**ECE 48800 Final Submission Check List**

**Project Title: \_\_\_\_**ASEE Robotics Competition\_\_\_\_\_\_\_\_\_**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Project Sponsor: Name: \_\_\_\_\_**Michael Golub\_\_\_**\_\_\_\_\_\_\_\_email: \_\_**migolub@iupui.edu**\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Each group member’s credit for this project:**

Name: \_Blake Bennett\_\_\_\_ credit: \_\_\_% Member’s Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: \_Dallas Thornton\_\_ credit: \_\_\_% Member’s Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: \_Glenn Haag\_\_\_\_\_\_ credit: \_\_\_% Member’s Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: \_Torben Eid\_\_\_\_\_\_ credit: \_\_\_% Member’s Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ credit: \_\_\_% Member’s Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Two identical paper folders with pockets that each folder contains**

Yes \_\_\_ The folder cover with the course number, project name, names and e-mail of group members, the sponsor’s name, phone number, and e-mail

Yes \_\_\_ The paper copy of the final report

Yes \_\_\_ This Final Submission Check List page on the first page of the report

Yes \_\_\_ A flash drive that stores all items generated in the course the course (all written reports, software, notebook, references, manuals, schematics, PCB design files, 5 minutes hardware/software/design demonstration video, etc., except hardware).

**Two copies of folders must be submitted**

Yes \_\_\_ Two identical folders (one for sponsor and one for course instructor) submitted

1. **Copy all files in the flash drive to Canvas in a folder called ProjectName-Final-Folder-Date**

Yes \_\_\_

**Submission Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**3/22/2018\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Final Project Report for ASEE Robotics Competition

ECE48800 Spring 2018

Sponsor: Michael Golub, ME Department, IUPUI

[migolub@iupui.edu](mailto:migolub@iupui.edu)

(317)–486–3490

Blake Bennett [[blakbenn@iupui.edu](mailto:blakbenn@iupui.edu)]

Torben Eid [[teid@iupui.edu](mailto:teid@iupui.edu)]

Glenn Haag [[ghaag@iupui.edu](mailto:ghaag@iupui.edu)]

James Thornton [[jamthorn@iupui.edu](mailto:jamthorn@iupui.edu)]

3/23/2018

**Table of Contents**

1. Project Motivation………………………………………………………1
2. Project Background Information ………………………………………..1
3. Project Requirements……………………………………………………1  
   3.1 Technical Requirements…………………………………………….1  
   3.2 Standards…………………………………………………………….2  
   3.3 Constraints………………………………………………………….. 2  
   3.4 Safety Requirements…………………………………………………2  
   3.5 Other Requirements………………………………………………….2  
   3.6 Schedule Requirements………………………………………………3
4. Design Options…………………………………………………………..3  
   4.1 Locomotion…………………………………………………………..3  
   4.2 Navigation…………………………………………………………...4  
   4.3 Retrieval System…………………………………………………….5  
   4.4 Sorting Mechanism………………………………………………….6  
   4.5 Deposit System………………………………………………………7  
   4.6 Design Options Summary……………………………………………8
5. Design Details……………………………………………………………9  
   5.1 System Block Diagram………………………………………………9  
   5.2 Locomotion Subsystem……………………………………………..10  
   5.3 Navigation Subsystem………………………………………………12  
   5.4 Retrieval Subsystem…………………………………………………13  
   5.5 Sorting Subsystem…………………………………………………..15  
   5.6 Deposit Subsystem………………………………………………….18  
   5.7 Software…………………………………………………………….19  
   5.8 Hardware Schematic………………………………………………..20  
   5.9 Bill of Materials……………………………………………………21  
   5.10 List of Standards………………………………………………….22
6. Implementation…………………………………………………………22
7. Testing Procedure……………………………………………………..25

7.1 Locomotion Testing……………………………………………….25

7.2 Navigation Testing………………………………………………..26

7.3 Retrieval Testing………………………………………………….28

7.4 Sort Testing……………………………………………………….28

7.5 Deposit Testing…………………………………………………...29

7.6 Full Robot Testing……………………….……………………….30

7.7 Testing Results……………………………….…………………..30

* 1. Testing Analysis …………………………………………………31

8. Conclusion……………………………………..……………………..31

8.1 ASEE Robotics Competition Conclusion………………………..31

8.2 Lessons Learned………………………………………………….32

* 1. Future Work………………………………………………………32

1. Appendices……………………………………………………………33

Appendix A: References………………………………………………33

Appendix B: Lesson Plans……………………….……………………34

**1. Project Motivation**

The motivation of this project stems from an attempt to garner more interest and participation in robotics at IUPUI. The university has taken part in a handful of competitions over the previous years and is looking to improve its standing among other schools. By creating the robot and mentoring younger students as they build their own unit for competition, it helps to create community amongst the students and foster more desire for participating in future competitions. A secondary motivation to this project is that IUPUI Robotics might bring home a win at the ASEE Robotics Competition, giving our school the recognition it deserves as a competitive engineering school.

**2. Project Background Information:**

The American Society of Engineering Education (ASEE) holds competitions each year for two-year universities and first and second year students of four-year universities in the field of autonomous robotics. This year the competition is being held in Salt Lake City, Utah in June. It will host students from multiple schools that will be competing on a board designed to mimic retrieving honey from a beehive. The competition has a points system where teams score ten points for each ball delivered to the correct pocket and five points for a ball delivered to the wrong pocket, which allows for a total of 130 possible points. Bonus time points are allotted based on the formula (120 – time to complete the run). If balls are placed in the wrong pocket but completed under time the bonus points are divided by two.

**3. Project Requirements**

**3.1 Technical Requirements**

The robot must be completely autonomous and cannot be touched once it is placed on the board. It must be placed before the balls are placed. It must navigate the board and retrieve all thirteen balls. Each ball may be one of two colors and must be identified as such so that it can be placed in the corresponding pocket at one of the four corners of the board. This must all be accomplished within a two-minute time limit. The robot must be able to retrieve the balls without breaking, or damaging them in any way. The robot must also not damage the board in any way.

**3.2 Standards**

The competition can only have first and second year students from community colleges or four-year universities. Due to this, we are only acting in a mentoring competition and our build is for instructional purposes only. The robotics students that participate will build their own version of the robot.

