



UNIVERSITY OF CENTRAL FLORIDA

Robot Basketball

Divide & Conquer - Senior Design I, Group 9, Summer 2019

Brandon Gross • Electrical Engineering

Suvrat Jain • Computer Engineering

Cory Ricondo • Computer Engineering

Mathew Schneider • Computer Engineering

Project Narrative:

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.

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.

Requirements, Constraints and Standards:

Constraints

Table 1 Project constraints

The project shall...
1. Cost no more than \$1000
2. Be transportable in a standard-sized sedan
3. Be designed by August 2, 2019
4. Be built and tested by November 15, 2019

Table 2 Arena constraints

The Arena shall...
1. Cost no more than \$500
2. Be powered by a standard US 120V 60Hz wall outlet
3. Be able to rest on two standard folding tables
4. Utilize a custom PCB

Table 3 Robot constraints

The Robot(s) shall...
1. Cost no more than \$300
2. Utilize a custom PCB
3. Be powered by a battery

Requirements

Table 4 Project requirements

The project shall...
1. Contain two high-level subsystems capable of communication: Arena and Robot
2. Allow a human-player to control the robot-subsystem to drive and launch a ball
3. Take efforts to ensure safety of both human players and subsystems

Table 5 Arena requirements

The Arena shall...
1. Be no larger than 2 meters length, 2 meters width, and 1.5 meters height
2. Weigh no more than 75 lbs. total
3. Contain at least 1 rubber ball that is no smaller than 1.5" diameter
4. Contain at least 1 basketball hoop no smaller than 1.5" diameter
5. Have flat ground with scale basketball court markings
6. Contain a display of at least 17 inches with 720p Resolution capable of playing sounds
7. Communicate with the robot subsystem at a rate of at least 50Hz
8. Support a camera for top-down view of the court
9. Support an Embedded Controller capable of running a traditional Operating System
10. Convert voltage from 120V AC to 12V DC
11. Convert voltage from 12V DC to 9V and 5V DC
12. Support at least two gamepads
13. Support vision-based position tracking of the ball and robots in the court with update rate of at least 60 Hz

Table 6 Robot requirements

The Robot(s) shall...
1. Weigh no more than 8 lbs.
2. Communicate with the arena subsystem at a rate of at least 50Hz
3. Be capable of holonomic locomotion
4. Contain at least 3 Drive motors
5. Contain a single motor for launching mechanism
6. Contain a launching mechanism capable of launching a 1.5" diameter rubber ball
7. Contain an intake mechanism for acquiring a 1.5" diameter rubber ball from ground level
8. Traverse in one direction at minimum 0.3 m/s
9. Be powered by a 12V battery
10. Convert voltage from 12V DC to 9V DC and 5V DC
11. Support an embedded controller capable of processing controls for a minimum 5 motors
12. Maintain at least 75% shot accuracy from anywhere on the court

Standards

Table 7 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

Similar Projects

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

Robocup competition (<https://www.robocup.org/>) 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.

VEX Robotics Nothing but Net (<https://www.youtube.com/watch?v=A8daR6qBw3M>) 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 involved several unique launching mechanisms and locomotion systems for a basketball-like challenge.

Stanford's 2015 battle of the bots(<https://www.youtube.com/watch?v=fXsB7fXcWO8>). 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 proposed for this project. This challenge provides a baseline to compare our robots against.

