UNIVERSITY OF CENTRAL FLORIDA

Robot Basketball

Project Documentation - Senior Design I, Group 9, Summer 2019
Brandon Gross • Electrical Engineering
Suvrat Jain • Computer Engineering
Cory Ricondo • Computer Engineering
Mathew Schneider • Computer Engineering

Table of Contents

Table of Figures

Table of Tables

1.0 Executive Summary

The project documentation is centered around the design of a Robot Basketball Arcade game. The arcade basketball arena sits on top of a tabletop and one to two players can pick up a controller and move around the robot basketball player to intake and launch a small basketball into a hoop. The arena garners attention from near and far with an exciting display of robot athleticism and engaging displays and sounds. The project provides an exciting platform to task the team with modern engineering challenges such as robotics, computer vision, game development, and embedded systems.

The major systems are designed with one ultimate goal in mind: player engagement. The final product is ultimately meant to be fun and entertaining such that people want to keep playing the game. In order to achieve this goal, the project's features and functions are fully described in requirement specifications and constraints, and relevant standards are researched and implemented where appropriate. The project is split into 3 major systems: Robot, Arena, and Game Systems. Each system contains many subsystems and components that interface with one another to implement functionality and features. The robot is responsible for picking up and launching a ball with a fast and capable mechanical system that feels fluid to the player. The arena handles high level logic and computer vision to maximize robot intelligence and autonomy. The game system provides a high-fidelity representation of the robot and arena to guarantee the player can fully engage with the system with minimal frustration. The game system also gives full control to the player to customize the robot's functionality to match the user's playstyle. The project includes both high and low-level software to hide complexities from the player to ensure maximum usability and accessibility.

The following report details the full design process including project description and narrative, engineering requirement specifications, realistic design constraints, system architecture, a detailed breakdown of system components and an administrative approach. Each system component contains a description, relevant research, design, and prototyping and testing sections. The component description translates the project's requirements and features to a narrative discussion detailing the various design aspects of that particular component. The research sections discuss in detail the possible technologies, high-level designs, or purchasable components that satisfy requirements for the component. The design section fully defines the ultimate design that the team utilizes to solve the requirements for the project. The prototyping and testing sections describe how the design is to be built and tested to ensure that the component actually solves the problem within required specifications. Each section is designed and described with the previous section's design decision in mind but attempts to be agnostic to it. Non-critical interfaces are defined in their relevant sections, but critical interfaces are designed and developed separately in another section to mitigate risk.

2.0 Project Description

2.1 Motivation

Entertainment is an essential part of life in the City of Orlando. Amusement parks, arcades, sports, movies and television retire us of our tiredness and fulfill our lives with optimism and sheer excitement. The Robot Basketball game project is chosen to create dynamic, interactive entertainment for everyone to enjoy.

This project is proposed in the spirit of Robocup challenge; Robocup is a standardized soccer-based robotic competition with a variety of leagues. In general, robots compete against one another utilizing complex algorithms developed by engineers. In the case of Robot Basketball, two human players can compete against one another by controlling the robot to move and shoot the basketball. However, due to perception and coordination problems that come from remotely operating robots, the players may need some assistance to maximize amusement. This introduces a complex engineering challenge that involves some level of machine intelligence to achieve high control fidelity.

The team proposes this project as a foundation for learning a wide variety of skills including Robotics, Computer Vision, Machine Learning, PCB Design, Bluetooth communication, Game and App development, and real-time control.

2.2 Goals and Objectives

The overall goal in this project is to create an arcade-style entertainment system that is both robust and intelligent. The product should be able to fit on typical foldable tables and should be playable by at least one, but preferably two people. The system should be designed modularly such that different subsystems can be designed, tested, and created independently without disassembling the entire system. The system should incorporate both high level software and low-level hardware interfacing. The robot should be low cost such that multiple robots can be created. The robot should be capable of collecting and launching the ball into a scale hoop with high accuracy and precision. The robot should be quick to traverse the court to increase mid-game activity. The system should assist the player by performing calculations to increase shot accuracy. The arena should display information to the player including game type, score, and debugging information. The final product should be engaging and attractive.

2.3 Design Process

The design process for this project follows the following pattern: Define the system features from market requirements, Define the subsystem components, Determine the requirements for each subsystem requirement specifications, define the tests to evaluate the subsystem requirement specifications, research components, design subsystem, prototype subsystem, and test subsystem. This pattern is chosen because it follows the logical progression of system development such that a final product meets the actual market requirements defined by customer. Each test defined early in the process is directly traceable to an engineering requirement specification. The tests are defined before the design is complete in order to create an objective set of tasks to be completed such that the requirements are fully satisfied. This prevents changing tests in order to ensure the test passes. The pattern is shown graphically in Figure 1.

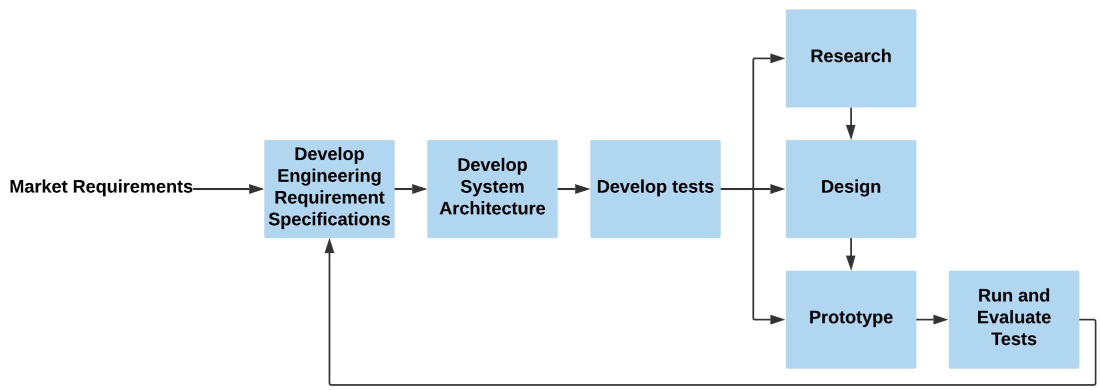


Figure 1 Project Design Process

2.4 Realistic Design Constraints

These constraints are those placed upon the project by environmental factors such as transportation, budget, or customer requirements. The constraints arise from the need to present the project in appropriate settings, and also to constrain the team adhere to deadlines and restrictions placed upon the project by the senior design committee.

2.4.1 General

General constraints pertain to the constraints enforced by the university or by team members due to environmental factors or arbitrary distinctions. The identified constraints can be found in Table 1, Table 2, Table 3, and Table 4.

Table 1 Project constraints

Constraint	The project shall…
C.P.1	Be transportable in a standard-sized sedan
C.P.2	Be designed by August 2, 2019
C.P.3	Be built and tested by November 15, 2019

C.P.4	Utilize GitHub as a version control system

Table 2 Arena constraints

Constraint	The Arena shall…
C.A.1	Be powered by a standard US 120V 60Hz wall outlet
C.A.2	Be able to rest on two standard folding tables
C.A.3	Have only 1 cable that plugs into the wall

Table 3 Robot constraints

Constraint	The Robot(s) shall…
C.R.1	Utilize a custom PCB that fits within size constraints required by the project
C.R.2	Utilize a PCB that contains limited through-hole soldering
C.R.3	Be powered by a battery

Table 4 Game constraints

Constraint	The Game shall…
C.G.1	Utilize a market-available Game Engine

2.4.2 Economic Constraints

Economic constraints are constraints that pertain to the microeconomic and macroeconomic factors that affect design decisions. These factors can include things such as taxes, impacts to stock markets, and the general cost and value of a product. In the case of Robot basketball, the primary economic factors are those that limit the quality or quantity of the parts the project can afford. Further, if the project is to be utilized in an actual arcade, some analysis must be done to ensure marketplace viability. The identified constraints can be found in Table 5.

Table 5 Economic Constraints

Constraint	Economic Constraint
C.ECON.1	The project shall cost no more than $1000
C.ECON.2	The robot shall cost no more than $300
C.ECON.3	The arena shall cost no more than $400
C.ECON.4	The game system shall cost $0

C.ECON.5	The robot design and cost shall be scalable to multiple copies

2.4.3 Environmental Constraints

Environmental constraints pertain to the consideration of environmental impacts such as disposal, energy efficiency, or carbon footprint. For this project, the environmental considerations directly relate to the energy efficiency and battery technology. The identified constraints can be found in Table 6.

Table 6 Environmental Constraints

Constraint	Environmental Constraint
C.ENV.1	The project shall be energy efficient
C.ENV.2	The project shall utilize organic materials where feasible
C.ENV.3	The project shall utilize rechargeable batteries where appropriate

2.4.4 Social Constraints

Social constraints pertain to human factors such as psychology, social etiquette, privacy, education, and accessibility. Social constraints are the largest driving force in this project due to the nature of human interaction with the final product. The identified constraints can be found in Table 7.

Table 7 Social Constraints

Constraint	Social Constraint
C.S.1	The project shall be easy to utilize
C.S.2	The project shall display appropriate information to enhance understanding by the user
C.S.3	The project shall implement accessibility for people with disabilities if time and budget permits
C.S.4	The project and associated documentation shall ensure appropriate terms (Pronouns, avoid trigger words, etc.) are utilized

2.4.5 Political Constraints

Political constraints pertain to the government as an overseer and as a customer. There are no driving political constraints for this project outside of following governing laws and regulations.

2.4.6 Health and Safety Constraints

Health and safety constraints pertain to the safe operation of a product and ensuring no harm comes to a person by being associated with the product. There are several health and safety constraints for this project. The identified constraints can be found in Table 8.

Table 8 Health and Safety Constraints

Constraint	Health and Safety Constraint
C.HS.1	The project shall ensure all electrical components are properly secured and grounded. No bare wires are to be accessible without a locked enclosure
C.HS.2	The project shall ensure all flying objects are appropriately secured and cannot leave the Arena
C.HS.3	The project shall ensure no user can interact with the robot while it is under active power
C.HS.4	The project shall ensure ergonomically considerate devices are utilized when feasible

2.4.7 Manufacturability Constraints

Manufacturability constraints pertain to the construction of the physical device and development of any software required to operate the device. This includes utilizing widely available standard components such as screws, bolts, and designing custom devices that can be made with available tools and machinery. For this project, several mechanical devices are required, and effort is put in to ensure the product can be manufactured by University students with available resources. The identified constraints can be found in Table 9.

Table 9 Manufacturability Constraints

Constraint	Manufacturability Constraint
C.MANU.1	The project shall utilize ISO hardware where needed
C.MANU.2	The project shall be designed with the following available machinery in mind: Saw, Table Saw, Jigsaw, Drill, Laser-Cutter, 3D Printer, Heat Gun, Soldering Iron
C.MANU.3	The project shall utilize as few parts as possible
C.MANU.4	The project shall utilize as few custom components as possible

2.4.8 Sustainability Constraints

Sustainability constraints pertain to the maintenance and support of the project after development and release to reduce or eliminate the need for additional resources. Additionally, renewable resources are to be utilized to ensure the long-term sustainability of the planet. For this project, sustainability indicates the ease of repair, changes, and expandability of the product by the end of the term, and the use of organic materials where possible. The identified constraints can be found in Table 10.

Table 10 Sustainability Constraints

Constraint	Sustainability Constraint
C.SUS.1	The project's mechanical design shall be maintainable
C.SUS.2	The project's mechanical design shall utilize locking hardware where feasible
C.SUS.3	The project shall utilize source control for software
C.SUS.4	The project shall include expandable hardware to meet future requirements

2.5 Engineering Requirement Specifications

The Engineering Requirement specifications found in the following tables are requirements developed by the project team such that the project is fully defined and constrained. The requirements are a guiding force behind the entire project, and each design decision made in the following sections are traceable back to these defined requirements.

2.5.1 Project Requirements

The project requirements in Table 11 define the major subsystem components and the large overarching requirements for the entire product. They act as a governing set of requirements that the project must achieve in order to be considered successful.

Table 11 Project requirements

Requirement	The project shall…
R.P.1	Contain three high-level subsystems capable of communication: Arena, Robot, and Game
R.P.2	Allow a human-player to control the robot-subsystem to drive and launch a ball
R.P.3	Take efforts to ensure safety of both human players and subsystems
R.P.4	Identify high-risk interfaces and fully define & design them

2.5.2 Robot Requirements

The robot requirements in Table 12, Table 13, Table 14, Table 15, Table 16, and Table 17 describe and define the functionality of the robot. The major subsystems under the robot are described in individual tables labeled appropriately.

Table 12 Robot requirements

Requirement	The Robot(s) shall…
R.R.1	Weigh no more than 8 lbs.
R.R.2	Contain a launching mechanism capable of launching a 1.5” diameter rubber ball
R.R.3	Contain an intake mechanism for acquiring a 1.5” diameter rubber ball from ground level
R.R.4	Be sturdy, robust, and resilient regardless of subsystem weight
R.R.5	Perform required functionality regardless of ball holding status
R.R.6	Be resilient to hitting the ball while driving
R.R.7	Be resilient to collisions

Table 13 Robot Base Requirements

Requirement	The Robot’s Base shall…
R.R.B.1	Be capable of holonomic locomotion
R.R.B.2	Contain at least 3 Drive motors
R.R.B.3	Traverse in one direction at minimum 0.3 m/s
R.R.B.4	Traverse the court without unintentional slipping
R.R.B.5	Be able to maintain a shot angle while driving

Table 14 Robot Launcher Requirements

Requirement	The Robot’s launcher shall…
R.R.L.1	Contain no more than two motors
R.R.L.2	Maintain at least 75% shot accuracy from anywhere on the court
R.R.L.3	Be capable of launching a ball with different forces for a required distance

Table 15 Robot Intake Requirements

Requirement	The Robot's Intake shall…
R.R.I.1	Contain no more than one motor
R.R.I.2	Intake the ball while stationary and moving from a variety of angles

Table 16 Robot Electrical Requirements

Requirement	The Robot's Electrical system shall…
R.R.E.1	Utilize a battery that can safely operate at the loads required for the systems
R.R.E.2	Convert voltage from 12V DC to 9V DC, 7.2V DC and 5V DC with high efficiency
R.R.E.3	Support an embedded controller capable of processing controls for a minimum 6 motors
R.R.E.4	Be power efficient in operation to run more than 10 minutes
R.R.E.5	Utilize a microcontroller capable of I2C, SPI and UART communication protocols

Table 17 Robot Software Requirements

Requirement	The Robot's software system shall…
R.R.S.1	Communicate with the arena subsystem at a rate of at least 30Hz
R.R.S.2	Utilize sensor data to close feedback loops on relevant actuators at a reasonable update rate
R.R.S.3	Utilize software that is fully unit tested
R.R.S.4	Utilize a robust deterministic state-machine

2.5.3 Arena Requirements

The arena requirements in Table 18, Table 19, Table 20, and Table 21 define the features and functionality of the Arena system and its respective subsystems. Each major subsystem's requirements can be found in the appropriate table.

Table 18 Arena requirements

Requirement	The Arena shall…
R.A.1	Be no larger than 2 meters length, 2 meters width, and 1.5 meters height
R.A.2	Weigh no more than 75 lbs. total

R.A.3	Contain at least 1 rubber ball that is no smaller than 1.5” diameter
R.A.4	Contain at least 1 basketball hoop no smaller than 1.5” diameter
R.A.5	Have flat ground with scale basketball court markings
R.A.6	Be easy to put together and take apart (Less than 3 minutes each)
R.A.7	Contain a surface that is level
R.A.8	Be resilient to impacts such as falling or dropping
R.A.9	Contain walls such that the ball or robot does not go through
R.A.10	Contain accurate basketball court markings
R.A.11	Utilize a ball that weighs no more than 5 grams
R.A.12	Utilize a ball that is not severely impacted by aerodynamic conditions
R.A.13	Securely mount the hoop to the frame
R.A.14	Contain a hoop that can fit a ball no greater than 2.5”
R.A.15	Contain a display to show players and spectators game status
R.A.16	Contain LED lights for status indication and consistent lighting on the court
R.A.17	Contain software that is fully unit tested

Table 19 Arena Display and Sounds Requirements

Requirement	The Arena Display and Sounds shall…
R.A.DS.1	Contain a display that is widescreen with a refresh rate of at least 60 Hz and 720p resolution
R.A.DS.2	Contain a display that can be viewed outdoors from a distance of 10 feet
R.A.DS.3	Have speakers capable of being heard from 10ft away

Table 20 Arena Electrical Requirements

Requirement	The Arena Electrical System shall…
R.A.E.1	Utilize an AC-DC adapter capable of powering the required DC loads at a high efficiency
R.A.E.2	Contain a DC-DC adapter that converts from the voltage provided by the AC-DC adapter to the required DC voltages at a high efficiency
R.A.E.3	Communicate with the robot subsystem at a rate of at least 30Hz

R.A.E.4	Support a camera for top-down view of the court
R.A.E.5	Support an Embedded Controller capable of running a traditional Operating System
R.A.E.6	Convert voltage from 120V AC to 5V DC
R.A.E.7	Support at least two gamepads
R.A.E.8	Contain sensors to detect when a goal is made

Table 21 Arena Computer Vision Requirements

Requirement	The Arena Computer Vision System shall…
R.A.CV.1	Support vision-based position tracking of the ball and robots in the court with update rate of at least 30 Hz

2.5.4 Game Requirements

The game requirements in Table 22 defines the features and functionality of the game system. Each requirement indicates a particular aspect of the subsystem that must be accomplished in order for the project to be considered successful.

Table 22 Game requirements

Requirement	The Game shall…
R.G.1	Create a 2D visual representation of the Arena and Robot Status
R.G.2	Have a menu to start, pause, and reset a timed match
R.G.3	Display current score and game time
R.G.4	Playback past 10 seconds of gameplay upon a goal
R.G.5	Play a 3D animation of the ball making it into the goal
R.G.6	Perform collision detection between the different objects
R.G.7	Employ software that is fully unit tested
R.G.8	Utilize collision detection to prevent dangerous actions

2.6 Standards

The standards found in Table 23 are relevant engineering standards that can simplify or increase the capabilities of the designs chosen. Utilizing standards results in inter-operability between various systems. It also streamlines decision designs in the event of an available standard that meets requirements.

Table 23 Relevant Standards

Standard	Name/Field
ICS 29.020	Electrical Engineering
ICS 29.060	Electrical wires and cables
ICS 29.100	Components for electrical equipment
IEEE 1872-2015	Standard for Ontologies for Robotics and Automation
IEEE 1012-2016	Standard for System, Software, and Hardware Verification and validation
IEEE/ISO/IEC 29418-2018	International standard – Systems and software engineering – Life cycle processes – Requirements engineering
IEEE 802.15.1	Bluetooth qualification
IEEE/EIA 12207	Life Cycle Process
IEEE 1540	Software Risk Management
IEEE 1471	Recommended Practice for Architectural Description of Software -intensive systems
ISO/IEC 14882	Programming Language C++
Unicode 12.1.0	Unicode standard
ICS 31.020	Electronic components in general
ICS 31.180	Printed circuits and boards
ISO 3833-1977	Road vehicles – Types – Terms and definitions

2.7 Project Research

There are several similar projects that are utilized as inspiration for the design, operation, and requirements for this project.

2.7.1 RoboCup

The Robocup competition introduces a challenge for competitors to develop complex algorithms to enhance the capabilities of robots in sports. There are several academic papers published on the topics of computer vision, control, and robot architecture. A useful solution for tracking robots that was developed for Robocup is the usage of an overhead camera utilizing computer vision to solve the localization and mapping problem. This is discussed in depth in the paper *Tracking a robot using overhead cameras for RoboCup SPL league* [1]*. The camera solution is shown graphically in* Figure 2*.* [1] [2]

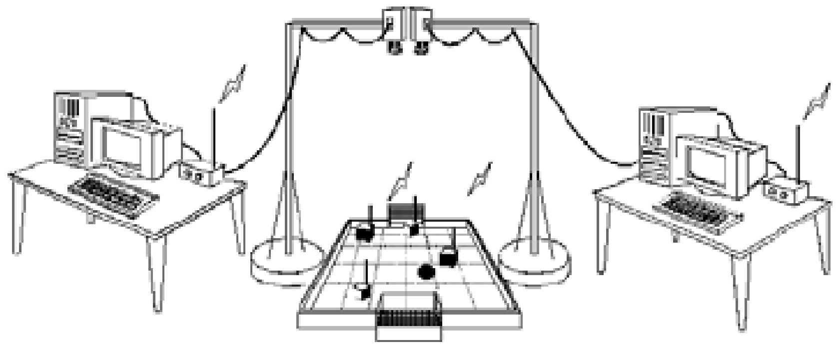


Figure 2 Typical RoboCup arena setup
Pending permission

2.7.2 VEX Robotics

The VEX Robotics platform provides a plethora of cost-effective robotics parts that can be utilized for this project. In addition, the Nothing but Net challenge from 2015 and Turning Point from 2018 involved several unique launching mechanisms and locomotion systems for a basketball-like challenge. The Vex robotics platform is a starting place for the mechanical aspects of the robot. The challenge provides a plethora of designs for launching a ball at different forces and ranges, and an inordinate amount of designs for locomotion in a competitive arena. [3]

2.7.3 Stanford's Battle of the Bots

Stanford's 2015 battle of the bots. This challenge very closely matches the scope and scale of our project. The students developed many unique robots that launch balls in a basketball competition at a very similar scale to the one initially considered for this project. This challenge provides a point of comparison for the scale, size, and capabilities for launching a small tennis ball in a basketball context. [4]

2.8 House of Quality

The house of quality diagram shown in
Figure 3 indicates the relationships between engineering requirements and market requirements. Additionally, the diagram indicates the relationship between different engineering requirements. In summary, some requirements that should be maximized causes an increase in a requirement that should be minimized. For example, increasing the shot accuracy of the project would result in an increased cost of the project.

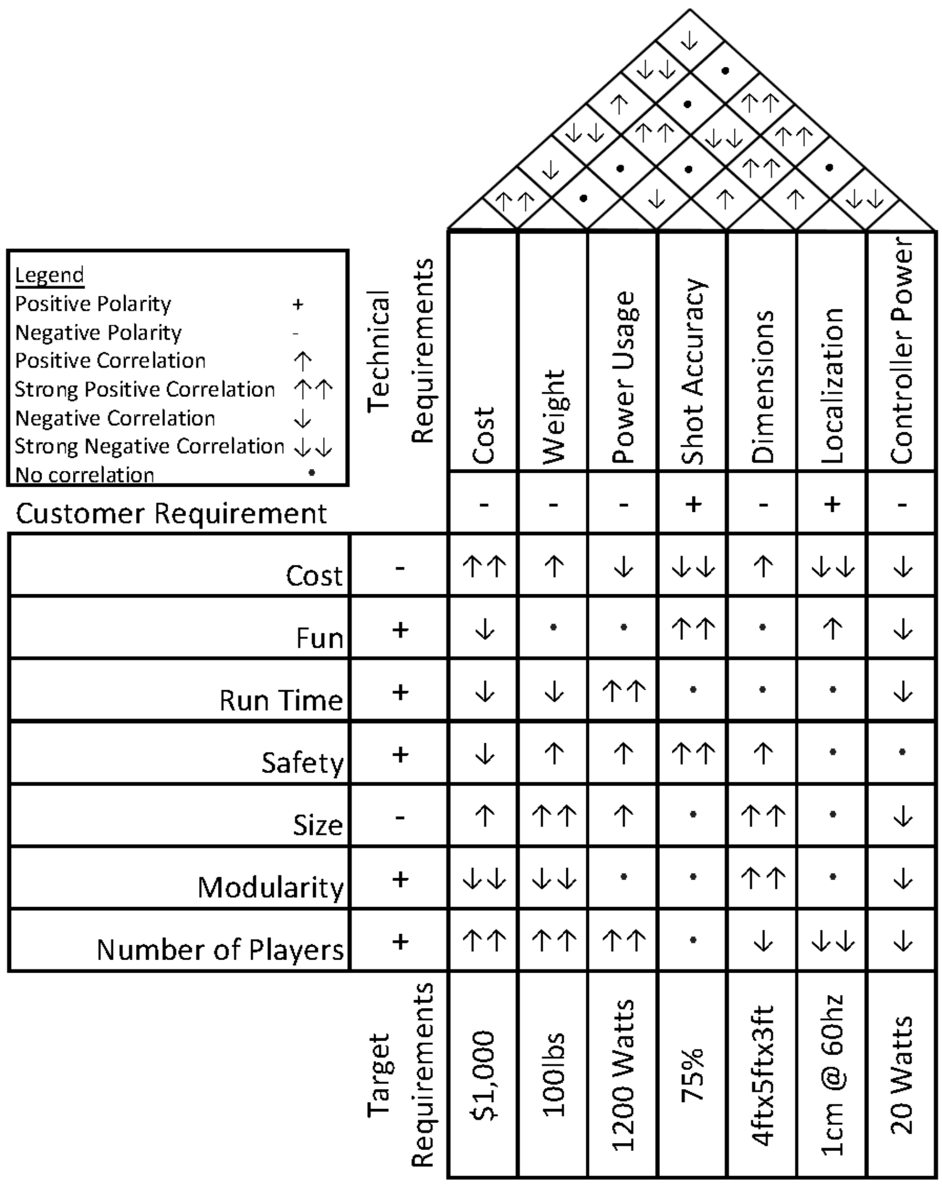


Figure 3 House of Quality

2.9 System Architecture

The system architecture defines the various systems included in the project, and their interactions between one another. The architecture is the highest-level guiding structure for all solutions to the engineering requirement specifications for both hardware and software systems.

2.9.1 System and Interface Identification

The Project is split into three primary systems: Arena, Robot, and Game. The Arena system encompasses all things related to the basketball court, basketball, physical frame structure, and computer vision. The arena contains a control system for high-level planning and control for commands that are sent to the robot system. The Computer vision subsystem is to determine the position and orientation of the robot on the court. Additionally, it must track the position of the ball on the court. The Game System involves taking data in from the player and displaying information such as game and robot status, instant replays, and other high-level functionality. The robot system is the device for physically interacting with the basketball court and basketball. The robot receives commands from the arena control-system and executes them.

The subsystems identified to achieve the requirements are the mobile base, intake, launcher, and control subsystems. These systems are discussed in depth in the subsequent diagrams. There are several critical interfaces identified for this project. These are looked at separately from their own subsystem such that the individual subsystems can be designed independently. However, this introduces risk that the systems are not compatible. Further, interesting behaviors can emerge when complex systems are put together. Thus, these integration systems are fully designed and tested in conjunction with the individual systems to ensure robustness and consistency. The system architecture is shown graphically in Figure 4.

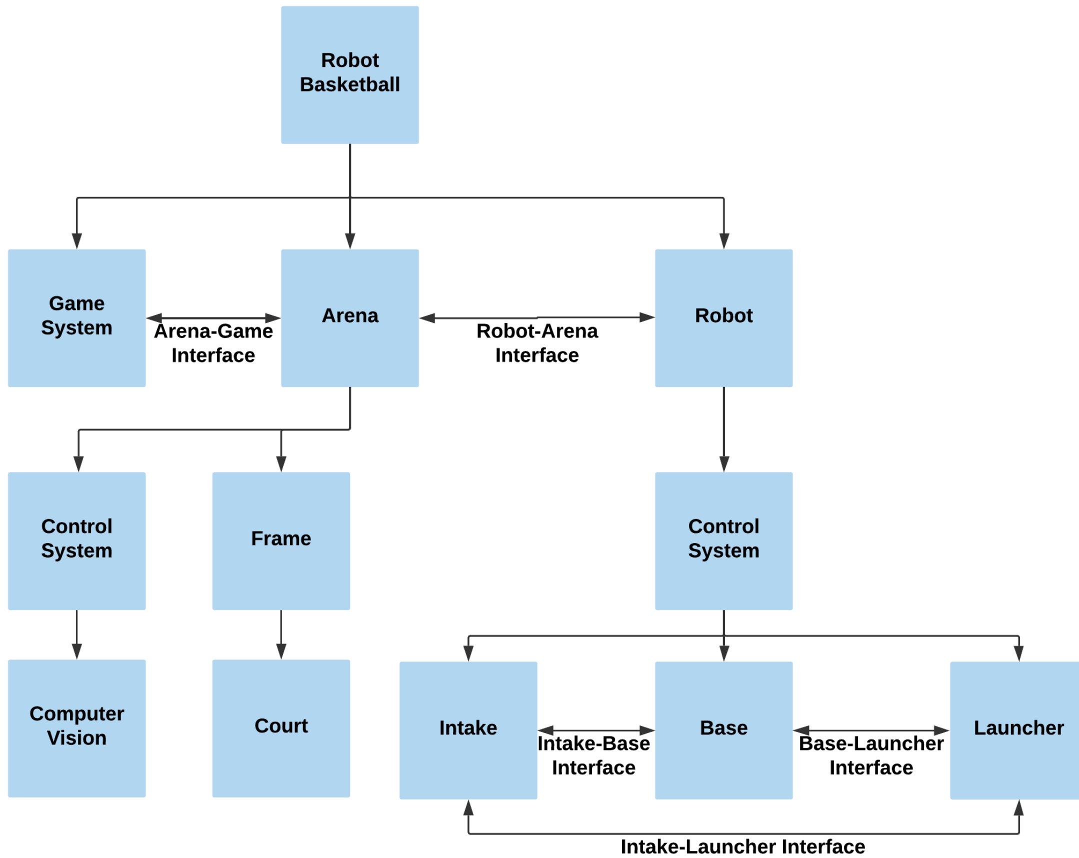


Figure 4 System Hierarchy and Interface Identification

2.9.2 Distributed Architecture

The project is designed and presented as a distributed system. A distributed system is an architecture that contains multiple independent systems that often rely on one another's components. In this case, the robots' responsibilities are separate from that of the arena both physically and computationally. This type of architecture is chosen due to the possibility of scaling the system to a larger number of robots without significantly increasing costs. Dozens of the robots could be built and the arena could be scaled up to a larger size, and the arena cost would remain the same as a single robot cost. The robot is treated as a *slave device* that does minimal processing. The higher-level control systems, computer vision, and hardware are handled by the *master device* (arena). This reduces cost and complexity for the robot by eliminating the need for a camera and a high-power processor. The arena can have increased complexity without significantly changing the system by only replacing a single arena device as the number of arenas or size of arenas increase. The distributed architecture diagram is shown in Figure 5.

