Race of Doom - Cyber Security

DESIGN DOCUMENT

Team 43
Client: Timothy Bigelow
Advisor: Timothy Bigelow
Team Members/Roles

Andrew Kraft -
Jack Doe - Project Manager
Gavin Petrak - Team Organization
Jacob Nedder - Testing, Team Coordination
Peter Wissman - Client Interaction

Team Email: sdmay24-43@iastate.edu
Team Website: https://sdmay24-43.sd.ece.iastate.edu

Revised: 12/3/23 (Version 2)

Executive Summary

Development Standards & Practices Used

At this time, the standards for circuits, hardware, software, and so on are not yet set out and will be determined between the various teams in the Race of Doom. When they are, this document will be properly updated to reflect these standards.

Summary of Requirements

- Develop an RC vehicle capable of autonomous steering to navigate a hostile environment at speed.
- Coordinate with the other Race of Doom teams to develop Terms of Engagement for vehicle and track design.
    - This document will develop to reflect the proper requirements once they have been set out.
- Provide weekly reports on the progress of our design to the client.

Applicable Courses from Iowa State University Curriculum

Cpr E 288

EE 230

Com S 227/228

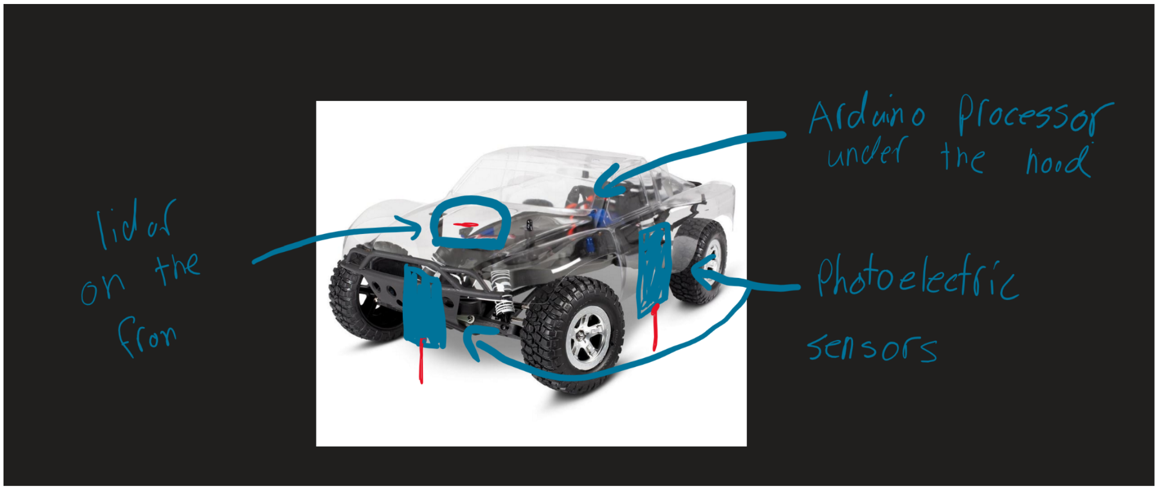
New Skills/Knowledge acquired that was not taught in courses

- Remote control of a vehicle.
- Autonomous Steering algorithms.
- Design techniques to develop an all-terrain vehicle.
- Shielded transmission to prevent remote car hijacking.
- Ergonomic controller design.
- Energy-efficient electronic vehicle wiring.

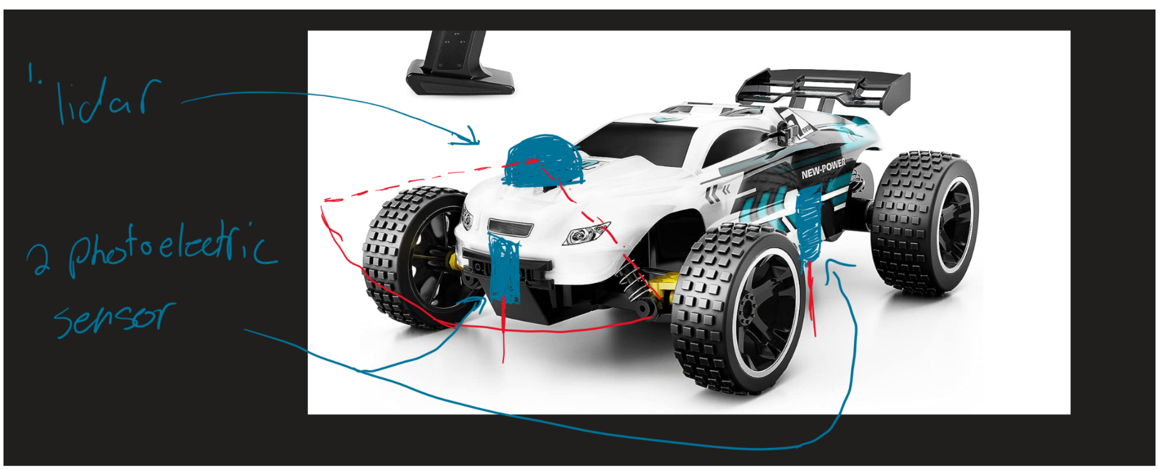
Table of Contents

List of figures/tables/symbols/definitions

Initial design of project:



Revised design of project:



1 Team, Problem Statement, Requirements, and Engineering Standards

1.1. Team Members

Andrew Kraft, Jack Doe, Gavin Petrak, Jacob Nedder, and Peter Wissman

1.2. Required Skill Sets for Your Project

- Remote control of a vehicle.
- Autonomous Steering algorithms.
- Design techniques to develop an all-terrain vehicle.
- Shielded transmission to prevent remote car hijacking.
- Ergonomic controller design.
- Energy-efficient electronic vehicle wiring.
- Proper documentation and communication with clients.
- Ability to communicate with other teams to develop and run a race.

1.3. Skill Sets Covered by the Team

- Ability to communicate with other teams to develop and run a race. - Jacob Nedder, Peter W
- Shielded transmission to prevent remote car hijacking. - Jacob Nedder
- Experienced in multiple coding languages - Peter W, Gavin P
- Embedded systems and external communicators - Gavin P

1.4. Project Management Style Adopted by the team

We have decided we are going to use an Agile approach for managing project work. As the development of the RC car will be rapid and ever-changing, we must be properly prepared to be as flexible as possible once construction of the RC car's features is underway.

1.5. Initial Project Management Roles

Andrew Kraft - Testing, Circuit Design

Jack Doe - Project Manager

Gavin Petrak - Team Organization

Jacob Nedder - Testing, Team Coordination

Peter Wissman - Client Interaction

1.6. Problem Statement

Our project is attempting to continue the development of self-driving vehicles for purposes such as crash detection, crash prevention, and artificial intelligence learning. We are striving to ensure a vehicle can drive through multiple types of obstacles, even those that might not necessarily be on a road, to ensure system adaptability and decision-making.

1.7. Requirements & Constraints

Team Requirements:

- Weekly meetings with the individual team, and periodic meetings between all Race of Doom teams.
- The full design document must be complete by the end of Semester 1.
- The general idea is to modify an existing car with new specs so every car team starts at the same point.
- A functional car must be fully operational by the end of Semester 2.