House of Quality

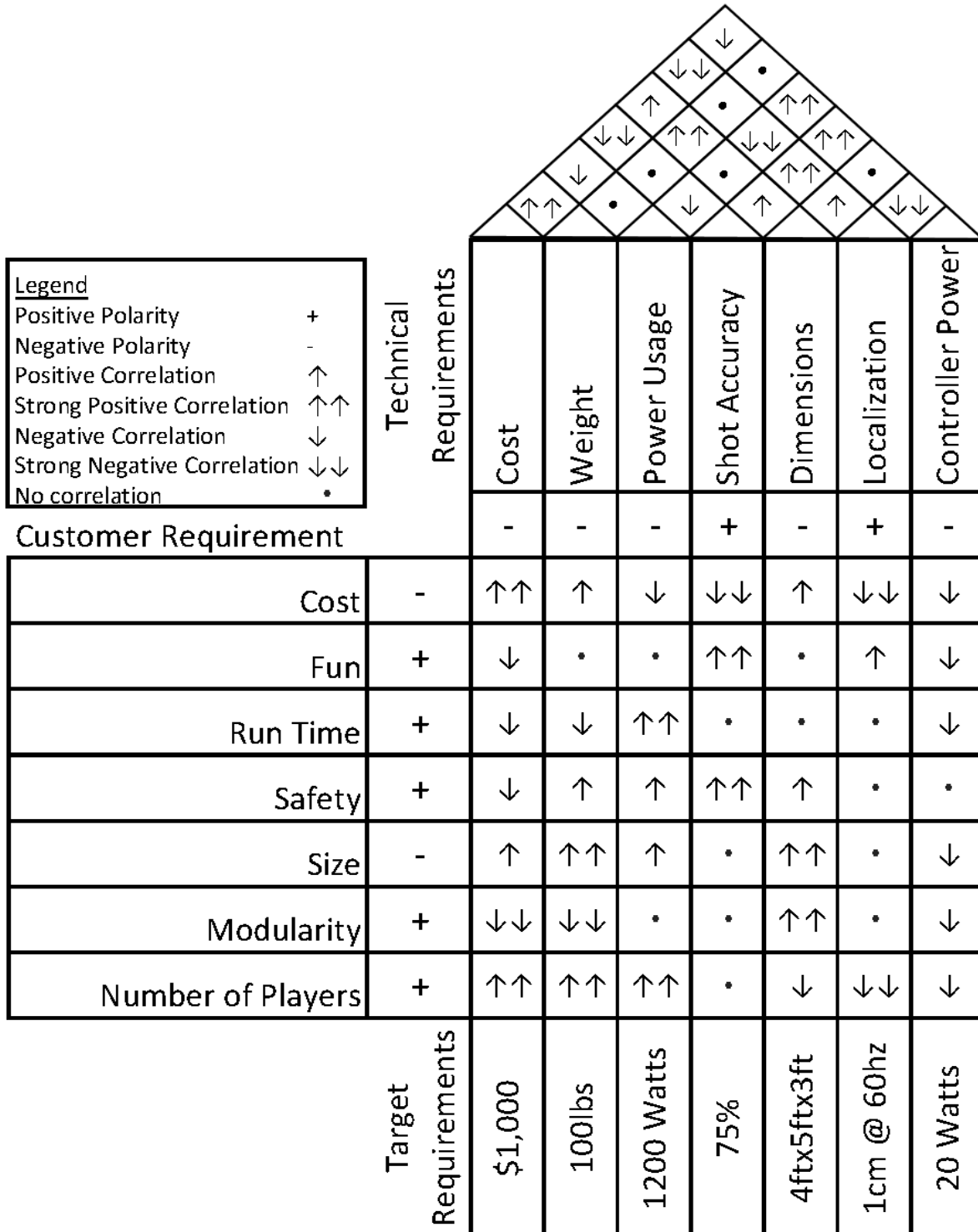


Figure 1 House of Quality indicating customer and technical requirements and their correlations.

System Diagrams

System and Interface Identification

The Project is split into two primary systems: Arena and Robot. The Arena system encompasses all things related to the basketball court, basketball, physical frame structure, computer vision, and user experience systems. 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 user experience 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.

