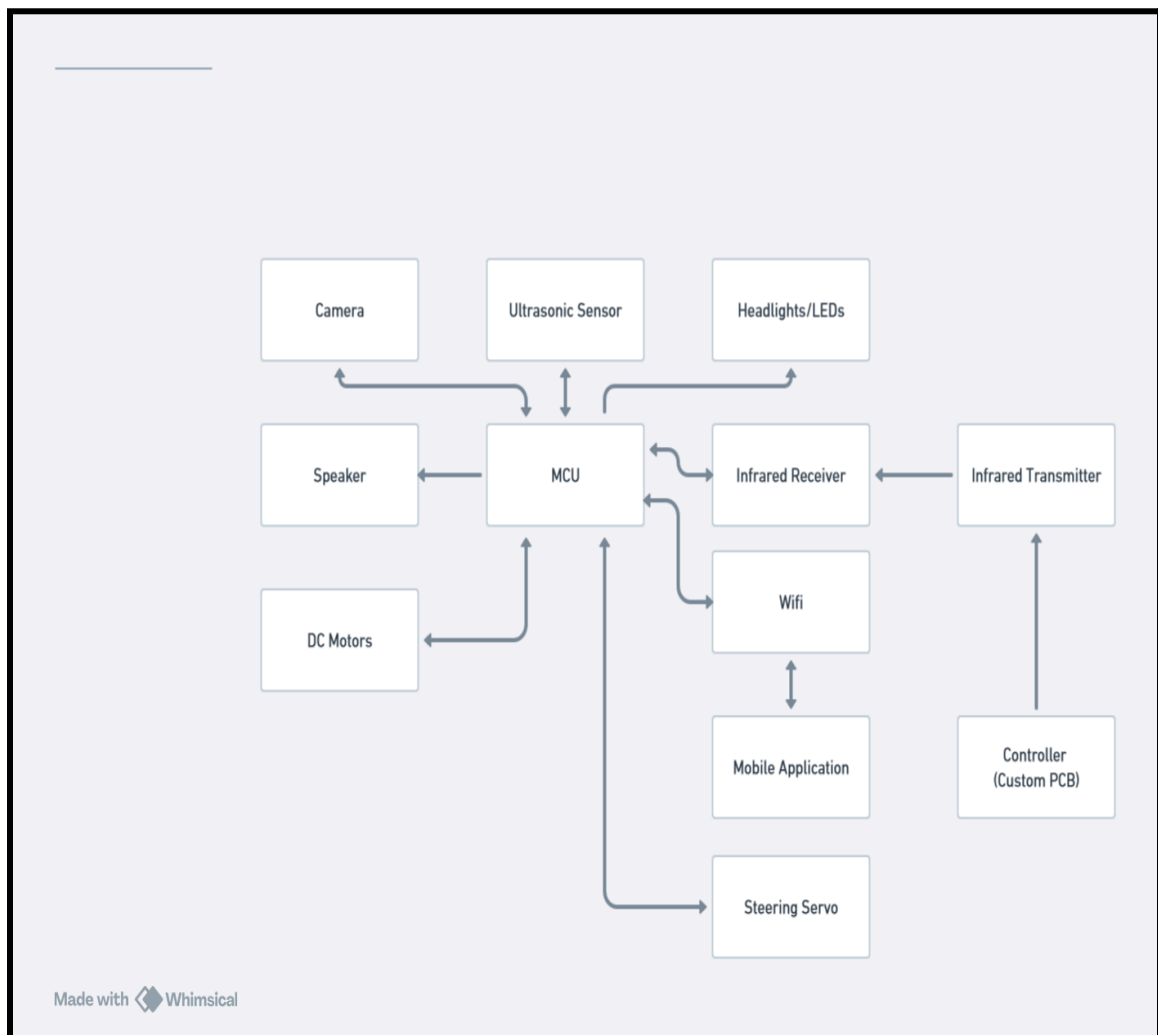
#### **5.1.1 Main Block Diagram**

The main block diagram shows how the three main components of our project are connected to each other. The three main components of our project are: the mobile application, the remote control car, and the controller. The controller is connected to the remote control car via an infrared transmitter and receiver. The controller communicates to the remote control car via the infrared transmitter and the infrared receiver on the remote control car receives the communication. The mobile application connects to the remote control car via a Wifi interface.



### **5.1.2 Remote Control Car Block Diagram**

The hardware diagram for the remote control car is shown below. This hardware diagram was derived from the initial hardware diagram. Initially, we were going to connect the mobile application and the controller to the remote control car via Bluetooth. However, it was decided to connect the mobile application to the remote control car via Wifi and it was decided to connect the controller to the remote control car via an infrared transmitter and receiver. The remote control car will feature a steering servo, DC motors, a speaker, a camera, an ultrasonic sensor, and headlights. The steering servo was used to steer the car left and right. The DC motors will move the car forwards and backwards. The camera can be used to view a POV from the car via the mobile application. The ultrasonic sensor will detect if the car gets too close to an object and the speaker will produce a beep if the ultrasonic sensor detects that the car gets too close to an object.

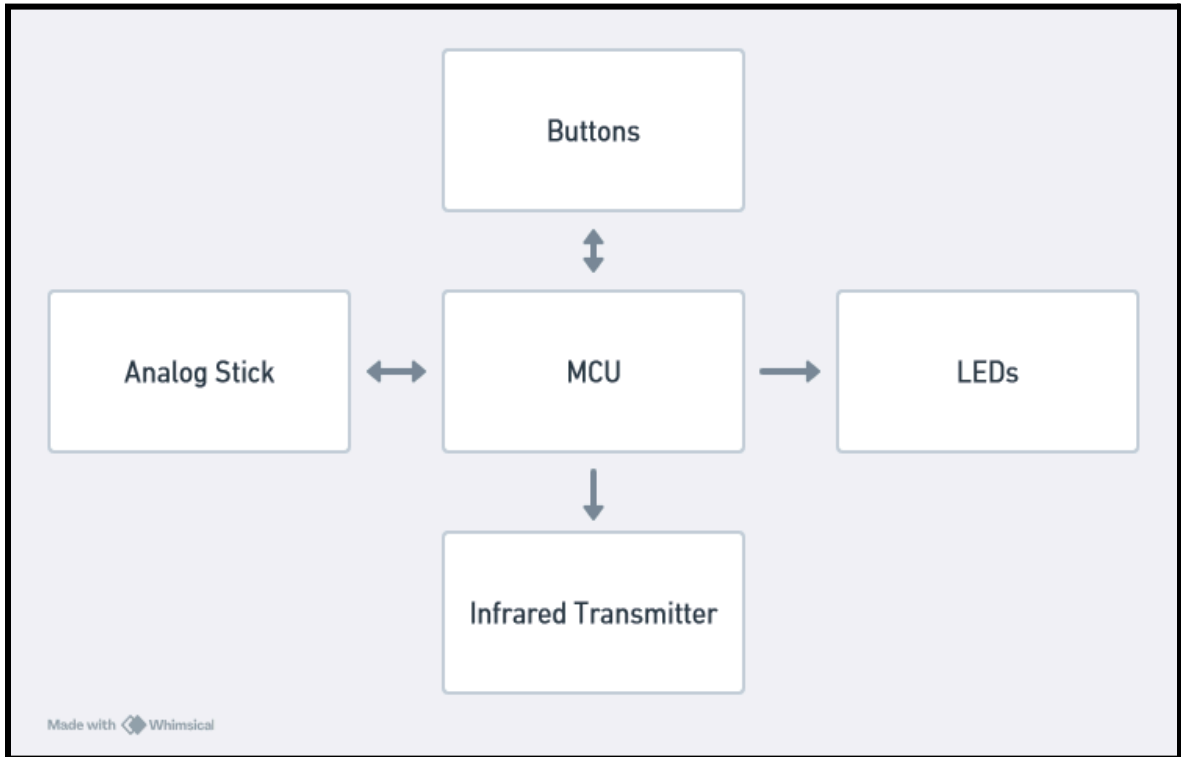


### **5.1.3 Controller Block Diagram**

The hardware diagram for the controller for the remote control car is shown below. The controller uses a custom PCB design that will feature buttons, an analog stick, LEDs, and

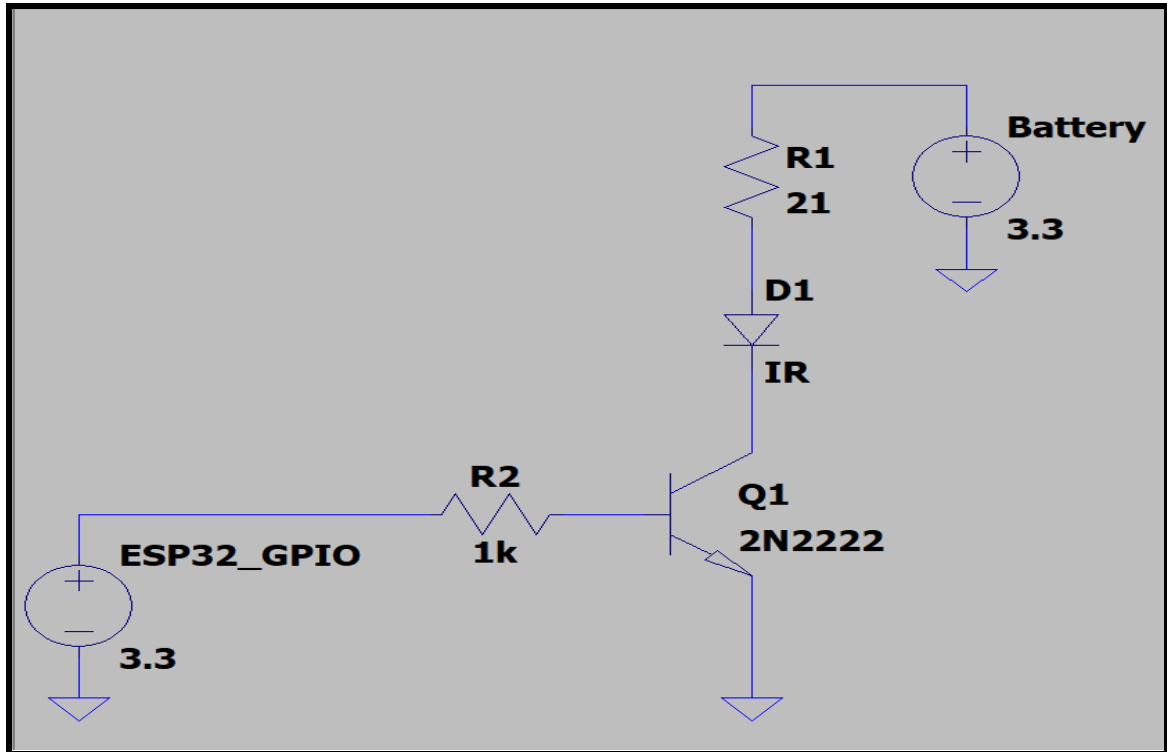


an infrared transmitter. The buttons are used to toggle different features of the remote control car. The buttons would have been able to toggle the headlights and the ultrasonic sensor on the remote control car, but we ran out of time. The analog stick be used to control the movement of the remote control car. The infrared transmitter communicates to the receiver on the remote control car in order to control it. And of course, all of these peripherals are connected to a microcontroller.



## **5.2 IR Emitter Circuit**

Shown below is the circuit design required to connect the IR LED to the ESP32 microcontroller.



