Senior Design Project Proposal

IEEE SoutheastCon Hardware Competition

Auburn University

January 21, 2019

Matthew Castleberry, Joe Hinely, Josh Jabłonowski, Nia Perkins

Project Overview

The purpose of this project is to create a robot to compete in the 2019 IEEE SoutheastCon held in Huntsville, AL. The purpose of the conference is to share and demonstrate technical advancements in engineering. We will be competing in the hardware competition against other universities from across the southeastern US.

The purpose of the robot is to move around a square carpeted area with a side length of 244 centimeters, and pick up small objects on the field and place them into one of four colored home bases based on the color of the object. The arena will be split into two zones. Zone one features four home bases corresponding to four different colors. The robot will start in one of the four home bases. Zone two is separated by a one inch thick white line. Zone two is where twelve objects will be placed for the robot to pick up: eight two inch wooden cubes, and four 2.5 inch diameter pit balls. The arena will feature a wooden box in the middle that will serve as an obstacle as well as four LED lights situated at the edge of zone two on the one inch line. The four home bases will be placed at the four corners of the arena in zone one. An arena was built in order to test the robot as shown in Figure 1. This allows testing of the robot in a controlled area leading up to the competition.

The robot has several constraints that limit the size of the robot, materials used in the construction that assure the safety of participants and other robots [1]. The robot's width and length must be within a nine by nine inch area, and a max height of eleven inches. The nine by nine area must include a bumper covering eighty percent of the perimeter. The bumper must be 1.5 inches from the ground and have a minimum vertical cover of one inch, and must not have a radius of curvature less than one centimeter. This is to prevent damage to the arena and other robots in the event of a collision. The robot can also raise a mechanical arm, with a max length of three inches, with a flag presenting the competing school's name.

The robot utilizes a Raspberry Pi chip to control the motors of the robot, while using an attached camera to identify the color and shape of the objects on the field. A LIDAR will be used to localize the robot on the arena to help avoid the four lights and the wooden box in the middle of the arena. The robot will move the objects to the appropriate home base by running over the objects, closing a small door on the front of the robot so the robot can go backwards without losing any objects, hold them under the robot and move towards the appropriate home base of the object.

Figure 1: Field Constructed in Broun Hall room 368

Management Approach

This project was initiated by Auburn University's Student Projects and Research Committee (SPARC) at the end of Spring 2018 semester. The team is split into two sub-teams: mechanical and electrical. The mechanical team is responsible for the design and constructing the robot's mechanical system focusing on the frame, mobility system, debris collection system, and flag raiser. The electrical team is responsible for selecting and wiring the electrical hardware components as well as writing the software to make them operate as desired. In order to increase the amount of effort put into the project, this senior design team has been formed to be a part of the electrical sub-team.

One of SPARC's primary functions is to display the status of its projects at E-Day during late February each year. Because this project is considered a major SPARC project, it is desired that it be in a minimally operational state by February 22, 2019. The goal is the robot would be able to autonomously push debris into the corresponding corner. After this deadline, the remaining work will be focused on tuning and diversifying the algorithm as well as accounting for an opponent robot.

Based on the previous development and this semester's two-phase development plan, this project most closely corresponds with the agile methodology. The tasks will be grouped into sprints in which each "has a defined duration (usually in weeks) with a running list of deliverables, planned at the start of the sprint" [2]. The project has multiple components that must be developed separately and then later integrated thus the first half of a sprint will be component focused and the second half will be focused on integration.

Technical Approach

Mechanical Strategy

The robot had three initial designs, a conveyor belt, a forklift to pick up the object, or pushing the object around. Each design worked on the basis of the robot running over the objects, however the conveyor belt and forklift used a tank to hold the objects in after lifting them. The conveyor belt design involved moving over the object and lifting them into a tank on the robot using notches on the belt to catch the objects. The forklift featured foam fingers located under the robot that picked up objects, lifted them through the robot and placed them into the tank. The final design removed the tank on the back of the robot, instead holding the objects under the robot while moving.

