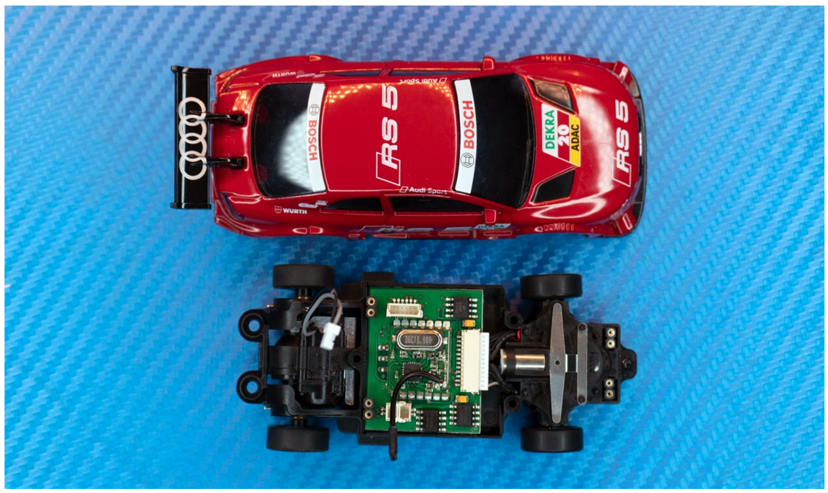
UCF

RACE: Remote Access Control for Electric Vehicles

Internet-Connected RC Car Racing Experience



Group 30

Authors:
Sean Chaney - Computer Engineering
Jesvany Colon - Electrical Engineering
Michael Mallette - Electrical Engineering
Austin Silva - Electrical Engineering

Review Committee:
Dr. Mike Borowczak (ECE Faculty)

Mr. Mark Maddox (ECE Faculty)

Dr. Suboh Suboh (ECE Faculty)

Donor:
Josh Jeinkins

Chapter 2 - Project Description

2.1 Project Background and Motivation

Remote-controlled (RC) cars have long been a popular hobby amongst children and enthusiasts who enjoy both the technical aspects of building and modifying the vehicles and the thrill of competing in races. As simple as an RC car may seem, it is responsible for very fond memories for many individuals. Unfortunately, traditional RC car racing is limited by physical proximity, requiring participants to be in the same location. This project will help to alleviate this issue and help to withhold the joy of RC racing that many have held dearly.

Originating from a heavy interest in robotics and gaming, we wanted to create a project that is not only entertaining but also interactive amongst people across the world. Going into the brainstorming for the project we knew we wanted to implement a car in some way, as a member had been doing some prior research on implementing cars and internet connectivity. As avid gamers, we also felt like creating a competitive aspect where users can race head to head or against time.

The motivation behind this project is to combine the excitement of RC car racing with the accessibility and interactivity of online gaming, creating a dynamic and engaging platform where users can race against each other remotely. By incorporating elements such as live streaming, we aim to bring a video-game-like experience to the real world, offering an innovative, entertaining, and technically challenging product for hobbyists and competitive racers alike. While the world is moving in a more virtual path, this project is a fun medium that brings the physical and virtual world together.

2.2 Current Technologies and Existing Products

There are a few current technologies and products that are similar to our idea such as, “Anki Overdrive”, “Vrombr”, and FPV Drone Racing.

2.2.1 Anki Overdrive

“Anki Overdrive” is a popular app-controlled racing system that featured AI-driven cars, customizable tracks, and various game modes with powerups, similar to Mario Kart. Players could control their cars using a smartphone app, but it was limited to local play rather than internet-connected racing.

Strengths:

- ❖ **Advanced AI Technology**: The cars are powered by sophisticated AI, allowing for autonomous racing with dynamic behaviors such as learning the track layout and competing against players intelligently.
- ❖ **Customizable Track**: It comes with magnetic, snap-together track pieces, enabling players to build a wide variety of track designs.

- ❖ **Mobile App Integration**: Players control the cars using a mobile app (compatible with both Android and iOS).
- ❖ **Multiplayer:** The game supports multiple players. Users can race against friends or AI-controlled cars.

Weaknesses:

- ❖ **Cost**: The starter kits and individual add-ons (such as new track pieces or cars) are relatively expensive compared to other toy-based racing games.
- ❖ **App Dependency**: The game relies heavily on its mobile app, which can be limiting if the app experiences glitches or compatibility issues with certain devices.
- ❖ **Battery Life of Cars**: Players may find that they can only race for short periods before needing to recharge the vehicles.
- ❖ **Limited Availability and Support**: Anki, the company behind Overdrive, ceased operations in 2019, so the product has seen limited support.