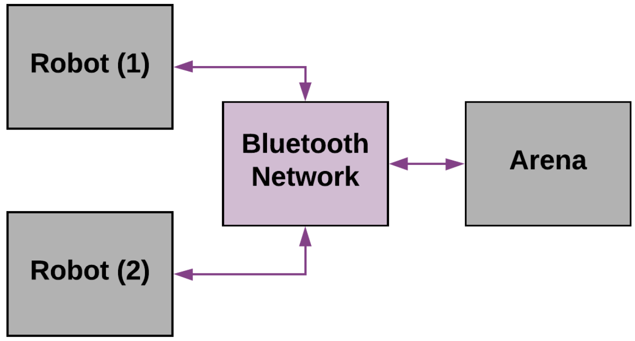


Figure 5 System Communication Diagram

2.9.3 Robot Control Architecture

A robot architecture defines how data should flow such that the robot can effectively interact with its environment. The architecture introduces constraints that drive the design and development of a robotic system. A deliberative robot architecture is chosen for this project because it provides a robust solution to systems that operate in a well-defined space. Due to the nature of the project, most variables related to the operating conditions of the robot such as the number of objects, color of objects, speeds and behavior of objects can be adjusted such that the robot performs adequately under the conditions provided. The general approach to this architecture is to take in data from peripheral devices such as encoders, computer vision, and a-priori knowledge to construct a virtual model that is then deliberated over to plan and act according to a set of pre-defined rules. The architecture is shown graphically in Figure 6.

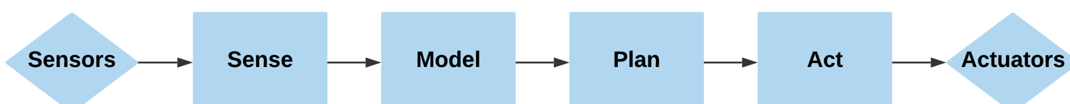


Figure 6 Deliberative Robot Architecture
Introduced by Rodney A. Brooks in
"A Robust Layered Control System" For a Mobile Robot at MIT in 1985

3.0 Robot

The robot subsystem is comprised of all the components required to pick up and launch a ball from different places on the court. The diagram in
Figure 7 denotes the primary systems and their various connections to other systems.

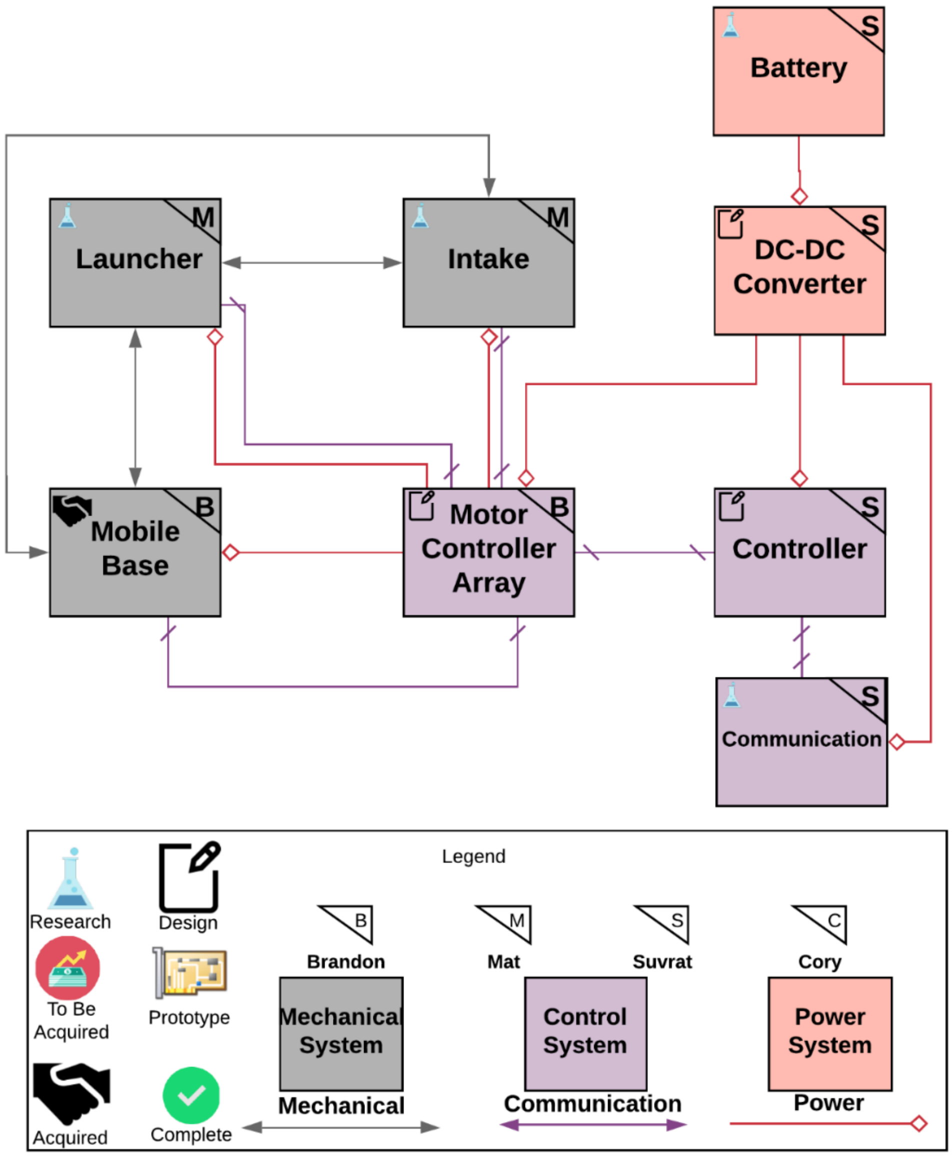


Figure 7 Robot Subsystem Power and Signal Diagram

3.1 Base

The mobile robot base is the locomotion piece of the system. It is the sole source of robot movement on the court. The mobile base is to be fast and agile in order to increase player engagement. If the robot is slow, the player will feel like they are not in control of the robot's actions, and they are not excited to play the game. If the robot is fast and agile, the player can perform complicated maneuvers and make exciting plays. The player experience is also significantly enhanced if the robot does not need to turn significantly to move around and shoot the ball. This way, the player can focus on moving the robot to specific positions and not worry about if the robot can rotate and shoot from that position.

The base platform serves as the main structure for the other subsystems. The Intake and Launcher must seamlessly integrate with the base to ensure robustness and consistency. For example, there must be space for the launcher to extend and retract, and the intake must be able to mount and reach the ball on the ground without interrupting the intaking motions. In the likely event of collisions between robots or between the robot and the arena, the base must be sturdy and stable. The electronics on the robot also must remain safe throughout various operating conditions, and they should be secure and resilient to impacts. The drive system should also be relatively low power to lengthen run-time, as most of the power in the robot is designated to the subsystem. Finally, the robots are generally the focal points of the entire project, thus they should appear both professional and exciting.

3.1.1 Research

3.1.1a 3-Wheel Holonomic

The 3-wheel design has omni-wheels mounted at 60-degree angles to one another. This allows for full holonomic motion with only three motors. There is power loss driving in cartesian directions because only two of the motors are contributing to the motion. An example of an available 3-wheel holonomic kit is shown in Figure 8.

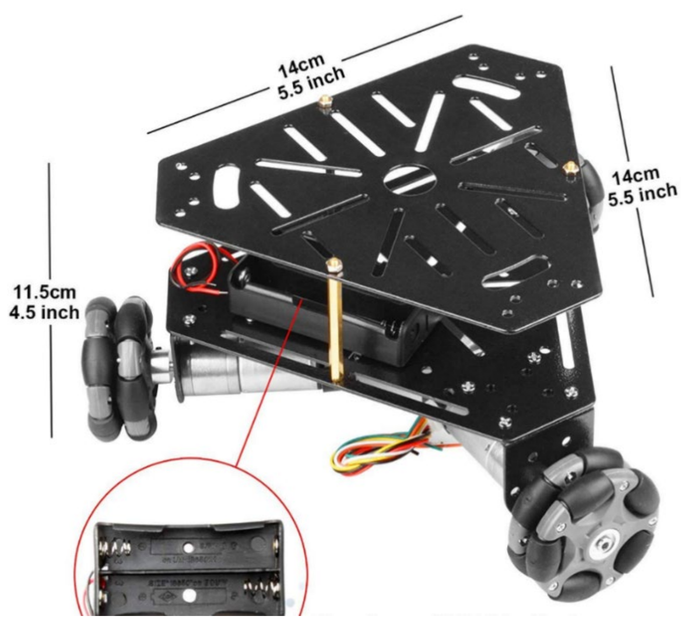


Figure 8 Example Omni-wheel base
Pending permission from Heneng

3.1.1b 4-Wheel Holonomic

The 4-Wheel Holonomic design is the same in principle as the design discussed in 3.1.1a 3-Wheel Holonomic. However, instead of three wheels at 60-degree angles, there are 4 wheels mounted at 45-degree angles. There is significantly more power in this design than in the three-wheel design because all four wheels are contributing to the motion at any given time. Additionally, the output speed is faster than the actual wheel rpm due to vector multiplication at the cost of torque. An example 4-Wheel holonomic kit is shown in Figure 9.

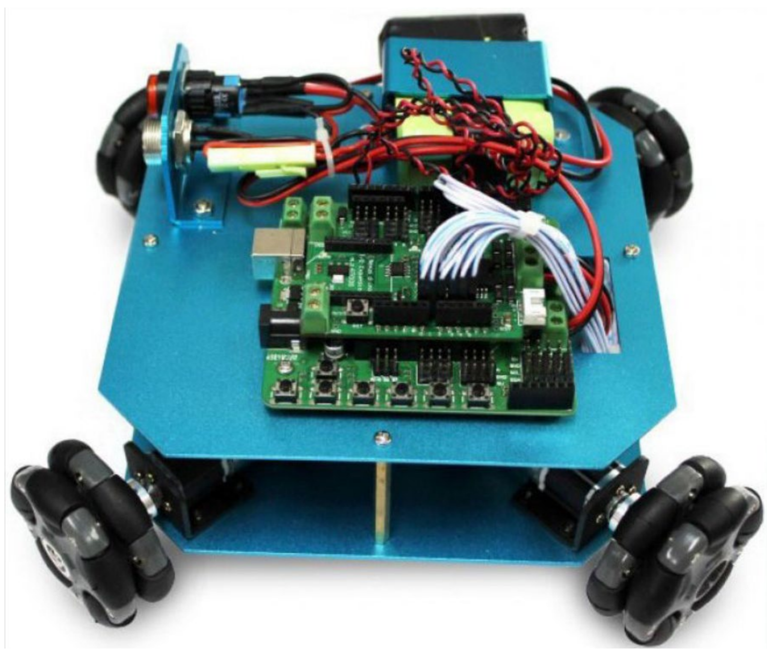


Figure 9 4-Wheel Omni Kit
Pending Permission from Robotshop

3.1.1c Differential Drive

The differential drive design is a traditional approach to mobile robotic bases. This design generally involves 2 to 4 wheels mounted square to the base. Either two or four of the wheels contributed to the motion of the drive. This design is very robust and provides significant power, however it is not holonomic. This base requires turning of the entire robot to drive in directions that are not forward or backward. The wheels are not required to be Omni-directional, thus traditional wheels or treads could be utilized. In order to achieve the requirements, an additional mechanism for turning the launcher and/or intake would be required. This would ultimately achieve the same thing as the holonomic motion regarding launch angle, but it reduces the agility of the robot and ultimately the player engagement. An available differential robot kit is shown in Figure 10.

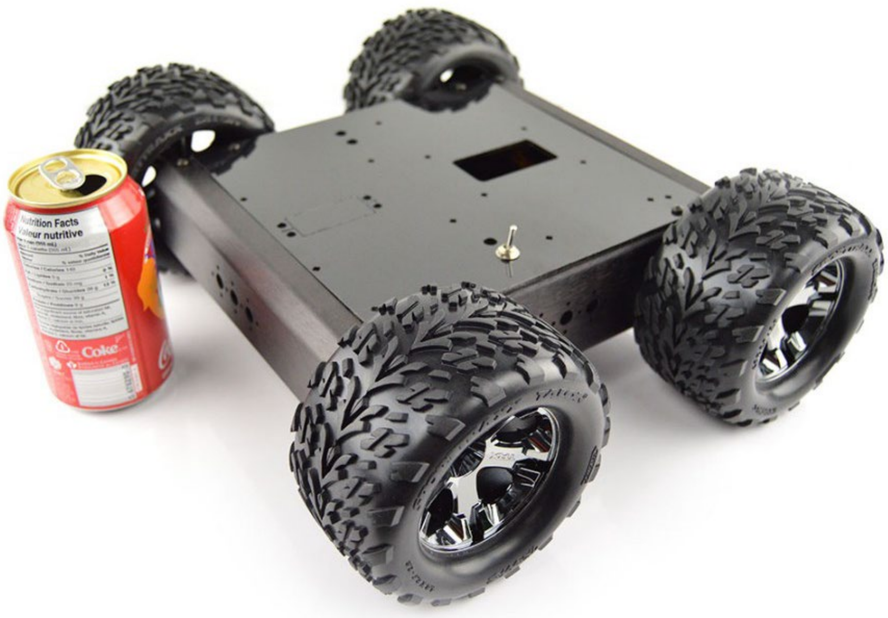


Figure 10 Differential Drive Robot
Pending Permission from RobotShop

3.1.1d Actuators

There are many available actuators with a variety of parameters that distinguish the different products. The actuators cost, RPM, voltage, current, power, control, and feedback types are the parameters that directly impact design decisions. A summary of the devices investigated in detail is shown in Table 24.

Table 24 Actuator Comparison

Actuator	Cost	RPM	Voltage	Current	Power	Control	Feedback
Heneng DC Motor	$15	100	9V	1.2A	10.8W	External MC	2 CPR Quadrature Encoder
Feedback 360 High Speed Continuous Rotation Servo	$28	140	6V	1.2A	7.2 W	PWM	2 CPR Hall Effect
High Speed Continuous	$17	180	7.4V	.6A	4.44W	PWM	None

Rotation Servo							

3.1.1e Wheels

There are many wheels available to choose from, each with a variety of properties that affect design decisions. The wheel type, cost, size, and material are the main factors investigated for this project. A summary of the products investigated is shown in Table 25.

Table 25 Wheel Comparison

Wheel	Type	Cost	Size	Material
Robotshop Omni	Omni-Wheel	$15	60mm	Aluminum + Rubber
Lego Omni	Omni-Wheel	$7.60	58mm	Plastic
UniHobby Omni	Omni-Wheel	$15	38mm	Plastic
Micnaron Luggage Wheel	Standard	$10	60mm	Rubber

3.1.1f Frame Materials

The frame is a critical component in the base subsystem, and a huge selection of materials are available to achieve the requirements defined for the project. The parameters investigated are cost, modularity, strength, and manufacturability. The modularity property indicates how easy it is to adjust, modify, or change the design of the frame given designs of other subsystems. The strength is the sturdiness of the material. Manufacturability is how easy the material is to work with given the tools available. A summary of the investigated materials is shown in Table 26.

Table 26 Material Comparison

Material	Cost	Modularity	Strength	Manufacturability
Wood	Low	High	Medium	High
Aluminum	High	Low	High	Low
Plastic	Low	Medium	Low	Medium

3.1.2 Design

The researched designs are summarized in Table 27. Based on this information, the design chosen is the 4-wheel holonomic design. The design provides maximum usage of the power available in the motors and provides better mounting places for the launcher and intake systems. However, it is more expensive and there are no low-cost kits available.

The design chosen is a mash-up of custom parts fabricated by the team, and pre-existing components. The motor/encoder combination is to be a continuous rotation servo. The continuous rotation servo is like a DC-motor except that it has built-in open-loop position control and motor driver. This substantially reduces the complexity of the PCB required for the robot. Standard servo mounting plates are used to interface the servo with the frame. The best servo considering long-term goals is the Parallax High-Speed Continuous Rotation servo with feedback shown in Figure 11. This servo for $27 provides up to 160RPM with high torque and accurate position control. Although this is more expensive, it provides a way to close the control loop to improve base performance. The servos are mounted asymmetrically to allow for the wheel to be in the center of the hexagonal side, and to allow a channel underneath the robot to allow space for cuts and mounting of the launcher and intake. The final design is shown in Figure 13. The holonomic motion is described in Figure 14.

The main frame piece for the design is a wooden plate cut on a laser cutter to quickly and accuracy cut out all the holes for the various hardware, and the cut-open sections that give space for the intake and launcher systems. It is also possible to utilize traditional tools such as a jigsaw and drill to build the design with enough tolerances.

The chosen Omni-wheels shown in Figure 12 are 60mm in diameter and are a mixture of aluminum and rubber. They are purchased from Robot-Shop for $15 each. This is the cheapest omni-wheel at this size. The size is chosen because it is just large enough to allow the ball to roll underneath given the wheel mounted directly center of the plate. Additional clearance is given by mounting it directly to the servo which is underneath the wooden frame. The drive servo directions indicated in
Figure 14 show how the frame successfully achieves the holonomic requirements in each cartesian direction and both rotations.

Table 27 Base Design Comparison

Design	Wheels	Motors	Speed	Cost	Agility	Strength	Modularity
3-Wheel Holonomic	3	3	Low	Med	Med	Low	Low
4-Wheel Holonomic	4	4	Med	High	High	Med	Med
Differential Drive	2-4	2-4	Med	Low	Low	High	High

Figure 11 Parallax Feedback 360 Degrees High Speed Servo Pending Permission from Parallax

Figure 12 60mm Omni wheel Pending Permission from RobotShop

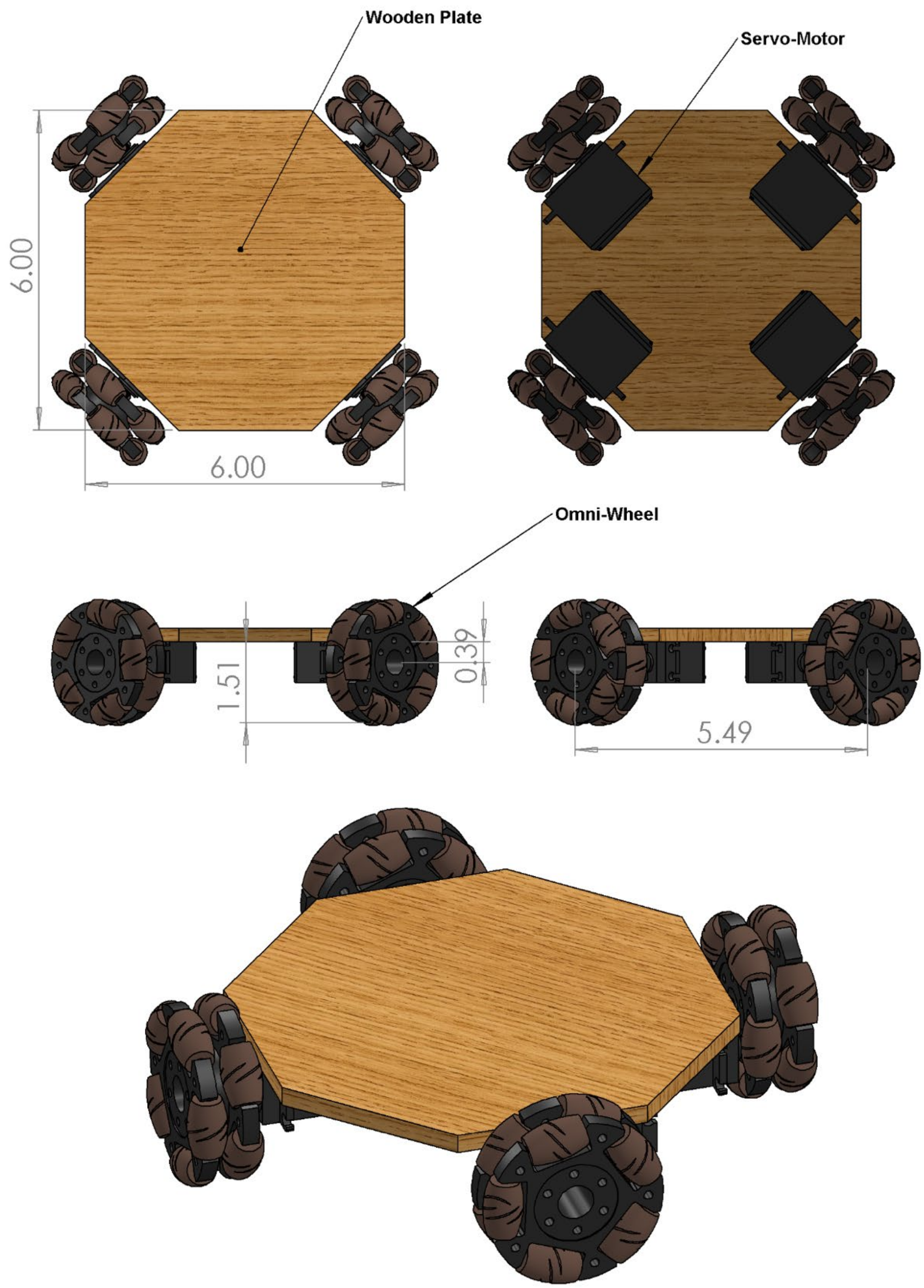


Figure 13 Robot Base Design

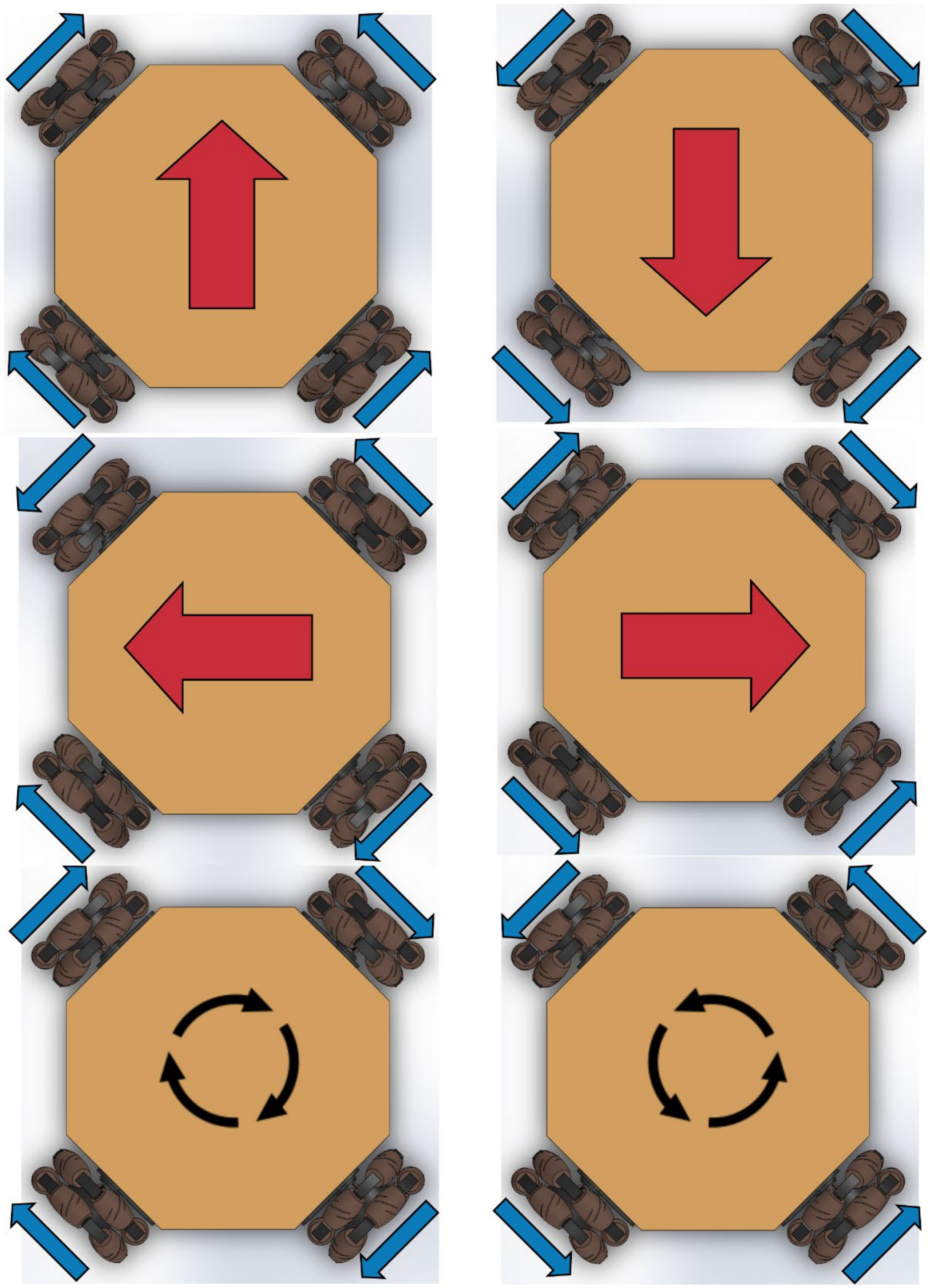


Figure 14 4-Wheel Holonomic Drive Configuration

3.1.3 Prototyping and Testing

The prototyping can be accomplished with a simple wooden plank of an appropriate dimension that is cut by a jigsaw or hacksaw and drilled appropriately. Once tested, a more accurate, tolerance-sensitive version can be manufactured on a laser cutter. The electronics can be individually bench-tested utilizing a servo driver, power supply, and Arduino. The Servos and wheels can be purchased directly from their respective manufacturers.

The tests in Table 28 indicate the various tests required to evaluate the performance and capabilities of the Base design. Each test corresponds to a requirement or constraint. The equipment required to adequately complete the test is also determined such that the equipment can be acquired prior to manufacturing.

Table 28 Base Tests

Requirement	Test	Required Equipment
R.R.B.4	Determine if the base traverse the court without slipping	Court, rope
R.R.B.1	Determine if the base drives forward, backward, left, right, and rotates in both directions	Arduino, long USB cable, windows laptop
R.R.4	Determine if the base plate is sturdy enough to support the additional weights of the other subsystems	Weights
R.R.4	Determine if the base is heavy enough to support a moment about the expected launching axis	Weights
R.R.4	Determine if the base moves in all directions when additional load is added	Arduino, long USB cable, windows laptop,
R.R.5	Determine if the base has enough height for the ball to roll underneath on the side that the intake is mounted to	Ball
R.R.5	Determine if the base has low enough height to block the ball from rolling under on the sides that the intake is not mounted	Ball
R.R.4	Determine if the robot remains active after an impact	Rubber Mallet

3.2 Launcher

The launcher on the robot must be able to shoot the ball from anywhere on the court being played on. In order to accomplish this, the launching mechanism must be adjustable in some way, shape or form. This feat can be accomplished in a multitude of ways, however, to make it an achievable goal, the team narrowed the possible designs down to two ways: either lock the angle and have variable force or lock the force and adjust the angle. These paths require different solutions and steps to be able to work properly, and the same type of mechanism may not work for both, or either of the ways chosen by the team and can influence other design choices. With a fixed angle, the force of the mechanism must be able to be easily and reliably changed. This makes the overall mechanism more complicated because more parts are required to make the launcher behave in the intended manner. A fixed force and variable angle bring up a different set of problems, such as the platform the launcher rests on will need to be more complicated instead of the launcher itself, and the equations become more complicated due to the changing height at each point of launch. Another point the team must keep in mind is that due to the steeper angle that would be required at some points on the field, the ceiling must be higher than it would be with a fixed angle. As previously mentioned, this would have an influence on the size and weight of the field, which has the potential to clash with our field requirements. The three main ways of implementing a launcher on the robot being explored are a flywheel, puncher, and catapult. These three methods were chosen because most of the ways to launch the ball reasonably will fit into one of these categories and the team can narrow it down more easily within the category before deciding which type overall to use.

3.2.1 Research

3.2.1a Flywheel

There are two main ways to implement a flywheel launching mechanism, using one or two wheels. Both offer their own specific problems that must be considered when doing calculations for the projectile coming out of the launcher. These situations are outlined in Table 10 below.

Table 29 Flywheel design problems

Flywheel problems	**Outcomes**
Wheel not up to full speed before shot	Shot comes out short
Ball enters wheel at different speed every shot	Shot is either short or long depending on speed and is hard to track and correct
Ball hits different part of wheel (isn't compressed as much or compressed more)	Length of shot is once again affected. Could also put a different spin on the ball
Wheels are not spinning at same speed (double flywheel specific)	Curve is put on the ball. This could also potentially change every time the ball is fired.

All these situations boil down to a flywheel just being too unpredictable at any given time. There are ways to remedy these problems, such as finding ways to finely control the speed of the ball entering the wheel, making sure the channel the ball follows into the launcher is a tight fit for the ball to disallow the ball to enter the wheel from a different angle each shot. The solutions to many of the problems presented by a flywheel are mechanical in nature and are something that the team isn't built to implement well. Something that can be looked at positively about using a flywheel, however, is that it will allow the robot to put a more natural spin on the ball compared to the other options under consideration by the team. Since a huge part of basketball is getting spin on the ball to help make shots off the backboard, this is a rather good thing to be able to do. The flywheel design also would easily be able to fulfill our requirements of varying force, by adjusting the velocity the wheel spins at, and the ability to fix the angle that the ball is launched at easily. This could be done the other way around rather easily as well.