**3.3 Constraints**

The robot must fit inside a box of dimensions 8” x 12” x 10”. There may be no use of complete commercially available vehicles, robots, or entire kits such as RC cars, Legos, K-nex, Fischer-Technics, Parallax, or erector sets may not be used. The use of Lego Mindstorm microcontroller bricks are prohibited. Individual components from these cars, robots, or kits (except the Mindstorm Brick) may be integrated into a team’s robot as long as the majority of the robot’s components are not from the same car, robot, or kit source. The cost of purchasing all components must not exceed $400.

**3. 4 Safety Requirements**

The robot may not emit any gases, liquids or solids and the energy source cannot present a safety hazard. The use of drones or any other flying robots is prohibited due to the competition being held in a crowded hall.

**3.5 Other Requirements**

The competition is being held in June, so the initial project stated that a projected design be presented by February 1, so that the robotics team has time to build the robot. There is also an exhibit portion of the competition where teams will present their robot with any supplementary poster boards or multimedia presentations they deem necessary. There will be judges to evaluate design evolution, robot operation, fabrication methods, design analysis and exhibit quality. This will be part of the overall score for the weekend in addition to the actual time trials.

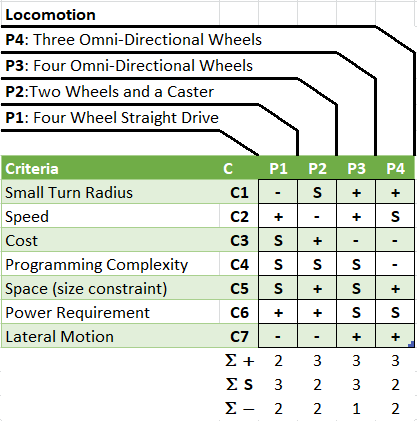
**3.6 Schedule Requirements**

All lesson plans are due by the first of February so that a group of students to be mentored have enough time to learn how to construct and program a robot. The mentoring process, if started by this point, will run concurrently with the design and testing of the ASEE Robotics Competition Senior Design robot.

**4. Design Options**

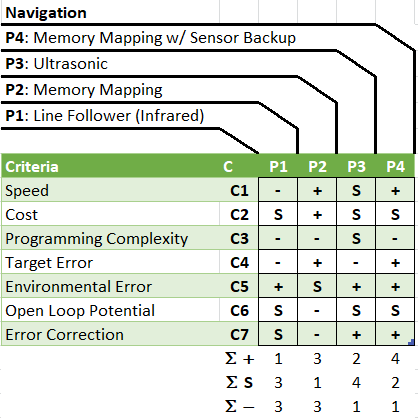
**4.1 Locomotion**

For our robot to move, we have envisioned four possible locomotion systems. The first (P1 of Figure 1) is a typical four wheel, straight drive system, similar to a car. This system will have an axle to turn the front wheels and will be rear wheel driven. This design requires two motors. The second design (P2 of Figure 1) involves two wheels and a caster wheel. This alleviates the difficulty of controlling the front axle, as the caster is free moving. The caster would be placed in the front-center of the robot, at the midpoint of where the axle would be in a four wheel, straight drive system. The two wheels are motor driven and also require two motors. The third design (P3 of Figure 1) requires four omni-directional wheels with each wheel located similarly to a car, but skewed at a 45 degree angle, pointing inward. This allows the robot to drive forward and backward and also rotate on its central axis, allowing it to pivot and move laterally unlike the four wheel, straight drive system. This design requires four omni-directional wheels and four motors. The final design (P4 of Figure 1) is a three omni-directional wheel system. The system has one wheel on each side, and one wheel at the front and one at the back. Both side wheels face forward while the front and back wheels are rotated at a 90 degree angle from the side wheels, making them perpendicular. Because of the nature of omnidirectional wheels, this will allow the robot to move in any direction with a minimal amount of rotation. This design, while slightly more costly than the three omni-directional wheel design, will likely be faster and more robust. All of the designs were considered and evaluated with a Pugh Matrix (Figure 1). From that, it was determined that the four omni-directional wheel system (P3 of Figure 1) excelled in speed, turn radius, and ability to move laterally. Its size, programming complexity, and power requirement were deemed satisfactory. Its only downfall was cost, though compared to our four hundred dollar budget, their cost was not much of a concern. Overall it was decided that the four omni-directional wheels would best meet the needs of our robot.

****Figure 1: Locomotion Pugh Matrix

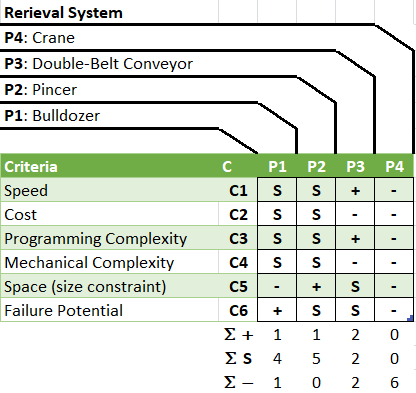
**4.2 Navigation**

For our robot to navigate, four design options were considered. The first (P1 of Figure 2) was a line following algorithm that requires the use of an infrared sensor. Based on the value read in from the sensor, the robot can follow the black electrical tape located on the competition board. The second design option (P2 of Figure 2) is a memory map. Basically the robots position is mapped in its memory and altered based on an empirically tested motor speed to distance travelled ratio. This design was first hypothesized by Jaime, one of Dr. Tovar’s PhD candidates. The third option (P3 of Figure 2) considered was allowing the robot to navigate using ultrasonic distance sensors, giving it a relative position based on the location of all four walls. It will be able to navigate freely and determine its position based on the signals sent from all four sensors, mounted on each side of the robot. The fourth option (P4 of Figure 2) is a combination of memory mapping with ultrasonic distance sensors used as an error checking mechanism to correct any slight errors that may occur in the travel of the robot. This allows for the use of positive attributes of the two options as well as minimizing their negative attributes. All of the designs were considered and evaluated with a Pugh Matrix (Figure 2). From that, it was determined that memory mapping with sensor backup (P4 of Figure 2) excelled in speed, target error, and environmental error. Its cost and open pool potential were deemed satisfactory. Its only downfall was programming complexity, which was the least of our worries as it will give us a chance to test our programming skills in a real world environment. Overall it was decided that the memory mapping with sensor backup would best meet the needs of our robot.