Design Requirements:

- The project must stay within the given CPR E program's budget
- The RC Car shall autonomous steer away from obstacles on the track
- The RC Car shall stay within the bounds of the track
- The RC Car shall be protected from the Tracks hacking source
- The Driver shall control only the speed and acceleration of the RC Car
- The RC Car shall make multiple laps around the track

Constraints:

- Our car must cost less than $500 to build, modify, and develop. This is subject to change pending funding from Caterpillar.
- The car should be able to sense its environment using a variety of sensors installed on the vehicle.
- Cars should be somewhat autonomous: the car can only move forward or backward by user input, and steering will be determined by sensors on the car.
- Car communications security must be student-built to allow testing of the cyber security aspect of the project.
- Each obstacle for the car should be overcome as quickly as possible to win the race.

1.8. Engineering Standards

IEEE 802.11 (Wi-Fi) Standards: IEEE 802.11 standards for Wi-Fi communication are relevant for remote control and data exchange between the remote control devices and the cars.

ISO 6469 - Safety of Electrically Propelled Road Vehicles: While the vehicles are small in scale if they are electrically powered, ISO 6469 might still apply to address electrical safety aspects, especially if they use lithium-ion batteries or other electrical components.

IEEE 1275 - Open Firmware Standard for Embedded Systems: If the remote control cars use embedded systems or microcontrollers, adherence to relevant firmware standards can be important for compatibility and reliable operation.

Radio Frequency (RF) Standards: Depending on the communication technology used for remote control, there may be specific RF standards that apply to ensure proper signal transmission and interference avoidance.

Electromagnetic Compatibility (EMC) Standards: EMC standards can be relevant to ensure that the operation of the remote control cars does not interfere with other electronic devices and vice versa.

1.9. Intended Users and Uses

The intended users of this project will be those who are interested in the progression of autonomous vehicles, as we will be able to pass on the research we have done during our Senior Design Project over to the university so future students can benefit.

Self-driving vehicles are an ever-growing industry, and creating more test cases like these is instrumental in the growth of these vehicles to minimize risk and promote machine learning. Even on a small scale, there is a plethora of information available to be obtained and researched.

3 Project Plan

3.1 Task Decomposition

- Identify which type of RC car we would like to modify for this project.
- Determine which sensors need to be added/changed for project criteria.
- Research how a computer could communicate with the RC car using various signals.
- Create a bridge connecting the RC car to the computer and/or host.
- Ensure a secure method of communication between the car and the computer is made.
- Build software to handle information sent from the RC car to the PC.
- Create autonomous features based on the software previously built.
- Thoroughly test both the RC car and software before the final race.

3.2 Project Management/Tracking Procedures

Will use a waterfall+agile approach to manage our project. This project management style works well with the system we'll use to track our progress over the year. This management style will also make tracking tasks through the project lifecycle easier and more efficient for the team.

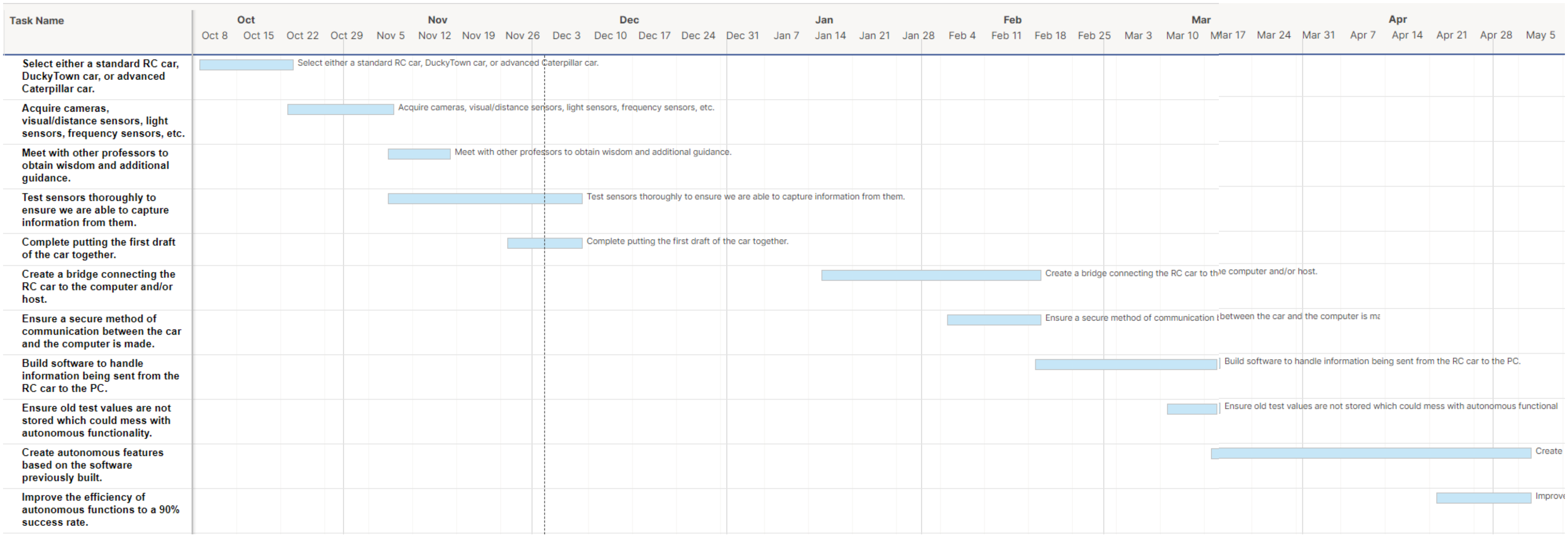
We will be tracking our goals using Gitlab. This will make creating, editing, viewing tasks, and a To-Do list much easier and straightforward to obtain for all developers.

3.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

- Identify which type of RC car we would like to modify for this project.
  - Select a standard RC, DuckyTown, or advanced Caterpillar car.
- Determine which sensors need to be added/changed for project criteria.
  - Acquire cameras, visual/distance sensors, light sensors, frequency sensors, etc.
- Create a bridge connecting the RC car to the computer and/or host.
  - Know which radio frequency to transmit and which computer program to use for connectivity.
- Ensure a secure method of communication between the car and the computer is made.
  - Create cyber security measures to prevent signal hacking.
  - Make a secure method for reconnection in case a signal is lost at any time.

- Build software to handle information sent from the RC car to the PC.
    - Obtain values from sensors and store them for later processing.
    - Create functions to send information for turning, acceleration, etc.
- Create autonomous features based on the software previously built.
    - Use the information stored in the car's processor to make decisions and call functions for action.
    - Ensure old test values are not stored which could mess with autonomous functionality.
- Thoroughly test both the RC car and software before the final race.
    - Improve the efficiency of autonomous functions to a 90% success rate.
    - Ensure any user input can quickly be inputted into the car.

3.4 Project Timeline/Schedule



Tasks in order:

- Select a standard RC, DuckyTown, or advanced Caterpillar car.
- Acquire cameras, visual/distance sensors, light sensors, frequency sensors, etc.
- Create a bridge connecting the RC car to the computer and/or host.
- Ensure a secure method of communication between the car and the computer is made.
- Build software to handle information sent from the RC car to the PC.
- Ensure old test values are not stored which could mess with autonomous functionality.
- Create autonomous features based on the software previously built.
- Improve the efficiency of autonomous functions to a 90% success rate.

