AR Kart XRP

RC Car with VR Game Integration

Final Documentation

Fall 2025

Table of Contents

AR Kart XRP Overview

Background

The AR Kart uses a Radio Control (RC) car developed to work within an Augmented Reality (AR) game environment, made in partnership with RIT. The style of the game is intended to be in the format of Mario Kart Live: Home Circuit. The user will be able to create their own track with different elements, such as turns, indicated by designated AprilTags. These AprilTags will be read by the camera onboard the kart and translated into the video game developed by the student team at RIT.

Purpose

The purpose of the AR Car is to assist in creating a racing game, with educational applications, for the XRP. We expect the AR kart XRP to ultimately be used in educational settings such as classrooms to facilitate learning. Students will be able to design their own tracks and learn about AprilTags and how they work in relation to the video game.

Semester Objectives

The primary goal for Fall 2025 was to prepare a working prototype of AR Kart that works with the game developed by RIT.

The mechanical team worked on developing the RC car with Ackermann steering and mounting on all the required components.

The electrical team focused on creating a Kalman filter and movement control of the AR Kart by testing a standard XRP robot.

The computer science (CS) team worked on implementing camera streaming.

Mechanical

Kart Body and Steering

Introduction

The AR Kart (RC Car) is a joint effort between CUP Robotics and RIT to develop both a physical RC Car and a video game that interacts with the physical car. The inspiration behind this project is the Mario Kart Live: Home Circuit developed by Nintendo, where players are able to control a physical Mario Kart and drive around a user customized track around their home. Our goal for this project is to create something similar but with the XRP platform, as well as keeping assembly easy and fully 3D printable.

Design Process

The first design of the RC Car is a very simple one, adapted from the XRP board mounts and motor mounts and simply rearranged, this created a very quick prototype of the car. On the chassis, there are locations and pins for an outer shell to slide on top, since the initial design philosophy was to make it look as much like a production car as possible. However, this leads the car body to be very long, which in a classroom setting could be detrimental to the usage of the car.

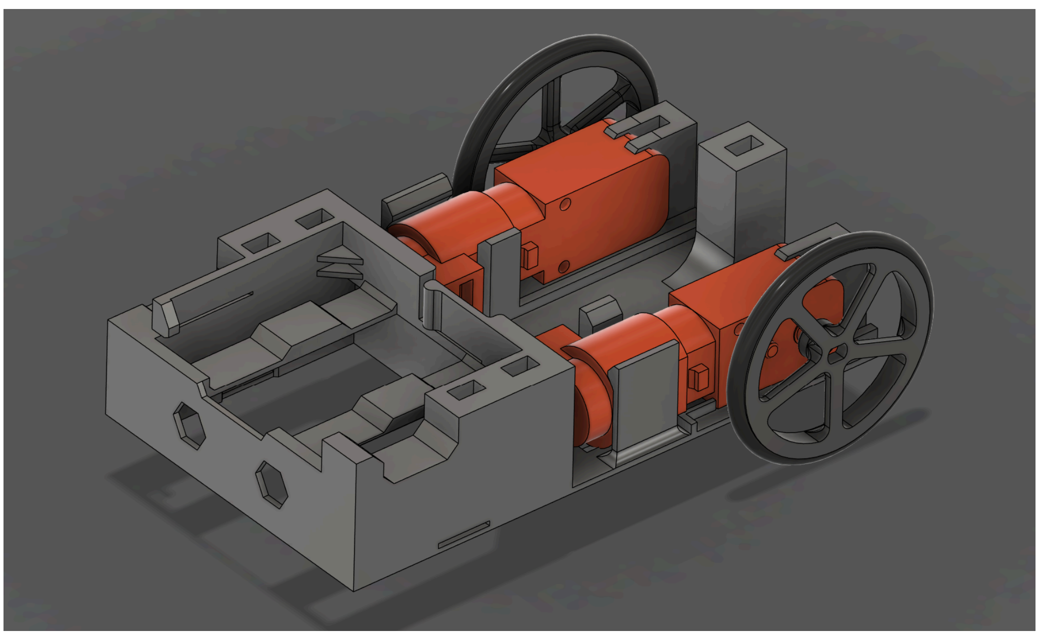

Figure 1: First Iteration of the RC Car

The next iterations of the body, we aimed to shorten the assembly as a whole, as well as have the XRP board be placed in a vertical orientation in order to facilitate the screen that will be attached to the RC Car in the final version. By flipping the mount vertically, we not only save space, we also gain the screen to show the emotion system. The downside of this, is that although the overall shape of the car is shorter, the taller build made the aesthetics more difficult to get down since it no longer looks like a real production car.

