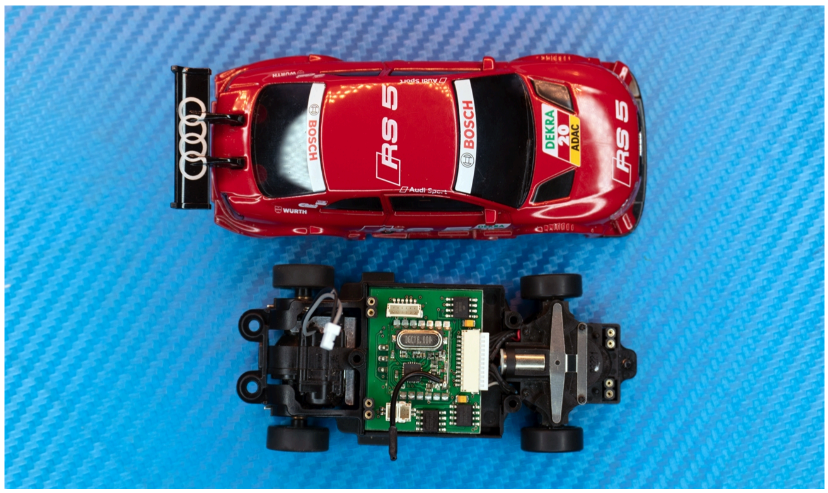
RACE: Remote Access Control for Electric Vehicles

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

Table of Contents

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”, “Remocon”, “Surrogate.TV”, and FPV Drone Racing.

2.2.1 Anki Overdrive

“Anki Overdrive” was 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 set up 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 race tracks 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.
- **Advances in Flight 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.
- ❖ **Legal and Regulatory Challenges**: Pilots must be aware of local aviation laws and restrictions, which may limit where and how they can race.

2.2.6 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 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

There are several goals and expectation that must be met through the progression of our project to ensure a quality product, this includes:

- ❖ Create an immersive Internet-Controlled (IC) car 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

To guarantee our overall goal of a quality product, we must achieve numerous objectives that we've culminated, such as:

- ❖ Design 2 car models, using driving motors, steering motors, and microcontrollers.
- ❖ Implement FPV 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 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:

To fully realize our IC car racing system, we determined this list of critical features and functionalities for this project:

- ❖ 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 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	≥15 minutes

Video FPS	20-40 FPS
Video Resolution	426x240
Connectivity Distance	10-30 ft
Total Cost of 2 Cars and Finish Line	≤$250
Finish Line Time Accuracy	±0.1 second

2.5 System Diagrams

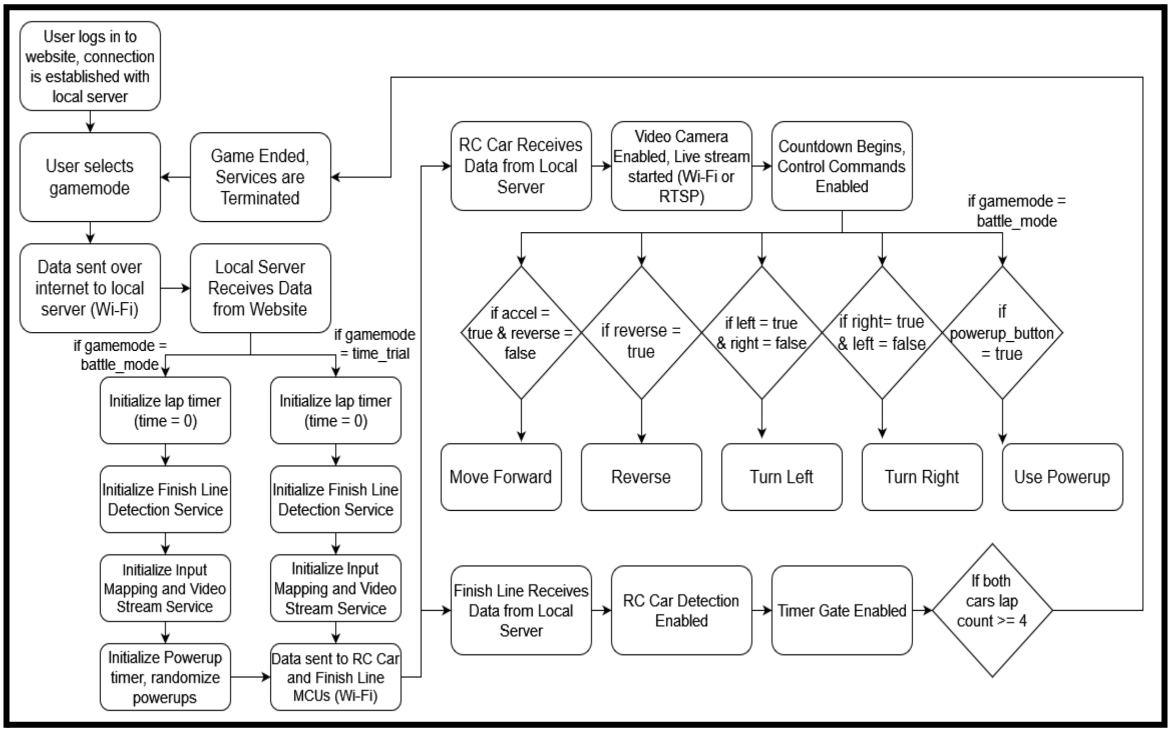


Figure 2.1. *Software Flowchart by Author*

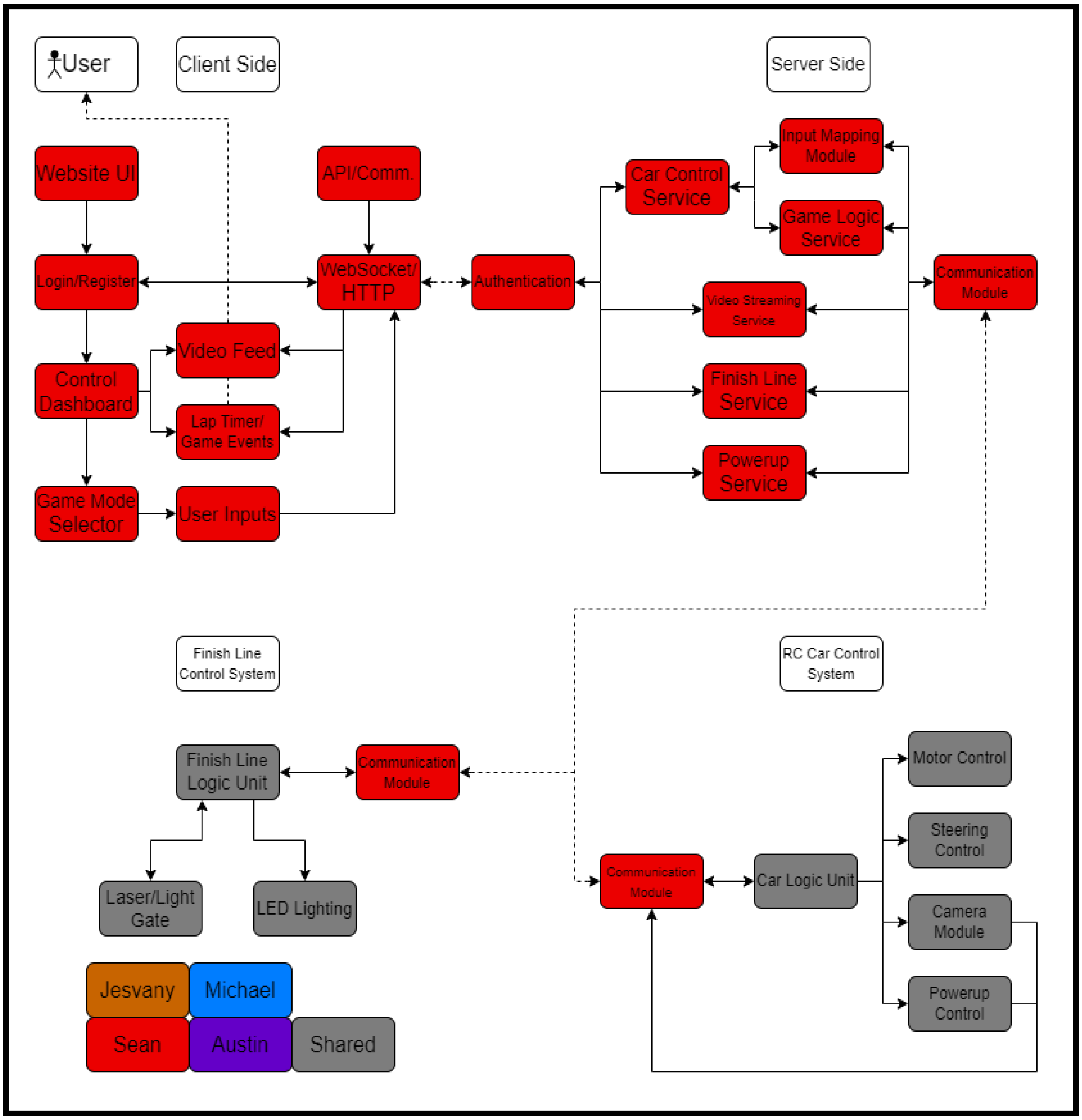


Figure 2.2. *Signals Block Diagram by Authors*

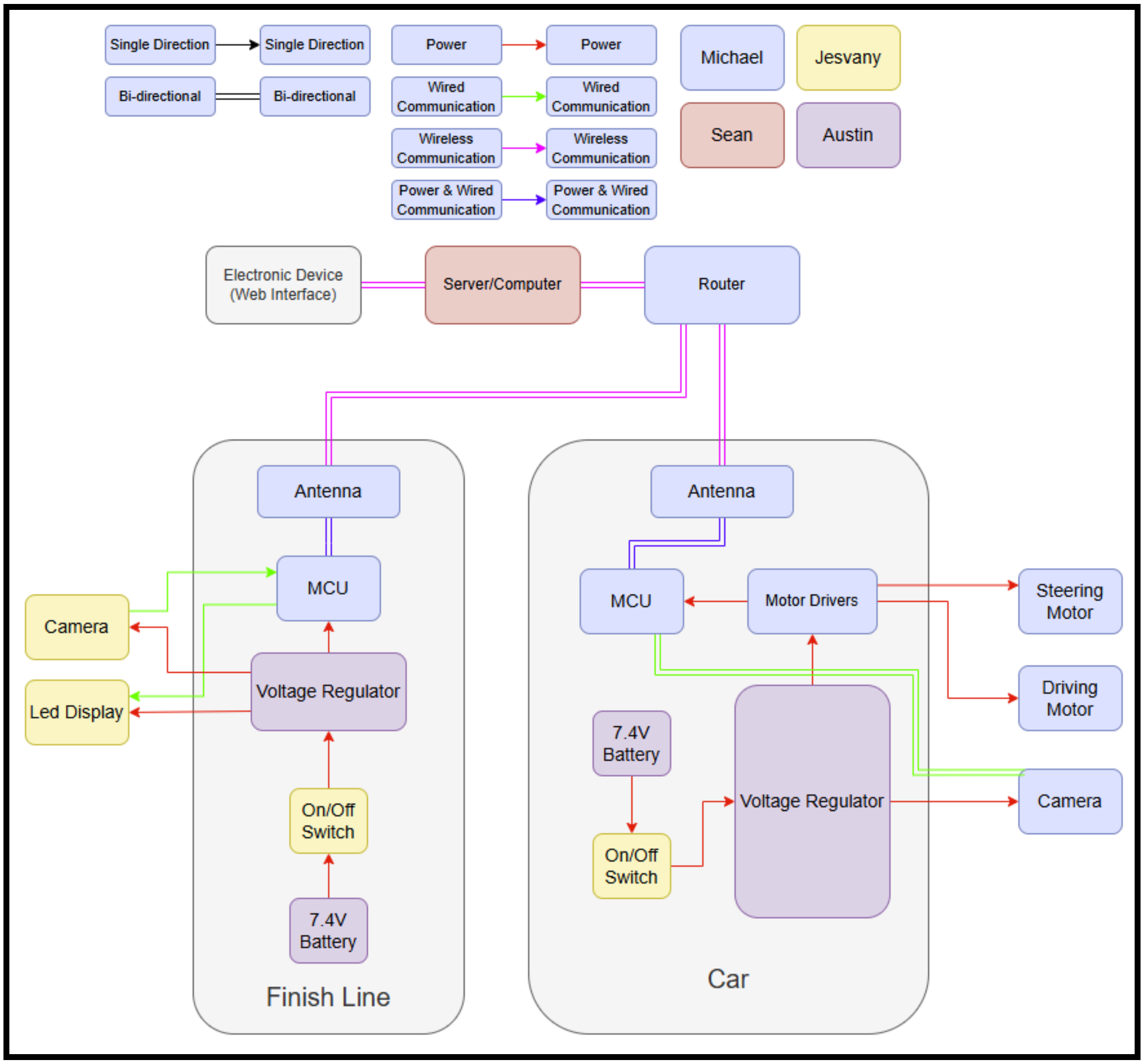


Figure 2.3. *Hardware Diagram by Authors*

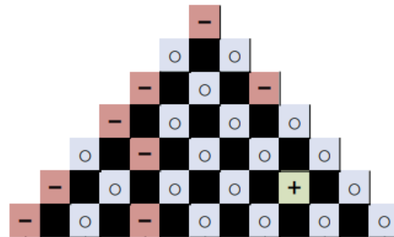
2.6 House of Quality

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 3 - Research and Investigation

3.1 Hardware Components Overview

This section is an introduction and general overview of the hardware components that will be used in this project, explaining their purpose and function in the system we are creating. With this project utilizing many hardware components, each component must be thoroughly researched to determine the most optimal choice of technology and part selection to best match our project needs. Through this research process, it is imperative that we ensure all of the selected hardware parts not only be appropriately selected for their specific role in our system, but that all selected parts be able to effectively communicate, connect, and work with one another.

This project will require ten major hardware components, each serving a purpose within the overall internet-controlled car racing system that we are creating. Listed below are the ten hardware components with an overview of their role and purpose in this project, then each component will be further researched and investigated in the following sections of this chapter.

Microcontroller Unit (MCU): The MCU will serve as the brain of the IC cars we're designing. Each car will be outfitted with a MCU, which will act as the control center for all the car's operations such as sending video transmission and receiving motor control inputs. A MCU will be needed for the finish line system as well, being in charge of coordinating the selected camera/sensor's action for recording real-time race data as well as the LED sequence for the start of races.

Antennas: The antennas will serve a crucial role in this project as the means to transmit and receive signal wirelessly. We will be setting up a Wi-Fi router near the racetrack, having our cars and finish line system be equipped with compatible antennas will be a must for our devices to effectively communicate with the router.

RC Car Model: A working RC car model will definitely be needed for this project. Due to various constraints within this Senior Design project, we have opted to buy a RC car model and fully customize it instead of attempting to build a RC car model from the ground up.

Driving Motors: Within the RC car model, driving motors are used to enable the car to go forward and backwards, located at the rear set of wheels.

Steering Motors: Within the RC car model, steering motors are used to enable the car to turn left and right, located at the front set of wheels.

Motor Driver: The motor driver is an essential component for controlling the speed and direction of the motors used in the RC car. In this project, the motor driver acts as an intermediary between the microcontroller and the driving and steering motors. It receives control signals from the MCU and translates them into the appropriate voltage and current levels needed to power the motors effectively. The motor driver enables forward and reverse motion for the driving motors and directional control for the steering motor.

Cars Cameras: To fully implement an immersive FPV car racing system, an effective camera that can transmit live video feed will be a necessity. This camera will need to be carefully selected to be able to communicate with the router and to meet the video quality specifications we have for this project.

Finish Line Camera/Sensor: To fully implement any racing system, let alone a car racing system, some form of race/lap tracking is needed. For this project specifically, the finish line system will implement a camera or sensor that will be used to effectively capture/record data whenever a car crosses the finish line. Thus, by using this hardware component, it will provide the means to develop the system for accurate lap tracking and race timing.

Finish Line LEDs: For every car racing system out there, there is always some form of countdown sequence to signal the start of the race. With many applications using a light sequence to represent the race's countdown, we decided to incorporate a matrix of LEDs into our finish line system that would then be used to signal the countdown sequence for whenever a race is starting.

Power Supply Unit (PSU): The power supply unit is another critical component in this project, due to it being in charge of effectively meeting the power needs of all the other hardware components. The PSU will consist of an effective battery supply and needed voltage regulation for both the cars and the finish line system. In that, the PSU will be acting as a power flow channel, supplying and distributing power throughout both systems.

3.1.1 RC Car Product Comparisons and Selection

For this project, we originally intended to build a RC car model and fully outfit it according to our project's needs. We did intensive research and investigation into the RC car design process, learning the mechanical aspects of building a RC car as well as the electrical aspects of it, such as how to properly tune the driving and steering motors for optimal performance as an example.

During that research process though, we started to really recognize the challenges we'd encounter and the constraints of this project if we were to build our own RC car model. Specifically, the time constraints and manufacturability constraints of Senior Design were the two critical factors we recognized as major challenges.

With these constraints in mind, we decided to shift our attention to the possibility of purchasing an already built RC car model and then customizing it to our project's needs. We first did a comprehensive search into the different kinds of RC cars in the market, becoming more educated on the various features, specifications, and build of each of these cars types. We then factored everything into a thorough decision-making process, determining the kind of car that could best fit our project's racing system.

On-Road Car Racing

For the car racing system we are creating, we knew that we'd be making an indoor or "on-road" racetrack to set up the IC car races. In the world of RC car racing, specifying whether the racing application will be on-road or off-road is critical to finding the right car for the specified racing environment. Thus, any off-road RC cars such as buggies, short-course trucks, crawlers, or monster trucks were automatically disqualified as an option for this project.

Specifically for on-road RC cars, there is a wide variety of cars available, anywhere from micro-sized RC cars for tight indoor racing, all the way up to specialized drifting and Formula 1 RC cars for high intensity racing. However, with our project's scope, we only considered options with appropriate features of size, speed, and control, while also keeping price in mind. From that we were able to narrow our options down to basic Ready-to-Run (RTR) RC cars, Mini & Micro RC cars, and RC Toy-Grade cars.

RTR RC Cars

Ready-to-Run RC cars are RC car kits that typically come with a fully assembled RC car, transmitter (remote controller), battery supply, and battery charger. The RTR feature applies to many different makes and models of RC cars, models for both on-road and off-road purposes, with there being a range of speed and sizes available. We looked into quite a few basic and beginner level models, but we ultimately determined that these RTR car kits include too many features and add-ons that would simply take away from the engineering design process of this project.

Mini & Micro RC Cars

Mini and micro RC cars are smaller sized RC cars specifically designed for indoor racing in tight spaces. When doing our preliminary investigation for this project, looking at existing products that are similar to our project, we came across a few products that utilized mini and micro RC cars for their car racing systems, so we already knew that this RC car type could be a serious option for our project.

The RC car scaling system is a ratio representation of how the model car's size compares to the size of the actual vehicle. For example, a 1:12 scale RC car model of a ford mustang (mustangs are approximately 15.3 ft in length) would equate to the RC car version being around 15.24 inches (1.275 ft). For these small scale cars, Mini RC cars are typically between 1:18 to 1:12 in scale, and Micro RC cars take downsizing to another level, typically coming in at 1:28 scale or even smaller.

Figure 3.x. *RC Car Scaling Comparisons*

With these small scale cars, we found that they offer overall good size for the indoor racing system we are intending to build. These cars also possess the speed and control characteristics that we are looking for as well, excelling in maneuverability and handling with fair levels of speed. Going back to our engineering specifications table, we then hammered in our search for car models that would fit the ≤5”x8” specification we made for the size of our car. With that specification, the typical mini RC car models are actually too big, whereas the micro RC car models do fit the specification.

Before locking in a car product selection of a micro RC car model, we took a step back and factored in the customization process we’d be undertaking to fully realize the IC racing cars we’re looking to build. From that, we recognized that the majority of these micro RC car models would actually be too small for us to effectively customize them in the way we’d intend to. So, we realized that we needed to find a car model right in-between the size of a mini RC car and a micro RC car, somewhere around 1:24 scale.

RC Toy-Grade Cars

After going through that process of determining the ideal size of the cars, and becoming more educated on the limited speed and control features that our project needs, since many of the products we came across were simply too fast and had too many extra features our project didn’t need. Thus, we decided to look into RC toy-grade cars, which easily provide the base speed and control features we need while also giving us the basic setup we need to fully customize the car model.

Once we gathered all of our information needed, we had one of our teammates go to a local retailing store and browse the commercially available RC toy cars available to see if there could be a good match. He found a model at a low price that matched our specifications in size at 1:24 scale, with fair speed and simple features.

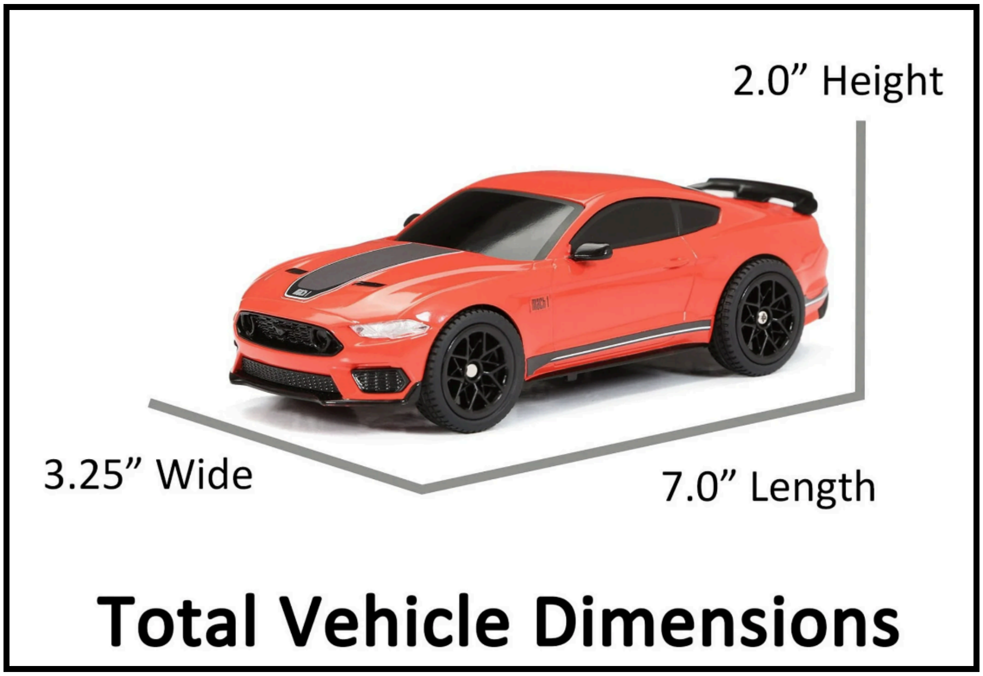


Figure 3.x. *Adventure Force’s Ford Mustang RC Car Dimensions*

After purchasing the model, we then had a look inside the car to see what the motors setup of the car is like, also determining how much space we might have to use for our customizations. Following that up we performed some elementary tests on the motors, assessing them to ensure that this RC car model will be sufficient for the racing system we’ll be creating. Ultimately, we determined that this RC car model will work for our project, and we obtained two: Adventure Force (1:24) Ford Mustang Mach 1 Battery Remote Control Red Sports Cars, 2423-13R.

