IEEE SoutheastCon Hardware Challenge Project Proposal

Brian Roskuszka, Cameron McSweeney, Daniel Hofstetter, Kendall Knapp

Advised by Drs. Jing Wang, Yufeng Lu, and In Soo Ahn

6 December 2016

Table of Contents

Introduction

The system described in this document is a robot built to compete in the IEEE SoutheastCon Hardware Challenge. The robot is required to fit in a 12 inch cube and be completely autonomous during the competition. The robot attempts to complete four tasks accurately within a four minute time limit. It uses inputs for location, navigation, and other task-specific applications, uses those inputs to make decisions, and then moves itself around the arena and outputs some task-specific actions.

In Task 1, there are six copper pads arranged such that five pads form a pentagon surrounding a sixth, central pad. Each of the five outer pads are connected to exactly one of the following components: a wire, a resistor, a capacitor, an inductor, and a diode. One end of each component is attached to the center pad, which acts as a common reference point for all the components. The robot identifies the location of each component and saves the information for Stage 3.

For Stage 2, there is a rod sticking up above the arena representing a light saber. A magnetic force is randomly generated five times for two-second intervals within a thirty-second time frame. The robot must hit the rod with a robotic arm while the magnetic force is on, and is penalized for hitting the rod when no magnetic force is being created.

At Stage 3, the robot recalls the order of the components from Stage 1. Each component has a number associated with it (1-5), and the robot turns a knob multiples of 360° in the same order the components were discovered in Stage 1. For example, if the code were 3-2-5-1-4, the robot turns the knob 3 revolutions clockwise, then 2 revolutions counter clockwise, then 5 revolutions clockwise, then 1 revolution counter-clockwise, followed by 4 final clockwise revolutions.

To finish up the competition, Stage 4 requires the robot to attempt to get up to 3 Nerf darts into a box at the far end of the arena. The robot may shoot the Nerf darts or drive to the box and simply drop the darts in, but the arena has increasingly tall steps to get to the box, encouraging competitors to shoot the darts from below.

Review of Literature and Prior Work

Only the official competition rules and specifications provided by the IEEE SoutheastCon officials were consulted for literature specifically related to this project [1]. From this, the rules and requirements were obtained and designs for the robot were created to accommodate those rules.

Other literature was consulted to learn how particular components of the robot worked to interface those components with the central microcontroller; datasheets were primarily consulted for these components.

As this competition was designed specifically for the upcoming IEEE SoutheastCon Conference, no prior work is useful to consult regarding this project.

Applicable Standards and Patents

Due to the fact that this is a robot designed from scratch for a friendly competition by all competing teams, there are no applicable standards or patents that must be accounted for.

Subsystem Level Functional Requirements and Specifications

I/O Summary

System Level Inputs

Power: The robot has a simple on-off power switch. In the off position, no power is provided to the robot and the circuit board. In the on position, power is supplied.

Start/Stop Signal: There is a button that, upon its activation, sends a signal to the robot to start the actions necessary to complete the competition. This is different than the power switch because action is not desired immediately upon startup.

Camera Feed: The robot utilizes visual inputs by way of a camera connected to a microcontroller that performs basic image processing. The vision identifies the direction the robot is facing, approximate distance from each task, and robotic arm location relative to the current task.

Distance from Walls: The robot has distance sensors located on each edge of the robot to sense the distance between the robot and the walls.

Component Voltage: For Stage 1, a 5V step is applied to a circuit in which a resistor and the component are in series. The transient voltage response of the component is measured and the pattern of the response will determine which component is being tested. The analog to digital converter is implemented to read these voltages.

Magnetic Field: The robot senses a magnetic field and uses the readings to complete Stage 2.

Detect Light Saber Hit: The robot detects when the lightsaber hits its counterpart on the arena.

Angle of Robot: The robot climbs the steps on the arena to create an advantageous position for itself before it shoots the darts. Having the vertical angle of the robot allows for adjustments to the launch angle of the darts.

System Level Outputs

Indicate Subsystem Status: An LED light turns on for each subsystem when it is functioning properly. This output is used exclusively for debugging purposes.

Move Robot: The robot moves around the arena to each stage.

Move Robotic Arms: The arms of the robot need to be moved slightly to correctly complete each task, even when the robot is correctly located in front of each task.

Component ID Step Output: In Stage 1 the robot excites the components to observe their voltage characteristics and identify each component. This is a simple 5V step output.

Numeric Code from Stage 1: The robot displays the order of the components identified in Stage 1 to gain points for correctly identifying the components in the event that Stage 3 is not completed correctly.

Hit Lightsaber: The robot uses an arm to hit the stand-in lightsaber on the arena at the prescribed time.

Turn Knob: Upon receiving a signal of how many turns to execute, this clamp rotates the knob the correct number of turns before reversing direction to execute the next number of turns.

Shoot Nerf N-Strike Dart: To complete Stage 4, three Nerf N-Strike darts must be shot into a hole at the far end of the arena.