3.5 Risks And Risk Management/Mitigation

Now that we have gotten a general idea of the requirements and constraints for our project, the main risk we face is to build a functional RC car by the end of the year. This includes designing and implementing changes to a preexisting RC car that the team can agree on. The current risks we can identify are taken from section 2.3 above, and the possible solutions are subject to change if we can invent a better idea when we come across said risks.

- Disagreement between which RC car we should use as a base (0.5)
    - This can be solved by using communication between the team, and if there is a split, we can use a voting process as a last resort
- Which parts should we purchase to be implemented, and how many parts to buy in case some break or malfunction (0.75)
    - This can be solved by researching the parts we need and only purchasing the minimum amount we need to test and implement them. If some parts take longer to get, we can purchase them earlier so we are never stagnant in our development.
    - This will also need good budget management for our expenses so we don't go over the limit set.
- Making sure our implementations are up to the team's standards (0.6)
    - This can be done through testing what the RC car can handle, physically and electronically, while also identifying possible risks we may encounter during the final race.
    - Ideas and communication between the team play an important role in this risk, as all team members must be honest about how they feel about the current state of the project, and willing to bring forward their ideas on how to make the project better.

As of now, those are the main risks we will most likely encounter while completing our project, but there is no doubt other issues may occur. When that happens, we will work as a team to generate ideas and solutions to those problems to ensure that all members of the group are satisfied with the final build of the project.

3.6 Personnel Effort Requirements

Task (taken from section 2.1):	Expected person-hours to complete:
1. Decide on RC	2
2. Determine Sensors	4
3. Research computer interface	4
4. Create a bridge connection	12
5. Ensure secure communication	16
6. Build information handling software	28
7. Create autonomous features	40
8. Thoroughly test the product	40+

1. Determining the RC car used in our project will be a process completed in conjunction with the other race teams at our weekly meetings. A decision should be reached within the course of a single meeting.
2. Determining the sensors we will use in our project will be a relatively simple discussion once again taking place with the other groups determining our joint parameters for the race to come.
3. The computer interface research can take place in conjunction with the decision on sensors, once we know the vehicle base we will be using for our project. This is solely an information-gathering phase, and as such it will be briefer than the following development.
4. Establishing the bridge connection will involve programming from the appropriate wireless libraries, and then a short period of debugging to ensure the connection functions properly.
5. Ensuring a secure connection will take longer than establishing the connection. It will involve rigorous testing of edge cases to ensure that a potential attacker cannot gain a foothold in our communications by taking input outside of the expected use cases.
6. Developing the information handling software will require all previous tasks in the project to be completed. It will likely involve every member of the group working on individual components of the whole before combining these parts into a single piece of software.
7. Creating the autonomous features for our vehicle will be one of the two largest tasks of our project. This will involve a combination of development by our computer/software/cybersecurity engineers, and the development of a physical component for our prototype.
8. The final task of our project will be stress testing, trial runs, and bug fixing. This task will occur in conjunction with the development of our autonomous features, to ensure the project ends in a successful prototype, and it will continue for as long as our group has time- thus, the allotted hours are subject to more variance than the other, easier to predict, tasks.

3.7 Other Resource Requirements

This project will require:

- An RC Car body
- A remote control
- 3D printing access for custom parts development
- A stable IDE compatible with the code used in our RC vehicle
- Mechanical components for any vehicle enhancements-
  - screws
  - axels
  - wheels
  - bearings
  - and other such parts
- Sensors for any vehicle enhancements-
  - Infrared Sensors
  - Radar Sensors
  - Sonar Sensors
  - Rotational Sensors

4 Design

4.1 Design Content

For our project, we need to design a RC car that will semi-autonomously navigate through a track with obstacles and hacking. The driver of the RC car will have access to increase or decrease the speed of the vehicle but have no control over the direction the vehicle goes in. The direction will be changed based on a sensor on the front of the bot that detects where there is an obstacle and where a path is available for the bot to travel through. There will also be sensors on the side of the bot that will detect the borders of the track. The border will consist of a specific color of tape that can be detected by a simple photoelectric sensor. The bot will have a processor on board that will compute everything related to detecting obstacles, borders, and deformities. This processor will tentatively be an Aurdino until we have found a better solution.

What we need based on requirements:

- A way to detect the borders of the track
- A way to detect objects in front of us
- A way to detect deformities in the track, for example, holes, bumps, and craters
- Steering is handled autonomously
- User controls speed of bot
- Protections against remote tampering
- Protections against potential physical hazards

Track team must guarantee:

- Car must stay flat to the track (can have ramps, banked turns, etc). The bot shouldn't be flipped from the track having a split level
- The car shouldn't be severely damaged by obstacles

4.2 Design Complexity

This project meets the requirements for technical complexity as it consists of multiple components and subsystems that work together using distinct scientific and engineering principles. Our end product is a partially autonomous RC car that can quickly navigate through a hostile environment to reach a goal. In order to accomplish this, we will need to modify an existing RC car, taking out its main control components and replacing them with a customized Arduino board. The only communication between our remote control and the car will be control of acceleration, which will be handled by the preexisting radio communication. The car itself will use LIDAR and photoelectric sensors to identify obstacles and remain on the track, feeding this data to the onboard systems that will make the decisions on when and where to turn the vehicle. These decisions will need to be programmed into the board, meaning our group will have to make use of an embedded coding language. Beyond this basic maneuverability, our bot will need to be durable enough to survive any of the obstacles included as part of the track design. Our car must be capable of navigating sharp turns, vertical changes in terrain, moving around holes, and avoiding other obstacles designed to damage the vehicle. This means we will need to redesign the shell and suspension of the car, implementing mechanical principles to minimize the damage of impacts and maximizing the chance of recovery from a flipped position.

4.3 Modern Engineering Tools

Embedded Programming Language: For coding the Arduino onboard processor, we plan to use the Arduino IDE. This software allows us to write code in a simplified version of C and C++. It's perfect for controlling the RC car's behavior, processing sensor data, and making autonomous decisions.

Sensor Data Processing: We'll be relying on the Arduino IDE to process data from our sensors, such as LIDAR and photoelectric sensors. This is where we write code in C/C++ to interpret data and make real-time decisions about obstacle detection and path planning.

Remote Control Interface: To create a user interface for remote control of the car's speed, we intend to use programming languages like Python or JavaScript. These languages are great for building a user-friendly control interface that communicates with the car through wireless communication methods.

Cybersecurity Implementation: Implementing cybersecurity measures to protect against remote tampering will be a crucial part of our project. We'll incorporate encryption and authentication protocols within our Arduino code to ensure the security of the communication.

Simulation and Testing Software: To test our algorithms and simulate the RC car's behavior before deploying it in the real world, we're considering using software tools like ROS (Robot Operating System) or simulation environments like Gazebo. These tools will help us iron out any issues before putting the car on the track.

Data Visualization and Debugging: For visualizing and debugging sensor data, we're looking at using software like MATLAB or Python with libraries like Matplotlib. These tools will help us analyze sensor data and identify any issues in the system.