Since each GPIO pin on the ESP32 is rated to have a maximum current of 40 mA, we can not connect the IR LED directly to the GPIO pin to control the transmission of data. This is because the IR LED has a maximum rating of 100 mA, and we need a little more than 40 mA to get a larger range and a stronger connection from the remote controller to the remote control car. If we connected the GPIO pin to the IR LED directly, it would risk damaging the pin or the microcontroller itself. Therefore, we must use a 2N2222 BJT transistor to act as a switch to control the flow of current through the IR LED. So, when the ESP32 GPIO pin provides a logical HIGH, then the transistor is on and current will flow through the IR LED and the IR LED was on. When the ESP32 GPIO pin provides a logical LOW, then the transistor is off and the current will not flow through the IR LED and the IR LED will also be off. We aim for a current of about 80% of the maximum current rating for the IR LED because according to the datasheet for our IR LED, the forward voltage can range from 1.2 V to 1.8 V, so by aiming for 80% of the maximum current rating, we can ensure that the current does not exceed the maximum current rating. The IR LED has an average forward voltage of approximately 1.46 V. According to the datasheet for our IR LED, the minimum forward voltage is 1.2 V and the maximum forward voltage is 1.8 V. Therefore, if we connect a 21 ohm resistor to the IR LED, we should see a current of 81.9 mA at 1.46 V and a minimum current of 66.5 mA and a maximum current of 93.8 mA. So, the current will not exceed the 100 mA maximum rating of the IR LED.

## **5.3 Rock N' Rover App Development**

The following section provides a comprehensive overview of the foundational elements underpinning the development of the Rock N' Rover app controller. It explores the platform chosen for programming, detailing its capabilities and suitability for the project's requirements. Additionally, this segment examines the array of features integrated within the app controller, highlighting their significance in enhancing user interaction and rover functionality. Furthermore, it scrutinizes the prototyping and design processes undertaken to ensure a smooth and intuitive user experience. By delving into these critical aspects, this section offers valuable insights into the careful planning and execution essential for the successful realization of the Rock N' Rover app controller.

### **5.3.1 Introduction to MIT App Inventor**

MIT App Inventor offers a user-friendly platform for individuals with varying levels of programming experience to create innovative applications for a range of purposes. When it comes to Arduino projects, MIT Inventor becomes an invaluable tool, simplifying the process of designing and deploying apps that interact with Arduino-based hardware. The platform's drag-and-drop interface, coupled with its visual programming language, allows users to swiftly develop functional applications without delving deeply into complex coding syntax.

One of the primary advantages of utilizing MIT Inventor for Arduino projects is its accessibility. Whether you're a novice or an experienced developer, the platform provides an intuitive environment for prototyping and deploying applications. This accessibility democratizes the development of Arduino-based solutions, empowering individuals with diverse backgrounds to bring their ideas to life without extensive programming knowledge.

Now, envision leveraging MIT Inventor to create an app to control the Rock 'N' Rover RC car. By integrating the app with the Arduino hardware embedded within the car, users can enjoy a seamless remote control experience. The app interface could feature intuitive controls such as directional buttons or a joystick, enabling users to maneuver the RC car effortlessly.

Beyond basic control functions, the app incorporates additional features to enhance the user experience. For instance, real-time telemetry data could be displayed on the app interface, providing users with insights into the car's speed, battery level, and other vital metrics. Moreover, incorporating features like customizable driving modes or automated routines adds depth to the app, catering to both casual users and enthusiasts seeking more advanced functionalities.

The versatility of MIT Inventor allows for further customization and expansion of the app's capabilities. For instance, incorporating Bluetooth connectivity enables wireless control of the RC car, freeing users from tethered connections and enhancing mobility. Additionally, integrating sensors or cameras into the car and interfacing them with the

app opens up possibilities for augmented reality experiences or obstacle detection functionalities.

In conclusion, leveraging MIT Inventor to create an app for controlling the Rock 'N' Rover RC car streamlines the development process while offering a plethora of possibilities for enhancing user interaction and functionality. By harnessing the platform's intuitive interface and robust feature set, developers can craft innovative solutions that redefine the remote control experience for RC enthusiasts and hobbyists alike.

### **5.3.2 MIT App Inventor Concepts**

#### **Components:**

Components serve as the foundational elements of an MIT App Inventor project, representing the diverse range of user interface elements and functionalities that constitute the app. These components include buttons, text boxes, labels, images, sensors, and communication modules like Bluetooth and Wi-Fi. Each component is added to the design canvas and configured to define its behavior and appearance within the app's interface.

#### **Events:**

Events are triggered by user actions or system events and are associated with specific components within the app. For instance, a button click event or a sensor value change event. When an event occurs, the corresponding event handler block executes, enabling you to define the app's response. Events are fundamental to the app's interactivity and functionality, dictating how it behaves in response to user input or external stimuli.

#### **Blocks Editor:**

The Blocks Editor is the primary workspace where you define the app's behavior using visual programming blocks. Each component in the app has associated blocks that control its properties and respond to events. These blocks are dragged and dropped onto the workspace and connected together to create sequences of actions and decisions. The Blocks Editor provides a visual representation of the app's logic, making it accessible to users without prior programming experience.

#### **Programming Logic:**

MIT App Inventor employs a visual programming paradigm based on blocks, simplifying the app development process. The programming logic is expressed through a combination of event-driven programming and sequential execution of blocks. Event handlers respond to user interactions or system events by executing specific sequences of blocks. These blocks can include conditional statements (if-else), loops (for, while), mathematical operations, variable manipulation, and function calls, allowing you to define the app's behavior and functionality.

#### **Variables and Data Storage:**

Variables enable you to store and manipulate data within your app. MIT App Inventor supports different types of variables, including numbers, text, and lists. Additionally, the

platform provides built-in storage options such as TinyDB for local data storage and CloudDB for cloud-based data synchronization. These capabilities empower you to create dynamic and data-driven apps that respond to user input and maintain state across sessions.

#### User Interface Layout:

The user interface layout is crafted using the Design Editor, where you arrange and configure the visual components of your app. This involves setting properties such as size, position, text, color, and visibility for each component. The layout design plays a crucial role in defining the app's aesthetic appeal and usability, influencing the overall user experience.

#### Testing and Debugging:

MIT App Inventor offers built-in tools for testing and debugging your app during development. You can connect your device to the MIT App Inventor companion app to test your app in real-time as you build it. Additionally, the Blocks Editor provides debugging features such as breakpoints and step-by-step execution, facilitating the identification and resolution of issues in your app's logic. Testing and debugging are integral parts of the development process, ensuring that your app functions as intended across different devices and scenarios.

### **5.3.3 Innovative App Features**

The MIT Inventor app designed for controlling the Rock 'N' Rover RC car can incorporate a plethora of features to elevate the user experience and extend the functionality of the vehicle. Here are some potential features that could be integrated into the app:

Directional Control: The primary function of the app would be to provide intuitive directional control over the RC car. Users could utilize virtual joysticks or directional buttons on the app interface to steer the car in various directions, allowing for smooth and precise maneuvering.

Speed Adjustment: Implementing speed adjustment controls within the app enables users to regulate the RC car's velocity according to their preferences or the specific terrain conditions. This feature could include sliders or buttons for adjusting the speed in real-time, offering flexibility and adaptability during driving sessions.

Custom Driving Modes: Introducing custom driving modes expands the versatility of the RC car, catering to different user preferences and scenarios. Users could create and save personalized driving profiles within the app, each tailored to specific driving conditions or purposes, such as high-speed racing, precise maneuvering, or obstacle traversal.

Telemetry Display: Real-time telemetry data displayed on the app interface provides users with valuable insights into the RC car's performance and status.

Information such as battery level, motor temperature, speed, and distance traveled can be showcased dynamically, empowering users to monitor and optimize the vehicle's operation.

Automated Routines: Incorporating pre-programmed automated routines into the app enables the RC car to execute predefined actions or sequences autonomously. Users could design and upload custom routines, such as figure-eight loops, obstacle courses, or choreographed movements, enhancing the entertainment value and showcasing the vehicle's capabilities.

Wireless Connectivity: Leveraging Bluetooth or Wi-Fi connectivity, the app facilitates wireless communication with the RC car, eliminating the need for physical connections and enhancing mobility. This feature enables users to control the vehicle from a greater distance and facilitates seamless integration with other smart devices or peripherals.

Camera Integration: Integrating a camera module into the RC car and streaming live video feed to the app allows users to experience immersive first-person driving or explore remote environments from a unique perspective. The app could include features such as image capture, video recording, and live streaming, enhancing the excitement and utility of the RC car.

By incorporating these and other innovative features into the MIT Inventor app, developers can create a comprehensive and immersive control interface for the Rock 'N' Rover RC car, unlocking new possibilities for remote control enjoyment and exploration.

### **5.3.4 App Implementation**

The communication between the Arduino and the controller is outlined in the software framework, detailing the execution and functionality of the program. Initially, the Arduino initializes, establishing connections with the controller. It then proceeds to gather sensor data to maintain environmental awareness, crucial for obstacle detection and navigation. Subsequently, the program configures the motors, enabling precise control over rover movements and manipulation. User inputs received from the controller trigger adjustments in rover movements, responding dynamically to sensor data analysis. After completing designated tasks, such as obstacle avoidance or target acquisition, the program halts the motors, marking the end of its execution until further inputs are received.

Creating an MIT App Inventor app involves several steps, each geared towards developing a functional and user-friendly application. The process begins with conceptualization and planning, where you define the app's purpose, target audience, and key features. Understanding the needs and preferences of potential users is crucial in shaping the app's design and functionality.

Once the concept is solidified, you move on to the design phase. Using MIT App Inventor's intuitive drag-and-drop interface, you can assemble the various components of

your app's user interface (UI). This includes buttons, text boxes, images, and other interactive elements that will form the backbone of your app's layout. Paying attention to UI design principles such as simplicity, consistency, and accessibility is essential to ensure a seamless user experience.

After designing the UI, you proceed to the coding phase. MIT App Inventor utilizes a visual programming language based on blocks, which simplifies the coding process even for individuals with limited programming experience. By arranging blocks in a logical sequence, you can define the app's behavior and functionality. This may involve setting up event handlers for user interactions, integrating external services or hardware components, and implementing algorithms to process data or perform computations.

Testing and debugging represent critical stages in the app development process. Once the initial version of the app is complete, you should thoroughly test it across different devices and operating systems to identify any bugs, glitches, or usability issues. MIT App Inventor provides built-in tools for live testing, allowing you to preview the app's behavior on a connected device in real-time. Iterative testing and refinement are essential to ensure that the app meets the desired performance standards and user expectations.

Finally, once the app has been thoroughly tested and refined, you can package and distribute it to your target audience. MIT App Inventor provides options for generating installation files compatible with Android devices, allowing you to deploy your app through various channels such as the Google Play Store or direct download. Additionally, you may consider soliciting feedback from users and iterating on the app based on their suggestions and preferences, thereby fostering continuous improvement and innovation.

### **5.3.5 App Description**

#### **User Interface Design:**

The user interface (UI) design is meticulously crafted to prioritize user experience and ease of navigation. The landing page serves as the initial point of interaction, featuring the application name prominently displayed to establish brand identity and recognition. The "Get Started" button is strategically positioned, employing contrasting colors or visual cues to attract user attention and encourage interaction. Upon tapping the button, users seamlessly transition to the controls page, where the layout is optimized for ergonomic usability. The directional buttons are intuitively arranged, mimicking the layout of a traditional RC controller to facilitate a seamless transition for users familiar with such devices. Additionally, the stop button is strategically placed to ensure immediate accessibility in emergency situations, enhancing safety and user confidence.

#### **Camera View Integration:**

The integration of a camera view option within the controls page represents a significant enhancement in user functionality and operational versatility. Leveraging the device's camera capabilities, users gain real-time visual feedback of the RC car's surroundings, enabling precise navigation and obstacle avoidance. The camera feed is seamlessly

integrated into the controls page, ensuring uninterrupted access without compromising the intuitiveness of the interface. Advanced features such as zoom and pan functionalities may be incorporated to further enhance user control and situational awareness, catering to diverse user preferences and operational requirements.