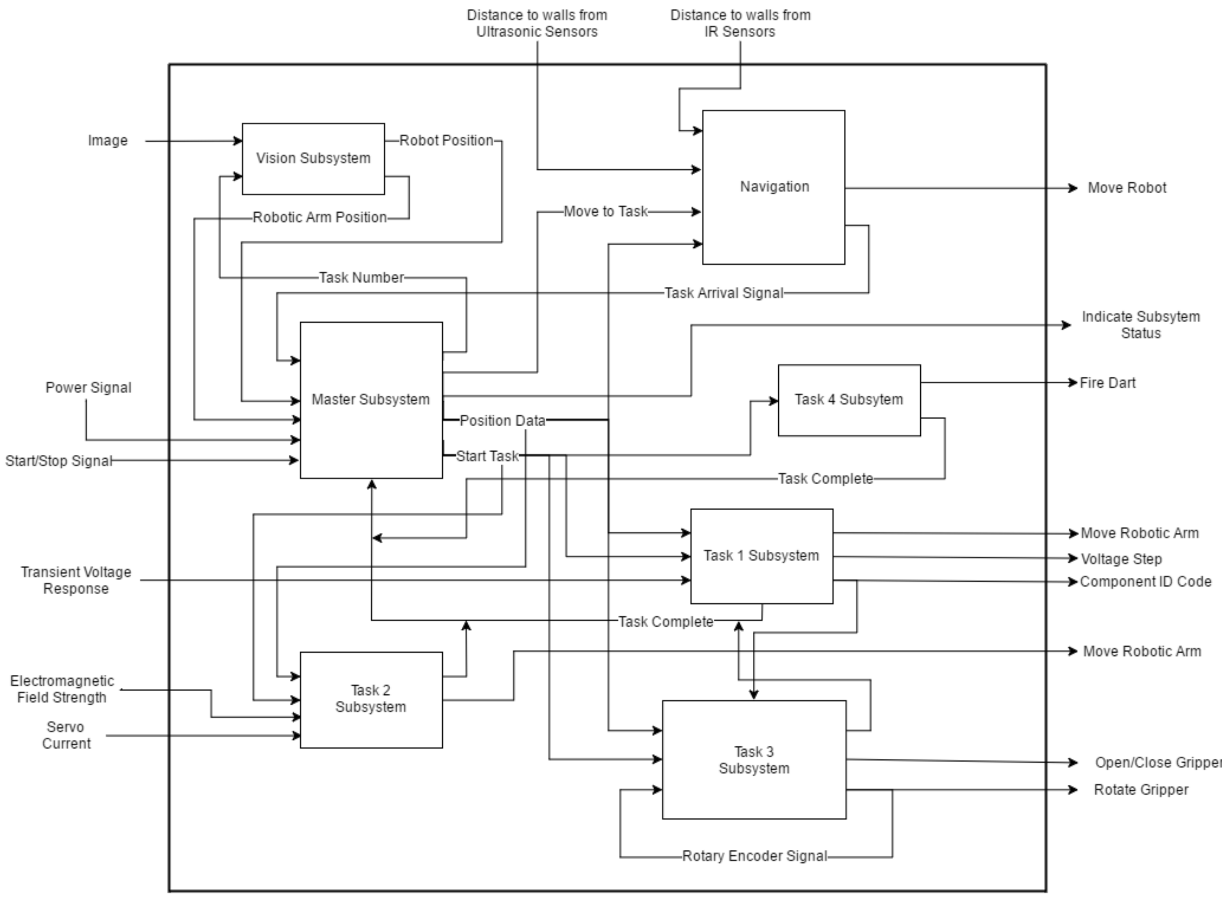
Modes of Operation

Status Check and Calibration: In the first state after robot is powered up, subsystems are checked and status is output to the LCD display. The robot locates itself in the arena in relation to each of the four stages.

Operating: After the Status Check and Calibration mode completes, the robot executes the functions of navigation and each of the four stages autonomously.

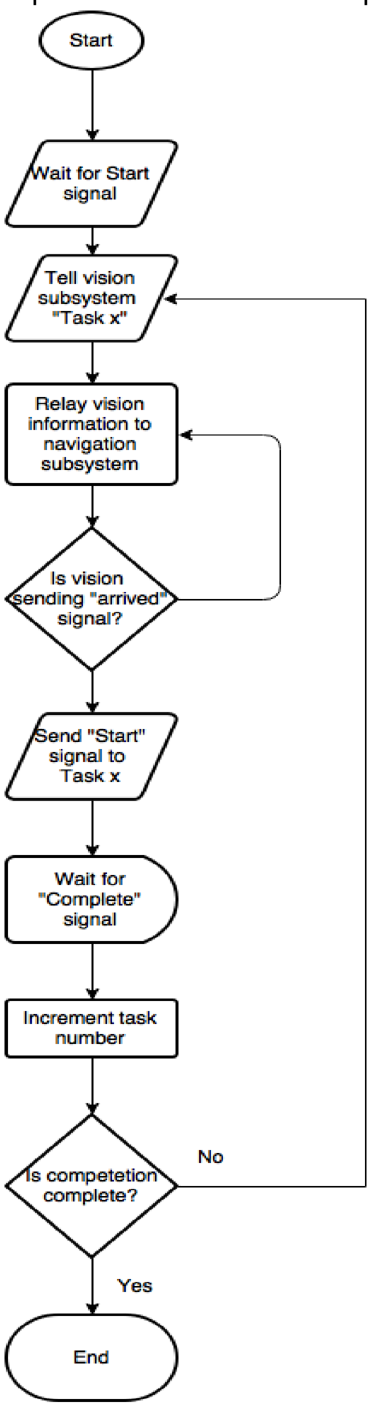
Mission Complete: The robot enters this mode upon completion of all the tasks and simply stays idle with no movement. This allows the power switch to be flipped and robot to be picked up safely.

System Level Block Diagram



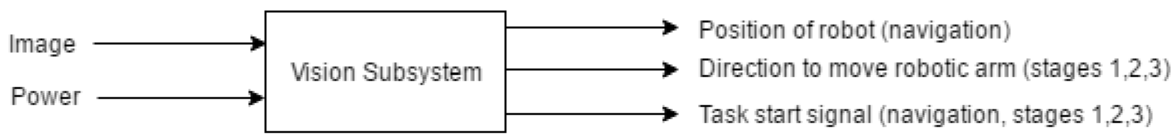
Master Subsystem

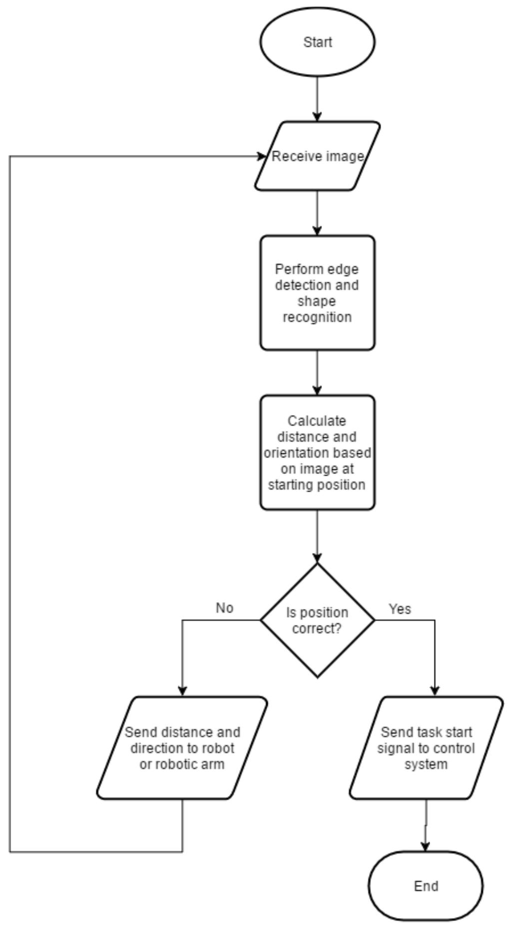
The master subsystem coordinates all other subsystems and runs the primary logic of the robot. This subsystem runs procedurally, coordinating efforts from each task in whichever order is prescribed, preventing, for example, the robot from attempting to complete Task 3 when it is positioned at Task 2. For the purposes of this exercise, this subsystem also relays information from one subsystem to other subsystems, but that does not mean such information sharing is implemented in a more complicated manner than is necessary.