Version Control: We'll use Git for version control. It's a great tool for managing code revisions, tracking changes, and merging contributions from our team members, ensuring a smooth collaborative development process.

Programming Environment: The Arduino IDE will be our primary programming environment for the Arduino onboard processor. We may also use text editors like Visual Studio Code or Sublime Text for writing and editing code, as they offer useful features like syntax highlighting and code completion.

Algorithm Development: If we create custom algorithms for obstacle detection and path planning, we'll use C/C++ within the Arduino IDE. We may also integrate additional libraries or external tools as needed to enhance our algorithms.

Firmware Updates: When it's time to update the firmware on the Arduino board or other microcontrollers, we'll use the Arduino IDE. This process involves uploading new code to the hardware components to implement improvements and updates.

Hardware: Things like the sensors being used for data collection, battery/power source, signal transmission between RC car and controller, and the prebuilt RC car to modify haven't been decided yet. The list below mentions the main components we will probably need once the criteria, requirements, and funding are finalized..

- Sensors: We will need distance sensors and light sensors at the very least for measuring distance and steering..
- Motors: We may need to replace the one that comes with the RC car already. Also, mounting a sensor on a servo motor will help us with data collection
- 3D printer: Designing the skeleton of the RC car and any of the physical components that may need to be remodeled inside and outside of the RC car
- Signal Connector: We will need a way for the controller to connect to the RC car without any interference from the track team
- Battery: May need to replace the battery depending on the power requirements of the added components

- Basic components and connectors: Just things like wires, cables, screws, etc, that will be needed for assembly and connecting devices

4.4 Design Context

Area	Description	Examples
Public health, safety, and welfare	In our evolving society, self-driving cars and autonomous vehicles are becoming a reality more and more every day. With this emerging technology, creating additional test situations such as our senior design project can help find reliable solutions for public safety and welfare by minimizing hacking attempts and malfunctions.	Further developing knowledge in the world of automotive vehicles, reducing risks of malfunction and/or hacking from third parties.
Global, cultural, and social	At the moment, self-driving vehicles and the technology behind them are only available to a select few, such as the upper class and possibly farmers when it comes to automated farming.	Development of this test is a smaller scale to reduce risks of harming humans, animals, or the environment.
Environmental	Our project will not use any gases or fossil fuels in general and will be run 100% on electricity. Utilizing the least amount of energy for optimization is a priority.	No usage of fossil fuels, and minimizing the amount of electricity used when the car is running is essential.
Economic	Even on our small scale, it is likely this experiment will cost at least $1000. Reasons like this are why automated vehicles are still in their infancy, so we will try to cut down on these costs as much as possible without sacrificing safety features.	Minimizing cost allows for further development of automated vehicles heading in a consumer-level market, but all safety requirements must still be met.

4.5 Prior Work/Solutions

When it comes to autonomous vehicles, a few clear examples come to mind. The recent rise of Tesla's self-driving cars is likely the most well-known example, whose engineers are attempting to allow fully automatic driving in any real-world scenario. Although Tesla brings some of the most advanced self-driving technology to the table, there are still plenty of bugs that prevent drivers from completely relying on the functionality of this software.

- Pros: Ideas for navigating a car on roads, knowledge of how sensors could be used to keep the car moving in a straight line, examples for visualizing obstacles in a computer environment.
- Cons: A high budget like the ones used for Teslas is not achievable with our funding, and comes from a large-scale model where constraints are likely different.

If we were to scale down the high aspirations of Tesla, another example of a group attempting to develop autonomous vehicles could be the show BattleBots. Here, groups compete against each other to use sensors and traps to fight their opposing vehicle. Our project is very similar to this example, where we must race around a track as fast as possible. We will be able to use ideas from a show like BattleBots and transform them into our designs for a race. However, we must also keep in mind budget is a large factor for a project like this, and we likely do not have the same funding as groups in the BattleBots show.

- Pros: Applications for modding/creating RC cars have been explored a bit, observable results from the show are easy to come across, and sensors are similar to ones we may be using.
- Cons: The objective of these RC cars is not the same as the project we have been handed, and the budget is likely not as high as those observed in the show.

4.6 Design Decisions

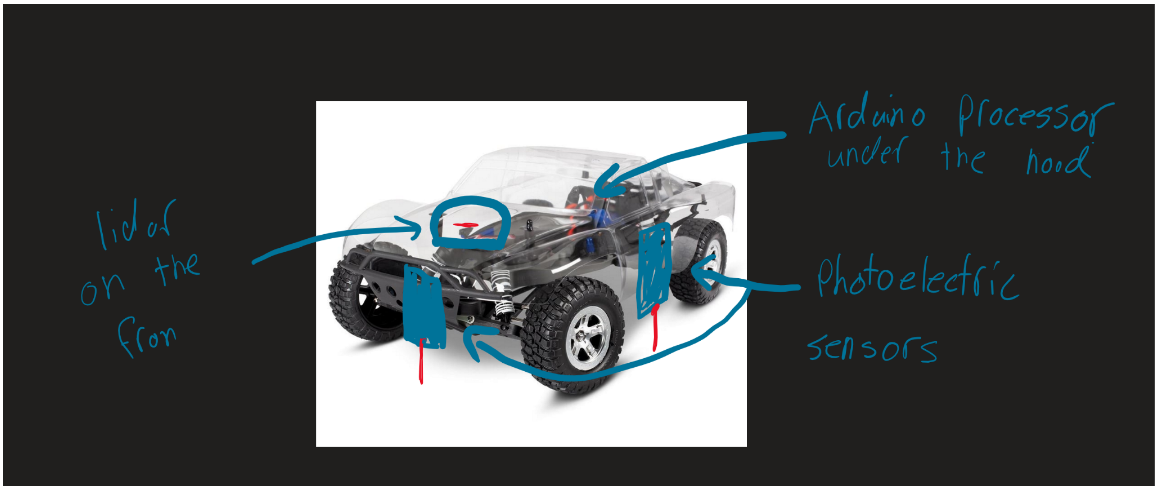
- Car selection: What kind/model of RC car would we like to modify? In turn, how much work will we need to put in when it comes to adding external sensors and software?
- Sensor selection: What kind of sensors would we like to have on our car? Are there any specific obstacles where a special type of sensor could come in handy? How easily can said sensors be connected to our Arduino to be integrated into our software design?
- Computer connection: What software would we need to connect our RC car and computer? Are there any limitations to this software? What programming languages (C, Java, Python, etc.) are we going to use for both connection and software development?

4.7 Proposed Design

Our team has begun to look into possible RC cars to buy. We've communicated with the other RC team what they are looking into buying, and both teams are going to have differing approaches. The other team working with an RC car has chosen a more expensive car that has more sensors built-in. Our team will be taking a more complicated approach, which focuses on an RC car that does not contain built-in sensors. We've also begun looking into what processor we want to use, which is going to be a Arduino processor as it is cheap and effective. We are also planning on utilizing a variety of sensors, such as photoelectric, Lidar, and more which will be plugged into the Arduino board.