Figure 2: Navigation Pugh Matrix

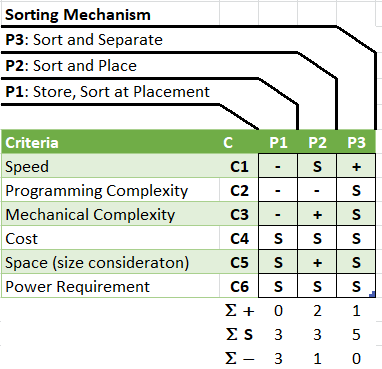
**4.3 Retrieval System**

The third design consideration for our robot is its retrieval system. The first option (P1 of Figure 3) considered was a bulldozer-like design. This design is intended to scoop the ball, but will also have a pincer-like closing mechanism located on the front of the dozer, intended to trap and push the ball to the back of the dozer. The second (P2 of Figure 3) is a stand-alone pincer arm which will grab the ball and drop it into its proper receptacle. The third option (P3 of Figure 3) is a double belt conveyor. This design uses two motor drive conveyors to pull the ball into the robot where it will be deposited into its proper receptacle. The final option (P4 of Figure 3) considered was a crane, which would grab the ping pong ball from above and drop it into its proper container. All of the designs were considered and evaluated with a Pugh Matrix (figure 3). From that, it was determined that the pincer design (P2 of Figure 3) excelled in size. Every other aspect considered in the Pugh Matrix was deemed satisfactory. The design had no significant downfalls. Overall it was decided that the pincer design would best meet the needs of our robot.

****Figure 3: Retrieval System Pugh Matrix

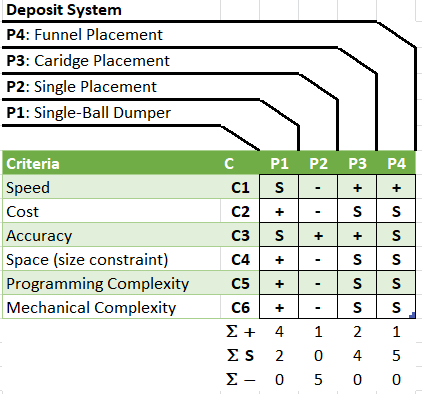
**4.4 Sorting Mechanism**

The fourth design consideration for our robot was the manner in which it would sort the colored ping pong balls. We were certain that this would require a color-to-light frequency converter, so rather than consider a mechanical design we focus on the methodology for this mechanism. The first possibility considered (P1 of Figure 4) was to store each ball as it was retrieved and then to sort them individually, placing each ball in its respective competition board receptacle. The second possibility considered (P2 of Figure 4) was to sort the balls as they were picked up and then place them immediately. The third possibility (P3 of Figure 4) was that our robot could sort each ball as it was retrieved and then place them into a color-specific container on the robot, allowing for quick depositing of the balls after they were all retrieved. All of the designs were considered and evaluated with a Pugh Matrix (Figure 4). From that, it was determined that the sort and separate method (P3 of Figure 4) excelled in speed. Every other aspect considered in the Pugh Matrix was deemed satisfactory. The design had no significant downfalls. Overall it was decided that the sort and separate method would best meet the needs of our robot.

Figure 4: Sorting Mechanism Pugh Matrix

**4.5 Deposit System**

Our final design consideration was for the robot’s depositing system. The robot needs to drop the ping pong balls in their proper receptacles. The first design considered (P1 of Figure 5) was something similar to a dump truck. The robot would position itself next to the proper receptacle and dump the balls into it. The second option considered (P2 of Figure 5) was a single placement system, like a robotic arm. This time consuming but precise design involves picking each ball individually and placing it lightly in the competition board receptacle. The third option (P3 of Figure 5) involved placing all like-colored balls into a cartridge-like device and placing or dropping the device into the receptacle. The final design option (P4 of Figure 5) is a funnel placement system. This system funnels all of the balls into a tube that rolls them gently into the receptacle. All of the designs were considered and evaluated with a Pugh Matrix (Figure 5). From that, it was determined that the single-ball dumper (P1 of Figure 5) excelled in speed. Every other aspect considered in the Pugh Matrix was deemed satisfactory. The design had no significant downfalls. Overall it was decided that the single-ball dumper would best meet the needs of our robot.

****Figure 5: Deposit System Pugh Matrix

**4.6 Design Options Summary**

By using Pugh Matrices as the crux of our design option selections we determined the final designs for each of our subsystems. The designs chosen were: four omni-directional wheels for the Locomotion Subsystem, memory mapping with sensor back-up for the Navigation Subsystem, a pincer for the Retrieval Subsystem, sort and separate as our Sorting Subsystem methodology, and a single-ball dumper as our Deposit Subsystem design. A summary of our selections can be viewed in the table below (figure 6).

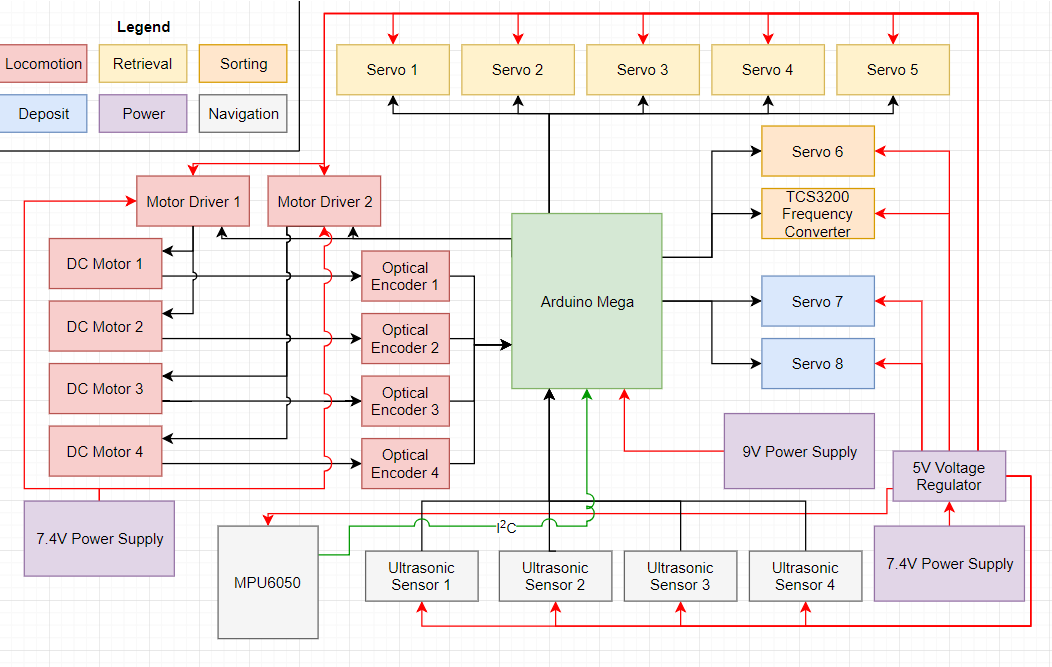
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Design Option 1 | Design Option 2 | Design Option 3 | Design Option 4 |
| Locomotion Subsystem | Three Omni-directional Wheels | **Four Omni-directional Wheels** | Two Wheels and a Caster Wheel | Four Wheel Straight Drive |
| Navigation Subsystem | **Memory Mapping with Sensor Backup** | Ultrasonic Sensors | Memory Mapping (No Backup) | Infrared Line Follower |
| Retrieval Subsystem | Crane | Double Belt Conveyor | **Pincer** | Bulldozer |
| Sorting Subsystem | **Sort and Separate** | Sort and Place | Store and Sort at Placement | N/A |
| Deposit Subsystem | Funnel Placement | Cartridge Placement | Single Placement | **Single-Ball Dumper** |