#### Navigation and Settings:

The navigation system is designed to offer intuitive access to various application sections while maintaining spatial coherence and visual consistency. A side menu layout is adopted, providing a persistent navigation panel accessible via a swipe gesture or a dedicated menu button. This approach ensures uninterrupted access to key functionalities, including the controls page and settings interface, without disrupting the user's workflow. Within the settings interface, users encounter a comprehensive array of customization options tailored to their preferences and operational requirements. Slider controls or numerical input fields may be employed to facilitate precise adjustments, ensuring a personalized and immersive user experience.

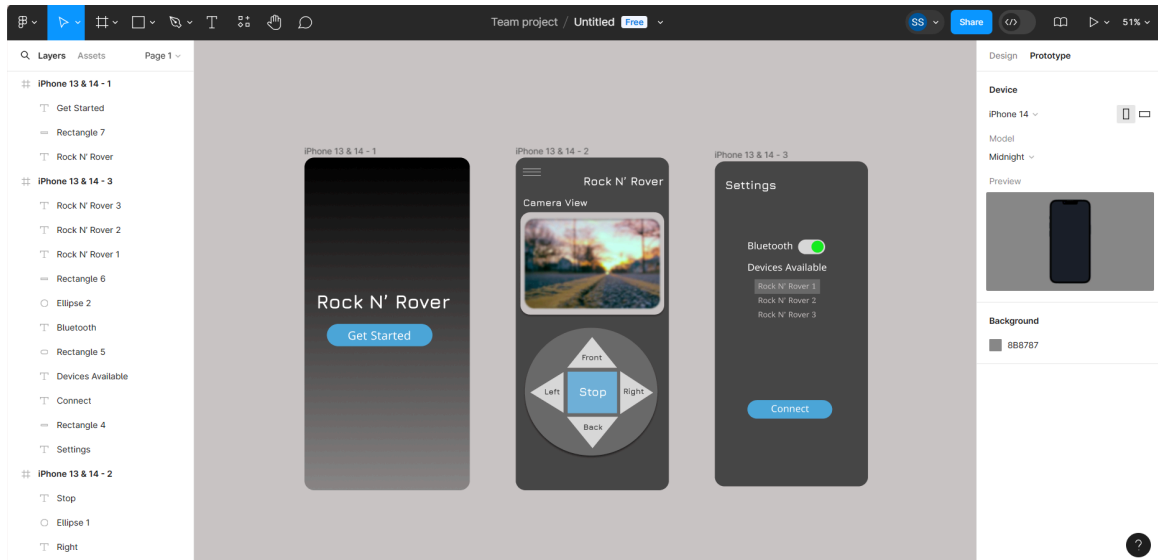
#### Backend Integration:

The seamless integration of the application with the Arduino microcontroller forms the backbone of its operational functionality and real-time responsiveness. Through Bluetooth or Wi-Fi connectivity protocols, the application establishes a reliable communication link with the Arduino board, enabling bidirectional data transfer. Advanced data parsing algorithms may be employed to interpret user inputs efficiently, minimizing latency and enhancing responsiveness. Error handling mechanisms are implemented to address communication disruptions or data inconsistencies, ensuring uninterrupted operation under diverse environmental conditions. Additionally, the application's backend architecture may incorporate modular components, facilitating future scalability and extensibility to accommodate additional features or device integrations seamlessly. Comprehensive testing procedures, including stress testing and compatibility testing, are conducted to validate the application's robustness and reliability across various hardware configurations and usage scenarios.

### **5.3.6 App Design**

The app controller for the Rock N' Rover was prototyped in Figma, a versatile and intuitive design tool. Figma stands out for its collaborative features, enabling multiple team members to work on designs simultaneously, regardless of their geographical locations. Its cloud-based nature ensures seamless sharing and real-time updates, fostering efficient communication and iteration within the design process. The image of the Figma development atmosphere is shown below. Moreover, Figma offers a wide array of interactive prototyping functionalities, allowing designers to simulate user interactions and test the app's usability comprehensively. By utilizing Figma, the team can refine the user experience of the Rock N' Rover's app controller with precision and agility, ultimately enhancing its functionality and user satisfaction.

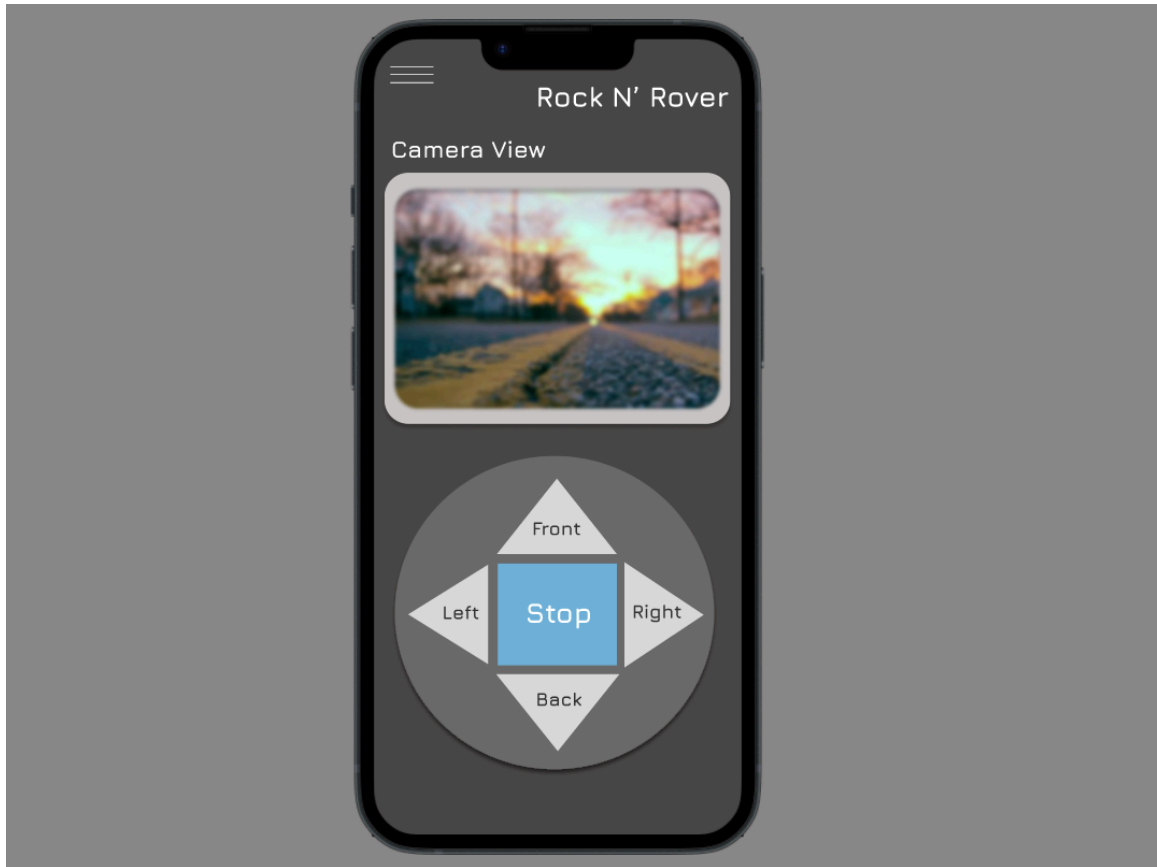




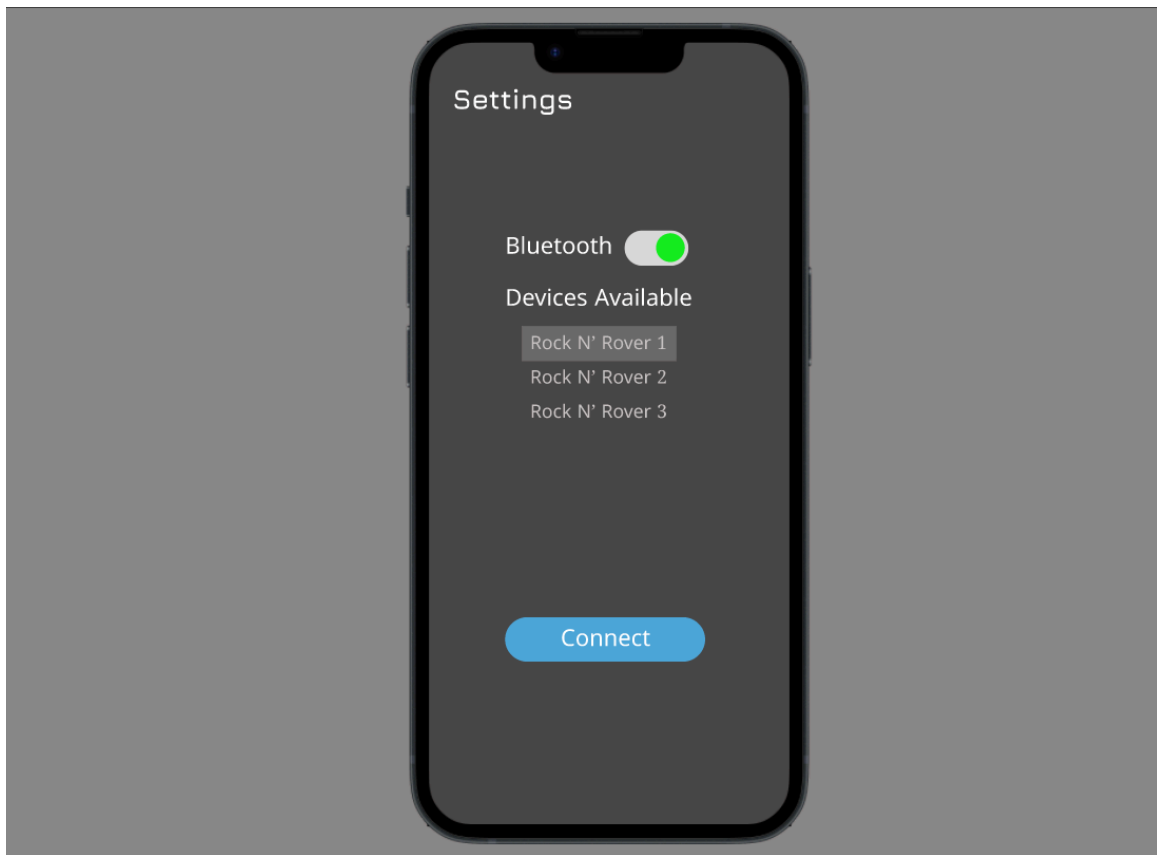
The Rock N' Rover app's design blueprint strategically positions the landing page as the initial encounter for users, marking the entry point into its immersive experience. Here, the app's title takes center stage, serving as a focal point to establish brand identity. Complementing this, a prominent "Get Started" button is strategically placed to prompt user interaction. Upon activation, this button seamlessly redirects users to the home page, marking the commencement of their journey. This intentional design not only ensures a user-friendly experience but also sets the standard for intuitive navigation throughout the app. Thus, users are promised an engaging and effortless exploration of the Rock N' Rover universe from the outset.



The Home page of the Rock N' Rover app presents users with a comprehensive interface designed to facilitate navigation of the rover. Central to this interface is the camera view, providing users with real-time visual feedback of the rover's surroundings. Adjacent to the camera view are the primary navigation controls, comprising the Front, Back, Left, Right, and Stop buttons. These controls empower users to maneuver the Rock N' Rover with precision and ease, allowing for efficient exploration of diverse terrains. By integrating both visual feedback and intuitive controls within the Home page, the Rock N' Rover app prioritizes user experience and functionality, ensuring a robust and immersive interaction with the rover's capabilities.