Table 3.x. *List of Official Product Information*

Features	Details
Scale/Size	1:24 Scale, 3.25x7.0x2.0 inches
Controls	Left-Right Steering Forward-Reverse Driving
Remote Controller	2.4 GHz Radio Transmitter
Battery Supply	3 AA Batteries
Design	Classic Ford Mustang Mach 1, Red
Tires	Rubber

3.2 MCU

3.2.1 MCU Technology Comparisons and Selection

MCU Overview

A microcontroller (MCU) is a small, integrated circuit that is designed to carry out specific tasks within an embedded system. It combines a processor, memory, and input/output (I/O) peripherals on a single chip, making it a powerful solution for handling control tasks in electronic devices. Microcontrollers are used to interface with hardware components such as sensors, motors, and other peripherals, allowing for automated, real-time processing and communication.

In our project, microcontrollers play a crucial role in managing key functionalities. They will be responsible for controlling the cars and the finish line, ensuring seamless, real-time communication between the system components and the server. This will allow for immediate control inputs from the server and real-time video feedback from the cameras in each car and at the finish line.

We are using three microcontrollers in total: one in each car and one at the finish line. The microcontroller in each car will handle two primary functions:

1. It will communicate with the server by sending live video from the camera and receiving control inputs to drive the motors.
2. The finish line microcontroller will also communicate with the server but with slightly different tasks—it will either send video footage or process and transmit the race time data if video processing is handled locally. Additionally, the finish line microcontroller will manage the start LEDs, ensuring synchronization with race start commands.

When choosing the right microcontroller, several factors need to be evaluated: size, Wi-Fi connectivity, operating voltage, memory capacity, the number of cores, and SRAM size. To streamline the project and reduce complexity, we will be using the same microcontroller for both the cars and the finish line, ensuring consistency and ease of use in development and troubleshooting.

Next, we will review potential development boards (DEV boards) that we can use for our prototype. These boards allow us to test the microcontroller's capabilities in a simplified and accessible environment before final integration into the project.

ESP32

The ESP32, developed by ESPRESSIF, is the first microcontroller we are considering for our project. It features a dual-core processor that can run at up to 240 MHz, but it also has the flexibility to reduce the clock speed to as low as 80 MHz when lower power consumption is required. This feature is particularly useful in our project, as it allows us to optimize performance while minimizing power usage during periods of inactivity or reduced demand.

The ESP32 is highly versatile, supporting both Wi-Fi and Bluetooth (including Bluetooth Low Energy or BLE). This dual-connectivity makes it ideal for projects that require both local device-to-device communication and internet-based data transfer. The operating voltage of the ESP32 is 3.3V, which is standard for many embedded systems. It also has 520 KB of SRAM, offering ample memory for managing multiple tasks simultaneously, such as handling the camera feed, receiving control inputs, and processing commands from the server.

The ESP32 also offers a wide range of GPIO pins, making it suitable for projects requiring multiple peripheral connections. Overall, the ESP32 is a strong candidate for our project due to its powerful dual-core architecture, connectivity options, and energy efficiency.

ESP8266

The ESP8266 is another microcontroller from ESPRESSIF and offers a lower-cost alternative to the ESP32. While it shares some similarities with the ESP32, the ESP8266 is generally less powerful. It features a single-core processor running at up to 160 MHz, which, while sufficient for many embedded tasks, falls short of the performance offered by the dual-core ESP32.

The ESP8266 operates at 3.3V, the same as the ESP32, but it lacks Bluetooth connectivity. This limitation makes it less versatile for applications that require both Wi-Fi and Bluetooth. Additionally, the ESP8266 consumes more power than the ESP32, especially during operations that require consistent processing, making it less efficient for battery-powered applications.

In terms of memory, the ESP8266 has 160 KB of SRAM, significantly less than the ESP32, which can limit its ability to handle multiple tasks at once, such as camera processing or managing multiple peripherals. The number of GPIO pins is also fewer on the ESP8266, making it less suitable for projects that need to interface with many components.

However, the main advantage of the ESP8266 is its cost-effectiveness. It is a cheaper option than the ESP32 and is still capable of handling many basic embedded tasks, making it an option worth considering for simpler applications where cost is a priority.

Raspberry Pi Pico

The Raspberry Pi Pico, based on the RP2040 chip, is another option we are considering for our project. This microcontroller features a dual-core ARM Cortex-M0+ processor, which runs at up to 133 MHz. While this clock speed is lower than that of the ESP32, the Raspberry Pi Pico is still suitable for a wide variety of embedded tasks, especially those that do not require intensive processing.

One notable limitation of the Raspberry Pi Pico is the lack of built-in Wi-Fi or Bluetooth connectivity. In order to enable wireless communication, an additional module would be required, which adds complexity to the project. Despite this, the Pico excels in projects where GPIO usage is the primary requirement, offering a wealth of well-supported libraries and tools to interface with external sensors, motors, and other peripherals.

The Raspberry Pi Pico operates at 3.3V and has 264 KB of SRAM, which is less than the ESP32 but sufficient for many simpler applications. It is also an affordable choice, making it attractive for projects where cost is a key factor and wireless communication is not essential.

Arduino Nano

The Arduino Nano is another microcontroller under consideration, though it presents several limitations compared to more modern options like the ESP32 or ESP8266. The Nano does not come with built-in Wi-Fi or Bluetooth capabilities, which means that an external Wi-Fi module would be necessary to provide wireless communication in our project. This adds extra hardware complexity and may increase power consumption.

The Arduino Nano runs a single-core ATmega328P processor at a much slower clock speed of 16 MHz, significantly slower than the ESP32 or even the ESP8266. The operating voltage of the Nano is 5V, which is higher than the 3.3V required by most modern embedded systems and may require additional voltage regulation when interfacing with certain components.

The Nano provides 14 digital I/O pins and 6 analog inputs, which is fewer than what the ESP32 offers. Additionally, it has only 2 KB of SRAM, which severely limits its ability to manage memory-intensive tasks like camera feeds or multiple sensor inputs. While the Arduino Nano is a well-supported and widely used development board, it is not ideal for our project due to its lack of built-in wireless capabilities and lower processing power.

3.2.2 MCU Part Comparisons and Selection

Table 3.x. *MCU Comparison Table*

Specification	ESP32	ESP8266	Raspberry Pi Pico	Arduino Nano
Microcontroller	Espressif (Tensilica Xtensa LX6)	Espressif (Tensilica Xtensa L106)	RP2040	ATmega328
Number of Cores	2	1	2	1
Operating Voltage	3.3V	3.3V	3.3V	5V

Clock Speed	240 MHz	80 MHz	133 MHz	16 MHz
Flash Memory	4MB	4MB	2MB	32 KB
SRAM	520 KB	64 KB	264 KB	2 KB
Wi-Fi	Yes	Yes	No	No
Bluetooth	Yes (v4.2)	No	No	No
GPIO Pins	30	17	26	14
PWM Pins	16	8	16	6
ADC Channels	18	1	3	8
Power Consumption	Moderate to Low	Low	Low	Low
Cost (Amazon)	6	8	10	6

Focusing on minimizing delay, we chose the ESP32 as the primary board for our project. This decision was straightforward because, out of all the microcontroller boards we considered, the ESP32 stands out as both cost-effective and high-performing.

One of the key factors is its powerful dual-core processor, which, while more than sufficient for our current requirements, also provides room for future scalability. The ESP32 also boasts a compact form factor, making it easy to fit within the physical constraints of our design.

Despite its low cost, it delivers impressive performance with a sizable amount of flash memory and RAM, giving us more than enough resources for current tasks while leaving plenty of room for future additions or enhancements.

3.3 Motors

3.3.1 Motors Technology Comparisons and Assessment

As previously mentioned in the overview section, we decided to buy an already built RC car and customize it to our project's needs. We came to this conclusion after intensive research and investigation into RC car design, determining that it would not be feasible for us to build a working model on our own with this project's constraints.

After making the decision to buy a RC car model, we knew that the model would need to have an effective driving motor and steering motor setup to match our project's scope. Taking that into consideration, we researched the various kinds of motor technology, investigating which is the best fit for RC car applications. We then used that research to assist in the RC car selection process, testing the car's driving motor and steering motor specifications to make sure the performance level would be a good match for our project's scope.

3.3.1.1 Motor Types

DC Motor

DC motors are simple, cost-effective, and commonly used in small-scale RC cars. They operate by converting electrical energy into mechanical motion through the interaction of a current-carrying conductor and a magnetic field. In this project, a DC motor would allow for direct control of the car's speed and direction using Pulse Width Modulation (PWM) signals from the microcontroller. This type of motor provides adequate torque and speed for small cars but lacks precise control, which may make it less ideal for applications requiring fine-tuned adjustments. The main advantages of DC motors include their simplicity, ease of control, and affordability, but they may experience wear over time due to their use of brushes. In regards to steering, a DC motor would not allow for fine control, but instead a binary control system. The three options are all the way left, all the way right, or center (centering is not due to the motor going into a centered state, instead the steering axis has a spring to naturally go back to the center. This is achieved by turning off the steering motor).

BLDC Motor

Brushless DC (BLDC) motors offer several advantages over traditional DC motors, such as higher efficiency, longer lifespan, and better torque-to-weight ratio, making them an attractive option for this project. These motors operate without brushes, reducing wear and increasing reliability. Additionally, BLDC motors are controlled using electronic speed controllers (ESCs), which allow for more precise control of speed and torque compared to brushed DC motors. For our project, BLDC motors could improve energy efficiency, prolong battery life, and enhance performance when fine control over movement is required. However, they tend to be more expensive and complex to implement due to the need for ESCs and more advanced control algorithms.

Stepper Motor

Stepper motors offer high-precision control over position, which could be useful for applications requiring incremental movement, such as robotics or systems requiring exact positioning. They function by moving in discrete steps, which are determined by the number of magnetic coils and the input signals sent to the motor. While stepper motors are known for their precision, they are not ideal for applications that require continuous rotation at varying speeds, like driving a small RC car. In this project, stepper motors would be less efficient compared to DC or BLDC motors due to their higher power consumption and relatively lower speed. They are more commonly used for applications that prioritize positioning over speed, making them less suitable for this specific application.

Servo

Servo motors are specialized motors that provide high precision in control over angular or linear positioning, making them a common choice for steering mechanisms in RC car applications. Servo motors typically have a greater torque compared to a DC motor from gear ratios, but move much slower. Servo motors operate by receiving a Pulse Width Modulation (PWM) signal, which determines the angle of rotation. The motor then adjusts its position until the feedback signal matches the input command.

Table 3.x. *Motor Types Comparison Table*

Motor Type	Advantages	Disadvantages	Best Application
DC Motor	Simple, cost-effective, easy to control, adequate torque and speed for small RC cars	Limited precise control, wear over time due to brushes, binary steering control	Basic driving and speed control in small RC cars
BLDC Motor	High efficiency, longer lifespan, better torque-to-weight ratio, precise control with ESCs	More expensive, complex implementation with need for ESCs and advanced algorithms	Applications needing precise and efficient control with better performance and longevity
Stepper Motor	High-precision positioning, suitable for applications requiring incremental movement	Not suitable for continuous rotation at varying speeds, high power consumption, lower speed	Robotics or systems needing exact positioning, not suitable for continuous driving
Servo	High precision control over position, suitable for steering, good torque due to gear ratios	Slower movement, not suitable for continuous high-speed operation	Steering mechanisms needing precise angular control

3.3.1.2 Motor Assessment

The RC car utilized in our project incorporates two main types of motors: a driving motor and a steering motor. These motors are integral to the car's overall functionality and performance.

Driving Motor Overview

The driving motors are a critical component of our project, responsible for moving the cars forwards and backwards. Selecting the appropriate motors requires careful consideration of various factors to ensure they can handle the load and not move the car too fast or too slow. The motors will need to provide sufficient torque to move the cars, fit within the size constraints of the vehicle, and operate at our cars operating voltage.

Driving Motor Assessment

The driving motor used in the RC car we selected is a DC motor. The operating voltage is 0.5V-3V. We found that the car goes too fast at 3V and is a good speed at 1-1.5V. The motor has a measured max current consumption of 1.5A. We will be using PWM on this motor via our motor driver in order to fine tune the speed and ensure a fair race between the 2 cars.

Steering Motor Assessment

The steering motor used in the RC car we selected is also a DC motor. The RC car gives this steering motor 4.12V. We found this to be a suitable voltage for the strong torque of the steering, allowing the car to steer when stationary. The measured current consumption of this motor is 0.5A.

3.3.1.3 Motor Drivers

Motor drivers are crucial for managing the driving and steering functions of RC cars, converting low-power control signals from the microcontroller into high-power outputs needed by the motors. Our motor driver selection is specifically based on the specifications of the motors provided with the RC cars. These specifications, including voltage and current requirements, help ensure that the chosen motor driver can adequately power the motors and handle their performance demands, whether controlling one motor or multiple simultaneously.

Motor Driver Selection

Table 3.x. *Comparison Table of Motor Drivers*

Motor Driver	Voltage Range	Current Handling	Features	Pros	Cons
L298H	4.5V - 46V	Up to 2A per channel	Dual H-bridge, simple and robust, suitable for basic applications	Widely available, simple to use, handles higher voltages	Large size, less efficient due to older technology, heat dissipation issues
DRV8871	6.5V - 45V	3.6A peak, 2A continuous	Single-channel H-bridge, efficient, built-in	High current handling, good for higher	Single channel, may need two for dual

			protection features	power motors, reliable protection	motor control
DRV8833	2.7V - 10.8V	Up to 1.5A per channel	Dual H-bridge, compact, efficient, capable of driving two motors	Compact size, efficient power usage, suitable for low voltage motors	Lower current handling compared to others, limited to lower voltage applications

We chose the DRV883 motor driver because it operates within the voltage and current specifications required by our motors, ensuring reliable and safe performance. Additionally, its dual motor control capability allows us to efficiently manage both the driving and steering motors using a single module, which simplifies our design and reduces the need for extra components. This combination of compatibility and functionality makes the DRV883 an ideal choice for our project.

3.4 Cameras

3.4.1 Camera Technology Comparisons and Selection

Cars System Camera Overview

With our project differing from traditional RC cars, using internet-controlled features instead of remote-controlled features, the effective implementation of a camera for our cars will be paramount to our users' ability to race these cars in real-time with accurate video transmission directly from the cars POV (point of view). Upon initial investigation into camera technologies used in RC cars and drone racing applications, the most common camera technology used is general FPV (first person view) cameras, which are specifically designed for low latency and a wide field of view, while being both compact in size and lightweight. The other applicable camera technology found is onboard action cameras, HD video recording cameras that are primarily used for high-quality performance in remote (in-person) races and to record high-quality video footage of races.

FPV Cameras

FPV camera technology is an integral component to remote-controlled vehicles racing. This technology allows users to experience real-time video feed directly from the perspective of their drone or RC car, providing that immersive FPV experience for all users. The key features of all FPV camera technology are low latency, wide FOV (field of view), and compact size, which makes them an ideal option for RC racing. For the specifications and technological aspects of these cameras, the major components include the image sensor type, the resolution, the frame rate, video transmission type, and potential OSD capabilities (on screen display).

The two main image sensors used for these cameras are CCD (charge-coupled device) and CMOS (complementary metal-oxide semiconductor) sensors. CCD sensors generally produce higher-quality images and have better low-light performance than CMOS sensors. However, CMOS sensors are cheaper, smaller, and less power intensive than CCD sensors while still providing good image quality and performance. The common resolutions for these cameras are 480p or 720p, and the typical frame rate is between 30-60 FPS. Those two aspects are critical for maintaining low latency, since latency is the delay between the camera capturing images and the resulting video being displayed to the user. The two video transmission types are analog and digital, where analog transmission is more common for FPV setups than digital, since digital transmission has higher associated costs and greater latency. With analog transmission, FPV cameras resolution is measured in TVL (TV lines) instead of pixel counts, with the common TVL resolutions being between 480-1000 TVL. Some FPV cameras do possess a built-in OSD that can display information onto the video feed as well.

With there being so many components, there is a wide variety of FPV camera options available, each having their own specifications to meet the user's needs. For this project's purposes, the main criteria for choosing our cars' FPV camera will need to be the latency level, size/weight of the camera, sensor type, TVL resolution, video transmission compatibility, and cost of the camera.

Onboard Action Cameras

Onboard action cameras utilize high-quality camera technology to capture detailed video footage in high resolution. Specifically for drone or RC car racing, they are an attractive technology option for real-time racing situations and for video recording races due to some of their distinct features, such as gimbaled stabilization and 360-degree view, being able to record smooth and accurate footage from the car or drone's perspective. Some of the major products on the market are GoPro and DJI Action cameras, with these cameras being quite versatile, not just having applications in drone racing or aerial photography, but also in action sports, documenting travel, recording major events, and much more.

For high-end drone and RC car racing applications, this technology is especially popular because many of these high quality camera models do have built-in capabilities to stream video in real time via Wi-Fi or Bluetooth, with these capabilities allowing them to connect to a local device for live streaming. The main drawback to account for is the potential latency in video transmission. Many of these products have higher latency than specialized FPV cameras, which would limit the practicality of using the camera for real-time feedback in certain racing scenarios depending on the setup.

While this technology does possess many great features, with this project's scope though, using this technology would not be a wise use of our resources. The reason for that is the significantly high cost of acquiring such a high-quality camera, with these products easily exceeding $100. There's also the size/weight of the camera to consider, where it would be a challenge to configure it into our car's design. Thus, when considering the parameters and needs of our project, utilizing any kind of onboard action camera would not be practical.

Table 3.x. *Comparison Table of Cameras for RC Car Racing Purposes*

Features	FPV Cameras	Onboard Action Cameras
Video Transmission	Typically Analog (5.8 GHz), Real-Time Feed	Wi-Fi Streaming, Internal Video Recording
Latency	Very Low, Typically 20ms	High, Can exceed 100ms
Field of View	Wide FOV, Either 120° or 170°	Variable FOV, Ranges from 90° to 170°
Resolution	Lower, 480p to 720p	Higher, 1080p to 4K
Image Sensor	Typically CMOS	Typically CMOS
Size	More Compact	Bulkier
Weight	Lighter, Usually < 30g	Heavier, 100g or More
Battery Life	Shorter, Dependent on Power Source	Longer, Typically lasts for Multiple Hours
Cost	Range of $20-$80	Range of $100-$500+

Research Moment

Initially, after researching the specialized FPV cameras used in RC car racing, our group was going to move forward and select a FPV camera product to use for our project. However, by following the iterative engineering design process, we discovered that the RF (Radio Frequency) analog signals from the FPV cameras would not be compatible with Wi-Fi, meaning that the Wi-Fi router we'd be using would not be able to interpret the raw video signal from any of these FPV RF-based cameras.

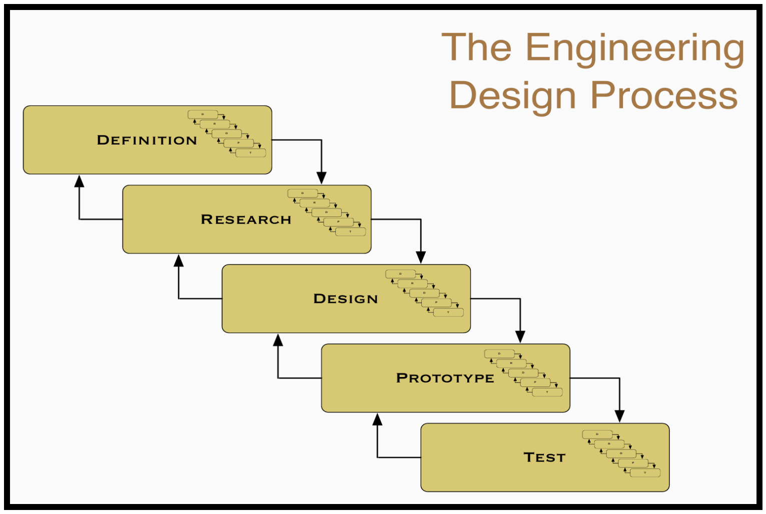


Figure 3.x. *The Engineering Design Process from UCF Senior Design Lecture*

We found a variety of reasons for why direct communication between a Wi-Fi router and RF camera isn't possible, the main reason being that Wi-Fi follows specific communication protocols (such as IEEE 802.11 standards) and modulation methods for internet data that cannot "interpret" raw analog video signals. Due to these Wi-Fi standards, Wi-Fi packets adhere to the standard by encoding data into headers, payloads, and trailers to format, transmit, and receive data in a wireless local network. Since the RF signal sends raw video without any encoding, the Wi-Fi router cannot "interpret" the data and can instead understand the signal as unstructured noise.

To determine a solution, we first brainstormed the potential idea of acquiring a designated RF receiver that we could connect and configure to the Wi-Fi router, trying to find a way to establish communication between the Wi-Fi router and RF receiver. However, we quickly dismissed the idea and instead chose to find a new camera technology that would be compatible with Wi-Fi video streaming capability.

Finding The Solution

Since we are using an ESP32 as our MCU of choice for this project, we shifted our research focus into finding the specific camera technologies used for video transmission over the internet, then looking for products that are directly compatible with the ESP32 while also meeting the necessary specifications needed for the car racing system. The best option we found is standalone camera modules that can directly pair with microcontrollers, to then create a system for video transmission over the internet. In that way, the effective combination of the camera module and microcontroller will enable the IP (Internet Protocol) camera functionality we're looking for.

The other option we found were webcams, which could work really well for this project's scope, with the camera's high quality and wireless connectivity through Wi-Fi. The challenges with this option would be any additional software and hardware setup needed for the ESP32, and the fact that ESP32 only supports certain camera interfaces. So, finding the appropriate camera product and determining how to configure it with the ESP32 could bring many unpredictable challenges and setbacks. We did find a variety of IP camera products as well, that could have been selected to offer high-end video transmission over the internet and provide easy connection to a Wi-Fi network without much additional setup required. However, similarly to the Onboard Action cameras, the price range and size/weight of the products did not make them very practical options for this project.

Overall, for our engineering design project of creating this overall car racing system, we decided to go the camera modules route, tasking ourselves with finding the best camera module to integrate with the ESP32.

3.4.2 Camera Part Comparisons and Selection

The camera modules we found are all CMOS image sensors with features that suit IoT and embedded systems applications. When searching for ESP32 compatible modules, the most popular options seem to be the OV2640 and OV7670 image sensors. The OV2640 is the most versatile sensor, coming in a variety of modules, and it even has a direct ESP32 chip integration with the ESP32-Cam development board. There are also the OV5647 and OV5642 image sensors that can come in a variety of modules as well, however, these two are predominately compatible with other MCU such as Arduino Uno and Raspberry Pi, where connection compatibility is not as certain for the ESP32. We could create a dual-MCU scheme where the ESP32 would handle the controls and the Raspberry Pi would handle the video streaming, but that would create unnecessary complications in our design and implementation of the cars system.

Thus, we narrowed our selection scope to the two image sensors options most compatible with our project, the OV2640 and OV7670 image sensors.

