J. MIKE WALKER '66 DEPARTMENT OF MECHANICAL ENGINEERING
TEXAS A&M UNIVERSITY

AUTONOMOUS COMBAT VEHICLE: FINAL REPORT

John Lyons, Stuart Day, Tejas Murali, Abdulelah Alamri

Texas A&M University

12/08/2021

Fall MEEN 402-503

Dr. Dale Cope

EXECUTIVE SUMMARY

Game theory mathematically predicts and models decision-making in complex environments, which is ideal for analyzing military situations. During World War II, U.S. mathematician John Von Neumann and other economists created the study of game theory. However, due to computational limitations, the U.S. military ignored game theory applications for the battlefield. Recently though computer and software advancements have made the study of game theory practical; therefore, the study of game theory has taken a renewed interest. This project's mission is to further the advancement of the study of game theory and its applications. The Autonomous Combat Vehicle (ACV) team has developed and integrated Donkey cars with manual control to play one versus one Capture The Flag (CTF) game. Additionally, the Donkey cars have a plug-in for Artificial Intelligence (AI) control in order to showcase to the U.S. Army the importance of game theory in complex strategic decision-making situations.

After meeting with the primary sponsor and the director of the Mays Innovation Research Center (MIRC), Dr. Korok Ray, the team received important information about the problem and needs of the project. Yan Yao, a recent mechanical engineering graduate, initially built the Donkey cars and started the ACV research project under Dr. Ray. Mr. Yao is now the team's technical advisor, and he assists the team with coding and integrating the software of the Donkey cars. Dr. Guni Sharon is a stakeholder in this project and also the leader of the Computer Science (CS) team overseeing the development of an AI program that plays CTF autonomously. After the ACV team completes the project and installs the AI hardware, the CS team will install software capable of performing AI CTF simulations into the Donkey cars.

In the final stage, the ACV teams structured the project into four subfunctions: physical car adjustments for AI hardware, full manual control with a gaming controller, shooting mechanism with IR blaster and receiver, and CTF mechanism using a camera. The physical car adjustment is responsible for redesigning the chassis of the Donkey cars to stabilize the camera, Infrared (IR) blaster and receiver, and adding receiver diodes for 360 degrees receiving angles. The team redesigned the chassis of the Donkey cars to fit a NVIDIA Jetson Nano, which satisfies the stakeholder's need for AI capable hardware. The ACV team final project also includes full manual control of the Donkey cars using Logitech joystick controllers. With joystick controllers, operators can adjust different speeds and directions of the Donkey cars. The team has also completed the firing mechanism for the Donkey cars using an IR blaster operated via a joystick controller. When the Donkey car gets shot by the enemy's IR blaster, the vehicle goes offline for approximately five seconds then it will come back online. For the CTF mechanism, the team chose to use Open Source Computer Vision Library, known as OpenCV, to carry out a color detection process to identify yellow and green flags virtually. The team created a CTF game logic, where the red Donkey car's mission is to scan the yellow flag first then scan the green flag to score a point. The blue Donkey car's mission is to scan the green flag first then scan the yellow flag to score a point. Additionally, the team created scoreboards for each of the Donkey cars, where operators can check the current score for the Donkey cars.

The Autonomous Combat Vehicle team has successfully met all the customer needs of the project. The ACV is manually controlled with a joystick controller, programmed to shoot the IR blaster with the controller's trigger buttons, and plays the CTF game by identifying the yellow and green colored flags with software developed by the group. Future work for the project include fixing minor adjustments on the electrical receiver, which include increasing the count of receiver diodes. The adjustment will increase the stability of how the CTF game operates. Additionally, the team is considering making an instruction guide for the CS team, who is responsible for integrating the AI CTF simulation into the Donkey cars.

Table of Contents

Glossary Table

Vocabulary	Definition
ACV	Autonomous Combat Vehicle
AI	Artificial Intelligence
CS	Computer Science
CTF	Capture The Flag
Donkey Car	Remote Control Car + Raspberry Pi
FMECA	Failure Mode Effects and Criticality Analysis
FTA	Fault Tree Analysis
GPU	Graphic Processing Unit
IR	Infrared
JetRacer	An autonomous AI racing car powered by NVIDIA Jetson Nano
MEEN	Mechanical Engineering
MIRC	Mays Innovation Research Center
NGCV	Next-Generation Combat Vehicle
OpenCV	Open Source Computer Vision
PLA	Polylactic Acid
RAM	Random Access Memory
RC	Radio-controlled
ROS	Robot Operating System
OpenCV	Open Source Computer Vision
SwIRL	South west Innovation Research Lab

1. Introduction and Problem Definition

One of the priorities of the Army is to work towards modernization which includes creating a next-generation combat vehicle that would have the capability of autonomous control with Artificial Intelligence (AI) machine learning [1]. Because of the military advantage when using autonomous capability, some areas of the military are quickly adopting to use this new technology. The crux of autonomy is when multiple autonomous characters are required to interact with each other, researchers and the military do not fully understand the outcomes of such scenarios. The team is currently developing a situation where game theory, the study of autonomous characters interacting with each other, can be demonstrated to the US Army.

Autonomous Combat Vehicles (ACV) are Donkey cars that operate on Raspberry Pi's, equipped with a shooting mechanism using Infrared (IR) blasters and receivers, and interact using a central computer. The main controlling software utilizes a program called Robot Operating System (ROS) and also the Linux operating system. The IR blasters and receivers are programmed using Arduino with C++ and operate similar to how a television remote does. The current Donkey cars allow manual control, but the project aims to create a setting for cars to develop partial and eventually complete autonomy. This scenario of creating an arena with Donkey cars playing Capture the Flag (CTF) will act as a demonstration so the researchers can show human-machine interactions for military situations. The focus of the Autonomous Combat Vehicles (ACV) team is to develop and integrate Donkey cars to play CTF and allow AI capability to be "plugged in" to the vehicles. Encoding the IR blasters and receivers, as well as keeping scores are both steps in transitioning into full autonomy.

Four personnel who are connected to Texas A&M's Engineering and Business Schools are sponsoring the ACV project. The primary sponsor is Dr. Korok Ray, an associate professor at the Mays Business School of Texas A&M University and is also the director of the Mays Innovation Research Center (MIRC). Another sponsor is Dr. Sivakumar Rathinam, an associate professor at the Mechanical Engineering Department of Texas A&M University who specializes in motion planning and control of autonomous vehicles. Yan Yao, a recent mechanical engineering graduate, previously built the Donkey cars and started the ACV research project under Dr. Ray is the ACV's technical advisor. Mr. Yao has assisted the team with the software integration of the Donkey cars. Dr. Guni Sharon is an assistant professor at the Computer Science (CS) Department of Texas A&M University and is a stakeholder. He is the CS department team lead for AI CTF simulations and oversees an AI program of CTF. After the ACV team finishes the project, the CS team will be adding AI CTF simulations and software code into the Donkey cars.

The ACV team has successfully satisfied all the customer needs and the mission statement from MEEN 401 of "Design a scenario for RC cars to play a one versus one CTF game where the Donkey cars have manual control and a 'plug-in' for AI capability."

1.1 Need Analysis

The needs of the project are broken up into four main items as seen in Table 1.1.1 and general descriptions for each of the listed needs are then given below.

Table 1.1.1: MEEN 402: Customer's Needs for the ACV Team

	Customer Needs
1	The arena is the South west Innovation Research Lab (SwIRL).
2	The Donkey cars operate with full manual control with joystick controllers.
3	The Donkey cars have the capability of a plug-in (or hardware) for Artificial Intelligence (AI) control.
4	The Capture The Flag (CTF) game must be functional with a firing and CTF mechanism.

1. Arena is the South west Innovation Research Lab (SwIRL)

From the customer needs, the ACV team utilized the entire lab space, South west Innovation Research Lab (SwIRL), as the physical arena. Figure 1.1.1 shows the SwIRL space with the work tables.

Figure 1.1.1 South west Innovation Research Lab

The team decided to utilize the work tables in the SwIRL as obstacles for the CTF game. Figure 1.1.2 shows the obstacle work tables for the CTF game. There are eight work tables in the SwIRL and they are easy to move if necessary.

Figure 1.1.2 Obstacle Work Tables in the SwIRL

Figure 1.1.3 shows a floor map of the SwIRL showing a layout of obstacle workbenches, the positions of the flags, and the command station which holds the laptops displaying the scoreboards.

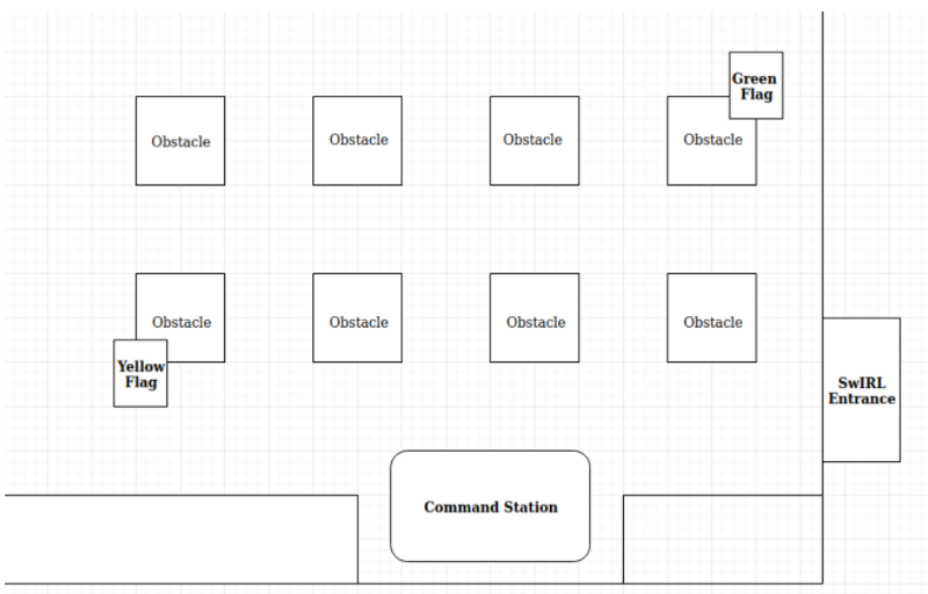

Figure 1.1.3 Map of SwIRL and Flag Locations

2. The Donkey Cars Operate with Full Manual Control

Control of the Donkey cars is a key aspect of this project, and the control can be implemented in multiple ways for an operator. However, to apply intuitive manual control for an operator, the options for manual control are narrowed. Imitating controller button key layouts for racing games is desirable to obtain an operator's intuitive control of a Donkey car. Figure 1.1.3 shows the button layout for the controller that is used to control the Donkey cars. RB button acts as ignition but must be pressed continuously during the movement and shooting of the Donkey cars.

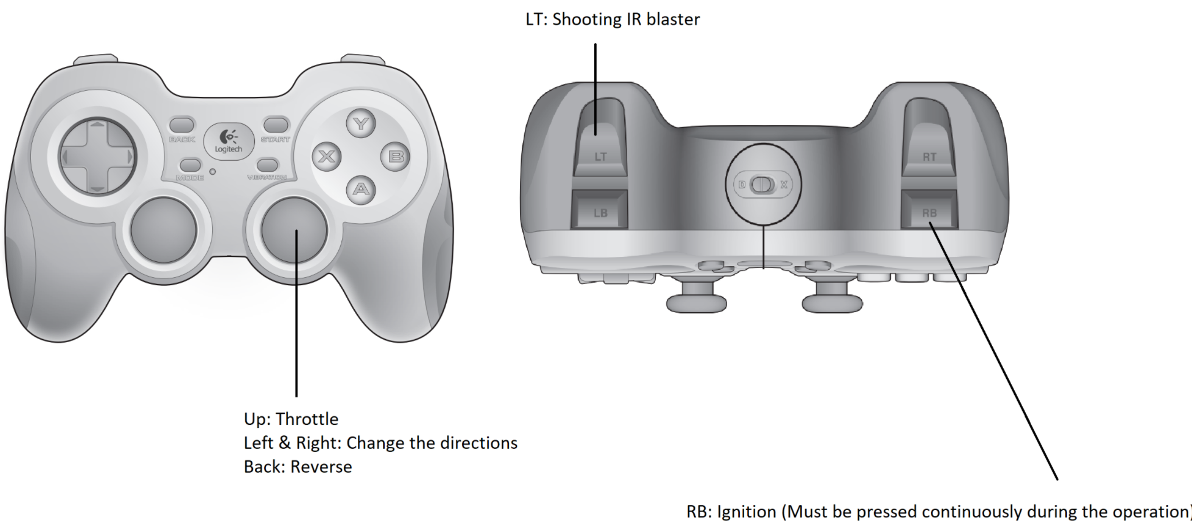

Figure 1.1.3 Donkey Car Logitech Controller Button Layout

3. The Donkey Cars Have the Capability of a Plug-In for Artificial Intelligence (AI) Control

A differentiating factor for this project is the need for the Donkey cars to be able to be controlled by artificial intelligence (AI). After consulting with the Computer Science (CS) team, who is responsible for developing an AI, the team found out the specific requirements that are needed for running AI onboard the car. That AI hardware requirement is to have a Jetson Nano GPU onboard the donkey car to process the algorithms that are needed to play the game autonomously. The Jetson Nano is seen in Figure 1.1.4 and needs to be located onboard the Donkey car.