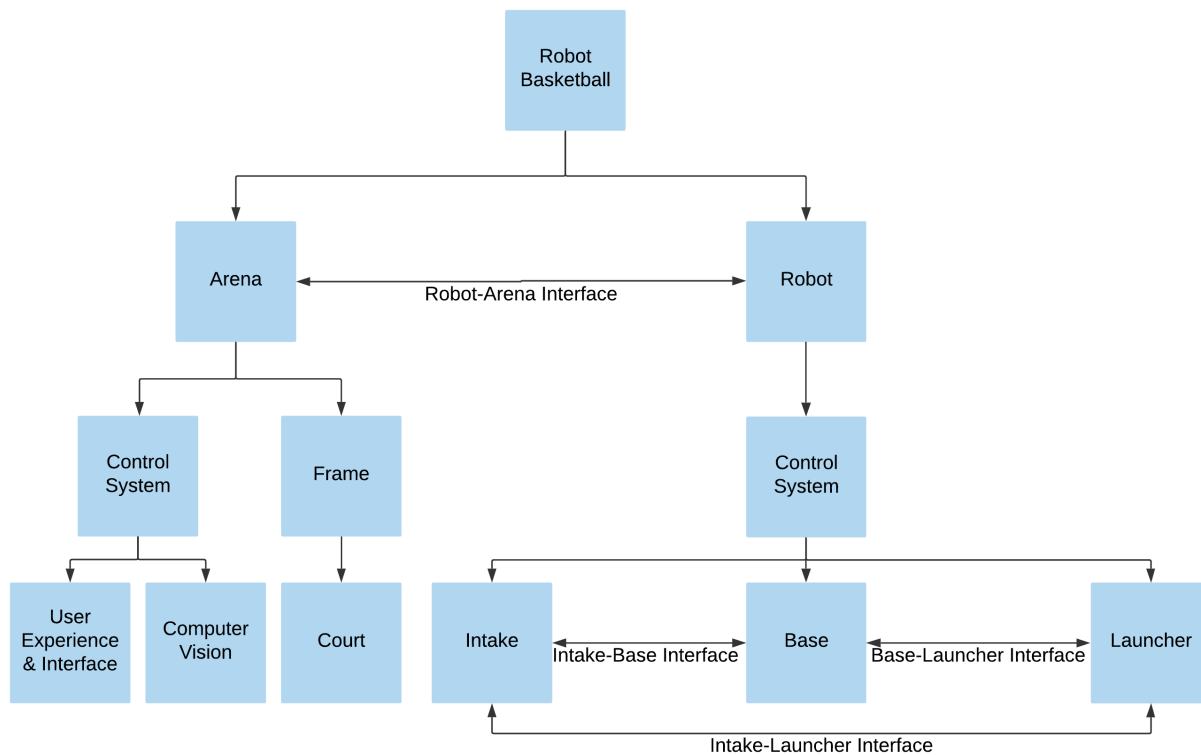


Figure 2 System Hierarchy and Interface Identification

Distributed System

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. 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.