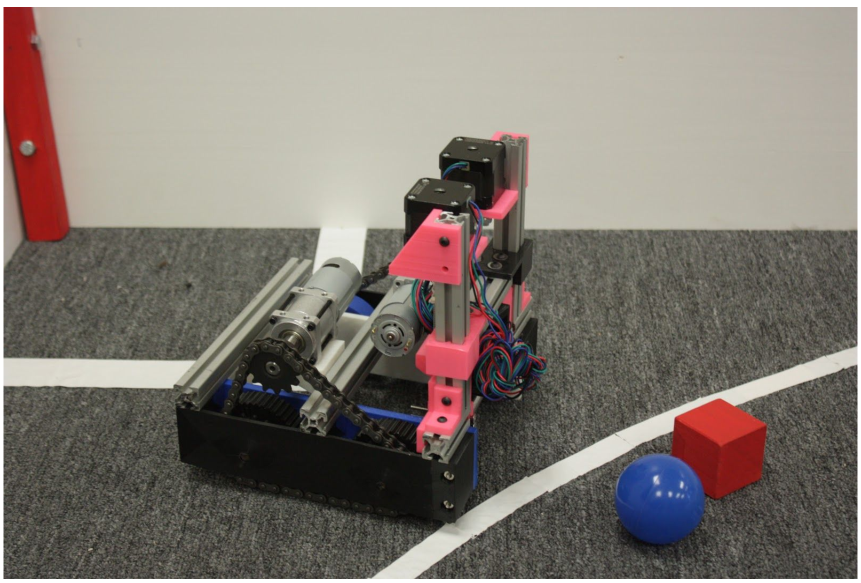

Figure 2: Current Status of Mechanical Platform

In order to keep the objects under the robot while moving backwards, the robot will have a 3D printed gate that is moved up and down using two small motors. The 3D printed sections can be identified by the pink and blue sections of the test robot, with the black motors on the front of the robot controlling the gate shown in Figure 2. The gate is raised to gather objects and closed to hold the objects under the robot until it reaches the appropriate home base. The front of the robot will also use a pole covered in flex seal that turns in order to pull objects under the robot.

For easy modification and building, the wheels and connections of the robot are 3D printed, the wheels forever have little to no traction, this is fixed by covering the wheels in flex seal in order to create traction against the carpet arena. The robot has two main motors. Due to availability, the robot uses two 70 Amp motors to drive the wheels and two motor controllers for the smaller motors controlling the front gate.

Electrical

Because this is an autonomous competition, our robot must be able to identify colors, locate them on the field, figure out the distance between the debris, other debris and how far it is away from its drop location while also recognizing where its current position is in the field to avoid spacetels and the centerpiece. Our approach to being able to identify space debris is with a camera mounted on the robot. OpenCV is used to program a debris recognition algorithm which is explained more in the software section. The camera selection was between a single camera or a stereo camera. The stereo camera is a two camera setup that gives vision more similar to human eyesight. This would be nice however this would require more processing power than our raspberry pi can handle. The single camera focuses on color instead of depth which is what we need, so the single was the better choice.

For the robot to be able to tell where its location is with respect to the playing field we need a device. The electrical team tackled with two approaches LIDAR vs. four distance sensors. The four sensors seem like a better approach, but actually, it requires a more complex algorithm since you have four different points to process. With the four sensors you have four data points to determine the robotics location on the field, which depending on the robotics location it may not be able to give an accurate location with the four sensors. The LIDAR however gives you a location with multiple data points which allows for a more precise location.The LIDAR does give more data but this allows for a more precise location.

Lastly, we had to decide rather use one battery or two. From you the pugh chart in Table 1, the two batteries is a better choice. This allows us to use a larger battery for the motors and smaller battery for the Pi. This prevents from brownout during a round.

Table 1: Pugh chart for number of batteries

Criteria	Weight	One	Two
Safety	2	--	+++
Reliability	2	--	++
Cost	1	+	-
Robot space	1	++	-
+		3	5
0		0	0