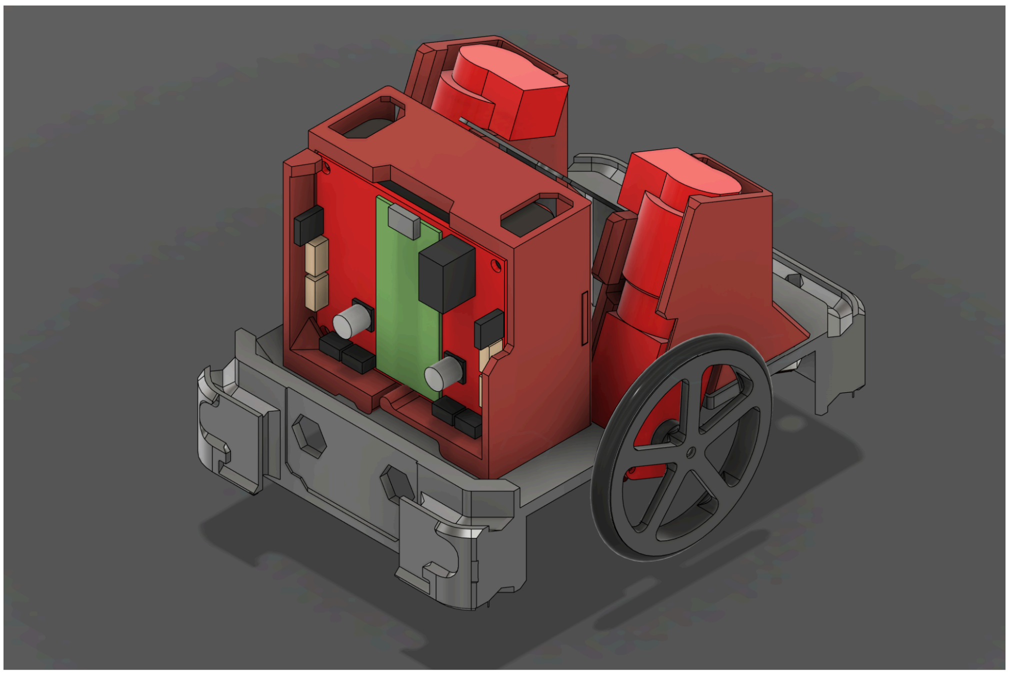

Figure 2: Second prototype of the RC Car body

This iteration of the RC Car is simply a modified version of the existing XRP. By cutting down the sides, we reduce the overall weight and footprint of the car, but there are also the XRP rails kept on the front and back, enabling the same XRP expandability that was always offered. This version of the body also has a proper Pi mount at the end, which facilitates the camera communications.

The next major part of the AR Kart is the steering system. Initially, we discussed the possibility of a differential drive, where it will simply be the two driving motors that turn. By powering the motors in opposite directions, we are able to achieve a turn similar to the XRP. However, this made the whole project too reminiscent of just a normal XRP, maybe with a camera attachment. Thus, in order to differentiate this project, we decided to implement an Ackermann steering system instead. This also aligns with our inspiration for this project, since most go karts and similar previous mechanisms all have Ackermann steering, which helps the car turn smoothly due to the inner wheel turning at a sharper angle than the outside wheel.

The first iteration of the steering system aimed to use a print in place configuration. Since the XRP is focused on easy assembly as well as ease of printing, a print in place system seemed obvious.

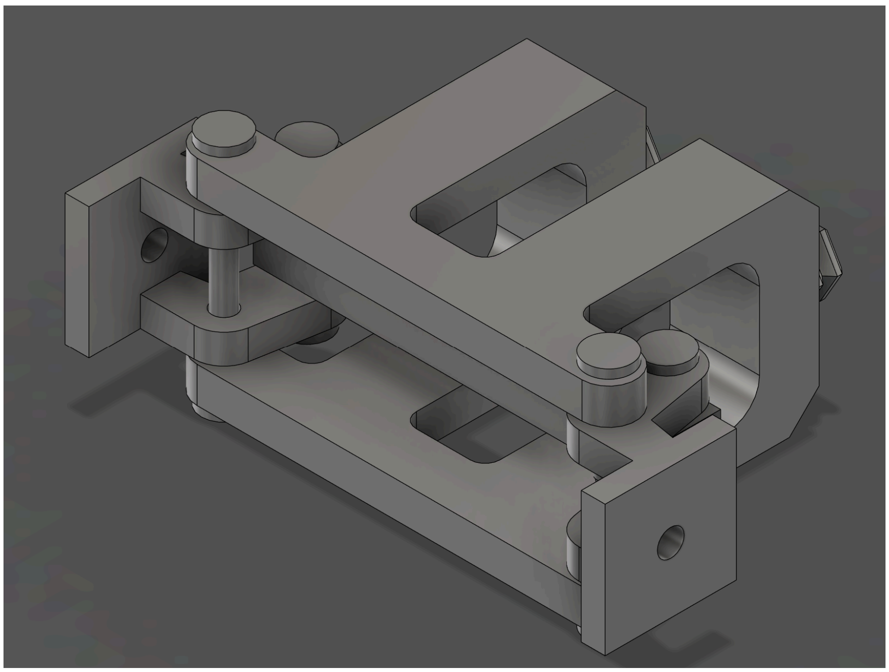

Figure 3: First iteration of the steering mechanism.

However, as the first prototype was printed, several issues became obvious. Since each rod was so thin, the overall steering mechanism was very fragile. With mishandling causing the crucial steering rods to break, it required the user to reprint the entire assembly. Another issue was that the print in place idea caused a lot of excess filament and supports to be on the crucial pivots, thus the turn was not smooth due to excess friction.

Later revisions changed the four axles to print separately and then assembled. The new axles are a lot thicker thus more durable. This new thickness allowed the axles to be printed with a hex bolt attachment on top. Although there is now some assembly required, the hex bolts still make the assembly very easy. This new assembly basically resolved all the issues we had with the first prototype. The assembly is now in multiple pieces, but still simple. The axles are strong and easily replaceable, and the entire assembly is very low friction throughout, allowing the servo to turn the wheels better.

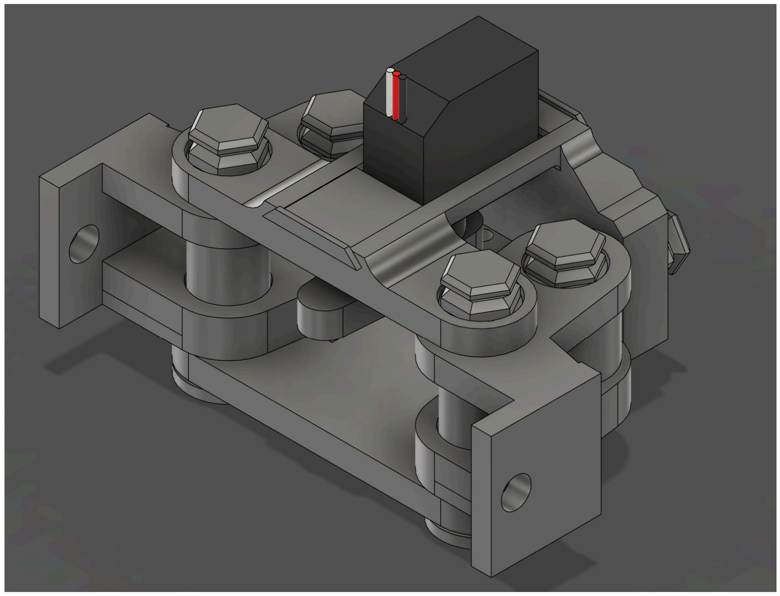

Figure 4: Updated steering module

Now with the steering module itself finished, one major challenge remained with how to attach the servo to the steering module. This challenge exists due to the fact that the servo horns must be attached to another object using a screw, but the goal of this project is to not use any screws. To resolve this issue, we implemented a servo horn housing, which slides over the actual servo horn. Then, the servo horn housing will be attached to the actual steering module via a rod that is push fit.