The Settings page within the Rock N' Rover app serves as a pivotal hub for user customization and control over various functionalities. Among its key features is the ability to manage access to Bluetooth devices, affording users the flexibility to enable or disable wireless connections as needed. Notably, the app controller establishes a wireless connection with the Rock N' Rover, enhancing convenience and mobility during operation. Additionally, the Settings page introduces a range of supplementary features, which may be integrated based on project timelines and primary feature completion. These include music streaming capabilities, enabling users to enjoy their favorite tunes while exploring, as well as LED light control functionality for enhanced visibility in low-light environments and creating an ambient environment. Furthermore, the inclusion of volume control for music playback further enriches the user experience, emphasizing the app's commitment to versatility and user-centric design.



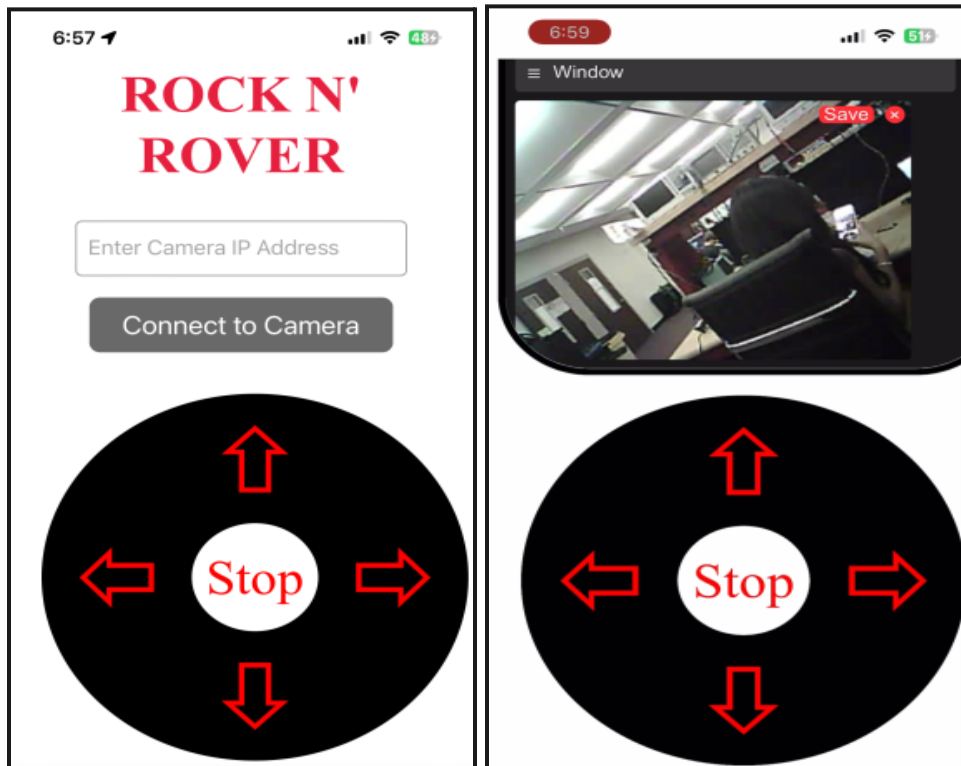
### **5.3.7 Changes in Approach: Final App Design**

Three weeks before the completion of our Senior Design 2 project, our team decided to transition from using MIT App Inventor to developing a web app with HTML, CSS, and JavaScript. MIT App Inventor, while user-friendly, proved too restrictive for our needs, particularly in terms of design flexibility. We encountered significant challenges in sending movement commands and integrating the camera module, which prompted us to pivot to a more versatile and robust solution.

The current app design is much more user-friendly and efficient. The landing page features intuitive buttons for maneuvering the RC car, making navigation straightforward. At the top of the page, a search box allows users to enter the car's camera IP address for a real-time video feed. Upon entering the correct IP address, camera settings appear, allowing users to configure their camera. From these settings, users can start the video stream and begin navigating using the provided controls.

We faced several challenges during this transition. Driving the car with the IR receiver allowed for the car to stop when no IR signal was received. Our app included a stop

button to manually halt the car, causing conflicts with the IR receiver's default behavior. To resolve these conflicts, we had to rebuild the app from scratch. MIT App Inventor's limitations also posed significant obstacles, as it was too restrictive for our design needs and complicated the process of sending movement commands and integrating the camera module. By switching to hard coding with HTML, CSS, and JavaScript, we gained greater control over the app's functionality and design. The new code sends HTTP requests to the car whenever a command button (Front, Left, Back, Right, Stop) is pressed, ensuring smoother and more reliable operation.



## **6. Project Construction**

In this section, we will discuss some topics related to the construction of the project.

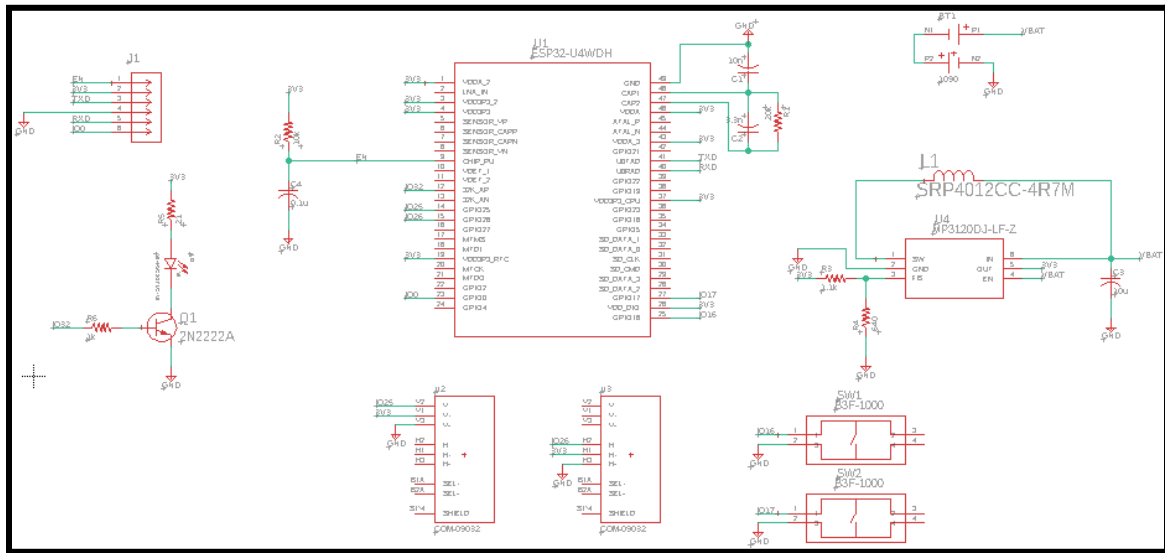
### **6.1 PCB Schematics**

For our project, we had two custom PCBs. One PCB is for the remote controller to send commands to the car. The other PCB is for controlling the car itself.

#### **6.1.1 Remote Controller Schematic 1st Version**

The remote controller PCB is used to send commands to the RC car via infrared transmission. The remote controller PCB consists of a microcontroller, a AA battery holder, a step-up converter, two buttons, two analog sticks, an infrared transmitter, and a 6-pin interface for flashing the code to the microcontroller. The overall schematic for the remote control car is shown below.

Figure 6.1.1 - Remote Controller PCB Schematic

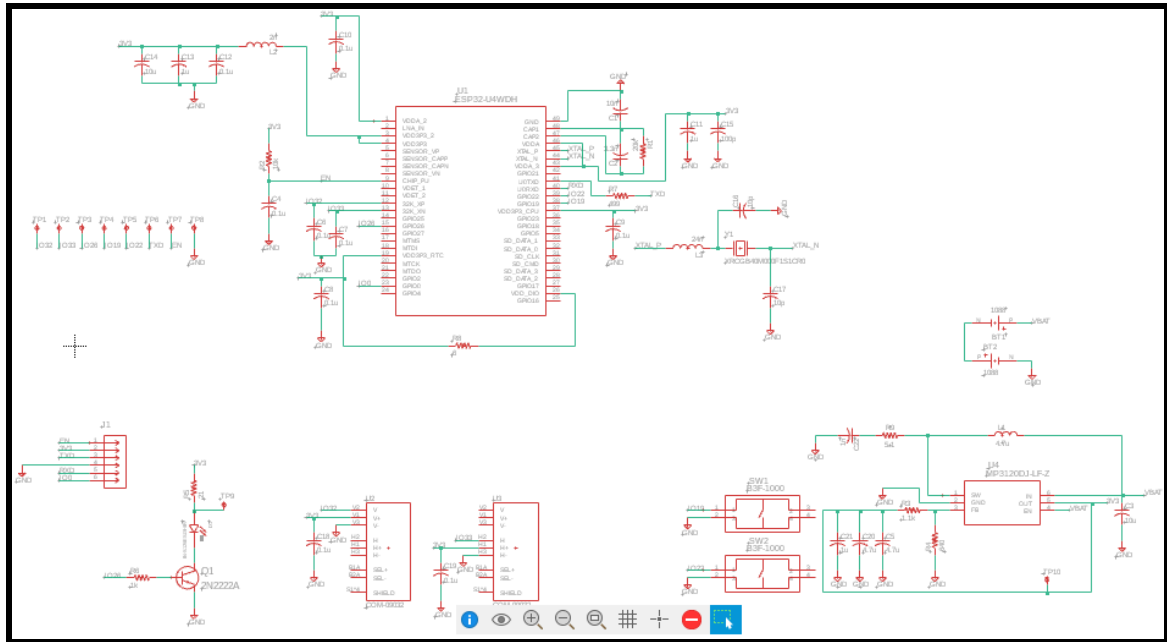


In the middle of the PCB, you can see the ESP32-U4WDH microcontroller. In the top right corner, there is the AA battery holder. It holds two AA batteries and will output a voltage of 3 V at full charge. Directly beneath the AA battery holder, there is the step-up converter and the circuitry required for it to step-up the voltage from the AA batteries to 3.3 V. Beneath the step-up converter, there are two buttons, connected to two separate GPIO pins to read the values of the buttons. The GPIO pins have internal pull-up/pull-down resistors so no external pull-up resistors are required. To the left of the buttons, you see two analog sticks. One analog stick is for left and right movement of the car, while the other analog stick is for moving the car forwards and backwards. In the top left corner is the 6-pin interface used to flash the code to the microcontroller via UART. Beneath the 6-pin interface is the infrared transmitter. The infrared transmitter draws 3.3 V power. The BJT transistor connected to the anode of the infrared transmitter is used to turn the infrared transmitter on and off, so the transistor will act like a switch. A GPIO pin is connected to the base of the BJT transistor and the GPIO pin outputs a logical HIGH to turn on the infrared transmitter and outputs a logical LOW to turn off the infrared transmitter. A second version of the remote controller PCB was needed.

## 6.1.2 Remote Controller Schematic 2nd Version

This is the second version of the schematic for the remote controller PCB.