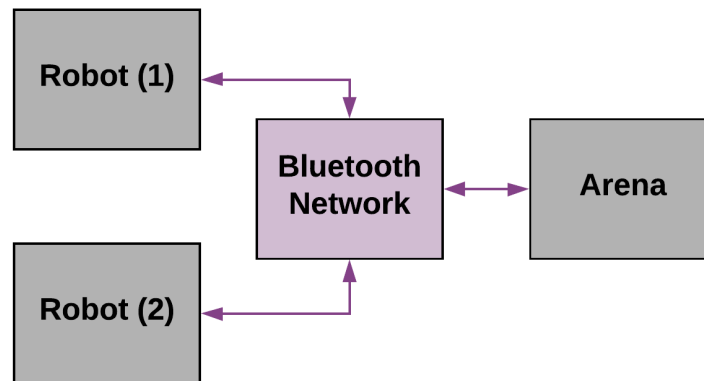


Figure 3 System Communication Diagram

Robot

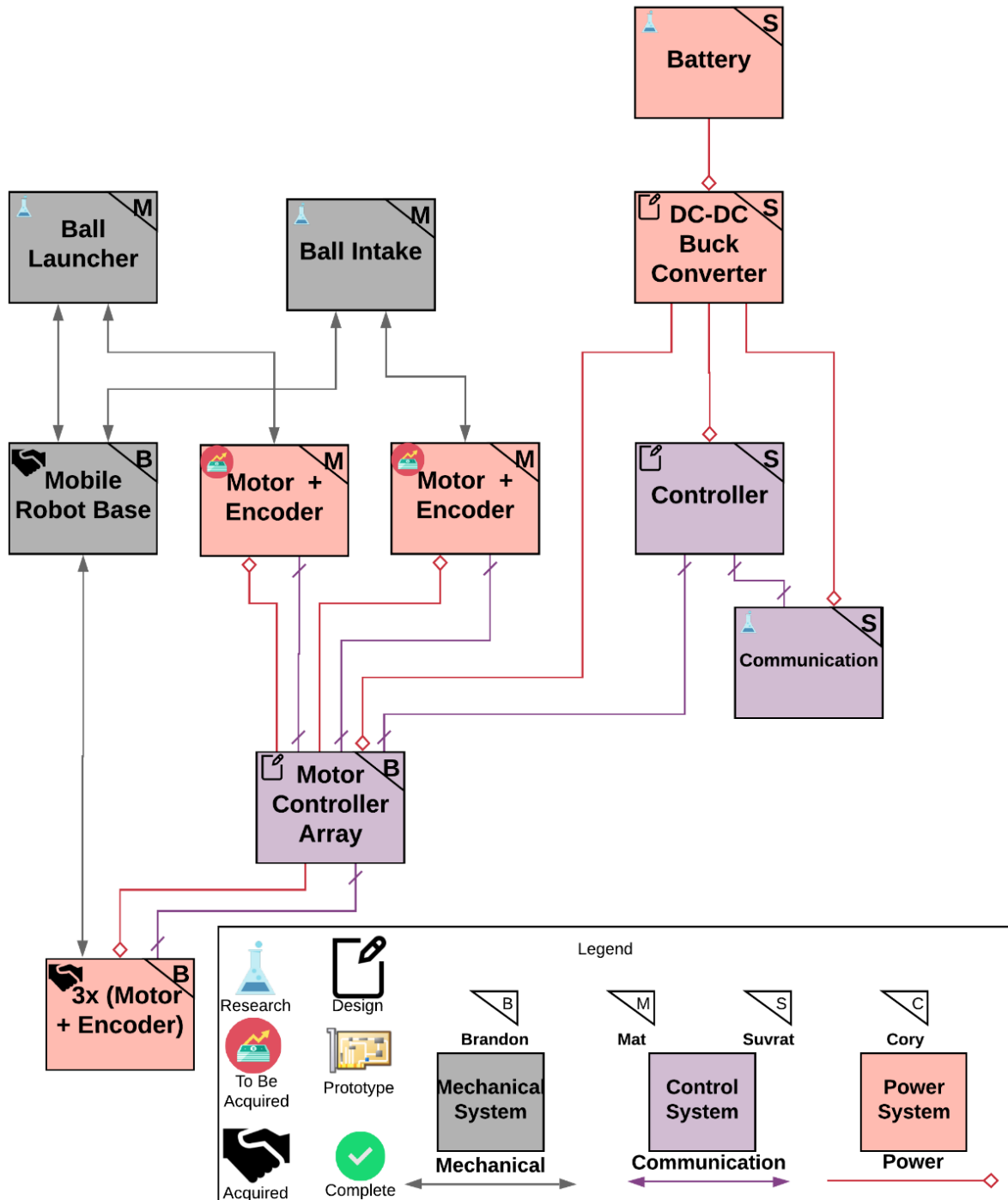


Figure 4 Robot Subsystem Power and Signal Diagram

Mobile Robot base

The mobile Robot base is the locomotion subsystem of this project. 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. Thus, a holonomic-type base is proposed to ensure motion without rotation. This is important such that the player can adequately avoid opponents without completely ruining launcher angle. A common way to achieve holonomic locomotion is utilizing a configuration of Omni-Wheels mounted at different angles.

Omni-wheels are traditional wheels with rollers mounted perpendicular to the actual wheel. This allows the wheel to rotate in two cartesian directions: in the direction of the wheel itself, and perpendicular to the wheel itself. This ultimately achieves a wheel that can move any direction in the ground plane. The wheel mounted at various angles from the base-frame allows for the wheel power to be added by component vectors. Due to the vector component addition, the actual speed of the robot does not directly correlate with the speed of the wheel; instead, it is affected by the trigonometric coefficients of the angles. For example, if the wheels are configured at 45-degree angles from the base frame, then the actual linear motion is a value of 1.41 times the driving RPM. This extra speed comes at the cost of torque, where the driving force is not in the direction of motion.

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. 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, thus they should be securely mounted on some shock-absorbing platform (E.g. Rubber Standoffs)

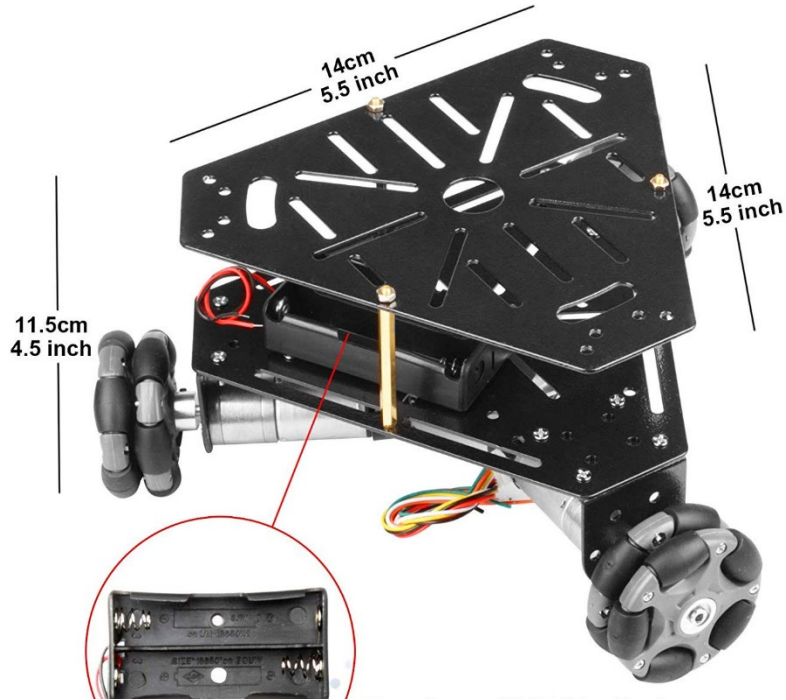


Figure 5 Example Omni-wheel base from Heneng on Amazon.com
[\[https://www.amazon.com/ap/product/B07LGT93BZ/ref=ppx_yo_dt_b_asin_image_o02_s00?ie=UTF8&pvc=1\]](https://www.amazon.com/ap/product/B07LGT93BZ/ref=ppx_yo_dt_b_asin_image_o02_s00?ie=UTF8&pvc=1)