-		4	3
Total		-1	3

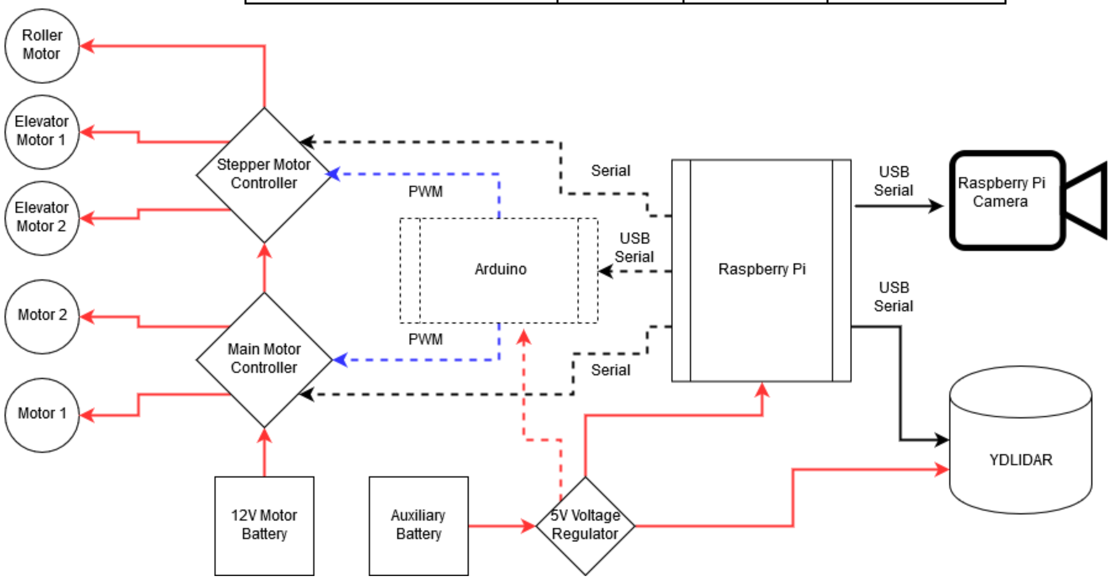

Figure 4: Electrical Block Diagram

Figure 4 shows a block diagram of the proposed electrical system. It is comparing two different configurations of controlling the motors. Currently on the test platform shown in Figure 5, the motor controllers require pulse width modulation (PWM) signal. Because the Raspberry Pi does not support multi-channel PWM, an Arduino microcontroller is used to supply the PWM. If new motor controllers with serial support are purchased, the Arduino can be eliminated therefore simplifying the design.

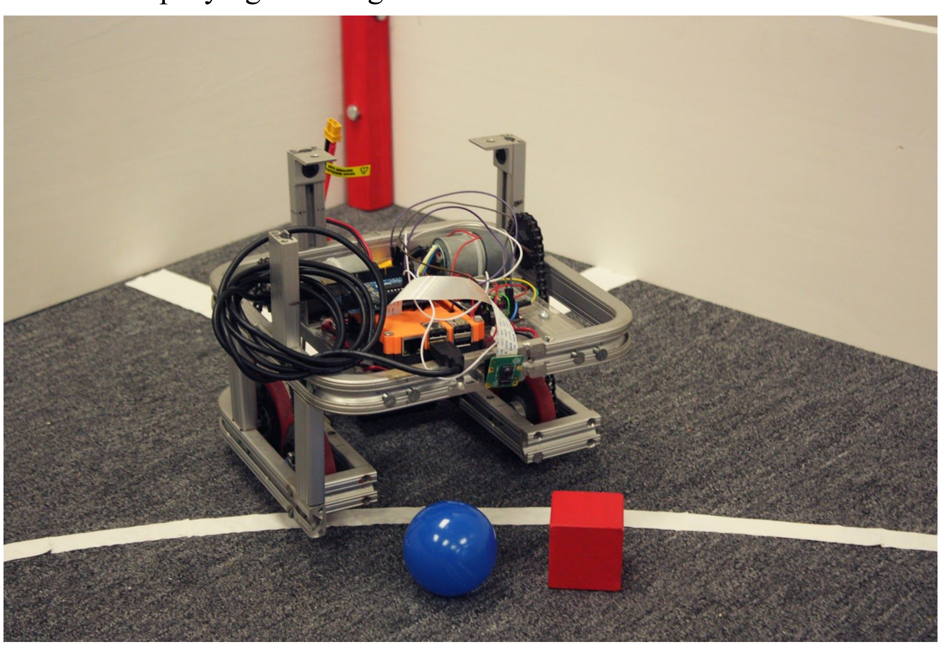

Figure 5: Electrical Test Platform

Software

Programming Language Selection

One of the first major decisions the team had to make for software development was which programming language we were going to use. We compared the two most popular programming languages for robotic applications, Python and C++. Both had support for the selected visual processing library, OpenCV. After considering the ease of development and online support of python versus the stability and processing speed of C++, we determined that C++ was the better option. See Table 2 for a pugh chart comparing the two languages.

Table 2: Pugh Chart for Programming Languages

Criteria	Weight	Python	C++
Ease of development	1	++	-
Stability	2	-	++
Online Support	1	+++	+
Processing Speed	2	-	++
Available Libraries	1	++	+
+		7	10
0		0	0
-		4	1
Total		3	9

Debris Detection Via Visual Processing

In order to detect the debris objects, a visual camera is placed on the front of the robot. The powerful open source C++ computer vision library, OpenCV, is utilized to interpret the images the camera receives. The program analyzes the colors in the picture by creating four pictures of a black background with white indicating the pixels of the corresponding color (Figure 6). OpenCV's contour detection function is then used to find all of the edges. A bounding box and circle are placed around each of the contours. By comparing the size of the bounding box and circles, the shape and approximate pixel size of the objects can be determined.

Figure 7 shows the resulting image by running the program on a static image. The corresponding data returned is listed in Figure 8.