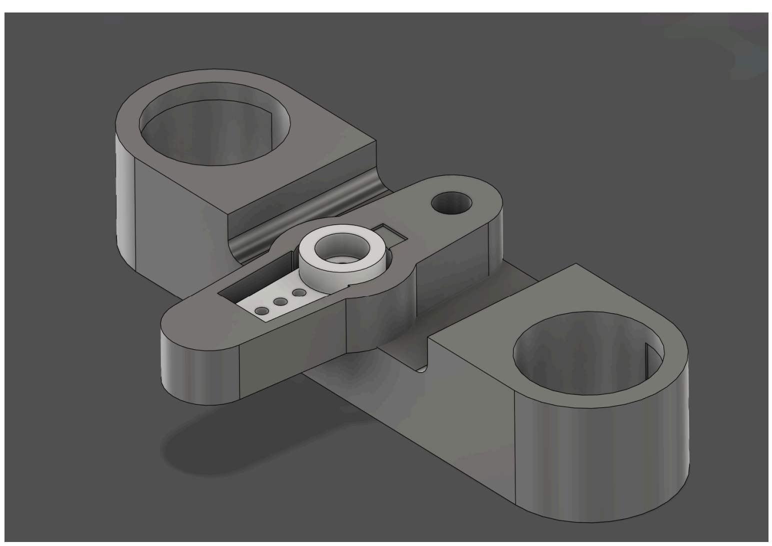

Figure 5: Servo horn attachment

Manufacturing process:

The whole assembly will be 3D printed, and assembled using hex clips.

Assembly:

Assembling the AR Kart can be done very easily. After printing all the parts and gathering the proper XRP components, simply align all the hex bolts to its corresponding location and attach the hex clip. For the steering module, the shorter axles go in the back and the longer axles go in the front.

Next Steps:

For now, all the mechanically interesting features have been implemented, what is left is only testing, awaiting the game from RIT, and external aesthetic pleasing pieces.

Bumper

Purpose

The bumper provides protection while also improving the car's function and appearance, and it is designed to support secure attachment of additional features. Inspired by the XRP system, the bumper creates a strong foundation for modular add-on components, allowing elements such as wheel covers and decorative pieces to integrate cleanly with the chassis. Extending the bumper along the sides reinforces structural stability and forms a continuous mounting pathway that supports customization. Through this combination of practical engineering and adaptable design, the bumper moves beyond simple protection and becomes a platform for future modification and personalization.

Design Process

The first idea for the bumper system was to combine the bumper and wheel covers into a single part that attached directly to the front of the XRP board. While this approach looked clean in CAD, it created several practical issues once prototyped. Because everything was connected, the piece was difficult to put on and take off, and any small adjustment required reprinting the entire assembly. More importantly, the design did not allow for customization, since changing the bumper would automatically affect the wheel covers.

To solve these problems, the design was split into separate components. The bumper became its own independent piece, while the wheel covers were redesigned to attach using a Lego-based system rather than mounting directly to the bumper sides. This approach allowed the wheel covers to connect to the central mounting piece, making them easier to customize, reposition, or replace without affecting the bumper.

The bumper itself was refined to better interface with the chassis. A hex bolt now runs through a hole in the front wheel mount and secures the bumper in place using hex clips, creating a stable yet removable connection.

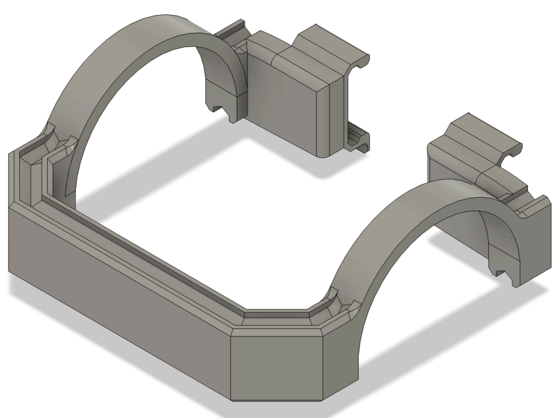

Figure 6: Bumper and wheel covers together attaching to the board

Figure 7: Bumper attaching to the front wheel mount

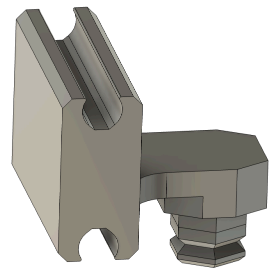

Figure 8: Smaller bumper attaching to the front wheel mount

A shorter middle bumper was also added to fit between the wheels without limiting their movement. This piece makes the front half of the car lighter and slightly shorter, which may help the car accelerate or handle better, though this has not been tested yet.

Implementation

Manufacturing Process

The components were designed in Fusion 360 and manufactured using rapid prototyping methods on 3D printers.

Assembly

To assemble the bumper, align the part with the front wheel mount. The hex bolt is placed through the hole in the wheel mount facing down, and hex clips are added to hold the bolt in place. The shorter middle bumper is assembled the same way and positioned between the wheels to ensure it does not block their movement.

Next Steps

The bumper system is considered complete and meets the design goals for this project cycle. It attaches securely to the chassis and supports modular additions. Although the current design performs well, there are still opportunities for future improvement.

One area for further work is testing the shorter middle connector piece. The reduced weight and length may improve speed and handling, but this has not yet been evaluated. Collecting performance data would help confirm these potential benefits.

Wheel Covers

Purpose

The wheel covers were created to enhance both the visual aspect and functional adaptability of the RC car. Their curved form adds a cohesive aesthetic to the front assembly while providing a stable platform for customization through Lego-based attachments. By designing the covers as modular components, they support future decoration, styling, and feature additions without requiring modifications to the chassis or bumper. The final design balances appearance, usability, and flexibility, allowing the wheel area to become a more expressive and configurable part of the system.