The strengths of Anki Overdrive are very advanced and worth pulling ideas from, considering its release date into the market. However, it being a now abandoned product shows that those strengths did not hold up well for users over extended time. The weaknesses of the Anki Overdrive will be essential to address, especially cost and the battery life of our IC cars. These weaknesses will be our strengths and ultimately our selling point, as we are not using advanced technology such as AI powered self driving cars or magnetic customizable tracks.

2.2.2 Vrombr

"Vrombr" is an augmented reality (AR) racing game that allows players to control physical remote-controlled cars through their smartphones. The unique feature of Vrombr is the integration of real-life racing with virtual elements, where the cars are equipped with cameras, giving players a driver's perspective as they compete in various game modes.

Strengths:

- ❖ **Augmented Reality Integration**: Combines real-world toy cars with digital elements like race modes, virtual obstacles, and in-game stats.
- ❖ **Real-Time Camera Feeds**: Cars are equipped with real-time cameras, offering a first-person view (FPV) during races.
- ❖ **Advanced Sensors**: Cars feature collision detection and sensors that can monitor the performance of the car in real-time.
- ❖ **5G Support**:Vrombr is designed to enable low-latency control, which is critical for fast-paced racing games.

Weaknesses:

- **Physical Limitation**: Since the cars are physical devices, any crashes, flips, or malfunctions require manual intervention.
- **Cost and Accessibility**: Reliance on physical cars could become expensive. Users must pay to upgrade their car.
- **Input Lag and Connection Issues**: Players mentioned the need for consistent, fast internet connectivity, which may limit its accessibility for some
- **Durability of Physical Components**: The cars, being 1:10 scale replicas with electronics, are prone to wear and tear, especially if used frequently in competitive environments.

"Vrombr" relates the most to our project, so we expect to run into similar issues and also pull inspiration from their successes. The weaknesses we want to address in our project, that is present in Vrombr, is the cost and accessibility, and the input lag and connection issues. We believe these are essential to mitigate in order to create a product that users will come back to our product for. If it is seemingly unplayable or overpriced it will drive users away from the product.

2.2.3 Remocon

"Remocon" was a platform allowing people to set up robots and let users control them, often for a price. While the site is still up, no robots have been online.

Strengths:

- **Expandable Platform**: Their platform was ever expanding, with users getting inspired to make their own robots.
- **Real-Time Camera Feeds**: Robots are equipped with real-time cameras, offering a view to control the specified robot

Weaknesses:

- **Physical Limitation**: Since the robots are physical, any crashes, flips, or malfunctions require manual intervention.
- **Input Lag and Connection Issues**: The latency varied between robots, some having delays of over 1 second.
- **No Activity**: While robots are listed, no robots seem to have been online within the previous month

2.2.4 Surrogate.TV

"Surrogate.TV" was an internet platform allowing people to set up small FPV cars and let users race them. It has since been discontinued for over 2 years.

Strengths:

- **Expandable Platform**: Their platform was ever expanding, with users getting inspired to make their own cars, races and tracks.

- **Real-Time Camera Feeds**: Tracks were set up with an overhead camera instead of FPV cameras. This allowed for a cheaper setup for many car races and saved bandwidth.
- **Waypoints:** The overhead camera tracked each car, ensuring that if a car went backwards into the finish line, they would not win the race.
- **Automatic Charging:** The owners setup their own track that featured nearly hands free maintenance with the cars being able to charge themselves using the overhead camera navigation.

Weaknesses:

- **RF signal instead of WIFI integration**: The owners, for their own racetracks and cars, used already existing RF model cars. They wired each controller to their server, allowing users to control each car. This gave them many problems in the long run with controllers connecting to the wrong cars.
- **Discontinued**: Surrogate.TV has since been discontinued

2.2.5 FPV Drone Racing

FPV (First-Person View) drone racing is a competitive sport where pilots control small, fast drones equipped with cameras that provide a live video feed directly to the pilot's goggles or screen. This setup gives pilots the sensation of flying from the drone's perspective, as if they were inside the cockpit. FPV drones are designed for speed, agility, and precise maneuvering, allowing them to race through challenging courses at high speeds.

Strengths:

- **Real-Time Video Transmission**: FPV drone racing uses real-time video feeds transmitted from the drone to the pilot's goggles. This gives an immersive experience, allowing precise control and quick reaction times in high-speed environments.
- **High-Performance Components**: Racing drones use lightweight frames and powerful motors to achieve high speeds.
- **Customizability**: FPV drones are highly customizable, allowing racers to build their own rigs based on personal preferences. This includes selecting different motors, flight controllers, cameras, and video transmitters to match their flying style and the race conditions.
- **Advanced Control Systems**: These drones utilize advanced flight controllers with programmable settings, stabilization, and GPS-assisted navigation. This gives pilots the ability to fine-tune their drone's flight characteristics, making it easier to adapt to different tracks and environments.