Figure 1.1.4 Jetson Nano Required for Autonomous Capability

4. Capture the Flag (CTF) Game Must Be Functional

The game mode that was suggested by the sponsor's problem statement was the game of Capture the Flag (CTF). A simple explanation of the Capture the Flag game mode is that a flag is placed in each team's base and the goal of each team is to capture the enemy's flag and bring it back to your base. The way Capture the Flag will take place with the Donkey cars is that the Donkey cars will pick up and drop a digital flag. The way the digital flag is picked up or captured is by a Donkey car camera scanning the "pick up" signal or the "drop off" signal. If a Donkey car is transporting a flag and becomes shot by an enemy then it will cause the flag to "drop" and the car will lose possession of the flag. A Donkey car that gets shot by an enemy, in addition to dropping a picked up flag, will also be penalized by freezing it's motor functions for a period of time.

1.2 Design Parameters and Performance Requirements

The section, 1.1 Need Analysis Sections 1-4, are brief descriptions of the overall task descriptions for this project. Section 1.2 Design Parameters and Performance Requirements goes into the detailed requirements of the project.

By using the customers' needs, the performance requirements and design parameters were created and are listed below in Table 1.2.1. These design parameters and performance requirements are divided into four parts, as outlined in Section 1.1 Needs Analysis.

Table 1.2.1: Performance Requirements

Customer Needs:	Performance Requirements:
Arena is the South west Innovation Research Lab (SwIRL):	1. The obstacles must not move when a Donkey car at full speed runs into it. 2. The layout of the arena must allow for a 4ft turn diameter . 3. The height of the obstacles is between 1 ft and 5 ft.
The Donkey cars operate with full manual control:	1. The car's controls must operate the car with variable speeds. 2. Controller's shooter button will be assigned to a trigger or bumper button of the controller. 3. Using manual control the car must reach max speed and have the ability to turn.
The Donkey cars have the capability of a plug-in for Artificial Intelligence (AI) control:	1. The chassis must contain the 3.9 in x 3.1 in x 1.1 in (Jetson Nano dimensions). 2. The chassis must hold 8.8 ounces of weight. Movement control, shooting, receiving, and picking up the flag must be done digitally.
Capture The Flag (CTF) game must be functional:	1. The Infrared (IR) receiving range on the Donkey car must have a circular 360 degree range. 2. The maximum range of picking up and dropping the flag is 3 ft. 3. A shot Donkey car must not move for 5 seconds.

1.3 Function Structure

Functional modeling is the standardized process of representing different functions within a product in an understandable and communicable way. Functional modeling is used "to describe and compare product functionality and create a formal function representation that would advance design methods and lead to repeatable models" [2].
We divided the design into four main sub-functions. The sub-functions are: manual control, IR shooting and receiving, Capture the Flag mechanism, and physical car's chassis adjustments. Figure 1.3.1 shows the black boxes of the first sub-function: manual control.

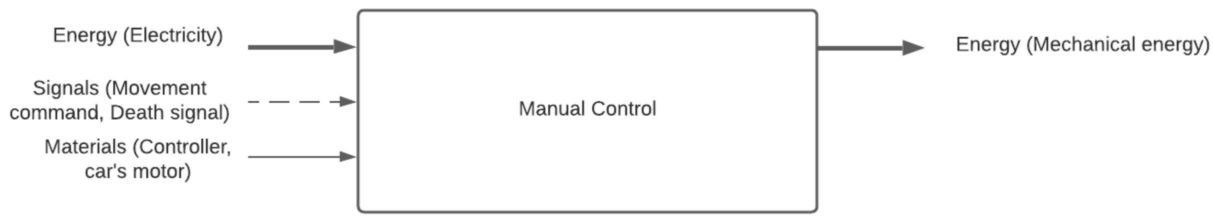

Figure 1.3.1: Black Box Model of the Manual Control

The input energy is the electricity stored in the battery of the Donkey car. The inputs signals are the movement command received from the controller and the death signal received from the IR Blaster. The input materials are the controller and the motors of the car. The controller and the motors are necessary devices for the Manual Control sub-function. The main output of the manual control sub-function is the mechanical energy observed as the movement of the car. This sub-function has no relevant output signals or materials. Figure 1.3.2 goes into more details of the manual control's functional block diagram. The figure shows the functions and how they interact with each other.

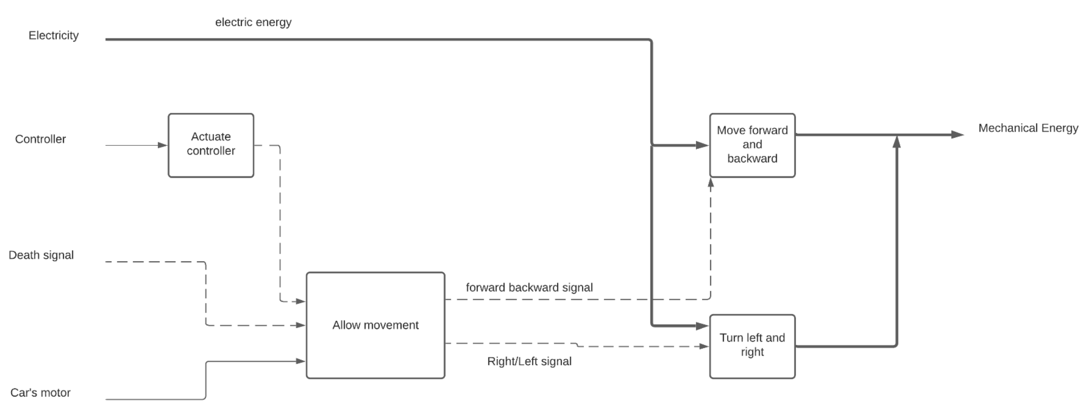

Figure 1.3.2: Functional Block Diagram of the Manual Control Sub-System.

Figure 1.3.3 and 1.3.4 are the black box and functional block diagram, respectively, of the IR shooting and receiving sub-systems. The enemy's shots are IR signals sent by the other Donkey car. The shots, in the output signals, are the IR signals that the car itself sends. The IR Blaster is used to detect enemy shots and shoot when commanded.

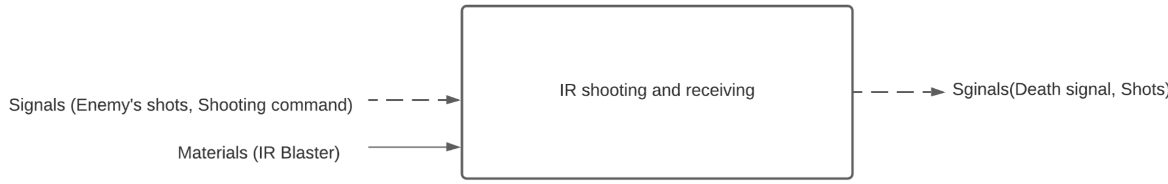

Figure 1.3.3: Black Box of the IR Shooting and Receiving Sub-System.

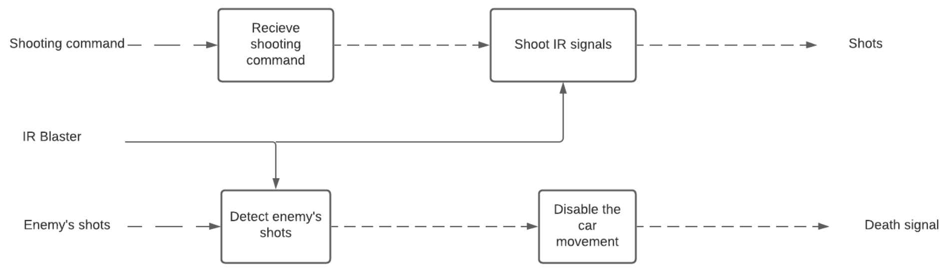

Figure 1.3.4: Functional Block Diagram of the IR Shooting and Receiving Sub-System.

The CTF Mechanism using a camera black box is shown in Figure 1.3.5. The input signals are the death signals received from the IR Blaster and the live video stream is captured by the camera. The materials are the camera to record the live video and a laptop screen to show the score. The energy input and output were not relevant to this function. This subfunction also has no output materials. As a result, we omitted the non-relevant energies and materials.

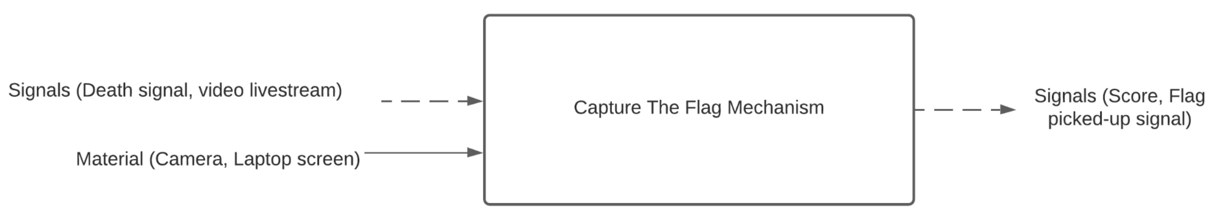

Figure 1.3.5: Black Box Model of the CTF Mechanism.

The detailed functional block diagram for CTF is shown in Figure 1.3.6. When the flag is picked up, a signal should notify the players that the car is carrying the flag, this signal is identified as the Flag picked-up signal in the figure.

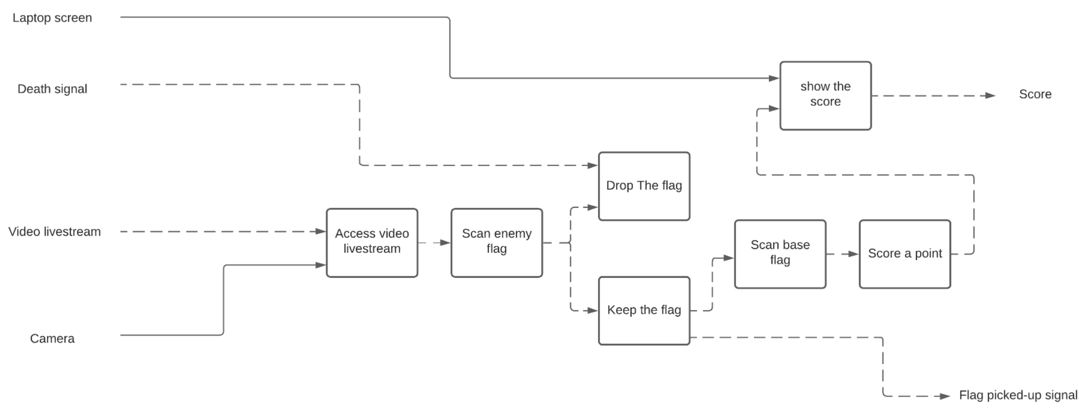

Figure 1.3.6: Functional Block Diagram of the Capture the Flag Sub-System.

The last subsystem is the car’s chassis adjustments. The car chassis main function is to hold the different components together securely while allowing communication between these components. Figure 1.3.7 shows the blax box model, while Figure 1.3.8 shows the detailed functional block diagram. The only inputs are materials to be secured and the output is the car chassis.

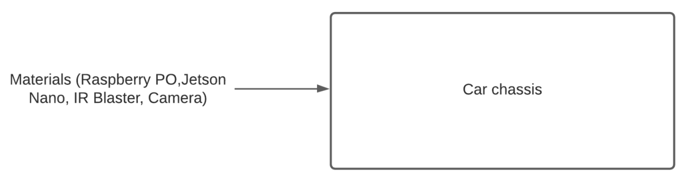

Figure 1.3.7: Black Box of the Car Chassis

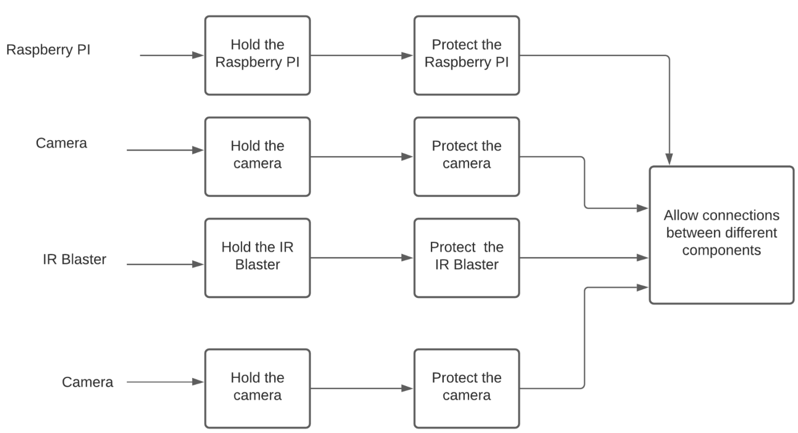

Figure 1.3.8: Functional Block Diagram of the Car’s Chassis

2. System Description

Connecting to the Donkey car was the first step towards gaining manual control. Two options are available for manual control, the Pi-Fi connection or the Donkey car's access point. The Pi-Fi connection allows multiple devices to connect to the same network and communicate with each other. Using the Donkey car's access point means the different Dokey cars cannot communicate with each other. Each car has to be connected to a different laptop to control it. The team chose to use the Donkey car's access point to control it because of the simplicity of the connection. The team has no background in networking and Pi-Fi connection. The car is connected to a laptop through its own access point. The laptop is used to run commands necessary to operate the car. The next step is to use programming packages to drive the motors of the car. The following packages were used to drive the motor: i2cpwm_board and Donkey_llc. For more information about the packages, please refer to the Appendix C. Teleopjoy is a node that is used to convert the joystick controller signals to vehicle movement commands. The final layout of the nodes in ROS are shown in Figure 2.1. To run the car, use the numbered commands in Figure 2.1.