Figure 6: Summary table of Design Options, selected options bolded and highlighted

**5. Design Details**

**5.1 System Block Diagram**

Our system block diagram, located in Figure 7, below, shows how our hardware components will be connected. The red blocks represent the locomotion subsystem, yellow the retrieval subsystem, orange the sorting subsystem, and blue the deposit subsystem. Everything is connected to the Arduino Mega. Purple blocks represent our power supplies for the system and use red traces to show their connections to various components.

Figure 7: System Block Diagram for WorkerBot

**5.2 Locomotion Subsystem**

The locomotion system features four omni-directional wheels. There are two pairs of wheels. Two wheels that make up a pair are parallel with each other, while the two pairs are perpendicular to each other, creating a cross or plus sign configuration (Figure 8). Due to the nature of omni-directional wheels, this locomotion system will allow our robot to travel in any direction on the x/y plane without rotating. Should rotation be necessary (proper orientation for retrieval and depositing) the robot can rotate in place without altering its x/y position. Each of the four wheels will be driven by a DC motor which will be powered using 7.4 volts, powered through a motor driver. Two motor drivers connected to the Arduino Mega will provide input to each pair of motors. The locomotion system’s chassis will serve as the base for the Arduino Mega as well. A 3D model of our chassis can be seen in Figure 9. The motors are driven through software by using pulse width modulation, which sets a duty cycle to an output pin, mimicking an analog signal to affect the speed. The robot ran typically at a little less than half speed with a digital value of 112 out of 255 bits of resolution. This equated to roughly 3.25 volts. When trying to adjust for error correction the motors were move at an even lower analog voltage with digital values of 70 – 90. This allowed for much lower speed so that the corrections could be more fine grained and we could dial in to the exact place in space that we want the robot to be.

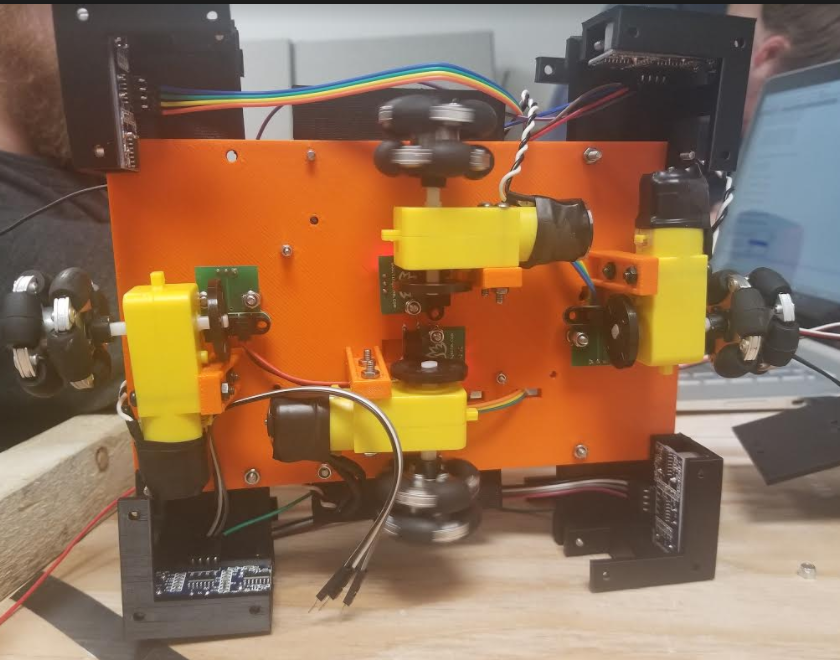
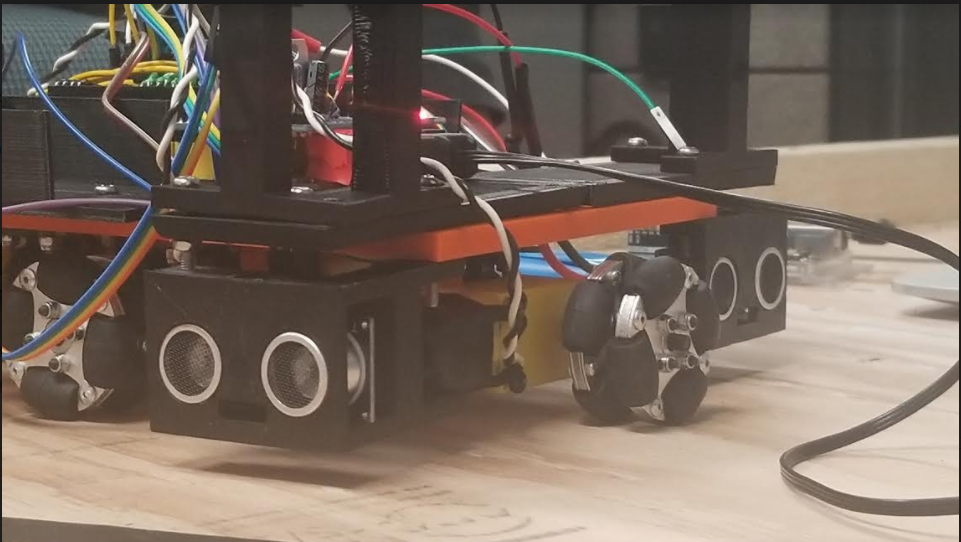
  