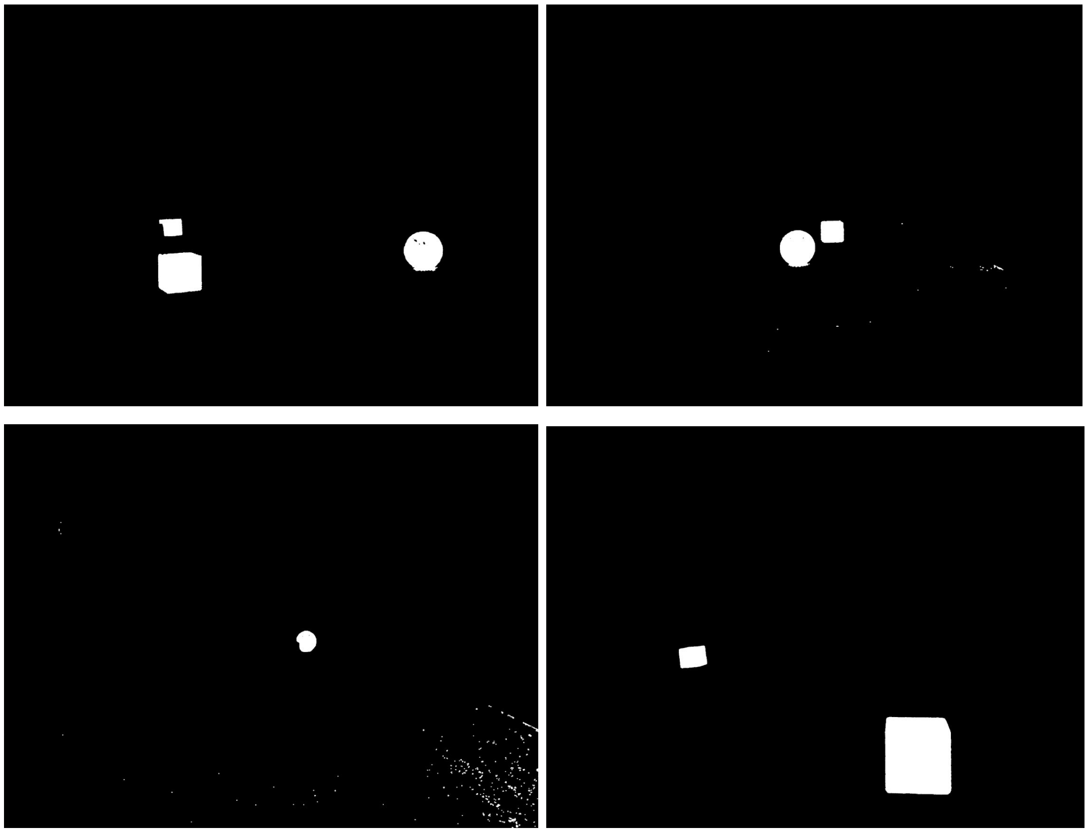

Figure 6: Color Filtered Analysis of Debris, red (top left), yellow (top right), green (bottom left), blue (bottom right)