Figure 6.1.2 - Remote Controller PCB Schematic

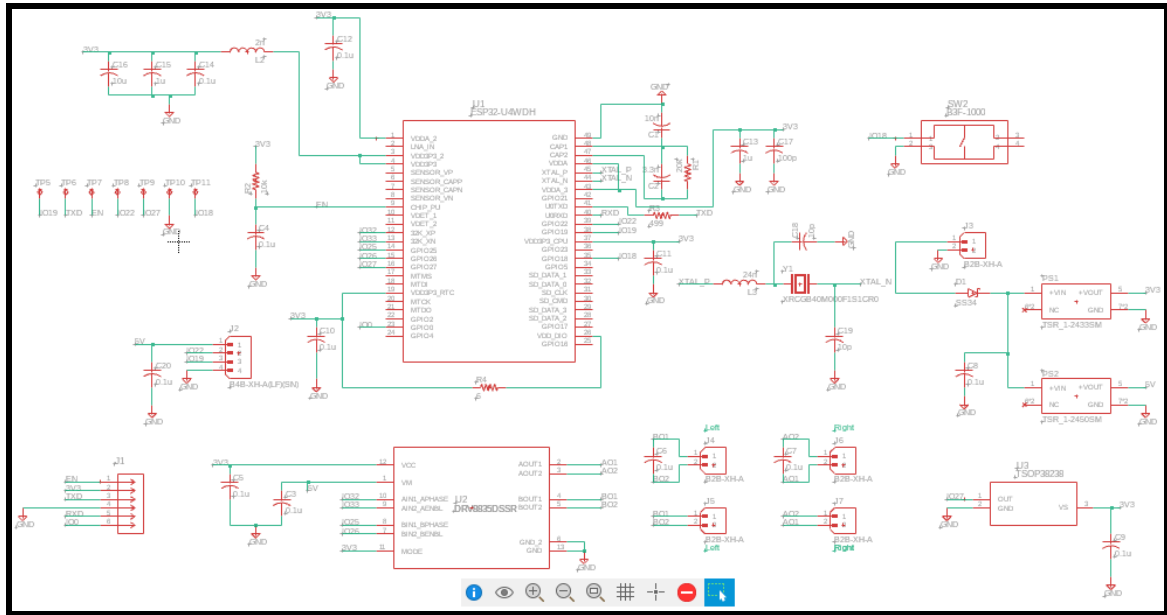


A second version of the remote controller was needed because there was an issue with the step-up converter where the voltage output had spikes, which was affecting the performance of the system. In order to remove this spike, an RC snubber was added to the switching loop of the step-up converter IC and the 10 uF output capacitor was replaced with two 4.7 uF capacitors and a 1 uF capacitor. Also, in the board layout, the external components were moved closer to the step-up converter IC.

### 6.1.3 RC Car Schematic

The RC car PCB is used to receive commands from the remote controller via infrared transmission and is used to control the DC motors of the car and the ultrasonic sensor. The RC car PCB consists of a microcontroller, several 2-pin interfaces, two DC/DC converters, an infrared receiver, a motor driver IC, a 4-pin interface, and a 6-pin interface. The overall schematic for the RC car is shown below.

Figure 6.1.3 - Car PCB Schematic



In the middle of the PCB, there is the ESP32-U4WDH microcontroller. In the top right corner, there is a 2-pin interface to connect our 7.4 V battery. The 2-pin interface is connected to two DC/DC converters. One DC/DC converter outputs 3.3 V and the other DC/DC converter outputs 5 V. Both DC/DC converters have internal bypass capacitors, so no external bypass capacitors are needed. Below the DC/DC converters is the infrared receiver. It is connected to a GPIO pin that reads from the infrared receiver. Beneath the infrared receiver, there are four 2-pin interfaces to connect the DC motors to the DC motor driver IC. There are two 2-pin interfaces for controlling the front and rear motors on the left side and there are another two 2-pin interfaces for controlling the front and rear motors on the right side. To the left of the 2-pin interfaces is the motor driver IC. The motor driver IC is connected to the 3.3 V power supply to power the logic and is connected to a 5 V power supply to power the motors. There are a total of four input pins. Two input pins control the left side motors and the other two input pins control the right side motors. Each input pin is connected to their own GPIO pin on the microcontroller. In the top left, there is the 6-pin interface to flash the code to the microcontroller. Beneath the 6-pin interface is a 4-pin interface to read from the ultrasonic sensor. No further version of the car PCB was needed.

## **7. Project Testing**

In this section, we will discuss how we tested individual components used in the project.

### **7.1 Kit components Testing**

The most important testing that we will do is to make sure the components in our kit work. We had a few ways to do this, but we decided the most effective way would be to assemble it just like the manufacturers intended, and then see if everything works. There are a couple of extra things that we will not need in our final design, like the arduino shield and the line tracking component, but the shield was amazing for testing. It included male header pins, which allowed us to make perfect connections with all of the components in the kit. We used the exact same thing on our PCB design. The only issue we ran into is the companion app that is provided. It has a joystick to control the car, but we had a hard time steering the car. This was worrisome initially, but we hoped it was just because the app design was extremely small, so our fingers moving on a dot on the screen was not going to be an accurate portrayal of what the joystick will be. After plugging the DC motors into our own arduino, we were able to get much more controlled responses, which led us to think our initial hypothesis was correct.

### **7.2 Headlights Testing**

This was an extremely simple process. Due to us not even connecting the line tracking module in the assembly of the RC car, we had an open 3 pin header to connect our two female pins to. One of them is 5V and the other is GND. We again went the simple route first, and just connected the lights to 5V. Once we saw that that worked, we connected the 5V end to a digital GPIO pin, and after outputting HIGH, the headlights worked again. After that, we mounted it to the frame of the car, and are awaiting our PCB design next semester.

### **7.3 Hand Held PCB Testing**

This was a bit trickier, as all of the components we used here were surface mount, and we are going to have them assembled at the manufacturer. With that being said, the codes are extremely easy for each component, so we purchased a random component kit and made sure that the code would work.

### **7.4 Infrared Transmitter/Receiver Testing**

The infrared transmitter and the infrared receiver were tested on a breadboard. The Arduino Nano ESP32 board was used, the TSAL6400 infrared transmitter was used, and the TSOP32838 infrared receiver was used. The TSAL6400 is a different transmitter from the one we are using. The IN-S126ESGHIR is the infrared transmitter we used on our PCB, however, it is a surface-mount component so for testing purposes, a similar through-hole component was used. The TSAL6400 is a similar through-hole infrared transmitter. As discussed in a previous section, the microcontroller board's GPIO was connected to a transistor and the transistor was used as a switch to turn the infrared transmitter on and off. The infrared transmitter was pointed at the receiver.



```

10:16:27.191 -> Protocol=NEC Address=0x0 Command=0x6 Raw-Data=0xF906FF00 32 bits LSB first
10:16:29.748 -> Protocol=NEC Address=0x0 Command=0x8 Raw-Data=0xF708FF00 32 bits LSB first
10:16:31.863 -> Protocol=NEC Address=0x0 Command=0x1 Raw-Data=0xFE01FF00 32 bits LSB first
10:16:33.999 -> Protocol=NEC Address=0x0 Command=0x3 Raw-Data=0xFC03FF00 32 bits LSB first
10:16:36.181 -> Protocol=NEC Address=0x0 Command=0x5 Raw-Data=0xFA05FF00 32 bits LSB first
10:16:38.271 -> Protocol=NEC Address=0x0 Command=0x7 Raw-Data=0xF807FF00 32 bits LSB first

```

To test the infrared transmitter and receiver, the terminal was used to print the protocol, the address, and the command that was received. The raw command was also printed in hexadecimal format. The code to test the infrared transmitter and receiver consisted of a for loop to repeatedly send commands from the transmitter to the receiver. The NEC protocol was used to encode and send the commands. The for loop sent commands 1 through 9, however, the receiver was only receiving every other command. This is likely due to a bug in the code. After each command was sent, the microcontroller would check if the command was received and decoded by the receiver, and if it was, then the microcontroller would read the data from the receiver and print the data to the terminal. Then, the interrupt for the receiver would be reset and the receiver would resume to receive and decode the next command. Some issues arose during testing. The first issue was that only one command was being received, however, this was because the interrupt was not reset in the code after a command was decoded and received. The second issue was that it was not receiving the appropriate commands unless there was an object placed behind the receiver. It appeared that the signal was bouncing off of the object and then was received by the receiver rather than directly being sent to the receiver within the line-of-sight. If there was not an object behind the receiver, the receiver would just be receiving random commands. This issue was likely due to interference and/or the positioning of the infrared transmitter or the infrared receiver on the breadboard. To fix this issue, some of the wires were moved away from the transmitter and the receiver. It seemed like one of the wires was causing interference with the receiver, but the wires near the transmitter were also moved to minimize interference. This fixed some of the issue, but the commands that were being sent were still not being received reliably. After some adjusting of the positioning of the transmitter and the receiver, the receiver was receiving the commands with minimal interference.

## **7.5 Senior Design 2 Testing**

During Senior Design 2, the PCBs were tested to make sure that the design was working as intended. The PCB to control the peripherals of the remote control car was working during the first iteration and no further iteration was needed. The remote controller PCB to send infrared commands to the car had an issue with the step-up converter and needed a second iteration.

### **7.5.1 Car PCB Motor Driver**

In order to test the functionality of the car PCB and the motor driver and the motors, a test code was created to repeatedly toggle the motors on and off in a loop. After the test code was uploaded, the motors were being turned on and off as intended, so the functionality of the PCB, the motor driver, and the motors were verified to be working correctly.

### **7.5.2 Car PCB IR Receiver**

In order to test the functionality of the infrared receiver on the car PCB, the code was written to receive a test infrared command and if that infrared command was received, the motors would turn on. Since at this point, we did not have the remote controller PCB working, we used a remote that could be purchased online, just to verify that the infrared receiver could receive commands correctly.

### **7.5.3 Remote Controller PCB IR Transmitter**

To test the remote controller PCB and the infrared transmitter, the code was written to send out different infrared commands based on the position of the joysticks. Upon testing, it was discovered that the voltage output of the step-up converter had spikes, which was interfering with the voltage that the analog/digital converter was reading from the joysticks. These spikes were also seen in the infrared signal that the microcontroller was sending to the infrared transmitter. Therefore, the schematic and the board layout of the external components of the step-up converter IC needed to be changed to get rid of these voltage output spikes because it was affecting the performance of the entire system. It was also discovered that the infrared transmitter was mounted the wrong way as the anode and the cathode of the LED needed to be connected the other way around. Also, even though this was a side-mount component, there were issues getting the LED to have a proper connection to the copper pads on the board, so the LED could not be mounted on its side and had to be mounted facing upwards.

### **7.5.4 ESP32 CAM**

During senior design 1, we were going to use the Elegoo WROOM cam. We found out how difficult it was to download code and interface with that module, so we decided to use a much cheaper and simpler alternative, the ESP32 CAM. This allowed us to use more resources online and all we had to do was go onto the arduino IDE, download their camerawebserver example code, and we were able to get the camera up and running. We modified that code to make the camera more usable for our application.

## **8. Administrative Content**

In this section, we will discuss planned milestones and a planned schedule for completing the project. We will also discuss the planned budget for our project.

### **8.1 Project Milestones/Schedule**

	Senior Design I
Week	Description and Due Dates
1-2	Develop a detailed project plan
3-4	Research and Benchmarking
5	Begin the design process for the robotic machine components
6	Create initial prototypes
7-8	Conduct initial tests to validate design concepts
9-10	Present progress to advisors and adjust design plans
11-14	Finalize detailed designs for each module and select specific components, sensors, and materials for construction
15	Assemble a basic prototype to test the integration of key components
16	Perform initial functionality tests

Senior Design I:

#### **Week 1-2: Develop a Detailed Project Plan**

During the first two weeks, the focus is on laying down the groundwork for the entire project. This involves creating a detailed project plan that outlines the scope, objectives, milestones, and tasks to be completed. An extensive project plan was used as a roadmap for the team, ensuring everyone is aligned on the project's direction and responsibilities. It should include timelines, resource allocations, risk assessments, and communication protocols. By dedicating sufficient time to planning upfront, our team can mitigate potential risks and set a strong foundation for the project's success.

#### **Week 3-4: Research and Benchmarking**

Research and benchmarking are essential steps for gathering information and understanding the landscape of the project. This phase involves conducting literature reviews, studying relevant technologies, and analyzing existing solutions or products in

the market. By benchmarking against competitors or similar projects, our team can identify best practices, challenges, and opportunities. This information will inform the design process and help our team make informed decisions throughout the project lifecycle.