Figure 8: Locomotion Subsystem

  
Figure 9: 3D model of the WorkerBot’s chassis

**5.3 Navigation Subsystem**

The navigation system will utilize the fact that we are implementing our robot in a static environment. The goal of retrieving and depositing balls is tied to concrete points in space which never vary their relative space to one another. Knowing this, we can use an array of ultrasonic sensors (figure 10), mounted at each of the cardinal directions, north, south, east, and west, to track where the robot is at all times. By placing the robot in a desirable place to pick each ball up, we can measure the distance it is in from the wall and use it to program an array of goal distances in the software. By changing state between retrieving balls, we can update the robots next desired goal distance and run the motors in the direction of that ball. Likewise we can pivot and turn the robot, tracking the distance provided by the sensor that we intended to be parallel to the nearest wall to ensure the turn has been completed. Combined with a 3-axis accelerometer that we employ as an onboard compass, the robot should more accurately and precisely move to each of its intended destinations, retrieving and sorting balls along the way. The compass is employed in tandem with the ultrasonic sensors to set both the x-coordinate and y-coordinate in space and then the compass sets the rotational vector so that the robot is always directly facing the ball when it attempts a retrieval. In this same manner, we should be able to direct the robot to each of the corners, and using the south ultrasonic sensor, back up to the corner pocket to deposit the ball.

Figure 10: Two of the ultrasonic sensors that make up the ultrasonic sensor array

**5.4 Retrieval Subsystem**

The Retrieval Subsystem uses a pincer that hinges in two places to fit within the robot’s 8”x12”x10” space requirement. The base design of the pincer was a previously published STL model found on the open source website Thingiverse (Figure 11). This design was chosen because of its robustness and ingenuity. Likewise, its design was beyond what we felt our capabilities were using out 3D modelling skills in Creo, easily outperforming anything we could create from scratch. We then modified the design found on Thingiverse, giving the pincer arm a base that could pivot on the x/z-axis. Likewise we added a point of rotation further up the pincer arm, pivoting on the same axis of rotation. This second point was added both to fit the size constraints imposed on the robots as well as to angle the pincer arm directly above the sorting mechanism, ensuring that the ball is passed precisely from the retrieval system to the sorting system.

Another aspect of the Thingiverse pincer that was modified was the actual pincer claw itself. Instead of using the premade claw, we modelled one in Creo that was exactly 40 millimeters in inner diameter (figure 12), ensuring a tight grip on any ping pong ball that the robot attempt to retrieve. In total, there are four hobby servo motors that are utilized in the described design. Two of the servos control the pivot from the base of the robot, one servo is used to pivot from the top of the pincer arm, and the last servo is used to open and close the pincer claw. With these modifications to an already robust design, we feel confident in the robot’s ability to retrieve a ping pong ball, contingent on the accuracy of the locomotion and navigation subsystems.

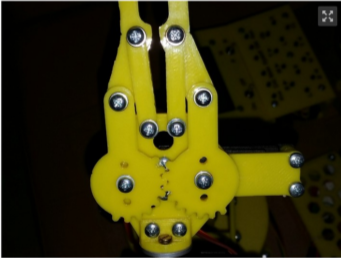


Figure 11: Picture of a 3D printed model of WorkerBot’s retrieval mechanism (pincer)

  
Figure 12: The modified Retrieval Subsystem pincer

**5.5 Sorting Subsystem**

The sorting methodology of sort and separate was chosen to maximize speed and accuracy of our sorting. By sorting each ball and storing them in separate containers we save time by depositing them all at once rather than individually. The storage system for the Sorting Subsystem is an open-air design that cleverly mimics the human digestive system as a way to save space (Figure 13). The inspiration behind this design was deceptive length of the human intestines, crammed into a tiny space. The way the intestines wind around each other was used to create a space that could store more ping pong balls than possible before in a straight space. The winding tubes that the balls file into are just large enough to store seven balls. Before entering the storage area, the ball is dropped into a pivoting cup (Figure 14). The TCS3200 frequency sensor is mounted laterally on the pivoting cup’s stand to determine the color of the ball prior to the cup’s pivot.

After a ball has been dropped into the pivoting cup, the TCS3200 frequency sensor determines the color of the ball based on RGB values and will pivot, using a servo motor to either the left or right. This dumps the ping pong ball into its respective open-air tube. The motor will always pivot to the left on the first dump, and all values after are compared to this first value. Similar values are dumped left, while differing values cause the pivoting cup to turn right. The software will never allow more than seven balls to be dumped into the same storage area, preventing overflow. A complete construction of the sorting system can be seen in figure 15.



Figure 13: 3D model of the holding area for the ping pong balls.