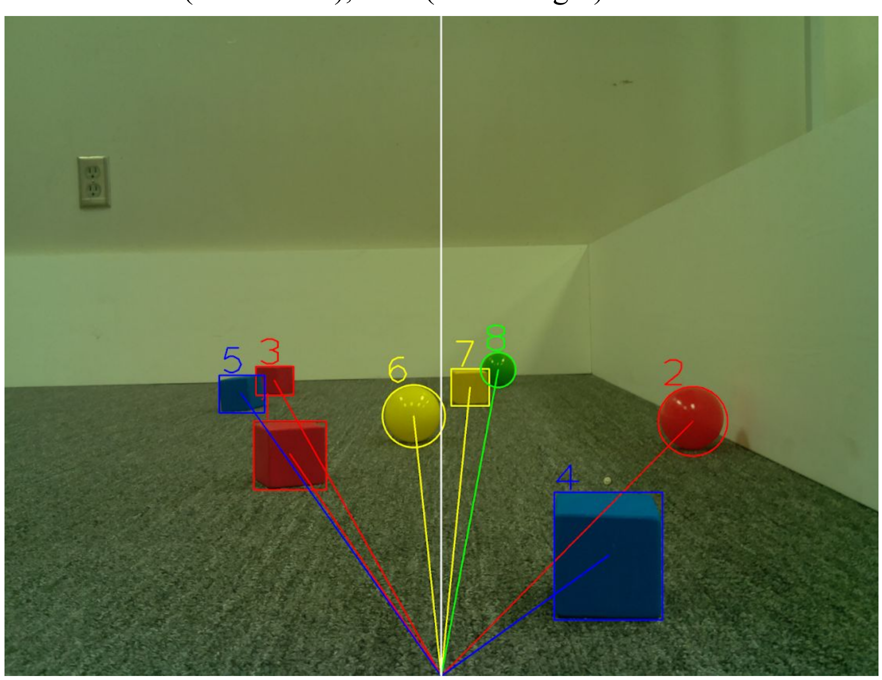

Figure 7: Demonstration of Debris Detection Program on Static Image

```
1. Red blck     @ [855, 1296]   -34.3 degrees   w/ width 211    and height 200
2. Red ball     @ [2029, 1201]  44.6 degrees    w/ radius 102
3. Red blck     @ [810.5, 1082] -29.4 degrees   w/ width 110    and height 84
4. Blue blck    @ [1784, 1592]  54.2 degrees    w/ width 316    and height 372
5. Blue blck    @ [713, 1120]   -35.3 degrees   w/ width 133    and height 109
6. Yellow ball  @ [1216, 1186]  -6 degrees      w/ radius 91
7. Yellow blck  @ [1380, 1102]  5.66 degrees    w/ width 111    and height 108
8. Green ball   @ [1462, 1051]  10.5 degrees    w/ radius 51.1
```

Figure 8: Output from Debris Detection Program

The remaining work to be accomplished is to determine the distance the objects are from the camera. There is an example python program accomplishing this by calibrating the camera to a known object [3]. Since the blocks and balls are known, definite sizes, this method should be fairly reliable. The entire visual detection program also has to be integrated other software in order to provide all of the functionality required for an operational robot.

LIDAR Analysis

Figure 9: Image of YDLIDAR

In order for the robot to know where it is on the playing field, we are using a LIDAR. A LIDAR sensor uses a spinning optical sensor to measure the distance to nearby objects very similar to how a RADAR sensor works. During the fall semester, a low-end 2D LIDAR sensor, the YDLIDAR, was purchased as shown in Figure 9. From work done last semester, a program was developed to record all of the points measured in a single rotation of the sensor. The results are displayed in Figure 10.

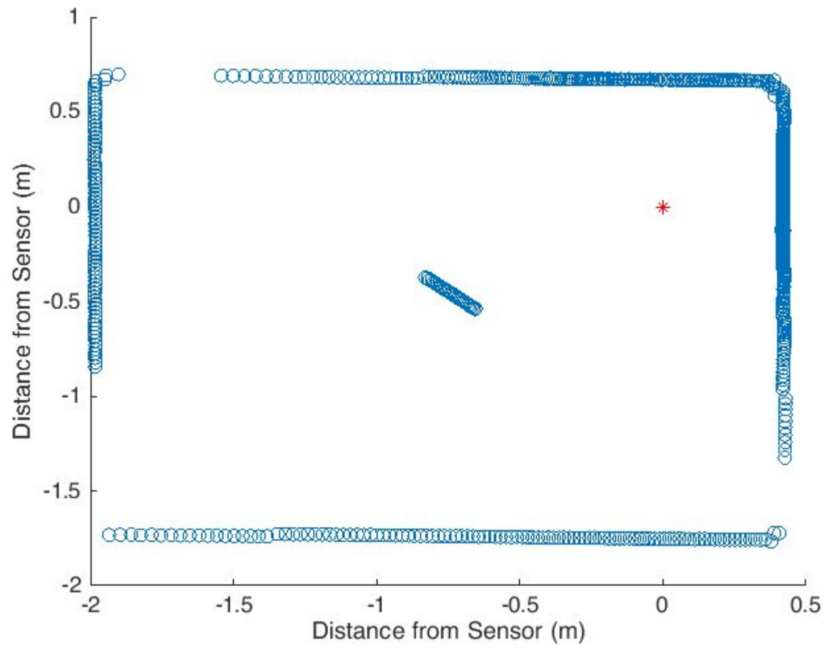

Figure 10: Plot of Detected Points from LIDAR

Another program was developed to approximate the x,y coordinate of the location of the sensor with respect to the walls. It uses a technique called rotating calipers to fit a square around the field as shown in Figure 11. From this information, we can program where the robot can and cannot go based on the current location. Further testing needs to be done to prove the program works in all cases especially as the robot approaches the center structure. If the program works accurately, then it will need to be integrated into rest of the functionality of the robot.