Weaknesses:

- **Signal Interference and Latency**: FPV drone racing often operates on 5.8 GHz analog signals, which are prone to interference from Wi-Fi, other drones, or even weather conditions. Signal drops, video static, or latency issues can disrupt the pilot's view, leading to crashes or poor race performance.

- **Battery Limitations**: Racing drones rely on LiPo (Lithium Polymer) batteries, which offer high power but limited flight time, typically between 3 to 7 minutes.
- **Complex Setup and Maintenance**: FPV drone racing involves assembling and maintaining drones, which can be complex for beginners. Pilots need a good understanding of electronics, soldering, and programming flight controllers.
- **Cost**: High-quality FPV drones and components such as goggles, transmitters, cameras, and batteries can be expensive. Pilots also will usually need backup parts and repair tools due to the frequent risk of crashes.

2.2.4 Summary

Of all these products, FPV drone racing has flourished the most as it has become a competitive sport and a rather pricey hobby. The main strength we can take away is the quality of components and high speed live streaming feed. The high stakes environment built around drone racing is something that we plan to take from in order to make sure our project gives the users a competitive spirit when racing our IC cars. Though in our case it will be different, as FPV drones use radio signals or digital FPV systems, which don't require Wi-Fi for video transmission. Due to different technologies, the weaknesses found with FPV drone racing is rather not comparable to our project as battery life will last longer than utilizing a LiPo battery, cost will be significantly lower and setup will be rather simple and not in the hands of the users.

Though these products did not inspire our project, they do hold significance for our research and future production of our project. Their strengths and weaknesses will act as a guide for us to follow, making sure to emphasize on the strengths and improve upon them and to avoid weaknesses by executing our set specifications.

2.3 Goals and Objectives

2.3.1 Goals

- Create an immersive Internet-Controlled (IC) racing experience.
- Provide real-time video streaming with a first-person view (FPV).
- Lap and time tracking for accurate race outcomes.
- Post race times on online leaderboards.

The overall goal of this project is to create an accessible and immersive Internet Controlled (IC) racing experience for users to race head to head or against time. This experience relies on the ability to maintain low-latency, real-time communication between users and their cars, ensuring that inputs are quickly reflected in the vehicle's movement to guarantee a smooth racing simulation. The challenge lies in optimizing this control responsiveness while minimizing delays caused by network transmission. We will need to create a robust server infrastructure and protocols like WebSockets or MQTT to facilitate smooth data exchange. The user interface (UI) should be intuitive, offering simple and effective controls for steering, throttle, and braking.

To enhance traditional RC car racing, the platform will include real-time video streaming, providing users with a first-person view (FPV) from the car, making the experience more immersive and engaging. The integration of FPV will allow users to feel as though they are in the driver's seat, offering a more realistic racing experience. Achieving this requires low-latency video streaming to ensure that the camera feed keeps pace with the real-time control of the car. WebRTC or similar streaming protocols would be ideal for this purpose. Proper camera placement and stability are essential to simulate the driver's viewpoint, giving an accurate representation of the car's environment. Additionally, considerations like bandwidth management become important when multiple users are streaming simultaneously.

Ensuring reliable lap tracking and fair race outcomes is another goal. A system will be implemented to automatically track lap times using sensors or markers at the start/finish line. Using IR sensors, RFID tags, or even a vision-based system could help achieve accurate lap counting. The system must distinguish between forward and backward movements to prevent false counts or errors when a car crosses the line repeatedly.

Posting users' race times on an online leaderboard is essential for curating a competitive spirit among participants. For this, a server-side database is needed to store race times and rankings, allowing users to see their position in real-time. The leaderboard system must update dynamically and accurately reflect each racer's performance.

2.3.2 Stretch Goals

One stretch goal we would like to implement in our project, if time permits, is a powerup system. For instance, after a specific amount of time (5-10 seconds) each car will get a random powerup (ex. boost self, slow other, damage others steering). The front of each car would have an infrared LED, and the back of each car would have an infrared receiver. The driver of a car can 'fire' their powerup. Powerups used on self, get instantly applied. For powerups that affect other cars, a hit is determined by IR. Due to its heavy dependance on software, with only 1 CpE member, we believe achieving this may be daunting.

Another stretch goal we are considering is creating a charging pit lane. This would be a system that allows a car to drive into a pit lane, get locked in place by pit lane, and start getting charged via contact port. There are various ways we can input this feature; we can have cars automatically drive into a pit lane by the mechanical and software system of the track and pit lane. The car can drive along the wall of the track, leading it into pit lane, then into the first available charging port. This avoids mapping and clunky components. Lastly, we could also add photonic sensors on the side to locate the wall of the track.

2.3.3 Objectives

- ❖ Design 2 car models, using motors, servo motors and microcontrollers.
- ❖ Implement onboard cameras for real-time video streaming and an immersive user experience.
- ❖ Develop an automated system for accurate lap tracking and race timing.