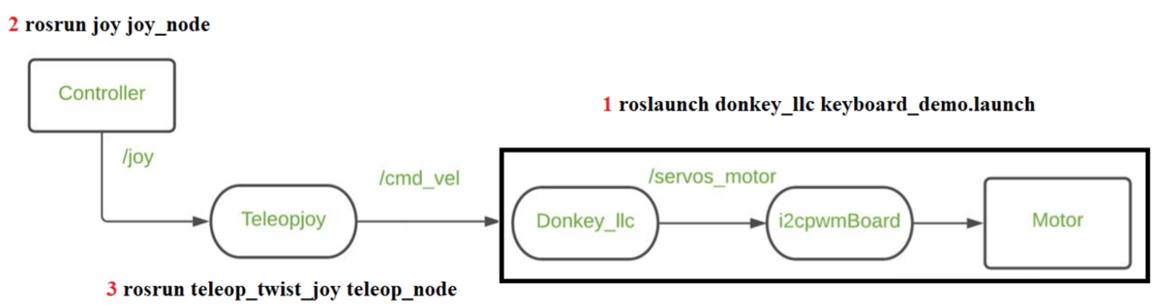

Figure 2.1: ROS Environment Design for Manual Control.

The car shoots using a Sparkfun IR Blaster. The IR Blaster can shoot Infrared signals using an IR emitter forward in a narrow cone shape path. Figure 2.2 shows the approximate range of shooting, in red, and receiving, in green.

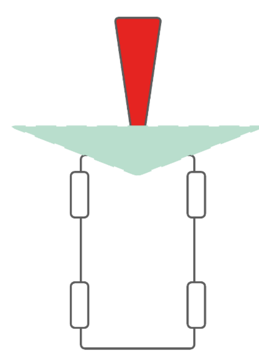

Figure 2.2: Overhead View of the Donkey Car and Its Shooting and Receiving Range. (Top Down View)

The team decided to increase the receiving range to 360 degrees. Figure 2.3 shows the expected new receiving range in green of the car. We will install three more IR receivers at the sides and the back, which will result in 360 degrees of receiving range.

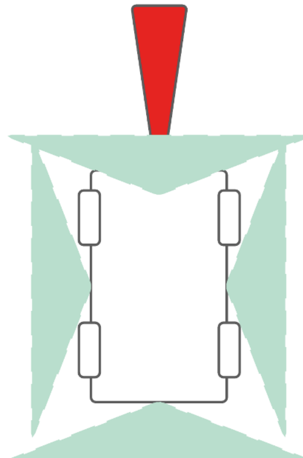

Figure 2.3: Overhead View of the Donkey Car with the Installed Receivers (Top Down View)

Figure 2.4 summarizes the code layout of the receiver. The code builds on the already existing one for the manual control. The shooter node will process any shots received and will send a command to disable the car movement for 10 seconds by disabling the Donkey_llc node using the /death_topic. The shooter node will receive signals from the controller using /joy topic. Then, it will command the IR Blaster to shoot through serial communication using the USB port connecting the IR Blaster and the Raspberry PI. The shooter node is an executable code written in Python.

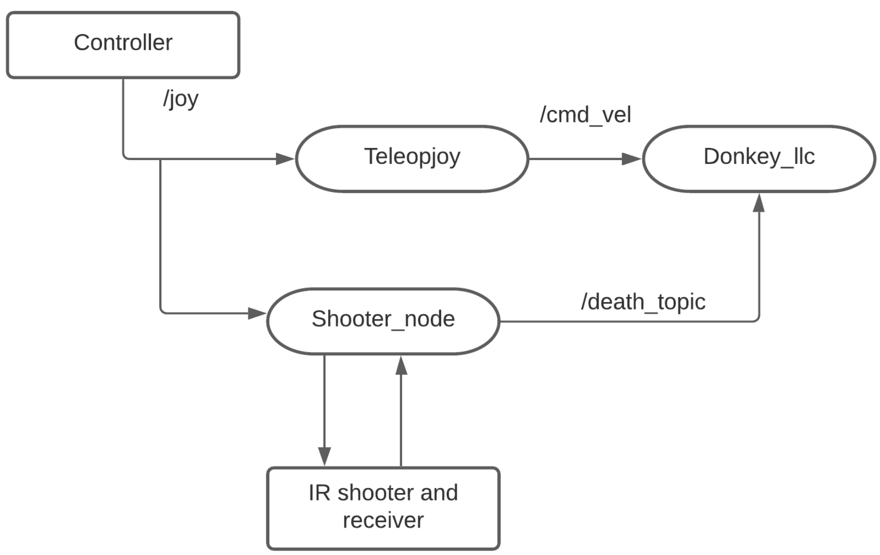

Figure 2.4: ROS Environment for the IR Shooter and Receiver.

To score a point in the CTF game, a Donkey car must obtain the enemy flag first. Then, while still alive, it should go back to its home base. Lastly, it must obtain its own flag. We designed the game to be completely digital. Thus, using a physical flag was not an option. The first iteration of the flag design was a QR code. A Donkey must scan a QR code to obtain the flag. However, most of the open source packages for the QR code do not function on a wide -angle camera, which is the one installed on the car. Also, the QR code needs high resolution video to be scanned. Although high resolution video can be captured using the existing camera, it requires high processing power which can affect the performance of the Raspberry PI. Thus, the team decided to change the flag to a colored object. Figure 2.5 illustrates the ROS environment required to play CTF.

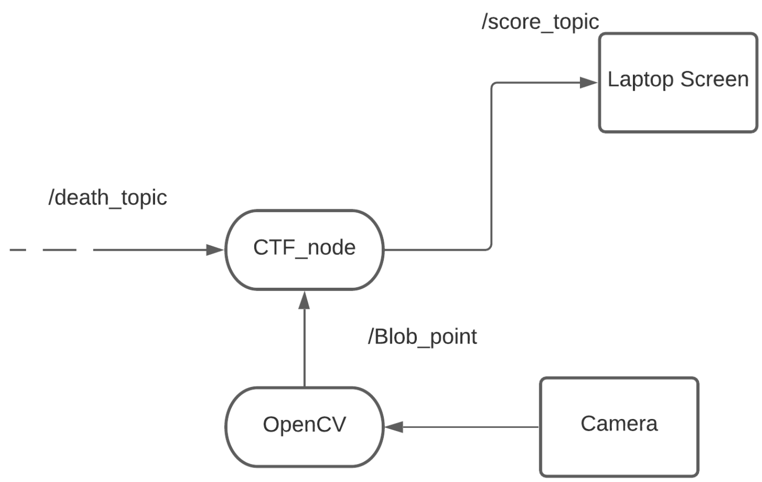

Figure 2.5: ROS Environment for CTF Mechanism.

OpenCV (Open Source Computer Vision Library) "is an open source computer vision and machine learning software library" [3]. We will utilize this library to detect the different colors for the flags. It can be integrated into ROS as a node as shown in Figure 2.5. The /death_topic is imported from the shooter node seen in Figure 2.4. The /death_topic is used to drop the flag whenever the Donkey car is shot. The OpenCV node's function is to detect whenever the enemy or its own flag is detected. The CTF node is responsible for counting the scores and updating it. The logic of the game should be processed in this node. Lastly, the score is shown on the laptop's screen connected to the donkey car. Each team will have its own differently colored object as a flag. The two colored objects are green and yellow flags as seen in Figure 2.6. These colors were chosen because they stand out from the surrounding environment and the Donkey cars.

Figure 2.6: The two flags used in the game.

.

The overall ROS environment is summarized in Figure 2.7 and is seen below..

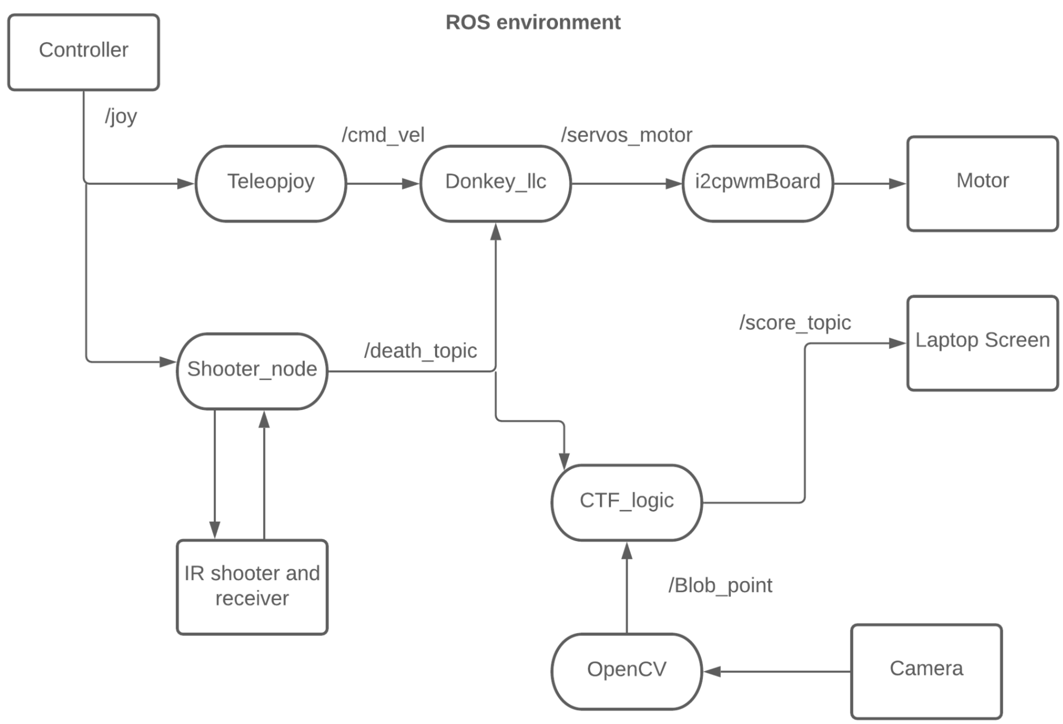

Figure 2.7: The Whole ROS Environment.

The chassis was modified to allow attaching the Jetson Nano. The new part was 3D printed using polylactic acid (PLA) material. PLA has high strength and hardness compared to the other options available such as ABS. The new chassis has modular design where new components can be attached easily. Figure 2.8 shows the dimensions of the chassis design in empirical units (in). The chassis attaches to the car using three screws connecting its legs into the car. And there are four mounts to secure the Jetson Nano to the top chassis.