Comparing the two types of flywheels, one or two-wheel, both have their own advantages as well. A one-wheel flywheel will take up less space overall but won't be able to put out the same force as a two-wheel flywheel using the same motors. Also, due to having only a single motor, the one-wheel flywheel solution would require less power to operate as well as have an overall simpler design to implement. The two-wheel flywheel would allow for more finely tuned spin on the ball and more overall launching power. However, the extra motor would need extra consideration as it adds more weight to the robot in the form of extra parts needed to hold and support the extra motor and removes space needed to implement other systems on the robot. Depending on the parts chosen, this could put unnecessary strain on the base and could affect how the base is constructed. The two-wheel variant of the flywheel also has a greater chance of failing due to the extra wheel. This would require careful monitoring of more variables than the single wheel method as any sort of disharmony between the speed or angle of the two wheels essentially make the calculations done by the other systems of the project useless as the real-life motion of the ball wouldn't be able to match the projected numbers. Overall, the flywheel method's variability is both its biggest strength and weakness, in the form of being flexible enough to meet the team's launcher requirements whichever way ultimately is chosen while being unreliable in accuracy and precision needed for this task.

3.2.1b Puncher

A punching mechanism is a lot more straightforward than either a flywheel or catapult design. With a puncher there is a lot more control possible with it because the ball is always launched from the same spot and orientation every time. The first major downside of a puncher is that in order to make the force of it variable is to more hardware is required. If the team was going to make a fixed force mechanism for a shooter, the puncher would excel at that as it could be solved with a mechanism such as a skip gear. However, due to needing to meet the requirement of a variable force on the ball, an additional mechanism such as a linkage, actuator, or even another motor, would be required to release the puncher.

The puncher design currently being considered will be a tension-based design powered by springs either extended or compressed with a sudden release. The spot that the puncher contacts the ball and the shape of the punch can be changed to produce different effects on the ball. Some examples of this are using a wedge and hitting low to produce a chip shot effect or to hit the ball as close to the center as possible to get little to no spin. As the puncher and rail can be attached at basically any angle and won't need to move, the team can experiment easily and find the best angle to use before locking the angle in place to fulfil our requirement of having a fixed angle, variable force launcher. Due to the puncher traveling in a straight line and only acting a short impulse upon the ball, the calculations end up being projectile motion equations.

The main pros and cons of the puncher are outlined below in Table 16. A huge con of the puncher design is the space required to implement it correctly. First, even though the slide component may look compact, it needs to be able to extend a certain amount outside of its at rest position, this size change can range from very little, like half an inch, to having to take up double the size of the initial position. Second, the extra component that would be needed in order to remove the gear from the slide to trigger the launch would have to include another motor or drive mechanism which also essentially doubles the space needed for the full system. However, if the puncher only needs to have a consistent force, the second part of the size requirement is removed, and it is only necessary to worry about the range of motion of the slide component.

Table 30 Pros and cons of a puncher

Pros	Cons
Consistent launch	Not easily converted to variable force
Consistent force	Spring/elastic mechanism can wear down
Angle easily changed	Large

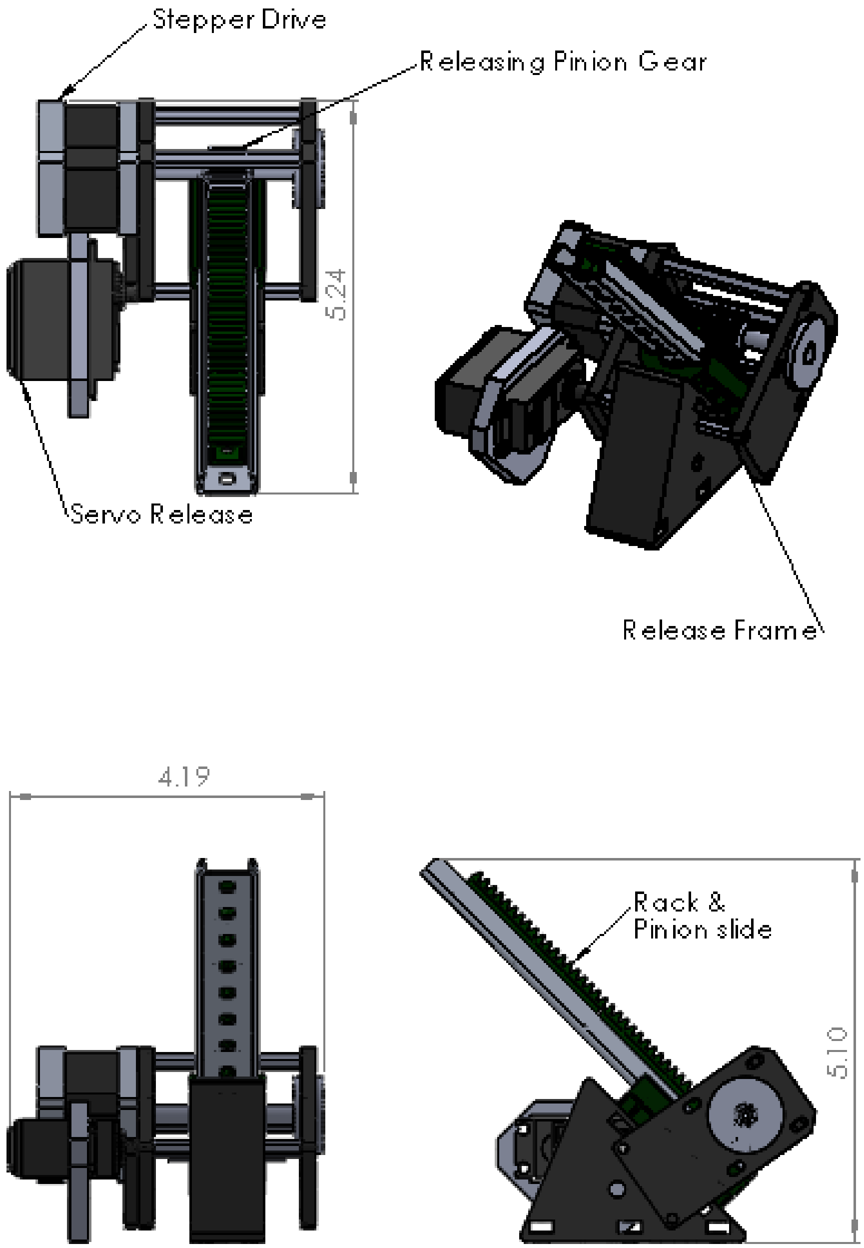


Figure 15 Launcher Design

3.2.1c Catapult

There are three main types of catapults, the ballista, the mangonel and the trebuchet. Since the construction of a trebuchet device would be unfeasible due to the complication of the design and the size constraint of our small robot, that idea was only very briefly explored. The ballista variant would be very similar in design to the puncher mechanism described above in section 3.2.2b, except for the fact that the ball would be pushed down the length of rail instead of a short, sharp contact to propel the ball. The ballista design shares a lot of the same advantages and disadvantages as the puncher except for being able to control the spin of the ball as it is launched. And in the implementation that would be used for this robot, the only difference between the ballista design and the puncher design being considered is a stopper that keeps the ball from falling into the channel left behind when the spring is drawn back.

The last type of catapult is the mangonel, which is what most people think of when they think of the word catapult. Using this design poses a lot of design problems. First, we would need to have a bigger and more complicated intake or put it in a place on the robot that doesn't make sense in order to load the arm of the catapult. Second, there would be little control over the angle unless the placement of the beam to act as a brake for the arm was very precise. Due to this, if the team was to try to make the robot have a variable launch angle, this design would immediately become unable to use as it would be difficult to get the correct placement dynamically on such a small-scale base. The team would also have to take special care to make sure that the arm was able to be fully drawn back, or at least drawn pack to a specific spot to be able to vary the force. The calculations for aiming the catapult and getting the correct drawback on the arm are more calculated than the relatively easier impulse and standard projectile motion formulas useable with something like the puncher.

3.2.2 Design

The design being. For all that the cons that it can potentially have, the team has decided that they are relatively easier to mitigate than having to design an entire separate mechanism that would be required to get the correct variable force behavior that is needed for the robot. The flywheel will be almost centered on the robot, slightly more towards the front side. The flywheel will be sunken into the robot so that it is close to the ground so that it will be able to function as the intake into the robot as well. The ball will travel around the wheel until the correct angle is for launch is reached and the ball is shot from the front of the robot. The wheel will be direct driven by a brushless motor. If the motor turns out to be much higher powered than what is required, it will have to be geared/chained down to burn off some of the speed, adding more complexity and pieces which could break down. The motor powering the wheel will have a Pololu Magnetic Encoder (Pololu Part #3499). This encoder provides 20 counts per revolution that allow the software to effectively track and alter the speed in order to make sure the flywheel is being spun at consistently the correct speed for the distance the robot is from the hoop.

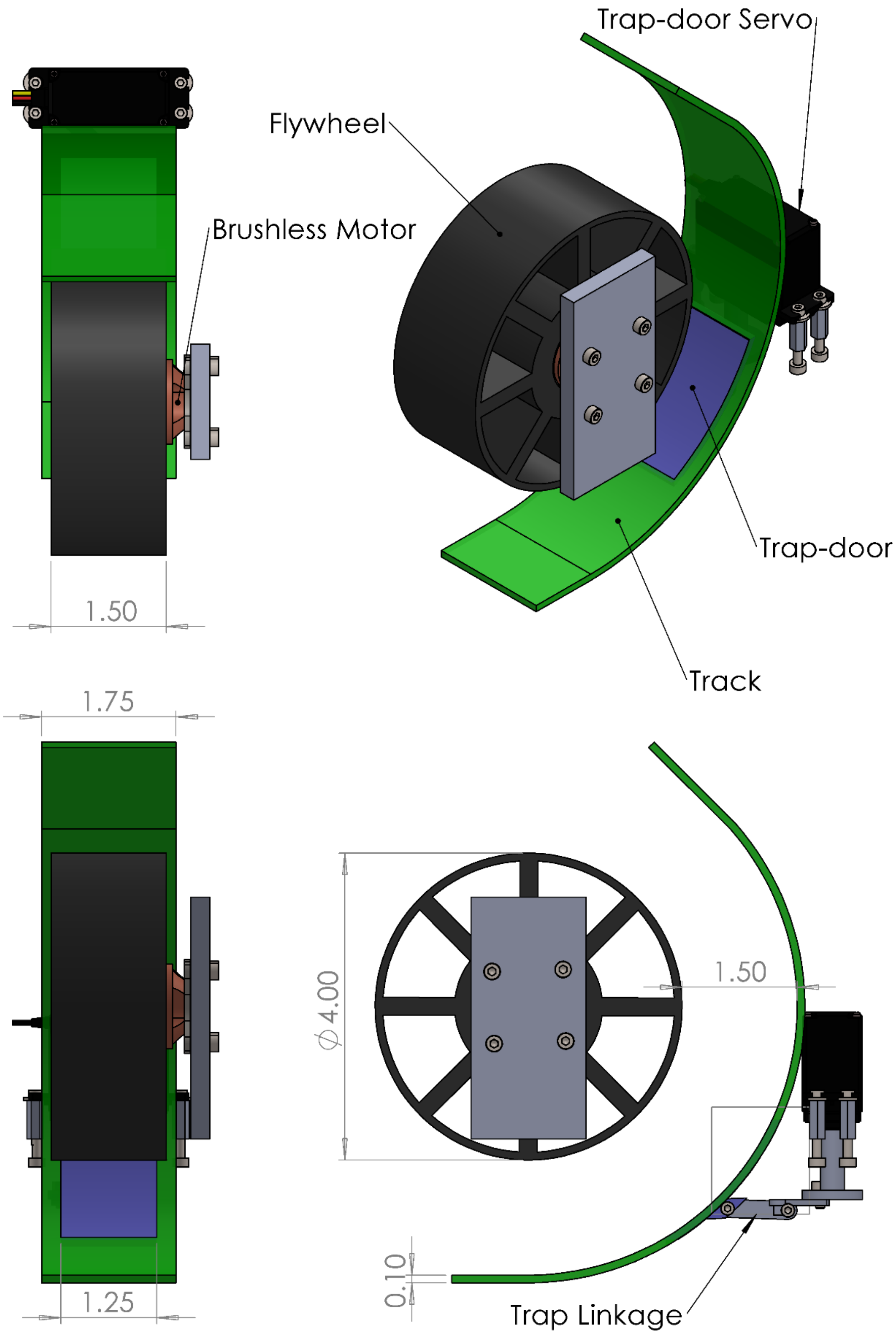


Figure 16 Launcher design drawing

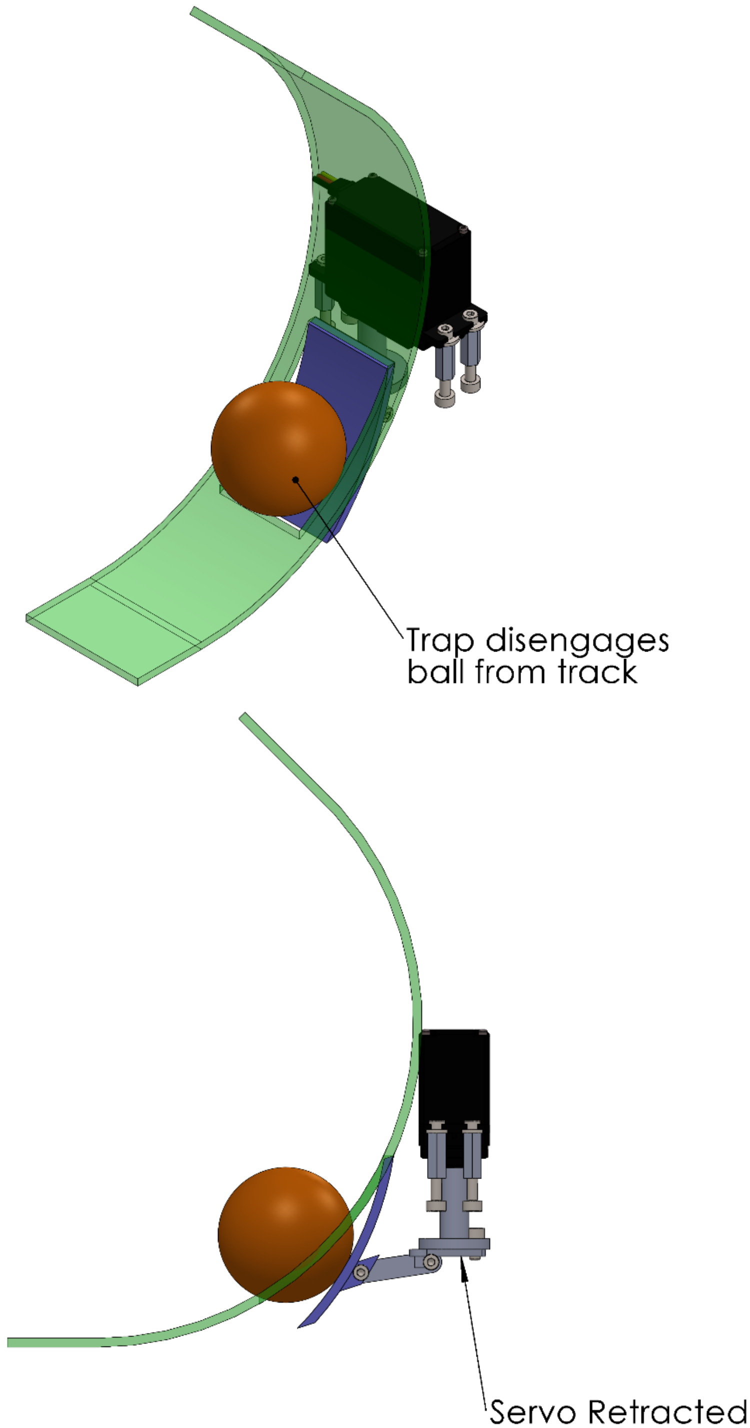


Figure 17 Ball-Trap interaction drawing

3.2.3 Prototyping and Testing

The launching mechanism that is going to be used is a single flywheel device. The team chose this after spending a large amount of time attempting to piece together a variable force spring mechanism which ended up being more complicated than what was initially thought. Testing for this mechanism will be conducted in stages, starting with force and making sure it is consistent before trying with different angles. Although there will not be varying angle capability included in the final design, it is important to test the angles in order to find the optimal one that uses less power and to make sure the path of the shot ball is contained within the arena that has been built for the robot. The angle will be controlled by a piece of material attached at the end of the track that the wheel slingshots the ball around.

Table 31 Launcher Tests

Requirement	Test	Required Equipment
R.R.L.3	Test the launcher with different forces. Determine distance.	Tape measure, carbon paper
R.R.L.3	Test the launcher with different angles. Determine distance	Tape measure, carbon paper
R.R.L.3	Test the launcher for accuracy and precision at different shooting configurations	Tape measure Carbon paper
R.R.2	Test if launcher resets properly between shots	
R.R.L.2	Check if the ball is hit consistently in the same area	
	Measure voltage and current draw across subsystem	

3.3 Intake

The intake for the robot must be able to pick up a ball and transfer it to the launcher mechanism. There are both passive and active options to pick up a ball that the team has explored. Passive solutions require no power, or significantly less power than active solutions, however, there is a higher chance for them to not consistently pick up the ball. Options researched for our intake mechanism include a telescopic lift, a conveyor belt, or a wheel-based design. This mechanism would place the ball directly into the spot it will be launched from. It's important that the ball is deposited in the same spot each time because that has a direct impact on the accuracy and consistency of the launcher due to the puncher having to hit the same spot on the ball each time. The team has narrowed the decision down to a series of wheels, a conveyor belt, and a telescopic lift like what is seen on a forklift.

3.3.1 Research

3.3.1a Wheels

The first design being considered, as well as the first of the two active intake mechanisms is a wheel-based mechanism to pick up the ball and pass it up the intake. Wheels for the intake can be done in two ways, either on one or both sides of a channel, much like a single or double flywheel design except with a lot less power. Wheels are more useful for the intake than for the launcher because less precision is required. The design and calculations for the intake don't depend on something as small as making sure the ball comes in at the same speed every time. Since all that is required is to get the ball to the launcher, using wheels is necessary. A wheel-based intake mechanism would most likely require the most hardware out of all the designs being considered as it would require more than one motor to implement. The wheel design the team is looking at is essentially a conveyor belt without the belt and the only major drawback besides the aspect of having to utilize more hardware is that if the wheels aren't placed in the right position the ball could get stuck between them or not move quick enough. Due to each wheel needing to be mounted individually, there is also more potential for a part to fail taking down the entire mechanism. The front of the wheels act as an active intake by spinning to physically pull in the balls, instead of just corralling the ball.

3.3.1b Conveyor Belt

The second active design being considered is a conveyor belt. There are only two versions of the conveyor belt that can be implemented for the robot. One with, and one without dividers in it. The only real distinction is that the one with tabs will have a more redundant mechanism for carrying the ball to the launcher. A conveyor belt can be implemented with a single motor potentially which makes it lightweight. The major failing point of using a conveyor design is that it must always be kept taut which requires a lot of attention and regular maintenance. If the conveyor belt isn't fully taut, the ball has the potential to just spin in place, which can be combated with plastic tabs that sweep the ball and act as a floor to prevent them from falling. Adding this to the conveyor belt doesn't come at the cost of too much hardware and extra weight typically. The two primary materials that the conveyor belt can be made from are either a smooth, continuous band or plastic links that look like tank tread. The tread design will allow the team to more easily. The conveyor belt is very similar to the wheel design in the fact that the front of the conveyor belt actively works to bring in the ball.

3.3.1c Telescopic Lift

The telescopic lift design is the only design being considered by the team that can be considered passive, as the end that contacts the ball would be like the fork on a forklift. The upside of this is that the fork part is simple to design and can be made from just about anything. It also has the perk of not being an active part that can break down and therefore must be replaced. The downside of telescopic lift is that the part that grabs the ball is passive. With a passive grabber there is a high likely hood of having to trap a ball in the corner to be able to pick the ball up. The inability to consistently pick up the ball is a huge detriment overall as it potentially leads to a huge loss of time in the game. The lift would be powered with either 1 or 2 motors attached to pulleys that would pull the different stages up. The lift could either go straight up or at an angle. To be able to drop the ball into the launcher at the top of the lift and to more securely hold the ball, a slight angle on the lift would be more beneficial than if it was perpendicular to the ground and base of the robot.

3.3.2 Design

The design that will be used is a dual functionality mechanism in the form of using the flywheel from the launcher. The flywheel will be placed low to the ground so that it can contact and intake the ball correctly. Just after the ball is taken in from the ground, it will enter a trapdoor-like mechanism that will keep the ball from being in contact with the wheel. This will allow the player to hold onto the ball until they want to shoot. When the signal to shoot is given, a servo or similar piece of hardware will be used to push the ball back into contact with the wheel, transforming it into the launching mechanism. When the ball is being picked up, the wheel will spin at a much lower RPM than when being shot, this will allow better control and put less stress on the trapdoor mechanism being used.

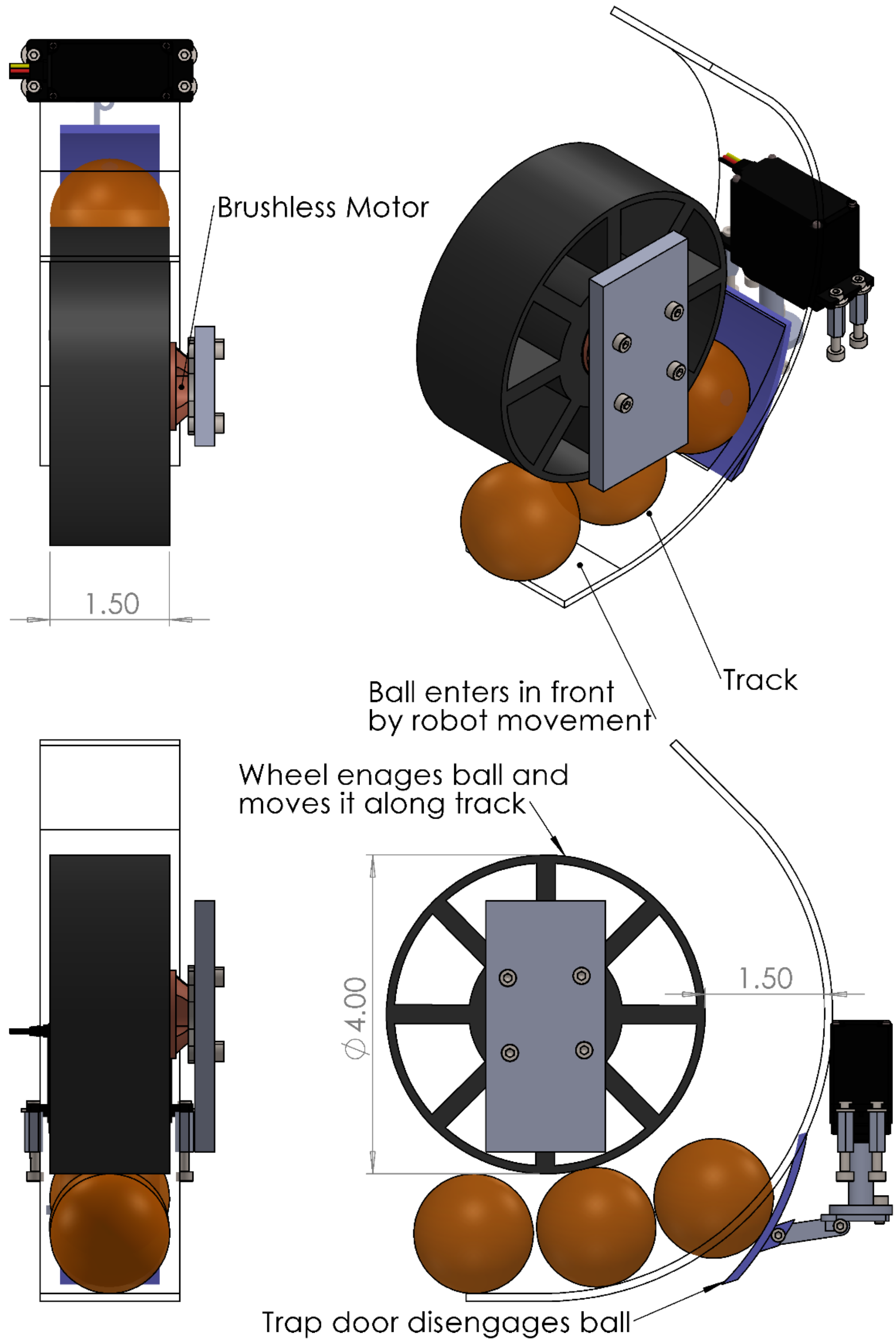


Figure 18 Intake Design

3.3.3 Prototyping and Testing

The intake will be prototyped and tested the same way that the launcher will be; first using premade parts and then getting them manufactured. As for actually carrying out the tests, until the intake is able to be mounted to the base of the robot, it will have to be hand moved to cover the tests that require the intake to be moving. The team will also be looking at the speed and consistency of the intake mechanism to determine what must be tweaked in order to make it better overall.

Table 32 Intake Tests

Requirement	Test	Required Equipment
R.R.I.2	Test if the intake can pick up a ball from different angles	Ball
R.R.I.2	Test the intake moving and pick up a stationary ball	Ball
R.R.I.2	Test the intake stationary and pick up a moving ball	Ball
R.R.I.2	Test the intake with both intake and ball moving	Ball
R.R.E.4	Measure voltage and current draw across subsystem	Multimeter

3.4 Actuator Control Array

The actuator array block exists primarily to interface the various actuator components of the Launcher, Base, and Intake systems to the electrical systems of the robot. This includes routing the signal parameters from the microcontroller to the motor controller, and routing power and ground to each device. The motor drivers for each of the drive motors exist within the servo itself, thus this component simply routes power and signal appropriately – There are no additional integrated circuits required. The intake and launcher systems are integrated into the same device, thus only a signal motor controller and servo controller port are required. The launcher motor is a DC brushless motor that requires an electronic speed controller to control. Thus, that device is investigated fully below. Additionally, components to simplify the control loop or servo control generation are also investigated.

3.4.1 Research

3.4.1a PCA9685 – I2C to PWM

This device is a I^2C to PWM IC. It can drive up to 16 PWM channels at once with 12-bit resolution at a fixed frequency. This can be used in conjunction with the chosen motor controller to reduce load on the chosen microcontroller. This also simplifies the motor control process. Additionally, between this device and a voltage regulator, several servos can be controlled without significant overhead.

3.4.1b TLC6C5912GQPWRQ1 - PWM Generator

3.4.1c 30A speed controller – Electronic Speed Controller

3.4.1d Hobbypower Rc ESC 10a Brushed Motor Speed Controller

3.4.2 Design

The motor controller array block consists of 4 L298P devices and a PCA9685. The PCA9685 acts as an I/O device that takes in inputs from the microcontroller and generates PWM signals to control the motors, stepper, and servo for mechanical systems. This device reduces the computational overhead, freeing the microcontroller to perform other calculations. The L298P is the cheapest and most widely available motor control device, and it controls two motors.

There are 4 motors for the drive, 1 servo for the intake, one servo for the launcher, and one stepper for the launcher. Thus 2 L298s are required for the drive, one for the launcher. The servos can be driven by the PCA9685 directly. This leaves no spare ports for future expansion for stepper or DC devices, but leaves some room for servo-based devices that have motor control built in.

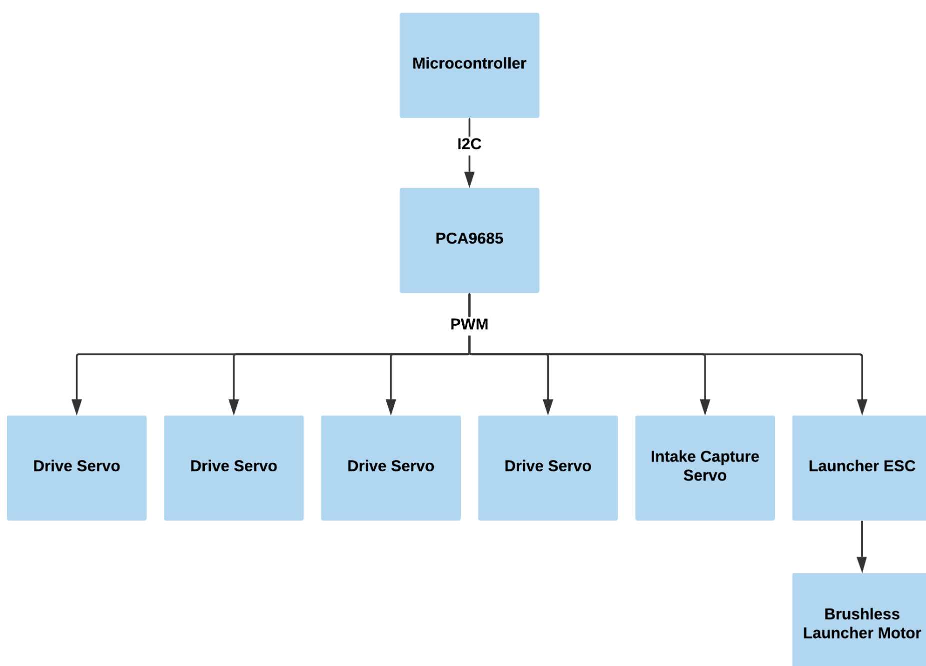


Figure 19 Control signal block diagram

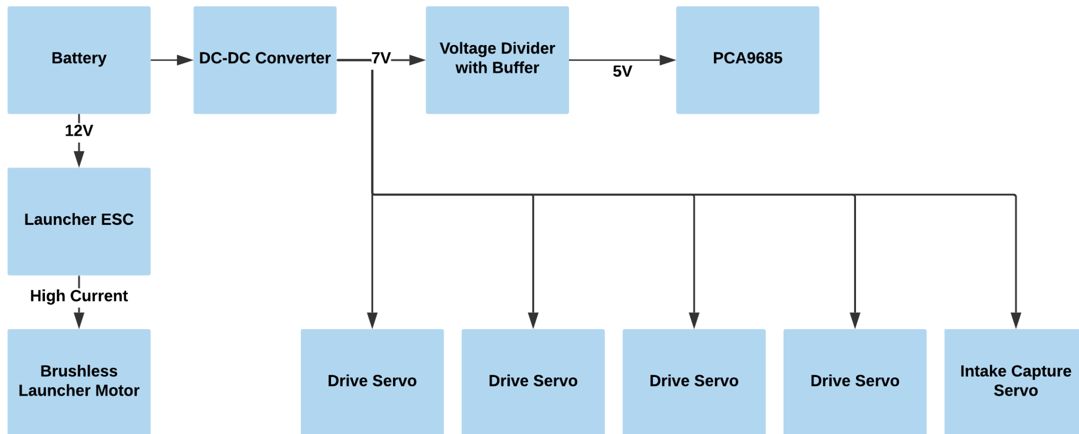


Figure 20 Power block diagram

3.4.3 Prototyping and Testing

The motor controllers can be evaluated utilizing evaluation boards available from Amazon. Each board can be purchased and tested individually to verify the design prior to the final PCB construction.