Vision Subsystem

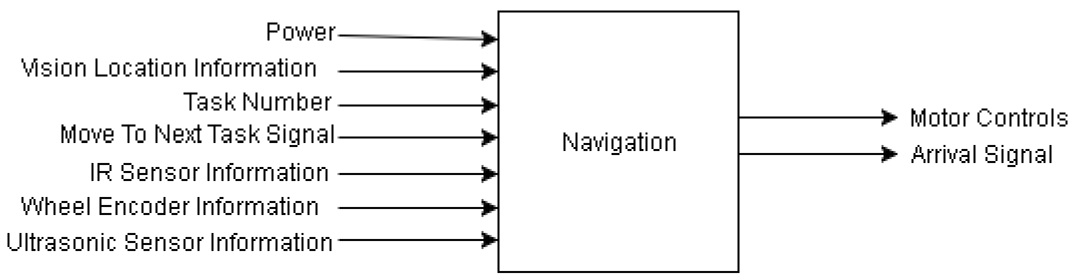
The input to the vision subsystem is a stream of images of the arena. The camera takes the images and sends them to the image processor. The image processing algorithm performs edge detection and identifies accepted shapes from the edges. These shapes include the components of the various stages and the robotic arms. Based on the position of the stage components and robotic arm, the image processor sends navigation instructions to the microcontroller regarding the direction to move the robot and/or robotic arm.

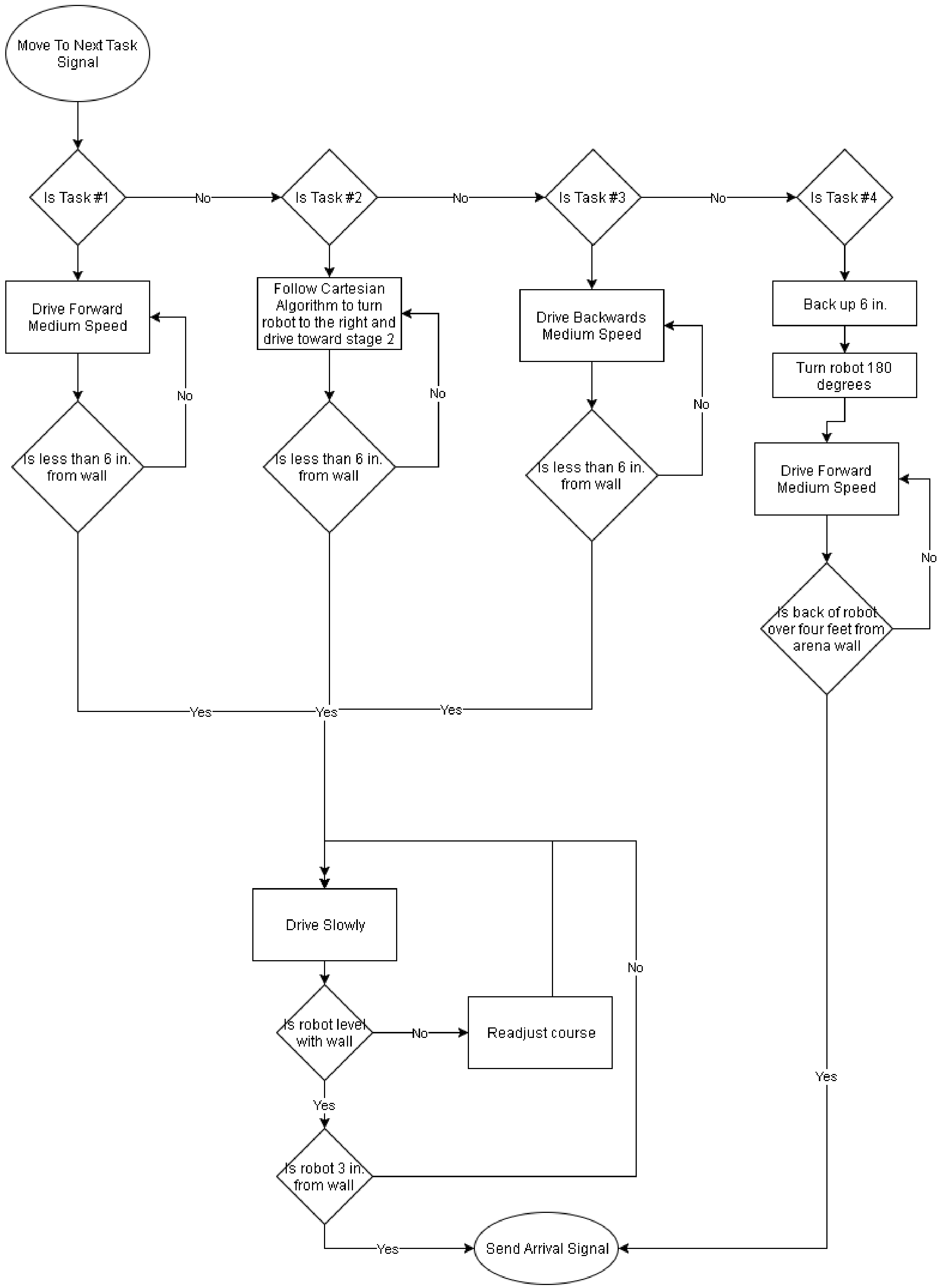




Navigation Subsystem

The navigation subsystem is in charge of correctly moving the robot to each task at the correct time. The subsystem keeps an internal record of where the robot should be located in the arena, and from each position the robot moves to the next task, which is dictated by the master subsystem. The vision subsystem provides orientation information to make sure that the robot is heading in the right direction. The wheel encoders provide a rough estimate of the distances and directions the robot has traveled, but due to the inherent error of encoders, the vision system and distance sensors correct and compensate for the encoder errors. Once the robot is close to the edge of the wall where the tasks are, the vision system is used to line the robot up with the task. This subsystem controls the drive motors and sends an arrival signal when it has arrived at the task it was commanded to move to.