Design Process

The wheel covers went through several design iterations. The first version consisted of three separate pieces that attached to the bumper using the rail mounts. While this worked, the team wanted the covers to integrate with Lego elements to allow easier customization, so the two side pieces were redesigned to connect through the Lego-based system instead. During testing, the covers were extremely secure, but having both the rail mounts and the Lego attachment made them very difficult to remove once installed. To simplify the design and improve usability, the rail mounts were removed entirely, leaving the Lego connection as the sole attachment method. This approach kept the wheel covers stable while making installation and removal much easier.

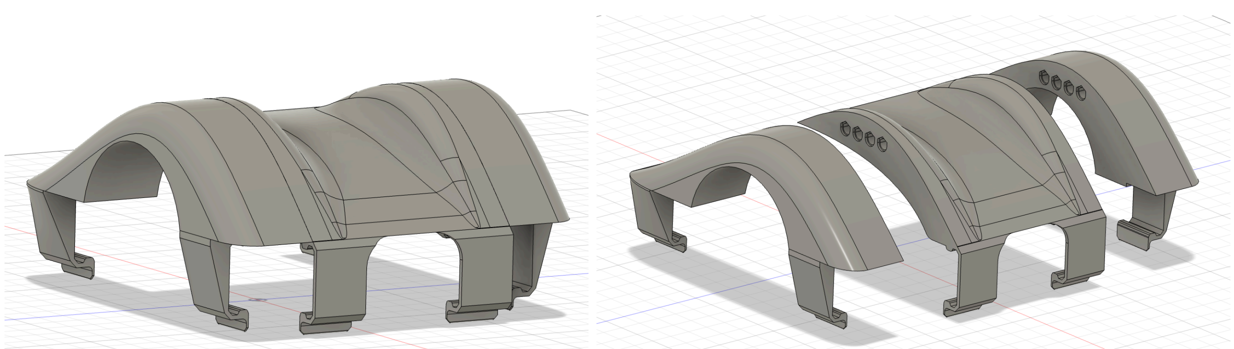

Figure 9: Wheel covers including rail mounts and LEGO integration

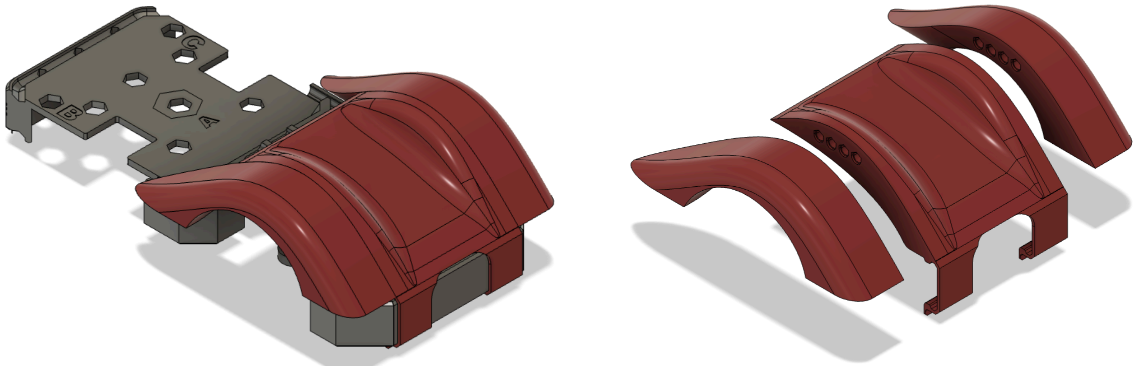

Figure 10: Wheel covers including only LEGO integration on side pieces

Implementation

Manufacturing Process

The components were designed in Fusion 360 and manufactured using rapid prototyping methods on 3D printers. They were printed on their sides to ensure the Lego holes formed correctly and maintained proper dimensional accuracy.

Assembly

Assembly of the wheel covers begins by attaching the main center piece to the middle part of the bumper. The two side pieces are then connected using the Lego-based connectors, which hold them firmly in place while still allowing easy removal. With the rail mounts removed in the final design, the Lego connection becomes the only attachment method, providing a stable yet accessible system for adding or removing the wheel covers as needed.

Next Steps

The wheel covers are complete and meet all of the goals set for this project cycle. They attach securely using the Lego-based system, support modular additions, and integrate smoothly with the rest of the vehicle. The design is stable, easy to assemble, and does not require further functional changes at this time.

If future work were to continue, the main opportunity would be to extend the aesthetic of the front half of the car to the rear. The updates made this semester focused primarily on the front assembly, and carrying the same visual style toward the back would help create a more unified and polished look. This would make the car feel more cohesive overall while preserving the modular, customizable features introduced through the bumper and wheel-cover designs.

One-Piece Shell

Purpose

The initial body shell was designed to provide the RC car with a clean, unified exterior while protecting the chassis. The design aimed to cover the car's top and sides in a single piece, creating a smooth surface that integrated the bumper, wheel covers, and front housing into a single continuous surface. The shell also included custom mounting points to secure attachment to the chassis. Its primary purpose was to provide structure, protection, and an improved appearance in a single integrated component.

Design Process

The first concept focused on creating a full-body shell that wrapped around the chassis and wheel area. The wheel openings were shaped directly into the shell, and internal supports were added to align it with the board. Taunting blocks were designed on the inside so the shell could attach firmly without needing additional hardware using a custom pin system. The top of the shell also included holes for a spoiler that used this same pin system. This approach produced a simple, clean exterior and eliminated the need for separate bumper or wheel-cover parts.

However, as the project progressed, the design requirements changed. The team decided to move toward a more modular system that allowed parts to be added, removed, and customized easily. Because the single-piece shell limited flexibility and made attachment and redesign more difficult, the concept was phased out in favor of independent bumper and wheel cover components.