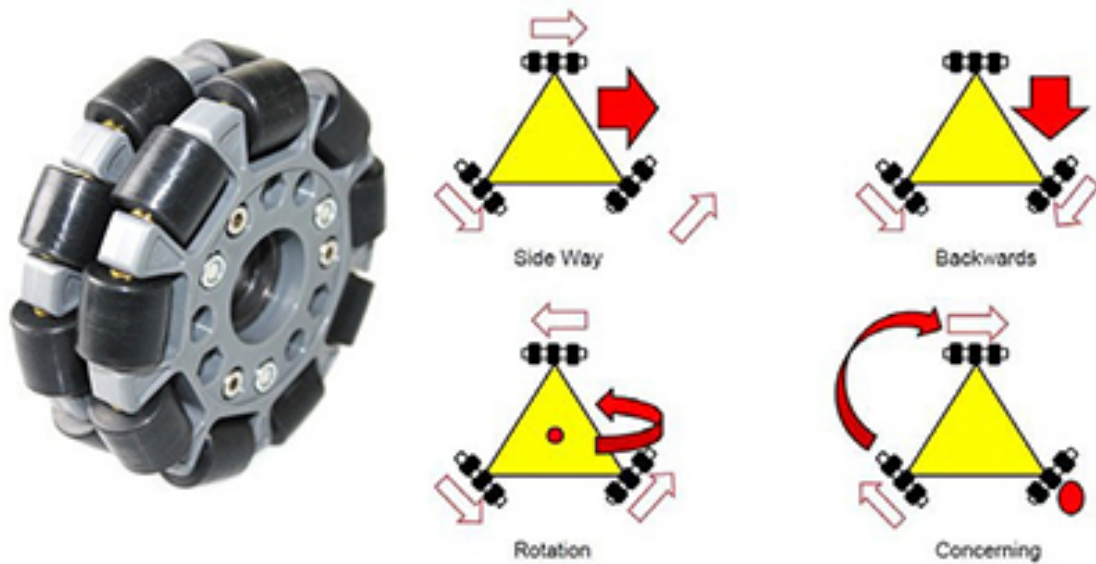


Figure 6 Example holonomic configuration. Image from <https://www.robotshop.com/en/3wd-48mm-omni-wheel-mobile-robot.html>

Battery

The battery is the main source of power for the robot. The battery is supposed to be able to power the motor controllers for omni wheels. This will assist the robot in conducting its locomotive activities. The battery must be strong enough to drive the robot, provide power to the ball launcher and intake mechanism. The longevity of the battery will depend on the battery technology. A normal 9V or a set of AA batteries will not be able to fulfill the power requirements due to low amp-hour ratings. Our current design assumes usage of 6 motors for basic functionality. Therefore, the battery needs to be able to support the voltage and current draw required by all of them. The battery also needs to be rechargeable to reduce cost.

The most reasonable option regarding cost and weight seems to be Lithium Polymer (LiPo) batteries. They are light weight and can provide high current draw at desired voltages. 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 is also required as LiPo batteries come with inherit risk of fire and cannot be over or under charged due to their chemical composition.

Controller

The controller is needed to control the various components on the robot. It receives commands in the form of a packet from the arena and drives the motors at the commanded velocities or positions. This packet contains the information of motor ID, motor velocity or position, and other necessary configuration settings. 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 video playback calculations 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 as discussed in Motor + Encoder.

Motor + Encoder

The largest loads for this system are the motors for the base, intake, and launcher. A few motor technologies are available: DC Brushed, DC Brushless, and Stepper. The pros and cons of these three technologies are listed in a table below. The motors provided by the proposed robot base are 9V DC brushed motors with built in quadrature encoders and gearbox. This combination allows for closed-loop control (Likely PD-control) over the motors' velocities to ensure accurate command execution. These motors are relatively low power at 1.8 W without load.

Speed control of the motors is required to achieve the holonomic motions expected by the player. While precise velocities are not required, introducing feedback for the robot base allows for more accurate locomotion to operate in a variety of conditions including variable battery conditions, mechanical wear and tear, and dead-reckoning in the event of computer-vision

failure. A stepper motor is likely to be used in situations where precise positional accuracy and high holding torque is required. These can be run as open-loop controllers, or additional feedback can be introduced such as potentiometers at the motor output to achieve closed-loop control. The motors will interface directly with the motor controller array located on the PCB.