### Week 5: Begin the Design Process for Robotic Machine Components

With a solid understanding of the project requirements and the findings from research and benchmarking, our team can now begin the design process. This involves brainstorming, conceptualizing, and sketching out ideas for the robotic machine components. The focus should be on translating requirements into feasible design concepts while considering factors such as functionality, manufacturability, and scalability. Collaboration and communication among team members are crucial during this phase to ensure diverse perspectives are considered, leading to innovative designs.

### Week 6: Create Initial Prototypes

Creating initial prototypes allows our team to materialize our design concepts and test them in a tangible form. Prototyping helps identify potential design flaws, technical challenges, and areas for improvement early in the development process. These prototypes don't need to be perfect but should be functional enough to validate key design decisions. Our team should be prepared to iterate on our prototypes based on feedback from testing and evaluation.

### Week 7-8: Conduct Initial Tests to Validate Design Concepts

During weeks 7-8, our team conducts initial tests to validate the design concepts and functionalities of the prototypes. This involves setting up test scenarios, performing experiments, and collecting data to assess the performance and reliability of the designs. The focus should be on identifying any discrepancies between expected and observed results and addressing them through iterative design improvements. Documenting test procedures and outcomes is essential for tracking progress and informing future iterations.

### Week 9-10: Present Progress to Advisors and Adjust Design Plans

Presenting progress to advisors provides an opportunity for external feedback and validation of the project's direction. Our team should prepare a comprehensive progress report or presentation that highlights achievements, challenges, and proposed solutions. Advisors may offer valuable insights, suggestions, or critiques that can help refine the design plans and address any concerns. We should be prepared to approach these meetings with an open mind and be receptive to constructive feedback, as it can ultimately lead to a stronger project outcome.

### Week 11-14: Finalize Detailed Designs and Select Components

During this phase, our team finalizes detailed designs for each module of the robotic machine and selects specific components, sensors, and materials for construction. This involves refining the design documentation, creating detailed drawings or schematics, and specifying the required parts and materials. Attention to detail is crucial to ensure the compatibility, functionality, and performance of the selected components. Our team should also consider factors such as cost, availability, and lead time when making decisions.

### Week 15: Assemble a Basic Prototype for Integration Testing

Assembling a basic prototype allows our team to test the integration of key components and subsystems. This involves physically putting together the individual modules or components according to the finalized designs. Integration testing helps identify any compatibility issues, wiring errors, or mechanical constraints that may arise during assembly. Our team should conduct thorough testing to ensure proper functionality and alignment with project requirements.

### Week 16: Perform Initial Functionality Tests

In the final week, our team performs initial functionality tests on the assembled prototype to evaluate its performance and validate its functionality. This involves executing test cases, measuring key performance metrics, and assessing the overall reliability of the prototype. Any inconsistencies or deficiencies identified during testing should be documented and addressed through iterative design improvements. By completing functionality tests, our team can gain confidence in the prototype's capabilities and readiness for further development or refinement.

Overall, each week of the project plan plays a necessary role in progressing through the various stages of design, development, and testing. Effective collaboration, communication, and attention to detail are key factors for achieving success in senior design projects. By following the outlined timeline and diligently executing each phase of the plan, our team can maximize our chances of delivering a high-quality robotic machine that meets the specified requirements and objectives.

	Senior Design II
Week	Description and Due Dates
1	Address any issues identified
2	Continue refining prototype
3	Begin integrating different modules for comprehensive system test
4-5	User interface and mobile app development

6	Implement customizable settings based on user feedback
7	Conduct extensive system testing to ensure all components work
8-10	Optimize software algorithms for enhanced performance, finalize assembly of the robotic machine, conduct testing
11-12	Prepare a comprehensive final presentation and documentation and submit and present the final project documentation

## Senior Design 2:

### Week 1: Address any Issues Identified

During the first week, our team focuses on addressing any issues or challenges identified from the previous phase. This may involve troubleshooting technical issues, resolving design flaws, or incorporating feedback from advisors or initial testing. Quickly addressing these issues ensures that the project remains on track and sets a solid foundation for further development.

### Week 2: Continue Refining Prototype

In week 2, our team continues refining the prototype based on the insights gained from addressing identified issues. This may involve making iterative design improvements, optimizing functionality, or enhancing user experience. The goal is to enhance the prototype towards meeting the project's objectives and user requirements.

### Week 3: Begin Integrating Different Modules for Comprehensive System Test

Week 3 marks the beginning of integrating different modules of the robotic system for comprehensive testing. This involves assembling various components and subsystems to create a unified system. Our team conducts initial system tests to evaluate the interaction between different modules and ensure seamless integration. Any integration issues or compatibility issues are addressed quickly to ensure smooth progress.

### Week 4-5: User Interface and Mobile App Development

During weeks 4-5, our team focuses on developing the user interface and mobile application for controlling the robotic machine. This involves designing intuitive user interfaces, implementing features, and ensuring compatibility across different devices and platforms. User feedback is solicited and incorporated into the development process to enhance usability and functionality.

### Week 6: Implement Customizable Settings Based on User Feedback

In week 6, our team implements customizable settings based on user feedback gathered during the UI and mobile app development phase. This may involve incorporating user preferences, adjustable parameters, or personalized features to enhance the user experience. Iterative testing and refinement ensure that the customizable settings meet user expectations and contribute to overall satisfaction.

### Week 7: Conduct Extensive System Testing

Week 7 is dedicated to conducting extensive system testing to ensure that all components and functionalities of the robotic machine work as intended. This involves executing predefined test cases, stress testing, and evaluating performance under various conditions. Comprehensive testing helps identify and address any remaining issues or deficiencies before finalizing the project.

### Week 8-10: Optimize Software Algorithms, Finalize Assembly, and Conduct Testing

During weeks 8-10, our team focuses on optimizing software algorithms for enhanced performance, finalizing the assembly of the robotic machine, and conducting thorough testing. This involves refining algorithms for efficiency, reliability, and accuracy, completing the assembly of mechanical and electronic components, and performing integration testing. The goal is to ensure that the robotic machine operates smoothly and meets all project requirements.

### Week 11-12: Prepare Comprehensive Final Presentation and Documentation

In the final weeks of the project, our team prepares a comprehensive final presentation and documentation. This involves summarizing key findings, achievements, and lessons learned throughout the project. Our team creates professional documentation, including technical reports, user manuals, and assembly instructions. The final presentation showcases the project's accomplishments and highlights its significance, innovation, and potential impact.

By following the outlined plan and executing each week's tasks, our team can successfully complete Senior Design II and deliver a high-quality robotic system that meets user needs and project objectives.

## **8.2 Initial Budget Breakdown**

We did not have a designated project sponsor, the funding for this endeavor was collectively contributed by our group members. We had set a budget ceiling of \$800, equating to \$200 per person. Although we possess certain testing components such as Arduinos, wires, breadboards, servos, and various sensors, the acquisition of additional project items did necessitate some expenditure.

## Cost Breakdown of Required Components:

ELEGOO UNO R3 Smart Robot Car Kit V4: \$80

This is the kit car that we chose. It includes nearly everything that we need except for the headlights and PCB controller.

ESP8266 (MCU to control the car): \$15

The ESP8266 microcontroller unit (MCU) plays a crucial role in controlling the car's functions and communication. Its affordability makes it a suitable choice for this project.

INJORA RC Headlights: \$10

We attached the headlights to the frame of the car in order to illuminate the front view, so the camera can see where it is going in the dark.

Battery: \$13

This was attached to the PCB and power our controller. A reliable power source is essential for powering the robotic system. The cost includes purchasing a suitable battery pack capable of providing sufficient voltage and capacity to support the system's operation.

PLA Filament: \$20

PLA filament is a 3D printing material used for fabricating various components and prototypes. The cost covers a significant portion of the budget, reflecting the importance of 3D printing in the project's development process.

## Additional Considerations:

- While some testing components such as Arduinos, wires, breadboards, small servos, ultrasonic sensors, and a PS4 controller are available for use at no additional cost, the acquisition of specialized components and materials requires extra spending.
- The project's budget allocation must be managed wisely to ensure optimal utilization of resources and adherence to the budget ceiling. Prioritizing essential components and seeking cost-effective alternatives can help maximize the project's value within the available budget.
- Regular budget tracking and monitoring are essential throughout the project's lifecycle to identify any overspending. Adjustments may be necessary to reallocate resources or find additional funding if required.

By conducting a thorough budget analysis and carefully planning the purchasing of required components, the project can effectively manage its finances and ensure the successful execution of the robotic system development within the specified budget constraints.



### **8.3 Updated Budget Breakdown**

As our project progresses, it's important to refine our budget allocation based on evolving requirements and emerging priorities. The updated budget reflects a more unrefined breakdown of estimated costs for individual components, providing greater transparency and accountability in financial planning.

Item	Price
Car Kit	\$68.15
Car Kit	\$68.15
PCBs (Version 1)	\$300.00
USB to Serial	\$20.00
USB to Serial	\$20.00
IR Kit	\$5.00
MCU	\$5.00
Miscellaneous	\$25.00
<b>Total</b>	<b>\$511.30</b>

Justification of Costs:

1. **Microcontroller Chip:** The microcontroller chip serves as the central processing unit for our robotic system, enabling control and coordination of various functions. Its low cost reflects its ubiquity and affordability in the market.
2. **IR Transmitter and Receiver:** Infrared (IR) transmitters and receivers are essential for wireless communication between components of the robotic system. Their minimal cost shows their fundamental role in creating data exchange.
3. **Motor Driver:** The motor driver circuitry regulates the power supply to the motors, ensuring smooth and precise movement of the robotic vehicle. Its cost reflects its critical function in maintaining operational efficiency.
4. **DC/DC Converters and Step-Up Converter:** These components are essential for voltage regulation and power management within the robotic system. Their higher costs are attributed to their specialized functionality and reliability requirements.
5. **PCB & Assembly:** The printed circuit board (PCB) serves as the backbone of our electronic system, providing a platform for component integration and electrical

connectivity. The cost includes PCB fabrication and assembly services to ensure quality and reliability.

6. **Other Components:** This category encompasses components and materials required for various aspects of the project, such as sensors, connectors, and mounting hardware. The estimated cost accounts for anticipated expenses based on project specifications and technical requirements.

#### Budget Management:

As we proceed with the project, diligent budget management was very important to ensuring optimal resource utilization and adherence to financial constraints. Regular monitoring and tracking of expenses, as well as periodic budget reviews, will allow us to identify potential cost overruns and implement corrective measures accordingly.

By maintaining a clear understanding of our budgetary requirements and effectively managing our financial resources, we can push through the complexities of project funding while remaining focused on achieving our objectives. The updated budget serves as a comprehensive roadmap for allocating funds and guiding financial decisions throughout the project lifecycle.

## **8.5 Budgeting Risks**

In undertaking our senior design project, it's necessary to identify potential risks and uncertainties that may impact project execution and outcomes. An approach to risk management includes assessing potential threats, evaluating their likelihood and impact, and developing emergency plans to diminish their effects.

#### Risk Identification:

1. **Technical Risks:** Potential challenges related to hardware and software integration, compatibility issues, and technical complexities in system design and implementation.
2. **Resource Risks:** Risks associated with resource availability, including budgetary constraints, limited access to equipment or materials, and dependencies on external suppliers or collaborators.
3. **Schedule Risks:** Risks stemming from schedule delays, unforeseen obstacles, and dependencies on project milestones or deliverables.

4. Quality Risks: Risks related to product quality, including defects, errors, and deviations from specifications, impacting performance and reliability.
5. External Risks: External factors such as changes in regulations, market conditions, or technological advancements that may affect project viability and success.