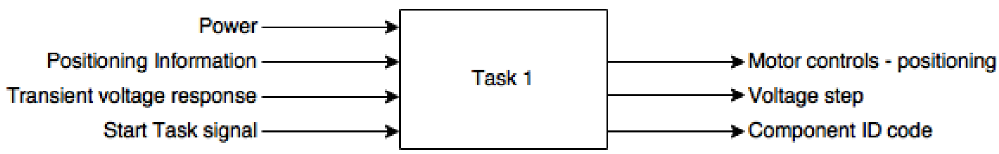


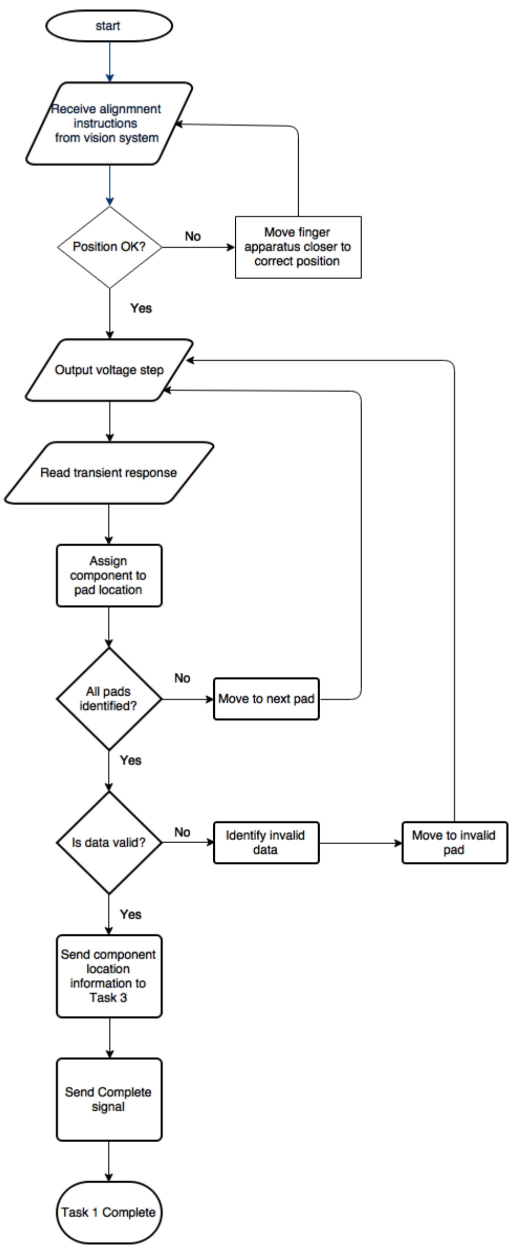


Move To Next Task Signal
Is Task #1
No
Is Task #2
No
Is Task #3
No
Is Task #4
Drive Forward Medium Speed
No
Is less than 6 in. from wall
Follow Cartesian Algorithm to turn robot to the right and drive toward stage 2
No
Is less than 6 in. from wall
Drive Backwards Medium Speed
No
Is less than 6 in. from wall
Back up 6 in.
Turn robot 180 degrees
Drive Forward Medium Speed
No
Is back of robot over four feet from arena wall
Yes
Yes
Yes
Drive Slowly
No
Is robot level with wall
No
Readjust course
Yes
Is robot 3 in. from wall
Yes
Send Arrival Signal
Yes

Component Identification - Task 1 Subsystem

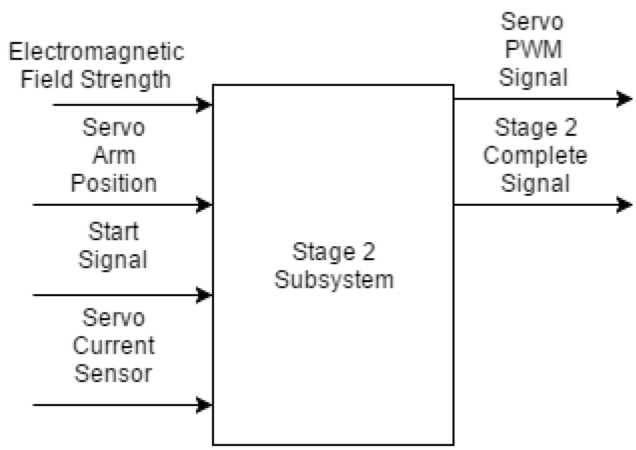
The component identification subsystem takes positioning information from the vision system and adjusts its fingers mechanism to align itself with the inset copper pads. When the start signal has been received from the vision system, the robot begins identifying the component connected to each pad. To accomplish this, the robot outputs a voltage step and then reads the transient response of each component via an analog-to-digital converter. Depending on the ADC results, the robot uses an algorithm and matches each component with a location. This component location information is output to an LCD screen and sent to Stage 3.