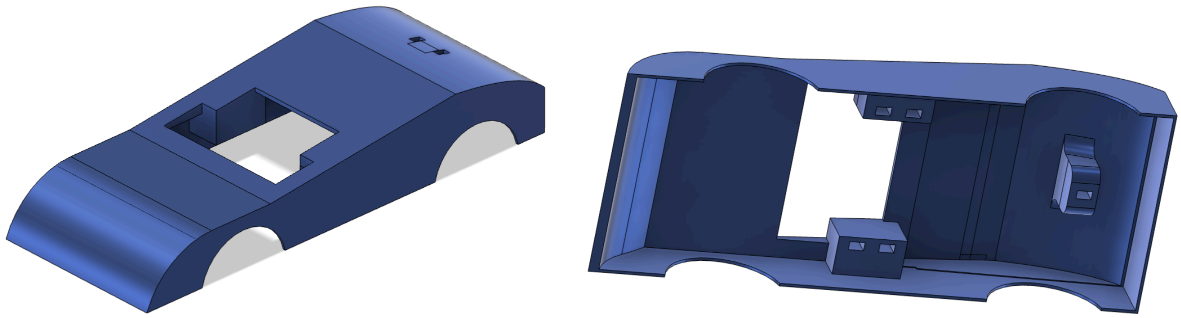

Figure 11: Bumper and wheel covers together attaching to the board

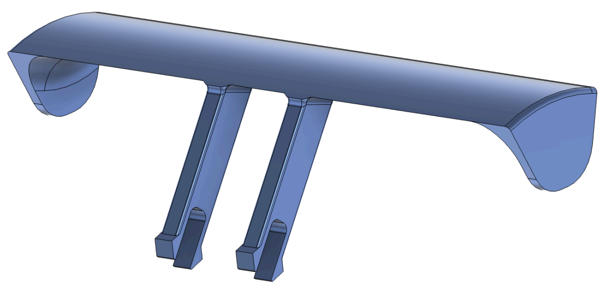

Figure 12: Spoiler

Next Steps

No further development is planned for this design, as the project has moved toward a modular system.

Camera Mount

Purpose

The camera mount was designed to provide a stable, forward-facing position for the camera while integrating cleanly with the RC car's existing layout. By attaching directly to the Blue Origin Lego core that holds the battery, the mount places the camera high enough to see over the wheel covers and capture a clear view of the path ahead. The design also supports quick removal, so the board can slide in and out of the chassis without obstruction. Through its flexible attachment system and secure camera housing, the mount provides a reliable, convenient way to integrate front-facing vision into the vehicle.

Design Process

The camera mount was designed to attach to the Blue Origin Lego core that holds the battery. Placing it on the top allowed the camera to face forward and look over the wheel covers, giving a clear view of the front of the car. Because the board needs space to slide in and out of the chassis, the mount had to be easy to remove without tools. To achieve this, the mount's ends were lengthened and thinned, with small cylinders added to each side. These cylinders compress when pushed in and expand into the holes on the Lego core, creating a secure but removable connection.

Finding the correct dimensions for these flexible attachment points took several iterations. Early versions were either too stiff to bend or too thin and broke during assembly. Adjusting the arm thickness and length helped balance strength and flexibility. The front piece that holds the camera in place was adapted from a previous project and integrated into the new design. The final result is a mount that is stable, easy to remove, and positioned for an effective forward-facing view.

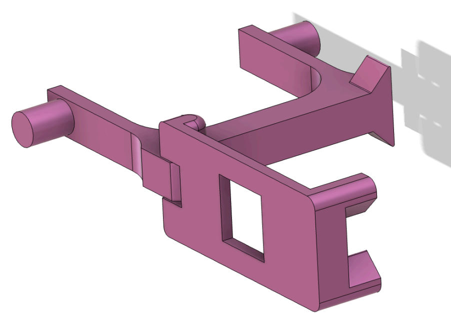

Figure 13: Camera Mount

Implementation

Manufacturing Process

The camera mount was modeled in Fusion 360 and produced using 3D printing. The part was printed on its side, shown in Figure 7, to ensure that the flexible attachment arms and cylindrical connectors formed accurately and sturdily. Several prototypes were built to refine the arm thickness and length so they could bend without breaking while still snapping securely into the holes of the Blue Origin Lego core.

Assembly

To assemble the camera mount to the car, press the two flexible arms inward and insert them into the matching holes on the top of the Blue Origin Lego core. Once released, the cylindrical connectors expand into place, securing the mount firmly. The front camera holder, adapted from a previous project, slides into the main frame and secures the camera. The mount can be removed by compressing the arms again.

Next Steps

The camera mount is not yet complete and requires further refinement. Although the current design securely attaches to the Blue Origin Lego core, the camera can still rotate up and down, preventing it from maintaining a fixed viewing angle. Future work should focus on adding a mechanism that locks the camera in place while still allowing for easy removal when the board needs to be accessed. Completing this feature will ensure that the camera remains stable during operation and provides consistent forward-facing visibility.

RC Gate

Purpose

The RC gates are used to construct the physical race track for the RC car game, with the RC car traveling underneath each gate as it moves through the course. They connect the physical track to the online version of the game. Each RC gate has an AprilTag attached, which the RC car reads using a camera mounted on the vehicle. By detecting these AprilTags, the RC car can identify what type of track piece lies ahead, such as a straightaway, left turn, or right turn. This information allows the car to map the physical race course into the digital version of the game in real time.

Using identical RC gates with interchangeable AprilTags also introduces modularity. Because any AprilTag can be attached to any gate, users are able to design their own unique race tracks by swapping tags. This modularity makes the game more engaging, as players can easily customize the layout of the course. In total, I designed eight different AprilTags corresponding to eight different race track pieces. A guide explaining what each AprilTag represents and how it is implemented in the game is provided in the following Google Doc:

https://docs.google.com/document/d/1Y2j1gwjwbXMPZ1A3llwvdSgl2kJZ_85u8ChK0dziiTY/edit?tab=t.0

Beyond modularity, the RC gate itself needed to meet several functional requirements. It had to be sturdy enough to withstand impact if a car hit it, large enough for multiple RC cars to pass through, and capable of holding an AprilTag securely. At the same time, the AprilTag holder needed to allow tags to be easily attached and removed to allow modularity. Additionally, the holder had to rotate so that different angles could be tested to determine which orientation worked best for the RC car's camera.