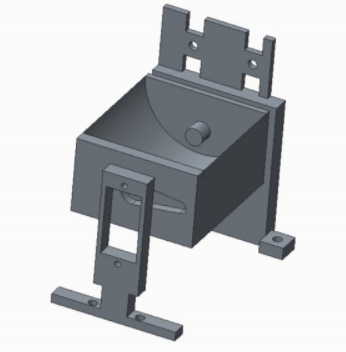
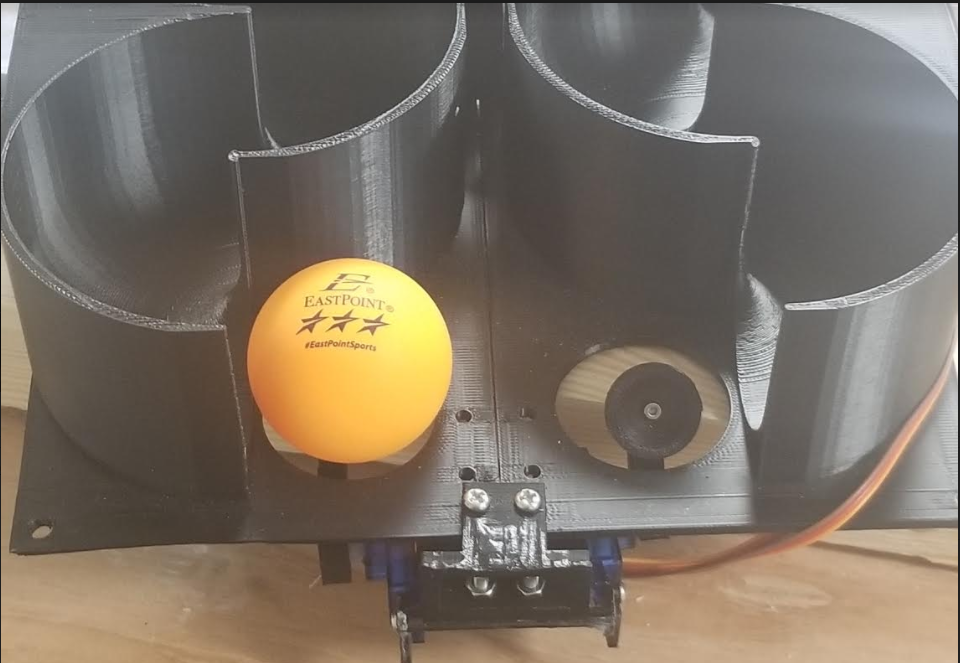


Figure 14: 3D model of the pivoting cup attached to its stand. TCS3200 to be mounted laterally with screws.

  
Figure 15: Overhead view of the assembled Sorting Subsystem

**5.6 Deposit Subsystem**

The Deposit Subsystem will be using a single-ball dump system. After filing through the open-air intestine-like storage, the ping pong balls come to rest in a hole that is 35 millimeters in diameter (figure 16), so the ping pong ball can rest in it but not fall through. Under each of these holes are a servo motor connected to a 3D printed pushing mechanism. When the deposit is triggered, the servo rotates, pushing the balls off the robot and into their proper receptacle on the competition board. At first we were concerned with the balls bouncing out of the receptacles, but by varying the signal sent to the servo, we can hit the ping pong balls at a shallow angle so they land in the pocket with more horizontal velocity than vertical, ideally preventing bounce. Likewise, the Retrieval Subsystem can be used as a lever (figure 17), pressing down against the competition board to lift the robot slightly, angling the storage area so the balls flow neatly downward toward the Deposit Subsystem. This action is similar to the way a dump truck angles its bucket to pour its payload out.

  
Figure 16: The Deposit Subsystem, prepared to dump a ping pong ball

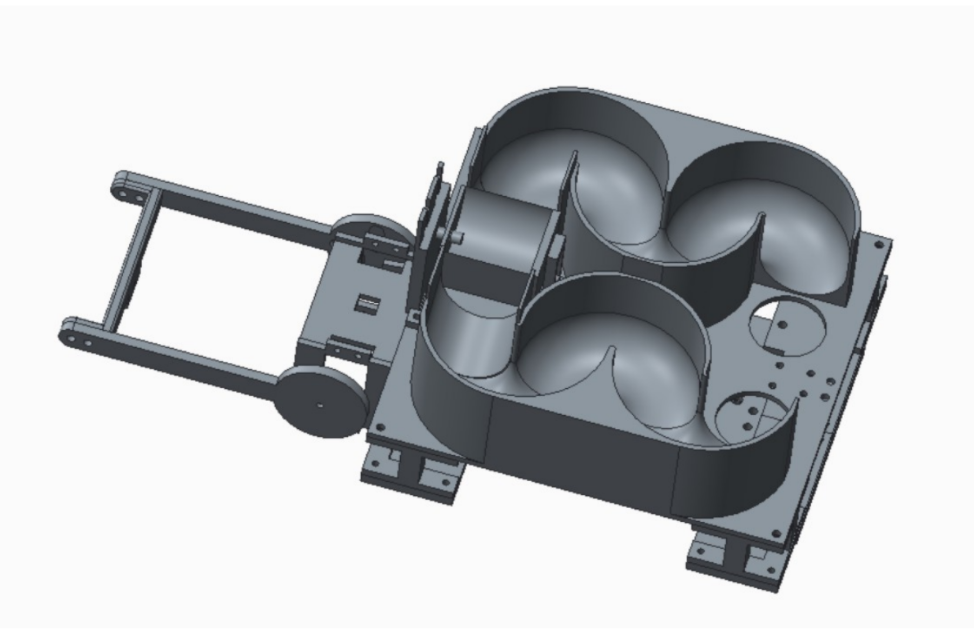


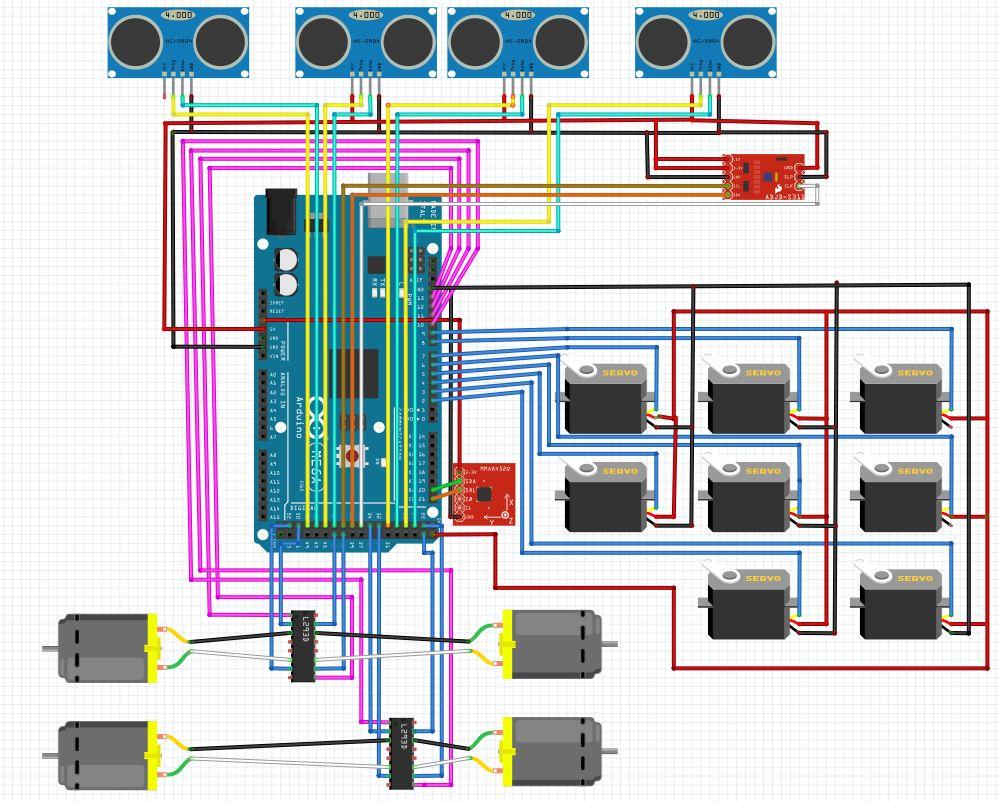
Figure 17: 3D model of the full build for our WorkerBot, displaying the Deposit Subsystem in relation to the Retrieval Subsystem, which can be used as a lever