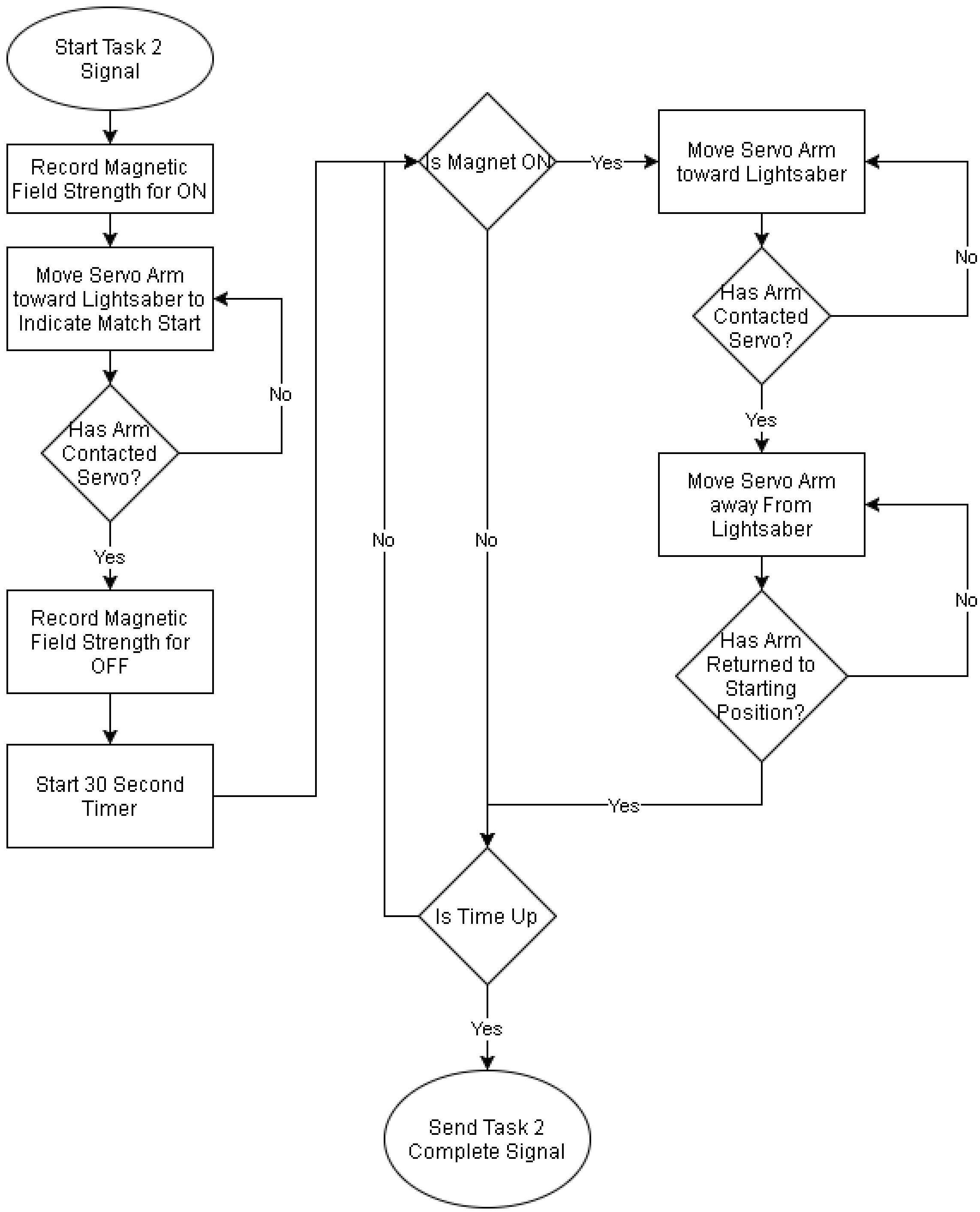




Lightsaber Duel - Task 2 Subsystem

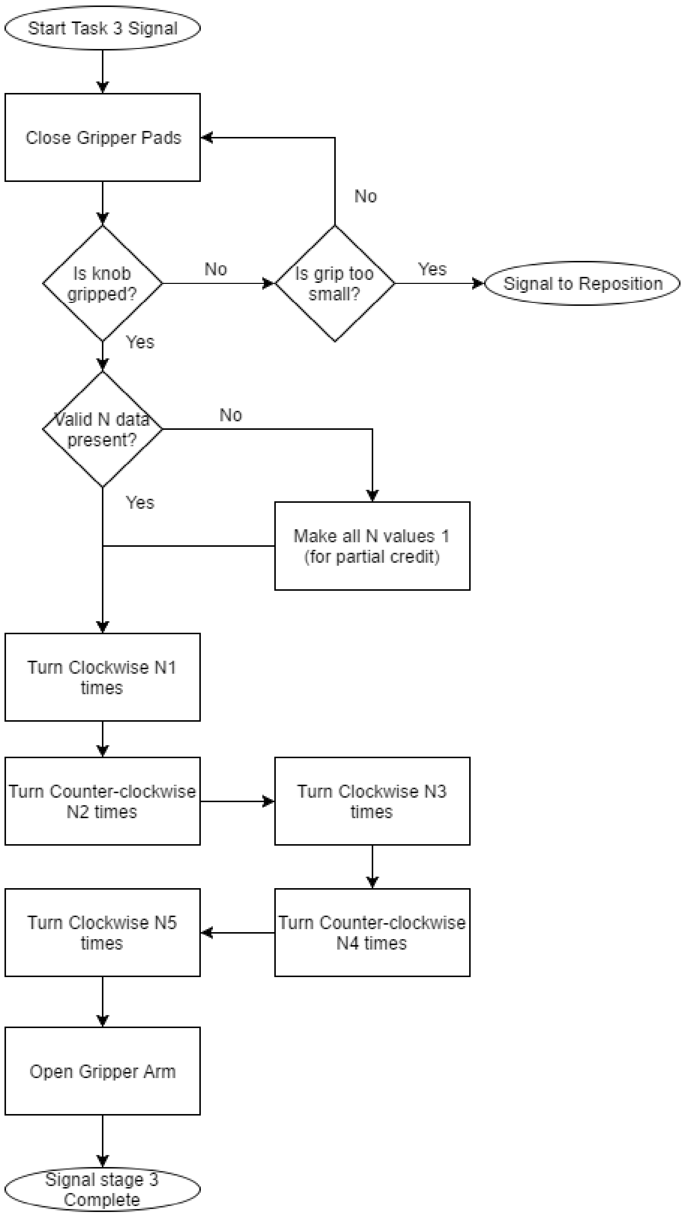
In stage 2, the robot must hit the hilt of the arena's lightsaber whenever the arena's electromagnet is activated. When the master subsystem sends the start command to the lightsaber subsystem, the servo swings at the lightsaber, indicating to the arena that the robot is ready for the duel. When this occurs, the arena switches off the magnetic field in preparation for the duel. Due to ambient electromagnetic noise, the robot uses an adaptive algorithm to measure the magnetic force before and after the first strike, thus ensuring accurate readings are made for on and off states of the magnetic field. When the field is activated, the arm quickly swings at the lightsaber. Current-sensing circuitry indicates when the servo's arm contacts the hilt of the arena's lightsaber, at which point the servo returns to the opposite direction. The duel lasts for thirty seconds, after which the subsystem sends a complete signal back to the master control subsystem.

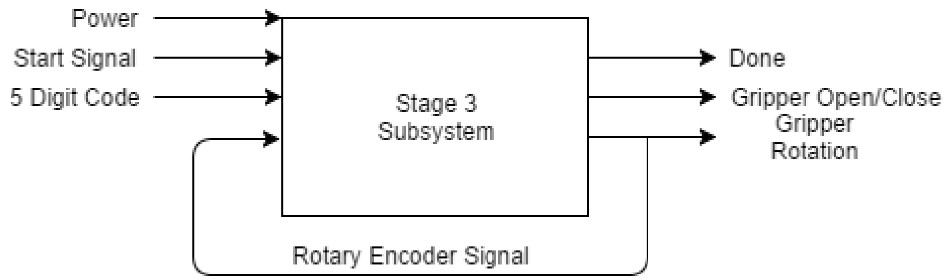



Start Task 2 Signal
Record Magnetic Field Strength for ON
Move Servo Arm toward Lightsaber to Indicate Match Start
Has Arm Contacted Servo?
No
Yes
Record Magnetic Field Strength for OFF
Start 30 Second Timer
Is Magnet ON
Yes
No
No
Move Servo Arm toward Lightsaber
Has Arm Contacted Servo?
No
Yes
Move Servo Arm away From Lightsaber
Has Arm Returned to Starting Position?
No
Yes
Is Time Up
Yes
Send Task 2 Complete Signal

Lock Turning - Task 3 Subsystem

The lock turning subsystem takes the code data collected from the component identification subsystem in stage 1 and uses it to turn an encoder the appropriate amount of turns.