4.7.1 Design 0 (Initial Design)

Design Visual and Description



Here we have a very simple mock-up of our car's design. We haven't decided on specific parts yet, but as a baseline, we have the following:

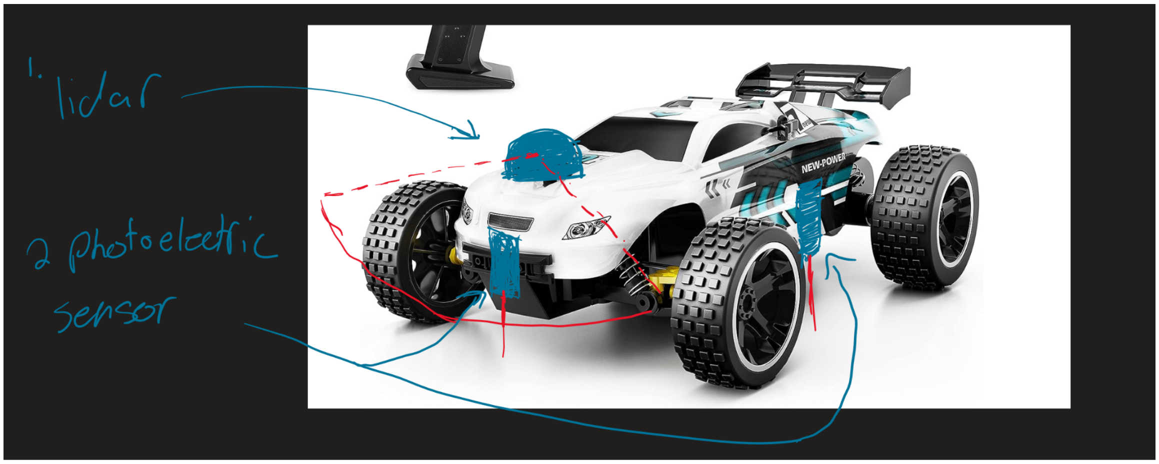
- Lidar on the front of the bot for object detection
    - This will allow the bot to detect obstacles on the track
- Photoelectric sensors on both sides of the bot as well as the front of the car
    - detects the borders of the track or holes
- Our processor will be hidden away underneath the hood of the car to protect it from damages
    - There are thoughts of incasing the processor with some sort of protection against crashing
- The possibility for other sensors are still being considered like for example, an IR sensor

Functionality

Our user will only be allowed to control the car's acceleration. All the turning and maneuvering will be completely controlled by the microcontroller inside of the RC's body. This initial design shows where the possible components will be placed onto the car but lacks many details for functional requirements.

4.7.2 Design 1 (Design Iteration)

After a more detailed look into the specific components, we've iterated on our previous design. It has a similar look and placement of components as design Zero.



We will use these specific components:

1. Lidar - DFR0315 from DFRobot. This lidar spins on its own, so we can detect objects in the track with this component. We won't need to build a motor and IR sensor combo
2. PhotoElectric Senors - SEN0239 from DFRobot. This simple photoelectric sensor will let us detect the track's borders by detecting the change in color
3. MicroProcessor - Raspberry Pi 4 - We've decided to use a Raspberry Pi for our processor because it is more versatile than an Arduino. This processor will also be hidden underneath the hood of the vehicle in order to protect it from the elements.
4. RC car - Tecnock RC Racing Car. This car is a simple and cheap RC car which allows us to spend more money on our components

4.7.3 Design Visual and Description

We have changed the vehicle to be slightly smaller and less expensive than in design zero, as it uses more elements you would find in a common RC car. This is so we can focus more of our budget into our data-collecting mechanisms, which will allow for better autonomous functionality.

4.8 Technology Considerations

We should have access to many different ways to code our microcontroller. Were hoping for an easy way to hijack the RC car to feed data from the processor to the car's motors. Understanding how a Raspberry Pi can communicate with an external machine like a PC will be one of the most critical pieces of functionality. On top of this, each individual sensor's datasheet will need to be understood so that we are able to work through the data each sensor is giving us.

4.9 Design Analysis

At this moment in time, there is not a lot of analysis to be taken with the information we have present. We still do not have a physical prototype, so a proper analysis on the design has yet to be done.

5 Testing

Testing is an **extremely** important component of most projects, whether it involves a circuit, a process, a power system, or software.

5.1 Unit Testing

Lidar Sensor

- Individually, we must determine how fast the LiDAR sensor can realistically scan its environment after taking into account possible errors that may come with scanning too quickly. We will also need to test that the connection to the Raspberry Pi is secure after we connect the Raspberry Pi to the computer. Bridging this gap will allow us to debug the LiDAR and come up with further test cases.
- When we put the LiDAR sensor on the RC car, we will need to test movement while a scan is taking place. If we can find a way to properly scan values while moving, our race can be completed much faster. Thus, it is worth our time to explore testing this feature. We will also need to attempt testing LiDAR scans while the car is on uneven ground, which may be present on the track.

Photoelectric Sensor

- Individually, we will need to connect the sensor to the Raspberry Pi to communicate with our computer to interact with the sensor's data. Then, we will need to see how much range our Photoelectric Sensor has been equipped with, as well as the margin of error present from multiple distances.
- The Photoelectric Sensor on the RC car will need to be tested for detecting walls, cliffs, and other obstacles present on the track. Our software will need to be adjusted to match values recorded by the sensor accordingly, as the RC car will take up space around the sensor.

Raspberry Pi

- Individually, we must test the connectivity of the Raspberry Pi to the computer for debugging. We should also test how far away the Raspberry Pi can be before disconnecting from the remote host, as this could cause problems with testing and debugging further down the line.
- When putting the Raspberry Pi on the RC car, we need to make sure no signals interfere with the RC car's basic movements, which can be done with some simple trial and error. Then, we must also make sure an ample amount of power is being supplied to the Raspberry Pi at all times and that this does not drain the battery of the RC car too rapidly.

RC Car

- We must make sure the RC car is able to properly traverse the variety of obstacles on the track before we can make the car autonomous. After all, if the dimensions or characteristics of the car cannot be fixed by sensors, we are going to have to look at alternative solutions for our car.
- We also must ensure that the car cannot run out of energy too quickly. We will need to determine the vehicle's maximum battery capacity once we acquire it to ensure it can pass the trials on the track.
- Also, we need to determine how far the RC car can travel away from the host before losing signal. This can be done by setting the vehicle far away and still attempting to communicate with it.

5.2 Interface Testing

Everything we add to the RC car will be running simultaneously during the final design, so each part must function correctly while interfaced with every other individual part. The main interfaces we can come up with are:

Between Lidar and Photoelectric sensors

- Must identify which sensor has priority function over the other at any given time (object detection using lidar sensor vs course detection using photoelectric sensor)
- Must also be able to communicate between each other to "build" the course in real time
- Testing will be done for each individual part to ensure they work correctly for their specific purpose, then used together to ensure the correct priority is being run by the Raspberriy Pi

Between Raspberry Pi and Sensors

- Raspberry Pi acts as the brain of the RC car so it must allow data to be received and interpreted from the sensors to determine the best course of action
- Must also link the sensors together so both data streams can be interpreted
- Testing will consist of displaying the interpreted data to the user

Between Raspberry Pi and RC car

- The Rathe spberry Pi will get information from the user (acceleration) and interpret the data incoming from each receiver to tell the RC car where to go
- RC car will execute those commands with correct functionality
- For testing, we will see how the RC car moves based on a variety of different situations it may come across in the race.