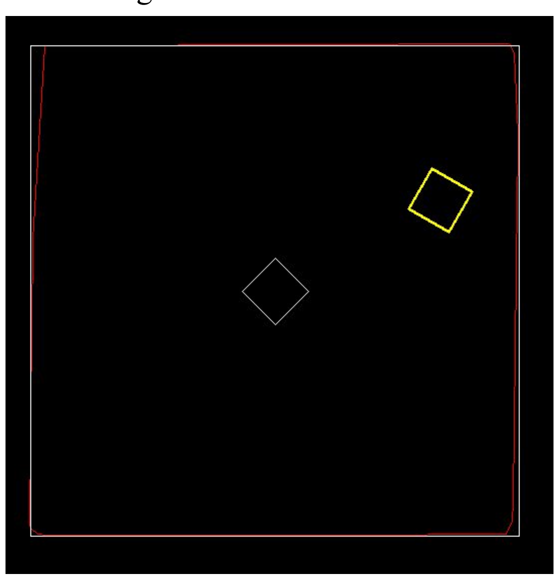

Figure 11: Analysis of LIDAR Data

Utilization of Robot Operating System (ROS)

A popular robotics C++ library, Robot Operating System (ROS), exists to ease the integration of various sensors and devices on robots. Much discussion has taken place on whether or not this library would be helpful. The benefits are it may have many algorithms and tools built in to help make sense of the data received from the sensors. The downside is that it is most commonly used on systems with communication back to an external computer and the learning curve is fairly steep. The alternative is a custom integration of the individual software components. At this point in time, it does not look beneficial to utilize ROS but it will be reconsidered at a later point if needed.

Integration

Since there are many significant parts to the robot with many design decisions, it is important to consider what challenges may take place in integrating major parts.

The mechanical and electrical teams must integrate their parts to ensure electronics have space to sit on the finished robot. Motor controllers must be selected to match the requirements of the motors selected by the mechanical team. The battery that is selected must also be able to support the motors and other mechanical systems on the robot.

Electrical and software integrations must also take place for the systems to communicate properly. After a decision has been made on what type of communication the motor controllers will require- serial or PWM- appropriate wires must be terminated at the Arduino and Raspberry Pi. These connections must also be made to the motor controllers.

Since the software is divided into two major components, object detection and LIDAR localization, there will need to be software written to fuse the information appropriately. This will happen a few weeks before each development cycle. In addition to sensor development, an efficient algorithm will need to be developed based on the inputs received by the software.

Budget

The chart in Table 3 includes an expected Bill of Material (BOM) of the parts that will be needed for this project, separated by the portion of the robot they are for. i.e. mechanical, electrical, software. All funding for parts will come through SPARC, approved by Dr. Roppel. Some parts have already been purchased, and are marked accordingly. Note that the components used for the arena are not included in this BOM.

Table 3: Predicted Budget

MECHANICAL				
Part	Quantity	Price/per	Total	Ordered?

2020 Extrusion	96in	$0.16	$17.32	Y
2020 nuts m4	50	$0.21	$10.50	Y
100x M4 Button Screws	1	$9.90	$9.90	Y
Flanged Bearings	8	$3.97	$31.76	Y
Shaft Collars	8	$3.89	$31.12	Y
12" Shaft	1	$7.72	$7.72	Y
6-32 Thread Size, 1-1/4" Long, wheel screws	1	$6.80	$6.80	Y
6-32 Thread Size,wheel/frame nuts	1	$3.40	$3.40	Y
6-32 Thread Size, 2" Long, frame screws	1	$6.90	$6.90	Y
Intake Rod	1 ft	$5.45	$5.45	Y
NEMA Screws	1	$4.18	$4.18	Y
Hex Nuts	2	$2.14	$4.28	Y
Lead Screws	1	$5.14	$5.14	Y
Lead Screw Coupling Tubing	10ft	$7.00	$7.00	Y
50x 10-32 Screws	1	$6.06	$6.06	Y
motor sprockets	2	$13.85	$27.70	Y
Motors	2	$7.25	$14.50	Y
Gearboxes	2	$56.50	$113.00	Y
Gearbox Mount Accessories	2	$4.50	$9.00	Y
Chain	1	$16.88	$16.88	Y
Chain Tools	1	$21.80	$21.80	Y
NEMA 17 Vertical Bar Actuator	3	$12.99	$38.97	Y
MECHANICAL TOTAL BUDGET			$399.38	
ELECTRICAL				
Raspberry Pi 3B+	1	$30.00	$30.00	Y
Arduino Uno	1	$20.00	$20.00	Y
3 Stepper Motor Driver	1	$30.00	$30.00	N
Dual DC Motor Driver (70A inrush)	1	$30.00	$30.00	N
Auxillary LiPo Battery	1	$30.00	$30.00	N
Main 12V LiPo Battery	1	$30.00	$30.00	N
5V Voltage Regulator	1	$10.00	$10.00	N
Raspberry Pi Camera V2.1	1	$20.00	$20.00	Y
Raspberry Pi Camera Ribbon Cable	1	$10.00	$10.00	N
USB-B to USB-A Cable	1	$5.00	$5.00	Y

YDLIDAR Module	1	$100.00	$100.00	Y
Various Connectors and Wires	1	$0.00	$0.00	Y
ELECTRICAL TOTAL BUDGET			$315.00	
FULL ESTIMATED PROJECT BUDGET			**$714.38**	

Major Roles

Due to the challenge of managing a large team, this project will require major roles to be divided evenly amongst the team members. The distribution is shown below in the organization chart in Figure 12. Note that the members of the senior design team are in yellow, and are all a part of the electrical team. The primary focuses of the senior design team will be electrical hardware, LIDAR localization, computer vision algorithms, navigation algorithms, and overall team management.