- ❖ Design a user-friendly interface for remote car control and live video access.
- ❖ Prioritize ease of use for an enjoyable and accessible user experience.
- ❖ Optimize system performance for low latency and multiple users.

Designing two car models will involve integration of motors, servo motors, and microcontrollers to ensure they operate smoothly and can be controlled remotely. Each car will need a motor to drive the wheels and a servo motor for steering. Selecting the right type of motors will be crucial, as balancing speed, torque, and power efficiency to match the racing experience we want to create. The microcontroller acts as the brain of the car, interpreting remote commands and controlling the motors accordingly. The microcontroller will also have Internet connection capabilities to assure that the cars can remotely be controlled via Internet.

The cars will be equipped with onboard cameras capable of streaming high-quality video to the user's device, offering an immersive FPV experience. The real-time video feed will be crucial in providing users with a view from the car's perspective during the race. The cameras need to be lightweight and capable of transmitting a steady video feed with minimal delay. This involves not only selecting the appropriate camera but also integrating it with the car's microcontroller and the server infrastructure. The camera should be securely mounted at a position that mimics the driver's perspective, giving users an accurate and engaging view of the race. To ensure smooth video transmission, we'll need to use protocols that prioritize low-latency streaming, and possibly incorporate adaptive bitrate streaming to adjust video quality based on network conditions.

Another major objective is the development of a precise lap tracking system. Technologies such as RFID or infrared sensors will be researched to automate lap counting and timing laps at the finish line, helping to determine the race's outcome. We are going to create a PCB to house the sensors and other components needed to track the lap count and race timing. This data would then have to be communicated in real-time to the server. This component will be set at the start/finish line in order to accurately count lap and time progress in the race, which will help determine the winner.

A user-friendly interface will be created, allowing players to control their cars via a web browser or mobile app. Alongside the controls, users should have access to the live video feed from their car's camera, ideally displayed in a way that is easy to navigate and view even during fast-paced racing. The user experience will be key in ensuring that players can easily navigate the system and enjoy the racing experience.

Lastly, testing and optimization will be crucial to ensure that the platform performs reliably under real-world conditions. The system will undergo rigorous testing to reduce latency and ensure that it can handle multiple users racing at the same time without performance degradation. Network latency can significantly affect both car control and video streaming, so the system needs to be optimized to minimize delays in both areas. Using efficient communication protocols, a high-performance server, and possibly even edge computing solutions could help ensure that the system remains responsive even when multiple users are racing simultaneously.

2.3.4 Project's Features and Functionalities:

- Internet-Based Remote Control: Users can control IC cars from anywhere in the world via an internet connection. The system utilizes a web-based interface, allowing players to send commands to the cars with minimal latency.
- First-Person View (FPV) Camera System: Each car is equipped with a real-time camera that streams video back to the user, providing a first-person view (FPV) as though they were inside the vehicle.
- Multi-User Support with Scalable Server Architecture: The system is designed to allow multiple users to race simultaneously. The server architecture is built to scale, ensuring smooth performance even with several concurrent users.
- Web-Based User Interface: Users will interact with the system through a web browser interface. The interface will provide control inputs, live video feed, and race stats.
- Lap Tracking System: A sensor will be implemented to automatically track laps. This ensures accurate race results and eliminates the need for manual lap counting.
- Custom Printed Circuit Board (PCB) for Communication: The car will feature a custom PCB that integrates a microcontroller (such as an ESP32 or Raspberry Pi) responsible for receiving commands from the server and executing them on the car's motors, steering, and sensors.

2.4 Engineering Specifications and Requirements

Table 2.1. *Table of Required Specifications*

Specifications	
Car Size	≤5"x8"
Car Control Delay	≤1.5 seconds
Car Chassis Maintainability	Chassis lasts for 100 runtime hours
Battery Life	≥10 minutes
Video FPS	20≤x≤40 FPS
Video Resolution	1280x720
Distance From Server/Router	10≤x≤30 ft
Total Cost of 2 Cars and Finish Line	≤$250
Finish Line Time Accuracy	±0.1 second