Table 3.x. *Comparison Table of Image Sensors for Video Streaming over the Internet*

Features	OV2640	OV7670
Resolution	2 MP (1600x1200)	VGA (640x480)
Frame Rate	Adjustable, up to 60 fps but 30 fps recommended	Typically 15 fps, but can go to 30 fps
Video Output	JPEG	Raw RGB or YUV
Connectivity	Directly Compatible with ESP32	Compatible, but requires additional interfacing
Field of View	160° FOV	Small FOV
Power Consumption	Moderate	Moderate

Setup Complexity	Less programming	More programming
Cost	$1.81	$1.03

After comparing the two options, we determined that the OV2640 image sensor would be the best match for our project, serving as the car's camera module.

3.5 Sensors

3.5.1 Sensor Technology Comparisons and Selection

Finish Line System Sensor Overview

In order to effectively time each lap and determine winners for races between multiple users, we will need an efficient and easily distinguishable method of the cars' detection across the finish line. This starts at our selection of sensors. The quality of our sensor type and method of use in our application will be extremely vital in creating an efficient and consistent system that will deliver accurate results. There are various options for sensors that can be utilized in our project but in consideration of our goal of effective timing and detection, we narrowed our options to IR sensors, RFID sensors, laser gate sensors and photo-finish cameras (using color sensing). Also, for quality and aesthetic we will implement red and green LEDs to simulate a racing countdown.

Infrared (IR) Sensors

Our first line of thought began with Infrared (IR) sensors, where there are 2 main types, active and passive. In the case of this project, both passive and active IR sensors could be used despite their differing functionalities. Here we will explore their differences and what best fits our needs and constraints.

Passive IR Sensor

Passive Infrared (PIR) sensors work by only receiving radiation, enabling them to detect moving objects that pass across the sensor by recognizing the heat emitted from the object. Some advantages of this device include low power consumption, cost-effectiveness, and reliable detection of warm-bodied entities like humans and animals, which makes it ideal for security and automation applications. However, due to its passive nature, objects being detected by this sensor must emit a significant amount of heat to be recognized and must also be in motion. These limitations mean that PIR sensors are unsuitable for detecting non-living objects or vehicles, which generally lack sufficient heat emission or may remain stationary. Additionally, environmental factors, like ambient temperature changes, can reduce the sensor's accuracy. Consequently, while effective in specific applications, PIR sensors have limited utility in scenarios requiring precise detection of various object types, as marker detection or other camera-based systems provide more comprehensive tracking and differentiation capabilities.

Active IR Sensor (Reflective VS Break Beam)

The main active IR sensor technologies we will explore include reflective and break beam sensors. Reflective IR sensors work by emitting infrared light through an IR LED and detecting the light reflected off nearby objects using a photodiode or phototransistor. When an object enters the sensor's range, it reflects the emitted IR light back to the receiver. The amount of reflected light depends on the object's distance and reflectivity, allowing the sensor to detect proximity. There are many advantages in utilizing this sensor type as it is very cost effective, ranging from $1-$5 for a basic short-range sensor. The response time of these sensors are relatively fast and could be easily implemented in our detection system, quickly registering when a car crosses the finish line and outputting precise trial times. The downsides for this method are the various possible interferences that can come up with this technology. For one, the IC cars would have to be reflective, limiting us to the type of material we can use for the chassis of the IC cars. If we did use a reflective material, we would have to make sure that the reflection can accurately hit the sensor, if not there can be inconsistent reads. Also, the lighting and other objects in the room will have to be accommodated in order to avoid any outside influence to trigger the sensor. Lastly, there is no real way to distinguish the 2 IC cars in this method, making it difficult to name a clear winner. Due to these obstacles we decided to move away from using Reflective IR sensors for our finish line system, as the advantages of this type of sensor can be met with other sensors and even better efficiently.

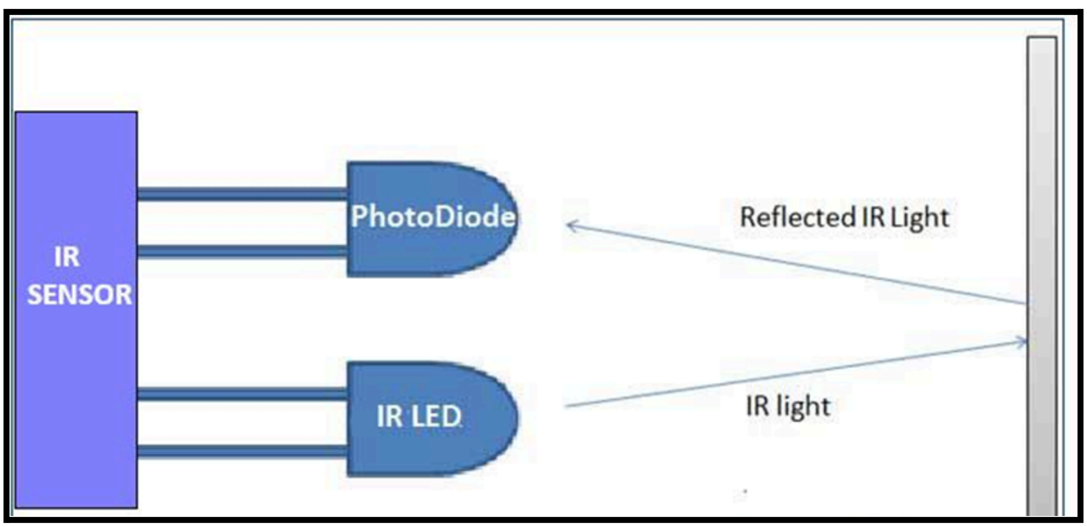


Figure 3.x.

In comparison, break beam sensors work by creating an invisible "beam" of infrared light between a transmitter and a receiver. The transmitter emits a beam of infrared lights through LED or laser diode, and the receiver faces the transmitter, continuously detecting the infrared light beam with phototransistors or photodiodes. When an object crosses the beam, and interrupts the path to the receive, a detection event is triggered. This makes it a highly precise way to detect an IC car crossing the finish line, as it only covers a small concentrated area unlike the reflective IR sensor. Break beam sensors are commonly used for applications where precise, fast object detection is critical, making it ideal for our finish line system.

General specifications of break beam sensors include a response time that ranges from about 1 microsecond to 10 milliseconds, depending on quality of product. Distance detection ranges from 10cm in general short-range models to 50m for long range industrial models. For our project, a long detection range will not be necessary as the track finish line will be about 20-30 cm long, making the sensor at both sides (left/right) only having to detect up to 10-15 cm. The energy consumption is also low, as the sensors typically operate with supply voltages at 3.3V or 5V, and consume up to about 10mA to 50mA for our applications. The wavelength of infrared (850nm-950nm) is significant in its function in the sensor. IR break beam sensors are ideal as they are not affected by visible light interference from ambient sources like sunlight or artificial lighting. This makes IR sensors more reliable in outdoor or well-lit environments.

One method of application we have considered for the usage of IR break beam sensors include positioning two sensors at differing heights, meaning that creation of different chassis heights for each car would be required. This approach is to ensure no interference between the IR sensors with each other, and also to distinguish between the different racers to confirm winners accurately. Creating a simple yet robust system is important, and IR break beams sensors can do just that as they are compatible with Arduino and other various MCUs, allowing ease of replication.

Another method we speculated about was having the sensors overhead of the finish line. Having the IR break beam sensors over the track promotes more precise reading as we can position the sensor just above the height of the IC cars. In this approach, there is also less likely a chance of outside interference such as lights or object obstructions. We ultimately moved away from this system as there are a lot of "what-ifs" in question. To individualize the IC cars and the users, we would have to pair this system with another sensor system such as RFID, cameras, color-pattern sensors, or combined with other IR sensors. To go this route, we would have to sacrifice our cost for a system that may over-complicate implementation than needed on the software end and performance may not be as accurate or consistent as other sensor options.

IR break beam sensors are a great valued sensor type that performs well for small scale time sensitive projects. The downside of these sensors is their susceptibility to light related outside interferences. In further investigation we see that there are better options in terms of speed, accuracy, reliability and identification.

Radio Frequency Identification (RFID) Sensors

Radio Frequency Identification (RFID) sensors work by using radio waves to wirelessly identify and track objects. The system consists of a reader and a tag, each with antennas that communicate via radio signals. When a tag (which can be passive, using the reader's energy, or active, with its own battery) enters the reader's range, it transmits its unique identification data. The reader captures this information and processes it, enabling applications like inventory management, access control, and logistics, all without the need for direct line-of-sight scanning.

RFID sensors were instinctually our first sensor type to consider as we knew that they bring individuality to each IC car and user. There are various tactics we can conduct when dealing with RFIDs for our finish line system. When deciding on a RFID sensor we need to consider the differences between passive versus active RFID sensors, and the suitability of their varying frequencies (low, high, ultra-high) for our project.

Passive RFID

Passive RFIDs have no internal power source, so they rely on the energy of the RFID readers and their antennas. When scanned, the RFID antenna emits an electrical signal converted into electromagnetic RF energy, activating the RFID tags in the scanning area.

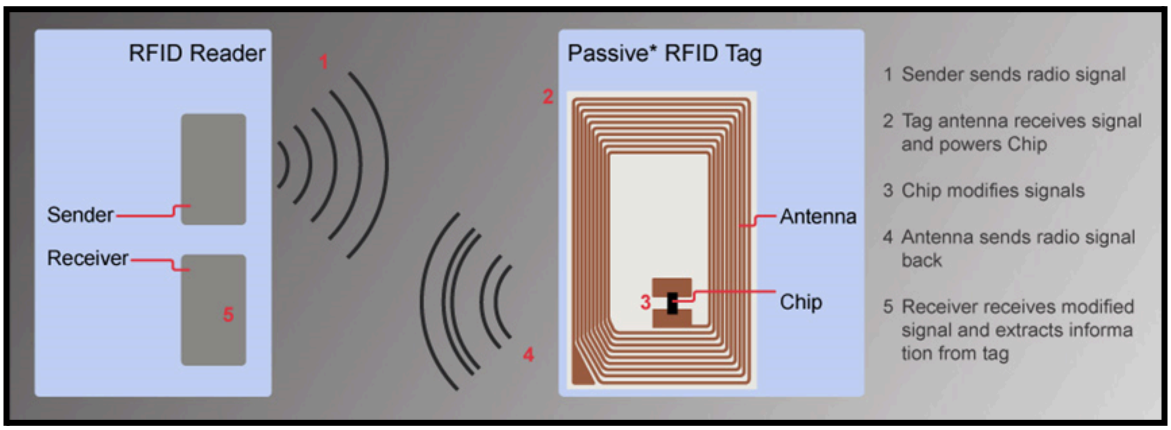


Figure 3.x.

These tags respond with programmed information, requiring minimal energy to power their internal RFID chips. This makes them easily programmable as their identifying data is stored in the RFID tag. When paired with a RFID reader that has read and write capabilities it enables the programmer to write to the tag. This is necessary in order to be able to get time measurements of each racer, and make it a straightforward process to determine a winner. From this we ask how we can get this information in an efficient manner, that will reflect lap times as accurately as possible considering potential latency. Frequency is the determining factor for the range, data transmission speed, and how the tag interacts with different materials of the RFID tag.

Low Frequency (LF) Passive RFID

The frequency range of LF passive RFID sensors spans from 30 kHz to 300 kHz, with most products operating at 125 kHz or 134 kHz. This frequency range is preferred for numerous applications due to its specific characteristics. Notably, the magnetic field within the LF band does not experience dispersion, modulation, or refraction, allowing LF RFID tags to be read even near or above metals, though the signal itself cannot penetrate metal surfaces. LF signals can pass through other materials with ease, making them versatile in various environments. Although radio-frequency technology typically struggles to function in the presence of liquids, lower frequencies, such as 125–134 kHz, are minimally affected by them. Additionally, the LF band supports proximity readings within a range of approximately 10–15 cm. One trade-off of using this low-frequency band is that tag reading and writing occur more slowly; however, this slower rate is advantageous when precision in reading is required.

LF passive RFIDs are mostly used for tracking, access control, and security applications. For example, in warehouses and retail stores they are used for tracking goods, managing inventory, and streamlining supply chain processes. They allow for automatic and remote scanning of items without the need for direct line of sight. They are also commonly used to allow access into buildings or areas with badges, cards or tags that are embedded with a passive RFID chip.

The similarities between these applications show the strengths and weaknesses of this technology. Strengths we can take away from these examples include its longevity, versatility and scalability. This type of RFID is able to penetrate through materials, with little to no effect from metal surfaces and water. Making this a great option for environmental applications and long-standing systems that can be modified. Another strength of this technology is its ability to collect individual data and be implemented in a system as seen with access control systems. This is an important feature for us to consider for the finish line system as the RFID reader will have to be able to read the car, collect data (time), and then be stored in our database to output the results of the race to users.

Unfortunately, the strengths of LF passive RFID do not translate to the needs of our finish line system. Due to its low frequency, the data transfer rate is too slow. Usually, the data transfer rate is between 4 kbps and 8 kbps for most LF passive RFIDs**.** This makes them not ideal for large amounts of data to be transmitted quickly. The tag read speed is also relatively slow, at speeds ranging from 30 to 50 milliseconds per tag. Since the finish line system is expected to read and transmit data in real-time, limited delays are essential to guarantee accurate timing feedback. These constraints alone led us to reconsider other options. As we discovered a lack of speed to be the issue, we decided to move our focus to high frequency passive RFIDs.

High Frequency (HF) Passive RFID

Passive high-frequency (HF) and ultra-high-frequency (UHF) RFID tags offer distinct advantages over their low-frequency (LF) counterparts, especially in terms of performance and versatility. Here we will explore the general specifications of these 2 differing frequencies and if they fare well as an option within our expectations.

HF RFID tags, which operate around 13.56 MHz, offer a balance between read range and interference resistance. While they share LF's strong resistance to environmental interference, HF tags extend their read range to about 10 cm to 1 m. This makes HF tags suitable for applications like access control, ticketing, and smart cards, where proximity-based interaction is important, but the tags still require a reasonably close read distance. HF RFID also provides moderate data transfer rates, making it more efficient than LF technology in transferring data over short distances. Data transfer rates range between 26 kbps and 848 kbps, depending on the communication standard the HF RFID uses. In addition, the cost of HF RFIDs is the least of the three RFID options due to its design. HF tag antennas are usually made of a copper, aluminum, or silver coil with three to seven turns. These are easier to manufacture and are the most used RFIDs, making it the most cost effective.HF RFID technology uses inductive coupling to facilitate communication between the tag and the reader. In this setup, the reader and tag interact through magnetic fields, with the reader generating an alternating current at 13.56 MHz to create a magnetic field. When a tag enters this field, its antenna captures the magnetic energy, producing a small electrical current that powers the tag's integrated circuit (IC). Data exchange between the reader and the tag is accomplished via modulation. The reader employs amplitude-shift keying (ASK) to modulate its signal, sending commands to the tag. In response, the tag modulates the reflected signal using methods like load modulation, where it alternates between absorbing and reflecting the reader's signal. This modulation technique enables the tag to send data back to the reader.

Despite these clear improvements in comparison to LF RFIDs they aren't quite enough for our expectations, and believe it could potentially have the same issues as an LF RFID. The roadblocks we can potentially see with this product is transfer data rates that may be still too slow to give accurate lap times, as most HF RFIDs are on the lower end of the potential data transfer rate. Like the LF RFID, it is dependent on multiple variables such as environment, communication standards, positioning, proximity, IC car speed and communication standards. Bringing to light these possible issues we looked further to limit any doubt that the system will work. So, we looked into ultra-high frequency (UHF) RFIDs, as we assumed the strengths we discovered in HF RFIDs would be amplified in UHF RFID systems.

Ultra-High Frequency (UHF) Passive RFID

Lastly, UHF RFID systems operate in the 860 to 960 MHz range, and use backscatter coupling, where the tag uses the energy from the reader to reflect a modified signal back, carrying information like its unique ID. This pushes performance even further with read ranges extending up to between 3 to 10 meters (sometimes up to 20-30 meters in good conditions) by leveraging far-field electromagnetic waves.

Although this detection range exceeds our needs, there are ways to adjust it to make it shorter. The selection of what type of antenna the UHF RFID plays a big part in reaching shorter ranges, as directional or near-field antennas can fit the reduced range we want. With directional antennas we can help focus the UHF RFID signal within a smaller area, effectively limiting the read range and making it more precise. Near-field antennas can operate within less than a meter, making it ideal for short range applications. Another way includes adjusting the transmitted power of the UHF RFID reader. By lowering the power output, the read range lowers in itself. This is important because if the detection range is too far out, there is more susceptibility to interference..

One of the key benefits to this RFID in particular is the data transfer rate. UHF RFID offers the fastest data transfer rates among the three passive RFID types, making it ideal in high-speed implementation. The data transfer rate range lies typically between 40 to 640 kbps (for EPC Gen2 standard, the most commonly used in UHF RFIDs), noticeably a significant jump from LF and HF RFID rates. In translation, the UHF RFID readers can read tags on objects moving at speeds up to 60 to 125 mph in optimal conditions. Though our IC cars will not likely reach these speeds, it gives reassurance that the UHF RFID has the ability to detect the IC cars rapidly. For reference, these are often used for highway tolling systems because of their reliability to detect and scan fast moving vehicles.

As a result of a RFID system with high data transfer rates, collision effects in turn can be reduced. Because the tag and reader can communicate at high speeds, the reduction in time each tag spends transmitting data enables the auto collision protocols to be more effective. In our potential utilization of this product, a responsive auto collision protocol is crucial to ensure that the time measured for both IC cars at the finish line are recorded and relayed back to the users correctly. If not, the IC cars may interfere with each other and create inconsistent reads that will result in a poor racing experience. This creates a balance as it responds to one of the few weaknesses of UHF RFIDs. Their reliance on high frequency electromagnetic waves allows for signals to be easily disrupted by the environment. For instance, electromagnetic noise from nearby devices like Wi-Fi or machinery can interfere with the RFID signal, and multipath interference occurs when signals reflect off surfaces, leading to conflicting data. To mitigate this, we will have to consider our positioning of the RFID tags, readers, Wi-Fi connected components and other potential metals in the surrounding area.

Ultimately, we forgo the utilization of UHF RFIDs in our finish line system. Considering our previous research of RFIDs, we acknowledge the possibility of implementing UHF RFIDs to be much easier than other passive RFID systems, however, it does leave us with many uncertainties. For this system to work every process will have to work in a timely and effective manner. From the communication between the tag and the reader, the reader to the database, and then to the user, there are many problems that can arise. Adjustments on the positioning of the RFID reader will be tedious as the range exceeds the distance we will use as the track will be a smaller scale. There are also vulnerabilities to Wi-Fi interruption, which can cause delays on top of the expected latency we'd experience between the application, the server, and the users. In addition, the cost of a basic level UHF RFID tag and reader range from $50 to about $300, which may not justify the cost for our use.

Active RFID

Active RFID tags, unlike passive RFIDs, are battery-powered devices that autonomously transmit signals over long distances, typically ranging from 100 meters to 1,500 meters. Operating in frequencies like 433 MHz, 860-960 MHz, or 2.4 GHz, they offer adjustable data rates and two-way communication. These tags have larger onboard memory (128 bytes to 512 KB) than passive RFID, allowing them to store additional data, and provide location accuracy within 1-3 meters in RTLS applications. Their battery life ranges from 1 to 5 years, and they are often housed in durable, weather-resistant enclosures for use in harsh environments.

An active RFID system would not be ideal for the IC car finish line due to its long read range, higher cost, unnecessary complexity, and size for this specific application. Active RFID tags have their own power source, allowing them to transmit signals over much longer distances, which could lead to inaccurate readings by detecting the car before or after it crosses the finish line. This lack of precision makes it difficult to accurately determine the exact crossing point. Having its own power source also adds extra bulk to the tag that could potentially weigh the IC cars down. Additionally, active RFID systems are more expensive than passive alternatives, both in terms of the tags and infrastructure, and they require ongoing maintenance to replace or recharge the tag batteries. For a small-scale IC car project, this added cost and upkeep are impractical. Since the finish line requires short-range, highly accurate detection, passive RFID systems provide a more reliable and cost-effective solution. Therefore, the long range and complexity of active RFID make it unsuitable for this application.

Table 3.x. *Comparison Table of RFID Sensors*

Features	Active RFID	Passive RFID (LF)	Passive RFID (HF)	Passive RFID (UHF)
Read Range	100m to 1km	Up to 10 cm	10 cm to 1 meter	1 to 10 meters
Cost of Tags	High (up to $20 or more per tag)	Low (typically <$1 per tag)	Low (typically <$1 per tag)	Low to moderate (about $0.10 to $2 per tag)
Reader Cost	Moderate to high	Low to moderate	Low to moderate	Moderate to high
Tag Lifetime	Limited by battery (3 to 5 years)	Indefinite (no battery)	Indefinite (no battery)	Indefinite (no battery)
Data Transfer Rate	100 kbps to 1 Mbps (High)	0.6 kbps - 5 kbps (Low)	26.7 kbps - 53 kbps (Moderate)	40 kbps to 640 kbps (High)
Environmental Sensitivity	Affected by temperature, durable	Performs well near metals/liquids	Performs well near metals/liquids	Sensitive to metals and liquids

Read Speed	100 to 500 tags/sec	1 to 50 tags/sec	10 to 100 tags/sec	100 to 1,000+ tags/sec
Ideal Applications	Long-range tracking, real-time data	Animal tracking, access control	Close-range, precise tracking	Mid-range tracking, bulk detection

Camera-Based System

Using a camera-based finish line system for our IC car race offers several significant advantages that enhance the racing experience. First, it provides a high level of accuracy in determining the winner by capturing real-time visual data, which minimizes the risk of errors often associated with manual tracking methods. Additionally, camera systems can easily integrate advanced image processing techniques to detect various factors, such as color, patterns, or markers, allowing for flexible configuration based on the cars' designs. This setup not only eliminates the need for physical sensors that can be cumbersome to install and maintain, but it also offers scalability for future enhancements, such as incorporating more complex tracking algorithms or adding additional features like lap counting. Overall, a camera-based system combines reliability, ease of use, and adaptability, making it a worthy option for efficient race monitoring. Here we will look at various methods a camera can be implemented in our finish line system, and which is the best approach to take.

Color Detection

Color detection is a key technique in computer vision, used to identify and track objects based on their unique color attributes within the frame. By leveraging differences in color, this method enables efficient object recognition and tracking, making it particularly useful in scenarios where distinct visual features can be exploited for classification. The underlying process involves isolating specific color ranges in an image, allowing for the detection of objects that match these color profiles. This simple yet powerful approach is widely used in applications ranging from industrial sorting systems to interactive robotic systems.

In the context of IC car racing, color detection provides a practical and cost-effective solution for determining race outcomes by tracking the cars based on their unique color schemes. The system would work by capturing real-time video footage of the race, typically focusing on the finish line area. The captured frames are processed by the computer vision system, where each frame undergoes conversion from the standard RGB (Red, Green, Blue) color space to a more suitable color space such as HSV (Hue, Saturation, Value) or LAB (Lightness, A-B color components). This conversion is crucial because HSV and LAB color spaces allow for better separation of color and brightness information, making it easier to isolate specific hues even in varying lighting conditions.