5.3 Integration Testing

There are a couple of different paths that are critical for our design to work. Because our project involves a combination of physical sensors and computer programming, we need all of our different components to be able to communicate with each other. The following integration paths are integral to our project:

1. The Lidar needs to communicate with the Raspberry Pi
2. The Photoelectric sensors need to communicate with the Raspberry Pi
3. Our Raspberry Pi must send signals to the RC car's onboard processor
4. The RC car's onboard processor must send signals to the Raspberry Pi
5. Our software needs to decode and interpret the information sent from the lidar and photoelectric sensors
6. Our Raspberry Pi needs to be the hub of communication between all the software and sensors

All of the above paths must be completed for our design to be successful. Paths 5 and 6 are especially critical concerning our requirements. These paths must be thoroughly tested and need to work for our autonomy requirement to be fulfilled. Without communication between the sensors and the Raspberry Pi, our car will not be able to seer autonomously.

All of these paths with be tested through individual tests based on what that path needs to accomplish. For example path 1, we will connect the lidar and Raspberry Pi for path one. Then, we will test the lidar with different objects in front of it and see if our software on the Raspberry Pi can get the data. After multiple rounds of this testing, we can determine whether or not this path is completed. The same method would be used for the rest of the paths when it comes to testing

In terms of tools, our main tools will be whatever debugging software we use to build the Raspberry Pi. This will likely be VSCode's debugging to test the software.

5.4 System Testing

System Level Testing Strategy:

For system testing, we will perform a comprehensive set of tests to ensure that all components work together as intended. The critical focus will be on end-to-end testing, covering both functional and non-functional requirements.

Testing Components:

1. Integration Tests: Continue validating the critical integration paths mentioned in the integration testing section. Ensure that communication between the Lidar, photoelectric sensors, Raspberry Pi, and the RC car's onboard processor is seamless.

2. End-to-End Functional Tests: Simulate the entire process of the RC car moving autonomously on the track. This involves starting from the input from sensors, processing on the Raspberry Pi, and output to control the RC car. Check if the car navigates accurately and responds appropriately to different scenarios.

3. Non-functional Tests: Include tests for system performance, reliability, and security. Given the potential for cyber attacks, evaluate the system's resilience against unauthorized access and ensure the security protocols are robust.

Tools for System Testing:

- Simulation Environments: Use tools like Gazebo or ROS (Robot Operating System) for simulating the entire system and testing various scenarios without physical implementation.

- Performance Testing Tools: Tools like JMeter or Gatling for evaluating system performance under different loads.

5.5 Regression Testing

Regression testing for our project will occur after each major step of sensor implementation, and after any major changes to the code handling our automated steering. As an example, the following checkpoints mark times we will need to go back through the functionality of the car to ensure it is still working:

1. After the initial installation of the Raspberry Pi into the car's circuitry
2. After hooking up the Lidar sensor to the Raspberry Pi
3. After hooking up the photoelectric sensors to the Raspberry Pi
4. After implementing the initial tests for pathfinding and steering
5. After any major alterations to our code
6. After implementing any components needed to ensure the car can finish the race successfully

At each of these times we will need to verify that the car still responds to remote input to control acceleration, that it can still turn and navigate to avoid obstacles, that the sensors function and accurately return their data, and that any additional functionality previously implemented still works as intended.

To verify this functionality, we will run the car through a series of tests to ensure that each part still meets the requirements set out for it in its initial unit, interface, and integration tests. As more and more aspects of the project come together to form the whole, each set of tests will need to be more rigorous to ensure we do not neglect checking a previously functioning component.

5.6 Acceptance Testing

Demonstrating Requirements Fulfillment:

Acceptance testing will focus on demonstrating that the design requirements, both functional and non-functional, are met.

Testing Components:

1. Functional Requirement Validation: Run specific tests to verify each functional requirement. For example, validate that the RC car can navigate autonomously, change direction based on sensor input, and respond to external stimuli appropriately.

2. Security Testing: Simulate cyber attacks to ensure that the system can withstand potential threats and remains secure during operation.

3. Usability Testing: Involve end-users or representatives to interact with the system and provide feedback on the user interface, autonomous navigation, and overall usability.

Involving the Client:

1. Client Involvement in Testing: Schedule sessions where the client can observe the system in action and provide feedback. This can include live demonstrations or remote sessions where the client can interact with the system.

2. Feedback and Iteration: Gather feedback from the client on whether the system meets their expectations and requirements. If adjustments are needed, iterate on the design and testing process accordingly.

Tools for Acceptance Testing:

- User Acceptance Testing (UAT) Tools: Platforms like TestRail or Zephyr that allow clients to provide feedback and track the progress of acceptance tests.

- Security Testing Tools: Tools like OWASP ZAP or Nessus for assessing the security aspects of the system.

5.7 Security Testing

For our project, security testing plays a larger role than might be expected. While we will not be handling sensitive data like finances or personal information, one of the core components of the Race of Doom project definition was the inclusion of elements in the track that attempt to "hack" the cars, feeding false data to sensors, confusing the vehicle's pathfinding, or even attempting to assume direct control with conflicting radio frequencies. Security testing will help us to mitigate these attacks from the track by hardening our sensors against these interferences.

To be effective, we will need to test extreme cases to ensure that our vehicle can still function. As an example, a photoelectric sensor will return different values when the car is passing over the ground when compared to passing through a tunnel. Since we do not know the track design, our vehicle must account for both options. Our LIDAR sensor will also need to be able to filter out false positives potentially caused by more reflective surfaces or conflicting artificial signals. We will need to ensure our control over the vehicle's acceleration is maintained in the presence of other radio control frequencies as well. Finally, tests for the vehicle's ability to navigate in unusual terrain will be of utmost importance. The track will not be straight, so being able to turn corners, navigate twisting passages, or come to a premature halt with unexpected timing will all be necessary.

5.8 Results

We have not conducted any testing as all of our tests must be conducted with our multiple components. We have not received all of our components yet, but we will begin testing as soon as we do. Once we test our components, we will ensure they are complacent with our requirements by doing test runs on the track. If our car can properly drive through the track as intended, we will have proven our testing and design work as intended. We will also update this section once we have actual results.

6 Implementation

At the start of next semester, we would like to start getting into the coding of our project. The code we create will greatly impact our vehicle's performance for the final race. So, starting the coding process early will allow us to test more before the end of the semester. We would also like to start testing all of the individual components and getting a hands-on understanding of how it works. This will help us make more effective code sooner, allowing for more promising iterations of our system. Lastly, we will be looking into all the electric components of the vehicle to learn the best way of connecting all the components together to have a working prototype and final project.