Table 33 Table of Motor Controller Tests

Requirement	Test	Required Equipment
R.R.E.3	Drive each actuator utilizing the chosen motor controller and Arduino	Arduino, Power Supply, breadboard, actuators
R.R.E.3	Drive each actuator utilizing the chosen motor controller, PWM generator, and Arduino	Arduino, Power Supply, breadboard, actuators
R.R.E.3	Drive each actuator utilizing the chosen motor controller, PWM generator, I/O generator, and Arduino	Arduino, Power Supply, breadboard, actuators
R.R.E.3	Drive each actuator simultaneously using each evaluation device (Motor controller, PWM generator, I/O generator, and Arduino)	Arduino, Power Supply, breadboard, actuators
R.R.E.4	Determine the final load of each actuator at full speed simultaneously	Arduino, Power supply, breadboard, actuators, multimeter
R.R.E.1	Determine the stall torque of each actuator, and the current at which it stalls	Arduino, Power supply, breadboard, actuators, multimeter
R.R.E.1	Determine the actual range of the servo motor	Arduino, Power supply, breadboard, servo, protractor

3.5 Microcontroller

The robot requires an onboard processor to perform the necessary calculations for locomotion and making shots. However, it is still a slave device to the arena and therefore, a microcontroller and not a microprocessor is used. A microprocessor can carry calculations at nanosecond speeds whereas a microcontroller, well, in microseconds. To provide a rich user experience millisecond latency will be enough and therefore, due to cost requirements and constraints a microcontroller is used to control the robot.

The controller is needed to control the dedicated tasks on the robot. These tasks require real time executions. The controller receives a packet from the Arena in a timely fashion and decodes them. The format of this packet is designed by the team. In excess to the overhead that comes with Bluetooth communication, the packet contains data that has substantial information for the robot to perform its activities. The update frequency of Bluetooth communication has to be 30Hz to meet the design requirement as it allows for a rich user experience. This high update rate will allow for error detection and correction most of which is inherently designed in Bluetooth's protocol allowing little to no lag on user end.

The packet received by the robot will have information on motors, velocities, configuration settings etc. Each motor is given an ID helping the microcontroller and the engineers in easily distinguishing them and applying varying velocities based on information contained in the packet. These motor values are converted to discrete values by the microcontroller and then fed to motor controller ICs using Pulse Width Modulation (PWM). The microcontroller also sends sensor data back to the Arena for feedback and makes the arena aware of the robot's location. The microcontroller also performs PD calculations for the motors to ensure accurate closed-loop control for the systems that require it. There is a myriad of options available in the market to use as Robot's "brain", however, due to the listed requirements and constraints only certain of them are feasible.

3.5.1 Research

Based on the market research there are many microcontrollers available to perform the job. The requirements however constrain the team from choosing just any microcontroller. As mentioned earlier, the microcontroller needs to control 6 motors and have the capability of getting encoder data for monitoring the velocities. When the arena sends the robot a Bluetooth packet, the onboard microcontroller parses the packet and breaks it into its respective components such as motor ID, velocity for that motor ID, Intake action commands, Launch action commands, no motion command and the like.

Due to the aforementioned tasks, the microcontroller is required to have Bluetooth compatibility for communication. There are several workarounds for this. Solution one is to get a controller with a built in Bluetooth module and have a Bluetooth stack available for programming it to send and receive data. However, microcontroller boards with built in Bluetooth tend to be expensive. Another option is to buy a simple microcontroller and have a separate Bluetooth module and use it via Universal Asynchronous Receiver Transmitter, also known as the UART. The UART is preferred method for exchanging data between the microcontroller and the Bluetooth mainly because the data format and transmission speeds are configurable.

Additionally, the microcontroller also needs to be able to send motor commands using Pulse Width Modulation and receive encoder commands via interrupts. There are boards available in the market which allow configuring every single pin as PWM and interrupt however, they tend to be expensive and constrain us in our spending limit. Therefore, the microcontroller needs to have a minimum of sending 6 PWM signal and have 8 interrupts for encoders. There are also multiple ways to work with this. First option is to buy a board with all features on board whereas another option is to buy modules and either find or create custom libraries to interface with them. Keeping such specifications in mind a list of required features was created and appropriate microcontroller technologies were studied. A summary of the findings can be in Table 34.

Table 34 Compare and Contrast of Different Microcontroller Technologies

Processor	ATmega 328P	ATmega 2560	MSP430G2553
Cost ($)	16.90	30.80	23.40
I2C	2	2	2
UART (Rx, Tx)	1	4	1
SPI	1	1	2
Interrupts	2	6	24
Digital IO	14	54	24
Analog IO	6	16	N/A
PWM	6	15	24
Operating Voltage	5V	5 V	5 V
Input Voltage	7-12V	7 - 12 V	5 V
CPU Speed	16 Mhz	16 MHz	25 MHz

EEPROM (KB)	1	4	N/A
SRAM (B)	2K	8K	512
Flash (B)	32K	256K	16K
USB	Regular	Regular	Regular

3.5.2 Design

3.5.3 Prototyping and Testing

Table 35 Controller tests

Requirement	Test	Required Equipment
R.R.S.1	Determine that microcontroller uses Bluetooth Low Energy as a serial device and can send/receive data to/from it.	Oscilloscope, Serial Monitor
R.R.S.2	Determine that the encoder interrupts increment and/or decrement motor velocities	Oscilloscope, Serial Monitor
R.R.E.5	Determine that the microcontroller successfully implements I2C, SPI and UART communication protocols	Oscilloscope
R.R.S.2	Determine that the encoder channels properly interrupt the microcontroller	Oscilloscope

3.6 Communication

The communication subsystem allows the robot to receive commands from the arena. To accomplish this, the robot must have a communication system on board and receive data over a wireless link. The communication subsystem needs to have a data update frequency of 30Hz at the minimum. Failure to do so can cause latency in robot’s motion. This latency hinders the robot from receiving data in a timely manner and constraints it from shooting successfully 75% of the time as per out requirements.

Another reason why the communication system needs to be wireless is that the robot will be moving in the field. Having cables or wires can restrict the robot and introduce noise in the communication signals. Using a differential pair is a possible solution however, the robot could damage the cables by running over them. Therefore, wireless communication is a priority to prevent any potential damages to the entire game. However, with wireless communication comes with a possibility of potential packet loss and data corruption. This can inherently introduce the similar problem of latency due to which the communication system has to have error detection and correction schemes implemented. This achievable using TCP/UPD or Bluetooth. The received packet from the Arena is designed by the team. It has information regarding the motors to be operated (i.e. motor ID), the velocity for that motor, information regarding intake, launch, and other necessary configurations.

The robot is a slave device to the arena that will receive data over the radio to perform its actions. The implemented communications protocol will also allow the robot to send its sensor data back to the arena for monitoring and debugging purposes. This data is shown by the Arena on a screen to give users more information regarding their robot. These stats could include current motor velocities, battery status, communication link status etc.

3.6.1 Research

There are multiple radio technologies for conducting Arena-Robot communication:

3.6.1a Bluetooth

Another technology which is under consideration is Bluetooth v4.2. Bluetooth is low power communication protocol which will allow the entire system to be portable and cost effective. It can be easily mounted on microcontroller and has libraries allowing wireless data exchange between the arena and the robot.

3.6.1b Wi-Fi Direct

3.6.1c Wi-Fi

One option is to use Wi-Fi. However, a router is not power efficient for the small scale of the arena. It requires 120V input and therefore, will require a dedicated outlet. The TCP/IP available with Wi-Fi will help transmit data at faster rate.

3.6.2 Design

3.6.3 Prototyping and Testing

Table 36 Communication tests

Requirement	Test	Required Equipment
R.R.S.1	Packet is successfully generated by the master/slave	Serial Monitor
R.R.S.1	Packet is successfully received from the master	Serial Monitor, Oscilloscope
R.R.S.1	Packet is successful transmitted to the master	Serial Monitor, Oscilloscope

	The system goes into sleep mode when no communication is occurring to save energy and system resources	Serial Monitor, Oscilloscope, Multimeter

3.7 Battery

The battery is the main source of power for the robot. The kind of battery to be used depends on the application and power requirements of the system. As mentioned earlier, the robot will have an onboard computer, up to 6 motors, sensors including motor drivers, analog to digital converters, DC to DC converters, and communication systems such as Bluetooth or Wi-fi Direct. Therefore, the battery needs to be strong enough to power it all.

For the motors, the robot has intake and launch mechanism that is implemented quite frequently. During this action, adding a load to the system will increase the current draw from the power supply to the motors. Therefore, the battery needs to not only fit the voltage requirement but also the overall current requirement of the robot system. The battery supplies at least 12 to 9 volts and 8-10 amps to the system to overcompensate in cases of indeterministic power requirements. This is stepped down to a usable voltage for the microcontroller and it’s peripherals using a DC-DC converter and/or voltage divider with a buffer.

The battery technology is rechargeable mainly because it reduces the overall cost of the system. It allows reusability of the components and keeps the costs at minimum consequently meeting the project requirements and constraints. The battery has to have a safety rating that meets OSHA standards. The battery needs a voltage detection circuit to determine when it is going under its minimum voltage as for instance, LiPo batteries can catch fire when electrically over drained or mechanically damaged harming the user or the environment or both.

3.7.1 Research

3.7.1a Lithium Polymer

Lithium Polymer, or LiPo, batteries are quite popular due to their light weight and higher energy rate. A single cell can hold up to 4.2V when fully charged and they are sold as a pack of multiple cells such as 2S, 3S, 4S, 5S and even 6S or more. The S essentially signifies that they are arranged in series therefore, a pack of LiPo can provide voltage up to 12.6V in a 3S (3 * 4.2 V). This property makes them an attractive choice since different combinations can be used at an affordable market rate. They also have a low discharge rate which allows them to last longer. Therefore, depending on the power consumption by the robot, a LiPo can easily power the robot system for at least 30 minutes or more. A detailed calculation of this is done in section 3.7.2 Design based on which the desired battery is chosen. Another advantage of them is that unlike Lithium Cadmium batteries, LiPo’s do not require to be fully discharged before being charged again. They can also be used in parallel to increase the current source to the system. LiPo batteries are also environment friendly unlike Cadmium, Lead or Mercury batteries which is also an important design decision for longevity of the system. [5]

LiPo batteries are rated with respect to their current and capacity rating. Therefore, a 2200mAh LiPo battery at 25C can provide 55 Amps of current at 11.1V for 1 hour, or 5 amps of current for 11 hours at the same voltage. This makes LiPo batteries an attractive choice as the launcher might use variable force to throw the ball which in turn would change the load on the motors. In addition to purchasing the battery, a proper battery charger and monitor is required as LiPo batteries come with inherit risk of fire and cannot be over or under charged due to their chemical composition. Additionally, they are quite expensive and their price increases with their capacity rating and number of cells. Therefore, an important design decision is to choose whether two 3S LiPos at 2250 mAh or one 3S LiPo at 5500 mAh capacity as this causes a dilemma choosing between cost and weight and one has to be sacrificed for the other.

3.7.1b Nickle Cadmium

Nickle Cadmium is one of the oldest battery technologies that were revolutionary upon their arrival. They made low powered portable systems a reality however, lost their market share to Lithium batteries.

Some of the positive characteristics of NiCad include low internal resistance. This allows the energy to easily travel from battery to the system and therefore, is an important trait in choosing the battery technology. Modern digital systems require high current spikes from time to in operation unlike analog loads that work easily on steady current. Therefore, a lower internal resistance acts as an important factor in determining the battery to be used in building the robot system. NiCad batteries can be easily stored in charged or discharged state without harm unlike LiPo batteries that need to be at a certain voltage before being shelved for prolonged period of time. They are available in a large variety of sizes and capacities [6].

Some of the negative characteristics of NiCad batteries include their susceptibility to memory effect [6]. This effect causes the battery to remember its previous discharge state and hinders its next recharge cycle from reaching a full potential. This is usually prevented by either discharging the battery completely before recharging it or buying a charger with capabilities to carry out such operations. This can increase the cost of building the robot as such charges are expensive. Like LiPo batteries, NiCad are prone to damage by overcharging.

3.7.1c Lead Acid

Lead Acid batteries are an industry standard that are featured in robots, cards, industrial machinery, power supplied and much more. They are cheap and reliable which make them an attractive choice for a financial standpoint. However, one of their major limitations include their weight. They are typically used in situations where weight is not much of a problem or concern.

One of the major pros of Lead Acid batteries include its reliability. They have been in development for over a century and are scaled enough to be available at a cheaper price compared to LiPo or NiCad batteries. They are tolerant to abuse and overcharging and do not explode in strenuous environments unlike LiPo batteries. They have an indefinite shelf like which can a plus to the robot when kept dormant for prolonged periods and can deliver high currents required to run the flywheel for intake and launch and servos for locomotion.

However, their weight is a serious disadvantage. Due to their high reliability, they tend to come in bulkier packaging which will add on to robot's overall weight and put pressure on the electronics to function with ease. They also do not charge fast unlike LiPo and NiCad battery technologies which can deteriorate user experience exponentially. Finally, they overheat easily and can cause disruptions in sensor readings and wear the robot hardware [7].

3.7.2 Design

Based on the research conducted in section 3.7.1 Research a system power analysis was conducted to specify what battery met the desired requirements and specifications. The results can be seen in Table 37 that shows how much power each system will need to operate under worst case scenarios and the overall power robot will use to operate. Conclusively, LiPo battery seems like the optimal solution to driving the robot due to multiple reasons.

Table 37 Power Calculations of Robot's Subsystem and Components

Subsystem	Part Name/Number	Unit(s)	Voltage (V)	Current (A)	Power (W)
Bluetooth	CC2541	1	3.3	0.02	0.066
Microcontroller	ATmega328P	1	5	0.2	1
Encoder	TLE4946-2K	1	5	0.05	0.25
PWM Controller	PCA9685	1	5	0.04	0.2
Servos	Parallax #900-00360	5	6.8	1.2	40.8
ESC + Motor	A2212/13T	1	10	2	20
Total Power					**62.32**

A LiPo battery can charge quickly and discharges at a longer rate. This allows the robot to run for a prolonged period of time. The specification of the battery that will run this robot need to be at least a 3S LiPo that can provide anywhere from 5000mAh to 6000mAh charge rate capacity. However, a cheaper solution would be to use two 3S LiPo batteries in parallel with 2250mAh capacity each, but it will increase the robot weight and occupy more space than a single LiPo battery. The specified battery can run the robot for approximately one hour on a full charge and 40 minutes on the minimum safest cell voltage (i.e. 3.7V each). Therefore, the battery should be able to easily support the robot and its activities for more than one hour. A test will be conducted upon purchase to determine the actual time the battery can run the robot for.

3.7.3 Prototyping and Testing

The battery can be purchased and tested with the materials available in the Senior Design lab. No tests can be done prior to component purchase except making sure that the calculations in section 3.7.2 Design are correct.

Table 38 Battery tests

Requirement	Test	Required Equipment
C.R.3	Determine that the battery is not undercharged	Portable BMS unit, Multimeter
C.R.3	Determine that the battery is not over charged	Portable BMS unit, Multimeter
C.R.3	Determine that the battery provides the necessary voltage and current to the system	Multimeter, Active Load
R.R.26	Determine expected runtime of the robot	Active Load

3.8 DC-DC Converter

The battery provides a high voltage and high current supply to the entire system. This can be harmful for certain integrated circuits and sensor technologies. Most sensors work at a standard 5V transistor-transistor logic, or TTL voltage. However, it is not uncommon to come across technologies that run on 3.3V. The reason behind such vast changes in logic levels is inherent to manufacturers and power consumption requirements of the system. Lower voltage levels and current draw contribute to the longevity of systems. However, it could come at a cost of high performance, cost and rich user experience.

The DC-DC converter has to be 9V to 12V tolerant and therefore, a switching regulator is needed to maintain high power efficiency as the voltage is stepped down by this system. The power supplied to the robot with a LiPo battery is more than what a microcontroller can safely handle. The DC-DC converter takes in raw battery voltage and current and converts it into an acceptable power level for the system components. The microcontroller used for this project works on 5V Transistor-Transistor Logic and therefore, a 5V converter is required to power it on. The pins of the microcontroller can supply a maximum current of 20 mA and with a maximum of 20 pins a total of 400 mA can be drawn from Arduino pins at the same time which, however, is an overestimate as a phenomenon like this highly unlikely as per the design.

A 6V DC-DC converter is used to power the continual rotation servos that help the robot in its movement. The servos use 15mA of current when idle, about 150mA or current when rotating with no load, and 1.2A of current when stalled. Therefore, the converter needs to supply at least 4.8 amps of current in worst case scenario for all driving servos. This is an important requirement for the robot to move.

An additional 5V DC-DC converter is used for powering the microcontroller, encoder, and PWM Controller. These are low powered systems that use 200mA, 50mA, and 40mA of current, respectively. Therefore, a linear regulator that can support 1 A of current should suffice. The converter will take 9 to 12 V of battery input and will try to regulate the output voltage to a steady 5V with an error ±0.1V. An alternate solution would be using a voltage divider from a buffer that takes 6V regulated output as input. However, this causes issues such as failure in case the 6V switching regulator fails. Having separate voltage conversions will allow connecting a GPIO line from the microcontroller to the 6V DC-DC output that can interrupt the processor in case the line goes low. This can help in debugging the robot when it stops moving without the need of a multimeter. Another problem with the voltage divider with a buffer is that the output voltage is not regulated and therefore, it can be anywhere from 4.5V to 5.5V which is a huge change and loss of power.

A 3.3V DC-DC converter is used to power the Bluetooth Communication System. The Bluetooth communication system uses 50mA of current at maximum and hence it is low energy. Therefore, the converter takes 5V input from linear regulator and steps it down for the Bluetooth to use. To create these converters, an online tool named TI Webench was used which is discussed in section 3.8.1 Research.

3.8.1 Research

3.8.1a TI Webench

This tool generates a Bill of Material (BOM) and PCB layout and therefore, finding parts to generate this should be convenient. An important step in building a DC-DC Converter is efficiency. We are hoping to achieve an efficiency of over 90% which can be obtained by using a buck regulator. The current used by microcontrollers typically ranged from 1-2 A and this needs to be taken into consideration with the PCB design. Instead of traces, a copper pour will be used as it provides high efficiency to DC-DC converter.

3.8.2 Design

The schematics were generated using TI's Webench online tool. It recommends possible voltage converters based on input and generates a bill of material as well.

3.8.3 Prototyping and Testing

The DC-DC converter can be built with the appropriate components on a breadboard in the Senior Design Lab. All of the appropriate equipment and resources are available to verify the design.

Table 39 DC-DC Converter tests

Requirement	Test	Required Equipment
R.R.E.2	Determine that the DC-DC converter outputs desired voltage and current	Multimeter, Active Load
R.R.E.4	Determine that the DC-DC converter is power efficient and does not lose energy as heat to the environment	Multimeter, Thermal Analysis

3.9 PCB

The PCB component is the implementation and integration of the various electronic systems defined previously. It must connect the various integrated circuit components to the microcontroller and have slots for the peripherals to plug into. The Printed Circuit Board is required to simplify the design of the robot. The electronics that make up the robot consists of 6 motors, a microcontroller, a DC-DC Converter, Motor Drivers, H-Bridge Circuit, encoders, and communication system such as Bluetooth. The PCB connects all these systems together and gives the robot a sophisticated appeal. The PCB needs to be small enough to fit the robot and keep the costs as minimum.

The Printer Circuit Board is a 2-layer copper board with silk screen and soldering pads on it. It consists terminal blocks that intake power from the battery and then direct them to a voltage regulator by the means of traces. The width of the traces depends on the power it is carrying. The width of the trace carrying battery power will be thicker than the trace carrying the power to the microcontroller. The PCB contains test points to check voltage and currents at certain spots. Additionally, adding test points helps in determining the signals using an oscilloscope which can help in debugging communication protocol problems that might arise while implementing I2C, SPI, or UART. The PCB also contains a reverse voltage protection circuit leaving the user with minimal adjustments and focus on playing the game out of the box.

3.9.1 Research

3.9.1a Autodesk Eagle CAD

3.9.1b Diptrace

3.9.1c AutoCAD Electrical

3.9.2 Design

Table 40 I/O Schedule

	Type	Connected Devices	Pin Type	Power Requirements
Drive Front Left	Servo	PCA9685	PWM	7V 1.2A Stall
Drive Front Right	Servo	PCA9685	PWM	7V 1.2A Stall
Drive Back Left	Servo	PCA9685	PWM	7V 1.2A Stall
Drive Back Right	Servo	PCA9685	PWM	7V 1.2A Stall
Launcher / Intake	Brushless DC	PCA9685, ESC	PWM	10V 10A Max
Launcher Release	Servo	PCA9685	PWM	7V 1.8A Stall
Feedback Front Left	Hall	Microcontroller	1 Digital	
Feedback Front Right	Hall	Microcontroller	1 Digital	
Feedback Back Left	Hall	Microcontroller	1 Digital	
Feedback Back Right	Hall	Microcontroller	1 Digital	
Feedback Launcher	Encoder	Microcontroller	2 Digital	
PWM Generator	PCA9685	Microcontroller	I2C	
Bluetooth	HC-06	Microcontroller	Tx, Rx	

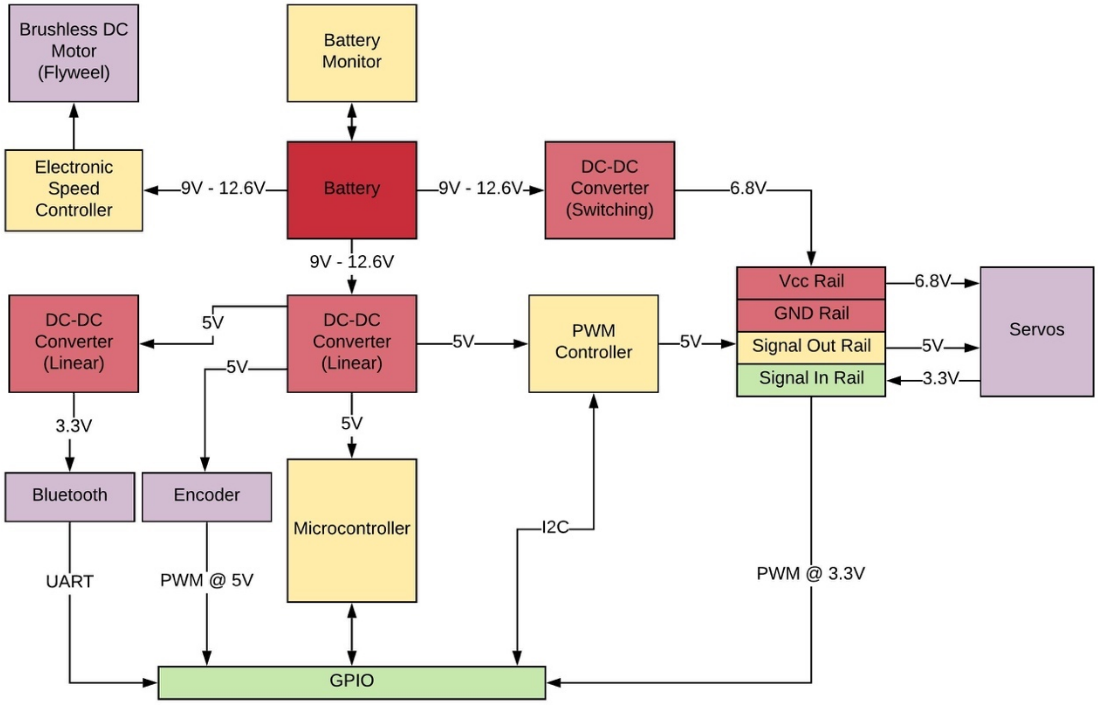


Figure 21 Robot Electrical Network Block Diagram

3.9.3 Prototyping and Testing

The PCB is to be designed and purchased through JLCPCB which sells 2 Layer PCB for $2 for 5 boards. Additionally, the stencils can be purchased for the board at just $6. The stencil is utilized in conjunction with the board and solder paste to quickly and accurately build the PCB. A heating chamber is required to evenly heat the board to prevent damage.

Table 41 Robot PCB Tests

Requirement	Test	Required Equipment
C.R.1	Determine that the PCB is an appropriate size to fit the robot	Ruler
R.R.4	Determine that the PCB does not have any shorts or opens	Multimeter

3.10 Software

The software component drives the hardware components in the PCB. This includes any control software such as various PID control, state machines, and Bluetooth communication. The software for the Robot must be at least Soft-Real time to ensure that inputs and outputs are processed in a context that does not affect the fidelity of the system. For example, an input from the motor encoders should be processed and utilized in outputs for the relevant motor within a single deterministic loop. If the motors outputs are updated too long from the motor encoder input, the data is no longer valid and could be harmful to the system. Thus, a firm data flow structure must be followed for the entire software system. Additional limitations on the software are based on the microcontroller chosen in section 3.5 Microcontroller. Ideally, the written software follows a strict architecture to enhance readability and debuggability. Debugging is critical for the robot because there are several factors that could lead to failure: Mechanical, Electrical, and Software issues. Often each one of these have issues are observable only in another area. Thus, the chosen software and libraries must have thorough documentation, and be thoroughly tested prior to usage. Each function or block of code should be fully documented and contain a unit test that corresponds to the requirement that drives the function.

3.10.1 Research

3.10.1a Arduino IDE vs Atmel Studio

The Arduino IDE is a very popular software that includes a full development environment including a text editor, compiler, boot loader, and serial monitor. It has a very easy-to-use interface and a large amount of documentation due to the prevalence of Arduino as a hobby device. The IDE is strictly designed around particular Arduino boards, thus the support for the chip itself is somewhat limited and requires extra effort to work with.

The Atmel Studio software is a software provided by Microchip that supports development and debugging for AVR and SAM microcontrollers. The application contains a fully-fledged IDE that supports text editing, compiling, debugging, and deploying to AT chips. However, this does require (similarly to the Arduino IDE) an extra chip that acts as a programmer device. The IDE can also import Arduino sketches and libraries.

Another option is the Visual Studio Code extension for Arduino that extends the capabilities of the Arduino IDE. The Extension provides full IntelliSense and all of the advanced capabilities of the Visual Studio Code application. This provides all of the capabilities of the Arduino IDE with a much better text editor.

3.10.1b Libraries

PID Libraries

PWM Libraries

I2C / PCA9685 Libraries

Servo Libraries

3.10.2 Design

The design for this software is developed to ensure maximum robustness, scalability, modularity, and maintainability. The general structure follows a strict I/O paradigm to guarantee real-time reliability. The design does the following: updates inputs, processes the system's data, and updates the outputs. This entire process must run each cycle with a deterministic scan time to ensure real-time operation. The process is shown graphically in Figure 22. The software and appropriate unit tests are to be developed and handled in Visual Studio Code with the C/C++, and Arduino IDE extensions.

The system defines three finite state machines shown in Figure 23,Figure 24, and Figure 25. The Master state machine defines the states that the robot system can be in, and their valid transitions. The transitions are determined by the master arena software and faults when errors occur in any of the processes. The actuators and sensors on the system have their own state machines to reduce system complexity in fault-tolerance and error checking. In the actuator states, the actuator transitions from offline to tracking such that individual actuators can be deactivated without disabling the entire system. The tracking state indicates that the actuator is actively following commanded positions or velocities, and the fault state indicate that some error has occurred such as tracking errors or invalid state transitions. The sensor state machine indicates the validity of the sensor's information. The reset state indicates that the sensor's information is invalid in this cycle and it must reset accordingly. Active indicates that the sensor is actively tracking velocities / positions and the information appears valid. Fault indicates that the sensor has had an error or invalid transition.

The software architecture and preliminary class diagram is shown in Figure 26. Four isolated layers are defined: Application layer, System layer, object layer, and library layer. The application layer contains the primary infinite loop for the robot's software that processes the inputs, data, and outputs in the appropriate scan time. It instantiates and calls methods from the classes in the system layer. The system layer contains classes defining robot-specific systems such as the intake, launcher, and drive systems. Each of these systems contain state-specific processing such as determining when an actuator should be active, how inputs from the master arena are handled, and general state machine I/O processing. The object layer contains abstract code for actuators, sensors, and state processing. This layer must be instantiated and operated by an upper level layer. However, the majority of the data of the system is processed and allocated to this layer. Finally, the system and object layer leverage existing libraries when possible. These libraries include the PID, Servo, I2C(PCA9685) and Bluetooth libraries available for the electrical components.