Once the image is in the desired color space, the system applies color thresholding, where specific color ranges, also known as "masks", are defined to isolate the colors of interest. Pixels within these predefined color ranges are highlighted, forming a binary mask where the target object's color appears white, and all other areas appear black. The system then processes the masked image using contour detection algorithms to identify the shape and boundaries of the (the cars) that match the specified color.

After detecting the contours, the system tracks the position of the cars as they move through the frames. For race applications, this position tracking is especially critical near the finish line. When a car's detected contour intersects with a predefined finish line in the camera's field of view, the system registers the crossing. By tracking the movement of each car based on its color, the system can determine which car crosses the line first, thereby identifying the winner of the race.

Color detection systems are favored for their simplicity, requiring only basic cameras and standard image processing libraries to operate. However, their performance can be affected by certain environmental factors, particularly variations in lighting conditions, which may alter the appearance of colors in the video. Shadows, reflections, and changes in light intensity can result in false detections or difficulties in distinguishing between similarly colored objects. To mitigate these challenges, color detection systems often require calibration, such as adjusting color thresholds dynamically or using controlled lighting environments.

The camera must meet several important requirements to ensure accurate and reliable performance. A high frame rate of 30-60 frames per second (FPS) is essential to capture smooth, continuous footage of fast-moving cars, reducing motion blur and ensuring real-time detection. The camera should also offer adequate resolution, typically at least 640x480 pixels, to clearly differentiate car colors at the finish line, with higher resolutions providing even more detail. A wide field of view (70-90 degrees) is necessary to capture the width of the track and ensure all cars are visible as they cross the finish line. High color accuracy and 8-bit or higher color depth allow the camera to distinguish between different colors accurately, while low latency is crucial for transmitting video data quickly to the processing unit. Depending on algorithm complexity and processing power, the latency can sit within a range of 10-50 ms. Cameras with fixed or adjustable focus ensure clear images at the required distance, and good performance in varying lighting conditions is necessary to prevent overexposure or shadows that could interfere with detection. Finally, the camera must be compatible with image processing algorithms and should have a reliable connection interface like USB or Ethernet to transmit real-time video data efficiently. These features ensure that the camera can capture precise, real-time footage needed for accurate race tracking and winner determination.

Consequently, this option is a very flexible and favorable option, offering significance in its simplicity in programming and installation, while maintaining accuracy and limiting risk in detecting each respective IC car. The compatibility with various processors, IDEs and programming languages offers a lot of freedom and circumstances that are ideal to work with. This detection method typically holds a +90% accuracy rate of detection in optimal conditions, and with higher quality processors and cameras, it can be paired with other detection methods such as line detection in order to

Line Detection/Tracking

Line detection is a process in computer vision that identifies linear patterns within images or video streams. It is typically used in applications such as object tracking, autonomous navigation, and industrial automation, among others. The general process begins with capturing an image or video feed through a camera. This raw image is often complex, so the first step is preprocessing to simplify it for analysis. The image is converted to grayscale, reducing the complexity by focusing only on intensity or brightness levels, rather than colors. Then, noise reduction techniques, like blurring filters, are applied to remove minor pixel variations or artifacts that could interfere with the accuracy of detection.

Next, the system applies edge detection, a crucial step that identifies where sharp changes in brightness occur, which typically signify the borders or edges of objects in the scene. Common algorithms for this include the Canny edge detector, which calculates gradients to highlight edges, and the Sobel operator, which computes differences in pixel intensity between neighboring areas. Once the edges are identified, the next step is to detect the lines within these edges.

One popular method for this is the Hough Transform, which transforms the detected edge points into a parameter space where straight lines can be identified by grouping edge points that align in a straight pattern. This approach is highly effective in distinguishing lines from other features in an image, even when the lines are partially obscured or noisy. The system can then analyze the characteristics of these lines, such as their length, angle, and location within the image.

For a finish line system, the camera would continuously monitor the finish line and detect when the IC cars intersect the defined line. As vehicles approach the finish line, the system tracks their edges and identifies when the finish line is crossed by recognizing the intersection of these edges with the detected line. This could trigger an event, such as recording the time of crossing. In more advanced systems, object detection algorithms can also be integrated to track and differentiate between multiple objects, such as identifying different cars in a race and ensuring accurate tracking of their finish times.

Marker Detection

Marker detection is a computer vision technique used to identify and track specific visual markers, such as shapes, patterns, or symbols, within a camera's field of view.. In the context of our finish line system, marker detection can accurately register when an IC car crosses a designated line. Marker detection relies on image processing algorithms and pattern recognition to identify unique markers within a live video stream or image.

The process begins with capturing the video feed from a camera that views the area where markers are expected to be present. The raw video frames undergo preprocessing, such as converting the image to grayscale to simplify analysis and applying filters to reduce noise. Once the image is ready, feature detection algorithms identify potential markers by looking for specific shapes or patterns that stand out. Many marker detection systems use easily recognizable shapes, such as rectangles or squares with high-contrast borders, to facilitate identification. One popular marker detection technique is AR marker detection (like AprilTag or ArUco), where the system identifies a distinctive, high-contrast black-and-white marker with unique internal patterns for differentiation.

For the programming side of marker detection, developers typically use computer vision libraries like OpenCV or ArUco, which provide pre-built algorithms for detecting and interpreting markers. The programming process generally involves initializing the camera and defining parameters for the detection algorithm, such as the expected marker size, orientation, and distance. The code continuously captures frames, processes each one to detect markers, and then extracts relevant data, such as the position and orientation of each detected marker. By comparing the position of the marker in consecutive frames, the system can determine when and where it appears or crosses a specific threshold, like a finish line.

In our case, markers could be placed on the vehicles themselves or embedded in the finish line area. When the camera detects a marker as a vehicle approaches and crosses the finish line, the system records the precise moment of crossing. Additionally, marker detection algorithms can distinguish between unique markers, allowing the system to track and differentiate multiple vehicles. This method provides accurate timing and helps ensure fair and reliable race results by automatically logging the exact time each vehicle crosses the finish line.

The main benefit of using marker detection is its accuracy and reliability in identifying specific objects or events, such as when a vehicle crosses a finish line. By recognizing unique markers, the system can differentiate between multiple objects, ensuring precise tracking and minimizing errors. Marker detection is also robust in various lighting conditions and can work in real-time, making it ideal for dynamic environments like racing. Additionally, marker detection systems are often easy to implement using existing computer vision libraries, making them both cost-effective and accessible for applications needing precise object recognition.

Table 3.x. *Comparison Table of Detection Techniques*

Aspect	**Color Detection**	**Marker Detection**	**Line Detection**
Method	Detects cars based on their color as they cross the finish line	Detects unique markers (e.g., QR codes, color markers) on the cars	Detects when a car crosses a specific finish line based on position

Accuracy	Moderate (can be affected by lighting or similar colors)	High (markers are unique and distinct)	High (detects the exact moment the car crosses the line)
Setup Complexity	Simple (only requires defining color thresholds)	Moderate (requires attaching and calibrating markers)	Simple (requires setting up a reference line in the camera's field of view)
Programming Difficulty	Low (basic image processing)	Moderate (requires marker recognition algorithms)	Low (basic line detection algorithms)
Speed	Fast (real-time color detection)	Fast (real-time marker detection)	Fast (real-time line crossing detection)
Lighting Sensitivity	High (may struggle in poor or changing light conditions)	Low (less affected by lighting changes)	Moderate (requires good contrast for line detection)
Cost	Low (requires only basic camera and software)	Low to Moderate (costs depend on marker production)	Low (basic camera setup with simple processing)
Maintenance	Minimal (occasional color calibration)	Moderate (may need to replace or adjust markers)	Minimal (adjust line detection thresholds as needed)
Car Differentiation	Limited to different colors	High (each car can have a unique marker)	None (only detects when something crosses the line)
Scalability	Limited (works best with a small	High (works with any number	High (can track any number of

	number of distinct colors)	of unique markers)	cars crossing the line)
Real-Time Feedback	Yes (provides immediate feedback)	Yes (provides immediate feedback)	Yes (provides immediate feedback)

Technologies Comparison & Selection

Table 3.x. *Comparison Table of Technologies*

Feature	IR Sensor Reflective	IR Sensor Break Beam	RFID LF	RFID HF	RFID UHF	Camera-Based Detection
Operating Frequency	30–40 kHz	30–40 kHz	30–300 kHz	13.56 MHz	860–960 MHz	10–60 Hz (frame rate)
Detection Range	0.5–10 cm	0.5–5 m	10–15 cm	10–100 cm	Up to 12 m	Up to 100+ m (depends on lens)
Response Time	~1 ms	~1 ms	50–500 ms	10–100 ms	1–10 ms	50–100 ms
Data Rate	N/A	N/A	4–8 kbps	25 kbps	Up to 640 kbps	Depends on resolution, 1–10 Mbps
Power Consumption	10–20 mW	10–20 mW	~50 mW	~100 mW	100–500 mW	1–5 W

Material Penetration	Minimal	Minimal	None (blocked by metal)	None (blocked by metal)	Low (affected by liquids/ metal)	None
Environmental Limits	Sensitive to ambient IR light	Sensitive to physical obstructions	Minor interference from liquids	Minor interference from liquids	Affected by liquid and metal	Affected by lighting/weather
Typical Applications	Short-range object detection, robotics	Safety barriers, people counting	Animal tagging, key cards	Contactless payment, library tags	Inventory management, tolling	Security, object recognition
Cost (per unit)	\$1–\$10	\$2–\$15	\$0.5–\$2	\$0.5–\$5	\$0.5–\$10	\$10–\$100+ (depending on specs)

3.5.2 Finish Line Camera Part Comparisons and Selection

In order to support these techniques we need an adequate camera to perform them. When researching these products we want to consider these factors: cost, ease of integration (with other hardware/software), compatibility, latency, resolution, and field of view (FOV). Upon initial research for a product that covers these necessities, I found that the PIXY 2, JeVois A33 Smart Camera, and ESP32-CAM all have the capability to execute efficient detection for our finish line system.

PIXY 2

The PIXY 2 camera is a powerful, compact vision sensor tailored for projects that require efficient color-based object tracking, line following, and marker reading. It comes with pre-installed firmware, enabling it to recognize and track multiple colored objects out of the box without requiring extensive programming or complex setup. The PIXY 2 detects objects based on color signatures, tracking position, size, and color simultaneously, which makes it ideal for applications such as robotics and automation. It integrates easily with various microcontrollers like Arduino and Raspberry Pi via multiple communication protocols, including SPI, I2C, UART, and USB. With extensive library support and intuitive software, PIXY 2 is a go-to choice for researchers and hobbyists looking to incorporate fast, reliable vision processing in their projects without the need for advanced programming skills. Here are the technical specifications found on the official PIXY 2 documentation:

- Processor: NXP LPC4330, 204 MHz, dual core
- Image sensor: Aptina MT9M114, 1296×976 resolution with integrated image flow processor
- Lens field-of-view: 60 degrees horizontal, 40 degrees vertical
- Power consumption: 140 mA typical
- Power input: USB input (5V) or unregulated input (6V to 10V)
- RAM: 264K bytes
- Flash: 2M bytes
- Available data outputs: UART serial, SPI, I2C, USB, digital, analog
- Dimensions:1.5” x 1.65” x 0.6”
- Weight: 10 grams
- Integrated light source, approximately 20 lumens

The PIXY 2's specs make it highly suitable for a finish line detection system. Its NXP LPC4330 processor (204 MHz dual-core) ensures fast image processing, allowing for real-time tracking as vehicles cross the finish line. The Aptina MT9M114 sensor captures high-resolution images (1296x976), enhancing accuracy in object recognition. The 60-degree horizontal and 40-degree vertical field of view offers ample coverage for tracking vehicles without missing key visual data. Its low power consumption (140 mA) and USB or flexible voltage input make it energy-efficient and adaptable to different setups. Additionally, multiple data output options (UART, SPI, I2C, USB) provide versatility for integration with various microcontrollers. The integrated 20-lumen light source ensures reliable detection under varying lighting conditions, and its compact, lightweight design makes it easy to mount and position.

The PIXY 2 camera features streamlined software capabilities tailored for color-based object tracking, line following, and basic barcode reading. Its software is designed to identify and track objects by recognizing specific color signatures, allowing it to detect multiple objects and output information about their position, size, and color. The PIXY 2 also includes a specialized line-following algorithm, ideal for robotics applications, which can detect and follow line patterns and intersections, making it useful for navigation. It supports color-coded tags for multi-object tracking, making it possible to uniquely identify specific targets or patterns. The camera’s firmware is pre-installed and easily customizable via an intuitive software interface, PixyMon, where users can adjust color signatures and monitor the camera's output in real-time. The PIXY 2 is compatible with a range of microcontrollers and platforms (Arduino, Raspberry Pi, etc.), with libraries provided to integrate seamlessly, making it accessible for users needing reliable object detection without extensive programming.

JeVois A33 Smart Camera

The JeVois A33 Smart Camera is a versatile, Linux-based vision processing camera designed for complex computer vision tasks. Equipped with a powerful onboard processor, it supports advanced functions such as object detection, feature tracking, marker reading, and even certain deep learning tasks. The JeVois A33 is programmable using C++ or Python, allowing for custom vision algorithms that run directly on the device. It interfaces with external systems through serial communication, making it compatible with microcontrollers and larger computing platforms. Due to its high flexibility and processing capabilities, the JeVois A33 is ideal for robotics, research, and applications requiring sophisticated vision processing in a compact form. The capabilities and details of this camera are reflected under the following the specifications:

- Processor: Allwinner A33 quad core ARM Cortex A7 processor @ 1.34GHz with VFPv4 and NEON, and a dual core Mali-400 GPU supporting OpenGL-ES 2.0.
- Memory: 256MB DDR3 SDRAM.
- Solid state disk: micro SD slot for firmware and data, supports up to 25Mbyte/s.
- Camera sensor: 1.3MP camera with
    - SXGA (1280 x 1024) up to 15 fps (frames/second)
    - VGA (640 x 480) up to 30 fps
    - CIF (352 x 288) up to 60 fps
    - QVGA (320 x 240) up to 60 fps
    - QCIF (176 x 144) up to 120 fps
    - QQVGA (160 x 120) up to 60 fps
    - QQCIF (88 x 72) up to 120 fps
- USB device interface: mini USB port for power, video streaming (UVC webcam standard), and serial-over-USB command-line interface and text data streaming.
- Hardware serial port: 5V or 3.3V (selected through VCC-IO pin) micro serial port connector to communicate with Arduino or other embedded controllers.
- Power: 3.5 Watts maximum from USB port. Requires USB 3.0 port or Y-cable to two USB 2.0 ports.
- Thermal: Integrated cooling fan, maintain stable 1.34GHz speed under full processor load, no overheating or thermal throttling.
- LED: One two-color LED: Green means power is good. Orange means power is good and the camera is streaming video frames.
- Size: 28 cc or 1.7 cubic inches (plastic case included with 4 holes for secure mounting). Only 17 grams.

It is also equipped with advanced software capabilities, enabling it to perform a wide range of computer vision tasks. It supports deep neural networks (TensorFlow, Darknet) for recognizing up to 1,000 different objects and can identify key items in complex scenes. The camera can detect and decode QR codes, barcodes, and ArUco markers, making it versatile for tagging and tracking. It includes visual attention and gist analysis to identify notable items and recognize places. With NEON-accelerated and

GPU-optimized processing, it achieves fast image processing for tasks like background subtraction, eye tracking, and optical flow. Multicore support enables efficient, real-time edge detection, feature extraction, and image segmentation. For autonomous applications, the JeVois A33 can detect roads and follow paths, making it ideal for robotics and machine vision projects. Additionally, it supports video capture to microSD and can operate as a USB camera at up to 120 fps, making it a powerful tool for both analysis and real-time applications.

The JeVois A33 Smart Camera is well-suited for a finish line detection system due to its powerful hardware and versatile image processing capabilities. The camera's Allwinner A33 quad-core ARM Cortex-A7 processor at 1.34GHz, with NEON and VFPv4 optimizations, ensures efficient and fast image processing, enabling it to handle real-time object recognition and movement detection as cars cross the finish line. Its 1.3MP camera sensor supports multiple resolutions, including up to 120 frames per second (fps) at lower resolutions, providing the ability to capture fast-moving objects accurately. The dual-core Mali-400 GPU with OpenGL-ES 2.0 further boosts performance, particularly for video-intensive applications.

The JeVois A33's mini USB port for power and data transfer, along with a flexible micro-serial interface (5V or 3.3V) for connecting to microcontrollers, makes it easy to integrate with a larger system and capture and transmit finish line data. With a compact size, low weight (17 grams), and sturdy mountable case, it can be securely positioned at the finish line. The integrated cooling fan maintains stable processing speeds without thermal throttling, ensuring consistent performance under high loads. Combined with onboard storage for firmware and data on a microSD card, these specs make the JeVois A33 a reliable, high-performance choice for accurately capturing and processing finish line data in real-time.

ESP32-CAM

The ESP32-CAM is a low-cost microcontroller module with built-in Wi-Fi and Bluetooth, powered by the ESP32 chip. It includes an OV2640 camera capable of capturing images and streaming video, making it popular for IoT, surveillance, and remote camera projects. The module features a microSD slot for storage, GPIO pins for connecting additional sensors, and can be programmed using Arduino IDE or Espressif's ESP-IDF. Its compact size, wireless capabilities, and camera integration make it ideal for embedded and remote sensing applications. Some features and specifications of the ESP32-CAM include:

- Microcontroller: ESP32-S with dual-core Xtensa LX6 processor, up to 240 MHz
- RAM: 520 KB SRAM
- Flash Memory: 4 MB PSRAM
- External Storage: MicroSD card slot (supports up to 4 GB)
- Camera Sensor: OV2640, 2 MP
- Image Resolution: Up to 1600x1200 (UXGA)
- Adjustable Frame Size: Various options (up to VGA, SVGA, etc.)
- Lens: Standard, fixed-focus

- Wi-Fi: 802.11 b/g/n (2.4 GHz)
- Bluetooth: BLE and Bluetooth 4.2
- GPIO: Up to 9 GPIO pins available
- PWM: Yes (for controlling servos, LEDs, etc.)
- Communication Protocols: I2C, SPI, UART
- LED Flash: White LED for low-light conditions, connected to GPIO 4
- Operating Voltage: 5V (via 5V pin) or 3.3V (via 3.3V pin)
- Power Consumption: Approximately 160 mA during operation
- Sleep Mode: Low-power mode available for energy-saving applications
- Dimensions: 27 mm x 40.5 mm
- Weight: Around 10 grams
- Antenna Options: Onboard PCB antenna and U.FL connector for external antenna

This camera supports video streaming over Wi-Fi, basic face detection and recognition, and can act as a standalone web server. With compatibility across Arduino IDE, ESP-IDF, and MicroPython, it's accessible to developers of varying skill levels and can control peripherals, log data to a MicroSD, and be updated over-the-air (OTA). However, its limited memory and processing power can restrict high-resolution image handling and complex processing tasks. Stability issues with video streaming and Wi-Fi connections can arise, especially in less reliable network environments. Additionally, the ESP32-CAM's fragmented documentation and occasional library compatibility issues may increase troubleshooting time, and OTA updates can be prone to failure without careful setup. These strengths and challenges make the ESP32-CAM an excellent choice for straightforward, low-power projects but potentially difficult due to its high reliance on programming for all functions of the camera.

Using this product is a potentially good choice for a finish line system due to its ability to capture high-resolution images, process data quickly, and transmit results wirelessly. Its built-in camera ensures clear, accurate recordings of race completions, while the microcontroller's dual-core processing and ample PSRAM handle real-time image processing efficiently. The Wi-Fi capability allows for seamless integration with a central server for live streaming or instant result uploads, and its GPIO pins enable easy integration with sensors or timing systems. Compact, low-power, and versatile, the ESP32-CAM offers a reliable and cost-effective solution for automated race tracking.

Products Comparison & Selection

After careful consideration of these three options we decided to utilize the PIXY2 Smart Camera despite its higher cost at about $70. This is mainly due to its built-in processing, ability to work with different microcontroller software environments, and relatively low power consumption. It will best give the ability to create an accurate method of determining winners, as we are able to utilize its color detection and line tracking feature simultaneously in order to capture the moment the car crosses the finish line.

3.6 LEDs

3.6.1 LED Technology Comparisons and Selection

An important aspect of this system and user experience is a countdown to determine when the race is starting. To do this we are going to utilize LEDs that are programmed and timed in order for limited delay or latency in communication between the countdown sequence to the users. There are various types of LEDs that can be used but due to the simplicity and function we settled on 2 different options, including RGB LEDs and Bi-Color LEDs.

RGB LED

RGB LEDs are a versatile type of light-emitting diode capable of producing a wide range of colors. They achieve this by combining three individual red, green, and blue LEDs within a single package. Each of these LEDs operates independently, allowing for the blending of their light to create various colors. By adjusting the brightness of each LED through Pulse Width Modulation (PWM), it's possible to generate millions of color combinations. The typical RGB LED has four leads: one common anode or cathode shared by the three LEDs and one lead each for controlling red, green, and blue. This design allows precise control of each color channel, enabling smooth transitions and vibrant color displays. Some RGB LEDs, such as addressable LEDs, have built-in driver chips. These chips allow each LED to be individually controlled via a single data line, making them ideal for applications where complex lighting patterns or animations are needed. These LEDs provide the ability for more customization, color range, and possible pattern creation. However, for the simplicity of the countdown function, we saw it to be unnecessary. While it is an attractive choice for its capabilities, we found that it would consume more power, cost slightly more and overcomplicate the engineering of the countdown system.

Bi-Color LED

Bi-Color LEDs can produce two distinct colors from a single package. They achieve this by housing two separate LED chips, typically of different colors such as red and green. These chips can be independently controlled, allowing the LED to emit either color depending on how it is powered. They typically come in two common configurations: two-lead and three-lead. In the two-lead version, the color is determined by the polarity of the voltage applied. For example, applying positive voltage to one lead produces red, while reversing the polarity produces green. In the three-lead version, there is a shared common lead (either anode or cathode) and two independent leads for controlling each color. This allows for more straightforward control, as both colors can be lit simultaneously to produce a third color, often amber or yellowish.

These LEDs are straightforward to integrate into circuits, requiring only basic components like resistors or transistors for current control. Their ability to switch between two colors makes them ideal for scenarios where a change in state needs to be communicated clearly and efficiently.In terms of power consumption, Bi-Color LEDs are highly efficient. Each LED chip typically operates within a current range of 10 to 20 mA, similar to standard single-color LEDs. When only one color is active, the power consumption is minimal.

Single Color LED

Single-color LEDs are the simplest and most commonly used light-emitting diodes, designed to emit light in a specific color such as red, green, blue, or white. They operate by passing current through a p-n junction, where electron-hole recombination produces light through electroluminescence. Known for their high efficiency, these LEDs typically require a forward voltage of 1.8V to 3.5V (depending on the color) and draw 10 to 20 milliamps (mA), resulting in power consumption as low as 0.02 to 0.07 watts.