Table 8 Motor Technology Comparison

	Cost	Power	Torque	Sensing	Motor Control
DC Brushed	Low	Medium	Medium	Requires feedback	Simple H-Bridge
DC Brushless	High	High	High	Requires feedback	Requires ESC
Stepper	Medium	Medium	High	Steps can be counted	Sequence of pulses

Ball Launcher

The launcher must be able to shoot the ball from anywhere on the court (5 ft by 4 ft). The team doesn't want the person controlling the robot to have to worry about controlling the amount of force so the launcher will receive commands from the field. The launching mechanism must be adjustable to make shots from anywhere. This can be accomplished in a multitude of ways, but the designs are narrowed down to two ways: fix the angle and have variable force or fix the force and adjust the angle. Both ways would be done with a spring powered punch (extended via gear rotation) that will propel the ball. With the variable force, there will be a mechanism that will either lift or shift the gear off its track to release the puncher. This raises the problem of which way would be easier to implement and cause less stress on the mechanism. With the fixed force, the mechanism would use a skip gear to get the same amount of rotation on the gear with each shot. The launching mechanism would need to be on a vertical swivel to vary the angle. Changing the angle requires changing the launcher height and would add an extra variable to calculate to get the correct number needed. Another con of the varied angle is the potential to need a much higher ceiling on the field. The team briefly explored the option of a single or double flywheel mechanism as the launcher, however decided that it would be too variable and unreliable to work effectively. Problems that could arise with the use of flywheels are outlined in the table below.

Table 9 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.

Ball 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. Using something like a conveyor belt, or a powered wheel (which is feasible to use for this case because we don't have to worry as much about the input and output speed of the ball) would put more strain on our electrical system, however, would be more likely to work more often. After being picked up, the ball must be transported. Options researched include a telescopic lift, a conveyor belt, or more wheels. This 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.

Motor Controller Array

The motor controller array block is a block that exists on the PCB that contains terminal connections for each of the motors. Each motor contains at least two wires for power. Additionally, some motors also include wires for encoder feedback, or additional power wires in the case of brushless or stepper motors. A typical H-Bridge IC like L298N can power two DC motors or one 4-wire stepper. The array is expected to consist of at least six motor modules that route power in from the DC-DC convertor and out to the motor terminal blocks. Additionally, the PCB in this section should route the signals from the controller to the motor controllers, and from the motor sensors to the controller.