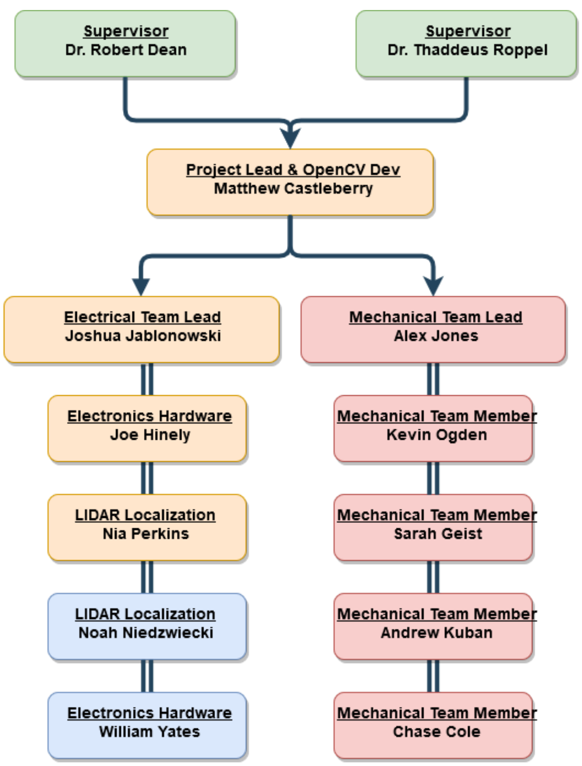

Figure 12: Organizational Chart

Timeline

Table 4 lists the weekly tasks for each aspect of the team. Figure 13 summarizes the table in the form of a Gantt chart. The project is split into three different cycles. The first cycle is to create a minimally operational robot that can move debris into the corresponding corner. This milestone is set to be completed for a demonstration at E-day on February 22nd. The second cycle is to finish the design, test alternative algorithms, and construct a second robot. A second robot will be useful in testing so that we can test the program for the final rounds. It also is a good backup to have two full robots for the competition in case of part failure. The third phase will take place after the competition and will be focused on preparing for the final presentation.

Table 4: Weekly Tasks to be Accomplished

Week	General	Mechanical	Electrical	Senior Design
Jan 7-11	Team Leads meet to discuss current progress, immediate actions, and competition	Continue Assembling prototype #1	Test serial control and vision detection program developed last semester	Form Team
Jan 14-18	First general SPARC meeting of the semester	Cont.	Tune full program, develop LIDAR program	Develop project proposal
Jan 21-25		Cont.	Cont.	Keep up with required documentation
Jan 28-1		Prototype #1 mobility system completed, start integrating electronics	Start integration on hardware	
Feb 4-8		Improve prototype #1	Further tuning and integration	
Feb 11-15		Cont.	Cont.	
Feb 18-22	E-Day Demonstration: Prototype #1 can put debris in the correct corner			
Feb 25-1	Travel Team Applications are released	Design prototype #2	Develop alternative programs,	
Mar 4-8		Start assembling prototype #2	Cont.	
(Spring Break) Mar 11-15	Deadline for travel team applications (Mar 11), Register for conference and hotel			
Mar 18-22		Test both robots on field	Test both robots on field	
Mar 25-29		Cont.	Cont.	
April 1-5		Cont. and 3D print extra parts	Cont.	
April 8-12	Week of Competition			

April 15-19		Cleanup and organize labs	Cleanup and organize labs	Prepare poster, practice demonstration
April 22-26				Senior Design Fair