Table 3.x. *Comparison Table of LED Technologies*

Feature	**RGB LEDs**	**Bi-Color LEDs**	**Single-Color LEDs**
Colors Available	Full RGB spectrum (millions of colors)	Two colors (e.g., Red and Green, plus Amber when combined)	One specific color (e.g., Red, Green, Blue, White)
Number of Chips	3 (Red, Green, Blue)	2 (Typically Red and Green)	1
Number of Leads	4 (or 3 for common anode/cathode models)	2 (polarity-based) or 3 (independent control)	2

Control Complexity	Requires PWM or built-in driver for color blending	Simple polarity or individual lead control	Simple on/off control
Power Consumption	20-60 mA per LED depending on brightness and usage of all three chips	10-40 mA (20 mA per color when both active)	10-20 mA typical for a single chip
Forward Voltage	2.0V - 3.5V (per color chip)	1.8V - 3.5V (varies per color)	1.8V - 3.5V depending on color
Brightness	High; dependent on individual chip intensity	Moderate; depends on active color(s)	Moderate to high depending on size/type
Cost	Higher due to complex functionality	Moderate; slightly higher than single-color	Low; cost-effective for basic use
Ease of Use	Requires microcontroller or driver circuit	Easy to use with basic circuit setup	Very easy; direct control

Typical Size	Available in various sizes, including matrices	Small; common for indicators	Very small; ideal for compact circuits

3.6.2 LED Part Comparisons and Selection

When considering our approach for the countdown light sequence, we decided that taking individual LEDs and connecting into a matrix-like format would create clutter and unwarranted complexity. Fortunately, there are LED matrix modules that are capable of connecting to most microcontrollers. Considering our needs and research of LED technology, we can find the best product that will allow us to execute our vision for the countdown light sequence.

Adafruit Bicolor LED Square Pixel Matrix with I2C Backpack - Qwiic / STEMMA QT

The Adafruit Bicolor LED Square Pixel Matrix with I2C Backpack is an 8x8 LED matrix capable of displaying 64 pixels in red, green, or a combination of both (amber). It features an integrated HT16K33 driver chip for easy control via I2C, requiring only two pins (SDA and SCL) for communication. The matrix is equipped with Qwiic/STEMMA QT connectors for plug-and-play compatibility, allowing seamless daisy-chaining with other I2C devices. It operates at both 3.3V and 5V, making it compatible with a wide range of microcontrollers, most importantly the ESP32 that we will be utilizing. The device measures 1.5" x 1.5" (38mm x 38mm) and is ideal for simple displays. It is supported by the Adafruit GFX and Adafruit LED Backpack libraries, it's a compact and versatile solution for creating low-resolution visual displays. This product is ideal since it has an integrated driver that allows us to easily connect to our ESP32. It being Bi-Color lets us implement a red and green countdown sequence to represent when to start driving for the users.

Adafruit Small 1.2" 8x8 Bi-Color (Red/Green) Square LED Matrix

The Adafruit Small 1.2" 8x8 Bi-Color (Red/Green) Square LED Matrix is a compact display featuring 64 LEDs arranged in an 8x8 grid. Each pixel can light up in red, green, or a combination of both to produce yellow. With its small 1.2" x 1.2" (30mm x 30mm) footprint, it's perfect for creating compact visual displays, including icons, patterns, and scrolling text. The matrix requires an external driver for control, such as the HT16K33 I2C Backpack for simplified interfacing. Designed for versatility, it operates at standard logic levels and can be easily integrated into projects using microcontrollers like the ESP32, Arduino, or Raspberry Pi. This matrix is a good option as it is the most compact and is bi-color, enabling us to create a red/green sequence to represent a 'go' signal. One downside that makes this matrix option not as attractive as the others is its lack of a driver, leaving us to order and solder onto the LED matrix pins.

HiLETGO MAX7219

The MAX7219 8x8 LED Matrix is a single color, versatile display module featuring 64 individual LEDs arranged in an 8x8 grid, capable of displaying text, numbers, and basic animations. Powered by the MAX7219 driver chip, it simplifies control by using only three microcontroller pins for communication via SPI. The module supports brightness adjustment and cascading, allowing multiple matrices to be connected in series for larger displays. It operates at 5V and is compatible with a wide range of microcontrollers, including ESP32, Arduino, and Raspberry Pi. Compact and easy to use, it's ideal for creating visual outputs in projects such as scoreboards, clocks, and message boards. This option is also worthy of consideration as its integrated driver makes it simple to directly connect to the ESP32 and program. However, this is only a single color LED, meaning we would have to create an alternate sequence in order for users to understand when the race is starting.This may cause a higher power consumption and more complexity in the programming and timing of the process.

Products Comparison & Selection

Table 3.x. *Comparison Table of LED Matrix Products*

Features	**Adafruit Bicolor LED Square Pixel Matrix with I2C Backpack**	**Adafruit Small 1.2" 8x8 Bi-Color (Red/Green) Square LED Matrix**	**MAX7219 8x8 LED Matrix**
LED Matrix Size	8x8 (64 LEDs)	8x8 (64 LEDs)	8x8 (64 LEDs)
LED Color(s)	Red and Green (Bi-Color)	Red and Green (Bi-Color)	Single Color (usually Red)
Driver Chip	HT16K33 (I2C)	External driver (requires an I2C backpack or controller)	MAX7219 (SPI)

Interface	I2C (with Qwiic/STEMMA QT connectors)	Requires external driver (typically I2C)	SPI
Voltage	3.3V to 5V	3.3V to 5V	5V
Dimensions	1.5" x 1.5" (38mm x 38mm)	1.2" x 1.2" (30mm x 30mm)	2.5" x 1.5" (64mm x 38mm)
Microcontroller Compatibility	ESP32, Arduino, Raspberry Pi	ESP32, Arduino, Raspberry Pi	ESP32, Arduino, Raspberry Pi
Control Pins	2 (SDA, SCL)	Depends on the external driver (typically I2C)	3 (SPI: MOSI, CLK, CS)
Brightness Control	Yes, via I2C	Yes, with external driver	Yes, via MAX7219
Multiple Matrix Cascading	Yes, via I2C	Yes, with external driver support	Yes, via SPI
Applications	Wearables, small displays, icons, scrolling text	Compact displays, low-res visuals	Clocks, scoreboards, message boards, simple graphics

Power Consumption	~20-30mA (depends on brightness)	~30-40mA (depends on brightness)	~50mA (depends on brightness and LEDs lit)

After careful consideration of these three options we decided to utilize the Adafruit Bicolor LED Square Pixel Matrix with I2C Backpack for our finish line system's LED Matrix. This is due to the product meeting all the needs for the countdown sequence we're going to create, and having the driver chip already included with the product was a major factor in the selection process since it reduces additional costs and setup complexity for the matrix.

3.7 Power Supply Unit

3.7.1 Hardware Components Power Needs

With all of the other hardware components having been selected, knowing the power needs from each device, we're now able to begin selecting the power supply and power delivery components in the form of battery supply and voltage regulation.

Table 3.x. *Peak Power Needs from IC Cars System Devices*

Device	Quantity	Voltage (V)	Current (mA)	Peak Power (W)
MCU	1	3.3	150-300	~1
Driving Motor	1	3 (Max)	1500	<5
Steering Motor	1	3.3 (Approximately)	500	~1.5
Dual Motor Driver	1	5	1500 (per channel)	7.5 (Max)
Camera	1	3.3	40-60	<1

Table 3.x. *Peak Power Needs from Finish Line System Devices*

Device	Quantity	Voltage (V)	Current (mA)	Peak Power (W)

MCU	1	3.3	150-300	~1
Camera	1	5	150-200	~1
LED Matrix	1	5	1000 (Max)	5

By gathering the voltage and current requirements from each device, then grouping them into the two systems as seen in the tables above, it enables us to make informed decisions for the battery capacity the systems will need and the voltage regulation these systems will use to deliver power to every device.

When examining the peak power needs for our IC cars, the MCU and the camera will have consistent power consumption at 3.3 V when the cars are in use, but the motors setup as a whole, both motors and the motor driver, will have varying power consumption levels when in use due to the motor controls activity. After our motors assessment of the RC car, it’s critical to note that the driving motor max voltage value will not be used since we’ll be limiting it to a range of 1-1.5 V for a more manageable speed. Also, for the steering motor’s voltage, while the RC car does give the motor 4.12 V, we determined that we could lower the voltage to 3.3 V with the steering motor still functioning as intended. This was done to help make the voltage regulation setup and power delivery system simpler since the MCU and camera takes 3.3 V as well. Then for the dual motor driver, while the input range is 2.7-10.8 V and both DC motors take 3.3 V or less, we researched and found that it is best to set the input to 5 V, for the driver and both DC motors connected to the driver to be sufficiently powered.

Overall, our IC cars will need a suitable 7.4 V battery with significant capacity (mAh) to efficiently and effectively power all of these devices for our cars’ desired battery life. They will also need 3.3 V and 5 V voltage regulators to step-down the 7.4 V for safe power delivery to their respective devices.

When examining the peak power needs for our finish line, the MCU will have consistent power consumption at 3.3 V and the camera will have consistent power consumption at 5 V. Then for the LED matrix, it will only be used for the start of races, so while it can consume upwards of 5 W, the countdown sequence we create will not require the matrix’s max power consumption (most likely 3 W on average) and it will only be active for a short period of time each race.

Overall, our finish line will also need a suitable 7.4 V battery to efficiently and effectively power these three devices. The finish line will also need a 3.3 V regulator for the MCU and a 5 V regulator for the camera and LED matrix.

3.7.2 Battery Technology Comparisons and Selection

Before investigating specific battery technologies, the question of which kind of battery our project needs must be addressed first. There are two main types of batteries, those being primary (non-rechargeable) and secondary (rechargeable) batteries. Primary batteries are designed for one-time use with a high shelf life for low-drain applications, they cannot be recharged, and they must be replaced after depletion. Secondary batteries are designed for multiple cycles of charging and discharging for high-drain applications, they can be recharged, and they are expected to keep sufficient capacity for hundreds to thousands of charge-discharge cycles. Common applications for primary batteries are smoke detectors, flashlights, and remote controls. Common applications for secondary batteries are smartphones, laptops, portable electronic devices, and electric vehicles.

By examining the needs of our project, secondary batteries are the optimal choice for both the IC cars and the finish line due to the high-drain power requirements of these devices when in frequent use, as well as the overall benefits of using rechargeable batteries over non-rechargeable batteries. With secondary batteries being the choice for our project, some of the important properties that will need to be kept in consideration during research are energy density, self-discharge rate, voltage range, temperature range, charging time, and cycle life. Specifically for the project's IC cars, some factors to consider are the size and weight of the battery, the cost effectiveness of the battery technology, and overall efficiency in meeting our performance requirements.

Lithium-Ion (Li-Ion) Batteries

Lithium-ion batteries are one of the most popular and versatile batteries in use today, having become an integral part of modern society by powering many electronic devices, electric vehicles, and even renewable energy storage systems. Due to the vast variety of their applications, lithium-ion batteries do come in many different shapes and sizes, however, all Li-Ion batteries possess the same battery cell anatomy and function.

Each Li-Ion battery cell consists of two electrodes, a cathode and an anode, an electrolyte, and a separator. The cathode is the positive electrode and is typically composed of a lithium metal oxide compound. The anode is the negative electrode and is typically composed of graphite. The electrolyte is the medium that facilitates the movement of lithium ions between the cathode and anode, and the electrolyte is typically some form of liquid or gel-like substance containing lithium salts dissolved in an organic solvent. Lastly, the separator is a thin, insulating material that is placed in-between the cathode and anode, which is used to prevent the movement of electrons yet allow the movement of ions.

Putting the chemistry together, the Li-Ion battery can efficiently charge and discharge energy. For the battery's charging process, lithium ions move from the cathode to the anode through the electrolyte, effectively storing energy. For the discharging process, lithium ions flow back to the cathode from the anode, releasing the previously stored energy.

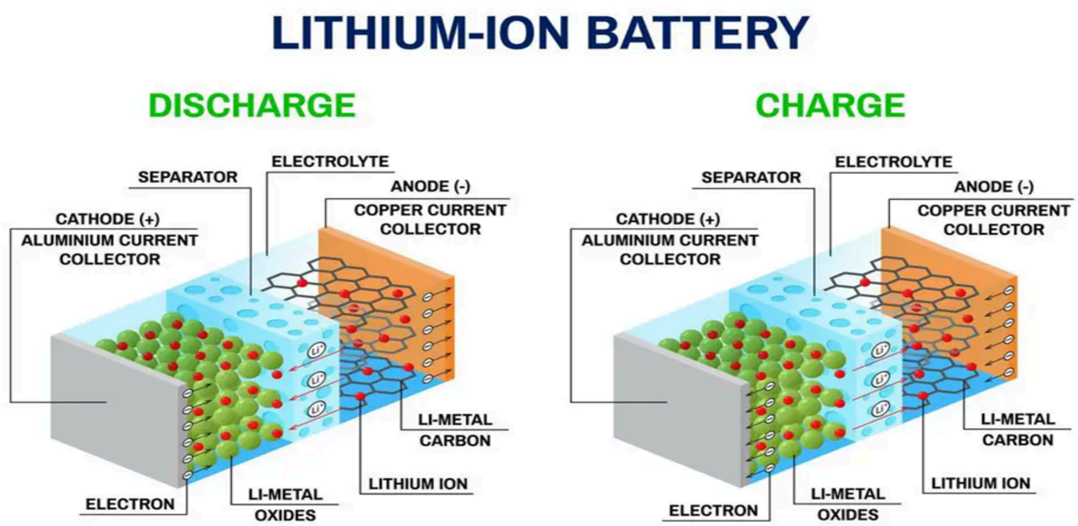


Figure 3.x. *Charging and Discharging of a Lithium-Ion Battery*

The advantages of Li-Ion batteries start with their high energy density, where they store large amounts of energy relative to their size and weight. Due to that energy-to-weight ratio, Li-Ion batteries are more suitable than other battery technologies for applications in need of lighter and smaller battery supplies. Li-Ion batteries also possess a lower self-discharge rate than other technologies, where they retain their charge well when the device is not in use. The other advantages of Li-Ion batteries are that they have no memory effect, and they generally have faster charging times compared to other technologies.

Li-Ion batteries do have limitations and disadvantages though, the main one being the potential risk of thermal runaway. Thermal runaway can occur when a battery has been damaged or mishandled, creating the risk of a fire or an explosion being produced if the battery is severely overheated. The other limitation is the battery's finite lifespan, where the battery will progressively degrade over a long period of time and use, eventually resulting in lower capacity and reduced performance. The only other disadvantage to note is that Li-Ion battery technology has a high cost due to the technology being more expensive to manufacture than other technologies.

Lithium Polymer (Li-Po) Batteries

Lithium polymer batteries are very similar to lithium-ion batteries, possessing the same characteristics and function of Li-ion batteries, using lithium-based materials for the electrodes of the battery cells. The main difference comes in the electrolyte that Li-Po has. Instead of a liquid-based substance, Li-Po batteries use semi-solid polymers with higher conductivity. Due to the electrolyte, Li-Po batteries have more lightweight designs than Li-Ion batteries, being housed in thin and flexible pouches instead of bulky metal casings that Li-Ion batteries have. For that reason, Li-Po batteries are a popular choice for applications with critical size and weight requirements, such as drones and RC vehicles.

With Li-Po batteries having the same advantages as Li-Ion batteries, they do have the same disadvantages and limitations as well. Though Li-Po batteries do have more of a safety concern than Li-Ion batteries since they are more susceptible to swelling or puncturing due to its inherent design, which can result in thermal runaway. Li-Po are typically more expensive than Li-Ion batteries as well, and with lithium batteries already having a high cost, this is a major factor to take into consideration for our project.

One more important consideration for both Li-Po and Li-Ion batteries is that they do have strict battery charging requirements to ensure optimal battery performance and maintain proper battery health to prevent overheating. It is recommended that the CCCV charging method be used to avoid overcharging or undercharging. The CCCV charging method consists of two phases, a constant current phase and a constant voltage phase. In the constant current phase, a fixed current is supplied to the battery until it reaches a certain voltage threshold. Once that voltage limit is reached, the charger switches to the constant voltage stage, where it maintains a steady voltage while decreasing the current as the battery nears saturation. Therefore, if lithium battery technology is used for our project, a certified charger with the proper safety features must be acquired and used to meet the charging requirements of these batteries.

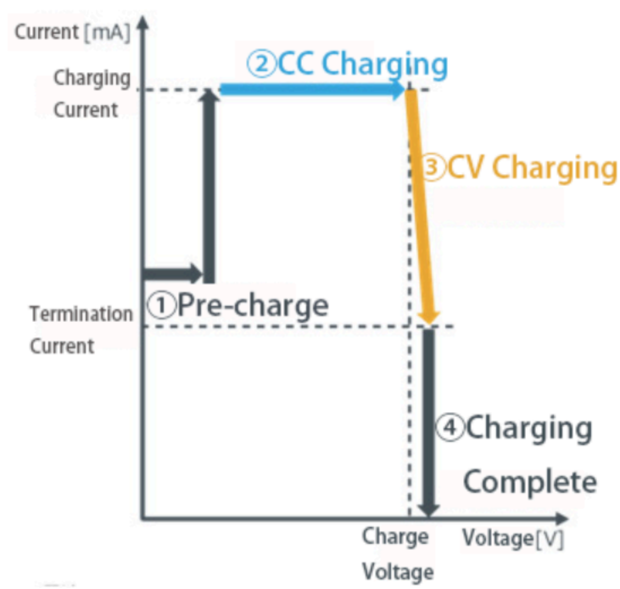


Figure 3.x. *CCCV Charging Process for Li-Ion Batteries*

Nickel-Metal Hydride (NiMH) Batteries

Nickel-Metal Hydride batteries is another popular rechargeable battery technology in use today, having become more popular than the older Nickel-Cadmium (NiCd) battery technology. Similarly to lithium batteries, NiMH batteries have a wide range of applications, the most common being in consumer electronics such as digital cameras and gaming devices, hybrid vehicles, and serving as rechargeable batteries for household devices, in both AA and AAA formats.

NiMH batteries consist of an anode with hydrogen-absorbing alloys, a cathode with nickel hydroxide, and a potassium hydroxide solution as the electrolyte. When the battery is charging, hydrogen atoms get absorbed by the anode through the electrolyte, forming a hydride in the anode and reducing the cathode to nickel hydroxide. Then, when the battery is discharging, the hydride releases hydrogen ions back to the cathode, which combine with the nickel hydroxide to regenerate nickel oxide, thus producing electrical energy in the process.

NiMH batteries are one of the most well-balanced rechargeable battery technologies in the market, with an even balance between energy density, power density, and efficiency. Energy density refers to the amount of energy stored per weight, power density refers to a measure of how quickly power can be delivered from the battery, and efficiency being determined by how well the battery converts the stored energy during charging into usable electrical energy for discharging. With power density, the discharge rates are especially important because these batteries can be discharged at high rates to provide substantial power for high-drain devices, such as digital cameras or power tools. Thus, NiMH batteries are very effective for devices that require bursts of power, being capable of delivering high current outputs.

NiMH batteries provide good energy density (greater than NiCd but less than Li-Ion), high power density, good efficiency (slightly less than Li-Ion), decent self-discharge rate (higher than Li-Ion), fair cycle life, little memory effect, and is a very environmentally friendly option. Again, they may not be the best in certain metrics such as energy density or efficiency, but they are a very well-rounded option with a better balance in cost and capacity compared to lithium-based batteries.

Lead Acid Batteries

Lead Acid batteries are the oldest rechargeable battery technology, and they are known for their reliability and cost-effectiveness. Lead acid batteries consist of two lead plates, one of them serving as the cathode and the other as the anode, and a filling of sulfuric acid (H2SO4) as the electrolyte. They are commonly used as the starter batteries in vehicles, sources of uninterruptible power supply, and in many industrial applications.

While lead acid batteries do have many advantages, the technology being thermally stable, reliable, easy to maintain, and possessing a relatively low cost, the disadvantages do outweigh the advantages for the purposes of our project. Those disadvantages being their low energy density, more limited cycle life, and the major issue of the size and weight of lead acid batteries.

Table 3.x. *Comparison Table of Secondary Batteries (Battery University)*

Specifications	**Lead Acid**	**NiMH**	**Lithium-Ion**		
			Cobalt	Manganese	Phosphate
Specific Energy	30-50	60-120	150-250	100-150	90-120

(Wh/Kg)					
Internal Resistance	Very Low	Low	Moderate	Low	Very Low
Cycle Life	200-300	300-500	500-1000	500-1000	1000-2000
Charge Time	8h-16h	2-4h	2-4h	1-2h	1-2h
Self-Discharge Rate (per month)	5%	30%	<5% Due to protection circuit consuming 3% per month		
Charge Temperature	-20 to 50℃	0 to 45℃	0 to 45℃		
Maintenance Requirements	3-6 Months (Topping Charge)	Full Discharge every 90 Days	Maintenance Free		
Safety Requirements	Thermally Stable	Thermally Stable	Mandatory Protection Circuit		
Cost	Low	Moderate	High		

3.7.3 Battery Supply Comparisons and Selection

Due to the properties of lithium-based batteries and effectiveness in RC car applications, we decided to look into Li-Ion and Li-Po battery products to be the power source for both the cars and finish line systems. Using the power needs tables in section 3.7.1, more specifically the current requirements, we used the following equation to determine the appropriate battery capacity for our desired runtime.

Determining Battery Capacities

$$\boxed{Battery\ Runtime = \frac{Battery\ Rated\ Capacity\ (in\ mAh)}{Load\ Current\ (in\ mA)}}$$

Figure 3.x. *Battery Pack Capacity Formula*

For the cars' battery life, one of our engineering specifications is to have the battery last for over 15 minutes before needing to be recharged, and we estimate that each race will last for at most 5 minutes. So in order to balance out the size/weight of the battery with the battery's capacity, we decided to aim for around 30 minutes of battery life, which would allow us to easily run multiple full races without the possibility of the batteries dying.

Taking all the current requirements into account, the total estimated load current is around 2310 mA with all of the devices powered and running. Adjusting the formula, we can take the load current (mA) * desired run time (hours) to find the battery capacity (mAh). The estimated battery capacity for 30 minutes would be: 2310 (mA) * ½ (hour) = 1155 mAh. Thus, the cars' Li-Po battery pack should have a 1100 or 1200 mAh battery capacity to sufficiently meet our desired runtime.

For the finish line's battery life, we would like the system to have at least one hour of runtime before needing to be recharged. Taking the current requirements into account, the total estimated load current is around 1040 mA (640 mA from LED matrix on average usage) if all devices are powered and running. For our system, the LED matrix will be on for a small period of time, so the total load current will be significantly reduced for the majority of the time the system is running. Using that same formula, the estimated battery capacity for one hour would be: 1040 (mA) * 1 (hour) = 1040 mAh. Again, since the LED matrix will only be used at the beginning of races, the battery life would be much more than 1 hour for a battery capacity of 1040 mAh.

Battery Packs Selections

Taking this information into account, we researched Li-Po and Li-Ion battery brands for products that are cost-effective, reliable, have a good C-rating, and possess a compatible form factor for our desired size/weight specifications. With most of the popular RC car battery brands having higher capacity and higher costs for the scale of this project, we found URGENEX to be a suitable choice for affordable and reliable lithium-based battery products that would work well for our project.