Figure 22 High level process flow

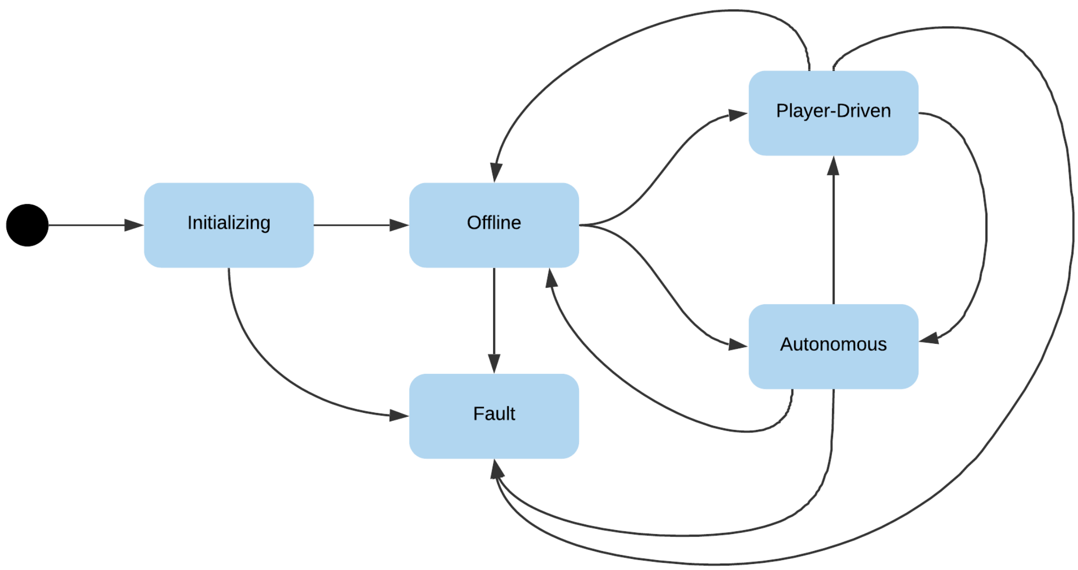


Figure 23 Master state machine

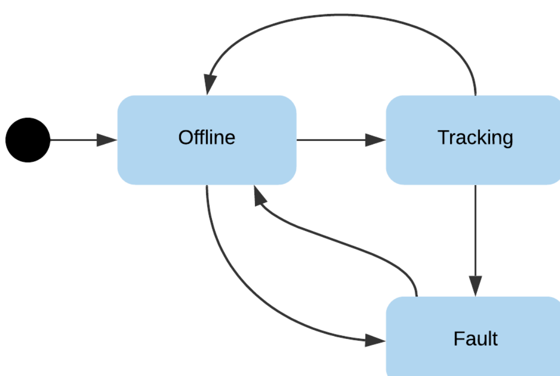


Figure 24 Actuator State Machine

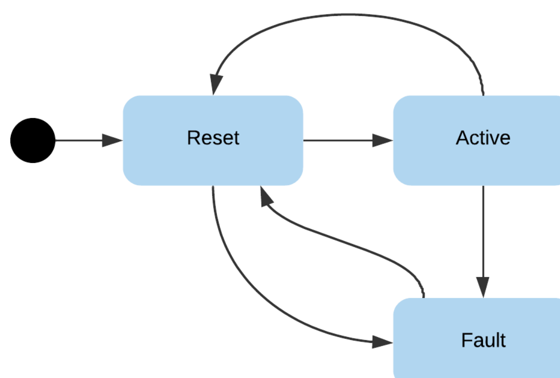


Figure 25 Sensor state machine

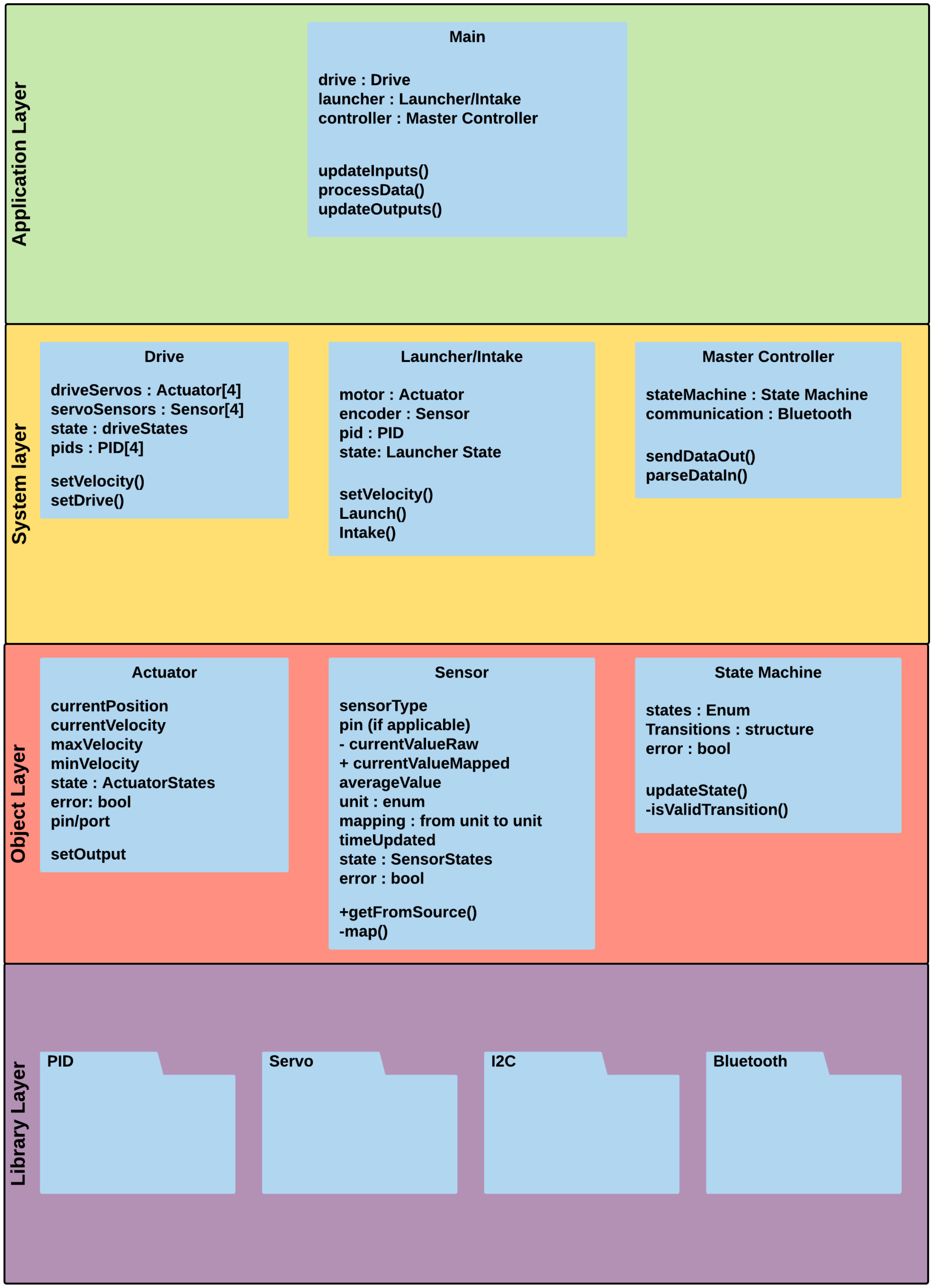


Figure 26 Robot software architecture design and class diagrams

3.10.3 Prototyping and Testing

In order to adequately prototype the software for the robot, an Arduino Uno and some evaluation boards are needed to mimic the functionality of the PCB if it is not finished yet. Otherwise, the software unit tests can be developed independently to the actual software such that each code block can be built and tested without the other blocks being complete.

Table 42 Software System Tests

Requirement	Test	Required Equipment
R.R.S.3	Communication Unit Tests (Bluetooth, Wired)	Bluetooth Module, USB Cable, Arduino, Power Supply, breadboard
R.R.S.3	Actuator Unit Tests (Base, intake, launcher)	USB Cable, Arduino, Power Supply, actuators, breadboard
R.R.S.3	Sensor Unit Tests (Encoders, Switches, battery, etc.)	Sensors, USB Cable, Arduino, Power Supply, breadboard
R.R.S.3	Control Unit Tests (PID, etc.)	Sensors, Actuator, Arduino, Power Supply, breadboard
R.R.S.3	Safety system Unit Tests (Heartbeat, etc.)	Arduino, Power supply
R.R.S.3	Full software tests (Operations at max capacity, timing, etc.)	
R.R.S.3	State machine Unit Tests (including operating modes)	Arduino, Power supply

4.0 Arena

The arena subsystem is the subsystem physical frame that the robot can be placed on, along with all the components required for basketball gameplay. It contains the computer vision component of the project for robot and ball tracking. The subsystem also contains all the player experience including lights, sounds, and display.

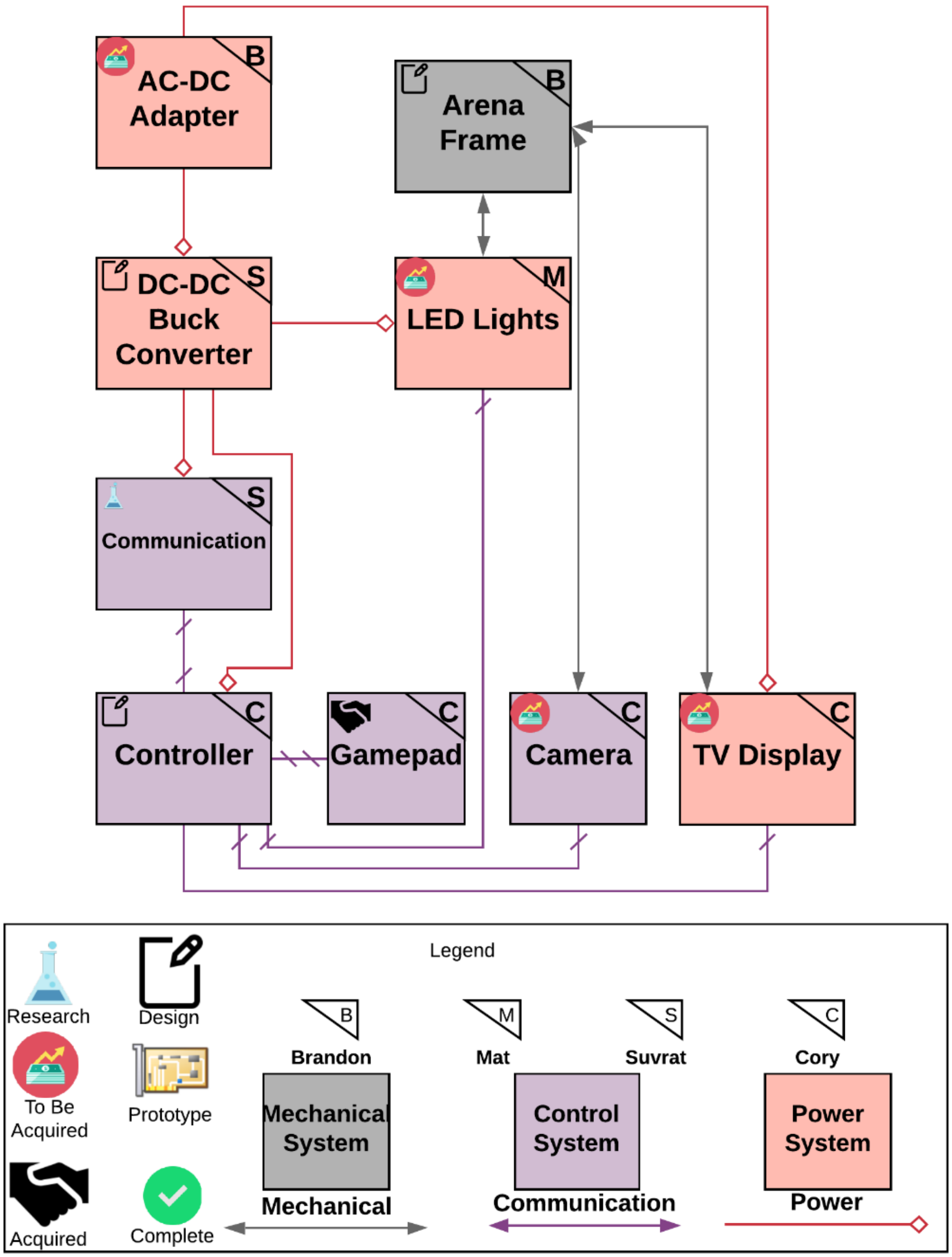


Figure 27 Arena Subsystem Power and Signal Diagram

4.1 Frame

The frame component is the structure to hold the robots, electronics, court, and other components physically. The frame can be quite large; thus, it is designed modularly such that it can be stored and transported in a small location. The court must be perfectly level to ensure the ball remains in place, thus the frame that supports the court flooring must include a leveling apparatus. The exterior of the frame will be closed off to prevent the ball from flying outside of the arena. The material must be transparent to make sure that spectators can see the entire field.

4.1.1 Research

4.1.1a PVC

PVC pipe is a light weight and low to medium cost material. The cost will depend on how many connectors are used as they are the most expensive PVC part to buy. The number of connectors used will depend on how much a PVC pipe can maintain level at varying lengths, if the PVC needs to be strengthened more to keep it level that increases cost. However, to alleviate the need for extra connectors we could use thicker pipe, again though the thicker the pipe the more cost increases. PVC pipe is a very portable material though since it does not need any hardware or glue to hold it together. The downside to this portability is that when taken apart and put together repeatedly, it could go together at a different depth than it did before depending on the force applied by the person putting the pieces together. This varying depth can throw off calculations if the arena ends up being off level. A unique problem to PVC is that it is the only round material considered, a round material is not easy to mount other parts on and supplies little support to the flooring of the arena. Additionally, the round property of PVC means that when together it can rotate in place, this can again affect calculations if the camera mount is not placed in the exact same height and position every time. Lastly PVC pipe placed into connectors will always have a lip between the connection point. Because of this lip a piece of plywood may need to be added to the frame in order to ensure the flooring can sit above the lip to make it level.

4.1.1b Metal

A metal frame will cost the most out of all options. To create a metal frame would consist of at least four L angled brackets made of aluminum, steel, or another lightweight inexpensive metal. These L angled brackets would be used as the wall mounts of the frame and additionally for securing the particle board or similar material to the frame that will be used to support the flooring. From our research, it was concluded that aluminum would have been the cheapest option for the L angled brackets with prices for all sides of the arena at over $100. Additional materials to construct the frame from metal would include the particle board or similar material to secure the flooring to the frame. In the research done the lowest cost material for this would be an OSB board which at the correct size for the arena would be roughly $8 at the lowest. The walls, hoops, and camera mounts would be connected to the L angled brackets using locking hinges so that the arena walls, hoops, and camera mount are capable of folding into the arena. These hinges were around $7 a piece and we would need at least two per wall, one per hoop, and one for the camera mount. The metal frame would supply the arena with the best compact design in that the arena would be able to fold into itself and carried. This material would also be near the highest weight of the researched materials. Even though the portability of this material is excellent, the minimum cost of around $180 is quite above the estimated budget for the arena. So, if this material is used either the budget of the arena would have to increase, the budget for the entire project would have to increase, or the budget would have to be lowered from another section of the project and applied here.

4.1.1c Wood

A wood frame would be one of the lowest cost options for the frame, but would require the most actual work to construct, in that there will need to be numerous cuts in the wood that require a tiny bit of skill in carpentry. Making a wooden frame would use four 2x4s as the walls and four 2x2s as the lengthwise flooring supports in the middle of the walls. Each cut in the four-foot sides (basket sides) will be a notch. One notch on each side will hold the five-foot sides (lengthwise sides) perpendicular but even height with the four-foot walls. The other four notches will be evenly spaced and will hold the 2x2 strengthening planks an inch below the surface of the outer walls. The flooring will then sit on these strengthening planks with no glue or hardware to hold them down. For this reason, the wood used will need to be strong, rigid, as straight as possible, and with as little knots as possible to ensure the cuts and notches will be even enough to hold the supporting braces level for the flooring. The wood will also need to be reasonably priced and accessible. For this reason, and all previous explained building choices, a kiln dried softwood, such as pine, spruce, or douglas fir will be preferable. This type of wood for a 2x4 would be about $5 for 10 feet of material and $2 for 8 feet of 2x2. The walls, camera mount, and basket mounts will be attached to the frame with PVC pipe and a bolt that will go through the pipe and pipe holder to hold it in place. The hardware for this would be roughly $10. Using wood would come to a total cost of around $30 to construct the frame. The savings in the arena budget if using wood could then be used on better parts or parts that make other parts of the project easier. The downside of using wood is that it is difficult to get notches cut very level and it’s of medium weight when needing to transport the eight planks together.

4.1.2 Design

The proposed size of the arena is approximately 4 ft width by 5 ft length by 3 ft height which is not to an exact scale of a real court. After creating prototypes of the arena in SolidWorks using the previously mentioned materials and putting together a parts list including price for each, our team decided to use wood because its inexpensive, requires little hardware, and can be easily disassembled and reassembled.

The design of the frame will use four 2x4 kiln-dried heat-treated spruce-pine-fir wooden pieces as the walls and base frame. Two pieces on each basket side will serve as the main notched pieces that will hold the middle supporting braces. Each of these two pieces will be cut to four feet in length with 1x1 inch notches cut two inches deep a half inch away from either side of the 2x4. These notches will be used to slot the lengthwise 2x4s into place. The lengthwise 2x4s will have the matching joint cut so that it fits tightly into the basket side notches and is level on the top and bottom of the joining pieces. With these four sides fit tightly together the inner perimeter of the frame should measure 18 feet or 4x4x5x5 feet on each side. Additional 2x1 inch notches shall be cut into the basket-side pieces two inches deep. These notches will be used to hold the supporting braces. The supporting braces will be made of the same type of wood cut to a 2x2 inch plank five feet two inches long. Each side of the supporting braces will have a joint cut to match the notch on the basket-side pieces. This cut should place the supporting braces one inch below the top of the frame. A cross support shall also be placed in a notch cut three inches deep in the middle of the long-side wall. With all walls and supports in place this should create a level platform in which to place the flooring of the arena. The frame will then have lead screws attached to each corner area to keep it level on any surface. Small levels will then be attached to each side to ensure the arena is always level. Figure 28 shows the design of the arena frame and walls.

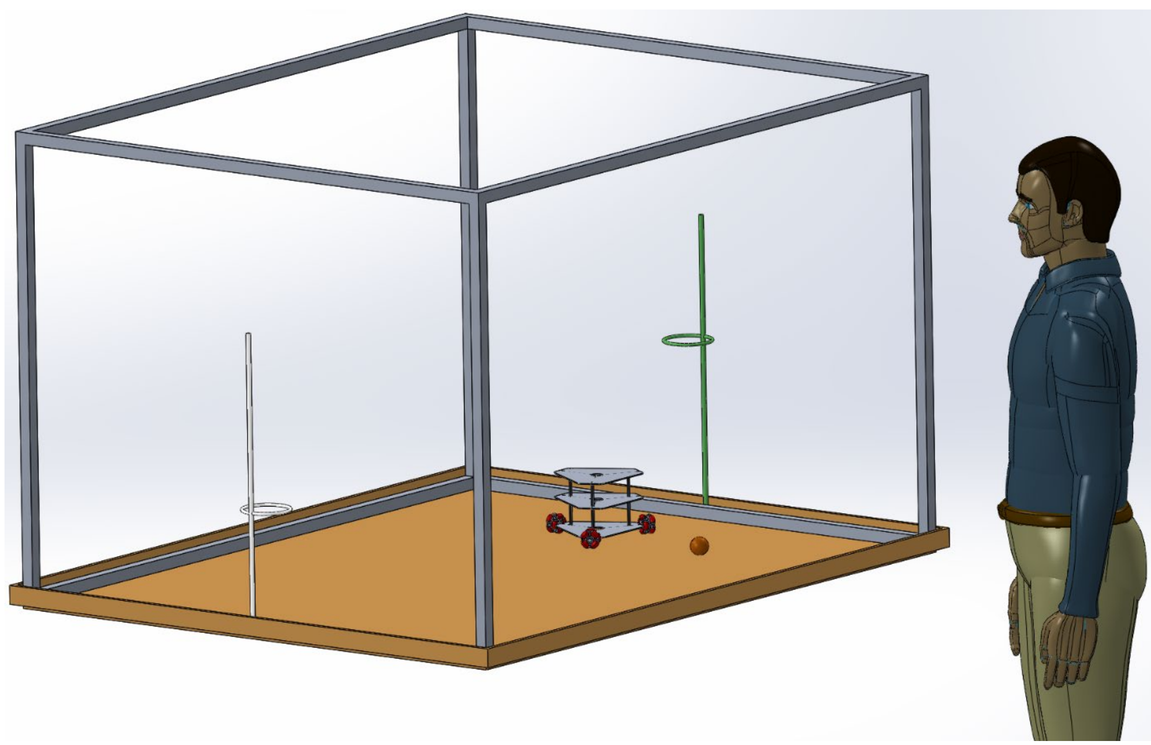


Figure 28 SOLIDWORKS Image of proposed scale Arena, Robot, and Ball

4.1.3 Prototyping and Testing

The frame components can be purchased from local hardware stores. The cuts required to correctly set up the frame need to be done with a jigsaw and table saw, both of which are available through team-member's families.

Table 43 Frame Tests

Requirement	Test	Required Equipment
R.A.6	Time how long it takes to put together	Stopwatch
R.A.6	Time how long it takes to take apart	Stopwatch
R.A.7	Determine how level the system is from different starting conditions	Level
R.A.8	Determine strength of the frame in event of falling	

4.2 Walls

The walls surrounding the arena exist to prevent a rogue ball or robot from flying out of the arena and hitting someone or something it is not supposed to. It also exists to prevent people from placing arms or objects into the arena while the robots are running.

4.2.1 Research

4.2.1a Clear Acrylic Plastic

The first material to be researched was clear acrylic plastic. This material is very rigid and the strongest material to be considered. This rigidity would make acrylic the best material for a camera mount alleviating the need for extra mounting material should the camera instead need to be mounted to the frame. The solid acrylic panels would also be an excellent choice for dampening any wind that could occur from outside forces and affect the calculations for shooting the ball, which would be necessary should this project need to be demonstrated in an outdoor environment. A clear acrylic wall would also make the best choice for viewing the robot, making it very easy for any player to see and control their robot. Attaching the wall to the arena is also made easier by the acrylics rigidity as hinges would be all that is needed to hold the walls and would make them collapsible for portability. The downside of the acrylic material is that it is much heavier and would add a tremendous amount of weight to the arena decreasing portability. In addition, the cost of clear acrylic plastic is tremendously more than any other material considered. In fact, this material is the absolute best choice considering all aspects, however the cost is so prohibitive that our team is unable to purchase the necessary quantity needed for this project.

4.2.1b Clear Vinyl Plastic

A clear vinyl material is a medium cost solid, but not rigid material. For this project at least six gauge or thicker vinyl would be used. This thickness would allow for hardware to be installed without ripping the vinyl material when it is pulled tightly. The vinyl will need to be pulled as tight as possible in order to make the material as clear as possible and help the ball bounce back into the court and keep the robot from falling out of the court. A vinyl material will also dampen wind almost as good as the rigid acrylic plastic if pulled tight enough. Due to the vinyl not being rigid the camera will have to be mounted on the frame, requiring more hardware. This is a lightweight material and will add almost no weight to the arena making it more portable. In addition, since the material is not rigid it can also be rolled up and carried separately. Attaching vinyl to the arena would entail the use of posts on each corner of the frame, again requiring additional materials.

4.2.1c Mesh

The third material considered is a mesh material, either plastic or nylon woven in a net like structure with one inch or less square holes. The mesh material will be the absolute cheapest material considered, costing only several dollars for many square feet of mesh. Mesh is also not a rigid material and thus will not be suitable for attaching the camera boom and so additional hardware will be required. Mesh is however extremely lightweight making it a very good material when considering portability. The ability of mesh to dampen wind is almost nonexistent, so there would require more work in the ball shooting algorithm to insure target goal probability. Mesh will also require additional materials to attach to the frame in the form of posts. These posts could be PVC or another sturdy low-cost material. PVC would be an easy solution as it is sturdy in short lengths and can easily be slotted into the arena by attaching PVC caps to each corner and placing the three-foot PVC pipe with mesh material attached into these caps.

4.2.2 Design

The final wall design will be a combination of PVC pipe to hold each corner upright and a mesh material used for the physical wall itself and some small ceiling hanging hooks to hold the base of the wall to the frame. Each wall will be made of ¼ inch woven mesh material three feet tall and either four or five feet wide depending on which side. The wall posts will be made of 1/2-inch PVC pipe three feet long each. These PVC pipe posts will fit into PVC plugs that are mounted to each corner of the arena frame. Each PVC pipe will have small holes drilled ¼ inch apart through both sides of the pipe down the full length of the pipe. The mesh will have single strands pulled through these holes and tied to keep the mesh tight to each PVC post. The frame will have small ceiling hanging hooks attached along the base of the wall area. These hooks will be used to hold the base of the mesh wall tightly in place. When completed there will be four PVC corner posts with mesh connecting them. This will be one piece and will be capable of rolling up, like a scroll, for easy transportation.

4.2.3 Prototyping and Testing

Table 44 Wall testing

Requirement	Test	Required Equipment
R.A.17	Test if the ball goes through the mesh thrown at different starting speeds and locations	Ball
R.A.9	Test if the ball can roll underneath the wall	Ball
R.A.9	Test if the robot can push through the wall siding	Robot

4.3 Court

The court component involves the actual floor of the arena that the robots drive around on. The flooring must be easily transportable and must be able to attach to the frame described in section 4.1 Frame. The floor paneling must lay flat on the frame and contain any required basketball court markings. This is because the ball will move roll around without input force and end up in a hotspot on the court. Additionally, the basketball court markings can be used as a way to ensure the court is placed together properly such that computer vision remains consistent between teardowns. The floor material should have a coefficient of friction high enough that the robot can consistently traverse the court without fear of slipping in the driven directions. If the wheels cannot grip properly on the court, the robot will not move, or the holonomic motions will be very inconsistent, leading to a poor player experience. The court is intended to mimic a full-size basketball court, however, after testing and prototyping, it may be advantageous to switch to a half-court style arena rather than a full-court arena.

4.3.1 Research

4.3.1a Laminate

Laminate flooring is a low-cost portable option for the court of the arena. Laminate flooring comes in many different color variations as well, which is helpful in choosing a color that works well with the computer vision tracking program. Laminate flooring generally comes in lengths of around 48 inches and widths of around 8 inches per plank. Each laminate flooring piece connects in a puzzle piece locking manner, when locked together the flooring has little to no bumps, groves, or creases. Additionally, as long as the frame holding the flooring up is level, the laminate flooring will also be level when locked together. Laminate flooring is also lightweight, about three pounds per flooring plank. For the entire arena to be covered, 4x5 feet of space, seven planks will be needed. Eight to nine planks will come in one package of laminate flooring, so only one package would need to be purchased. Each package depending on color, brand, and thickness will range in price from $12 to $20 making this a very inexpensive choice for the court even if choosing the highest priced options.

4.3.1b Metal

An aluminum court would in theory be a great choice as metal is generally flat with no impurities in the surface that would cause bumps in the court and is lightweight. However, in practice it would depend on the thickness of the aluminum and how we transport, cut, and mount it. A thinner aluminum like 0.032 inches would be ideal for low weight as a 4x5 foot sheet would weight about 10 pounds. A thinner sheet though would be flimsy and need a solid frame below for support. If not enough framing support is under a thin aluminum sheet it will start to develop waves in the metal, once the waves start to develop it is almost impossible to get the metal to be perfectly flat again. To fix this problem a thicker sheet of aluminum could be used, something like 3/16 of an inch. At this thickness the aluminum would not need much framing to support it and maintain its surface through transport. An aluminum sheet 4x5 feet at 3/16” thick will weight approximately 36 pounds, clearly a drastic increase in the weight of the arena. Regardless of the thickness of the aluminum sheet it will need to be cut in half either lengthwise or widthwise to make it portable. Because the sheet is cut in half it will have to be rejoined together when placed on the arena frame, this can be accomplished by either laying the two sheets down next to each other and hoping they don’t move or adding hinges to hold the two halves together. Whichever idea is chosen will create a small gap or possible difference in height between the two pieces which will have to be fixed somehow to make the court completely level again. Additionally, aluminum would be the most expensive material to create the court. From online quotes for a 4x8 foot sheet of 0.032-inch-thick aluminum sheets it would cost $109 and go up to $398 for an equal sized 3/16 of an inch-thick sheet. This would clearly break the budget for the entire arena assembly.

4.3.1c Particle Board

Particle board is an alternative to plywood. There are different types of particle board and the type chosen to discuss here will be OSB, oriented strand board, as it is the lowest cost while maintaining uniform construction and rigidity. OSB comes in a variety of sizes and can be bought and cut such that the whole arena, 4x5 feet, could be covered by only one piece of OSB. As one solid piece OSB is very sturdy and if bought at the correct thickness would not need any framing underneath to keep it level. While it sounds great to only have one piece for the entire floor this project has portability as a restriction and therefore a 4x5 foot sheet of OSB would not be portable. Thus, the sheet of OSB would need to be cut, at the least, in half to make either two 2x5 halves or two 2.5x4 halves. This half cut would be a detriment to the rigidity, levelness, and ease of the OSB sheet as joining the two halves together would almost certainly add a slight bump to the middle joint and add hardware to connect the two halves. Additionally, the longer and thinner width of the sheet the more likely the sheet is to start bowing thus increasing the need for frame supports or risking the flooring to be unlevel. OSB can easily be painted as well to any color and design that would work well with the computer vision program. However, OSB and all particle board, generally does not look very professional or sleek, even when painted. Lastly, for the amount of OSB that would be needed for this arena the cost would be around $8, which is clearly the lowest cost of any of the materials researched.

4.3.2 Design

The final design of the court will be using the laminate flooring material because it was the best combination of lightweight, portability, cost, presentability, and would maintain a flat level surface after multiple instances of being taken apart and put back together. It will require seven laminate flooring planks to cover the 4x5 foot area. Since each plank is 8.03 x 47.94 inches, seven planks will cover an area of 56.21 x 47.94 inches with no cutting of the planks involved. The court will be inserted on the arena frame one plank at a time with the lengthwise side parallel with the basket side frame wall. Each additional plank will be locked into the previous plank by inserting the protruding locking plank side into the docking side of the previous plank at an angle and then pushing in and down to lock the two planks together. A diagram of locking two planks together is in Figure ____ below. With all planks of the court together and aligned evenly the court markings will be drawn. The basic court markings will be general professional basketball court markings spaced and drawn to scale on this court. These basic court markings include the middle division line, drawn to separate the five-foot side length in two. The center circle, where the ball and both robots are located at the beginning of a game. A semi-circle free throw line for each basket. Finally, the free throw lane is drawn which is a rectangle and semi-circle that touches from the free throw line to the basket wall. Any additional markings and colors will be added for the computer vision to be able to locate distances on the court. All court markings are shown and labeled in Figure ____ below.