Power
Start Signal
5 Digit Code
Stage 3
Subsystem
Done
Gripper Open/Close
Gripper
Rotation
Rotary Encoder Signal

Dart Firing - Task 4 Subsystem

The dart firing subsystem takes a signal from the navigation system as an input. This input tells the dart to fire when the robot has been lined up with the target. The only output is the dart firing after aiming is complete. The match ends once the last dart is fired.

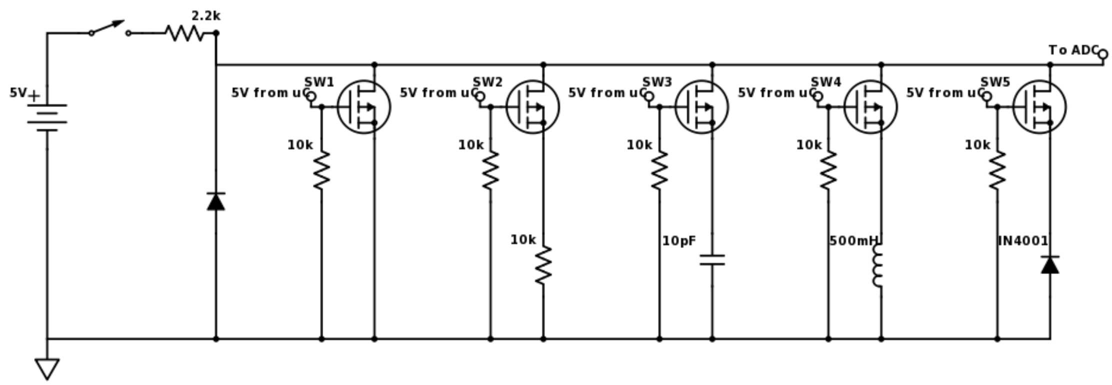
Engineering Efforts Completed To Date

Task 1 - Component Identification

The circuit used to generate a readable transient response has been created and tested to accurately identify each of the components used in the competition. Each component's transient response when placed in the circuit was recorded on an oscilloscope to verify timing and safe voltage levels. It was determined that a diode placed in parallel with the components was necessary to protect the analog-to-digital converter from a large negative voltage spike from the inductor when the step function switches to a low logic level.

Code has been written to sample the transient responses of the components and determine the pattern and voltage levels of each response. The microcontroller averages the first eight samples from the ADC after the step function switches to logic high and records the value as the starting average. After a delay another eight samples are averaged to determine steady-state voltage and records this value as the steady state voltage. A simple algorithm then determines which component was just tested by comparing starting average, steady state average, and, if those two values are similar, the steady state voltage to preset thresholds.

Once the microcontroller could accurately identify each component individually, a circuit was built to automatically switch between the five components using logic-level FETs that can be driven by the microcontroller. The schematic of the circuit is shown below.



Task 2 - Lightsaber Duel

The physical output of this task is very simple; the robot must move an arm that collides with the arena's lightsaber. The current design involves three components; a magnetic field sensor, an arm to strike the lightsaber, and a means of knowing when the arm has contacted the lightsaber. A multi-purpose sensor that has both a magnetic field sensor and an accelerometer (the accelerometer will possibly be used for task 4 or for navigation also) was chosen. The sensor uses I^2C to communicate with a microcontroller. A standard servo motor was the obvious choice for creating motion. A simple stick-like arm will be attached to the motor to act as the robot's lightsaber and make contact with the arena's lightsaber. The desired method of detecting that the lightsabers have struck one another is by monitoring the current going through the servo motor. When the robotic arm strikes the arena's lightsaber, the current draw of the servo will increase. By using current sensors on the motor power supply line, the current increase can be detected and the servo direction can be reversed when the arm hits the lightsaber. An alternate design is to mount an accelerometer on the robotic arm and detect the impulse created upon impact. Testing will be necessary to determine which method will work better.

All of the parts listed above, namely, the servo, magnetic field sensor, current sensor, and accelerometer have been obtained. However, progress on this task has been hindered. The electromagnet has not been constructed yet for testing the magnetic field sensor because there was a delay in obtaining the plastic spool for the electromagnet. Code has been written for generating a PWM signal for the servo motor, but has not been tested because the servo initially provided was faulty.

It is worth mentioning that while the current design uses a servo motor, there has been a desire to use either a linear actuator such as a solenoid or to simply drive the robot back and forth to strike the lightsaber. At this point, the plan is to evaluate both the plan of driving the robot back and forth along with the original plan of using a servo. Because the servo idea is the only one that has been designed, ordered parts for, and worked on, it is the only design described in detail here.

Task 3 - Lock Rotation

The physical output of the system is quite simple; the robot must grab onto the knob and then rotate it according to the code collected in the first stage. There are many ways to accomplish this task, and after consideration and testing, the following are the conclusions of what will be implemented onto the robot.

The part of the arm that will grab onto the knob is a simple servo driven gripper arm. The servo requires a PWM signal to switch from an open position to a closed position with enough holding torque to allow the arm to rotate without losing a grip on the knob. Functional PWM code has been created to accomplish the task outlined in the previous sentence.

There are many different ways to apply rotation to the gripper arm in order to rotate it according to the code from Task 1. It was determined that the motor used for rotation requires some form of rotary encoder or angle counter in order to be able to conform to the ±15 degree tolerance for the turns. There are 2 motor designs under consideration: a stepper motor and a DC motor with a rotary encoder. A DC motor will be tested first, and then switched to a stepper motor if the functionality is not determined to be adequate.

Task 4 - Dart Firing

The robot is designed to go up the first step and stop when it gets to the bottom of the second. This leaves a 2 ft distance the darts need to travel.