Table 3.x. *URGENEX 2s Li-Po Batteries Features*

Features	URGENEX 2S Li-Po Batteries
Voltage	7.4 V
Capacity Range	1000-1500 mAh
C Rating	20-35 C (continuous discharge)
Weight	50-70g

Ultimately, we decided on acquiring the URGENEX 7.4 V Li-Po Battery with 1100 mAh capacity for both the cars and the finish line's battery supply. This decision was made with simplicity and consistency in mind, where the 1100 mAh capacity, 35 C rating, and 52g weight would work well for both systems and our desired runtime. Using this same product requires us to only get 1 battery charger as well, saving us from additional costs of having to get two chargers.

3.7.4 Voltage Regulation Technology Comparisons and Selection

Voltage regulation is an essential component in many electrical circuit applications, and voltage regulation will play a significant role in our system due to the varied voltage requirements of the electronic devices involved. Voltage regulators are used to maintain a steady regulated output voltage despite variations in input voltage or load conditions. Thus, the correct voltage regulator(s) will be needed to continuously satisfy the power requirements of all the connected devices in our system, ensuring efficient and consistent power supply.

There are two main types of voltage regulators, linear voltage regulators and switching voltage regulators. Linear voltage regulators are simple in design, easy to use, and provide constant output power by dissipating any excess voltage as heat. However, they are less efficient than switching regulators and do not perform as well when there are large voltage drops (major difference in input and output voltages). On the other hand, switching regulators are more efficient and they possess the ability to both step-up or step-down the voltage due to the inductors and capacitors used in their design to convert voltage levels. A few important considerations to keep in mind are the load regulation and the transient response of the voltage regulator(s). With the sudden changes in demand via the motors used in our IC cars, the voltage regulator must be able to respond quickly, accurately, and efficiently to meet the load requirements when changes in demand occur.

Linear Voltage Regulators

One of the core components of linear voltage regulators is that they utilize a pass transistor. This transistor, regulated by an operational amplifier, adjusts the regulator's resistance in response to changes in load current or input voltage, producing the desired stable output voltage. The result of that is a change in the base current of the transistor via the operational amplifier, ensuring that the desired output voltage remains constant. Again, it is important to note that linear regulators can only step-down voltages ($V_{in} > V_{out}$), and the excess voltage resulting from large voltage drops ($V_{in} - V_{out}$) must be dissipated as heat, which does significantly reduce the efficiency of the regulator.

There are two types of linear voltage regulators, those being series and shunt regulators. As their names imply, series regulators have the pass transistor in series with the load and shunt regulators have the pass transistor in parallel with the load. The series regulator configuration enables more efficient voltage regulation for high load currents, whereas the shunt configuration is less efficient, but does have the ability to absorb excessive input voltage, offering greater protection to the circuit for large voltage drop scenarios. Therefore, depending on the regulator's application, the correct choice of configuration can be very useful for maintaining proper line and load regulation.

While linear voltage regulators do have their drawbacks, the simplicity of design, straightforward usage, low output noise, and stable output voltage regulation make them a top choice in many electronic applications. Some of the most popular applications include battery-powered devices, power supply for microcontrollers, and precision measurement devices such as audio, video, or medical instruments due to their low noise properties.

Switching Voltage Regulators

One of the core components of switching voltage regulators is that they utilize a switch-mode power supply to rapidly switch the input voltage on and off, converting a voltage level to the desired output voltage level. Since switching regulators incorporate inductors, capacitors, and diodes, their incorporation is what enables the switching regulator to store and transfer energy to produce a stable output voltage level, instead of having any "excess power" go to waste such as the case in linear regulators.

These regulators can do that because of the inductive and capacitive properties of LC filters. By using inductive filtering, the regulator can smooth an on/off waveform into a relatively stable voltage and deliver consistent load current despite the on/off switching. The magnitude of that voltage is determined by the waveform's duty cycle, thus PWM (Pulse-Width Modulation) is used at high frequency to create a filtered signal that is pushed into the inductor-capacitor powered circuit, creating the desired fully stable and consistent output voltage. By using feedback signals, the PWM duty cycle can be adjusted in accordance with various load conditions.

Due to these features, switching regulators can step-up or step-down output voltages, can achieve efficiency in the ranges of 80-95% with large dropout voltages (greater input voltage levels), and provide a wide range of possibilities for configuring a variety of output voltages. Some of the most popular applications include battery-powered-devices and battery chargers, power supply and management for electronic devices, and telecommunication equipment.

The most common switching regulator types (converters) are Buck, Boost, and Buck-Boost. Buck converters can step-down voltages, Boost converters can step-up voltages, and Buck-Boost converters can either step-up or step-down voltages.

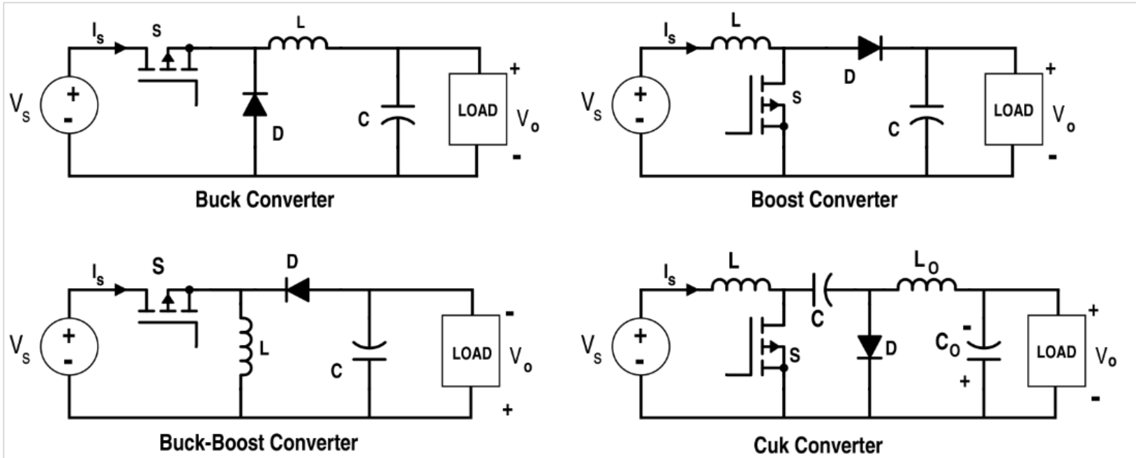


Figure 3.x. *Basic Converters Design (Analysis of DC-DC Converters in Equilibrium, All About Circuits technical article)*

Table 3.x. *Comparison Table of Voltage Regulators*

Specifications	Linear Regulators	Switching Regulators
Efficiency	Moderate	High
Design Complexity	Lower complexity with fewer components	Higher complexity with additional components
Thermal Performance	Poor due to more heat generation	Better due to less heat generation
Transient Response	Faster	Slower
Noise	Less	More (due to switching)
Input Voltage	Limited, needs input to be greater than output	Wide range, can step-up or step-down
Output Voltage	Fixed or Adjustable	Flexible
Cost	Lower	Higher

After investigation and comparison of the two types of voltage regulators, we selected switching regulation as our voltage regulation choice due to the higher efficiency, better thermal performance, and flexibility in voltages.

3.7.5 Voltage Regulator Comparisons and Selection

3.8 Modes of Communication

3.8.1 Communication Protocols Needs

In this project, we require a robust, real-time communication framework to control an internet-connected RC car remotely via a web interface. The communication setup must support the seamless transmission of control commands from the website to a local server, which will then relay those instructions to the RC car. This system should also facilitate live video streaming from the RC car's camera to the website, allowing users to monitor the car's environment and movements with minimal delay. The solution needs to be efficient, reliable, and capable of handling continuous data flow for both control inputs and video feedback, ensuring a smooth and responsive user experience regardless of the car's location on the network.

3.8.2 Communication Protocols Comparisons and Selection

Website to Local Server Communication

WebSockets

WebSockets is a protocol that enables full-duplex communication channels over a single TCP connection. It's ideal for real-time, low-latency interactions between a client (in our case a website) and a server. With full-duplex communication, WebSockets allow the client and server to send and receive messages at the same time without needing to establish a new connection. It is also designed for real-time communication, with reduced overhead compared to traditional HTTP polling or request/response models. Once a WebSocket connection is established, it stays open, making it ideal for continuous data streams like real-time control signals or sensor data. We can use WebSockets for real-time control of the RC car, allowing the website to send commands to the server and receive updates on the car's status with minimal delay. WebSockets, however, are not suitable for very large-scale data transmission (like video).

WebRTC

WebRTC is a protocol designed for peer-to-peer (P2P) communication between browsers and devices, enabling real-time video, audio, and data transfer without the need for plugins. WebRTC enables direct communication between the website and the RC car without going through an intermediary server, reducing latency. It has built-in support for video, audio, and data channels, making it ideal for live media streaming and sensor data, and automatically encrypts all data transmitted, providing added security. We can use WebRTC for streaming live video from the RC car's camera directly to the website, and it could also be useful for transmitting sensor data or control signals directly between the website and the car with low latency. WebRTC, however, is more complex to implement than WebSockets or RTSP and network traversal (NAT/firewall) can sometimes be complicated without the right configurations.

Local Server to MCU Communication

RTSP

RTSP is a protocol used for controlling the streaming of real-time data, such as video and audio. It's commonly used for streaming media from IP cameras or multimedia servers. RTSP can pause, play, and stop media streams, making it a popular choice for surveillance and media broadcasting. It is also optimized for low-latency streaming, making it a good choice for live video feeds. Lastly, RTSP is commonly used in systems that need to serve live media to multiple clients. In our project, RTSP can be used for streaming live video from a camera mounted on the RC car. The local server can then stream the video to the website for remote viewing. RTSP alone doesn't handle advanced media delivery mechanisms like peer-to-peer streaming (as seen with WebRTC) and is complex to set up for bidirectional data flow.

MQTT

MQTT is a lightweight messaging protocol used for IoT (Internet of Things) applications. It operates over TCP and is designed for constrained devices and networks, making it highly efficient for sending small, frequent messages. Clients can subscribe to topics, and when new messages are published to a topic, all subscribers receive the updates. This decouples the sender and receiver. MQTT is optimized for environments with limited bandwidth or intermittent connections and also keeps connections open, making it ideal for real-time sensor data transmission or control signals, especially for low-power devices.. In our project, MQTT can be used for sending control commands to the RC car or receiving sensor data such as speed and battery status. It's particularly well-suited if the car is transmitting small, frequent data updates. Compared to WebSockets, however, it is more complex to set up.

Bluetooth

Bluetooth is a short-range wireless communication protocol commonly used for pairing devices and transmitting data over short distances. Its effective range is around 10-100 meters depending on the Bluetooth version and is designed for low-energy usage, making it suitable for battery-powered devices. Bluetooth also provides encryption and pairing mechanisms to ensure data privacy. We can use Bluetooth for direct communication between the RC car and server, however, it can have less range than Wi-Fi and is not ideal for high-bandwidth uses like video streaming.

Wi-Fi (Both)

Wi-Fi (Wireless Fidelity) is a widely used wireless communication protocol that allows devices to connect to a local area network (LAN) and the internet. If the RC car has a Wi-Fi module, it could connect directly to the local server over the local network and then using Wi-Fi again, transmit data over the internet to the remotely hosted website. This setup would allow the server to control the car over standard TCP/IP. Wi-Fi supports high-speed data transfer, with the latest standards (Wi-Fi 6) offering up to 9.6 Gbps, has a range of around 30-50 meters indoors, and its bandwidth is good for general communication and video streaming. Latency depends on the network but is generally low in a local network setup. The RC car can connect to the local server through Wi-Fi, allowing real-time control and data transmission. A Wi-Fi connection is stable enough for sending control commands and possibly transmitting video if the bandwidth is sufficient. Wi-Fi, however, has range limitations and can have potential interference from other devices on the same network. In our case, range should not be an issue.

Table 3.x. *Communication Modes Comparison Table*

Mode	Type	Strengths	Weaknesses	Best Use in Project
Wi-Fi	Wireless (LAN)	High-speed, wide coverage, stable for data and control	Interference, limited range	Connecting RC car to local server and internet
WebSockets	Real-time communication	Low-latency, full-duplex, persistent connection	Requires WebSocket server setup	Sending real-time control commands
RTSP	Media streaming protocol	Low-latency video streaming, scalable	Not for bidirectional control	Streaming live video feed from RC car
WebRTC	P2P real-time communication	P2P, low-latency, supports video, audio, and data	More complex setup, NAT traversal issues	Live video and data streaming between car and website
MQTT	IoT messaging protocol	Lightweight, low bandwidth, publish/subscrib e model	Not suited for large data transfers (e.g., video)	Transmitting small data like sensor readings
Bluetooth	Short-range wireless	Low power consumption, secure	Limited range, lower data rates	Short-range control or monitoring without internet

There are several modes of communication that we will be using in our project. The first is Wi-Fi, which is crucial for connecting the RC car to a local server and enabling internet-based control. Next, we will use WebSockets which are excellent for real-time control commands between the website and server. Finally RTSP would be used for a low-latency video stream from the RC car to the website. With the combination of these modes of communication, we hope to achieve relatively low latency.

3.9 Antennas

For our internet-controlled racing system to be fully realized, there will need to be a wireless communication system in place for the cars to communicate with the server for both video transmission and motor controls, and for the finish line to communicate with the server for data transmission from the races. To implement this system, both the cars and the finish line need to be outfitted with compatible antennas to enable wireless communication to the server.

3.9.1 Antenna Technology Comparisons and Selection

Upon initial investigation into the different antenna technologies available for Wi-Fi connectivity applications such as this project, we discovered that there is a wide variety of technologies that come in many shapes and sizes, each with their own unique characteristics and advantages for their intended purpose. Those technologies range from the traditional dipole and monopole antennas to more advanced technologies such as MIMO (Multiple Input Multiple Output) and fractal antennas. For the purposes of this project though, the size constraints and overall scope of this project allowed us to specify our search to patch antennas which then lead into research of integrated antennas with chip antennas and trace antennas as potential options.

Patch Antennas

Patch antennas consist of flat, rectangular or circular sheets, “patches”, of conductive metal that are mounted on a dielectric substrate. The patch is suspended over a larger ground plane, separated by the dielectric material between the two sheets of metal, to form the conductive layer.

The working principle behind these antennas is that when a RF signal is applied to the patch through a feeding line, the patch gets excited, resonating at its natural frequency. These electromagnetic waves being emitted from the patch create a specific radiation pattern, and this directional radiation pattern typically has maximum gain being in the direction perpendicular to the surface of the patch. The feed line is either by a microstrip feed that is connected directly to the edge of the patch, or by a coaxial cable typically connected through a hole in the ground plane. Thus, for a desired specific frequency, the design dimensions of the patch, the dielectric substrate, and the feeding mechanism can be calculated to properly influence the natural resonance frequency to then be the desired resonant frequency, allowing efficient radiation of electromagnetic waves for the intended application.

These antennas are often referred to as microstrip antennas, and with their slim, lightweight, simple, and compact design, they are a suitable choice for many compact applications where size is a critical factor. Due to this compact form factor, patch antennas are easily integrated into devices’ circuit boards, making them a very popular choice for many modern electronic devices. With that, they are relatively easy to customize and can be fabricated directly into a PCB.

Integrated Antennas

For integrated antennas, they are typically some form of patch (microstrip) antennas purposefully designed to fit the contours of a PCB, directly integrated into the PCB during the manufacturing process. However, in the realm of integrated antennas, there are also slot antennas, antennas specifically designed as slots cut in the PCB, there are trace antennas, antennas made by printing traces directly onto the PCB, and there are chip antennas, ceramic antennas that are soldered onto the PCB.

With patch antennas specifically, using this PCB integrated method has many advantages, such as efficiency of space, improved reliability, optimized performance, ease of integration, and cost-effectiveness. The main advantages being space efficiency with the minimized physical footprint of the antenna enabling a more compact design, improved reliability via less external connections that incurs a reduced risk of failure, and integrated antennas having more of a seamless connection between the other devices on the PCB.

Trace Antennas

Trace antennas are created by etching or printing out conductive copper traces on the PCB, serving as the antenna element. Similarly to patch antennas, they are designed with the desired resonance frequencies in mind, meaning that size and shape of an antenna is determined by the desired resonant frequency, influenced by the length, width, and pattern of the trace. These trace antennas come in different types such as monopole, dipole, or meandered designs for different patterns and configurations.

These trace antennas give a cost-effective and compact solution for integrating wireless connectivity directly into the PCB design. The key advantages of a PCB trace antenna are its low cost, space efficiency, and ease of integration into the PCB manufacturing process. That is because the antenna is a part of the PCB itself, having no need for any additional or separate components, while also reducing overall complexity and material costs.

Chip Antennas

Chip antennas are very small antennas that are designed to be connected to a PCB via soldering. They are typically made from ceramic or plastic in some cases, with conductive elements included within them to facilitate RF communication. They possess many of the same advantages as the patch integrated antennas but differ in the sense that they are soldered into the PCB as a surface-mounted component instead of being directly integrated into the PCB during the manufacturing process.

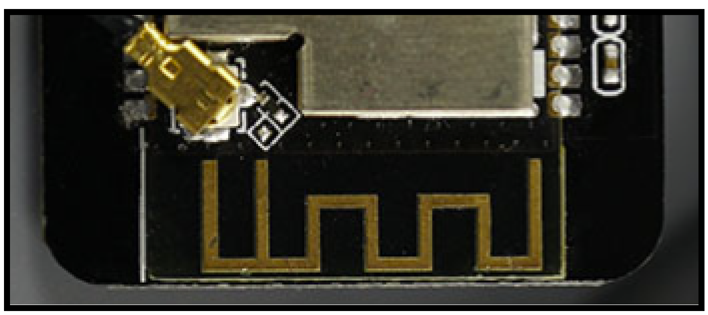
These chip antennas give a pre-manufactured option that offers a compact solution for wireless communication. It is a ready-to-use device that is soldered directly onto the

PCB, offering easier integration than a custom PCB antenna. Overall, chip antennas are another popular choice for applications when space is very limited, with good performance metrics even with their smaller size. It is important to note that impedance matching and placement of the antenna is critical to integrate with the rest of the circuit, and the antenna's specifications must match the performance requirements.

3.9.2 Antenna Part Comparisons and Selection

Antenna Type	Range	Integration Complexity	Cost-Effectiveness	Suitability for ESP32
Patch Antenna	Moderate to High	Moderate	Moderate	Good
Trace Antenna	Moderate	Easy	High	Excellent
Chip Antenna	Moderate	Easy	Moderate	Good

Figure 3.x. *Trace Antenna on ESP32*



We chose to use a trace antenna due to its range and compatibility with the ESP32 board, which we have used in testing. The trace antenna offers slightly more range than a chip antenna and is integrated directly into the ESP32 board. This setup minimizes additional hardware and maximizes performance within the project's scope, making it the most practical choice for our application.

3.10 Development Environment: Languages and Repositories

3.10.1 Programming Languages Comparisons and Selection

C

C is a compiled, procedural programming language known for its simple and concise syntax. It is considered a general-purpose language regularly used in systems programming, embedded systems, and other applications where performance is crucial. C is a low-level language, meaning it is closer in syntax to assembly and machine code than other languages, and also includes a variety of standard libraries that can be utilized in many different applications. This gives C many advantages when it comes to performance, portability, and control. Since C is a close-to-hardware language, it is extremely fast and efficient, and is important for real-time applications (such as building an internet-controlled car racing system). This makes C ideal for programming microcontrollers (MCUs). Because many MCUs have limited processing power and memory, the efficiency of the code is important for controlling multiple peripherals in a timely manner. C is also a highly portable language with a consistent syntax across many different platforms. Most MCUs on the market have a C compiler available, making it easy to implement in a variety of applications. C also provides high-level abstractions, such as functions and structures, that help developers write clean and maintainable code that is less dependent on the specific hardware specifications of a platform. Lastly, C provides a high level of control over system resources. It allows for precise control over memory allocation, hardware registers, and peripherals. However, due to its complexity with pointer arithmetic and manual memory management, as well as its lack of safety features compared to Java and Python, programming in C can make development times longer and can result in complicated and error-prone code. Despite its complexity, Cs advantages would make it a great choice for programming the MCUs in our project.

Java

Java is a compiled, object-oriented programming language designed to have a high degree of platform independence. It is regularly used in enterprise-level and web applications that take advantage of its object-oriented features. Java is a high-level language with a higher level of abstraction compared to C. This gives it a few advantages over C in terms of platform independence, object-oriented features, and robustness. Platform independence refers to the ability to run on many different devices or operating systems. Java accomplishes this by compiling the Java source code (.java files) to an intermediate form called bytecode (.class files) which can be run on any device with a Java Virtual Machine (JVM) installed. Java also includes a diverse set of standard libraries that developers can utilize without worrying about cross-platform compatibility issues. With its object-oriented nature, Java allows for modular and reusable code using concepts such as inheritance, encapsulation, and polymorphism. This typically results in shorter development times and code that is easier to read and write. Lastly, Java is a robust language, providing built-in memory management and error handling. The JVM can deallocate memory that is no longer in use (garbage collection) and can handle exceptions by using the try, catch, and finally blocks. Java, however, with its automatic garbage collection, introduces additional overhead which makes it notably slower than C in many cases. It is also considered a “verbose” language, generally requiring more lines of code than a comparable program written in Python. Lastly, the JVM consumes significantly more memory than C for example, making it less ideal for the limited memory present in many MCUs. Overall, Java provides many advantages for enterprise-level applications and may be useful in developing the web application for our project.

Python

Python is a high-level, interpreted programming language designed with an emphasis on code readability and developer productivity. It is a general-purpose language regularly used in web development, data science, and automation. Python has many advantages compared to C and Java, including its ease of use, versatility, and its extensive community support. Python's syntax is the most simple and clean out of the three languages described in this section. This makes it an easy language for beginners to learn how to program and allows for rapid development compared to most other languages. Like Java, memory management in Python is handled automatically with a garbage collector, reducing memory leaks and pointer-related errors. Python also uses a feature called dynamic typing, where the developer doesn't need to specifically declare a variable's type. Instead, a variable's type is determined at runtime, reducing the amount of code required and making the code easier to read. Python is also noted for its versatility, providing an extensive standard library and other frameworks, as well as supporting several programming paradigms, such as procedural, object-oriented, and functional programming. Python also integrates well with other languages like C and Java through the use of tools such as Cython and Jython. This allows developers to combine the ease of use of Python with the performance of lower-level languages. Lastly, Python's community is massive and continues to grow everyday. Developers can benefit from an assortment of external libraries and frameworks and can easily get support through the vast amount of online communities and forums. These advantages do come at a significant cost, however. Because Python is interpreted at runtime instead of compiled, it is generally much slower than both C and Java. Python's dynamic typing may also introduce runtime errors that would have been caught in statically typed languages when compiling. Overall, with its ease of use and extensive set of libraries and community support, Python is an excellent language that can provide functionality that other languages cannot.