Design Process

The project began with the need for a race track at MIT. For this initial setup, the track was created using cardboard cutouts that indicated the direction the RC car should drive. While this solution worked as a quick temporary fix, a more permanent and functional RC gate design was needed afterward. The first RC gate design focused on fitting two RC cars underneath while also supporting an attached AprilTag. However, this initial design (shown in Figure 14) was too large to print on a single print bed, so a dovetail connection was used to join the two sides of the gate. This version did not perform well. The gate was unstable and easily fell over, and the AprilTag holder was unreliable. I initially attempted to hold the AprilTag in place using a spring mechanism, but this didn't work well. Additionally, the AprilTag holder was attached to the gate using snap bolts.

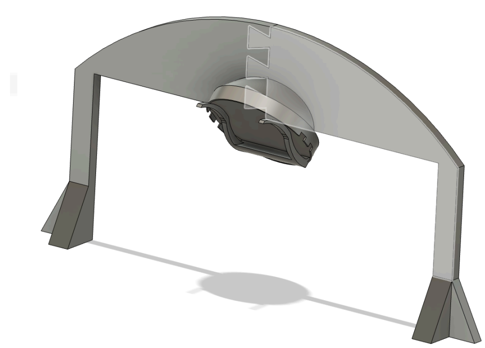

Figure 14: RC gate iteration 1

The second iteration of the gate introduced a sliding connection system that allowed the two halves to lock together more securely. This version also included support bars at the bottom of the gate to improve stability. While these changes improved the design, the iteration was not pursued further. At this point, the team realized that a more modular system would be beneficial, and reusing components from the AgXRP rail system offered a better solution.

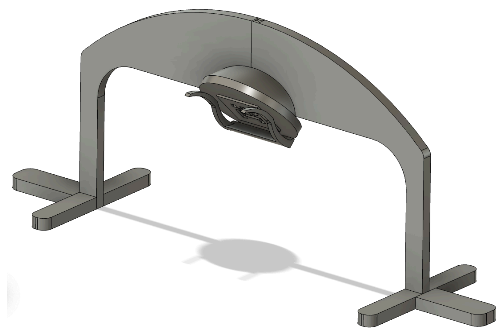

Figure 15: RC gate iteration 2

The AgXRP rail system is very modular, allowing both the height and width of the RC gate to be easily adjusted. This flexibility makes it possible to accommodate more RC cars by expanding the gate as needed. Additionally, the system was already proven to be stable and provided convenient ways to attach other components. To mount the AprilTag holder, I used the AgXRP rail mount that attaches directly to the top of the gate and is commonly used with other AgXRP parts.

Designing the AprilTag holder itself required many iterations. The first attempt involved a gear system that allowed the AprilTag to rotate. This design, however, was too loose, causing the gears to not rotate well. Through multiple iterations, the tolerances of the gear system were adjusted until it could reliably rotate in 15 degree increments. This made it possible to test different angles to determine the best orientation for the RC car's camera. Rather than creating a large stand for the AprilTag using snap bolts, I reused an existing AprilTag holder design from the ISS XRP, which allows the tag to snap on and off easily. This holding piece was mounted on top of a rod connected to the gear system I designed. The entire assembly used AgXRP rail mount, making it simple to attach and remove from the RC gate. The final AprilTag holder design is shown in Figure 18, and it successfully rotates, supports multiple AprilTag designs, and connects securely to the RC gate.

Figure 16: AgXRP rail used for RC gate

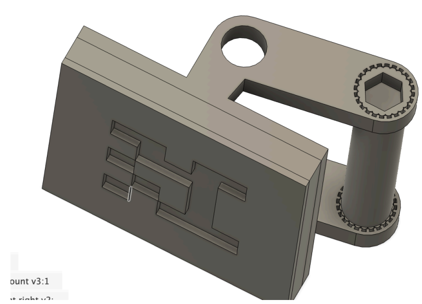

Figure 17: RC gate AprilTag holder iteration 1

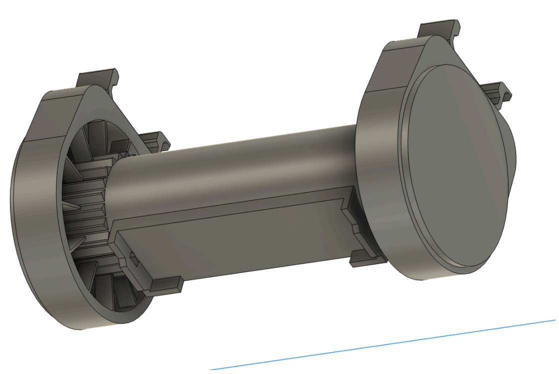

Figure 18: RC gate AprilTag holder iteration 2 (Final version)

Implementation

Manufacturing Process

The RC gate was designed in Fusion 360 and made completely with 3D printing for rapid prototyping.

Assembly

To assemble the RC gate, the end-to-end (ETE) connectors are first attached to two rail sections to form the vertical supports. The gate height consists of two rail sections, while the width is made up of three rail sections connected using ETE connectors. Right angle (RA) connectors are used at the corners to join the rail sections together. To ensure stability, a snap bolt and hex clip are added to secure each connector in place. The AprilTag holder is attached to the top middle rail of the gate using two side pieces with rail connectors. The AprilTag holder rod is then placed between the two geared connector pieces, creating a tight fit that still allows rotation. A 3D-printed AprilTag is inserted into the holder, after which the race track can be set up based on the meanings of each AprilTag. An example race track configuration, where each gate represents a unique track piece, is shown in Figure 19.