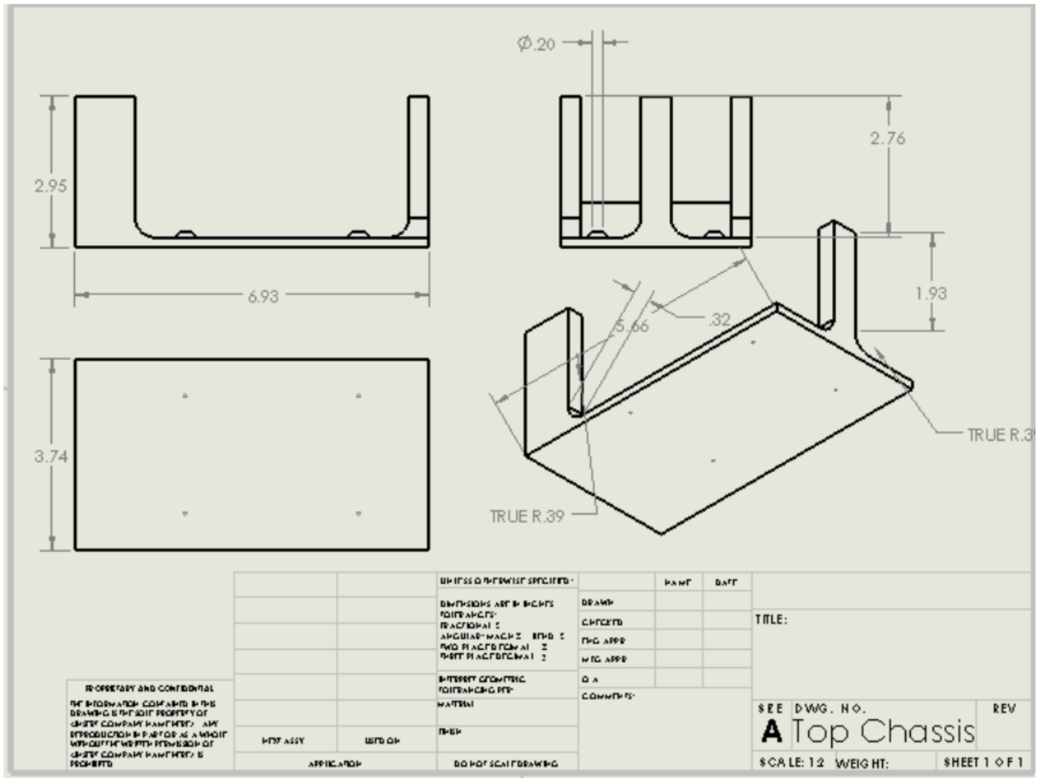

Figure 2.8: Engineering Drawing of Top Chassis (in.)

In Figure 2.9, the IR Blaster and Receiver is shown. The engineering drawings for the device are also provided. The material is also made of PLA for this holder as the mechanical properties are more accurate when printed compared to ABS.

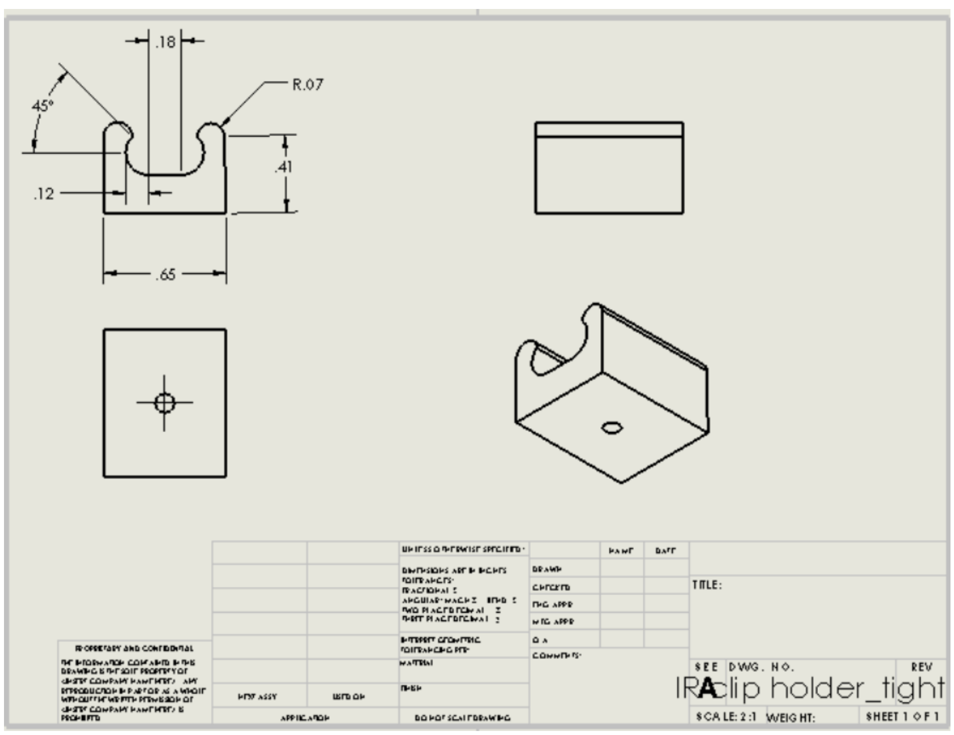

Figure 2.9: Engineering Drawing of IR Blaster Holder (in.)

The camera holder is used to attach the camera to the chassis. Its dimensions are shown in Figure 2.10. The camera holder was also 3D printed using PLA. The camera holder uses a sliding mechanism to hold the camera.

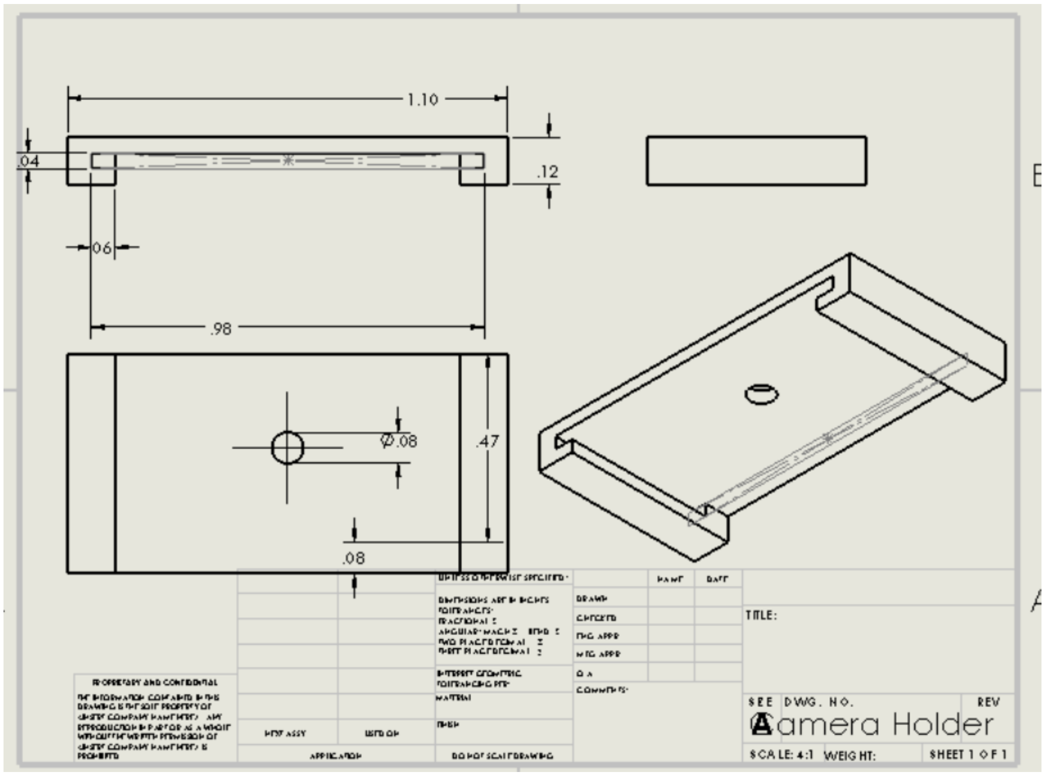

Figure 2.10: Engineering Drawing of Camera Holder (in.)

Figure 2.11 shows how the IR Blaster is clipped into the IR Blaster holder. Then the holder is screwed into the top of the chassis. The camera is inserted into its holder and the holder is screwed to the front panel of the chassis.

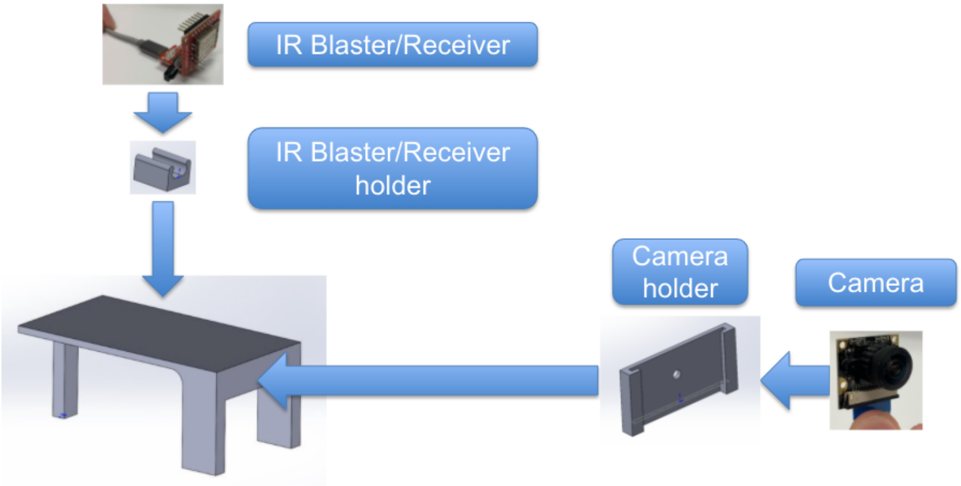

Figure 2.11: Assembly Process for the Camera and the IR Blaster to the Chassis

Figure 2.12 shows how the Jetson Nano is attached to the chassis. Four screws hold the Jetson Nano upside down to the bottom of the chassis. This allows the Jetson Nano to be securely placed into the donkey car.

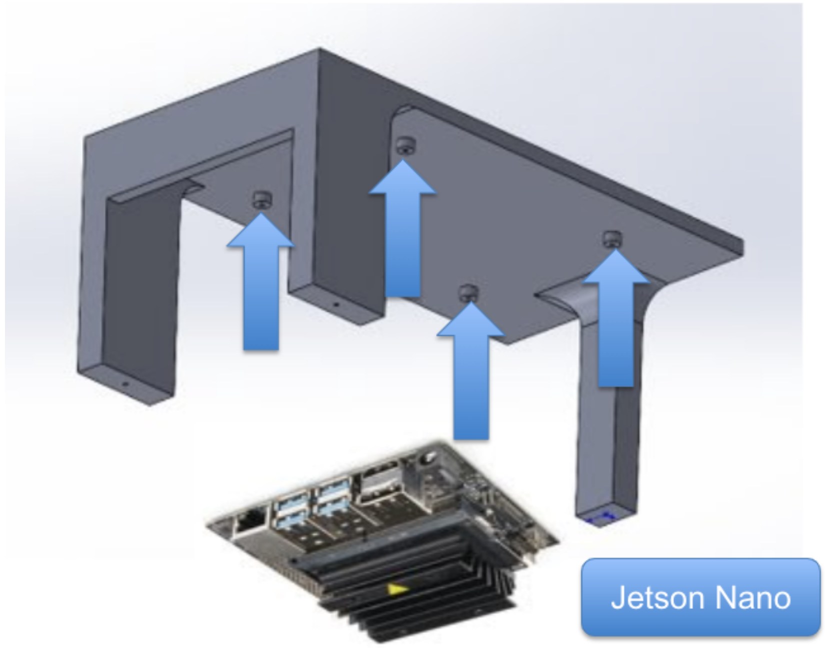

Figure 2.12: Assembly of the Jetson Nano to the Chassis

3. Analysis for Design

The analysis for design was completed using multiple processes such as the embodiment design checklist, Failure Mode, Effects, & Criticality Analysis (FMECA), and the Fault Tree Analysis (FTA).

3.1 Embodiment Design Checklist, FMECA, FTA

This section goes into detail about each of the specific processes used for designing and planning of the autonomous combat vehicle system. The embodiment design is also included in the appendix of the report.

3.1.1 Embodiment Design Checklist

In the Embodiment Design Checklist, Appendix A, all of the aspects of the project that affected the Embodiment Design were included. Some of these areas were more thoroughly addressed than others, because those areas were more relevant to the project. All of these aspects are discussed in detail in the Appendix A. The function section discussed the four customer needs of the SwIRL as the arena, manual control, compartment for AI control, and successful CTF mechanisms. The cars will be using their own access points instead of connecting to a local network. Concerning energy transfer and speed, the transfer of energy involves the batteries, motors, and actuators within the car. The cars use Logitech controllers that can control the acceleration and velocity of the car. No quantitative safety calculations were completed, but multiple safety concerns in the project are worth mentioning, including both the safety of the user and the car, such as wearing P.P.E. and keeping the car on top of a wooden block during testing. Tolerances are also discussed, such as having small tolerances for the IR holder, and the assembly of the car and IR Blaster are also mentioned. Operations, such as using the controller to operate and handle the car were discussed as well. Maintenance in troubleshooting and networking was also discussed, as these were two important recurring processes that the team had to go through to maintain the car's abilities for manual control. Finally, costs were definitely relevant to the project, as the team had ordered a camera module, diodes, and another starter kit. The Gantt chart for the entirety of the project was also made to show the teams progress and note which tasks were still needed to be completed by their respective deadlines.

3.1.2 FMECA

The team analyzed the risks for the full project using a Failure Mode, Effects, & Criticality Analysis (FMECA) [4] table is shown below in Table 3.1.2.1. The parts of the project, their potential failure modes, causes, effects, testing methods, and recommended actions are addressed. Also, numerical quantities such as the Severity (S), Occurrence (O), and Detection (D) were also mentioned to calculate the Risk Priority Number (RPN). The FMECA table was particularly helpful in the project, because it alerted the team to what parts of the project needed more focus. The team moved forward with color sensors and not QR scanners. The team also postponed completing the electrical diode upgrade until after the final demo. Scoreboards are present on each laptop since the members can connect to their cars. The LED lights were not used because they were not needed. Overall the risks for the project were present but the team tried their best to react to putting out brush fires and keeping on task.