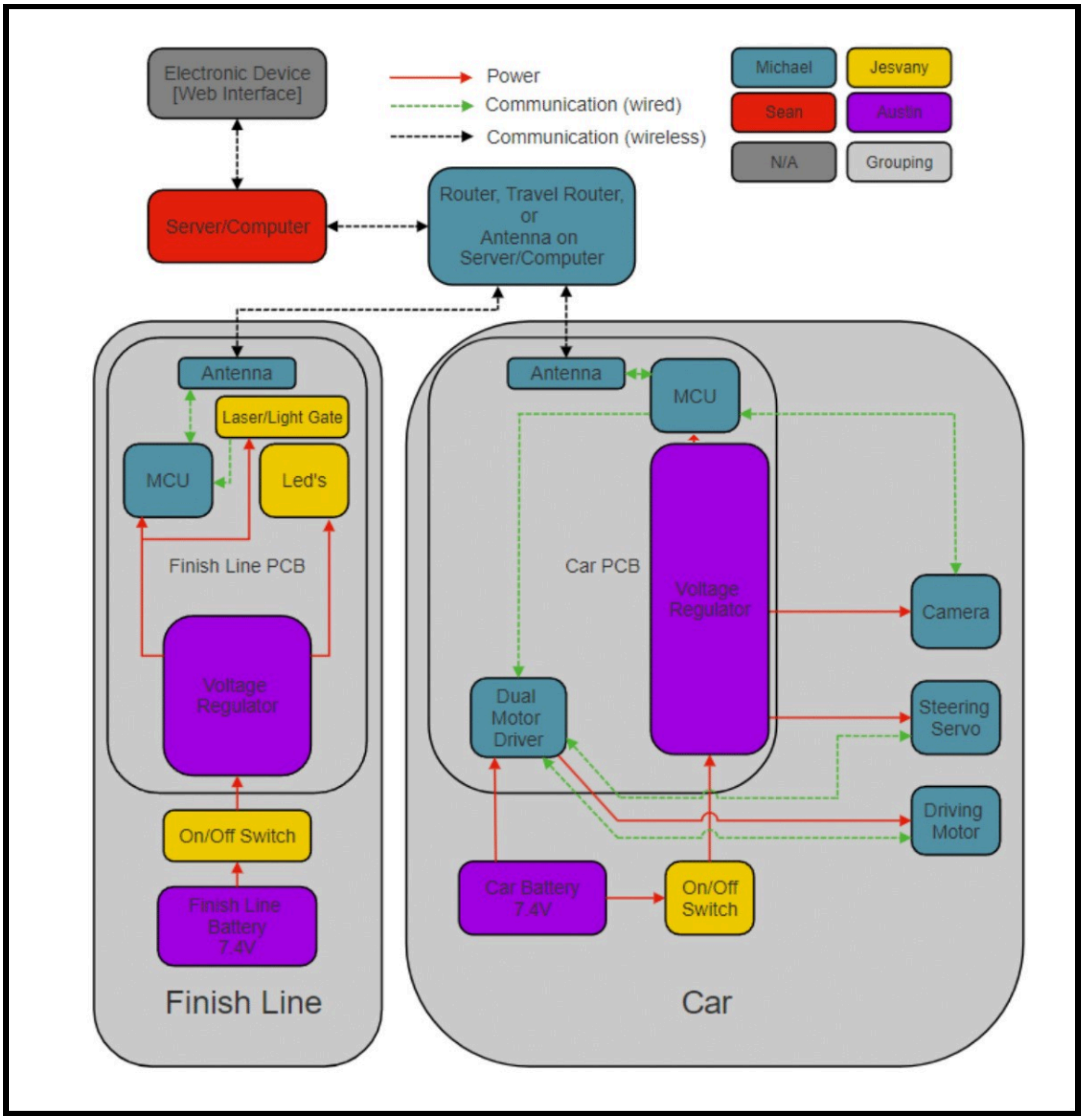


Figure 2.1. *Hardware Diagram by Authors*

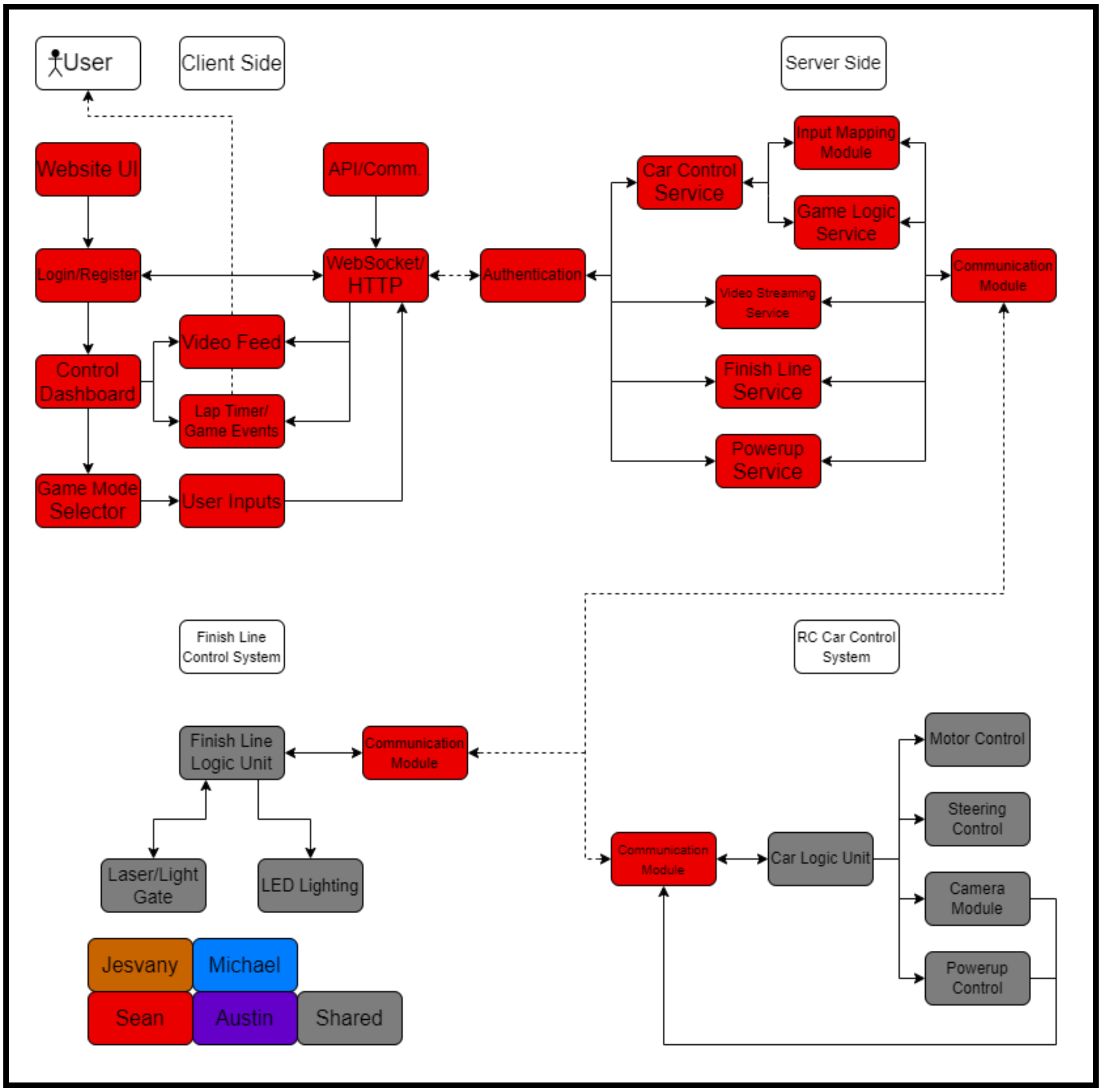


Figure 2.2. Signals Block Diagram by Authors

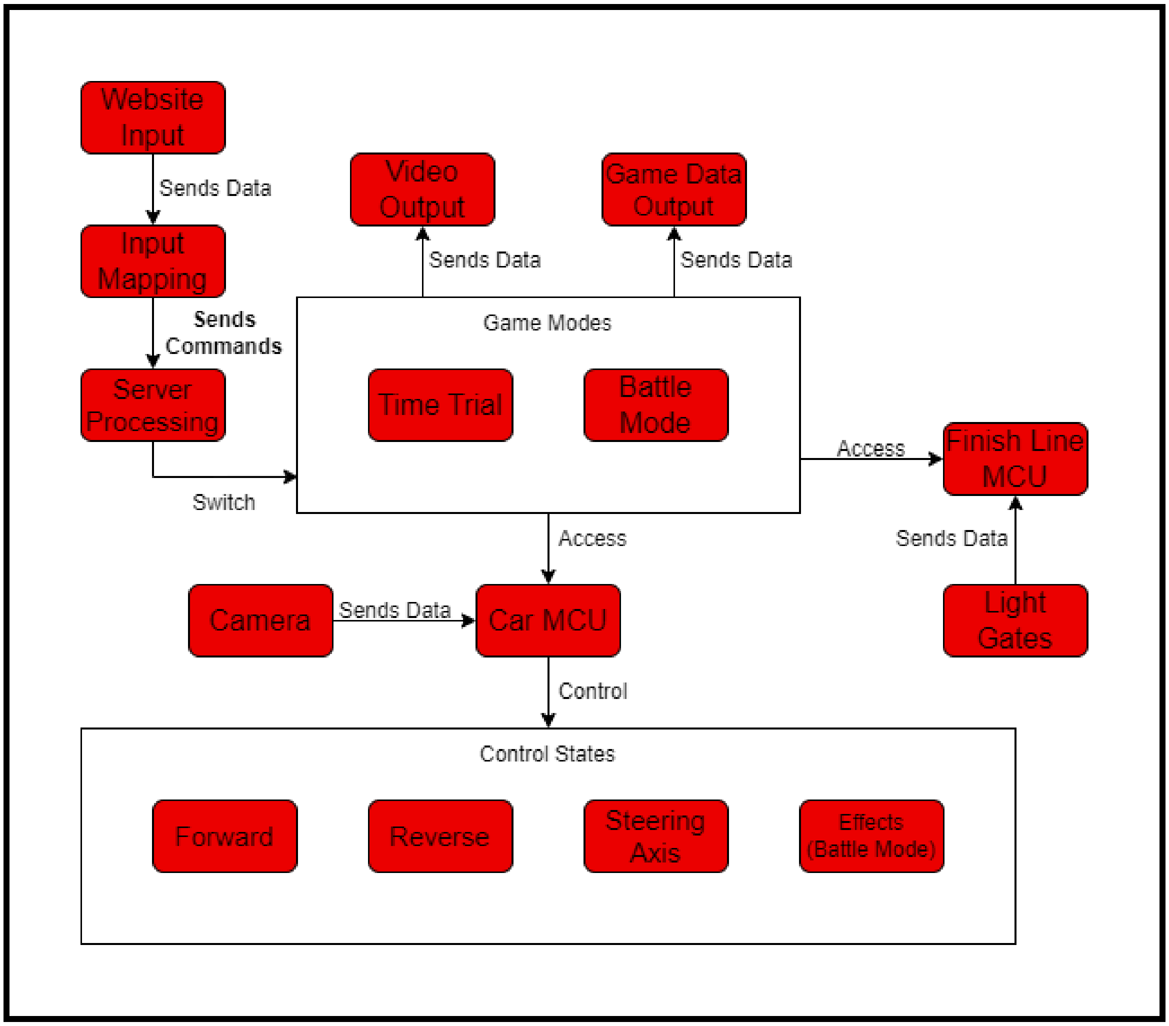


Figure 2.3. *Software Flowchart by Authors*

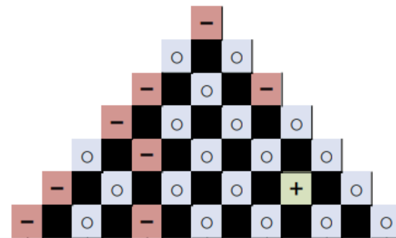
2.5 House of Quality

The House of Quality is a resourceful visual tool that enables us to effectively compare the engineering and marketing requirements of our project. It showcases the correlations and tradeoffs between the requirements, providing insight into the potential effects that would result from making changes to our project.

Correlations	
Positive	+
Negative	−
No Correlation	○

Relationships	
Strong	↑
No Relation	○
Weak	↓

Direction of Desirability	
Maximize	▲
Minimize	▼



Column #		1	2	3	4	5	6	7	8
Engineering Requirements / Customer Requirements (Explicit and Implicit)	Direction of Desirability	Total Cost of 2 Cars and Finish Line	Car Size	Car Control Delay	Distance From Computer/Router	Battery Life	Finish Line Time Accuracy	Car Chassis Maintainability	Video FPS