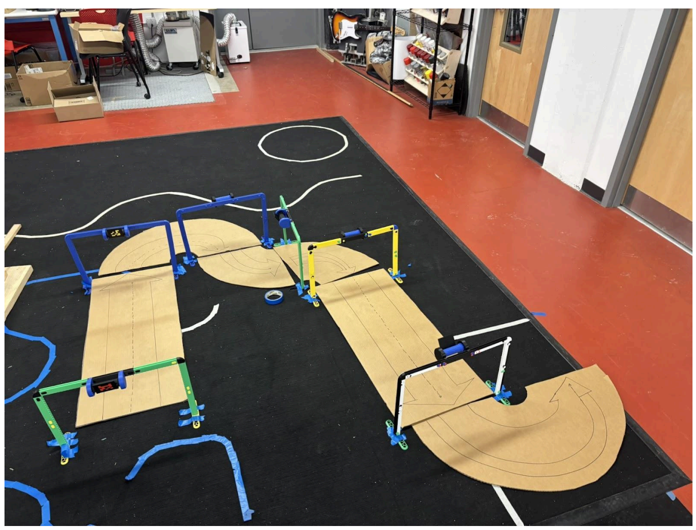

Figure 19: Example RC race track

Next Steps

Due to a lack of white filament, the current AprilTags do not have the correct black-and-white color contrast. New AprilTags with accurate coloring will need to be printed. In addition, the race track must be tested with the RC car to determine the best angle for reading the AprilTags. Furthermore, the code that connects the AprilTag track pieces to the game still needs to be completed. Finally, additional AprilTags representing more track pieces should be created to expand the game and provide a more complete racing experience.

Electrical

The Electrical team focused on fusing the XRP's wheel encoder data with that of its built-in IMU sensor. The sensor fusion will allow for more accurate state variable measurements and limit the impact of error, such as wheel drift, in such predictions.

Odometry & Movement Sensing

The AR Kart uses robust filtering techniques to properly parse sensor data and determine the correct position of the AR Kart.

Kalman Filter

The AR-Kart uses two different types of sensors: Two encoders (one for each wheel) and a gyroscope on the IMU. The ultimate goal is to create a Kalman Filter that fuses the data from the encoders and the IMU together and provides accurate next-state estimation from sensor readings. This semester was focused on the odometry calculations for each type of sensor. In particular, ECE focused on the calculations for the encoder, and CS focused on the calculations for the gyroscope. The immediate objective was to have the code generate continuous changes in x, y, and θ calculations that could be later fused with the corresponding gyroscope calculations.

Implementation

Before putting the calculations into code, the team modeled a specific situation (the robot turning in a circle in one direction) to get an idea of how the encoder data worked. Encoders give small measurements (ticks) every time the wheel moves. Using the circumference of the wheel, arc lengths can be calculated for the left side and right side of the Kart. The goal was to find out the arc length travelled by the middle of the Kart (the overall change in x, y, and theta for one movement). Trigonometry helped as well as a system of arc-length equations that gave the radii and change in θ, and then used that to find the target arc length. From there, calculations can be made for the change in x, y, and θ for one movement.

Since these whiteboard calculations were for a specific scenario, they were generalized with an algorithm in code. The code is a script that contains a class called DiffDriveOdom. It declares objects from the class Encoder (defined in encoder.py) and takes in hard-coded dimensions of the robot. Then, it performs the next-state estimation algorithm, which is called in the main() of the file. This script also samples and prints out estimated values, which were useful for testing on a sample robot because it made the progress of the continuous sensor-value-generation easier to observe.

Issues

Issues occurred with importing the Encoder class objects into the file. However, these were resolved by using the Drivetrain class to generate encoder data instead. The next steps will be tuning a Kalman Filter that fuses the sensor calculations from both the gyroscope and the encoders, as well as further testing of the odometry calculations from both sensors.

Testing

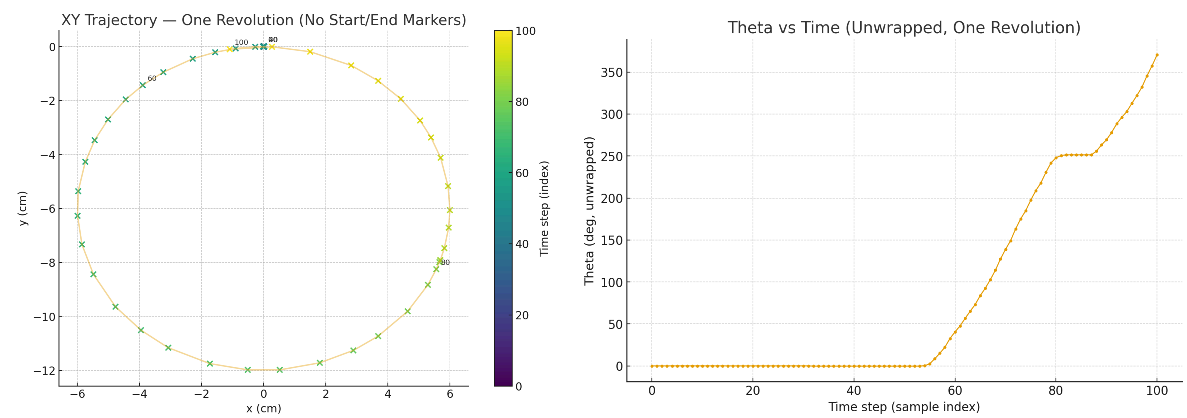

Figure 20: Predicted path of odometry with time steps over circular actual path (left); Theta displacement of robot through time over one circular revolution (right)