Table 3.x. *Languages Comparison Table*

Feature	C	Java	Python
Type	Compiled, Procedural	Compiled, Object-Oriented	Interpreted, High-Level, Dynamic
Performance	High (Close to hardware)	Moderate (JVM overhead)	Low (Interpreted, dynamic)
Memory Management	Manual	Automatic (Garbage Collection)	Automatic (Garbage Collection)
Ease of Use	Complex	Moderate	Easy
Platform Independence	Low (Depends on compiler)	High (JVM allows cross-platform)	High (Python interpreter available on most platforms)

Application Areas	Systems Programming, Embedded	Enterprise, Web Applications	Web Development, Data Science, Automation
Syntax	Low-level, concise	High-level, verbose	High-level, concise
Error Handling	Low (Prone to memory-related errors)	Moderate (Checked exceptions)	High (Dynamic typing, extensive standard library)
Development Speed	Slow (Low-level, verbose)	Moderate (Higher abstraction)	Fast (High-level, concise)
Community and Ecosystem	Large but specialized	Large and corporate-friendly	Massive, diverse, and growing

We will be using C as the primary programming language for our project. We considered using Python due to its ease of use, however, after researching the differences between each language, C presents the most advantages for our project. For an internet controlled RC car, where real-time control, efficient memory use, and direct hardware access are crucial, C is the best choice.

3.10.2 Repositories Comparisons and Selection

GitHub

GitHub is a cloud-based code hosting platform with an intuitive and easy-to-use interface. It's mainly used for organizing and storing open-source projects, collaboration within groups of developers, and code reviewing. Some key features of GitHub include free public repository hosting, collaboration and review tools, GitHub Actions, and integrations. With GitHub, users can host public Git repositories that make it easy to track changes in an application's development, collaborate with other users, and manage their code, all while being completely free. This has made GitHub the largest platform for open-source projects. GitHub provides many features that help users collaborate on a project. Branch management allows users to "branch off" of the main code in their project and make changes or add additional features while keeping their main branch intact. The developers can then merge the branch into the main branch, if they choose, after a code review. Pull requests allow other developers to work on a project and propose changes to the project's owner. If the owner likes the changes, they can then approve the request and merge it into the main branch of the repository with a single click. Also, users can track and manage bugs, feature requests, and other tasks with Issues, and have project-related conversations with Discussions. GitHub Actions is their Continuous Integration/Continuous Deployment (CI/CD) tool that allows developers to automate tasks such as building, testing, and deploying code directly from a repository by using workflows. Actions, however, is less flexible for complex CI/CD pipelines compared to GitLab. Lastly, GitHub has a vast ecosystem of third-party services and tools and a large marketplace of apps and extensions, providing developers with all the tools they may need for any type of project. Overall, GitHub is a great choice for smaller teams or individual developers looking for a simple platform with a focus on collaboration and open-source projects.

GitLab

GitLab is an end-to-end DevOps platform that is available as both cloud-based and self-hosted versions with a free and several paid tiers. GitLab is similar to GitHub, as it allows users to host Git repositories, includes collaboration and code review features, and many integrations. However, unlike GitHub, GitLab has an integrated CI/CD pipeline, DevOps integration, and a self-hosting option. GitLab's repositories include full version control capabilities like GitHub, but provides more flexibility for private repositories in the free tier. GitLab includes merge requests (similar to GitHub's pull requests) and a more advanced issue tracking system with labels, milestones, and assignees. GitLab also integrates with many third-party tools, however, it generally requires less integrations because it provides more built in features. Unlike GitHub, GitLab's CI/CD capabilities are directly built into the platform and provide support for parallel builds, triggers, and more complex workflows compared to GitHub Actions. Designed as an end-to-end DevOps solution, it covers the entire software development cycle with features such as Container Registry, Package Registry, Infrastructure as Code, Monitoring, and Security Testing all within a single platform. GitLab also provides a feature with automatic CI/CD pipelines that require minimal configuration called Auto DevOps that makes it easy for teams to get started with DevOps practices. Lastly, one of GitLab's biggest advantages is that it can be self-hosted, meaning it can be run on a company's private servers, and in turn, provides full control over their code and infrastructure. Overall, GitLab excels in enterprise environments and for teams that need more advanced CI/CD, security, project management features and the ability to self-host.

For our development repository, we will be using GitHub to store and share our code. Due to its very large community, easy to use interface, and the fact that we are already familiar with Github, it is the obvious choice.

Chapter 4 - Related Standards and Design Constraints

4.1 Industrial Standards

4.1.1 PCB Design Standards

Printed Circuit Boards (PCBs) are integral components of virtually all electronic devices. To ensure consistency, reliability, and safety in the design and manufacturing of PCBs, the IPC (Association Connecting Electronics Industries) has developed a comprehensive set of standards that are widely used across the electronics industry. These IPC standards cover various aspects of PCB design, performance, and assembly, helping manufacturers create high-quality products that meet industry requirements.

One of the most important IPC standards is **IPC-2221**, which serves as the generic standard for PCB design. It provides guidelines on essential elements such as material selection, electrical characteristics, mechanical properties, and thermal management. By adhering to IPC-2221, designers can ensure that their PCBs meet the necessary mechanical strength and electrical performance criteria while balancing factors such as manufacturability and cost-effectiveness. This standard is fundamental for creating reliable PCBs that function well in a wide range of applications. The IPC-2221 standard acts as an umbrella standard for several other sub-parts labeled as IPC-2222 through IPC 2226. Here we will see what makes up IPC 2221 and its importance in the industry.

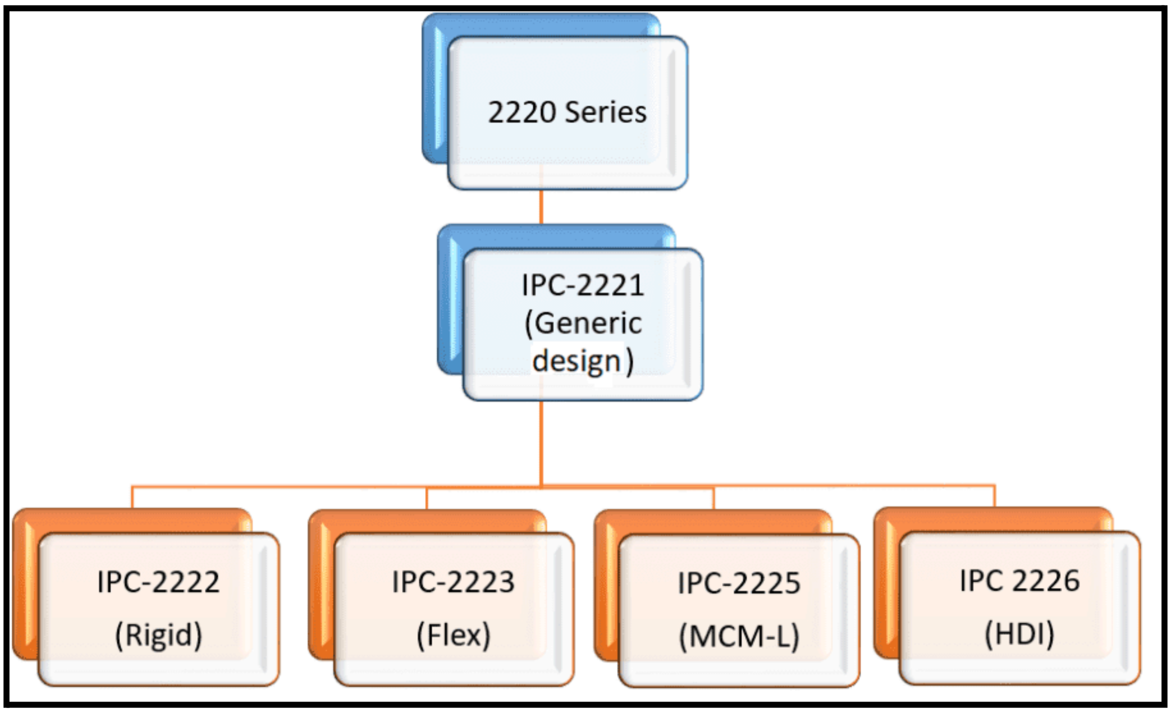


Figure 4.1. *IPC Standards Hierarchy*

IPC-2222 covers the detailed requirements for designing rigid PCBs, which are widely used in electronics where mechanical stability and long-term reliability are essential. It discusses material selection conductor characteristics, trace width and spacing, pad design, and signal routing. The standard addresses the electrical properties such as impedance control, signal integrity, and thermal management for rigid boards. It also emphasizes design rules that ensure manufacturability, addressing tolerances for features like drilled holes and plated-through vias. Designers are encouraged to consider mechanical stresses such as vibration and bending forces to prevent structural failure. IPC-2222 also highlights environmental factors like temperature cycling and humidity exposure, guiding material choices and surface finishes for boards used in demanding conditions. It also focuses on specific design constraints, such as current-carrying capacity and dielectric layer thickness, which directly affect board performance. This standard is critical for any designer working with rigid PCBs in applications like consumer electronics, industrial control, or automotive systems

IPC-2223 defines the design rules for flexible PCBs, which are used in applications requiring bendability, such as wearables, medical devices, or space-constrained electronics. It covers material properties specific to flex circuits, including polyimide substrates and flexible adhesives, which enable the board to bend or fold without damaging the conductors. The standard provides guidelines on layer stackups for single-sided, double-sided, and multilayer flex circuits. The document also addresses critical issues like bending radius, mechanical stress endurance, and the placement of components on flexible sections. IPC-2223C (a revision) introduces design considerations for rigid-flex PCBs, combining rigid and flexible sections. It focuses on balancing flexibility with the mechanical stability of the rigid sections and preventing cracks or delamination during operation. The standard also emphasizes strain relief techniques around flex areas to increase durability.

IPC-2224 focuses on High-Density Interconnect (HDI) technology, used for compact PCBs with high routing densities, such as those in smartphones, tablets, and other portable devices. HDI boards use microvias, which are much smaller than standard vias, allowing for more connections in limited space. The standard covers the design of microvias, stacked and staggered vias, blind and buried vias, and via-in-pad technology. The document highlights methods for minimizing signal integrity issues, such as crosstalk and signal delay, that can occur due to densely packed traces. It also emphasizes thermal management, providing techniques for efficiently dissipating heat in high-power applications with limited board space. IPC-2224 further discusses routing strategies for handling high-speed signals in multilayer HDI boards, focusing on maintaining signal integrity across the complex layer stackups.

IPC-2225 is the standard for organic chip carriers, which are substrates used to package integrated circuits. These chip carriers serve as intermediates between the die and the PCB, often used in advanced semiconductor packaging. The standard covers material choices like organic laminates or BT (Bismaleimide-Triazine) resins, which offer excellent electrical insulation and thermal performance. It also provides guidelines for designing interconnects between the chip and the carrier, including fine-pitch routing, ball grid arrays (BGAs), and leaded packages. Designers are instructed on ensuring dimensional stability of the carrier to prevent warping, which is critical during soldering and assembly processes. The standard highlights signal integrity, emphasizing low dielectric constant materials and optimized trace geometry to support high-speed data transmission, which is critical in applications like CPUs, GPUs, and network processors.

IPC-2226 deals with multilayer PCB designs, providing guidelines for designing boards with multiple stacked layers to accommodate complex circuits with high component density. The standard covers layer stackup design, ensuring that each layer maintains proper signal routing, power delivery, and ground planes. Special attention is given to impedance control and crosstalk minimization between adjacent layers, critical for boards handling high-speed digital signals. The standard discusses via types, including blind, buried, and through-hole vias, which allow interconnection between the layers without compromising board integrity. IPC-2226 also provides guidelines for managing thermal dissipation in multilayer boards, which can be a challenge due to the heat buildup in densely populated boards. The standard covers testing and quality control to ensure that the multilayer boards are free from defects such as delamination or short circuits, which could occur during the complex manufacturing processes required for these boards.

Together, these IPC standards help ensure that PCBs meet the necessary quality, performance, and reliability requirements. By adhering to IPC guidelines, manufacturers can create PCBs that are not only functional but also durable, safe, and environmentally responsible. IPC standards provide a common language and set of expectations across the industry, enabling efficient production and global interoperability in electronic devices.

4.1.2 Hardware Specific Standards

Connector and Wiring Standards

The use of connectors for data transmission and power distribution will follow **IEC 60603-7** standards, which define requirements for connectors, particularly those used in Ethernet and similar high-speed digital applications. This ensures reliable signal transmission and mechanical durability in connectors that will be subject to repeated insertions and extractions.

All wires selected for the PCB will comply with **AWG (American Wire Gauge) standards**, ensuring that wires can safely carry the current required by each subsystem without risk of overheating or voltage drop. For example, power lines carrying higher currents to motors or battery connections will use thicker gauge wires to minimize resistive losses and prevent overheating.

Power System Standards

Voltage regulators will be chosen to meet the requirements for power efficiency and voltage stability. Thus, adhering to **IEEE 1625** standards, which govern the design of efficient power systems for portable devices, will ensure that voltage regulators are able to efficiently step down or step up voltages to supply the correct power to the microcontroller and peripheral devices. This is particularly important for projects like this one, where minimizing power loss is essential to extend the battery life.

Motor and Motor Driver Standards

The motors used in the project must comply with **NEMA (National Electrical Manufacturers Association) standards**. These standards define motor specifications like voltage, power ratings, and safety, ensuring that the selected motors can operate within the electrical constraints of the system without risk of overheating or failure. NEMA MG 1-2011 guidelines, for instance, define the mechanical and electrical parameters of motors, including the protection ratings (IP ratings) and thermal management requirements.

For motor control, **IEC 60034-1** standards will be referenced to ensure the correct motor protection mechanisms are implemented. This will cover factors such as overvoltage protection, temperature sensors for overheating prevention, and the selection of motor drivers that can safely handle the current required by the motors.

The selected motor driver, for example, the DRV8833, will be able to handle the voltage and current required to drive the motors while providing protection against overcurrent. The motor driver's design must also account for thermal management, ensuring that heat generated by the driver does not exceed safe operating limits.

4.1.3 Communication Protocol Standards

Selecting the appropriate standards is important in developing a reliable and functional system, especially when it comes to communication and interoperability. One of the more relevant standards for our project is the **IEEE 802.11 (Wi-Fi) Standard**, which defines protocols for wireless local area networks (WLANs). This standard will play a role in enabling our RC car to communicate with the local server, process control commands, and transmit live video stream data.

The primary benefit of the IEE 802.11 standard is its role in enabling Wi-Fi communication. It ensures that devices connected to a network can communicate seamlessly, without interference or signal loss. The 802.11 standard ensures reliable video streaming, data transfer, and real-time interactions, even in environments where many devices are competing for bandwidth. By adhering to this standard, systems can maintain high-quality connections, important in systems that demand low latency or high data throughput.

Another key advantage of the IEEE 802.11 standard is its compatibility across devices. This standard has been widely adopted by manufacturers of routers, computers, smartphones, and other connected devices, ensuring that systems built with it can interact with a wide range of hardware. Interoperability is essential in connecting devices from different manufacturers and allowing them to communicate seamlessly over a common network. The 802.11 standard guarantees interoperability between devices, making it a foundational standard for any modern wireless communication system.

Security is also a crucial aspect of the IEEE 802.11 standard. With cyber security becoming increasingly important every year, it is important that all wireless communication systems include robust security measures. The 802.11 standard includes encryption protocols such as WPA2 and WPA3, which help protect data from being intercepted or tampered with during transmission. This level of security is crucial in ensuring that sensitive information remains private and only authorized users can access the network. By adhering to these security protocols, systems can ensure that their wireless communication is safe and reliable.

The IEEE 802.11 standard also allows for performance optimization. The 802.11n, 802.11ac, and 802.11ax (Wi-Fi 6) for example, provide options for enhancing network performance based on specific needs. Wi-Fi 6, specifically, offers increased speed, reduced latency, and improved performance in congested areas, making it ideal for high-demand applications like video streaming or data-heavy environments. By selecting the appropriate version of the 802.11 standard, systems can be optimized for the best possible performance, depending on their specific requirements.

To conclude, the IEEE 802.11 standard plays an important role in modern wireless communication systems. It ensures reliable Wi-Fi connectivity, broad device compatibility, robust security, and performance optimization. Adhering to this standard in our project will improve our system's functionality and ensure that it operates within recognized and established practices in wireless communication technology.

4.1.4 Software Specific Standards

The **ISO/IEC 9899:2018** standard, known as **C18**, is the latest international standard for the C programming language, enhancing its efficiency and portability. This standard updates syntax, libraries, and features to ensure that C code remains consistent across different systems, improving portability and stability in embedded and cross-platform applications. C18 is particularly relevant to performance-critical applications requiring low-level hardware control, such as embedded systems.

In our project, ISO/IEC 9899:2018 directly impacts the development of C-based code for the ESP32 microcontroller, which handles real-time communication with the RC car hardware. Adherence to C18 guarantees that our code is compatible across compilers and systems, ensuring reliable operation on various hardware. With C's low-level access to system memory and hardware functions, following this standard optimizes memory and processing efficiency, both essential for responsive RC car control.

Additionally, ISO/IEC 9899:2018 supports modern, secure programming practices, helping prevent common vulnerabilities like memory leaks and buffer overflows. This reliability is crucial for uninterrupted communication between the server and RC car, with industry-accepted best practices making the code safer, more efficient, and easier to troubleshoot.

4.2 Practical Design Constraints

4.2.1 Time

The most daunting type of constraint that we will need to manage for this project is the time constraints of Senior Design. With this project being a senior design project, we must adhere to all deadlines that are imposed by the senior design coordinators, effectively going through the engineering process in its entirety, and delivering a demonstrable product by the end of the second semester of Senior Design.

The first major challenge will be the time constraint of having only two school semesters to make our product. Throughout each semester there are many project milestones and deadlines that we must meet on time, meaning that there will be no room for any time extensions. With these hard deadlines and firm requirements to complete our project and pass this course, effective time management throughout the entirety of these two school semesters is of critical importance. With the use of effective time management, it will enable us to fully define our project's scope, research our project's components, design our product's system, build and test our prototype, and ultimately deliver a finished product that will be demonstrated to the senior design coordinators and our review committee, all in just two school semesters.

Specifically with Senior Design 1, the biggest challenge will be fully writing and completing a 120-page final report in under three months, from our project's approval in the Divide & Conquer meeting. We are required to complete the project definition/description, research, system hardware and software design, system fabrication, and system testing, all in that limited timeframe. To counter this daunting challenge, we have utilized a kanban board for our project's report, where we've effectively broken down these requirements into achievable steps, created realistic timetables for every task, and then delegated tasks amongst each other to work together in reaching our main goal of submitting a complete, 120-page length, final report of our project.

Another challenge that starts with Senior Design 1 and will carry over into Senior Design 2, is the potential time constraint that can arise once we finish our parts selection and begin ordering parts. From previous senior design projects, there is a realistic possibility that there could be delays in acquiring our parts, especially PCBs. Whether it be supply shortages or something else, we will need to factor in that possibility when planning our project's development, testing, and assembly.

Towards the end of Senior Design 1 and throughout Senior Design 2, one of the challenges we will encounter is the extensive testing phase of this iterative engineering design process. During this design process, especially in assembly and testing, we will most likely encounter setbacks, problems, errors, etc. that must be addressed and corrected in a timely manner en route to completing our product. Thus, with the time constraints involved with Senior Design, we must be prepared and ready to respond accordingly to the unexpected obstacles that we will encounter, or else the project's overall success will be in jeopardy if we cannot complete our product in time.

With all the time constraints involved with this project, it is imperative that we as a group of engineering students maintain integrity and go through the entirety of this project in a disciplined approach. This disciplined approach means that we must not rush any part of this process by taking shortcuts or anything like that. Instead, we must remain disciplined and complete our work in a step-by-step manner en route to our end goal.

To help mitigate these time constraints, we elected a team leader (Austin Silva), to effectively plan out our project's timeline and manage the team's kanban board on a weekly basis to ensure we remain on track as a team. Then, some of the ways we have kept each other accountable are by multiple weekly team meetings where we update each other of what we have individually accomplished each week, discuss any problems or issues encountered, and plan out the next week's objectives. Our team leader has facilitated each team meeting and has overseen management of our project's progression, encouraging clear communication and collaboration across all team members. Thus, through our team's time management and project planning, we will be able to overcome these time constraint challenges and successfully complete this project on time.

4.2.2 Economic

The next biggest type of constraint for this project is the economic constraints of our project's budget. Thankfully, one of our team members was able to get in contact with a donor interested in our project's internet-controlled car racing feature. The donor provided us with $200 for our project, and with our project's scope, we decided to create a fair project budget of $350. With this group-imposed economic constraint, we set our project budget in that way to push ourselves to balance cost-effectiveness and product performance by making engineering tradeoffs to ensure our project stays in our budget and meets all our engineering requirements.

With these economic constraints, we need to take our technology research and use the constraint to help us determine the best match between the costs of parts and optimal performance of parts for our project's needs. Though, we also must create space in our budget for multiple iterations of the design process, setting aside a portion of funds for if we need to purchase part replacements or exchange parts during the testing and assembly phase.

One economic constraint during parts selection has been in the selection of the cars' camera. After researching the various technological aspects of FPV cameras for RC cars and drone racing, the top-quality cameras can easily reach $50 or more with high-end features such as greater resolution, superior low-light performance, and a bigger field of view than the more beginner and mid-range cameras. With our project including two cars, meaning we need two FPV cameras, we'll need to effectively balance the performance metrics and technological aspects of the camera options available to us to make a smarter and more cost-effective selection that stills serve our project's needs, instead of simply going for the "best" option on the market without factoring in costs.

Similarly to the cameras, another economic constraint during parts selection has been in the battery selection for the cars. When investigating the NiMH and Li-Po battery products on the market for RC car racing applications, there are major cost factors that need to be considered. The battery packs alone have a price range of $20-$60, and an associated battery charger will be needed as well which can easily add another $20-$60 to the project. Therefore, this economic constraint has forced us to make some tough decisions in determining the capacity of our cars, choosing a suitable level of capacity for the performance of our car racing project.

One more major economic constraint is the PCB requirement of this project, with two cars and a finish line system, our project will need three custom PCBs. Each PCB is already a significant expense on its own, but with three PCBs needed, it is imperative that we effectively design these PCBs to smoothly integrate the components of our car and finish line systems. That way we can avoid having to re-order PCBs (which would be another time constraint) if the first design does not work as intended.

Overall, the economic constraints of this project have proved to be quite a challenge since it forces us to make the most economical-based decisions in selecting the parts and products to use for meeting our project goals, instead of simply going for the highest quality parts and products on the market. In this project, and in the engineering industry, economic constraints are a critical factor in the engineering design process since it effectively scales the project to its intended level of performance by limiting the amount of funds and resources available for the project through a budget.

4.2.3 Sustainability

The sustainability constraint of this project is another one to keep in heavy consideration as we go through the design process. Since our project does revolve around racing, we will need to ensure that there is sufficient protection for the critical components of our cars and that they are durable enough to withstand any crashes or impacts on the racetrack.

Again, due to the nature of our project's racing setup, the sustainability of our cars is a major component for delivering a successful product. If this aspect of engineering design were to be ignored or not taken seriously, then the product could easily become damaged or broken at any time the product is in use. With one of our engineering specifications being directly based off sustainability, *car chassis lasting for 100 runtime hours*, we decided to make this constraint a top priority in our design process and testing phase.

The car's critical components will need to be thoughtfully placed in the car's body, especially the battery supply, to ensure there is no possibility for the battery supply to be damaged by external factors such as a car crashing into another one or a car running into the boundaries of the racetrack. Following that up, it's imperative that these components are secure in the car's body as well, where the risk of them becoming loose or detached must be mitigated. Then, specifically for the chassis and body of the cars, sufficient research into similar products and materials will be needed to determine the right selection for the optimal durability.