Table 3.1.2.1 Failure Mode, Effects, & Criticality Analysis (FMECA) for the Project

Part # and Functions	Potential Failure Mode	Potential Effect(s) of Failure	S	Potential Causes and Mechanisms of Failure	O	Current Design Controls Test	D	Recommended Action	RPN
QR code and scanner	CTF cannot be played	Unable to meet the CTF need	8	Camera and scanner aren't compatible with car or stop working	6	Research if it has been done, scan a sample QR code prior	5	Switch to using color sensors for the CTF mechanisms for playing the game, find additional resources who could advise the team	240
Scoreboard	Unable to keep score	Unable to keep track electronically of which car wins each CTF game	5	One laptop cannot be used to connect all cars to a local network	3	Test to see if cars can shoot at each other and collect flags to score points	2	Use access points of cars instead of Pi-Fi connection so that one laptop corresponds to each car and each laptop can display if its car scored	30
LED Light	Unable to tell if a car is shot	Unable to tell when points are scored and	3	LED Lights are not compatible with the Donkey Car and IR Blaster and Receiver.	2	Test LED Light before-hand to see if it turns on when cars get shot	2	Use LED Light if it has been tested and works; it is low severity because it is not a must for	12

		unable to keep score						the game to be played	
IR Blaster and Receiver	Unable to shoot at other cars or receive shots	Cannot play the CTF Game	8	Arduino IDE code is not edited properly	2	Constantly test the IR Blaster and Receiver to see if it reacts	3	Keep continually testing the IR Blaster and Receiver to see if it can receive and shoot	24
Manual Control	Unable to manually control cars using controller	Cannot play the CTF game because the cars can't move	10	One of the car's battery is not fully charged, the controllers battery is not replaced, or the car internet doesn't show up	1	Make sure cars can be manually controlled every now and then using controller	1	Run the commands necessary to manually control the car at home in spare time	10
IR Blaster	Electrical Adjustment fails	CTF unable to be played	10	Diodes are not compatible working in parallel together	5	Test shooting while driving verify shot	4	Postpone electrical upgrade until after final demo	200

3.1.3 FTA

The team also created a Fault Tree Analysis (FTA) structure that was used for the project and is presented in Figure 3.1.3.1. The purpose of the FTA [5] is to further analyze the risks involved in an AND/OR diagram. The tree basically shows that the CTF game cannot be played unless there is manual control, functional IR Blaster and Receiver mechanisms, and working cameras to scan the flag. For the IR Blaster and Receiver to work, the Arduino code must be edited properly and the emitters and receivers must be correctly positioned and not broken. For the camera to work, it must be able to detect the flag, and OpenCV must be implemented the correct way to allow cars to pick up and drop the flag. For manual control, the car has to be connected to the laptop, the correct commands have to be written, and the battery for the servo-motor should be charged. However, the team has completed manual control and is almost finished with the IR Blaster and Receiver, so these risks are fairly low.

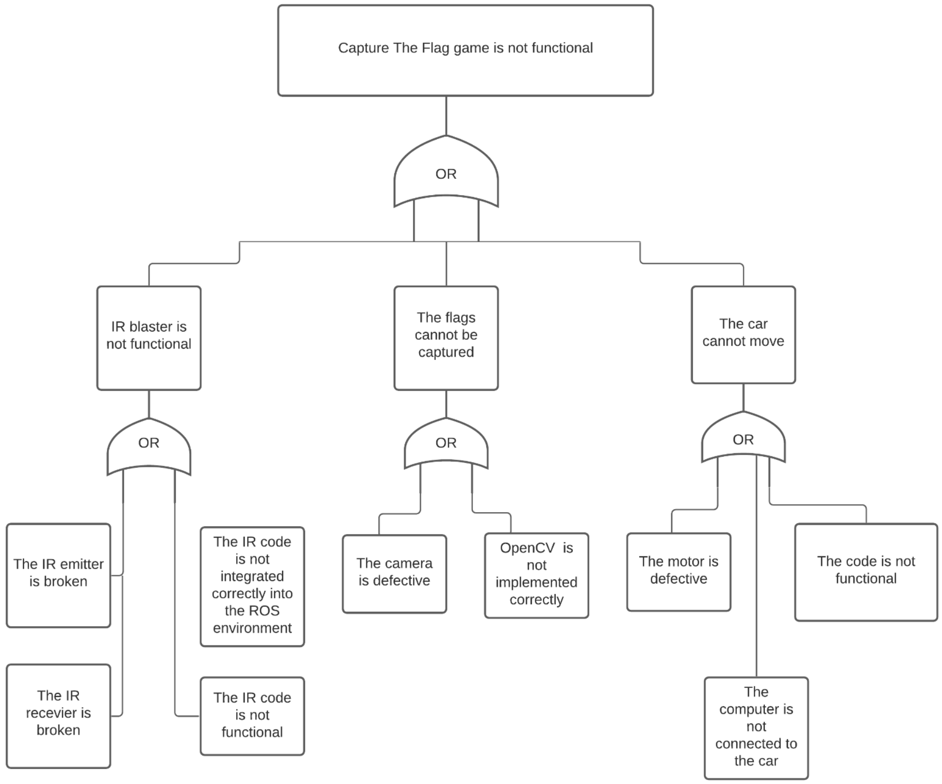

Figure 3.1.3.1: Fault Tree Analysis (FTA) of Issue that Could Cause the Game to be Not Functional

3.2 Design Validation

A validation plan was generated by taking the requirements listed in Section 1.2 Design Parameters and Performance Requirements and making specific requirements for each of the listed needs. The validation plan is laid out below in Table 3.2.1 and was performed by the team. Every Validation Task was completed successfully, however, Validation Task #10 was partially delayed for the purpose of risk mitigation, but the diode upgrade still needs to be tested once the soldering is completed. Otherwise, the team has successfully completed the Validation Plan, thankfully the validation plan has had minimal adjustments throughout the process.

Table 3.2.1 Validation Plan for the Project

Listed Need	Validation Task
The obstacles must not move when a Donkey car at full speed runs into it.	1. Drive Donkey car into workbench and verify workbench stays stationary.
The layout of the arena must allow for a 2ft turning radius.	2. Drive the Donkey Car in a circle and measure the diameter of the circle. Calculate the area of the circle and ensure it can fit within the SwIRL.

The height of the obstacles is between 1 ft and 5 ft.	3. Measure the height of the workbench and verify that it is within 1-5ft in height.
The car's controls must operate the car with variable speeds.	4. Verify the Donkey car can drive forward in slow speed and fast speed for 2 seconds each.
Controller's shooter button will be assigned to a trigger or bumper button of the controller.	5. Verify the Donkey car's IR blaster shoots using any of the following buttons on the Logitech controller: RB, RT, LB, or LT.
Using manual control the car must reach max speed and have the ability to turn.	6. While pressing forward turn the joystick to the left and right and verify the car turns without slowing down.
The chassis must contain the 3.9 in x 3.1 in x 1.1 in (Jetson Nano dimensions).	7. Visually check to see the Jetson Nano is attached to the car chassis.
The chassis must hold 8.8 ounces of weight.	8. Shake the Donkey car back and forth with your hands to see if the chassis deforms, if it does readjust chassis dimensions to account for the weak point.
Movement control, shooting, receiving, and picking up the flag must be done digitally.	9. Using the Logitech controller, drive the Donkey car and shoot another car to verify the IR blaster shooter and receiver works and that the enemy car is disabled. Using the Logitech controller, drive the Donkey car and pick up the digital flag.
The infrared (IR) receiving range on the Donkey car must have a circular 360 degree range.	10. Take two Donkey cars and use one to shoot the other one, position the shooting Donkey car in 8 positions separated by 45 degrees around the receiving donkey car with a minimum distance of 2 feet separating the Donkey cars.
The maximum range of picking and dropping the flag is 3 ft.	11. Verify the Donkey car cannot pick up the flag at a distance of 3.5 feet from the flag pick sign.
A shot Donkey car must not move for 10 seconds.	12. Using one Donkey car as the target, press forward on Donkey car #1 and shoot Donkey car #1 using Donkey car #2 and while still pressing forward on Donkey car #1 verify the motor control of Donkey car #1 stops for 10 seconds.

3.3 Comparison of Design to Requirements

In the table 3.3.1 below, the required tasks for the project are shown along with their current status. The main tasks are the Physical Car and AI Hardware, Manual Control, Shooting Mechanisms, and CTF mechanisms. All of these tasks are mentioned along with their performance requirements. The level of satisfaction of those tasks are also listed. As the project is already finished, all of these tasks are completed. There are still some minor adjustments that can be made with the electrical shooting mechanisms to allow the game to be played more efficiently, but other than that the team is finished with the project.

Table 3.3.1: Required Tasks for the Project and Satisfaction

Functional Requirement	Performance Requirements	Design Feature	Level of satisfaction
Physical Car and AI Hardware	Cars must be able to hold a Jetson Nano underneath the chassis, hold a raspberry pi and servo motor board on the frame, and mount an IR holder to the top of the chassis.	3rd Iterated top chassis, Donkey Car hardware, IR holder	All of these components have successfully met the performance requirements. The Jetson nano is held by the chassis, the raspberry pi is held by the frame, and the servo motor is also held by the frame.
Manual Control	Cars must be able to drive around the SWIRL environment so that they can shoot at each other using the IR Blaster and Receiver mechanisms and scan the flags at each of the enemy bases. The turn radius of the Donkey Car cannot be too large as it has to be able to quickly dodge obstacles within the area.	ROS nodes running successfully, Car batteries charged successfully, cables successfully wired to servo board and raspberry pi, controllers successfully working and connected to each of the cars.	These components have also all been met for our project. The team has successfully developed the nodes in ROS and allowed them to communicate with each other. There were additional nodes for the IR blaster and receiver and color detection.
Shooting Mechanisms (Receiver with IR Blaster)	The cars have to be able to shoot at each other using the IR Blaster and Receiver mechanisms and also receive shots from other cars. The cars must be able to deactivate for small amounts of time during the game after getting shot. The cars also need to be able to hold the IR Blaster mechanisms with an IR holder.	IR Blaster and Receiver device, Emitters and Receivers able to function properly and communicate with the cars, receivers on all sides around the cars.	The cars are successfully able to shoot at each other and receive signals for being shot. There are receivers added to all sides of the car and there will be one main shooting emitter. The team is still planning to add extra diodes for the receivers on all sides around the car.
CTF mechanisms involving a camera	The cars have to be able to scan flags in opposing bases and make it back to their base, which they will again have to scan. The cars need to be able to use the camera to scan the flags with differing colors.	Color code detection through different colored flags, Scoreboard to keep score after the cars get shot, working camera to scan both of the flags.	The team has successfully been able to complete this requirement by creating another node in ROS. The cars are able to use their cameras to scan each of the flags and return back to their bases. The scoreboard is able to keep track of the score. The cars are also able to deactivate for a small period of time if getting shot.

3.4 Cost Accounting and Cost Model

Due to an update of the customer’s needs from the sponsor, the ACV team no longer needs to design and build an interactive arena for the Donkey cars. Because the team’s primary sponsor, Dr. Korok Ray, wants to spend as little money as possible, both parties agreed to utilize the physical space of SwIRL as the arena. Using the SwIRL as the arena saved a significant amount of money for the team.

After the arena issue was concluded, the ACV team still required electrical components for the CTF and shooting mechanisms for the Donkey cars. Table 3.5 shows the quantity, item description, company provider, unit price, and total unit price of items required by the team. The team gained approval for the purchase request from Dr. Rathinam and Dr. Cope, and successfully ordered and picked up the items. The total cost of the items was $122.36, which is within the budget for the ACV team. According to the validation plans and budget analysis, the team does not require other components to achieve the objective and satisfy the customer's needs. Therefore, $122.36 will be the total budget required for the ACV team during the MEEN 401/402.

One of the complications of the Donkey cars is that the IR receiver can only receive shots from the front because only one IR receiver diode is installed in the front of the Donkey car. Three more IR receiver diodes will be soldered onto the IR blaster and receiver board. IR receiver diodes placement will cover the full 360 degrees around, so the Donkey cars can receive shots by the IR blaster from all angles. The team used the Arducam Raspberry Pi Camera Module and utilized the OpenCV library for making the CTF mechanism.