Direction of Desirability		▼	▼	▼	▲	▲	▲	▲	▲
Affordability	▲	↑	↑	○	○	↓	○	○	○
Video Quality	▲	↓	○	○	○	○	○	○	↑
Communication Speed	▲	○	○	↑	○	○	↑	○	↑
Communication Range	▲	↓	○	↓	↑	○	↑	○	↑
Battery Life	▲	↓	↓	○	○	↑	○	↑	○
Portability	▲	↑	↑	○	○	○	○	○	○
Ease of Mantainence	▲	↑	↑	○	○	↑	○	↑	○
Targets		Maximum $250	Maximum 5"x8"	Maximum1.5 seconds	Range of 10-30 ft	At least 10 minutes	Within 0.1 second	Chassis lasts for 100 runtime hours minimum	Range of 20-40 FPS

Figure 2.4. *House of Quality by Authors*

Chapter 10 - Administrative Content

10.1 Initial Budget and Cost Estimates

Our project has received a donation of $200 from a donor who is interested in our project’s internet-controlled car driving feature. The rest of our budget will be evenly funded by our group members.

We have decided to set the project’s budget at a maximum of $350, encouraging us to effectively minimize the cost where applicable and create space for multiple iterations, providing plenty of flexibility and spare funds if we need to purchase part replacements or exchange parts. Thus, we divided the project into three parts (subsystems), visualized below in Table 10.1, allocating our budget accordingly.

Table 10.1. *Budget Allocation for Project Parts*

Subsystem	Budget
Cars	$150
Finish Line	$150
Server	$50

Table 10.2. *Initial Cost Estimates for Each Component*

Component Description	Quantity	Estimated Cost
3D Printed Car Chassis	~500g of filament	$12
Driving Motor	2	$6
Steering Servo	2	$12
Camera	2	$2
3D Printed Finish Line Shell	~600g of filament	$15
Colored LED’s	~12	$3
Laser Sensor	1	$2
PCB’s & Switches	2	$80
Batteries	4	$30
Combined Total Cost		$162

Seen from the table above, Table 10.2, we did initial cost estimates for all of our project's components, and the total estimated cost is well below our initial budget. As previously mentioned, this will help us go through the project's iterative design process in a smooth manner, allowing for more margin of error and relatively low financial pressure.