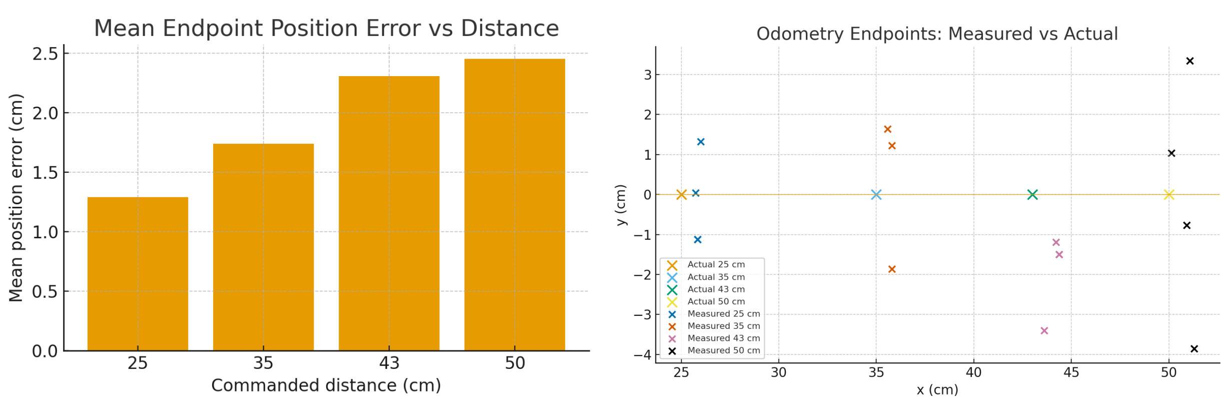

Figure 21: Results of odometry predictions versus actual travel for different distances (left); Summary of mean error of odometry prediction over different commanded distances (right)

The testing strategy involves two different approaches. The first was to test how well the current odometry system can accurately map a given trajectory. Figure 20 depicts a graph of the predicted trajectory of the AR Kart over different time steps to visualize how well the odometry system was able to map the perfect circle trajectory that it was given. For the next approach, multiple trials were conducted where the XRP robot was commanded to move a controlled displacement, then compared the robot's odometry-based position estimate to the actual measured displacement. The goal of these trials was to evaluate how accurately the odometry predictions tracked the robot's true motion and to quantify any systematic error or drift over repeated runs.

CS

Camera Streaming

Purpose

In the overall AR Kart project, the camera streaming is necessary in order to achieve the augmented reality aspect of it by providing the camera feed information needed in order to overlay the game UI on top of it. The overall requirements that the system needed were that the camera streaming needed to be able to send the camera feed over Wifi to Godot so that it can be utilized for the development of the game. Furthermore, the camera feed needed to be reliable and low-latency, so that the camera feed is able to keep up with the gameplay and the Kart driving around the racetrack.

Implementation

Assembly

The assembly of the camera streaming system involved attaching an Arducam OV5647, which was the camera chosen to take in the video feed, to the Raspberry Pi Zero W/2 with a ribbon cable. The overall installation and setup process is explained in more detail and step by step in the following linked document.

Next Steps

After setting up the Arducam and Raspberry Pi Zero W/2 according to the given instructions, there were a couple of things of note that may need to be addressed for an improved implementation.

First and foremost, the latency and quality of the streaming definitely need to be improved in order to be up to par and keep up with the AR-Kart racing around the track, and for the visuals to be accessible. This is likely because the access point set up on the Raspberry Pi Zero W/2 does not provide a strong enough connection in order to stream the data in real time and with good resolution. This could be remedied by adding modifications to the Pi that include an antenna that may help with the strength of the Pi's access point.

Next, it would be a good idea to streamline the installation and setup process even further so that there are no issues with the setup in any way.

Finally, the camera feed needs to be incorporated into the Godot gameplay so that the game UI and HUD elements are overlaid on top of the camera feed. Throughout the semester, RIT has already worked on many of the gameplay features, and it likely just comes down to combining the camera feed that the CS team has implemented with their gameplay elements.

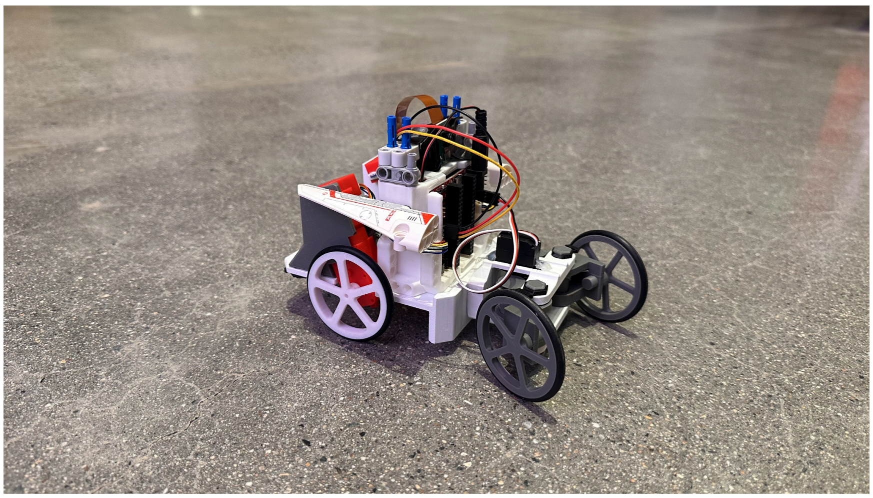

Figure 22: AR Kart prototype with Arducam strapped on the back

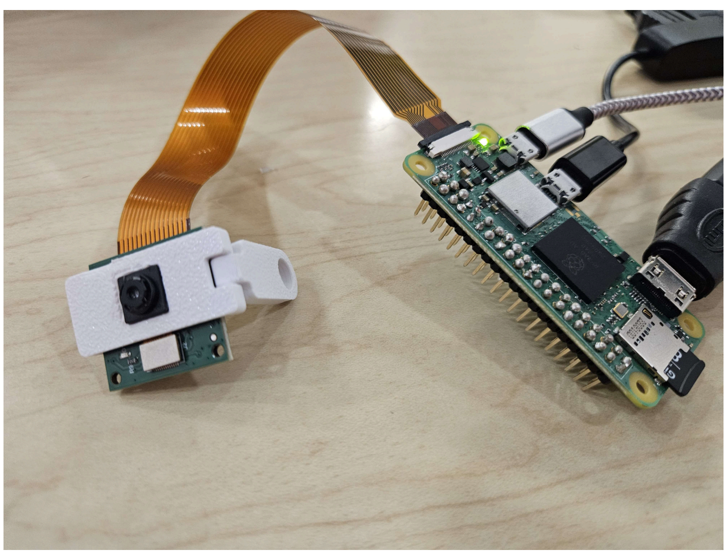

Figure 23: Separated Arducam connected to the Raspberry Pi Zero W/2