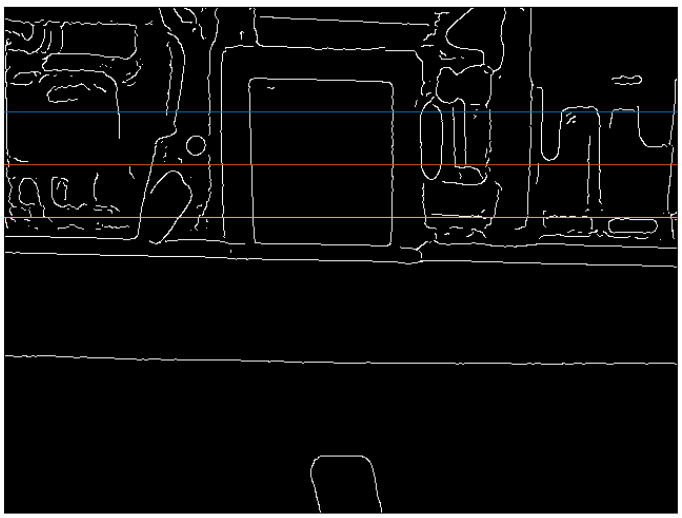
The purchased Nerf guns will easily traverse this distance, without having to be at an angle. They can travel for over 30 ft. However, the trigger takes a significant amount of force to pull. This would be difficult to accomplish with a linear actuator or motor. Each gun only holds one dart, so there would need to be three guns. Combined with the firing mechanism, this would take up a significant amount of room.

As an alternative, a custom prototype has been developed by Nick Schmidt. It uses a 0.5in hollow metal tube, slightly larger than the diameter of the dart. A spring is pushed into the tube and held in place by a pin pushed through a hole in the tube. Then, the dart is dropped into the tube. Pulling the pin out of the hole will release the spring and fire the dart. It travels more than 5ft. All of the loading can be done before the match starts. A firing mechanism is needed to pull the pin, but it will require much less force than the Nerf gun trigger. In addition, space will be saved.

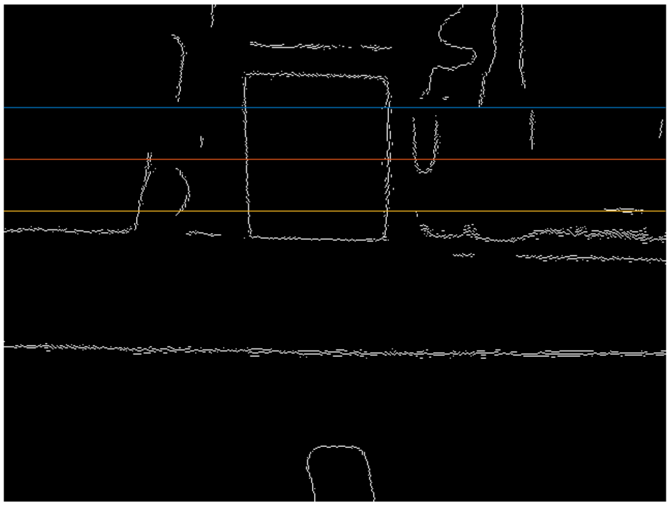
Vision Subsystem

Preliminary image processing testing was done in MATLAB using pictures taken of the constructed arena. Several difficulties were encountered, but a rudimentary distance measurement is developed. Still, a camera may not be the best way to navigate the robot.

The first method used was edge detection. The figure below shows stage 1 after performing Canny edge detection (no circles have been added yet). The goal was to use the vertical lines shown on the figure to measure the average width of the square. After measuring the width from images taken at various distances, a function for distance to the stage could be derived. Unfortunately, the background produces too many edges to make this method viable. It is very difficult to reliably distinguish between the arena and the background.

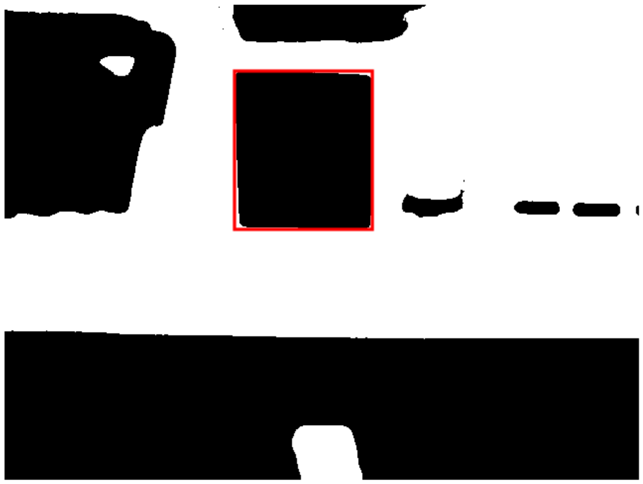


The team member tried to solve this problem by blurring the image prior to edge detection and using the Sobel method instead of Canny. These two changes result in much less background noise, but also less detail in the foreground. The resulting image is shown below. It may be useable, but a new method of measurement is needed.



Finally, the team member converted the image to a binary image using a threshold of 100. Anything with a grayscale value of more than 100 becomes white; everything else is black. This value was determined experimentally and could change with different lighting levels. Next, the regionprops() MATLAB function is used to get the major and minor axis lengths for each black region. One of these regions is the target stage. To automate the location of this region, several methods are used. First, the ratio of the minor to major axis is compared. A ratio greater than 0.8 is a possible match. The actual ratio should be close to one because the real stage is a square, but the constructed stages are currently incorrect and have a ratio of 0.83. Next, the candidate regions above a certain area are selected. This area varies with distance, but will always be fairly large. From the remaining regions the one with the ratio closest to 0.83 is selected.

While not perfect, this method reliably selects the appropriate region. The height of three regions are measured at 21in,15in, and 10in from the stage. From these data points a second degree polynomial is derived. Then, other distances can be calculated based on the height of the stage in the image. The results are within 0.5in of the expected distance. Factors causing error include variations in lighting that cross the threshold and inaccuracies in distance measurements when the picture was taken.



Detecting regions from a binary image provides a viable method of distance measurement using a camera. However, it works best when the camera is centered with the stage. If the robot is at an angle, the ratio changes and it the wrong region can be detected. Also, with obstructions and changes in lighting, it is difficult to get the precision necessary to determine the angle of the robot. More work needs to be done, but these results lead the team to believe ultrasonic sensors may be more effective at distance and angle measurements.

Navigation Subsystem

The Parallax Ping ultrasonic distance sensor has been tested and functions well. If the sensor is within 2ft of the arena wall, it has an accuracy of 0.1in. If the sensor is located further than 3ft from the wall, the reading can jump significantly. This is probably due to the low arena wall. Accurate readings still occur, but the outliers will have to be filtered out.