**5.7 Software**

The UML software diagram describes the model of how the different modules will interact with each other in code (Figure 18). The main logic will function as the main driver program for all the separate modules. Within this module it will have methods that will call the other sub-modules and have their functionality implemented. The robot needs are mostly linear in nature, so it will follow step by step to traverse the board, and as it gets to points of interest a set of events will occur sequentially. The main logic will call the drive method which will use a Drive object to control the motors and determine direction and speed, allowing for complementary functions such as correctYaw() and correctDirection() to make adjustments to the robots placement on the board as it travels. When it reaches the point of interest such as retrieving a ball, the retrieveBall() method will be used to call the instance of the Scooper that will implement its series of actions that will enact the servos to close the door on the scooper and retrieve the ball, then activate the vertical servo to deposit the ball into the sorting mechanism. This will return a value to let the main logic know it can continue on to the next module which is the series of functions that use the Sort object and chooses which deposit bin to send the ball based on its color. Once the main logic has completed that process 13 times and traversed the entire board it will then call the Deposit object to deposit the balls in the correct receptacle.

****Figure 18: UML Diagram for WorkerBot’s software component

**5.8 Hardware Schematic**

****Figure 19: Hardware Schematic

**5.9 Bill of Materials**

The bill of materials is listed below. Not included in the bill of materials are small things like the PLA plastic used in 3D printing, jumper wires, solder, and heat shrink insulation. None of the above would vary the cost of our robot by more than a few dollars, and considering we only spent roughly half of our budgeted $400, we stayed well within our range.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Team: | Group 12- ASEE Robotics Competition | Date: | 3/23/2018 |  |  |  |  |
| Sponsor: | Michael Ira Golub | Project: | Worker Bot | | |  |  |
|  |  |  |  |  |  |  |  |
| **Item** | **Description** | **Manufacturer** | **Part number** | **Qty** | **Unit Price** | **Ext. Price** | **Notes** |
| 1 | Omni-Directional Wheels | Nexus Robot | RB-Nex-57 | 4 | $ 12.00 | $ 48.00 | Locomotion |
| 2 | Color Sensor | WaveShare | TCS3200 | 1 | $ 5.89 | $ 5.89 | Sorting |
| 3 | Arduino Mega | Arduino | 2650 | 1 | $ 38.50 | $ 38.50 | Controller |
| 4 | Ultrasonic Distance Sensors | Banana Robotics | HCSR-04 | 4 | $ 3.00 | $ 12.00 | Navigation/error checking |
| 5 | 3-Axis Accelerometer | InvenSense | MPU-6050 | 1 | $ 13.99 | $ 13.99 | Error Checking |
| 6 | Servo Motors | Tower Pro | MG90S | 3 | $ 3.59 | $ 10.77 | Sorting/Depositing |
| 7 | Servo Motors | Tower Pro | SG90 | 3 | $ 6.00 | $ 18.00 | Pivoting Pincer |
| 8 | Parallax Servos | Parallax | 900-00008 | 2 | $ 13.00 | $ 26.00 | Moving pincer up and down |
| 9 | Motor Driver | Texas Instruments | L293NE | 4 | $ 3.68 | $ 14.72 | Logical control for motors |
| 10 | 210RPM Encoder Motors 6V | Uxcell | Hobby Type TT | 4 | $ 19.99 | $ 79.96 | Turns the wheels |
| 11 | Batteries (7.4V rechargeable) | Hoisin | 18650 | 1 | $ 8.82 | $ 8.82 | Powering the motors |
| 12 | Batteries (9V) | Duracell | MN1604 | 1 | $ 2.54 | $ 2.54 | Powers Arduino |
| **Total** |  |  |  |  |  | **$ 278.19** |  |

**5.10 List of Standards**

No standards are currently being implemented in this project, though IEEE 802.15.1 Bluetooth is something we are considering implementing in the future. This will allow us to read values from the Arduino wirelessly, making debugging easier and allowing us to read the specific values of the ultrasonic sensors and rotary encoders during operation.

**6. Implementation**

Integrating all of the subsystems of our robot has been challenging, but very rewarding. The implementation of our design started with a very simple process flow diagram (figure 20) to govern the movement of our robot.