Arena

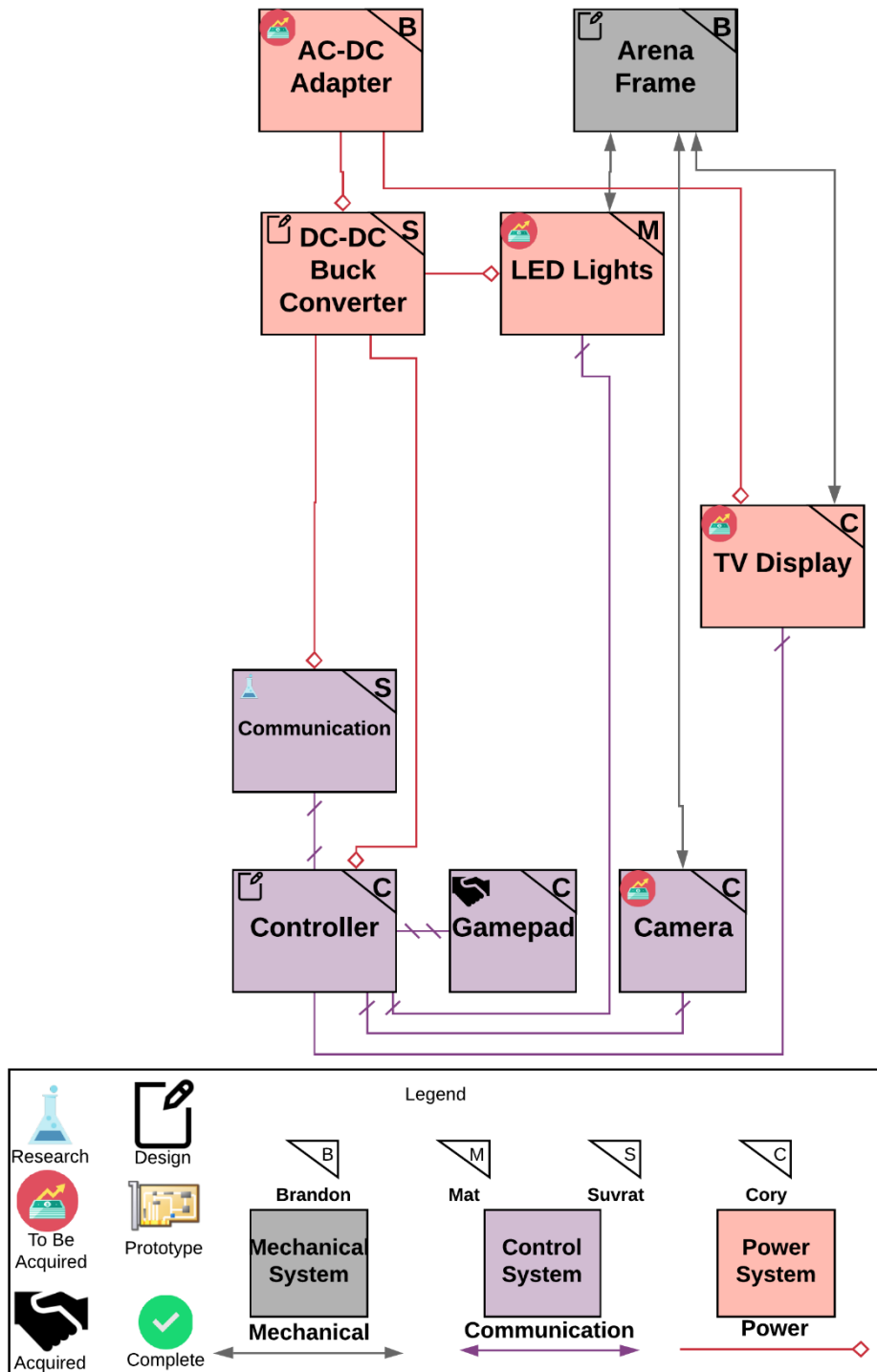


Figure 7 Arena Subsystem Power and Signal Diagram

AC-DC Adapter

This component is required to convert power from a standard US 120V 60Hz outlet to DC power. This component must be highly efficient; thus, it will be purchased. Some example

devices include the Apple USB Power adapter that converts to 5V DC at 2 Amps. A single outlet is expected to support the entire Arena subsystem including TV display, Controller, and other loads.

LED Lights

The arena will have some form of LEDs on them to show the status of the field to the players or audience. The LEDs will also need to be independently addressable so that we can control which ones are on and what color they are. This will allow us to come up with different codes for different statuses and not have to have all the LEDs on all the time as this would waste power and could be distracting. Another function of the LEDs would be to serve as an attention grabber and provide entertainment value. We are currently looking at NeoPixels to fulfill this role as we already have some experience with them and feel they are in a good price range.

Display and Sound

The arena contains a visual display unit, like a TV, monitor, or tablet, that is used to relay information to the players. The most important thing to display is the settings page, like a dashboard screen on a video game console. This is used to set up new player's robot, adjust sound, game mode, playback options, and other important settings related to the robot.

The display will also show important information related to the currently active game, such as the score of the game, the current period out of four total periods, and the remaining time for the current period.

Additionally, when a basket is made the screen will display a replay of the events leading up to the shot and the shot itself. We plan to accomplish this by recording the last 30 seconds of gameplay before a shot is made, and then visualizing the robot and arena in a game engine. The Unity game engine will most likely be incorporated here because it has copious amounts of documentation and support. In addition, the team believes the stylized graphics that Unity supplies would fit the aesthetic feel planned for the game.

The display will also be used for debugging and development purposes. This means that it will have to be capable of showing the real time view of the computer vision camera as well as any software used for programming the project, in addition to any other programs needed on our controller for debugging and development purposes.

Arena Frame

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. The arena must contain a flat surface that has some amount of grip to ensure consistency of the robot base locomotion. Two hoops on either side of the long end of the arena are mounted one foot above the ground plane. The frame should be collapsible so that it can be easily transported and assembled when necessary. The arena must be exceptionally level to ensure proper basketball physics, thus leveling leadscrews are to be included underneath the frame to level appropriately. The ball is to be roughly 1.6" diameter and rubber to ensure appropriate bounce and small enough scale. 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. A low-cost example is PEVA which can be found in shower curtains, car covers, and other typical household items. The frame must support the weight of the mounted TV, electronics box (Containing the controller and PCB), and the over-head camera frame. Additionally, the court must contain electronics for sensing if a goal is achieved such as an infrared sensor mounted within the basketball hoop rim.