4.3.3 Prototyping and Testing

Table 45 Court Testing

Requirement	Test	Required Equipment
R.A.7	Determine how level the flooring is in different conditions	Level
R.A.10	Determine if the court markings go back into the same place each time	Camera
R.R.B.4	Determine if the wheels slip on the floor	Wheel
R.A.7	Determine if the court lays flat inside of the frame	Level
R.A.9	Determine if the chosen wheel can roll over the frame wall	Wheel

4.4 Ball

The ball for this project represents a full-size basketball. However, it is much smaller scale and must be throwable by a small-size robot. The ball should not be heavily affected by aerodynamic forces to ensure repeatability. That is, the ball should not be so light that a small gust of wind would affect its motion. Aerodynamic drag is expected and will likely be utilized to gain lift based on the amount of spin on the ball. It should also bounce on the court but not all over the arena from a single throw. This is to prevent the ball from being too difficult to pick up and to prevent the ball from landing on a portion of the robot that it gets stuck on. The ball should be nearly spherical so that it has consistent rolling and launching. It cannot be deformable to the point that a force on the ball causes a permanent dent.

4.4.1 Research

4.4.1a Ping Pong Ball

Ping pong balls are 40mm in diameter and weigh about 2.7 grams. They are made from a thin plastic shell that is made of a material to meet a required bounce standard. The standard states that the ball should bounce “25 cm when dropped from 30.5cm.” This bounce is very significant and could lead to significant issues with the robot collecting the ball. However, the ball is very light and could be launched very easily.

4.4.1b Small Tennis Ball

A typical tennis ball is a bit too large for the scale of the robot. However, there are much smaller-scale tennis balls that exist for pets. This introduces a small difficulty as there are not standards related to the size and material. Thus, additional research must be conducted after the ball is picked and purchased because the material properties could differ from the documentation provided. The typical size for these tennis balls are about 1.5” in diameter, a perfect size for the scale of the robot. Virtually any tennis ball of this size can be utilized, and there are hundreds of options that are offered in a variety of colors, themes, and prices.

4.4.2 Design

The small Tennis ball is chosen for this project. It is a small ball that has some weight and grip on it, and it has a fair amount of grippy material covering the rubber ball. The tennis ball chosen comes in a sports pack from PetSmart that contains one basketball themed ball. This provides a color with sharp contrast to the court so that it can be tracked more easily by the computer vision software. Additionally, it fits the theming of the game.

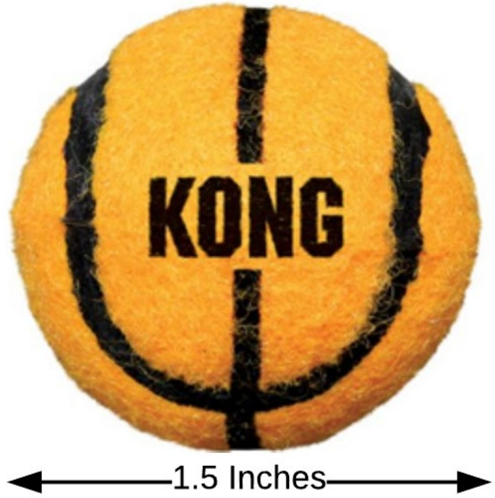


Figure 29 KONG basketball tennis ball chosen for this project
Pending permission

4.4.3 Prototyping and Testing

Table 46 Ball Tests

Requirement	Test	Required Equipment
R.A.3	Verify Ball size	Calipers
R.A.11	Verify ball weight	Scale
R.A.12	Test throw with different conditions including outside, inside, with different spins	Tape measure

4.5 Hoop

There are two basketball hoops located on either short side of the arena. The hoop is mounted and is set to a diameter that is feasible for the launcher to remain accurate under all conditions. Two hoops are chosen in order to maintain the traditional basketball feel. If two hoops are too many, it is easy to reduce back down to just one hoop in a half-court setting. Each time a basket is made, the score for the game must be updated, thus the hoop must sense when a ball makes it all the way through. It is possible that the ball goes halfway in and pops out, so the sensor must be designed such that it is resilient to false positives (I.E debouncing). The hoop structure should maintain the appearance of a basketball hoop including a backboard, a rim, and a net. Each of the pieces must remain sturdy when the ball inevitably misses and hits the structure. It should be designed in such a way that improves accuracy. The hoop size can increase or decrease based on robot performance, and the backboard should be angled in such a way that increases accuracy. The hoop can be broken into 4 sub-components: ring, post, backboard and sensor. The ring is the actual loop that the ball falls through, the post is the mounting interface for the backboard and ring, the backboard is the solid face that the ball can bounce in from, and the sensor determines when a goal is made.

4.5.1 Research

4.5.1a 3D Print

The hoop can be designed in SolidWorks and 3D printed in PLA or ABS plastic. This allows for easy integration between the hoop, hoop frame, and sensor technology by giving full control over the size, shape, and design of the hoop. 3D printers are readily available at UCF or by team-members with a variety of bed-sizes and printable materials. Thus, the actual cost of the print depends directly on the amount of print material required, and whether or not the design can be printed on a particular printer. However, the major disadvantage of the 3D print design is that the printer may not print the exact size or shape that is designed. It is common for prints to warp, bend, or shrink in the process of printing. Further, the strength of the design strictly depends on the material used and printing properties used such as infill density and infill pattern. Printing larger objects can also take up to days long which may affect the viability of the process.

4.5.2b Metal

A simple metal hoop can be utilized to fulfill the requirements for the hoop. Any metal material such as aluminum or steel can be utilized to form a ring. Additional hardware for mounting the ring, sensor, and post is necessary. Metal is sturdier than the 3D prints even at smaller sizes so it will be more resilient to impacts than the 3D prints regardless of diameter. The strength of the hoop design depends mostly on the interface between the ring and the post, as most of the force of an impact will go into a moment about the interface. It is most likely that the ring would bend downwards to the post upon impact.

4.5.1c Infrared Gate (Break Beam Sensor)

An infrared Gate utilizes Infrared light transmitter and receiver to determine when an obstacle is placed in the path between the transmitter and receiver. When the object blocks the light, the value of the receiver changes and that change is interpreted as a pulse by the microcontroller. The length of the pulse indicates how long the object has blocked the gate. Ultimately this shows whether or not the object actively passed through the hoop without bouncing out. These devices are relatively low cost and easy to set up. They are also contactless meaning they will not interfere with the object passing through the gate. Some key factors in determining the practicality of a particular gate is whether or not the beam can travel far enough to reach the receiver within the hoop, the width of the beam so that if the ball is not perfectly center the beam will still be broken, and the resilience to noise of the sensor. These sensors are dramatically affected by the amount of ambient light in a scene, thus outdoor use may affect performance.

4.5.1d Ultrasonic

An Ultrasonic sensor utilizes sound to determine the distance to objects. This sensor is like the IR gate in that it can detect when an obstacle passes in front of it by constantly determining the distance to a known plate on the opposite side of the hoop. Again, this can be interpreted as a pulse by the microprocessor and an appropriate response to the pulse can be executed. These sensors are more expensive and more difficult to work with than the Infrared Gate despite giving the same advantages. This sensor is not affected by ambient light.

4.5.1e Limit Switch

A limit switch can be utilized to detect if a ball has passed by opening/closing a digital circuit when interacted with. The major advantage of this is that the ball can be detected in a single direction from an angled switch. However, the device is contact-dependent thus it directly affects how the ball passes through the hoop. Similar to the previous devices, the digital output can be interpreted as a pulse and an appropriate response can be executed. This sensor is the most resilient to noise and environmental conditions.

4.5.2 Design

The final design is a combination of the metal and 3D print considerations. The metal ring is the sturdiest material and structure, but it suffers from poor interfacing with the post, backboard, and sensor. Thus, the metal hoop interfaces with a 3D printed bracket that integrates the sensor and backboard. The chosen sensor is the limit switch due to the ability to work in all environments, and it naturally prevents the problem of the ball bouncing from below the hoop being counted as a score. It is also the cheapest and easiest device to integrate into the rest of the project. The backboard is made out of polycarbonate to prevent warping or damage over continued use. All 3D printed parts are printed with a high infill density in ABS to maximize strength.

4.5.3 Prototyping and Testing

Table 47 Hoop Tests

Requirement	Test	Required Equipment
R.A.13	Test if the hoop is mounted securely and can take X force	Frame, Weight, Ball
R.A.14	Test if the ball can fall through the hoop	Ball
R.A.E.8	Test the sensors and verify accuracy	Ball
R.A.6	Test if the hoop is put into the same place each time the court is put together	Frame

4.6 Display and Sounds

The arena contains a visual display unit, like a TV, a monitor, or a tablet, that is used to relay information to the players. The display unit needs to be capable of clearly showing the settings page for the game, like a dashboard on a video game console. This page will be used to set up new player robots, game mode, playback options, the score of the game, the current period out of four total periods, the remaining time for the current period, and to adjust the sounds for the game. The display unit will also display the live action 2D top down position on the court in a game engine, this is so the player can glance at important game information on the screen and not lose their place on the court and can continue driving. Showing the live location on the court is also useful for spectators of the game that might not be able to see in the arena. It will also need to be capable of displaying debugging and development information such as the live computer vision feed for any debugging that might need to happen during a game.

When searching for a display that will work for these tasks there are some features that will need to be considered. A high definition or super high definition display will be ideal for spectators being able to see the information from a far distance very clearly. For this same reason a larger screen size is also preferred. In addition, only widescreen monitors will be evaluated so when the court, which is a rectangle, and robot location is displayed it can take up the whole screen space instead of making it smaller to fit on a square screen. The higher refresh rate on the display the better so that the game and settings will look smooth. Lower refresh rates might make the picture look choppy which can affect where the player thinks their robot is in respect to the court. Another consideration for choosing displays is how well it can display in daylight conditions. Should the game need to be played outdoors or near a window during daylight hours there may be too much ambient light to see the display.

There will be sounds enabled with the game and therefore speakers, either connected to the display or separate entities that will need to be able to supply loud enough sound for both players and spectators to hear. Sound is necessary for this project as it will supply feedback to the user as well as add an emersion element to the game. An additional feature that will be taken into consideration is the ability of the speakers sounds to be mixed with tactile feedback to enable a person who is blind to enjoy the game as well. In this case the speakers would have to produce adequately loud sounds in conjunction with the tactile feedback such as announcing location and orientation on the court. Finally, the price of the display and speakers should be reasonable for a self-funded college group of four to adequately purchase.

4.6.1 Research

4.6.1a Monitor

There will be two categories of monitors examined and researched, those with speakers and those with no speakers (sold separately). Regardless of which category is chosen the total price for this section of the arena should be less than 70 USD. The screen of the monitor should be no less than 18”. The display will be showing a live 2D position of the robot on the rectangular court. The monitor will need to display the camera feed for debugging. For these two reasons the display chosen should be widescreen to adequately scale the rectangular arena.
Any monitor with built in speakers must have an HDMI or DisplayPort connector to be considered, as these are the two best options for showing HD video and playing sounds through one plug. DisplayPort will be prioritized higher than HDMI for its superior video quality capability. The lowest refresh rate on monitors today is adequate for this project and so will not be a consideration. The weight of the display is taken into consideration as the arena must be portable. Table _______ shows the different monitors for consideration, their size, price, ports, and weight.

4.6.1b Speakers

The speakers for the arena need to be loud enough for the spectators to hear the game sounds and mountable or embedded into the display for portability. The cost of the speakers will also need to be low to meet the arena display and sound budget. For non-embedded speakers there are many choices available. Generally, all non-embedded speakers will be loud enough for our needs and are relatively inexpensive, starting at roughly $10.

4.6.2 Design

The final design for the display and sounds will use a combination monitor with embedded speakers that will be capable of mounting to the frame of the arena or stood up alongside the arena. The monitor is the ______ model which is ____ inches in size and weights ____ pounds. The ___(model)__ monitor can display in ____ resolution at ____ Hz. This meets our restriction for displaying the game dashboard, settings, and simulated 2D view of the arena. Additionally, the embedded speakers are well loud enough and can be heard from ____ feet away perfectly at max volume. The display can either be mounted on a post attached to the side of the arena or can be placed on the ground in the front of the arena depending on if the arena is on a table or not.

4.6.3 Prototyping and Testing

Table 48 Display and Sound Test

Requirement	Test	Required Equipment

Display Size	Measure the screen from the bottom corner to the top diagonal of the opposite corner.	Tape measure
Widescreen	Is the screen size 16:9?	Windows Laptop
Viewable Outside	View the display outside in a covered area with the correct settings. Stand in the player position and observe if the screen is clearly visible.	Windows Laptop
High Res	Use a program and run it on the display to determine the resolution	Windows Laptop
Distance Viewing	Turn on the display to the proper settings. Walk backwards until the display can no longer clearly be seen. Measure this distance to the display.	Windows Laptop
Refresh Rate	Use a program to test the actual refresh rate of the display.	Windows Laptop
Spectators hear sound	Play game sounds at max volume and continue moving backward until the sound can longer clearly be heard. Measure this distance to the arena.	Windows Laptop

4.7 Camera

The camera for this project is used for computer vision to track the robots, ball, and goal. It will need to be very accurate to ascertain the exact position of the robot in comparison to the goal so that the robot can make the goal within accuracy requirements. The camera will be placed above the arena a certain distance so that it may see the entire arena, robot, and goal without moving. The camera needs to supply clear video and bright colors along with fast speed so that we may update locations in real time, as accurately as possible. It will need to determine the robot's orientation in the arena so that we may use this to turn the robot toward the goal when the gamepad’s shoot button is pressed. In addition, the camera will need to be fast enough to track the ball going through the goal so that we may register a point and trigger the replay on the display. The camera will connect to the controller directly, so it must have compatible connections and firmware to be able to achieve this.

4.7.1 Research

4.7.1a Pixy2

The Pixy2 is a small camera that comes with computer vision and tracking built in making it an excellent choice if it can perform the necessary tasks adequately. The Pixy2 uses an Aptina MT9M114 image sensor capable of displaying video at 1296x976 resolution at 60 FPS, which in theory should be perfectly fine for our application. The camera has a 60-degree horizontal and 40 degree vertical field of view. With this field of view the camera would have to be mounted six feet above the arena to have a full view of the entire court. The arena is only three feet high, so a six-foot mounting height is a detriment in terms of aesthetics. Additionally, at six feet high the camera might not be able to distinguish and track the objects it needs to. The Pixy2 uses a color-based object detection algorithm that should be capable of following a ball or a shape that we design for the robots. It also has built in 20 lumen lights to keep the vision area cleanly lite at all times. The Pixy2 uses an NXP LPC4330 204MHz dual core processor with 264Kb of RAM and 2Mb of flash memory. It will consume roughly 140 mA of power with either a 5V USB input or an unregulated 6V-10V input. The Pixy2 outputs data through either a UART serial, SPI, I2C, USB, digital, or analog connection. This variation in output data connections is useful because depending on the controller we use, there may not be enough of a certain port on said controller for all items to plug into if everything uses USB or UART.

4.7.1b Logitech C922x

The Logitech C922x is a wide view full HD webcam. This camera was chosen to research because one of our team members owned it, there are other possible better options to research, but to stay in budget for the arena we will attempt to use parts already owned. For documentation purposes the Logitech C922x is sold for $100. This camera can produce a full high definition resolution of 1080p at 30 fps or 720p at 60 fps. This resolution is quite adequate for computer vision and is the best of the three researched cameras. Additionally, the C922x has an autofocus feature which is good for tracking quick moving objects. The camera is also wide view having a 78-degree field of view. This means the camera can be mounted at a minimum of 6.4 inches above the court in order to view the entire court without moving. However, the camera will be mounted at a minimum distance of three feet above the ground because it has to go over the arena wall. At three feet above the court the camera will be able to see 28 feet of a surface, five times the amount that needs to be viewed. This makes the center vision perfectly clear for the computer vision, but also makes it more likely that something outside the arena could be detected. This will have to be considered when designing the computer vision with this camera and further testing is needed to accurately tell if the extended view will actually cause problems.

4.7.1c Logitech C270

The Logitech C270 is a standard HD webcam. Like the previous Logitech camera this camera was also chosen for research it because it is already owned by a team member. For documentation purposes the Logitech C270 is sold for $40. This camera can produce a high definition resolution of 720p at 30 fps. This frame rate might not be good enough to follow fast moving objects like the ball flying, but further testing is needed to discern this. In conjunction with lower ability to track quick moving objects this camera only has a fixed focus which makes the previously mentioned quick moving objects harder to track. The Logitech C270 has a field of view of 60 degrees, meaning that in order to see the whole court it will have to be mounted 1.4 feet above the court. Again, since the camera must be mounted at a minimum of three feet above the court in order to go over the wall the camera will be able to view 10.4 feet of space. This is more than enough of a distance to view the entire court and will make the center vision clearer, however, things outside the arena could potentially impact the object detection and tracking of the computer vision. This will have to be considered when designing the computer vision with this camera and further testing is needed to accurately tell if the extended view will cause problems.

4.7.2 Design

For the camera design we first chose to use the Pixy2 as it simplified the computer vision object detection and tracking. However, after testing the Pixy2 it was discovered that it would not work for this project as it was incapable of detecting unmoving objects from six feet above the ground (the height needed to view the entire court) at a reasonable rate. The cameras actual video input quality was also very low, requiring many lights to make the court bright enough for even slight object detection.

The C922x camera will be the camera used in the final design and was chosen for its ability to see the entire court from well below three feet due to its widescreen camera. It’s also the best quality resolution of the three cameras researched. The two downsides of the C922x are the use of a USB connection for power and data transfer which will take up one of the few USB slots available on the arena controller and the fact that we will have to now write the computer vision software for the camera.

4.7.3 Prototyping and Testing

Table 49 Camera tests

Requirement	Test	Required Equipment
(Clear vision)	Set the camera up above the ground at the correct height. Place all objects being used in the camera vision. Are all objects clearly distinguishable?	Webcam, Varying sized objects
(Color vision)	Set the camera up above the ground at the correct height. Place varying known colors in the camera field of view. Is each color correct and distinguishable?	Webcam, Varying colored objects

(Field of view)	Set the camera up above the ground at the correct height. Observe and record the distances from each corner on the ground.	Webcam, measuring tape, location markers
(Compatible with other parts)	Read the camera documentation and the controller documentation to ensure the parts will be compatible	Camera and Arena controller documentation

4.8 Gamepad

The gamepad will be the first thing a player will touch when playing this game, so it's important to choose a gamepad that will feel familiar. When choosing the gamepad our team felt that an often overlooked, but important feature is tactile feedback. Tactile feedback aids in the feeling of control over the robot and adds another level of response to the player so they feel like their driving has an impact on the game.

Our communication between the arena processor and robot will be accomplished through Bluetooth. We anticipate that this communication will need to be very fast to make driving the robot feel good and reactive. Because of this we are keeping the amount of information sent over Bluetooth to the lowest amount possible and using a wired gamepad will remove information needed to be communicated over Bluetooth. The gamepad could also communicate using WIFI direct, but we have opted for wired because adding WIFI direct will add an additional module that will need to be purchased, which could break the budget requirement. Choosing a wired gamepad will also add a layer of reliability. If we use a Bluetooth gamepad and we are having problems with driving, is that a problem with our Bluetooth or our code for driving? We are eliminating the possibility of errors occurring from wireless communication.

Additionally, the gamepad should be easy to write code for and have thorough documentation. This will make working with the gamepad quick and easy and allow us to focus our efforts into other parts of the project. For these reasons we decided to pick between two popular gamepads; the Xbox One wired gamepad and the PlayStation 4 wired gamepad.

4.8.1 Research

4.8.1a Xbox One

The Xbox One wired gamepad is one of the most widely used gamepads for computer based and robot-based applications. Therefore, the Xbox One gamepad has a lot of documentation, especially for robotic applications like ours. The Xbox One gamepad was developed with comfortability in mind when holding the gamepad for long periods of time. Therefore, the gamepad fits comfortably in the hand while also allowing the user to be able to hit any button and any button combo with ease. This gamepad features ten digital buttons, a syncing button, two analog triggers, two analog sticks, and a digital D-pad. The two triggers feature independent rumble motors (Impulse triggers) that can be programmed to vibrate directionally. This rumble will be useful for giving the user an in depth experience, such as rumbling harder and harder while spinning the flywheel up to launch the ball when not in autonomous mode. The right side of the gamepad contains four of the ten digital buttons; the green ‘A’, red ‘B’, blue ‘X’, and yellow ‘Y’ buttons. These buttons are useful for main actions like ‘Choose’ or ‘Go Back’. The left and right side also contains one of two analog sticks each, these also contain a digital button activated when the analog stick is pressed in. Analog sticks are very important for driving and directional aspects of controlling the robot. In the center of the gamepad is two more digital buttons and the syncing button, generally used for pausing, menu, and turning the gamepad on and off. The left side of the gamepad also contains a digital D-pad generally used for choosing options quickly. Located on the shoulders of the gamepad are the two more digital buttons generally referred to as “bumpers”. Finally, the back shoulders of the gamepad each have one of two analog triggers. These triggers have the rumble feature and therefore can be used for processes that require feedback to make the game feel more natural. All together the Xbox One gamepad contains sixteen possible buttons, many more than this project should need to make it feel good to the player.

4.8.1b DualShock 4

The DualShock 4 is the gamepad used for the PlayStation 4. The DualShock 4 is not typically used in many robotics operations. The DualShock model line of gamepads has kept its design similar for many years, which could be seen as an advantage to players who have used this gamepad since the first generation, which was released well before the first-generation Xbox gamepad. The DualShock 4 is smaller gamepad compared to the Xbox One gamepad. It also contains two vibration motors, one inside the left handle and one inside the right handle. The right handle motor is smaller and less powerful than the motor on the left, this allows the vibration to vary based on what feedback the developer wants the player to feel. The DualShock 4 also incorporates a clickable two-point capacitive touch pad on the front along with motion detection through a three-axis gyroscope and accelerometer. The buttons on the DualShock 4 include two analog sticks, two analog triggers, two pressure sensitive buttons, ten digital buttons, and four directional buttons. Located on the right face of the gamepad are four of the ten digital buttons: green ‘triangle’, orange ‘circle’, blue ‘X’, and pink ‘square’. These are the main action buttons, such as ‘select’ and ‘back’. Also located on the right face is the right analog stick in addition to the fifth digital button activated by pressing the analog stick. Similarly, on the left face of the gamepad is the left analog stick and sixth digital button, again activated by pressing the analog stick inward. These analog sticks are generally used for movement, such as driving. On the left face of the gamepad is also located the four directional buttons: ‘up’, ‘down’, ‘left’, and ‘right’. These buttons are also generally used for movement tasks. On either side of the capacitive touchpad (located in the middle face) are the ‘options’ and ‘share’ buttons, which are two more of the ten digital buttons. On each side of the gamepad, located on the shoulder, lies the two pressure sensitive buttons, also referred to as “bumpers”. Lastly below each bumper on the shoulder of the gamepad are the two analog triggers, again which are usually used for performing action tasks like accelerating a car. The DualShock 4 gamepad is sold starting at $30.

4.8.2 Design

Between the two gamepads we believe the Xbox One wired gamepad will have the most documentation and support as well as ease of programming, thus it will be used for the final design. We are opting to use a wired controller for two reasons: lower the amount of information needed to be transmitted to the Bluetooth module and to add a layer of reliability. The Xbox One gamepad also has the individual rumble motors on each trigger button, which will add more immersion to the game. In full autonomous shooting mode, the right trigger will be used to launch the ball. When pulled, regardless of how hard, the flywheel will start spinning up which will enable the rumble feature, which will increase in intensity as the wheel spins faster and continue rumbling until the ball is launched. If autonomy is turned off the player will control the speed of the flywheel, this will be done by pressing and holding the right bumper button, the rumble in the trigger will begin just like in autonomous mode, but the ball will only be launched when the player pulls the right trigger. The left trigger will be used for intake and like launching the ball the flywheel will begin spinning and the left trigger will begin to rumble. This time the trigger will rumble while the fly wheel is spinning and stop either when the player releases the trigger, or the ball has reached the resting position in the launching mechanism. The left analog stick is used to main robot movement. When leaning the analog stick forward or backward the robot will move forward or backward. When leaning the analog stick left or right the robot will strafe left or right. All combinations of movement are supported as well: forward and strafe left or right, backward and strafe left or right. The right analog stick is used to rotate the robot. Moving the stick to the right rotates the robot clockwise and moving the stick to the left will rotate the robot counterclockwise. The green ‘A’ button is used as the ‘select’ button and the red ‘B’ button is used as the ‘back’ button. The ‘menu’ button is used for pausing the video game portion of the game to view video settings, exit the game, or restart the current game mode, in addition this menu is used if the player would want to invert their movement controls. This mapping is reset back to default every time the main dashboard is accessed. The ‘view’ button is used for pausing the game. The ‘Xbox’ button turns the gamepad on and off. The left bumper, D-pad, blue ‘X’, and yellow ‘Y’ button do nothing and will not have mapping. This control mapping should feel comfortable and natural to the player, whether they play with launching autonomy or manual launching mode and regardless of the players left hand or right hand preferability.

Table 50 Player input functions and gamepad mapping

Function	Type	# Of Axes
Forward/Backward + Strafing	Left Joystick	2
Rotation	Right Joystick	1
Launch Ball	Right Trigger	1
Flywheel Speed Control	Right Bumper	1
Intake	Left Trigger	1
Select	A Button	1
Back	B Button	1
Player/Game Settings	Menu Button	1
Pause	View Button	1
Gamepad Power On/Off	Xbox Button	1

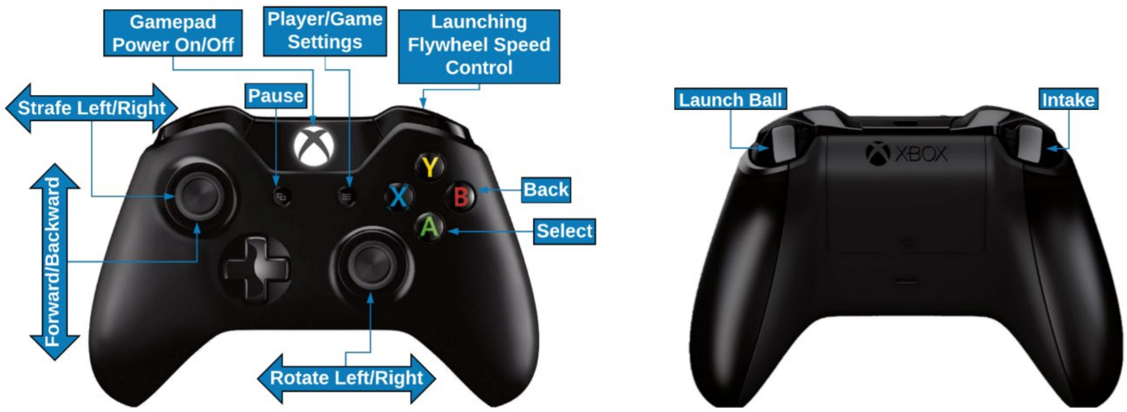


Figure 30 Gamepad control layout
Pending permission from Microsoft

4.8.3 Prototyping and Testing

To test the gamepad the following items are needed: the Xbox One wired gamepad, the robot, Bluetooth, arena controller, and video game portion need to be working and turned on. The first tests should be ran using launching autonomy, then the same tests shall be repeated for manual launching mode. Note that the right bumper button should only work in manual mode. Using the gamepad button mapping table and diagram press each button one at a time and observe that the correct function occurs. Ensure that when the triggers are pressed the rumble function works properly and is synced with the flywheel spin up. Next, change the driving controls to inverted and observe that the robot is still moving in the correct directions. Lastly, after inverting the driving controls and certifying that they work correctly, return to the main dashboard and back into a game. The driving controls should have reset to default, test this by driving the robot and observing that the drive controls are now back to default settings.

Table 51 Gamepad tests

Requirement	Test	Required Equipment

(Tactile feedback)	Set a function to rumble the gamepad on command. Perform the command and observe the rumbling feeling of the gamepad.	Gamepad, Gamepad software, Arena controller
(Button operation)	Ensure each button performs it's given task by pressing each button one at a time and observing the buttons output.	Gamepad, Gamepad software, Arena controller

4.9 LED Lights

The arena system uses computer vision to detect distance between the robot and the hoop. Based on these distances, the arena converts them into motor velocities for the robot to adjust and shoot the ball. The update rate of the computer vision system is 60Hz and therefore, the arena can perform calculations quickly. However, none of this is possible without proper illumination. This is where the LED lights play an important role.

Light Emitting Diode, or LEDs, are a common occurrence in present day. Therefore, due to such high-volume availability they are affordable. Using appropriate current limiting resistors and a microcontroller, one can turn them on or off in a timely fashion. Having a strip of them around the arena will not only illuminate the arena for computer vision, but also make it aesthetically entertaining. There are different colors of LEDs and they can be combined to form different colors by simply mixing their RGB values. Consequently, when the player makes a shot, a sensor will trigger a sequence of LED blinks and create an animation for user entertainment. Each action has an LED sequence preprogrammed into the arena. Making the shot causes the arena to turn green, a Bluetooth connection turns the arena blue whereas when pairing the arena blinks blue light. A lost connection or fault causes the arena to turn red.

LEDs tend to draw a lot of current to shine brighter and therefore, based on the kind used, their current and voltage requirements are used to calculate current limiting resistors. They surely can add an entertainment value to the project and make it more professional. There are multiple LED technologies available which are discussed in section 4.9.1 of this document.

4.9.1 Research

4.9.1a Adafruit NeoPixel

4.9.1b Traditional LEDs

4.9.2 Design

4.9.3 Prototyping and Testing

Table 52 LED Lights tests

Requirement	Tests	Required Equipment
R.A.16	Determine the voltage used by the LEDs to power on	Multimeter, Power Supply, Jetson Nano
R.A.16	Determine the current draw by the LEDs and contain their brightness using Ohm's Law	Multimeter, Calculator
R.A.16	Determine the animation sequence via timers	Oscilloscope, Serial Monitor
R.A.16	Determine logical value of each LED at a certain instance for debugging	Logic Analyzer, Multimeter, Oscilloscope

4.10 Controller

The controller for this project strongly depends on the computational power that is required by the various components. This controller performs calculations for computer vision, Bluetooth communication from arena to robot, calculations for robot location, calculations for force to launch the ball, and be able to show video on the display using the game engine. In addition, it will also control any LED lights that are installed in the arena. The controller will need to be capable of running an operating system to allow the use of certain software, like the computer vision and game engine software. For this reason, the controller will need to be powerful, but also compatible with the other parts chosen for the project. Lastly, the controller will need to have the proper slots for additional hardware that will have to be interfaced with.