#### Risk Assessment:

For each identified risk, we assessed its likelihood and potential impact on project objectives, timeline, and budget. This involved assigning a probability rating and severity rating to determine the overall risk level and prioritize efforts accordingly.

#### Contingency Planning:

In response to identified risks, we developed dedicated emergency plans aimed at minimizing their impact and ensuring project resilience. These included alternative approaches, resource reallocation, schedule adjustments, or stakeholder communication strategies to manage expectations and mitigate negative consequences.

#### Risk Monitoring and Review:

Throughout the project lifecycle, we maintained alert oversight of identified risks, monitoring their status and effectiveness of moder measures. Regular risk reviews and updates enables us to adapt our strategies as needed, proactively addressing emerging threats and optimizing project resilience.

By actively identifying, assessing, and managing risks, we enhanced project preparedness and resilience, minimizing the likelihood and impact of potential disruptions. Our comprehensive risk management approach highlights our commitment to project success and allowed us to face uncertainties with confidence and agility.

## **8.5 Work Distribution and Team Members**

### **Ryan Kohel:**

My name is Ryan Kohel. I am graduating in the Summer 2024 semester with my Bachelor's of Science in Electrical Engineering. My academic interests are computer architecture, signal processing, and embedded systems. My contributions to the report are: Initial Hardware Diagram, Hardware Diagrams, Microcontroller Selection, IR Transmitter Selection, IR Receiver Selection, DC/DC Converter Selection, Step-Up Converter Selection, Comparing Bluetooth vs. Wifi vs. IR, Motor Driver Selection, IR

Testing, PCB Schematics, and PCB Board Layout. During Senior Design 2, I also wrote the code for the motor control, I wrote the code for receiving the infrared commands, and I wrote the code for the custom remote controller PCB.

Sanya Wadhwa:

I'm a final-year Computer Engineering student with a passion for software development, specifically focusing in Frontend/UI Design. As I embark on my senior design project, I'm excited to showcase my skills in this area. Throughout my studies, I've focused on understanding how technology interacts with user experience. In this report, my contributions to the project were: Senior Design Website, Standards, Design & Constraints, Software Diagrams, App Development (Programming and UI Design), and App Prototyping.

Laila El Banna:

I am Laila El Banna, a Computer Engineering student with a passion for software development and UI design. Throughout my academic journey, I've concentrated on understanding how technology interfaces with user experience. In our senior design project, I've taken a role in app development, focusing on creating intuitive interfaces and ensuring smooth functionality. My contributions for this report include Budget Breakdown, Project Research, Wifi/Bluetooth/SPI Standards, Design Constraints, Administrative Content, Project Milestones and Schedule, and the Conclusion.

Michael Patalano:

I am in the electrical engineering program, and I went into this course with experience in Arduino IDE and robotics. I also had some experience with 3D modeling and printing. I hope to return to school and get my masters in biomedical engineering, and pursue a career in designing prosthetics. My contributions include: Introduction, Automotive History, Breakdown of How Cars Work, Car Kit comparisons, Headlight Comparison, Handheld PCB Breakdown, Joystick, Potentiometer, Powerbank, and On/Off switch comparisons, and ChatGPT analysis.

## **9. ChatGPT and other AI/LLMs**

The best way to figure out what ChatGPT is, is to ask it.

Prompt: “What are you and how do you work?”

Answer: I'm an AI language model created by OpenAI called GPT-3.5, specifically the ChatGPT variant. Essentially, I'm a computer program designed to understand and generate human-like text based on the input I receive. I work by processing the input text and using patterns and knowledge learned from vast amounts of data to generate relevant and coherent responses. My responses are generated based on probabilities learned during my training, which involved exposure to a wide variety of text from the internet and other sources. So, in a nutshell, I'm here to converse, provide information, and assist with whatever questions or tasks you have in mind!

ChatGPT is an artificial intelligence chatbot that uses natural language processing to create human-like dialogue. The language model can respond to questions of composing various written content. It is a generative AI, so it will only give you what you put into it. A lot of its features rely on how well you can prompt the AI. It is similar to automated chat services found on consumer service websites. “GPT ” stands for “generative pre-trained transformer,” and it is trained with reinforcement learning through human feedback. One of the largest examples of this is every time you enter a prompt and ChatGPT gives you a response. It will have multiple options, one of which is a thumbs down button. When you click that button, it's telling ChatGPT that this answer was not good and it says “tell us more: don't like the style, not factually, didn't fully follow instructions, refused when it shouldn't have, being lazy, more...” If you do not press the thumbs down button, it will automatically assume that the response is not only acceptable but preferred. This is where having an educated user base comes into play, because if they can verify the response themselves, now someone who is looking to learn from a prompt, can trust exactly what they are getting in response.

ChatGPT was created and launched in November 2022, funded by a group of researchers and entrepreneurs in 2015. AI is backed by several investors, with Microsoft the most notable. OpenAI also created Dall-E, which is an AI text to art generator. ChatGPT is different because it will only give you responses in the form of text. It cannot make any digital media, only text responses.

ChatGPT is only as good as its user base, because there is only a certain amount of information that one human being can program into it. For example, if you were to write a program for a calculator, it's physically impossible to hard code every single possible mathematical equation, but if you allow your code to accept any number, it will do the calculations. In ChatGPT's case, it will only continue to get better. At release, it had its knowledge and was useful for the most basic amount of things, and it acted almost like a Google search. It would look for the highest rated websites and find the information there. Once people provided feedback letting ChatGPT that information is acceptable, it would confirm it and now that is the responses that it will give you for those prompts. The programmers could not implement every single prompt there would be, so there

were times when ChatGPT was not able to do basic math or would give you different responses than you are looking for. This is a major downside to ChatGPT. If you were asking specific industry things, it might not know what the response should be, so you as a user should confirm the information.

In conclusion, ChatGPT is an incredible resource. By simply entering in prompts, you are able to get unlimited information at your disposal. Information will only continue to get better, as more and more educated people put in prompts, and verify that the responses are acceptable. Once this happens, ChatGPT gets more reliable. If you need to do  $1+2+3+4$ , you have to do  $1+2$  first, and then you can add 3, and so on. ChatGPT and other AI/LLMs build upon themselves, and create the best user experience. Is it now able to do some extremely intricate stuff, while also guaranteeing success in the basic tasks. You can enter an entire email, ask for points of emphasis, and instead of having to go through 10 paragraphs, you could just read the cliff notes version that ChatGPT provides. Everyday tasks are infinitely easier, and advanced tasks become more achievable, due to all the information available at your fingertips.

## **10. Conclusion**

As we approach the conclusion of our in-depth exploration into the Rock 'N' Rover project's development journey, it's become increasingly evident that our achievements in engineering extend far beyond basic technical skills. They are a combination of careful planning, smooth adherence to standards, and a commitment to perfection. Throughout this path of innovation and discovery, the industry standards have paved our path, steering us towards safety, reliability, and innovation. Our journey was marked by countless moments of collaboration, problem-solving, and creativity. As we get towards the final creation of our invention, we're filled with a sense of pride and accomplishment, knowing that we'll overcome challenges and push boundaries to create something truly remarkable. This experience also feels like the beginning of a new phase filled with even more opportunities for growth, learning, and impact. We're excited to see where this journey takes us next, confident that our foundation of teamwork, dedication, and passion will continue to guide us towards greater heights of innovation and excellence.

From the project's establishment, safety has remained our prominent concern. Through the implementation of robust safety protocols, comprehensive risk assessments, and the establishment of emergency backup plans, we have underscored our undeniable dedication to preserving life, property, and the environment. This proactive stance toward safety not only diminishes potential risks but also instills a profound sense of confidence in the integrity and success of the project. Additionally, regular safety checks and ongoing training sessions ensure that everyone involved is well-prepared to handle any unforeseen challenges that may arise. By prioritizing safety at every stage of the project, we create a secure environment where innovation can thrive and all team members can work effectively towards our common goal.

In the field of electrical engineering, we take seriously the responsibility to follow industry-approved standards and conduct thorough testing with unwavering determination. By sticking to established best practices and testing our creations rigorously, we aim to create a vehicle that not only works reliably but also proves to be durable and adaptable. Every part of the Rock 'N' Rover's design has been carefully thought out to handle exploration, overcoming challenges with consistent confidence and effectiveness.

When it comes to software development, our main focus is on ensuring quality. We do this by using proven methods, encouraging compatibility, and putting a lot of effort into quality assurance. Our goal is to create software that is stable, secure, and performs well. By writing precise and efficient code, we ensure that the Rock 'N' Rover runs smoothly in any situation, demonstrating excellence in software engineering.

Effective communication has served as the essence coursing through our collaborative efforts. By establishing clear guidelines for information exchange and collaboration, we have created an environment of transparency, accountability, and innovation. Our shared protocols and interfaces have provided a comfortable ground in which creativity and efficiency can flourish, propelling us toward the height of our collective aspirations.

Moreover, our unwavering dedication to quality assurance and careful documentation standards will have laid the basis where every aspect of the project rests. Through persistent documentation and accurate testing, we will have ensured that every component of the project adheres to the highest standards of excellence, enhancing reliability, customer satisfaction, and streamlining the development process.

As we take a look into the future, The Rock 'N' Rover project is ready to make a big impact in the world of remote-controlled vehicles. We've got a strong foundation based on following the rules, doing things well, and aiming high. This sets us up to explore new ideas and inspire future engineers and innovators. As we keep pushing the limits of what we can do, our project will not only make waves in robotics but also open doors for more discoveries. By sticking to our values and encouraging creativity and curiosity, we're set to shape the future of remote-controlled vehicles and make a lasting mark on the industry.

In conclusion, the Rock 'N' Rover project shows how working together, staying determined, and being passionate can really make a difference in achieving engineering greatness. As we move forward, we're filled with excitement and hope because we know that there are so many opportunities for new ideas, discoveries, and making things better. We're ready to dive into this next phase, knowing that it's full of potential for innovation, uncovering new things, and making big changes.