7 Professionalism

This discussion is with respect to the paper titled “Contextualizing Professionalism in Capstone Projects Using the IDEALS Professional Responsibility Assessment”, *International Journal of Engineering Education* Vol. 28, No. 2, pp. 416–424, 2012

7.1 Areas of Responsibility

Area of Responsibility	**Definition**	**NSPE Canon**	**IEEE Code of Ethics**
Work Competence	Perform work of high quality, integrity, timeliness, and professional competence	Perform services only in areas of their competence; Avoid deceptive acts.	to maintain and improve our technical competence and to undertake technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations;
Financial Responsibility	Deliver products and services of realizable value and at reasonable costs.	Act for each employer or client as faithful agents or trustees.	to be honest and realistic in stating claims or estimates based on available data
Communication Honesty	Report work truthfully, without deception, and understandable to stakeholders.	Issue public statements only in an objective and truthful manner; Avoid deceptive acts.	to seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, to be honest and realistic in stating claims or

			estimates based on available data, and to credit properly the contributions of others;
Heath, Safety, Well-Being	Minimize risks to safety, health, and well-being of stakeholders.	Hold paramount the safety, health, and welfare of the public.	to hold paramount the safety, health, and welfare of the public… to protect the privacy of others, and to disclose promptly factors that might endanger the public or the environment;
Property Ownership	Respect property, ideas, and information of clients and others.	Act for each employer or client as faithful agents or trustees.	to credit properly the contributions of others; to treat all persons fairly and with respect, and to not engage in discrimination based on characteristics such as race, religion, gender, disability, age, national origin, sexual orientation, gender identity, or gender expression;
Sustainability	Protect environment and natural resources locally and globally.		to strive to comply with ethical design and sustainable development practices
Social Responsibility	Produce products and services that benefit society and communities.	Conduct themselves honorably, responsibly, ethically, and lawfully so as to enhance the honor, reputation, and usefulness of the profession.	to improve the understanding by individuals and society of the capabilities and societal implications of conventional and emerging technologies, including intelligent systems;

Our addition to the Work Competence column is a copy of section I6 from the IEEE Code of Ethics. It discusses only taking on projects that you are fully qualified to handle, or that you have fully communicated the limits of your knowledge about. This ties directly into the idea of Work Competence, that work needs to be completed in a timely, quality, and competent manner.

Our addition to the Financial Responsibility column is an excerpt from section I5 of the IEEE Code of Ethics, in this case referencing a need to be honest and realistic both when returning the results of testing or experimentation, but also when delivering financial estimates or quotes. This has a direct link to the reasonable costs necessary for Financial Responsibility. The Need for quality is not directly referenced by the IEEE Code of Ethics in any of its sections.

Our addition to the Communication Honesty section is the full text of section I5 , as it is all pertinent to the concept. Truthfully reporting work means reporting errors accurately and truthfully, and it means being realistic about results and not taking credit for the work of someone else. If all of this is properly communicated to stakeholders, the concept is upheld.

Our addition to the Health, Safety, and Well-Being section is a combination of two excerpts from section I1. This row focuses on preserving the safety and well-being of stakeholders, both now and in the long term. Protecting the public, as well as promptly reporting any potential risks to the public and the environment, will ensure this principle is not neglected.

Our addition to the Property Ownership column comes from two sections of the IEEE Code of Ethics: section I5 and section II7. First, section II7 discusses treating all persons fairly, regardless of any characteristics. This is an important backbone to respecting any property, ideas, or information of any client. Section I5's excerpt references plagiarism, which fundamentally breaks this respect by taking the work of another and claiming it as one's own. To uphold the Area of Responsibility, no exceptions may be made.

Our addition to Sustainability comes from section I1 once again, in this case referencing the importance of ethical design and sustainable development. By ensuring both practices are upheld, the environment and natural resources can be properly protected and maintained.

Finally, our addition to the Social Responsibility column comes from section I2 of the IEEE Code of Ethics. Social Responsibility revolves around providing benefits for society and communities alike, and by working to improve understanding and develop new technologies to push the boundaries of our capability, we can effectively provide those benefits.

7.2 Project Specific Professional Responsibility Areas

Work Competence does apply within our project's professional context. It is always necessary to have a level of understanding about the project in order to be able to complete it to a satisfactory degree. In this regard, our group is performing at a Medium level in this area. We have a good understanding of code and hardware, but none of us have prior experience working with RC cars. However, we have access to many resources, such as the F1/10th website, that can help us to bridge these gaps in our knowledge.

Financial Responsibility plays a small role in our project's scope. Our final product will need to stay within the budget provided to us by the school, but it is a one-off product. We will not need to consider any optimizations for future production or bulk production, just our immediate costs. As such, our group is currently performing at a High level- we remain well within our budget at this time and do not anticipate any major uptick in spending.

Communication Honesty will have a notable application in our project's context. Already, we have been providing biweekly update reports on our progress with the design of the car. We will have to continue these reports, as well as presentations on our design and prototype, as the class continues. Honesty in these presentations is essential- representing our work as being more advanced or complete than it really is will lead to more issues in the future when we cannot deliver on these promises. In this area, however, our group already performs at a High level. We represent all our progress accurately.

Heath, Safety, and Well-Being will not play a role in this project. While the project is titled the "Race of Doom", the only thing at risk in the project is our vehicle. So long as we take basic safety precautions for any spectators, there should be no immediate risk, and there will be no long-term risks.

Property Ownership will also have only a minor application in our project's scope. The project has two teams designing cars to race against each other in the final presentation. The two teams need to develop their vehicles independently, so any attempt by us to make use of the other team's work, especially without acknowledging it, would be breaking the rules set out for the project. Thus far, our team's development has occurred completely independently, meaning we are performing at a High level.

Sustainability will play no role in this project. Our car is being designed for a one-off race that is intended to be potentially destructive to our vehicle. We will be creating only one prototype for a single use, and do not need to consider any future applications.

Finally, Social Responsibility will not apply within our project's professional context. Once again, this is a small, destructive race intended for entertainment. While there could be information taken from this project and applied to improve society, such as by improving the durability of autonomous rescue vehicles, any such applications are not the main intention of the project.

7.3 Most Applicable Professional Responsibility Area

The most applicable Professional Responsibility area to our project is Communication Honesty. Since this project ends in a competition, there is a strong temptation to break the rules of the project, perhaps by taking information we are not supposed to know from the other car team or even the track team to give ourselves an unfair advantage. Holding ourselves accountable and using only the information we are supposed to have access to fulfills one important part of Communication Honesty.

Another aspect of our project that will touch on Communication Honesty is reporting the sources we do end up using, be it the F1/10th website or an online code resource like GitHub. Once again, not reporting where our information comes from risks charges of plagiarism, but we can also benefit from recording what sources we use to track down any misinformation, or information that isn't quite what we needed at the time.

8 Closing Material

8.1 Discussion

Discuss the main results of your project – for a product discuss if the requirements are met, for experiments oriented project – what are the results of the experiment, if you were validating a hypothesis – did it work?

Requirements:

- The RC car can detect and avoid obstacles using lidar and photoelectric sensors
- The car can make decisions about how to move based on sensor data
- The car can be autonomously controlled by a raspberry pi
- The car can detect cyber attacks and effectively defend against them

Hypothesis:

Our team is able to design and construct an autonomous RC car capable of navigating a challenging track,evading obstacles, and equipped with cybersecurity measures to defend against potential cyber attacks. By using sensor technology, including lidar and photoelectric sensors, our car will seamlessly perceive its surroundings and make informed decisions in real time. A raspberry pi will serve as the cars central processing unit, tasked with interpreting sensor data and issuing commands to the cars motor. To safeguard against cyber attacks, we will implement a multi-layered security approach, including encryption techniques, intrusion detection systems, and secure communication protocols.