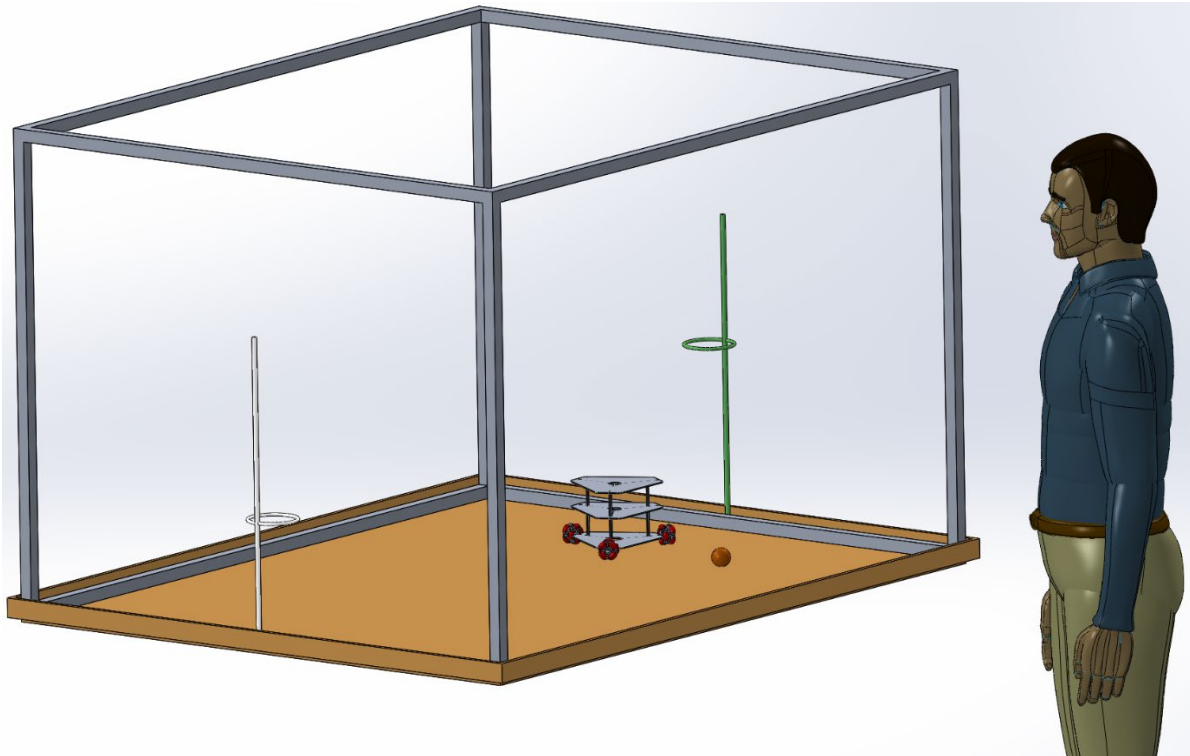


Figure 8 SOLIDWORKS Image of proposed scale Arena, Robot, and Ball

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. A Raspberry Pi 3 B+ was originally suggested because of its built-in Bluetooth capability 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. For this reason, a Jetson Nano may be used in this project. 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.

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.

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 add a layer of reliability. 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. Between the two gamepads we believe the Xbox One gamepad will have the most documentation and support as well as ease of programming and will be using it for our project.

Table 10 Player input functions and gamepad mapping

Function	Type	# Of Axes
Forward/Backward + Strafing	Joystick	2
Rotation	Joystick	1
Shoot	Trigger (Analog or Digital)	1
Intake	Trigger (Analog or Digital)	1
Select	Button	1
Back	Button	1

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.

The camera choices will be between a Pixy2 camera that comes with computer vision tracking already implemented or a USB webcam that we will have to write the computer vision tracking for. The Pixy2 camera would have to be placed at least six feet above the arena to cover the court entirely. A webcam would be able to be placed lower to the arena. The Pixy2 will display video in 1296x1976 resolution at 60 FPS. This might not be a fast and clear enough video for a very accurate location of the objects being tracked in the arena. After further testing the Pixy2 camera we will determine if it can supply the accurate computer vision that we require for the project. If we find that it does not meet the requirement, we will have to move to a USB webcam.

Common

Communication

There are multiple radio technologies for conducting Arena-Robot communication. 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. Another technology which 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. The arena will use camera information, update the TV screen, encode the movement instructions based on Gamepad input and Robot's position, and generate a packet for transmission. This communication protocol will also allow the robot to send sensor data back to the arena, which will display the robot stats on the screen and help the user debug and monitor its movements.

DC-DC Buck Converter

The power supplied to the robot with a LiPo battery is more than what a microcontroller can safely handle. Most microcontroller technologies function on 3.3V or 5V logic level. Due to this a DC-DC buck converter will have to be designed. TI's Webench tool will be utilized to create the design. 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.

The arena will mainly be powered by an AC power outlet. An AC to DC converter is used to convert the outlet voltage to 5V at 2A as mentioned before. This converter can either be designed using Webench or bought off the shelf depending on time, resource and complexity of the project. The AC-DC converter will be used to power the microcontroller of the arena which in turn will power all its peripheral devices.

Budget:

Table 11 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	\$ 15.00
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 12 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

Timeline:

Our group has a room in HEC reserved every Tuesday and Thursday from 12pm-1:30pm for design meetings. These operate as SCRUM stand-up meetings in our AGILE based development process. Due to the modularity and subsystem breakdown of the project, most components and software development can be completed in parallel to other tasks. The block diagrams in figures 3 and 4 indicate proposed task delegation mostly broken down by subsystem. Some tasks across subsystems are very similar, and thus are combined under a single owner in order to reduce duplicate work (Bluetooth in Robot and Arena, for example).

A proposed Gantt schedule can be found on the following page. The generalized process for task distinction is Concept -> Design -> Prototype -> Test. However, engineering is a highly iterative process thus many of these distinct components overlap or repeat over the course of the project. Many of the timeslots for the tasks are longer than the actual expected task length in order to account for this iteration and allow for some time slip as necessary.