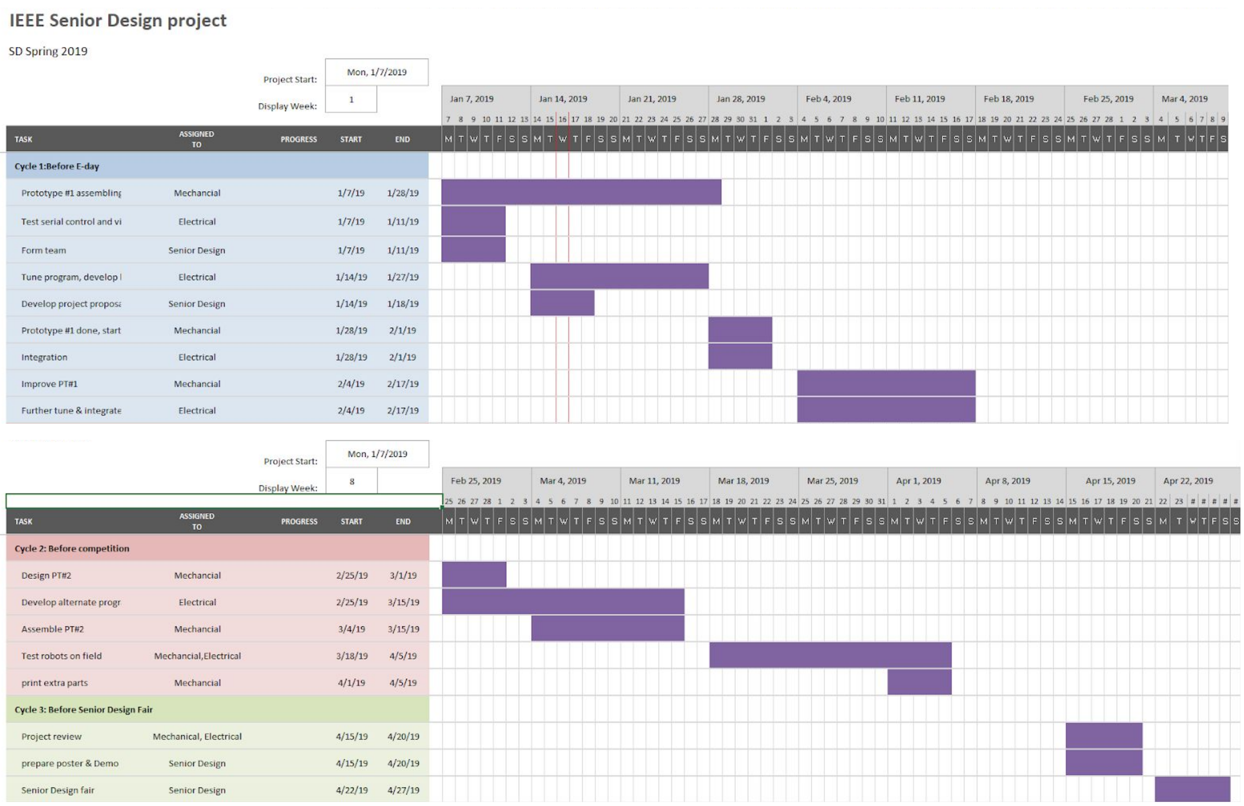

Figure 13: Gantt Chart of Project Timeline

Facilities to be Used

This project will utilize two labs located in Broun Hall 367 and 368. Both labs are supervised by Dr. Roppel. Broun 367, the SPARC lab, will be used for design and prototyping. Broun 368 will be used as a test area, housing the 9`×9’ test arena. All hardware will be stored in these two labs on campus for the duration of the project.

Conclusion

Overall, we are constructing a robot to autonomously move small blocks and balls into coordinated colored corners. The technical deadline for this project is April 10th with a major operational milestone of February 22nd. The operating technical budget is $714.38.

References

[1] “2019 SoutheastCon Hardware Rules v1.2.” IEEE, Huntsville, 18-Aug-2018. “http://sites.ieee.org/southeastcon2019/files/2018/09/2019-SoutheastCon-HW-Rules_v1_2.pdf”

[2] “Waterfall vs. Agile: Which Methodology is Right for Your Project?,” *Segue Technologies*, 20-Nov-2018. [Online]. Available: https://www.seguetech.com/waterfall-vs-agile-methodology/. [Accessed: 16-Jan-2019].

[3] “Find distance from camera to object using Python and OpenCV,” *PyImageSearch*, 05-Dec-2018. [Online]. Available: https://www.pyimagesearch.com/2015/01/19/find-distance-camera-objectmarker-using-python-opencv/. [Accessed: 16-Jan-2019].

[4] Jarema, R., & Jarema, R. (2018, April 12). Batteries - choose the right power source for your robot. [Online]. Available: https://medium.com/husarion-blog/batteries-choose-the-right-power-source-for-your-robot-5417a3ec19ca. [Accessed: 16-Jan-2019]

Appendix A: Disposition Agreement

Auburn University's IEEE SoutheastCon 2019 Hardware Competition Team is funded solely by donations generously given to the Student Projects and Research Committee and the Department of Electrical and Computer Engineering. Henceforth, all hardware is the property of the department and SPARC. At the conclusion of the projects, the hardware will be used for demonstrations or future SPARC projects.

Signature: [illegible] Date: 1/22/2019

Signature: [illegible] Date: 1/22/2019

Signature: Joshua Jablonowski Date: 1/22/2019

Signature: [illegible] Date: 1/22/2019