4.10.1 Research

4.10.1a Raspberry Pi 3 Model B+

At the time of writing this paper the Raspberry Pi 4 was released. This is not being considered due to it being sold out. Instead the older generation Raspberry Pi 3 B+ will be researched. The Raspberry Pi 3 Model B+ uses a 1.4 GHz Broadcom BCM2837B0 Cortex A53 64-bit Arm8 processor, it has 1GB of SDRAM. The Raspberry Pi has wireless LAN, Bluetooth 4.2, and Bluetooth low energy capabilities. The Bluetooth 4.2 is what will be used to communicate with the robot and is a very important feature. It also has a HDMI port and DSI display port, important for connecting the display and sound. The Raspberry Pi 3 Model B+ also requires a micro SD card for loading an operating system and storing data, adding an additional expense to this controller. Additional ports on this controller include: Extended 40-pin GPIO header, CSI camera, four USB 2.0, and a 4-pole stereo output and composite video. The controller is powered by 5V/2.5A DC power, a standard amount for a controller of this type. The Raspberry Pi 3 Model B+ sells for $35. The Raspberry Pi 3 B+ was originally suggested because of its built-in Bluetooth capability, low price, and processing power, however, due to the inclusion of the game engine video display, we believe an onboard GPU will be necessary for smooth video.

4.10.1b Jetson Nano

The Jetson Nano does not however include a built in Bluetooth module so that must be purchased. In addition, the Jetson Nano supplies much more processing and computational power than the Raspberry Pi 3 B+ and for this reason a webcam for computer vision with a high frame rate could now be implemented.

4.10.2 Design

4.10.3 Prototyping and Testing

Table 53 Controller tests

Requirement	Test	Required Equipment
(Bluetooth Compatible)	Look at the documentation for controller and Bluetooth module.	Part documentation
(Processing Power)	Run game video, Computer vision camera software, LEDs, and Bluetooth module. Observe and record specs using software and visually look for lag in video playback.	Controller Monitoring Software
(Proper slots for parts)	Check that the correct number of slots and slot type is included in the controller.	Controller documentation
(Supplies enough voltage to parts)	Using documentation for the controller check that the supplied voltage is capable of supporting the part that will be plugged in.	Controller documentation

4.11 Communication

The communication subsystem allows the Arena to send commands to the robot. To accomplish this, the Arena must have a communication system on board and send data over a wireless link. The communication subsystem needs to have a data update frequency of 30Hz at the minimum. Failure to do so can cause latency in sending commands to the robot and thereby an overall latency in the system's response. This latency hinders the robot from shooting successfully 75% of the time as per the requirements.

The Arena is a master to all the robots in it. It takes in commands from the controller and the computer vision system and combines them into a packet in a systematic way. This packet is sent to the robot(s) via a Bluetooth link. The packed is designed by the team and passes multiple checks to ensure accurate transmission of data. The robot is a slave device to the arena that will receive data over the radio to perform its actions. The implemented communications protocol will also allow the robot to send its sensor data back to the arena for monitoring and debugging purposes. This data is shown by the Arena on a screen to give users more information regarding their robot. These stats could include current motor velocities, battery status, communication link status etc.

The communication system is low energy because it will allow the onboard computer to use its resources for high powered activities such as the LEDs and running the game engine.

4.11.1 Research

4.11.1a Intel Wireless-AC Wi-Fi/BT adapter

4.11.2 Design

4.11.3 Prototyping and Testing

Table 54 Communication tests

Requirement	Test	Required Equipment
	Determine that the communication system successfully form a connection with the slave devices	Bluetooth module, Bluetooth App
	Determine that the commination system successfully reads the packet generated by the Arena from a buffer	Serial Monitor, Bluetooth Module
	Determine that the communication system successfully transmits the packet	Oscilloscope, Bluetooth Module, Bluetooth App
	Determine that the communication system successfully receives a packet	Oscilloscope, Bluetooth Module, Bluetooth App
	Determine that the communication system saves system resources by going to sleep when no communication is required	Multimeter, Serial Monitor, Bluetooth Module

4.12 Electrical System

The Electrical System subsection defines how the Arena components are wired and work together. Supplying power directly to a microcontroller from the power outlet can be dangerous. Due to this proper AC to DC conversion is required meeting the operation requirements of the Microcontroller and Arena peripherals. The AC-DC converter needs to be energy efficient and provide at least 30 to 40 watts of power for the entire electrical network of the Arena. One can design such a converter with enough time and resources. However, they also act as constraints for our purpose due to which an AC-DC converter is purchased instead. Overall, time vs cost analysis was done by the team to arrive at this conclusion.

Typically, microcontrollers run on 3.3 – 5 V Transistor-Transistor Logic (TTL) and so, the AC-DC converter needs to be able to step the power down to that voltage. Additionally, the NeoPixel LEDs, which are used to light up the arena, work on 5V input. The microcontroller can consume up to 4-6 amps of current as it is going to support the D-Pad controller, output video to the TV, and update the LED colors. The LEDs consume up to 2 amps of current when all of them are turned to full brightness. The TV monitor will run on AC output voltage. This allows the team to eliminate the need for a Printed Circuit board for the Arena thereby reducing the cost.

The electrical wiring will be hidden in a box which is attached to the arena. This electrical panel will give the arena a professional look and keep the electronics safe from potential damages caused by human interaction and carelessness. A power surge is required to power the monitor, Jetson Nano, and the LEDs. These technologies are further discussed in section 4.14.1 of this document.

4.12.1 Research

4.12.1a UPS / Surge Protector

The Power surge is attached to the arena to power the microcontroller, the NeoPixel LEDs and the display screen. For simplicity, the team decided to have one cable from the arena go into the power outlet. This also serves the arena requirement of being portable. To do so, a surge protector seems like an ideal option. One cable from the surge protector will go to the wall whereas all the components will connect to the surge protector. The surge protector needs atleast 2 outlets: one for the display and one AC-DC converters to power the LEDs and the microcontroller. However, depending on the current consumption and equipment protection, the LEDs and the microcontroller might use two different power adapters. There are multiple surge protectors available in the market with varying features and a comparison between them can be seen in Table 55.

Table 55 Comparing Surge Protectors

Brand	*Belkin*	*AmazonBasics*	*TonBux*
Number of Outlets	12	6	4

USB Ports	-	-	4
Length (inch)	15.6	11.9	12.2
Width (inch)	6.10	2.20	2.44
Height (inch)	2.10	1.75	1.26
Maximum Output Current (A)	15	15	16
Weight (lbs)	2.1	1.1	1.6
Energy Rating (J)	3940	200	1700
Cost ($)	24.99	11.49	33.99
Purchase Link	Amazon	Amazon	Amazon

As seen in Table 55, different surge protectors provide different benefits. The arena requirement states the surge should be able to support at least three plugs for the microcontroller, the LED and the display screen. However, it also depends on the type of connectors used to power the equipment. The microcontroller can be powered from a DC barrel jack connector, using GPIO pins, or using a standard micro USB cable. However, each of them provides different amounts of current to the system based on which the DC barrel jack is chosen as it provides adequate amount of current required to run Jetson Nano along with its peripherals. The NeoPixel LEDs can be powered with either standard USB type A or an AC-DC adapter. The specifics of these connectors and adapters will be discussed in more detail in sections 4.12.1b and 4.12.1c but this gives an idea on how many outlets and/or USB ports does the power surge need to support at the minimum.

The surge provided by Belkin has 12 outlets and can provide a maximum of 15 amps of current. The energy rating is 3940 joules and it costs approximately $25. The dimensions of this surge protector are 15.6 x 6.10 x 2.10 inches and it only weighs 2.1 lbs. However, it is quite big in size and it will not attach well to the arena making it a bad choice from an aesthetic standpoint. An important requirement for the arena is that it needs to be portable. Due to this the surge protector needs to be thin enough to able to be glued to the frame allow portability and ease of use. The surge protector provided by Amazon Basics consist of 6 outlet and can output 15 amps of current at maximum. The dimensions of this product are 11.9 x 2.2 x 1.75 inches and will definitely attach to the arena effectively. However, the energy rating is only 200 joules and therefore, it cannot protect against high voltage surges. Lastly, the surge protector by TonBux is a sure upgrade from Amazon Basic and Belkin but comes are a higher cost. It has built in wifi that connets to an app allowing toggling over the air. It has 4 outlets and 4 usb ports and can supply at most 16 amps of current. The energy rating is much higher than Amazon’s Surge but less than Belkin and it has a market price of approximately $34. The dimensions of this supply fit our needs however, it is the most expensive option out of all the three surge protectors.

4.12.1b AC-DC Adapters and Peripheral Connections

This component is required to convert power from a standard US 120V 60 Hz outlet to DC power. This component must be highly efficient; thus, it will be purchased. A single outlet is expected to support the entire Arena subsystem including TV display, Controller, and other loads. A household power strip will be attached to the frame to split AC power from the outlet to the TV Display and AC-DC adapter systems. The total power between the two must be calculated to ensure a single outlet is not tripped, however the TV display is separate from the AC-DC adapter power requirements.

The AC to DC adapter will power the Jetson Nano Controller which sends data to TV display using HDMI and communicates with D-Pad controller using USB. The TV Display needs 3 amps of current and the D-Pad controller uses less than 0.5 amps of current. Therefore, Jetson Nano needs at least 4 amps of current to power all its peripherals easily. The DC barrel jack can support 4 amps of current at 5 V which is more than enough required to run the system effectively. The AC-DC used for Nano is specified in the datasheet provided by NVIDIA however, that component is obsolete. Due to this, the technical parameters of the said component were studied, and an equivalent AC-DC converter was chosen. This converter has a 5.5 x 2.1 mm barrel jack connector that is compatible with Jetson Nano and successfully converts 100-240V to 5 V and can output a maximum of 5 amps of current.

The NeoPixel LEDs also require a 5V adapter however, the current requirements are at most 2 amps in case when all the LEDs will be lit up. This scenario is highly unlikely mainly because the LEDs are used for animation purposes and therefore, they will never use their maximum current. This can be used as an advantage and help save cost. The LEDs can be powered by Jetson Nano’s 5V pin which at maximum load can output 1.5 amps and at minimum load outputs maximum current of the power supply. Therefore, a 5V 4A barrel jack connector could be used to run the TV, D-Pad, Camera and NeoPixels. On the other hand, if Nano cannot supply enough current to the NeoPixels then there are two possibilities. One, they can run at low current and will be less bright and second, an additional 5V 25W AC-DC converter adapter can be used to power the LEDs and the data cable can be connected to Nano’s GPIO to address and program the LEDs.

The peripherals are interconnected using different connectors and exchange data using drivers that are built into Linux’s Kernel. The TV Display uses HDMI to receive and display data. The D-Pad sends controller commands via USB whereas GPIO pins and PWM is used to address and program the NeoPixels. The camera data is exchanged using USB as well. The Bluetooth modules is connected using a special M.2 Key E connector which is built into Jetson Nano and does not require any purchase. This information is also summarized in Table 56 in an organized fashion. The GPIO pins use approximately 0 Watts of power because they provide high impedance signal to their respective sensor/device.

Table 56 Arena I/O Schedule

	Type	Connected Devices	Connection Type	Power Requirements
Bluetooth	Intel Module	Microcontroller	M.2 Key E	~ 10 W
LED’s	NeoPixel	Microcontroller	GPIO	~ 0W
Display & Sound	TV	Microcontroller	HDMI	< 5W
Gamepad	Xbox controller	Microcontroller	USB	~ 2.5W
Gamepad 2	Xbox Controller	Microcontroller	USB	~ 2.5 W
Switch	Digital	Microcontroller	GPIO	~ 0 W

4.12.2 Design

Based on the research conducted in section 4.12.1 an overview of the Arena Electrical Network can be seen in Figure 31. The block diagram shows that the Arena will mainly be powered by a single outlet. The outlet will power a 4 port, 4 USB power surge to which the display monitor/TV and a 5V 15A AC-DC converter adapter is plugged in. The AC-DC converter has a barrel jack connector that powers the Jetson Nano, the brain of the arena. The barrel jack also powers the LED strips that go around the arena and perform animation for entertainment purposes.

The Jetson Nano will consume 4 amps of current while the LEDs will consume up to 2 amps. The camera, D-Pad, and HDMI plug into Jetson Nano using their appropriate peripheral connector cables and they all run on 5V input. Three GPIO pins on the Jetson Nano are used: one for addressing the NeoPixel LEDs, second for controlling Limit Switch 1, and the third for controlling Limit Switch 2. The Limit Switches are used to detect when the ball makes into the hoop. Second Limit switch is added for redundancy. For addressable NeoPixel LEDs the GPIO pin has to provide pulse width modulation signal as that is a requirement of its drivers.

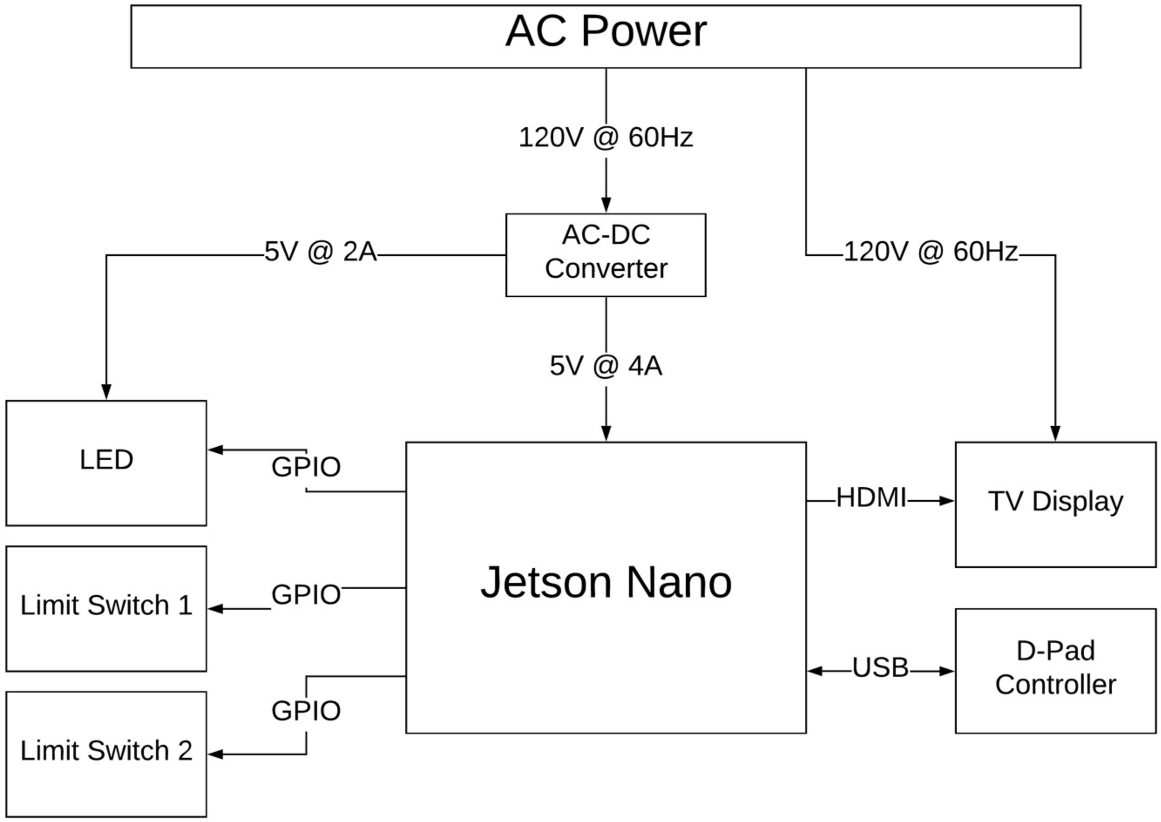


Figure 31 Arena Electrical Network Block Diagram

Table 57 AC-DC Convertor Tests

Requirement	Test	Required Equipment
	Test if the AC-DC adapter regulates voltage at the correct value	Voltmeter, active load
	Test if the AC-DC adapter can support the required loads value	Voltmeter, active load
	Test the AC-DC adapter efficiency	Voltmeter, active load

4.12.3 Prototyping and Testing

The electrical system will be tested using multimeter and oscilloscope. The LEDs work on Pulse Width Modulation signal and therefore, an oscilloscope will be used to decode the signal generated by the controller. A multimeter will allow to check the voltages and currents at input, output of the controller and all the peripherals allowing the team to make sure no excessive current or voltage spikes occur with the potential of damaging the system and harming the user. These tests will also verify the constraints specified in 2.4 Realistic Design Constraints section of this document. Table 58 shows how these constraints will be tested.

Table 58 Electrical System Tests

Requirement	Test	Required Equipment
C.A.3	Determine that the power surge supports all the adapters	Multimeter
C.A.3	Determine that the power surge plugs into the wall	Eye Test
R.P.3 R.P.4	Determine that the controller, peripherals, and the sensors are within acceptable voltage and current ranges	Multimeter, Oscilloscope, Datasheets

4.13 Computer Vision

The computer vision portion is one of the most vital aspects of this project. It will detect and track the location of all moving objects and defining court features. It will be capable of distinguishing between different robots and supply an accurate location to be used in other portions of the project, most notably, shooting the ball. This information will then be used for multiple parts of the project. The most important part is for calculating the force or angle needed to shoot the ball into the hoop. The game engine will also use the information from the computer vision to show the robots on the court in an overhead 2D visual representation that is a part of the Game system. For these reasons, the computer vision will need to be good enough to distinguish between robots and track them if one should be obscured by another. It will also simultaneously need to detect and track the ball of the court. All these calculations must be done rapidly such that the robot's control loop can be updated with adequate accurate information. If the information is outdated or inaccurate, nearly all systems in the project suffer.

4.13.1 Research

4.13.1a OpenCV

4.13.1b Neural Net

4.13.2 Design

4.13.3 Prototyping and Testing

Table 59 Computer Vision Tests

Requirement	Test	Required Equipment

(single Robot detection)	Place one robot on a blank, white or black, surface with the webcam directly overhead at the correct height and attempt to identify the robot.	Robot 1, Webcam, Arena Controller
(multi Robot detection)	Place two robots on a blank, white or black, surface with the webcam directly overhead at the correct height and attempt to identify both robots simultaneously.	Robot 1, Robot 2, Webcam, Arena controller
(single Robot tracking)	Place one robot on a blank, white or black, surface with the webcam directly overhead at the correct height and attempt to track the robot while in motion from one point to another.	Robot 1, Webcam, Arena controller
(multi Robot tracking)	Place two robots on a blank, white or black, surface with the webcam directly overhead at the correct height and attempt to track each robot from one point to another simultaneously.	Robot 1, Robot 2, Webcam, Arena Controller
(Obscured robot tracking)	Place two robots on a blank, white or black, surface with the webcam directly overhead at the correct height. Drive the robots so that robot one obscures the vision from the webcam of robot two. Observe that after robot two is unobscured that it is still being correctly tracked.	Robot 1, Robot 2, Webcam, Arena Controller
(ball tracking)	Place the ball on a blank, white or black surface, with the webcam directly overhead at the correct height. Move the ball and observe if the ball is being tracked while moving.	Ball, Webcam, Arena controller
(location accuracy)	Make two markers at a known location at a known distance apart. Place one robot on the surface at marker one with the webcam directly overhead at the correct height. Observe the output of the computer vision matches the known location. Drive the robot to marker two. Observe the output of the computer vision matches the known location.	Robot 1, Webcam, Arena Controller
(Tracking on court with multicolored surface)	Perform all previous tests, but now using the arena laminate flooring as the surface with the webcam the correct height above.	Robot 1, Robot 2, Ball, Webcam, Arena controller

4.14 Peripheral Software

The peripheral software involves all the software related to devices and hardware for the arena. This includes driving the LED lights, communication through Bluetooth, hoop sensing, and robot control. This software also needs to directly communicate with the Game system; however, this interface is discussed more thoroughly in section 6.5 Arena – Game. This code runs on the microcontroller selected for the arena; thus, it is not as critical to design memory efficient code. A strict structure is not required to achieve high performance, thus object-oriented design is more appropriate than functional design. However, the architecture of the code is dictated by the libraries available to achieve the functionality the software requires. The code in this section needs to quickly process the data from the other systems and generate outputs to maintain some semblance of real-time control. Data from the gamepad input in the Game System sent to the peripheral system and then finally sent and processed by the robot can introduce a huge amount of latency, particularly in the arena-robot interface.

4.14.1 Research

4.14.1a C++

4.14.2 Design

4.14.3 Prototyping and Testing

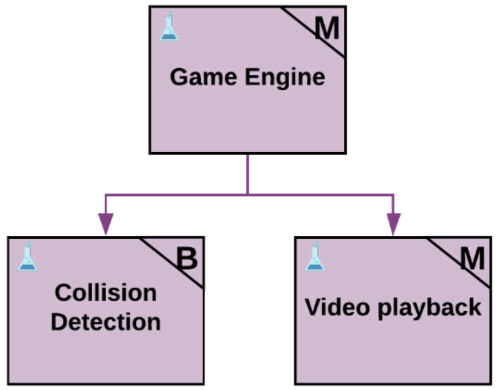
Table 60 Peripheral Software tests

Requirement	Test	Required Equipment
R.A.17	Communication Unit Tests	Bluetooth Module, power supply, Nano
R.A.17	LED Unit Tests	Nano, Power supply, LED's
R.A.17	Sensor Unit Tests	Nano, Power Supply, Sensors
R.A.17	Full software tests	Nano, Power Supply, terminal monitor

5.0 Game System

The Game system harnesses the power of a game engine to deploy commonly used features that are available in a virtual environment. The project requires players to be able to adjust settings, start and stop timers, display scores and other feedback, and assist the user by showing a 2D virtual representation of an environment. The game system is essentially the primary software arm of the

Arena system, but it can be developed and act independently from the Arena system. This system is the most feature-scalable system in the project. A large number of extra software functionality can be added to the project through the game system. These things include different robot settings based on a chosen player. This feature could adjust speeds, accuracy, or force limits to vary the player experience. Additional control logic such as autonomy or machine learning could be introduced into the game system to change the player experience quite dramatically.



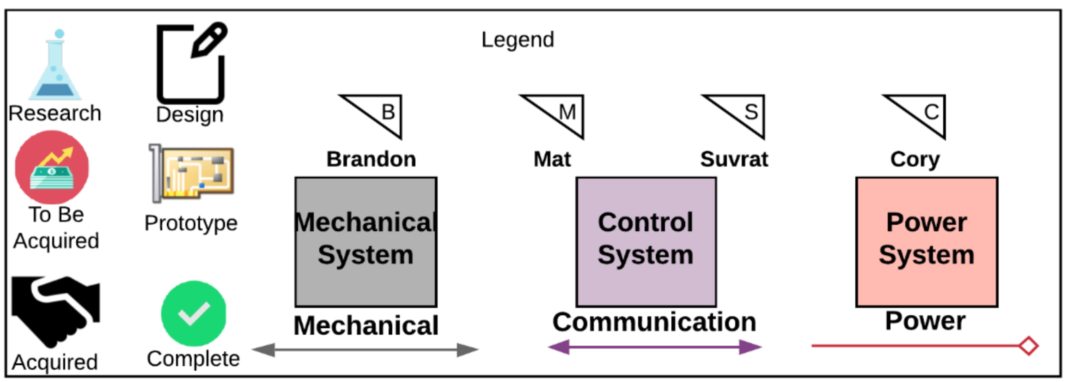


Figure 32 Game system block diagram

5.1 Game Engine

The game engine will be responsible for a few tasks overall and will act as a hub for data to flow in and out of. It must be able to handle data visualization such as showing a mockup of where the physical components such as the robot and ball are on the field or playing an animation when a shot is made or missed. Using data sent from the field and robot, the game engine will also handle collision detection and send feedback to the gamepad being used by the player. The game engine will also need to be capable of both 2D and 3D animation to accomplish its tasks.

5.1.1 Research

5.1.1a Unity

Unity is useable for both 3D and 2D games and simulation. Most of the group has used Unity before so there is some experience there. A major plus is that the base edition of this engine is free to use. Due to that Unity is under consideration for being used for both the 2D visualization and tracking as well as the animation after a made shot. Unity looks to be a primarily 3D based engine and there seems to be more material for tutorial in 3D rather than the 2D side. A downside of unity is that the UI can get rather cluttered and unusable. It also has a rather steep learning curve if the developer is just learning how to use Unity with C#. On top of that, there are solutions, but no simple answer to doing inter process communication between Unity and other things, such as a C++ program or Python script.

5.1.1b Godot

Godot seems to be slightly opposite of Unity in that it looks more 2D friendly than 3D. It comes with a lot of tools to help a first time Godot developer get started creating what they need to. Like Unity, it has its own suite of animation tools for the developer to use instead of using a separate software such as Cinema4D or Maya. As well as supporting C++ and C#, Godot also has its own language, GDScript which is a lightweight Python-like language. Godot in general is a more lightweight program and requires less resources to run. This may prove to be useful as the resources to run the game system through whatever controller is chosen for use may be limited. Godot, like Unity, has an asset hierarchy that dictates how and what items are allowed to interact with.

5.1.2 Design

The GUI for the game aspect of the project will be created using the Godot 2D gaming engine. It will consist of 4 main screens that are laid out in Figures 28-30 below. The main menu is the screen that will be shown first when the game is first initialized (Figure 28). It will consist of 3 options in the form of buttons for the users to pick, Play, Controls, and Exit.

The first one is “Play” which will bring the player to another option screen. The Play screen’s options consist of easy and hard mode. Easy mode will lead into a game where the robot will handle the different aspects of shooting for the player, such as power of the shot and making sure the robot is lined up with the basket. Hard mode will disable these assistive options and allow the player full control over the robot. The second option of the main menu is to bring up the controller menu. This screen will contain an image of the controller labeled with its mappings (Figure 25 in Section 4.8.2) as well as two checkboxes. One checkbox is to invert the x-axis of the controllers and the other is to invert the y-axis of the controllers. This allows the user to reconfigure the controller to match where they are standing around the arena to give them the easiest and least confusing control of the robot. The last option is to simply exit the game. Figure 29 contains a flowchart on how the screens are accessed

After the player selects their preferred mode of play, easy or hard, the game will start, and the screen will display a representation of the actual game arena. This screen contains all the elements on the physical field, the robot, the ball, and the hoop. All three will be tracked by our computer vision program, and the positions will refresh in real time to display an up to date position of the game components. This will also allow the hoop to be placed wherever the owner wants to configure it and have it still accurately shown in the simulation aspect of the game. The other two components are a match timer that displays the time remaining in the game, and a counter to display the score the player has accrued by making baskets. Baskets will also be scored by the distance the shot occurred from in order to provide more of an incentive to make shots from farther away.