# **11. Appendices**

## **11.1 Appendix A - References**

1. Fitzgerald, Weston. *Smartphone Control of RC Cars*,  
[digitalcommons.calpoly.edu/cgi/viewcontent.cgi?article=1633&context=eesp](http://digitalcommons.calpoly.edu/cgi/viewcontent.cgi?article=1633&context=eesp).
2. Rathod, Sujata, and Vinay Bansal. *Android Controlled RC Car Unit*,  
[www.researchgate.net/publication/354220199\\_Android\\_Controlled\\_RC\\_Car\\_Unit](http://www.researchgate.net/publication/354220199_Android_Controlled_RC_Car_Unit).
3. “Remote Controlled Bluetooth Vehicle.” *Kent State University*,  
[www.kent.edu/cae/remote-controlled-bluetooth-vehicle](http://www.kent.edu/cae/remote-controlled-bluetooth-vehicle).
4. Salgado-Bierman, Andres, et al. *RC Car*,  
[web.mit.edu/6.101/www/s2016/projects/nkwate\\_Project\\_Final\\_Report.pdf](http://web.mit.edu/6.101/www/s2016/projects/nkwate_Project_Final_Report.pdf).
5. *ESP8266EX Datasheet*, Espressif Systems,  
[www.espressif.com/sites/default/files/documentation/0a-esp8266ex\\_datasheet\\_en.pdf](http://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf).
6. *RP2040 Datasheet*, Raspberry Pi Ltd,  
[datasheets.raspberrypi.com/rp2040/rp2040-datasheet.pdf](http://datasheets.raspberrypi.com/rp2040/rp2040-datasheet.pdf).
7. *ATMEGA328P DATASHEET*, Atmel,  
[ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P\\_Datasheet.pdf](http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf).
8. *ESP32 Series Datasheet*, Espressif Systems,  
[www.espressif.com/sites/default/files/documentation/esp32\\_datasheet\\_en.pdf](http://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf).
9. Laukkonen, Jeremy. “Bluetooth vs. Wi-Fi: Wireless Tech in Your Home, Office, and Car.” *Lifewire*, Lifewire, 24 Feb. 2020,  
[www.lifewire.com/bluetooth-vs-wi-fi-4088218](http://www.lifewire.com/bluetooth-vs-wi-fi-4088218).
10. Ashish. “What’s The Difference Between Bluetooth And Infrared Transmission?” *Science ABC*, 19 Oct. 2023,  
[www.scienceabc.com/innovation/whats-the-difference-between-bluetooth-and-infrared-transmission.html](http://www.scienceabc.com/innovation/whats-the-difference-between-bluetooth-and-infrared-transmission.html).
11. *Hiwonder*, [www.hiwonder.com/](http://www.hiwonder.com/).
12. *Deerc*, [www.deerc.com/](http://www.deerc.com/).
13. “Understanding the Basics of Infrared Communications.” *DigiKey*, DigiKey, 6 Jan. 2021,  
[www.digikey.com/en/maker/tutorials/2021/understanding-the-basics-of-infrared-communications](http://www.digikey.com/en/maker/tutorials/2021/understanding-the-basics-of-infrared-communications).
14. *IS25LP016D IS25WP016D DATA SHEET*, ISSI,  
[www.mouser.com/datasheet/2/198/25LP\\_WP016D-1090979.pdf](http://www.mouser.com/datasheet/2/198/25LP_WP016D-1090979.pdf).
15. *LE25U40PCMC*, onsemi,  
[www.mouser.com/datasheet/2/308/1/LE25U40PCMC\\_D-1115453.pdf](http://www.mouser.com/datasheet/2/308/1/LE25U40PCMC_D-1115453.pdf).
16. *CSL1501R3T1 Data Sheet*, ROHM Semiconductor,  
[fscdn.rohm.com/en/products/databook/datasheet/opto/optical\\_sensor/infrared\\_led/csl1501rw-e.pdf](http://fscdn.rohm.com/en/products/databook/datasheet/opto/optical_sensor/infrared_led/csl1501rw-e.pdf).
17. *VSMB10940*, Vishay Semiconductors,  
[www.vishay.com/docs/84170/vsmb10940.pdf](http://www.vishay.com/docs/84170/vsmb10940.pdf).
18. *IN-S126ESGHIR*, Inolux,  
[www.mouser.com/datasheet/2/180/IN\\_S126ESGHIR\\_V1\\_1-2323377.pdf](http://www.mouser.com/datasheet/2/180/IN_S126ESGHIR_V1_1-2323377.pdf).

19. *TSOP22.., TSOP24.., TSOP48.., TSOP44.. IR Receiver Modules for Remote Control Systems*, Vishay Semiconductors, [www.vishay.com/docs/82459/tsop48.pdf](http://www.vishay.com/docs/82459/tsop48.pdf).
20. *TSOP382.., TSOP384.. IR Receiver Modules for Remote Control Systems*, Vishay Semiconductors, [www.vishay.com/docs/82491/tsop382.pdf](http://www.vishay.com/docs/82491/tsop382.pdf).
21. *GP1UM26RK0VF/GP1UM27RK0VF Series*, SHARP, [www.mouser.com/datasheet/2/365/gp1um26rk0vf\\_e-336717.pdf](http://www.mouser.com/datasheet/2/365/gp1um26rk0vf_e-336717.pdf).
22. *Technical Data Sheet Infrared Remote-Control Receiver Module*, Everlight Electronics Co., LTD., [www.mouser.com/datasheet/2/143/Everlight%20IRM-26xxA-1166026.pdf](http://www.mouser.com/datasheet/2/143/Everlight%20IRM-26xxA-1166026.pdf).
23. *DRV8835 Dual Low-Voltage H-Bridge IC*, Texas Instruments, [www.ti.com/lit/ds/symlink/drv8835.pdf](http://www.ti.com/lit/ds/symlink/drv8835.pdf).
24. *STSPIN948 Datasheet*, STMicroelectronics, [www.st.com/resource/en/datasheet/stspin948.pdf](http://www.st.com/resource/en/datasheet/stspin948.pdf).
25. *MTS2916A Data Sheet*, Microchip Technology, [www.mouser.com/datasheet/2/268/MCHPS04303\\_1-2520611.pdf](http://www.mouser.com/datasheet/2/268/MCHPS04303_1-2520611.pdf).
26. *V78E-1000-SMT*, CUI Inc., [mouser.com/datasheet/2/670/v78e\\_1000\\_smt-1480599.pdf](http://mouser.com/datasheet/2/670/v78e_1000_smt-1480599.pdf).
27. *R-78AA-1.0*, RECOM Power, [www.mouser.com/datasheet/2/468/R-78AA-1.0-1710061.pdf](http://www.mouser.com/datasheet/2/468/R-78AA-1.0-1710061.pdf).
28. *TSR 1SM Series, 1 A*, Traco Power, [www.mouser.com/datasheet/2/687/tsr1sm\\_datasheet-3049707.pdf](http://www.mouser.com/datasheet/2/687/tsr1sm_datasheet-3049707.pdf).
29. *EPM78Vx Non-Isolated DC-DC Converter*, Eaton, [www.mouser.com/datasheet/2/87/eaton\\_epm78vx\\_non\\_isolated\\_dc\\_dc\\_converter\\_data\\_sh-1923416.pdf](http://www.mouser.com/datasheet/2/87/eaton_epm78vx_non_isolated_dc_dc_converter_data_sh-1923416.pdf).
30. *VXO78-500-M*, CUI Inc., [www.mouser.com/datasheet/2/670/vxo78\\_500\\_m-2932771.pdf](http://www.mouser.com/datasheet/2/670/vxo78_500_m-2932771.pdf).
31. *R-78AA-0.5*, RECOM Power, [www.mouser.com/datasheet/2/468/R-78-0.5-1710058.pdf](http://www.mouser.com/datasheet/2/468/R-78-0.5-1710058.pdf).
32. *MP3120*, Monolithic Power Systems, [www.mouser.com/datasheet/2/277/MP3120-2945726.pdf](http://www.mouser.com/datasheet/2/277/MP3120-2945726.pdf).
33. *MP3414*, Monolithic Power Systems, [www.mouser.com/datasheet/2/277/MP3414-2945896.pdf](http://www.mouser.com/datasheet/2/277/MP3414-2945896.pdf).
34. *MP3221*, Monolithic Power Systems, [www.mouser.com/datasheet/2/277/MP3221-2945789.pdf](http://www.mouser.com/datasheet/2/277/MP3221-2945789.pdf).
35. Hetler, A. (2023, December 15). What is chatgpt? everything you need to know. *WhatIs*. TechTarget. Retrieved April 22, 2024, from <https://www.techtarget.com/whatis/definition/ChatGPT>
36. How does an electric motor work? (DC motor). (2020, June 10). *YouTube*. YouTube. Retrieved April 22, 2024, from [https://www.youtube.com/watch?v=CWulQ1ZSE3c&ab\\_channel=JaredOwen](https://www.youtube.com/watch?v=CWulQ1ZSE3c&ab_channel=JaredOwen)
37. Brain, M., & Hall-Geisler, K. (2000, April 1). How Electric Motors work. *HowStuffWorks*. HowStuffWorks. Retrieved April 22, 2024, from <https://electronics.howstuffworks.com/motor.htm>
38. Electric motor. (n.d.). *Electric motor - Energy Education*. Retrieved April 22, 2024, from [https://energyeducation.ca/encyclopedia/Electric\\_motor](https://energyeducation.ca/encyclopedia/Electric_motor)

39. Electric motor. (n.d.). *Electric motor - Energy Education*. Retrieved April 22, 2024, from [https://energyeducation.ca/encyclopedia/Electric\\_motor](https://energyeducation.ca/encyclopedia/Electric_motor)
40. Car anatomy: The basics / how cars work? (3D animation). (2022, October 10). *YouTube*. Retrieved April 22, 2024, from [https://www.youtube.com/watch?v=fPjOWekzeGI&ab\\_channel=CARinfo3d%28En%29](https://www.youtube.com/watch?v=fPjOWekzeGI&ab_channel=CARinfo3d%28En%29)
41. English online. (n.d.). *Cars | How cars work | Types of cars | History of the automobile | Safety features | The world's biggest carmakers*. Retrieved April 22, 2024, from <https://www.english-online.at/travel/cars/cars-and-how-they-work.htm#:~:text=Most%20automobiles%20are%20powered%20by,pistons%20slide%20up%20and%20down>.
42. Electric, T. B. (2023, July 21). Types of motors in electric vehicles. *Blog - Biliti Electric*. Retrieved April 22, 2024, from <https://bilitielectric.com/blog/types-of-motors-in-electric-vehicles/>
43. Types of car engines: List of different types of car engines. (2023, June 20). *Spinny Car Magazine*. Retrieved April 22, 2024, from <https://www.spinny.com/blog/index.php/types-of-car-engines/>
44. Car engine types: List of different types of car engines. (2024, January 17). *Acko General Insurance*. Retrieved April 22, 2024, from <https://www.acko.com/car-guide/car-engine-types/>
45. Group, M.-B. (n.d.). Benz Patent Motor Car: The first automobile (1885–1886): Mercedes-Benz Group. *Mercedes*. Retrieved April 22, 2024, from <https://group.mercedes-benz.com/company/tradition/company-history/1885-1886.html#:~:text=On%20January%2029%2C%201886%2C%20Carl,birth%20certificate%20of%20the%20automobile>
46. (N.d.). *COM-09032 Sparkfun Electronics | switches | DigiKey*. Retrieved April 23, 2024, from <https://www.digikey.com/en/products/detail/sparkfun-electronics/COM-09032/6823623>
47. Automobile history. (n.d.). *History.com*. A&E Television Networks. Retrieved April 22, 2024, from <https://www.history.com/topics/inventions/automobiles>
48. About The Author Amy Follow Us Position: Columnist Having writing articles about computer tech for a long time, & Us, A. F. (2023, July 19). PS4 controller buttons: Names/layout/functions [full guide] - Minitool Partition Wizard. *MiniTool*. Retrieved April 22, 2024, from <https://www.partitionwizard.com/partitionmagic/ps4-controller-buttons.html>
49. Arduino esplora. (n.d.). *docs.arduino.cc*. Retrieved April 22, 2024, from <https://docs.arduino.cc/retired/boards/arduino-esplora/>
50. RCCarAction.com. (2022, January 20). Get started in RC - everything you need to know. *RC Car Action*. Retrieved April 22, 2024, from <https://www.rccaraction.com/get-started-rc-everything-need-know/>
51. Ossum, & Instructables. (2020, November 14). Scratch build an RC car with CAD and rapid prototyping. *Instructables*. Retrieved April 22, 2024, from <https://instructables.com/Scratch-Build-an-RC-Car-With-CAD-and-Rapid-Prototy/>

52. Ossum, & Instructables. (2020, November 14). Scratch build an RC car with CAD and rapid prototyping. *Instructables*. Instructables. Retrieved April 22, 2024, from <https://instructables.com/Scratch-Build-an-RC-Car-With-CAD-and-Rapid-Prototy/>
53. What are Radio Control (RC) cars? (n.d.). *What are Radio Control (RC) Cars? - HPI Racing*. Retrieved April 22, 2024, from <https://www.hpiracing.com/en/beginners-guide>
54. Radio-controlled car. (2024, March 19). *Wikipedia*. Wikimedia Foundation. Retrieved April 22, 2024, from [https://en.wikipedia.org/wiki/Radio-controlled\\_car](https://en.wikipedia.org/wiki/Radio-controlled_car)