Overall, this project does have factors that will affect the durability and sustainability of our product, however, due to the scale of our project and careful design process, we are confident that we will be able to deliver a product with a sufficient lifespan to meet our engineering specifications and goals.

4.2.4 Remaining Constraints

4.2.4.1 Environmental

An environmental constraint for this project relates to the racetrack and the environmental conditions needed to set up the track for our project's demonstration and usage. The first set of conditions needed will be an indoors, hard and flat surface with sufficient surface area to fully set the outside boundaries of the racetrack. Then, the boundary lanes, barriers, and obstacles of the racetrack will need to be set up within that space for the track's design to be fully implemented. Lastly, we will need to configure the finish line with the track and do a couple of test runs to ensure the racetrack is ready for optimal racing.

This constraint is less of a factor than the other major constraints due to the ease of access we should have in setting up the racetrack. It is still a constraint worth factoring into our racetrack's design though, since it will help us to make more informed decisions in determining the size and scale of the track to best fit our project's scope and the environment we will be using.

4.2.4.2 Ethical

A component of the project's design and development process that needs to be kept in mind is the ethical responsibility that we possess as a group of engineering students. As a group of students, we must make sure to take the needed care to fully research and reference the work of professionals as we learn about the different parts of the engineering design process. From that, we will apply the knowledge we gain from these resources into our project's design process and not simply copy the work of another author. It is a simple responsibility, but one that must be upheld and one that we as a group take very seriously.

Another ethical constraint is the materials and parts we will be using for our project. For the racetrack, we will be prioritizing recyclable materials for the boundaries, obstacles, and barriers, to the best of our ability. At the end of the project's completion, we will be sure to properly recycle or re-use the materials for another use instead of letting them go to waste. For the cars and the finish line, we will be sourcing our parts from trusted companies only, with a proven track record of ethical compliance and responsibility. Overall, this ethical constraint does factor into the time constraint as well, with the additional research of the products and parts we will be using.

4.2.4.3 Social

Another constraint of this project is the social barrier that can occur when users are racing against each other. Our project is being designed to have a web-based user interface that will allow a user to race against another user while providing real-time racing stats throughout the duration of a race. Thus, since our project is not being designed to include a social application or feature for the users, it can be a potential drawback for users since it does differ from the traditional RC car racing experience, where users can easily communicate with each other by racing each other in person.

Due to this social constraint, we are going to highly recommend that users utilize social applications such as discord or zoom, or even something as simple as a phone call when using our product, that way users can further enjoy their racing experience when racing against one another.

4.2.4.4 Health and Safety

Health and safety are a critical priority in everyday life and this project will adhere to all safety practices applicable. The most significant health and safety constraint is the battery supply for our cars, where the proper measures will be taken to protect the battery packs from external damage and all the charging requirements will be strictly followed to maintain battery integrity to prevent the possibility of thermal runaway. Another safety precaution is for the racetrack moderator, where the outside boundaries of the racetrack must be firmly set in place to prevent the cars from going off track and potentially running into the moderator.

4.2.4.5 Manufacturability

The last constraint to keep in consideration for this project is the manufacturability constraint. Given that this is a project done by undergraduate engineering students, we are inherently limited to the Senior Design lab's resources. This in turn does limit us from the amount of manufacturing and construction that can be accomplished in our project's development.

With this constraint, our group has scaled our project's design to account for the resources at our disposal, opting to go for a simple yet effective mechanical design that allows us to assemble our product with the tools available to us. Instead of manufacturing the pieces of our project, we decided to purchase mechanical pieces and then assemble them together in accordance with our design. However, for the electrical components of our design such as the PCBs, we will need to have them manufactured.

Chapter 5 - Comparison of ChatGPT

ChatGPT is an advanced language model developed by OpenAI that can understand and generate human-like text. It's designed to assist with a wide range of tasks, from answering questions and providing explanations to engaging in conversation and assisting with writing tasks. The model uses machine learning techniques to generate responses based on the context of the input it receives, making it an incredibly useful tool for research in this project.

However, it is important to verify the results of ChatGPT with the use of a search engine, such as google or Microsoft edge, because ChatGPT pulls information from all over the internet and compiles it together in a summarized format which may not be 100% correct depending on the sources ChatGPT used. Thus, by using a search engine, the information received from ChatGPT prompts can then be checked by finding research papers and articles from trusted sources.

Case Study 1: Question, "Provide information on the top three secondary battery technologies"

"Here's an overview of the top three secondary (rechargeable) battery technologies currently in use:

1. Lithium-Ion (Li-ion)

- Chemistry: Utilizes lithium salts in organic solvents as electrolytes.
- Applications: Predominantly found in consumer electronics (smartphones, laptops), electric vehicles (EVs), and renewable energy storage systems.
- Advantages:
    - High Energy Density: Offers one of the highest energy densities (150-250 Wh/kg), making it ideal for portable applications.
    - Low Self-Discharge: Retains charge well when not in use, typically under 5% per month.
    - Long Cycle Life: Generally lasts 500-1500 charge cycles with proper care.
- Disadvantages:
    - Safety Risks: Susceptible to thermal runaway if damaged or improperly charged.
    - Cost: More expensive to manufacture compared to some alternatives.
    - Environmental Concerns: Lithium extraction can have environmental impacts.

2. Nickel-Metal Hydride (NiMH)

- Chemistry: Composed of nickel oxide hydroxide and a hydrogen-absorbing alloy.
- Applications: Commonly used in hybrid vehicles, rechargeable batteries for household devices (AA, AAA), and some consumer electronics.
- Advantages:
    - Moderate Energy Density: Offers energy densities between 60-120 Wh/kg, which is adequate for many applications.
    - Environmental Benefits: Less toxic and more environmentally friendly than some lithium-based batteries.
    - Stable Performance: Performs well at various temperatures and has good cycle stability.
- Disadvantages:
    - Higher Self-Discharge Rate: Can lose charge more quickly than Li-ion batteries.
    - Lower Energy Density: Not as efficient as Li-ion in terms of energy storage.
    - Limited Cycle Life: Typically lasts 300-500 cycles.

3. Lead-Acid

- Chemistry: Consists of lead dioxide, sponge lead, and sulfuric acid.
- Applications: Widely used in automotive batteries, uninterruptible power supplies (UPS), and for grid energy storage.
- Advantages:
    - Cost-Effective: One of the most affordable battery technologies available.

    - Established Technology: Well-understood and widely recycled, with a long history of use.
    - Reliable Performance: Provides stable voltage and discharge characteristics.
- Disadvantages:
    - Heavy and Bulky: Has a low energy density (30-50 Wh/kg), making it less suitable for portable applications.
    - Short Cycle Life: Typically lasts 200-300 cycles, especially if regularly deep-discharged.
    - Environmental Concerns: Lead is toxic, although recycling programs exist.

These three battery technologies are dominant in their respective markets, each with specific advantages and limitations that influence their applications and overall effectiveness." (ChatGPT conversation, 2024)

As I began research into the battery supply for our project, I started by asking ChatGPT a variety of questions like the one above. This method of prompting ChatGPT to provide information on the different rechargeable battery technologies, listing the advantages and disadvantages, making comparisons between different technologies, and more, was very helpful in gaining a base understanding of each battery technology and determining the criteria to use for making technology comparisons. I specifically mention that this method was useful for gaining a base or foundation of understanding, because from that I branched out into using Microsoft edge to find more in-depth material on the chemistry of these batteries, the specific characteristics of each technology, and resources for the charging requirements for each technology, which enabled me to expand my understanding of the various battery technologies.

From my experience using ChatGPT, it was most helpful in getting a baseline of information for the technology I was researching, easily explaining how the technology works, and providing solid pros and cons analysis. As mentioned already, I could not use it for the entirety of my research since I needed to access more specific and in-depth information on the technology, which ChatGPT was not able to provide me with. Thus, by using ChatGPT it provided an excellent base of understanding for research. So by using the information I received, it helped me to narrow down the scope of what I needed to specifically search for with the use of Microsoft Edge.

Case Study 2: Question, "Here is a PDF of our current project report. Give me the following data for MCU's ESP32, ESP8266, Raspberry Pi Pico, and arduino nano:

number of cores, hz, connectivity(wifi, bluetooth), memory, then give key features of each.

###BREAK###

Next, give a general suggestion of which wifi capable MCU would be best fit for our project."

“MCU Specifications Comparison

MCU	# of Cores	Hz	Connectivity	Memory	Key Features
ESP32	2	80 - 240 MHz	Wi-Fi, Bluetooth, BLE	520 KB SRAM	Dual-core processor, versatile connectivity, low power modes, and suitable for real-time video streaming.
ESP8266	1	80 - 160 MHz	Wi-Fi	160 KB SRAM	Low-cost, single-core, good for basic Wi-Fi applications, but lacks Bluetooth and has less memory.
Raspberry Pi Pico	2	133 MHz	None (requires external module)	264 KB SRAM	Affordable, great GPIO options, lacks built-in connectivity, more for GPIO-heavy projects.
Arduino Nano	1	16 MHz	None (requires external module)	2 KB SRAM	Very limited memory and processing power, best for simple projects, lacks built-in wireless connectivity.

Key Features

- **ESP32**:
    - Dual-core processor allows for high-performance multitasking.
    - Integrated Wi-Fi and Bluetooth, making it flexible for various wireless applications.
    - Large memory size (520 KB SRAM), ideal for handling tasks such as video streaming and real-time controls.
    - Low power consumption with power-saving modes.

- **ESP8266**:
  - Single-core, Wi-Fi-only microcontroller, good for cost-sensitive projects.
  - Suitable for projects where basic internet connectivity is needed, but less memory and power compared to ESP32.
  - Ideal for simple IoT applications or basic internet-controlled devices.
- **Raspberry Pi Pico**:
  - Dual-core ARM Cortex-M0+ processor.
  - No built-in Wi-Fi or Bluetooth (requires external module).
  - Great for projects that need extensive GPIO usage, but less suitable for internet-connected applications without additional components.
- **Arduino Nano**:
  - Single-core, very basic processing power and memory (2 KB SRAM).
  - No built-in Wi-Fi or Bluetooth, requiring external modules for wireless applications.
  - Best for small, simple embedded applications, unsuitable for complex, internet-connected projects.

General Suggestion

For your **internet-controlled car racing project**, the **ESP32** would be the best fit. It provides both **Wi-Fi** and **Bluetooth** connectivity, and its **dual-core** processor ensures that real-time control and video streaming can be handled smoothly. The ample **memory (520 KB SRAM)** makes it suitable for handling more demanding tasks, such as live streaming, controlling motors, and processing commands with minimal latency.

The **ESP8266** is a solid alternative if you're working with a tighter budget or don't require Bluetooth functionality, but its limitations in processing power and memory make it less ideal for more complex, multi-tasking operations."

As I worked on researching the microcontroller (MCU) options for our project, I found ChatGPT to be a useful tool in getting a quick overview of key specifications for different MCUs like the ESP32, ESP8266, Raspberry Pi Pico, and Arduino Nano. I used prompts like, 'What are the specifications of the ESP32?' and 'How does the ESP32 compare to the ESP8266?' to quickly pull together basic information. ChatGPT provided a breakdown of details like the number of cores, clock speed, connectivity options, and memory for each MCU, which allowed me to get a general idea of how they stacked up against each other.

This method of using ChatGPT to gather an initial set of data proved to be an effective way to streamline the beginning of my research. While I quickly gathered general specifications, I soon realized that some of the technical details were either outdated or needed to be verified against more trusted sources. Therefore, I cross-checked the information I received from ChatGPT with datasheets and official documentation for accuracy. Despite this limitation, using ChatGPT in the early stages sped up the process significantly, as it gave me a clear starting point for deeper research and allowed me to narrow down which MCUs to focus on.

In addition to the technical specifications, ChatGPT also provided suggestions on which MCU might be best suited for my project based on the needs I described, such as real-time video streaming and low-latency control. This suggestion didn't replace in-depth research but provided me with a good sense of direction on what to explore further.

Overall, ChatGPT served as a helpful tool for establishing a foundation of understanding and offered a quick comparison of options. It wasn't suitable for providing detailed, specialized insights—such as specific application notes or real-world testing scenarios—but by using it to build a general framework, I was able to refine my search and dive deeper into trusted sources like official datasheets and technical forums. This combination of initial insights from ChatGPT and detailed follow-up research helped me approach the project with both speed and accuracy.

Case Study 3: Question, "Provide a detailed description and compare C, Java, and Python languages"

C Programming Language

Overview:
C is a compiled, procedural programming language developed by Dennis Ritchie at Bell Labs in 1972. Initially created for system programming, especially for building operating systems, C remains a general-purpose language. It is low-level and close to machine code, with syntax requiring manual memory management. C's performance is highly efficient, making it ideal for performance-critical applications. It requires compilation to convert source code into machine-specific code, which means that C programs must be recompiled for each platform. Developers manage memory manually using functions like `malloc` and `free`.

Strengths:
C's primary strength is its speed and efficiency due to its proximity to the hardware. This makes it an excellent choice for systems programming, embedded systems, and any application where performance is key. Additionally, C is highly portable at the source code level; although platform-specific adjustments may be needed, it can be compiled for various architectures. C also offers precise control over system resources through pointers and direct memory manipulation, giving developers a level of control that higher-level languages don't offer.

Weaknesses:
The complexity of C's low-level operations, such as manual memory management and pointer arithmetic, can make the language difficult to master and error-prone. Additionally, C lacks modern safety features like automatic memory management and bounds checking, which can lead to vulnerabilities like buffer overflows.

Java Programming Language

Overview:
Java is a compiled, object-oriented, platform-independent language created by Sun Microsystems in 1995. It was designed to minimize implementation dependencies, allowing developers to "write once, run anywhere" using the Java Virtual Machine (JVM). Java's syntax is more abstract and higher-level compared to C, with a focus on object-oriented programming principles such as inheritance and encapsulation. While Java's performance is generally slower than C due to the overhead of the JVM, modern optimizations like Just-In-Time (JIT) compilation help improve execution speed. Java includes automatic memory management through garbage collection.

Strengths:
Java's key advantage is its platform independence, thanks to the JVM. Once code is compiled into bytecode, it can run on any platform that supports a JVM. Java also promotes robust software development with its object-oriented design and built-in error handling mechanisms, making it suitable for large-scale, enterprise-level applications. Automatic garbage collection reduces memory-related errors like leaks, contributing to Java's overall robustness.

Weaknesses:
The abstraction layer introduced by the JVM and garbage collection introduces performance overhead, making Java slower than C. Additionally, Java's syntax is often considered verbose, requiring more lines of code to achieve tasks that might be simpler in languages like Python. Java applications also tend to use more memory due to the JVM, especially for larger-scale applications.

Python Programming Language

Overview:
Python is an interpreted, high-level, dynamically-typed language created by Guido van Rossum in 1991. Its main focus is on readability and developer productivity, making it a general-purpose language used in various domains, including web development, data science, and automation. Python's syntax is simple and easy to understand, emphasizing code clarity. It is slower than both C and Java because it is interpreted and dynamically typed. Python code is interpreted at runtime, though it can be compiled to bytecode for minor performance improvements. Memory management is automatic via garbage collection.

Strengths:
Python is known for its ease of use and is highly favored for its simple, readable syntax, making it an ideal language for beginners and rapid development. It has a versatile ecosystem of libraries and frameworks that support a wide range of applications, from web development to machine learning. Python also benefits from a massive and active community, which contributes to its extensive resources and continuous growth.

Weaknesses:
Python's interpreted nature and dynamic typing make it significantly slower than statically-typed, compiled languages like C and Java. Additionally, Python's Global Interpreter Lock (GIL) in the default CPython implementation can limit the performance of multi-threaded programs. Python's dynamic typing can also lead to runtime errors that might have been caught at compile-time in other languages.

Comparison Summary

- **C**: Offers maximum performance and control, especially for systems programming and embedded systems. However, it requires detailed memory management and can be error-prone due to its low-level nature.
- **Java**: Ideal for platform-independent, large-scale applications and enterprise environments. It strikes a balance between performance, ease of use, and portability, though it incurs some performance overhead due to the JVM.
- **Python**: Best for rapid development and applications requiring high-level abstractions, such as data analysis or web development. Its simplicity and vast ecosystem make it popular, despite its performance limitations.

Each language has distinct strengths, and the choice depends on project-specific needs such as performance, development speed, and portability.

End of ChatGPT Response

When getting started with the research on the software side of the project, I started out by asking for a detailed comparison between the major programming languages that we could choose from for programming the project. This response provided me with a great amount of detail to use when writing the Development Environment section for our research in Chapter 3. When first deciding on the language I would use for this project, I wanted to write in Python as I have never written in Python before and it would be a great opportunity to learn a new language. However, after reading through this comparison, and fact-checking a lot of things just to be sure it was correct, it seemed that C would overall be better for our project. This conclusion was then verified by ChatGPT by asking it, "If I was programming an MCU for a remote controlled car, which language should I use?" ChatGPT then listed 5 different reasons why C would be the language of choice for our project, and is now the language we are going to actually write in.

Continuing with writing the Development Environment section of our research, there were a few things that I wanted more information about, as ChatGPT did not go into as much detail as I wanted for a few things. I asked it a few times to expand on a certain topic, such as asking it to "expand on the portability of C", "briefly expand on Java's platform independence", "expand on python's ease of use", and several more. This provided me with a few more details that were not covered in the first prompt, and allowed me to have a better understanding of each language's strengths and weaknesses, and also to be able to provide a better explanation myself when writing our research section.

Overall, this method of research by using ChatGPT allowed me to get most of my information all in one place that is also concise and to the point. This allowed me to have a better understanding of a certain topic before I went and researched that topic myself on Google in order to fact-check and look into real world examples of the concepts in action. ChatGPT was also a major influence in deciding which technologies, programming languages, and online code repositories would work best for our project.

Case Study 4: Question: “What are some general specs of high and ultra-high frequency passive RFIDs?”

“Passive RFID tags in the high-frequency (HF) and ultra-high-frequency (UHF) ranges differ in terms of performance, operating principles, and general specifications. Here's a breakdown of their common specifications:

High-Frequency (HF) RFID:

- Frequency Range: Typically 13.56 MHz
- Read Range: 1 to 10 cm (up to 1 meter in specialized cases)
- Data Transfer Rate: Moderate
- Antenna Type: Typically uses inductive coupling (coils)
- Typical Applications: Access control, ticketing, smart cards, and near-field data exchange
- Cost: Generally lower than UHF tags
- Pros:
    - Better performance in environments with metal or liquid.
    - Easier to shield or restrict.
- Cons:
    - Shorter read range compared to UHF.
    - Slower data rates.

Ultra-High-Frequency (UHF) RFID:

- Frequency Range: 860 to 960 MHz (depending on region)
- Read Range: Up to 10 meters (some can reach 20-30 meters with ideal conditions)
- Data Transfer Rate: Faster than HF
- Antenna Type: Uses radiative coupling (dipole antennas)
- Typical Applications: Inventory tracking, supply chain management, asset tracking, and toll collection
- Cost: Slightly higher than HF tags
- Pros:
    - Longer read range.
    - Faster data transmission.
    - Suitable for high-speed scanning and bulk reading.

- Cons:
    - More affected by interference from metal or liquids (though modern designs mitigate this).
    - Higher power requirements for readers.

Both types of RFID tags do not have an onboard power source (passive), relying on energy from the RFID reader's signal to power the tag.

These tags are chosen based on the required range, environment, and cost considerations for the application." (ChatGPT conversation, 2024)

Throughout my research process, usage of ChatGPT has been very handy in getting a general idea of various technologies and products. In this case, ChatGPT did well in giving a concise summarization of HF and UHF passive RFIDs and their specifications. As shown, there is a variety of sub-topics revolving around these technologies that are considerable in research. This included frequency ranges, data transfer rates, read ranges, antennas, costs, typical applications, etc. From this, I was able to look deeper into these subjects and find any specific information. I filled these gaps of lack of information with more descriptive and quantitative details using Google Chrome and finding various websites that go in more detail for these technologies. Asking ChatGPT for these specific details is also an option, however, I found that values given by ChatGPT contradicted those found in other sources. Realization of this problem arose after prompting ChatGPT for estimated data transfer rates for each RFID type at two different occasions and received different values each occasion. This shows how ChatGPT should not be solely depended on and is prone to making mistakes when it comes to particular information.

From this experience, we learn that ChatGPT is a great tool to get general information that can help us to break down the topics we do want to cover and share to our readers. In order to effectively use ChatGPT we must be selective and use other sources to verify that the information it has given us is consistent across multiple platforms. This method has worked very well, as it has given structure to research in our initial approach, and encourages extensive research across different sources in order to find a common truth in the information.

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 racing 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	$200
Finish Line and Racetrack	$100
Server	$50

10.2 Bill of Materials

Table 10.2. *Initial Bill of Materials*

Component	Part Description	Quantity	Price Estimates
Cars			
RC Car	RC Toy-Grade Car	2	$10
Microcontroller	ESP32	2	$10
Motor Driver	DRV8833 Dual Motor Driver	2	$10
CMOS Image Sensor	OV2640 Image Sensor	2	$3
Battery Supply	7.4 V Li-Po Battery, 1100 mAh Capacity	2	$20
Antenna	Trace Antenna	2	$2
PCB Estimate		2	$60
Finish Line & Racetrack			
Finish Line Enclosure	3D Printed Enclosure	1	$12
Micocontroller	ESP32	1	$5
Sensor/Camera	Pixy 2 Camera	1	$60
LED Matrix	Adafruit Bi-Color LED	1	$15

	Square Pixel Matrix		
Battery Supply	7.4 V Li-Po Battery, 1100 mAh Capacity	1	$10
Antenna	Trace Antenna	1	$1
PCB Estimate		1	$30
Racetrack Materials	Materials to Build the Racetrack		$30
Combined Total Cost			$278

10.3 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	Cars Design & Construction
	Project Lead
	Software Design & Implementation
	MCU & Motors Implementation
	Communications & Antenna
Jesvany Colon	Finish Line PCB Design
	Finish Line Sensor Selection & Implementation
	Finish Line Design & Construction

	Administrative Management
Austin Silva	Cars PCB Design
	Team Lead & Project Management
	Power Supply Selection & Implementation
	Support Cars Design & Construction
	Support Finish Line Design & Construction

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.

10.4 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 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, and start designing system
10	10/21/24-10/25/24	Submit 60-Page Report	Submitting 60-Page progress report on time, and making parts selections
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 ordering parts
12-13	11/04/24 - 11/15/24	120-Page Report Draft and Mini Demo Video	Finalizing system design and conducting system testing, creating the mini demo video and completing the 120-Page final report draft
14-15	11/18/24 - 11/26/24	Final Report Submission	Reviewing the final report and mini demo video, submitting both on time, updating team website as well

Table 10.5. *Senior Design 2 Timeline*

Senior Design 2			
Week(s)	**Dates**	**Task**	**Description**
1-3	01/06/25 - 01/24/25	PCB Design	Work on PCB design and research prototype assembly methods
4-6	01/27/25 - 02/14/25	Order PCB	Order PCB once design is complete and begin assembling the prototype
7-8	02/17/25 - 02/28/25	System Testing and Integration	Assembling system prototype and integrating all components, testing

			prototype's functionalities
9-10	03/03/25 - 03/14/25	Optimize Prototype	Completing and optimizing 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.