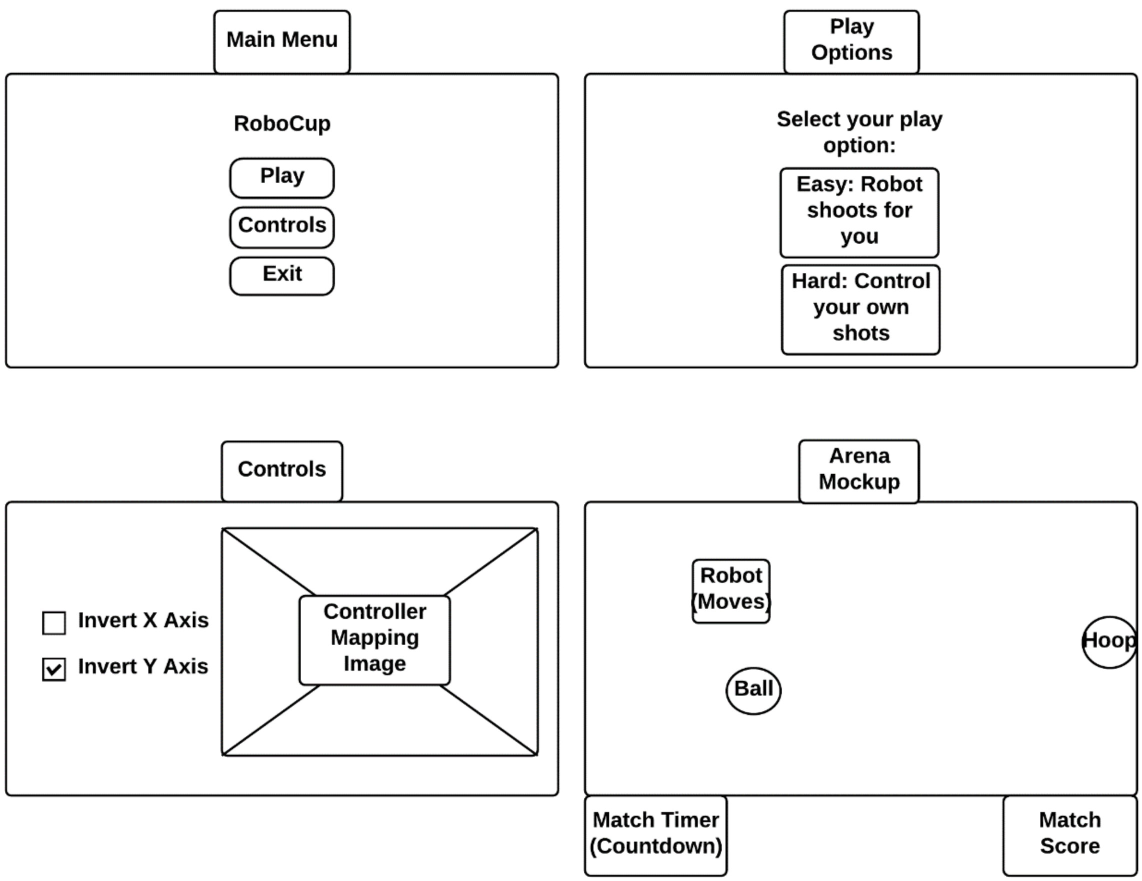


Figure 33 Wireframe screen layouts Screens of the game system

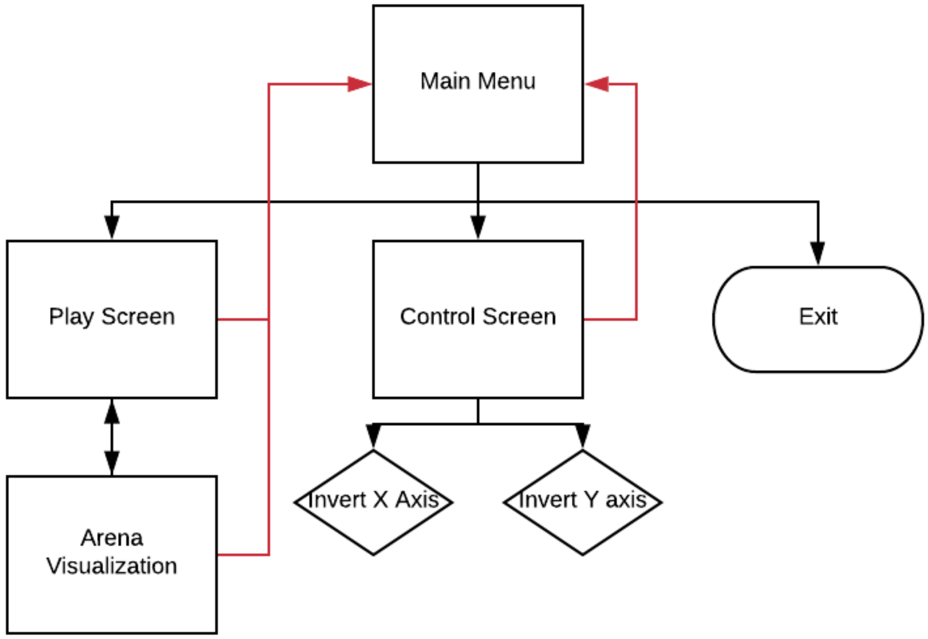


Figure 34 Flowchart for screen navigation

5.1.3 Prototyping and Testing

Prototyping will be done using Godot. The first objective for prototyping the game system will be to create a base scene that takes the shape of a rectangle to act as the arena. On this scene there will be a few shapes that will act as the ball, rim, and robot. After the scene is set up, the first 2 tests from Table 40 below will be able to be tested. The tests will be conducted in the order of the requirements fulfilled as each requirement is a concrete subcategory of the game system. Verifying that the game submodules work effectively together will require the rest of the subsystems work first, therefore it doesn't explicitly fall under a requirement listed in section 2.5

Table 61 Game engine tests

Requirement	Test	Required Equipment
R.G.1	Moving shapes programmatically	Laptop
R.G.1	Moving shapes via gamepad input	Laptop
R.G.2	Check data is displayed properly	Laptop
R.G.3	Check date and time are accurate	Laptop
R.P.4	Verify game submodules work together in the intended fashion	Laptop

After the tests are validated, the parts will be combined into one scene that contains the visualization for the court, as well as the data display parts and rechecked to make sure the components still function properly. Once that is verified it will be made into the official design.

5.2 Collision Detection

The game engine system is responsible for protecting the robot in events of poor user input. For example, if the player constantly runs into the wall, the robot would either drive over the wall and flip, or it would burn out the motors and cause electrical or structural damage. Another instance that requires collision detection is when two robots run into each other. Again, these events could cause electrical or structural damage and prevent consistent playing. In both instances, the collision detection should be aware when a robot is entering a zone that could be dangerous and protect the robot. The protections could be reducing motor power, slowing the robot, or preventing input entirely. Another useful feature of collision detection is automatic intaking when the ball is near the front of the robot. This is a player-assist feature that can have adjustable settings.

5.2.1 Research

5.2.1a Game-Engine Collision Detection

Collision detection is possible with both Godot and Unity, and is done in a very similar way in both engines, with the game objects being designated as collision objects, and then monitoring the different objects in order to check if they are overlapping, and sending a signal when two objects are found to be overlapping.

5.2.1b Optimization

5.2.1c Collision Response

5.2.2 Design

The collision detection will be set up in such a way that as the robot moves closer to the designated wall area of the arena, the controller will begin to vibrate, and the intensity of the vibration will increase the closer that the robot's position to the wall is. This is accomplished by layering bands of detection objects in a procedural manner leading up to the perimeter. Each band will be assigned a value for vibration that will be triggered upon the robot's sprite in the visualization entering its area. It is important that there is a reliable scale between the visualization and the actual arena. If this is not the case, the controller may vibrate for no reason, or not vibrate when it should be doing so. It is also possible to attempt to send a command to the robot to not allow it to move in a specific direction, preventing the continued attempt to move into a wall, which potentially can damage motors and components.

5.2.3 Prototyping and Testing

Table 62 Collision Detection Tests

Requirement	Test	Required Equipment
R.G.8 R.P.4	Verify Safety System	Robot, Display, Frame
R.G.6	Verify Collision Avoidance algorithms	Robot, Display, Frame
R.G.6	Verify accuracy of simulation versus physical	Robot, Camera, Court. Tape Measure

5.3 Video Playback

It is very common to have a replay of events that happened prior to a score in any sport. When a player scores a goal, it would be exciting and useful for spectators to see the motions of the robot and ball in the time leading up to the robot shooting the ball. This requires a storage buffer containing the positional data of the robots and ball, and timing for ball entering the hoop. At the time of scoring, a short playback of the positional data (in 2D) and then a pre-rendered 3D animation of the ball being launched and going into a hoop play. This is very similar to what bowling centers do for different types of pins being knocked down. The pre-rendered 3D animation reduces complexity of the simulation while still providing the feeling of experiencing the goal again.

5.3.1 Research

5.3.2 Design

5.3.3 Prototyping and Testing

Prototyping of the video playback does not require the actual video that will be played to be done in order to be completed. The team can substitute any video to use for testing purposes and just swap it with the correct rendered animation once it is complete. The testing will follow an order outlined below consisting of unit tests that slowly scale up until we get the full project.

Table 63 Video Playback tests

Requirement	Test	Required Equipment
R.G.5	Manually trigger any video	Laptop
R.G.5	Make sure our video is rendered properly	Laptop
R.G.5	Manually trigger our rendered video	Laptop

R.G.5	Trigger scene on basket score/non-score	Nano

6.0 Subsystem Integration

The system integration section identifies high-risk interfaces that must be carefully designed and tested to avoid problems that occur when multiple systems are designed in parallel. Three robot interfaces are identified, and two major system interfaces are identified. The robot interfaces are high-risk because they are the most likely point of failure in the project, and the entire project depends on the robot's capabilities to be completed appropriately. Although these components exist within the same subsystem, their critical risk status elevates the importance of integration. The system interfaces are not within the same subsystem; thus their integration is not discussed within their respective system discussions. As such, the interfacing between the major systems is developed in this section.

6.1 Base – Intake

The Base-Intake integration is identified as the mounting interface between the base subsystem, and intake subsystem. This interface ensures the compatibility between the intake and the wheel locations, and the existence of mounting locations for the intake to be attached to the base. The ball must be able to be picked up from the ground and in various orientations around the court. Corners are particularly difficult for the intake to reach in, so the intake-base integration must ensure that the intake can reach the ball from each orientation at each position in the court.

6.1.1 Design

The intake requires the ball to go to a particular location without getting stuck. Thus, the design involved for this component is a shovel/gate type apparatus that directs the ball into the correct location. The intake is the same as the launcher thus the mounting and cuts are the same as in section 6.2 Base – Launcher. This design does not solve the problem of picking up the ball from corners as the intake is located within the frame. In the event that this becomes a larger problem, additional designs / components will be introduced.

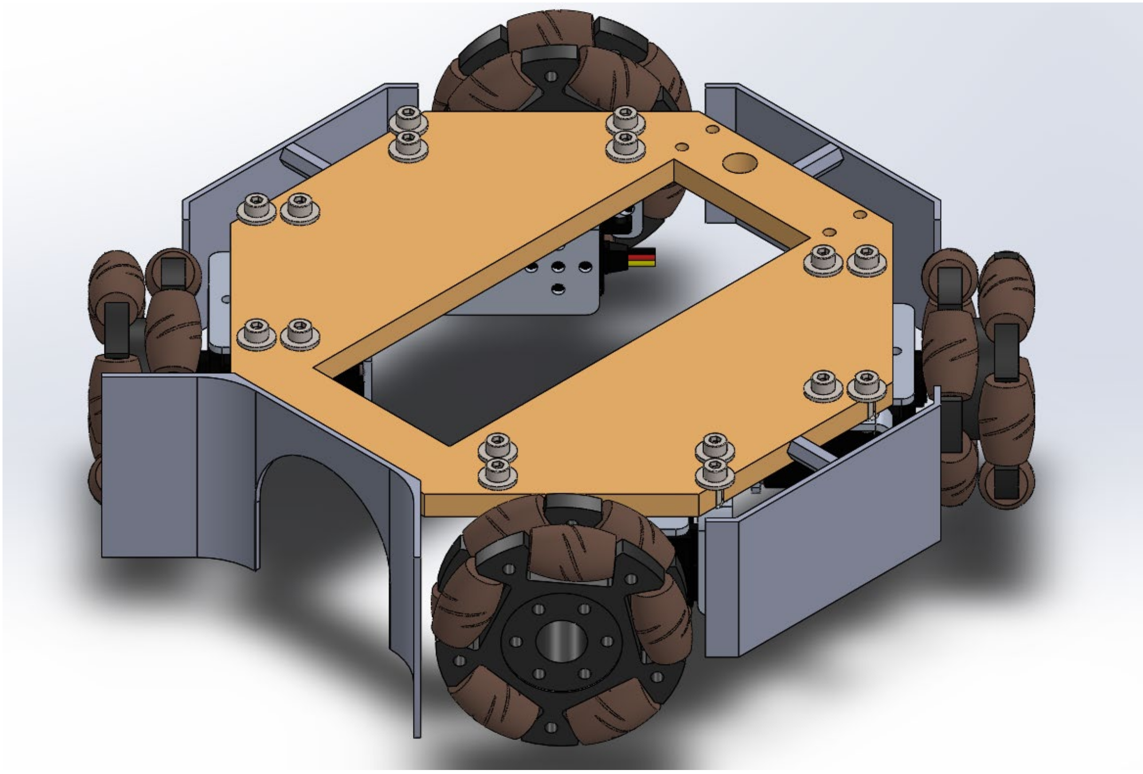


Figure 35 Ball-Prevention plates

6.1.2 Prototype and Testing

Table 64 Base-Intake integration tests

Requirement	Test	Required Equipment
R.P.4	Does the intake mount securely to the Base?	Base, Intake, hardware tools
R.P.4	Does the intake reach the ground to pick up the ball?	Base, Intake

6.2 Base – Launcher

The base-launcher integration is identified as the mounting interface between the Base subsystem and Intake subsystem. This interface ensures the compatibility between the base and the launcher, including the existence of mounting locations for the launcher to be attached to the base, and clearance for the launcher mechanisms to fully actuate. In particular, the launcher must be able to fully extend or retract the slide, and the release linkage must be able to fully engage or disengage the gear.

6.2.1 Design

The Base-Launcher integration design consists of a cutout for the wheel and track mechanism, and mounting holes for the various subsystem components required to operating the systems. This includes mounting holes for the lever servo, and the wheel motor bracket. The wheel size and cutout are variable such that the best sized-wheel can be printed or adjusted after additional testing. However, the cutout must be small enough that the frame remains strong despite the large hole in the center.

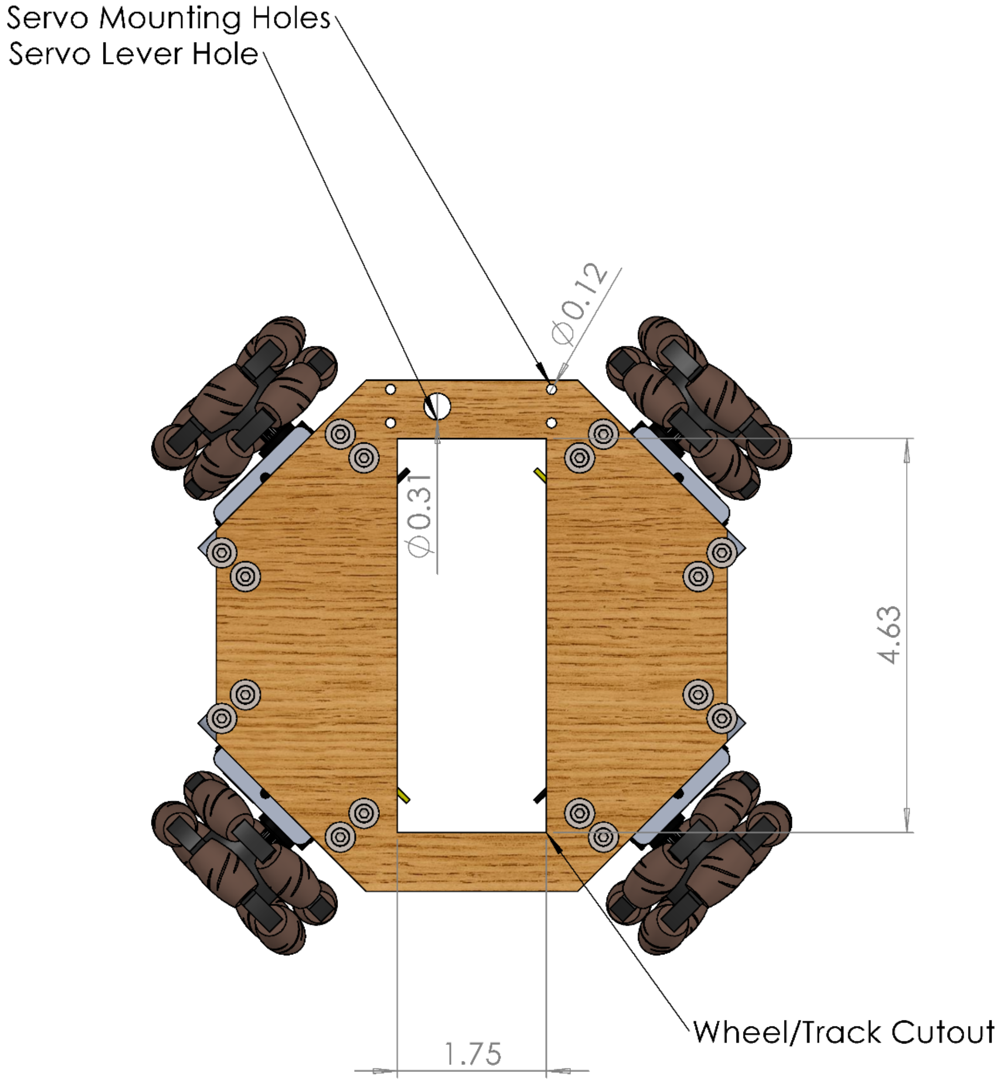


Figure 36 Base-Launcher Integration

6.2.2 Prototype and Testing

Table 65 Base-Launcher Integration tests

Requirement	Test	Required Equipment
R.P.4	Does the Launcher mount securely to the Base?	Base, Launcher, Hardware tools
R.P.4	Can the launcher slide fully extend and retract?	Base, Launcher
R.P.4	Can the launcher release fully engage or disengage the gear?	Base, Launcher

6.3 Intake – Launcher

The Intake-Launcher integration is identified as the design interface between the intake and launcher such that the intake places the ball in the correct location each time for the launcher to hit consistently. Further, the intake must not interfere with the launching mechanism; in particular, the launching mechanism must be able to fully extend or retract regardless of the location or orientation of the intake.

6.3.1 Design

Currently the design we are going with for the Intake-Launcher integration is going to incorporate them into the same component. We will be using a single large flywheel that is lowered close to the ground to be able to grab the ball off the floor. After the ball enters the mechanism, it will slot into a trapdoor to wait to be fired. This will allow the wheel to spin freely without moving the ball. When the player is ready to shoot, the wheel will spin up to the correct speed and the trapdoor mechanism will be reversed through the use of a servo or a similar piece of hardware. Once the ball contacts the wheel again, it will continue along its path and be shot out the other end. It is important that whatever is used to reverse the trapdoor has a high enough torque rating to keep the ball and wheel from pushing back out against it. Setting up this integration this way will allow the team to utilize both passive and active mechanisms to make the overall component use less pieces.

6.3.2 Prototype and Testing

Table 66 Intake-Launcher Integration Tests

Requirement	Test	Required Equipment
R.P.4	Does the intake place the ball into the correct location for the launcher?	Intake, Launcher, Power Supply
R.P.4	Can the launcher shoot without interference from the intake at any rotation?	Intake, Launcher

6.4 Camera-Arena

The Camera-Arena integration component is defined as the interface between the camera and arena. Specifically, the mounting of the camera such that the camera's field of view does not prevent the camera from seeing all of the components on the field. This directly affects the mounting height of the camera.

6.4.1 Design

6.4.2 Prototype and Testing

6.5 Arena – Game

The Arena-Game integration involves interfacing between the Arena system and the game system. The game system requires position data of the robots and the ball from the camera stationed above the arena to accurately update the locations of the simulated versions in the game engine. Additionally, the game system must send the gamepad data to the arena system to process the player input's and send them out to the robots.

6.5.1 Design

Inter-process communication can be accomplished in a variety of ways between scripts and programs of different languages. The two options the team were most comfortable with were TCP/IP sockets and shared memory. Shared memory is much easier with a C++ or Python based environment while TCP/IP sockets would be better for use with something like C# and Unity.

6.5.2 Prototyping and Testing

The Arena-Game interface prototype simply requires the Arena and game systems to be completed. Once the systems are completed, a single software section must be built and tested to interface the Arena and Game software systems.

Table 67 Arena-Game Integration Tests

Requirement	Test	Required Equipment
R.P.4	Send and verify signals between arena and game	N/A
R.P.4	Check visualization matches data sent from the arena	N/A

6.6 Robot - Arena

The Robot-Arena integration is identified as the interface between the components of the Robot and Arena. For example, the ball must interface correctly with the

Intake subsystem, and launch subsystem. The fiducials for computer vision to track on the robot are designed in this section. The robot and arena share information via Bluetooth. Arena uses camera information and controller inputs, combines them into a packet and sends it to a buffer. The Bluetooth system reads the buffer periodically and sends the commands to the robot. The robot then parses the packet into useful information and carries out the commanded tasks.

6.6.1 Design

6.6.2 Prototype and Testing

Table 68 Robot-Arena Integration Tests

Requirement	Test	Required Equipment
R.A.2 R.P.4	The robot and arena can communicate bidirectionally	Robot, Arena, Terminal
R.A.17 R.P.4	The arena can control the robot	Robot, Arena, Terminal
R.P.4	The computer vision system can track the robot's position	Robot, Arena, Terminal, Display

7.0 Administrative

Overhead is required as project size increases. The overhead involved for this project relates to task management, scheduling, budgeting, and communication. Each of these is necessary to achieve the requirements set forth by the team.

7.1 Budget and Bill of Materials

Table 69 Robot Budget

Item	Price (USD)	Quantity	Subtotal (USD)
Launching Hardware	$ 20.00	1	$20.00
Drive Hardware	$ 30.00	1	$30.00
Intake Hardware	$ 20.00	1	$20.00
Intake Motor	$ 15.00	1	$ 5.00
Drive Motor	$ 20.00	3	$60.00
Launch Motor	$ 20.00	1	$20.00
Controller	$ 20.00	1	$20.00
Battery	$ 30.00	1	$30.00

PCB	$ 20.00	1	$20.00
Bluetooth Module	$ 10.00	1	$10.00
Voltage Converter	$ 15.00	1	$15.00
5x Motor Controller	$ 13.00	1	$13.00
Total per Robot			$273.00

Table 70 Arena Budget

Item	Price (USD)	Quantity	Subtotal (USD)
Frame Hardware	$ 100.00	1	$ 100.00
Camera	$ 40.00	1	$ 20.00
Controller	$ 100.00	1	$ 100.00
Power Supply (AC-DC)	$ 20.00	1	$ 20.00
PCB	$ 20.00	1	$ 20.00
Bluetooth Module	$ 10.00	1	$ 10.00
Ball	$ 5.00	1	$ 5.00
Court Hardware	$ 25.00	1	$ 25.00
Voltage Converter	$ 15.00	1	$ 15.00
LEDs	$ 25.00	1	$ 25.00
Gamepad	$ 25.00	2	$ 50.00
TV Display	$ 70.00	1	$ 70.00
Total			$ 460.00

Project Total for 1 Robot: $733, Project total for 2 Robots: ~$1000

Table 71 Bill of Materials for Project

Item	Budget Item	Price (USD)	Quantity	Subtotal (USD)
Arduino Uno	Robot Controller			
L298N	Motor Controller			
PCA9685	Motor Controller			
MCP23017	Motor Controller			
Servo	Launcher Motor			
Stepper	Launcher Motor			
Robot Kit	Drive Hardware, Drive Motor			
DC-DC Convertor	Voltage Converter			

Bluetooth module	Bluetooth Module (Robot)			
Intel Module	Bluetooth Module (Arena)			
Level Shifter	PCB			
Jetson Nano	Arena Controller			
SD Card	Arena Controller			
DC Power Supply	Power Supply (AC)			
SD Card Reader	Arena Controller			
DC Power supply 2	Power Supply (AC)			
Arena hardware	Frame Hardware			
Frame Material	Frame Hardware			

Table 72 Bill of Materials for Manufacturing and Reproducing

Item	Budget Item	Price (USD)	Quantity	Subtotal (USD)
PCA9685	Motor Controller			
Servo	Launcher Motor			
SD Card	Arena Controller			
PCB	PCB			
Jetson Nano	Arena Controller			
Bluetooth module	Communication			

7.2 Milestones

Figure 37 is a Gantt chart that shows the various major milestones and project timelines required to successfully complete the project in Senior Design 1 and Senior Design 2 courses. The major critical path is that of the PCB design, purchase, and fabrication due to the long lead time to purchase and build the PCBs.

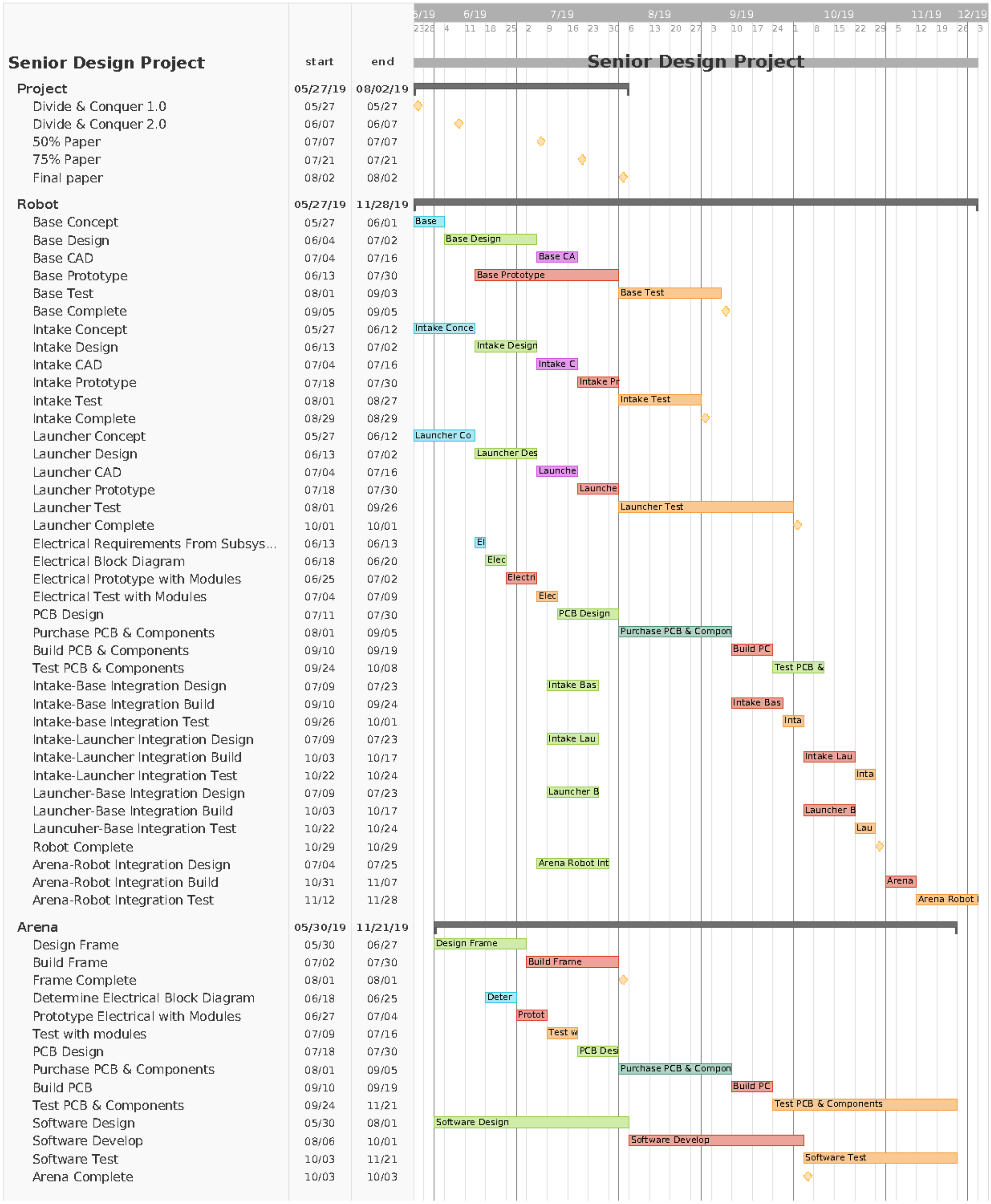


Figure 37 Gantt Chart indicating critical milestones and work timelines

7.3 Communication

Communication is critical to the project team's success. A thorough use of documentation and sharing tools allows the team to work at peak efficiency regardless of physical location or project timeline. Although there are many tools available to achieve this, three critical tools that can seamlessly link together are utilized to reduce the number of sites or applications to download.

7.3.1 Microsoft SharePoint

SharePoint and other Microsoft products are used for this project because it can act as a one-stop shop where all the materials required for the project can be found. SharePoint itself is a website platform that has various pages and plugins. Each research topic has its own page that the team fills out as the research and design is completed. This allows all our research to be compiled real time and is organized such that information can be retrieved when necessary. The plugins utilized within SharePoint include Microsoft Planner, a tool that allows users to add tasks with information like assignee, due date, and relevant files. The tasks are tracked as cards that can be moved around with order of importance, or have reminders set so that things are finished on time. Everything with the SharePoint is synced and stored on OneDrive, Microsoft's cloud storage platform. This allows for version control of all the documentation required for the project.

7.3.2 Discord

Discord is a free VoIP software that provides chat, screen-sharing, file-sharing, voice and video calls in an easy to use platform. Discord is chosen over Slack, Skype, and other chat software because it provides the required features for free, it is stable, and the team has utilized it for other projects in the past. This tool provides us a way to store any text messaging between members of the group and return to it at any point in the future.

7.3.3 GitHub

GitHub is a cloud application that integrates with the git version control scheme. The team can work on their local machine and develop any files or software required, and then when finished, upload the file to the cloud that other members can update from. The tool is very powerful when simultaneously working on the same file because git can merge different versions of the file based on changes made. This is particularly helpful in software that are modularized into functions or blocks that multiple members can work on simultaneously without losing progress in another block.

8.0 Project Summary and Conclusion

Appendix I Copyright Permissions

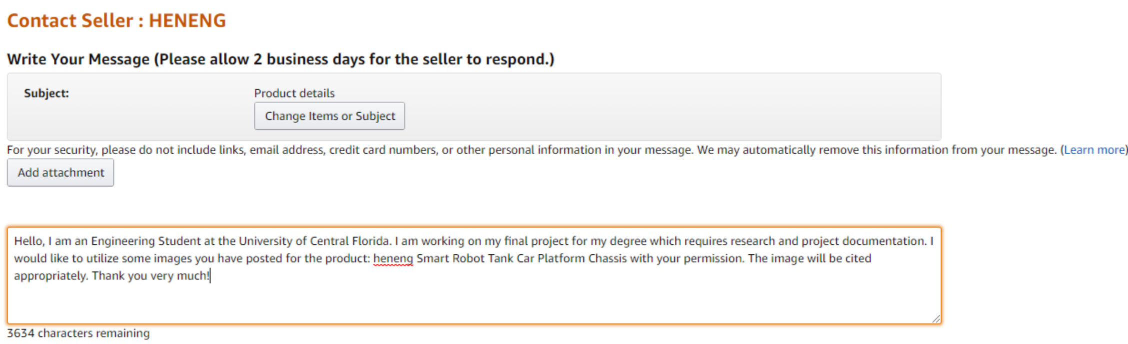


Figure 38 Heneng Permission Pending

Appendix II Data Sheets

References

[1] "Researchgate," [Online]. Available: https://www.researchgate.net/figure/Typical-RoboCUP-robot_fig3_4083675.

[2] "Robocup," [Online]. Available: https://www.robocup.org/.

[3] " Youtube VEX," [Online]. Available: https://www.youtube.com/watch?v=A8daR6qBw3M.

[4] "Youtube Stanford," [Online]. Available: https://www.youtube.com/watch?v=fXsB7fXcWO8..

[5] Genstattu, "The LiPo Battery Characteristics and Applications," [Online]. Available: https://www.genstattu.com/blog/the-lipo-battery-characteristics-and-applications/.

[6] "Electropedia," [Online]. Available: https://www.mpoweruk.com/nicad.htm.

[7] "Battery and Energy Technologies," [Online]. Available: https://www.mpoweruk.com/leadacid.htm.

[8] "Atmel Studio," [Online]. Available: https://www.microchip.com/mplab/avr-support/atmel-studio-7.

[9] "Arduino IDE," [Online]. Available: https://www.arduino.cc/en/main/software.