Table 3.5: Budget and Cost Model for the ACV team

#	Item	Description	Company	Unit Price	Total
8	IR Receiver Diode	Increasing DC receiving shot radius	Sparkfun	$1.95	$15.60
2	Smraza Basic Starter Kit	Breadboard, Power Supply, Jumper Wires, Resistors, LED	Amazon (Smraza)	$13.39	$26.78
2	Arducam Raspberry Pi Camera Module V2-8 Megapixel	Raspberry Pi Camera for color or QR code detection	Amazon (Arducam Store)	$39.99	$79.98
Total Cost					**$122.36**

4. Summary

4.1 Work Breakdown Structure

The work breakdown structure (WBS) for creating a new ACV is listed in Table 4.1.1. This WBS is used with the current tools, Solidworks' files, and adjusted catkin_ws that have been made. If a WBS is created without these tools, the WBS would involve a significant amount of small details that the team learned during the process of completing the project and is unfeasible to list holistically. So, it is suggested to use the already created tools and adjusted files for generating a new ACV.

Table 4.1.1 Work Breakdown Structure for New Donkey Car

		Task Name	Duration	Predecesso
1		Physical Adjustments	**8 days**	
2		Purchase Default Donkey Car	1 wk	
3		Print Top Chassis CAD file	2 days	
4		Print IR Clip Holder CAD file	1 day	
5		Print Camera Holder CAD file	1 day	
6		Connect Printed Parts to Default Donkey Car	2 days	2,3,4,5
7		Connect Electrical Parts to Donkey Car	1 day	6
8				
9		**Software**	**15.5 days**	
10		Connect to raspberry pi	3 days	
11		Install Linux OS on raspberry pi	5 days	10
12		Install & run [catkin_ws] code on raspberry pi	1.5 wks	11

The WBS used this semester for the team's plan of the car's controls and the shooting mechanism is listed in Table 4.1.2

Table 4.1.2 Work Breakdown Structure for Car's Controls & Shooting Mechanism

Task Name	Durat	Start	Finish	% Con
Car's Controls	**6 wks?**	**Mon 8/30/21**	**Fri 10/8/21**	**100%**
Manual Control	4 wks	Mon 8/30/21	Fri 9/24/21	100%
Validation Task #1	0 days	Mon 9/13/21	Mon 9/13/21	100%
Validation Task #2	0 days	Mon 8/30/21	Mon 8/30/21	100%
Validation Task # 6	0 days	Wed 9/8/21	Wed 9/8/21	100%
Remap controller keys	2 wks	Mon 9/27/21	Fri 10/8/21	100%
Validation Task #4	0 days	Tue 10/5/21	Tue 10/5/21	100%
				0%
Shooting Mechanism	**45 days**	Mon 9/6/21	Fri 11/5/21	**100%**
Edit arduino code for IR blaster and receiver.	3 wks	Mon 9/6/21	Fri 9/24/21	100%
Save code and test shooting mechanism to see how operates with car.	2 wks	Mon 9/27/21	Fri 10/8/21	100%
Trial and error by editing code to see how shooting mechanisms reacts.	2 wks	Mon 10/11/21	Fri 10/22/21	100%
Test shooting mechanism with new changes and test to sat perf.	2 wks	Mon 10/25/21	Fri 11/5/21	100%
Validation Task #5	0 days	Fri 10/29/21	Fri 10/29/21	100%
Validation Task #10	0 days	Fri 10/29/21	Fri 10/29/21	100%
Validation Task #12	0 days	Wed 11/3/21	Wed 11/3/21	100%

The WBS used for this project is seen in Table 4.1.3 for the CTF mechanism, physical car, and electrical shooting adjustment. The time it took for these tasks is roughly the amount of time it took the team to complete the project.

Table 4.1.3 Work Breakdown Structure for CTF Mechanism, Physical Car, & Electrical Shooting Adjustment

Task Name	Durat	Start	Finish	% Con
CTF Mechanism	**50 days**	Wed 9/15/21	Tue 11/23/2:	**100%**
purchase materials for QR code scanner	2 wks	Wed 9/15/21	Tue 9/28/21	100%
Integrate QR code scanner digitally	2 wks	Sat 10/23/21	Thu 11/4/21	100%
Integrate QR camera physically	2 wks	Sat 10/23/21	Thu 11/4/21	100%
Integrate shooting code to drop flag	2 wks	Wed 11/10/2	Tue 11/23/2:	100%
Validation Task #9	0 days	Mon 11/22/2	Mon 11/22/2	100%
Validation Task #11	0 days	Mon 11/22/2	Mon 11/22/2	100%
				0%
Physical Car	**37 days**	Mon 9/13/21	Tue 11/2/21	**100%**
Validation Task #3	0 days	Mon 9/20/21	Mon 9/20/21	100%
CAD new Chassis	3 wks	Mon 9/13/21	Fri 10/1/21	100%
Print new Chassis	2 wks	Thu 10/7/21	Wed 10/20/2	100%
Place Jetson Nano	1 wk	Mon 10/18/2	Fri 10/22/21	100%
Validation Task #7	0 days	Wed 10/20/2	Wed 10/20/2	100%
Validation Task #8	0 days	Wed 10/20/2	Wed 10/20/2	100%
Place QR camera	1.5 wks	Mon 10/18/2	Wed 10/27/2	100%
Electrical Shooting Adjustment	**3 wks**	**Mon 9/20/21**	**Fri 10/8/21**	**67%**
Buy 6 reciever diodes	1 wk	Mon 9/20/21	Fri 9/24/21	100%
Solder reciever diodes	2 wks	Fri 9/24/21	Thu 10/7/21	50%

4.2 Final Gantt Chart

The general overview of the semester tasks are seen in Figure 4.2.1. The gantt chart below has five subsections: Car's Controls, Shooting Mechanism, CTF Mechanism, Physical Car, and Electrical Shooting Adjustment. The electrical shooting adjustment is almost completed which will put the 402 project completion percentage from 97% to 100%.

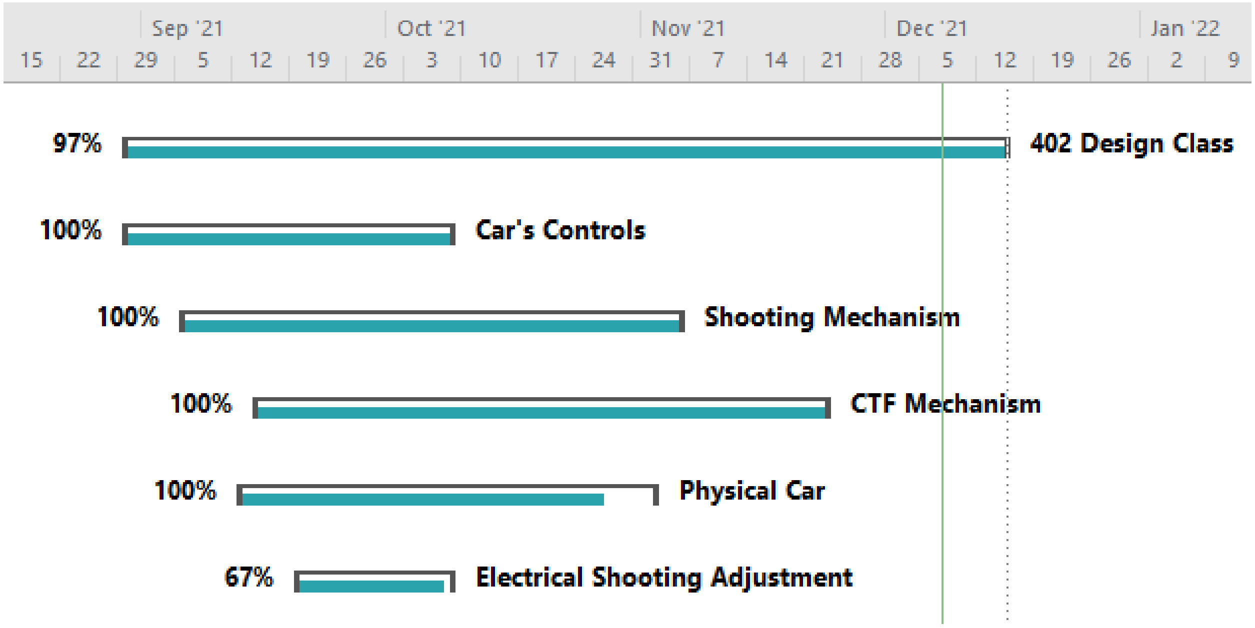

Figure 4.2.1 General Schedule for ACV Project

All of the validation tasks have been completed. Figure 4.2.2 shows the car's controls and shooting mechanism tasks.

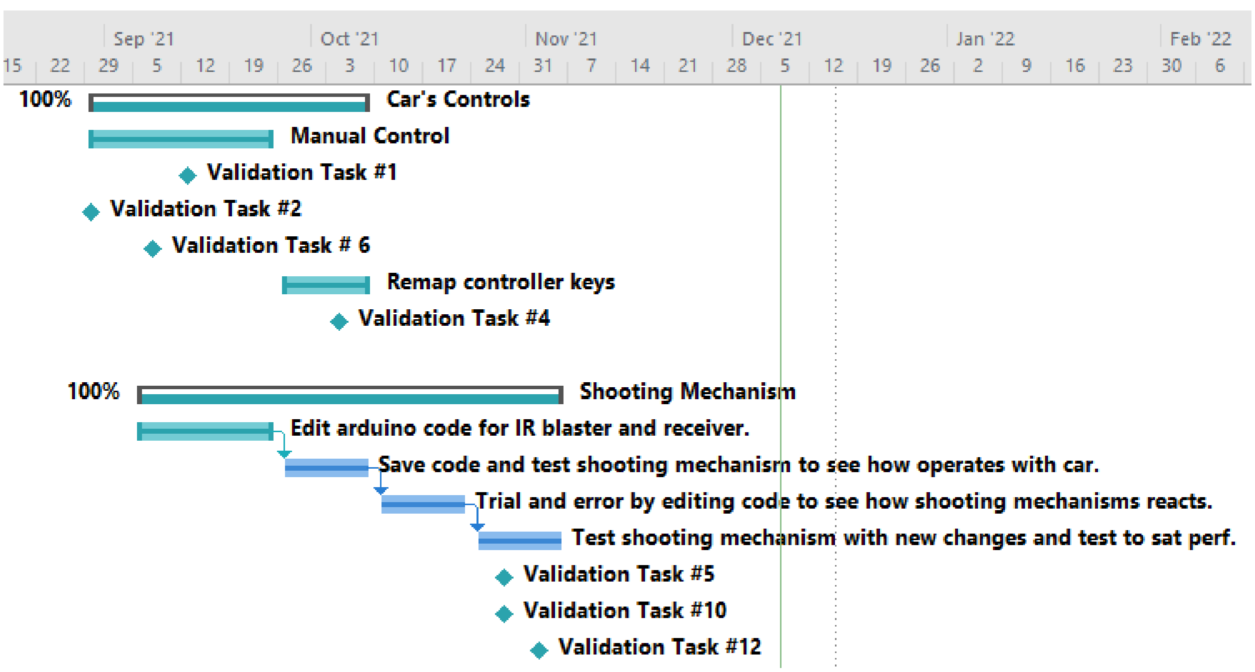

Figure 4.2.2 Gantt Chart Car's Controls & Shooting Mechanism

Figure 4.2.3 shows the in-depth tasks for the CTF Mechanism, Physical Car, and Electrical Shooting Adjustment. All of which are completed except for the final electrical shooting adjustment of increasing receiving range.