The ultrasonic sensors will also need to determine the angle of the robot in relation to the wall. Mounting two sensors on each side of the front and measuring the difference between the two readings should accomplish this. More testing needs to be done to determine if there is sufficient accuracy.

Infrared sensors were also tested, but could only be used as proximity sensors to sense when the wall is close, not to measure the exact distance. Additionally, the distance from the wall varies with the lighting conditions and color of paint, making them unreliable. Ultrasonic will work in any lighting conditions.

Touch sensors are more promising. Simple push buttons on the front of the robot could signal that the robot is touching the wall. With one on each side of the front, it would be simple to position the robot perpendicular to the wall.

The sensor data would be useless if they cannot be used to precisely control the movement of the robot. Therefore, a motor control system with encoder and sensor feedback is necessary. So far, PWM code has been written that uses the internal timers to automatically generate dual PWM signals for the left and right motors without using any interrupts. This will free up the microcontroller for other tasks, like monitoring the encoders and performing control algorithms. The H-bridge chosen to interface the motors to the microcontroller is the L298 dual H-bridge. This chip has a rating of 2.5 amps max current per channel with an 80% duty cycle. Because the stall current of the motors is 2.5 amps, this will work well. A stall condition should never be entered considering that the friction between the tires and the wooden arena platform is low enough so that the wheels will spin instead of locking up. A testing circuit for the H-bridge has been built, and the PWM code is ready to be used to navigate the robot.

Robot Chassis Construction

The robot chassis is completed. The design includes a 10” square base platform upon which two 2 ⅝” drive wheels are attached at the ‘front’ of the robot. These drive wheels are driven by 2 DC motors with rotary encoders powered by connection to a dual H-bridge. The rear of the robot contains two 1” ball bearings which allow for the robot to precisely navigate in both the forward and backwards direction.

The rotating platform above the base is another 10” square upon which the components of the robot can be attached. The rotation is done with a Lazy Susan bearing rotator, which allows for a simple and smooth rotation relative to the base configuration. This rotation allows for the ‘left’, ‘right’, and ‘front’ of the robot to all be at the front of the robot when executing the tasks.

There is currently no motor in place to rotate the top of the platform as that will be determined according to which type of motor has enough power and will be small enough to allow for all of the other components to fit on. A high torque servo motor will be tested to see if it has enough power to rotate and hold the top platform. If that solution does not work, the platform will instead be rotated by a high torque stepper motor.

The platform is built and ready for each individual component to be mounted on. The space below the robot is 3” and the height of the base of the top platform is approximately 4.5”.

Parts List

A list of parts that have been purchased or may be purchased in the future is shown below. The current total of $332 is well within the budget of $1000. While more purchases are expected, the total should not exceed $500.

Description	Quantity	Cost (if known)	Item Obtained
Arena			
Wood		$40	Yes
Paint	2 Quarts	$20	Yes
Medium Vibration Sensor	1	$0.95	Yes
Clear Plastic Knob	1	$0.95	Yes
Rotary Encoder - Illuminated	1	$3.95	Yes
Task 1			
Linear Actuator	1	$20	Yes
Button	1		No
Logic-Level FETs	5		No
Low forward voltage diode: < .3V	2	On hand	Yes
LCD	1		No
Task 2			
MPU-9250 Magnetic Field Sensor	1	$15.99	Yes
Standard Servo Motor	1	On hand	Yes
ATmega168A, DIP	1	On hand	Yes
Hall Effect Current Sensor	1	$4.50	Yes
Task 3			
Lynxmotion Little Grip Kit	1	$15.99	Yes
HS-422 Servo	1	$9.69	Yes

Task 4			
Nerf Jolt Guns	3	$18.97	Yes
Vision			
Raspberry Pi 5MP Camera Board Module	1	$22.95	No
Raspberry Pi 3 Model B	1	$35.70	No
Chassis			
Parallax Ultrasonic Sensor	4	$29.99	Yes
Auto EC 68 mm wheels x4	1	$9.99	Yes
4'x8' MDF Sheet	1	$19.17	Yes
6" Lazy Susan Everbilt	1	4.48	Yes
#3 screws x10	2	$6.28	Yes
1" Zinc Coated Caster Ball Vestil	2	$6.50	Yes
Pololu Wheel 42x19 mm	2	$13.96	Yes
12V DC Motor W/rotary encoder	3	On hand	Yes
Right Angle DC Motor mount 37mm	3	$26.97	Yes
Total:		**$332**	

ECE 499 Deliverables

- Final Project Report
- Final Project Presentation
- Final Project Demo
- Industry Advisory Board Poster Presentation
- Bradley Student Scholarship Expo

Division of Labor

Each team member is responsible for one stage, designing the components necessary to complete it. In addition, everyone will be in charge of a functionality of the integrated robot, making sure the different parts are combined successfully.

Team Member	Functionality	Stage
Cameron	Robotic Controls	1: Component ID
Daniel	Navigation	2: Lightsaber Swing
Brian	Circuit Design Robot Chassis Layout/Design	3: Turning Knob
Kendall	Vision System Navigation Sensors	4: Launching Dart

Schedule for Completion

The robot must be ready for the competition on March 30. This is a sooner deadline than most projects. As a result, the team will need to work during break in order to complete the project in time. Each team member will have their individual tasks complete soon after break is over. Then, the semester can be spent integrating the subsystems and fine-tuning the system. The timeline assumes 6 hours of work weekly for each team member, for a total of 24 hours per week. The majority of time is given to integrating subsystems and fine-tuning the system in order to account for unexpected problems.

Task	Deadline
Have moving robot	Dec. 8
Individual tasks working	Jan. 27 (requires work during break)
Integrate subsystems	Feb. 24
Fine-tune system	Mar. 24
Competition	Mar. 30

Discussion and Future Directions

Progress has been made on each task. However, more work needs to be done before the subsystems can be integrated. Completing the chassis is a significant step that will help future efforts. Once the robot is integrated and tested, the problems that need to be focused on will become apparent. The camera may be eliminated in favor of distance sensors. Testing will determine how accurate robotic arm movements need to be and whether the dart firing mechanism is sufficient without aiming functionality.

As the competition nears, the team will need to reevaluate the system requirements. One or more tasks may need to be eliminated or reduced in priority in order to complete the tasks most likely to succeed. This will maximize points at the competition. While the team would like the robot to function under any conditions, risks may need to be taken to complete the project in the short time frame.

References

[1] *SoutheastCon 2017 Hardware Competition Rules* [Online]. Available: http://sites.ieee.org/southeastcon2017/student-program/student-hardware-competition/