10.2 Work Distribution

Listed below is the worktable distribution for each team member, showcasing our individual responsibilities and assigned tasks for our project.

Table 10.3. *Distribution of Work*

Team Member	Key Responsibilities
Sean Chaney	Software Design & Implementation
	Website Design & Management
	Server Management
	Communications & Antenna
Michael Mallette	Car Design & Construction
	Project Lead
	Software Design & Implementation
	MCU & Motors Implementation
	Communications & Antenna
Jesvany Colon	Finish Line PCB Design
	Administrative Management
	Finish Line Design & Construction
	Laser/Light Gate Selection & Implementation
Austin Silva	Car PCB Design
	Team Lead & Project Management
	Support Car Design & Construction
	Support Finish Line Design & Construction
	Power Supply Selection & Implementation

While each team member does have their own responsibilities, this is a heavily collaborative project, where every team member will be working together and have input on the project's subsystems in a substantial way. As such, each team member will be supporting each other's work through verification of design and implementation components by all team members as we progress throughout the various stages of the project. Overall, ensuring that everyone is involved in the process and that all of our work will be effectively incorporated into the subsystems.

10.3 Project Timeline and Milestones

Listed below are two tables that showcase a general timeline that we plan to follow for continual project progress throughout Senior Design 1 & 2. These tables do also include descriptions of the steps we intend to take on a weekly basis to reach our project's milestones.

Table 10.4. *Senior Design 1 Timeline*

Senior Design 1			
Week(s)	**Dates**	**Task**	**Description**
1	08/19/24 - 08/23/24	Group Formation	Forming a well-balanced team
2	08/26/24 - 08/30/24	Brainstorm Ideas	Sharing project ideas and determining the project scope
3	09/02/24 - 09/06/24	D&C Document and Review Committee	Writing Divide & Conquer document and forming a review committee
4	09/09/24 - 09/13/24	D&C Group Meeting and Project Approval	D&C meeting with advisor, once project is approved, begin researching
5-8	09/16/24 - 10/11/24	Project Research	Researching and selecting the technology and parts for the project, as well as defining project's constraints
9	10/14/24 - 10/18/24	Reaching 60-Page Report Milestone	Finish writing final report content for Chapters 2-5 and Chapter 10, designing system, and ordering parts
10	10/21/24-10/25/24	Submit 60-Page Report	Submitting 60-Page progress report on time
11	10/28/24 - 11/01/24	60-Page Group Meeting and Progress Update	60-Page report meeting with advisor, continue writing final report (should be at the 90-page mark), and system testing
12-13	11/04/24 - 11/15/24	120-Page Report	Completing the 120-Page final report,

		Draft and Mini Demo Video	continued system testing and breadboard prototyping, creating mini demo video, and designing PCB
14-15	11/18/24 - 11/26/24	Final Report Submission	Reviewing the final report and submitting report on time, updating team website, and ordering PCB

Table 10.5. *Senior Design 2 Timeline*

Senior Design 2			
Week(s)	**Dates**	**Task**	**Description**
1-3	01/06/25 - 01/24/25	PCB Testing	Receive PCB and test functionality to ensure PCB works to specifications
4-6	01/27/25 - 02/14/25	System Testing and Integration	Constructing/assembling system prototype and testing prototype's functionality
7-8	02/17/25 - 02/28/24	Corrective Measures	Researching and applying corrective solutions to any mistakes or errors encountered
9-10	03/03/25 - 03/14/25	Optimize Prototype	Optimizing and completing the prototype
11-14	03/17-25 - 04/11/25	Project Demo Preparation	Completing all SD2 requirements before final presentation and demonstration
15-16	04/14/25 - TBD	Final Presentation and Demonstration	Presenting and demonstrating prototype to SD coordinators and review committee

For us to remain on schedule and effectively work as a team, we have decided to conduct weekly in-person team meetings to discuss project progress in the areas we are working on each week. We are also utilizing a project management tool called Trello, which provides us with the ability to manage a kanban board for our project, allowing us to effectively assign tasks to each other and visualize our project's progression as we complete our tasks.

References:

ChatGPT, "Conversation with an AI assistant," OpenAI ChatGPT, Sep. 4, 2024. [Online]. Available: https://chat.openai.com/.

Surrogate.tv, "Driving A Tiny RC Car Over The Internet Using ESP8266," Make: DIY Projects and Ideas for Makers, Sep. 12, 2019. https://makezine.com/article/maker-news/driving-a-tiny-rc-car-over-the-internet-using-esp8266/ (accessed Sep. 2, 2024).

"Remocon.tv - Remote control an internet of robots," Remocon.tv, 2020. https://remocon.tv/ (accessed Sep. 2, 2024).