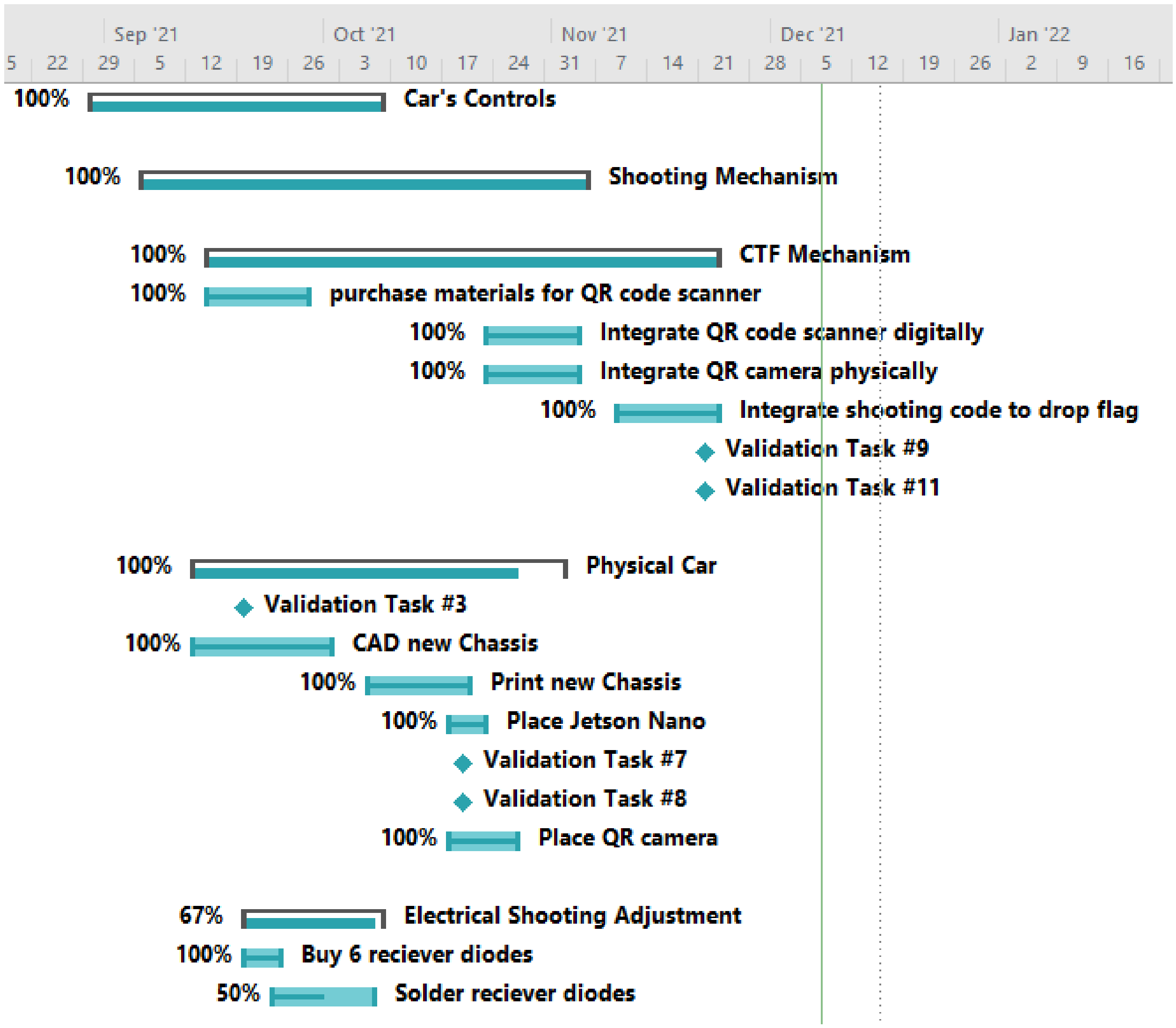

Figure 4.2.3 Gantt Chart of CTF Mechanism, Physical Car, & Electrical Shooting Adjustment

4.3 Technology Development

At this point, the team has successfully completed all stages in the project and there are no further technology advancements necessary. The team is only working on adding to the electrical shooting components, to make the game play smoother. If there are issues with testing after this adjustment has been made, the team will go back to the original design, since this adjustment is more of a benefit than a mandated requirement. There are no issues with the project running as of right now, and even if the CTF mechanisms do not function properly, the team also devised a back-up CTF mechanism, shown in Figure 4.3.1, which is already 3D printed and just needs to be picked up from the laboratory. In short, the team would just like to add a bit more to the electrical component so that the cars can shoot at each other more effectively.

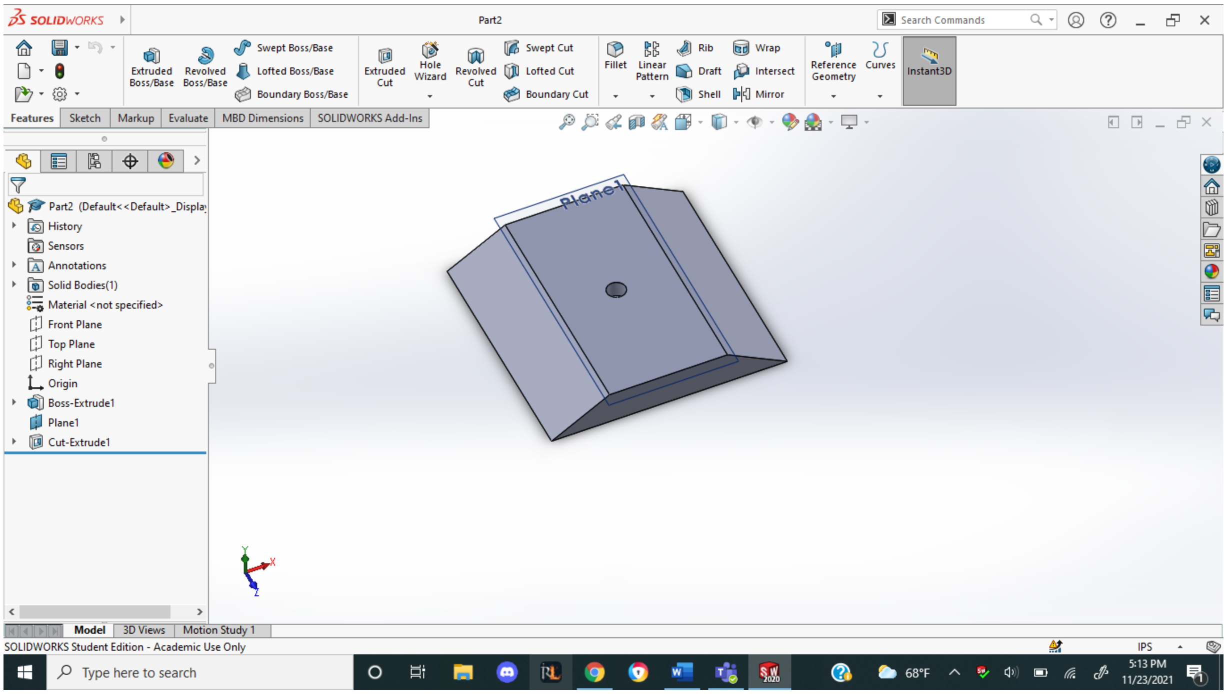

Figure 4.3.1: Picture shows the CTF backup mechanism designed in SolidWorks

4.4 Limitations of Current Solution

There are few limitations of the current solution since the project is finished, but are worth mentioning. One limitation is that when playing the game, there is no way for the players to know when their cars get shot or score a point. This can only be known by an operator managing the computers that the cars are connected to when playing the game. This would be nice to know for the players for convenience sake, rather than requiring a 3rd member to monitor the computers. Also, another limitation of the game is that the cars are not able to make a sound as they get deactivated. The only way for the players to know that they got shot is by pressing buttons and seeing if their cars can move around. If the cars are not able to move for 5 seconds, that means they got shot. Again, these are minor limitations and the game can still be played successfully without fixing them, but doing so would make the experience more helpful for the operator.

4.5 Future Work

The Autonomous Combat Vehicle team has successfully met all needs of the project. The team has manually controlled the cars, programmed the IR blaster and receiver using the Arduino software, and developed color code detection nodes to play the CTF game with 2 colored flags. Future work includes fixing minor adjustments for the electrical shooting mechanism. However, the team has already demonstrated playing the game at the final presentation. These adjustments could be useful for the virtual project showcase if the team decides to include another demonstration of the game. The team is also considering making a guide for the CS team to view when they begin the implementation of the Jetson Nano. In the entirety of the project, the team used resources such as the ACV guide and online sources, but often were left with many questions as the guides were not specific enough. To account for this in the future, the team wishes to create a guide for the CS department as they will be taking over the project after the end of the semester. The CS team will then be able to see the process the team went through to finish each part of the project. This will prove to be very useful when they work with the Jetson Nano to allow autonomous capabilities for the cars.

5. References

[1] Johnson, R. (2021, January 27). *Army to Stay Course on Modernization Priorities, Official Says*. Potomac Officers Club. https://potomacofficersclub.com/army-to-stay-course-on-modernization-priorities-official-says/.

[2] Hirtz, Julie, et al. *A functional basis for engineering design:Reconciling and evolving previous efforts*. 13 ed., Research in Engineering Design, 2002.

[3] "About OpenCV." *OpenCV*, 4 Nov. 2020, https://opencv.org/about/.

[4] Asq.org. 2021. *What is FMEA? Failure Mode & Effects Analysis | ASQ*. [online] Available at: <https://asq.org/quality-resources/fmea> [Accessed 8 December 2021].

[5] *Fault Tree Analysis (FTA) – Definition & Examples | EdrawMax Online*. https://www.edrawmax.com/fault-tree-analysis/. Accessed 8 Dec. 2021. Accessed 8 Dec. 2021.

6. Appendices

A. Embodiment Design Checklist

3.2.1 Function – *Are the customer's needs satisfied? Is the stipulated product architecture and function fulfilled? What auxiliary functions are needed?*

There are 4 main customer needs in the project. The first need is that the arena is the Southwest Innovation and Research Lab environment. This need is satisfied because recently, the sponsors announced the Grand Opening of the SwIRL Aggie Robotics Lab. The team showcased the cars with full manual control and confirmed with the sponsors that the space within the lab is enough, the tables can function as obstacles, and the cars can move across the floors. The second need is that the cars must be fully manually controlled. The team has already satisfied this need. Since the beginning of MEEN 401, the team has made continuous progress in manual control, and the team now has 3 cars that can all move using a Logitech Controller. The third need is that the Donkey Cars need to have a compartment for a device that allows AI capabilities. The team has also satisfied this need as well. The team has 3D printed a second iteration of the Donkey Car chassis at 90° angles with a flat top that matches the surface area of the Jetson Nano. The fourth need is that the CTF game must be able to be played. This need is finished, as it was the team's last focus. The team has also finished testing the IR Blaster and Receiver and made it able to shoot at cars and receive shots from each car. CTF mechanisms are also being finished; the team is using color sensors to play CTF. The project architecture involved in the project was originally the need to design a physical arena, but after communicating with the project sponsors and stakeholders, this need is no longer required, as the existing SwIRL lab environment can be used and the project is more software focused. Thus, it can be considered fulfilled. The auxiliary functions would be removing extra 3D printed material during designing the chassis and IR holder.

3.2.2 Working Principles and Form Solutions – *Do the chosen form solutions (architecture and components per function) produce the desired effects and advantages? What disturbing noise factors may be expected? What byproducts may be expected?*

The ACV team designed a project architecture in the last semester's Design Report 3. This model showed the cars communicating with each other and shooting at one another while playing the game. The only update for this model was the fact that the car's are using their own access points instead of connecting to a local network. This will result in the advantages such as being able to have manual control without the need for the internet. There are no noise factors to be considered, and byproducts are mentioned in the *Function* section above.

3.2.3 Layout, Geometry, and Materials – *Do the chosen layout, component shapes, materials, and dimensions provide minimal performance various to noise (robustness), adequate durability (strength), efficient material usage (strength to mass ratio), suitable life (fatigue), permissible deformation (stiffness), adequate force flows (interfaces and stress concentrations), adequate stability, impact resistance, freedom from resonance, unimpeded expansion and heat transfer, and acceptable corrosion and wear with the stipulated service life and loads?*

As the ACV project is more software focused and less hands-on, there were little to no factors related to mechanical or thermal properties of materials. Since this section is mainly focused on these concepts, the team will cover the answers in other areas of this checklist.

3.2.4 Energy and Kinematics – *Do the chosen layout and components provide efficient transfer of energy (efficiency), adequate transient and steady state behavior (dynamics and control across energy domains), and appropriate motion, velocity, and acceleration profiles?*

Regarding the energy and kinematics involved in the project, the Donkey Cars are manually controlled and can function using a Logitech controller, so the user can control the velocity and turn angle of the car. Key re-mapping was done to allow the cars to be able to be controlled similar to modern day racing games. Files were edited in ROS to change information about the cars maximum velocity and acceleration. The cars originally picked up velocity fairly slowly, and after some troubleshooting, the controller and joystick can now be used to control the speed of the cars. Also, the Lithium-Ion battery is responsible for powering the servo-motor. This would be a process of converting chemical to electrical to mechanical energy. The motion of the car could best be described as transient until it reaches its maximum velocity and steady state afterwards.

3.2.5 Safety – *Have all of the factors affecting the safety of the user, components, functions, operation, and the environment been taken into account?*

Although there were no quantitative measures in the project so far, such as calculating a factor of safety, there are still some safety factors affecting some of the components in the project. The Donkey Car must not be kept out in extreme temperatures and is probably best kept at room temperature conditions. The Raspberry Pi circuit board, servo-motor and board, Lithium-Ion battery, battery bank, IR Blaster and Receiver, Logitech controller, and other electrical equipment will malfunction if exposed to unusual temperatures of heat, and moisture. Also, the car is often kept on a wooden block when it is on to ensure that it doesn't accidentally move when the Logitech controller is nearby and being tested. This protocol is very important because the car accelerates very fast and could get damaged with the momentum it takes during impact with the wall. As far as safety precautions taken for the team, the team ensured that there was adequate A.C. at the SwIRL during the meetings that it gathered to work on the project every Monday and Wednesday. The SwIRL was also made sure to be locked during the hours that the team was not present, and duplicate keys were made for each member in the team. During work done in the FEDC, appropriate clothing and P.P.E. was worn when 3D printing materials such as the chassis and IR holder. As far as environmental concerns, since the project is mostly software orientated, there weren't any materials or processes that could potentially affect the surrounding environment.