8.2 Conclusion

Goals:

- Design and construct an autonomous RC car capable of navigating a challenging track, evading obstacles, and equipped with cybersecurity measures to defend against potential cyber attacks.
- Use sensor technology, including lidar and photoelectric sensors, to seamlessly perceive the car's surroundings and make informed decisions in real time.
- Utilize a Raspberry Pi as the car's central processing unit to interpret sensor data and issue commands to the car's motor.
- Implement a multi-layered security approach, including encryption techniques, intrusion detection systems, and secure communication protocols, to safeguard against cyber attacks.

The best plan of action to achieve these goals is to continue to develop and test the car's hardware and software, finalize the car's design and documentation, and prepare for the final competition.

Constraints:

- The limited amount of time, resources, and budget
- The complexity of the project
- The teams lack of experience with certain technologies pertaining to the project

For Future design/implementation:

- Familiarize everyone on the team with what exactly has been tested and what components on each level (hardware, embedded systems, software) are capable of doing.
- Continue to communicate with the team on changes on vision, issues, or concerns.

The team is confident that they can achieve their goals by following the plan of action outlined above. The team is excited about the potential of the Race of Doom project to make a real difference in the world by helping to develop autonomous vehicles that are safe and secure.

8.3 References

LiDAR Sensor Datasheet:

[1] Rplidar A1M8-R6 - 360 Degree Laser Scanner Development kit, https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/828/DFR0315_Web.pdf (accessed Dec. 4, 2023).

Photoelectric Sensor Datasheet:

[1] Gravity: Digital adjustable infrared proximity for (0~200 cm), https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/2225/SEN0239_Web.pdf (accessed Dec. 4, 2023).

Raspberry Pi Datasheet:

[1] Raspberry pi 4 computer, https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-product-brief.pdf (accessed Dec. 4, 2023).

F1Tenth website:
[1] "F1tenth," F1TENTH, https://f1tenth.org/ (accessed Dec. 3, 2023).

8.4 Appendices

LiDAR Sensor Datasheet:
https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/828/DFR0315_Web.pdf

Photoelectric Sensor Datasheet:
https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/2225/SEN0239_Web.pdf

Raspberry Pi Datasheet:
https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-product-brief.pdf

F1tenth website:
https://f1tenth.org/

8.4.1 Team Contract

Team Name: Race of Doom - Cyber Security

Team Members:
1) Andrew Kraft 2) Gavin Petrak
3) Jack Doe 4) Jacob Nedder
5) Peter Wissman

Team Procedures

1. Day, time, and location (face-to-face or virtual) for regular team meetings:
   Wednesdays at 2 PM in Coover 1301 (Senior Design Lab).

2. Preferred method of communication updates, reminders, issues, and scheduling (e.g., e-mail, phone, app, face-to-face):
   Weekly meetings face-to-face, Discord for remote communication, Gitlab for file sharing.

3. Decision-making policy (e.g., consensus, majority vote):
   All members will attempt to reach a consensus until a majority vote is necessary.

4. Procedures for record keeping (i.e., who will keep meeting minutes, how will minutes be shared/archived):
   A Google Doc will be shared for each meeting to allow each member to enter their notes.

Participation Expectations

1. Expected individual attendance, punctuality, and participation at all team meetings:
   All members are expected to be at all meetings unless all members are aware of a conflict. Showing up on time and being ready to communicate with one another is a requirement for a meaningful meeting.

2. Expected level of responsibility for fulfilling team assignments, timelines, and deadlines:
   Every member should work throughout the week, splitting up their work to consistently meet deadlines and provide work for the team. If an issue arises and the timeline must be extended, that member should be open about their issues with the entire team to properly adjust that expectation.

3. Expected level of communication with other team members:
   Members should check their email and/or Discord at least once a day to see if any assistance or action is necessary.

4. Expected level of commitment to team decisions and tasks:
   When deciding on tasks and issues as a team, each member should have input as to what the deadline and quality of work should be for that item. Expectations should be stated when a new task arises.

Leadership

1. Leadership roles for each team member (e.g., team organization, client interaction, individual component design, testing, etc.):
   Andrew Kraft - Testing, Circuit Design
   Gavin Petrak - Team Organization
   Jack Doe - Project Manager
   Jacob Nedder - Testing, Team Coordination
   Peter Wissman - Client Interaction

2. Strategies for supporting and guiding the work of all team members:
   In weekly meetings, each member will have time to voice what they are working on, if any impediments are present, and give their game plan for the work they will be doing the following week. Discord will allow an instant method of reaching out if there are any concerns.

3. Strategies for recognizing the contributions of all team members:
   Issues will be assigned to the member who is working on that part and successes will be shared during weekly meetings to recognize those who have made progress.

Collaboration and Inclusion

1. Describe the skills, expertise, and unique perspectives each team member brings to the team.
   Andrew Kraft - Power Systems, Circuit Design, Electrical and Physical Schematics
   Gavin Petrak - Java, C, Embedded Systems Programming
   Jack Doe - Java, C, HTML, CSS, Javascript, Python, SQL, etc
   Jacob Nedder - Python, Information and Software security
   Peter Wissman - Java, C, Embedded Systems

2. Strategies for encouraging and support contributions and ideas from all team members:

   Communication and being open to new ideas is integral for a good team. As a team, we will speak up and ask questions about the project as it sparks discussion and will create a better final product. We will also be looking to go outside our comfort zone and learn more about the different skills we aren't the best at.

3. Procedures for identifying and resolving collaboration or inclusion issues (e.g., how will a team member inform the team that the team environment is obstructing their opportunity or ability to contribute?)

   Issues may be brought up either during weekly team meetings or through the Discord group chat depending on both the severity and the time sensitivity of the issue. If an issue of inclusion arises, the whole team will engage to ensure proper adjustments to workload and assigned tasks are complete.

Goal-Setting, Planning, and Execution

1. Team goals for this semester: Present a working final product for senior design
2. Strategies for planning and assigning individual and team work: weekly meetings and checkups
3. Strategies for keeping on task: Holding eachother accountable and weekly meetings

Consequences for Not Adhering to Team Contract

1. How will you handle infractions of any of the obligations of this team contract?

Initial infractions will result in warnings from the team. Failure to complete work within the allotted deadlines will result in another team member being brought into the task to help ensure the work is brought up to standard and to assist in reducing the delay to project completion.

2. What will your team do if the infractions continue?

Once three infractions on the same issue have occurred, the team will bring up the problems at the weekly TA meeting to reach a consensus as a group.

****

a) *I participated in formulating the standards, roles, and procedures as stated in this contract.*

b) *I understand that I am obligated to abide by these terms and conditions.*
c) *I understand that if I do not abide by these terms and conditions, I will suffer the consequences as stated in this contract.*

1) _________Andrew Kraft_____________________________ DATE ____9/10/2023______
2) __________Jacob Nedder_____________________________ DATE ____9/10/2023______
3) ___________Jack Doe ___________________________ DATE ____9/10/23________
4) __________Peter Wissman___________________________ DATE:_____ 9/10/23___________
5) __________Gavin Petrak____________________________ DATE ____9/10/23________