Figure 3.2.3: Entire robotics kit used for Donkey Car work, including safety block

3.2.6 Ergonomics – *Have the human-machine relationships been fully considered? Have unnecessary human stress or injurious factors been predicted and/or avoided? Has attention been paid to aesthetics and the intrinsic "feel" of the product?*

Ergonomics, efficiency, comfort, and convenience are definitely considered in the duration of this project. One of the most obvious situations that led to stress and inconvenience that the team had to deal with was connecting the car to the local network. During this process, the members on the team had to deal with several cables and wiring. When working at home for example, an ethernet cable had to be connected to the laptop, both batteries on the car had to be connected to the wall or charged prior, and the member had to be able to access zoom meetings to meet with teammates. This tedious process was often inconvenient for the members, so they often spent several minutes prior to meetings to ensure that all devices were set up correctly to result in an efficient meeting. Even with the current status of the Donkey Car, the team is still required to connect to the car's Wi-Fi at times for manual control. This process can also be quite difficult, because the laptop will not be able to access the internet during the time it is connected to the car, and the user will not be able to access important resources such as the ROS wiki or zoom. In this case also, prior planning was done to ensure the members set up another computer or phone before meetings to be able to access the internet when needed. Finally, during the first few weeks of the semester, the SwIRL Lab did not have internet. This issue was soon fixed as well because the team needed access to Wi-Fi in the lab to have its weekly meetings. Also, as mentioned in the other sections, higher quality chairs and A.C. were added to allow more physical comfort.

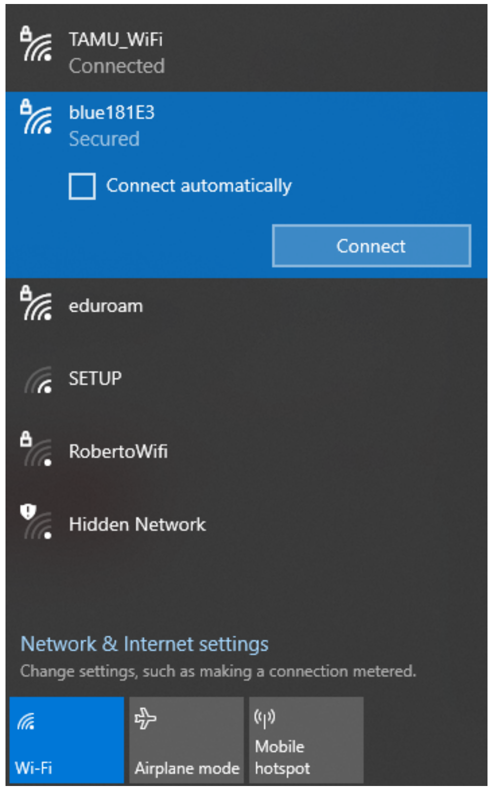

Figure 3.2.4: Internet settings panel shows that the car Wi-Fi can be connected to a laptop

3.2.7 Production – *Has there been a technological and economic analysis of the production processes, capability, and suppliers?*

The technological analysis has been described in other sections, such as System Description. Also, economic analysis and costs are mentioned later in this *Embodiment Design checklist*. Costs are also mentioned in the Budget and Cost Model.

3.2.8 Quality Control – *Have standard product tolerances been chosen (not too tight)? Have the necessary quality checks been chosen (type, measurements, and time)?*

There have been some tolerances that the ACV team has worked with. The Donkey Cars all had a turn radius of approximately 5 feet in the past, but that changed when the team figured out how to change the degree of turn angle of the cars. However, given the new needs of the sponsors, the team realized that this was no longer an issue because the arena could be the entire space of the SwIRL. The team also worked with very small tolerances when 3D printing the 2nd iteration of the chassis and 3rd iterations of the IR holder. The IR holder was just a millimeter too small during the second iteration, and then just a millimeter too big during the third iteration. However, the team managed to find the correct dimensions for the IR holder. The new chassis at 90° angles also holds the surface area of the Jetson Nano. These are all the main quality checks necessary for the project.

3.2.9 Assembly – *Can all internal and external assembly operations be performed simply, repeatably, and in the correct order without ambiguity? Can components be combined (minimize part count) without affecting modular architectures and functional independence of the product?*

The main assembly involved in the project was the prior installation of the Donkey Car hardware and IR Blaster and Receiver, both of which were done prior to the start of the project by a graduate student named Yan Yao. Yan built a guide called the ACV guide that the team is currently following for software installations and progression. The ACV guide covers all steps in building the Donkey Car and IR Blaster and Receiver in detailed, orderly steps. However, the team would like to keep track of their progress and create another guide in more detail so that the C.S. team that works on the project next can easily repeat the steps.

3.2.10 Transport – *Have the internal and external transport conditions and risks been identified and solved? Have the required packaging and dunnage been designed?*

Due to the fact that the ACV project is mostly software related, there are no packages or other items that need to be transported. The items ordered recently however, such as the camera module, Smraza kit, and diodes will all arrive through mail.

3.2.11 Operation – *Have all of the factors influencing the product's operation, such as noise, vibration, and handling, been considered?*

As far as the Donkey Car's operation is concerned, the car makes a small noise when the Lithium-Ion battery is turned on to indicate that it is on. The car also makes sounds as it is driving. The car also has fairly decent handling, as the joystick can be used to turn it and move forward or backward. The IR Blaster and Receiver will be able to shoot once it is fully programmed, and this will be done using the back keys on the Logitech controller.

3.2.12 Life Cycle – *Can the product, its components, and its packaging be reused or recycled? Have the materials been chosen and clumped to aid recycling? Is the product easily disassembled?*

Some of the components and materials in the project can be reused and recycled. The entire robotics kit will be reused by the next team that works on the project in the following semester. Also, the entire set of new mechanisms, such as the QR code and scanner as well as color sensors will be reused as well. Potentially basic materials such as tape and cardboard may be used to quickly mark an arena within the SwIRL, and these can be recycled afterwards.

3.2.13 Maintenance – *Can maintenance, inspection, repair, and overhaul be easily performed and checked? What features have been added to the product to aid in maintenance?*

Troubleshooting and networking is a main part of the maintenance of the Donkey Cars and IR Blaster and Receiver. It is important to continuously test the devices that the team has worked with since the start of the project to make sure they still run without errors. The team has saved a backup Donkey Car with all of the files used in manual control, as well as a backup IR Blaster and Receiver.

3.2.14 Costs – *Have the stipulated cost limits been observed? Will additional operational or subsidiary costs arise?*

Regarding the budget and costs of supplies in the project, the team has made three main purchases recently. The team has purchased an electronics and circuits kit, a camera module, and extra diodes for the IR Blaster and Receiver. The team has received the Donkey Cars as well as their hardware and the IR Blaster and Receiver from the sponsor at no cost. The team has also received the required cables needed for the cars as well as the Logitech controllers required to control them. In the *Budget and Cost Model* section, there is a more detailed table listing the components that the team has purchased so far and their corresponding costs. The team also used color sensors for the CTF mechanisms in the project. If these costs are decided and set in stone, another budget approval request form will be sent to the sponsor.

3.2.15 Schedules – *Can the delivery dates be met, including tooling? What design modifications might reduce cycle time and improve delivery?*

According to the Gantt chart addressed later in the report the team has completed the project on time. The IR Blaster and Receiver are being tested and the cars are able to shoot each other. The team is communicating with the C.S. team stakeholders to finalize plans for the CTF mechanisms and how the game should be played. The team believes that more and more communication with the C.S. team will lead to faster implementation of the Jetson Nano. The team worked with Yan Yao to finish the electrical shooting adjustments and then focused on the CTF needs in the final month.

B. Manual Control of the Donkey Cars

1. The Raspberry Pi does not come preloaded with the correct operating system configuration, so a custom one must be downloaded and installed onto the SD card, which acts as its hard drive.
2. Once installed, some settings must be configured, and general updates/installations must be fully operational.
3. Using your computer, search for the WiFi network called “ubiquityrobotXXXX” and sign in using the password “robotseverywhere”
4. Once logged on, launch the terminal
   a. Use the command “ssh ubuntu@10.42.0.1”
   b. Use the password “ubuntu”
   c. If there is an error, wait approximately 5 minutes for Raspberry Pi to turn on
2. Repeat step 2 two more times in another terminal
3. Use the command “roslaunch donkey_llc keyboard_demo.launch
4. Use the command “rosrun joy joy_node”
   a. A beep sound will come from the Donkey car (motor control)
5. Use the command “rosrun teleop_twist_joy teleop_node”
6. Use the Logitech controller to control the Donkey car manually

C. Nodes and Topics Description

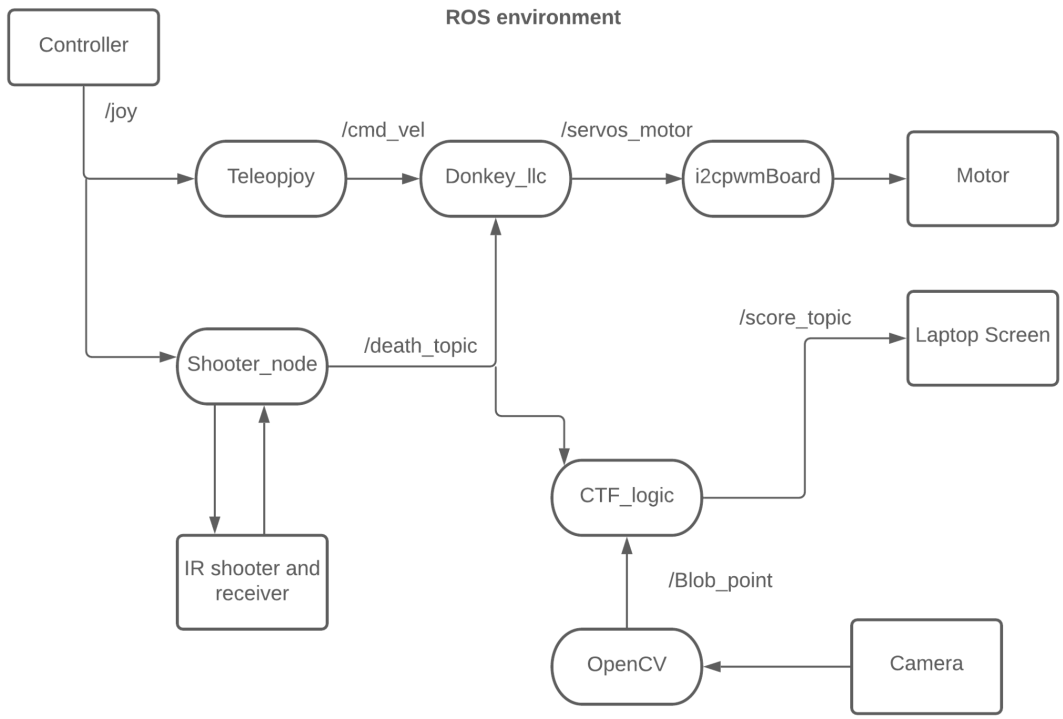

Figure C.1: ROS environment of the whole system

Controller (Hardware component):
The actual controller is treated as a node in the environment which publishes the topic /joy. The topic joy contains information about which button is pressed.

Publish: /joy

Teleopjoy (C++):
Import the buttons pushed on the controller and convert them into axial and angular velocities. It is a library that was modified to remap the controller keys.

Subscribe:
/joy

Publish:
/cmd_vel

Donkey_llc (Python):
Donkey low level control (Donkey_llc) is a node that converts the axial and angular velocities into values understood by the motor driver node.

Subscribe:
/cmd_vel

Publish:

i2cpwmboard:
This node is using I2C communication control to send signals as PWM. It is the controller of the motor.

Subscribe:
/servos_motor

Shooter_node (Python):
Command the IR Blaster to shoot and detect incoming shots. This is a custom node that will send commands to shoot whenever a button is pressed on the controller.

Subscribe:
/joy

Publish:
/death_topic

OpenCV (AI Library):
OpenCV is a real-time Computer Vision Library used for color

Subscribe: None
Publish: /Blob_point

CTF_node (Python):
This node contains the game logic. it will award a team a score if they scanned the enemy flag then its own flag. It also drops the enemy flag whenever it gets shot. It will also indicate that a Donkey car is carrying the flag through a signal

Subscribe:
/FlagPickUp_topic
/death_topid

Publish:
/scoreUpdate_topic

Screen_node (Python):
Screen_node is used to Show the score on a Laptop screen. It will only show the score of one team.

Subscribe:
/scoreUpdate_topic

Publish:
/Score_topic

Camera (Hardware component):
A wide angle camera is used as a node in the ROS environment. Its live feed will be processed by the OpenCV node

Motor (Hardware component):
The motor on the car is used as a node in the ROS environment. It is controlled by the i2cpwmboard node.

IR shooter and receiver (Hardware component):

The IR shooter and receiver (IR Blaster) is a hardware node. it will send and receive information from the Shooter_node

Subscribe:
/shoot_topic

Publish:
/death_topic