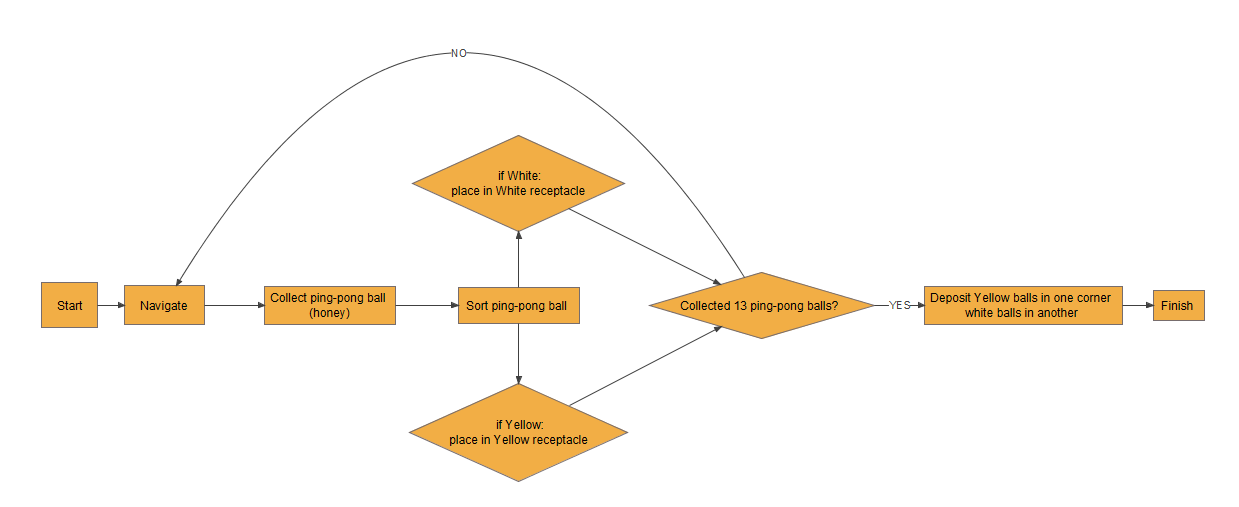
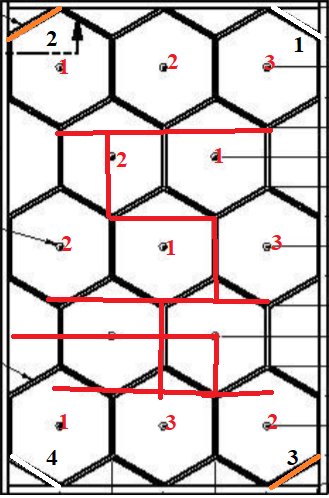


Figure 20: Process Flow Diagram for the ASEE Robotics Competition WorkerBot

From this process flow diagram we extrapolated to determine a path that the robot will take on the competition board. Many paths were considered, including shortest paths from ball to ball, least amount of turns, paths ending closest to a corner, but we finally settled on a snake-like path that made use of long, straight drives forward. This path can be viewed in Figure 21. The red line shows the path from the wall that the robot will take. The robot pick up the second row of balls first, followed by the first row, third row, fourth and fifth. The order of balls picked up per row is numbered in red. The corners are numbered (and colored by pocket color) to show what order the robot will drop its payload. The robot will run along the wall to accomplish this. The reason we chose this path is due to its navigation relying heavily on the ultrasonic sensor array. The ultrasonic sensors read flat surfaces much better than angled or rounded surfaces due to the way the sound waves bounce on them. Because of this, we decided it was best for the robot to stay parallel to the wall as much as possible and only deviate while pivoting or depositing balls.

  
Figure 21: Visual representation of the robot’s intended path. The corners are labeled in the order the robot will deposit in.

With the basic algorithm and path decided on, we have begun working on the software to fully accomplish the proposed tasks. While the software is still being worked on and tested, its basic methodology can still be explained. The robot starts at the wall and immediately picks up the first ball. It then moves forward reading the east wall with the north sensor until it is in range. The robot will then attempt to correct its yaw and longitudinal/latitudinal distances. It picks up the ball and then proceeds to the first row of balls. It picks them up, reading various directions and self-correcting its yaw before every retrieval. The robot will continue in this manner, picking up all of the balls in each row, moving from the third to the fourth, to the fifth, and sorting them. The robot will then pivot and back into the wall until the south sensor reads less than three centimeters. At that point we will drive all the motors a predefined time to ensure we are backed up against the corner and deposit part of the white payload. The robot will shimmy to ensure the the balls have moved down the tube into the deposit area. The robot will then follow the wall to the orange #2 pocket and follow the same procedure. It follows the same method for pockets 3 and 4. At this point the robot’s job is complete and it will brake all motors and no longer move.

**7. Testing Procedure**

The majority of testing occurred at the Subsystem level for our ASEE Competition robot. Each of the Subsystems were tested individually, with the exception of the Navigation Subsystem, which required the Locomotion Subsystem in order to function.

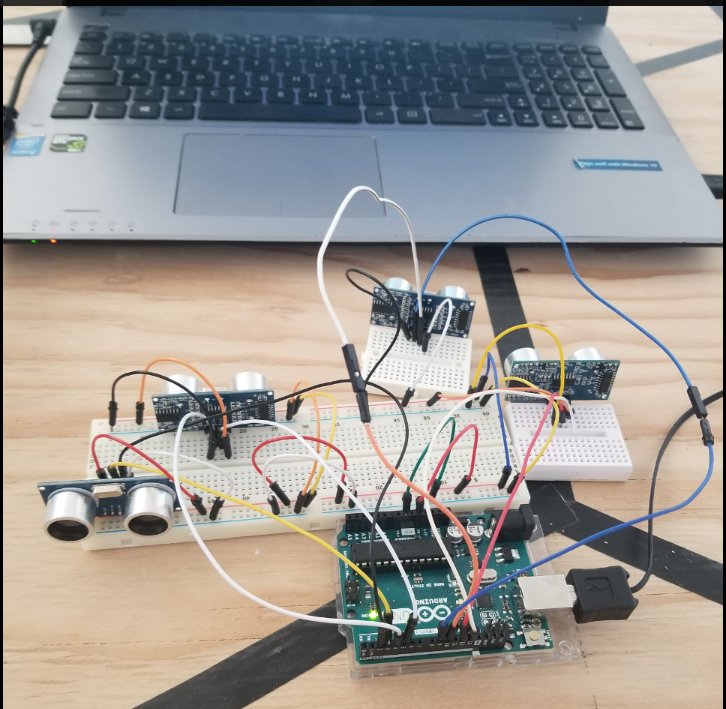
**7.1 Locomotion Testing**

The Locomotion Subsystem was tested by prototyping the motor controller circuit on a breadboard and attaching the DC Motors to them. We wrote specific unit tests to run each motor, ensuring that we were sending the correct pulses to the motor controller’s control and enable pin. After completing this, we assembled the base of the robot with DC motors, omni-directional wheels, and a 3-axis accelerometer. The Locomotion subsystem was tested by running each of the functions that it used such as driving forward, reverse, left, right, pivot clockwise and pivot counter-clockwise, making sure that the motors were moving in the expected direction at all times. The software testing was pretty straightforward; however, the hardware testing was much more time intensive due to components failing and needing to be fixed or replaced and wires needing to be reinstalled, negating any previous testing. A variety of unit tests were written to ensure proper use of the rotary encoders, proper breaking of the wheels, and proper turning, pivoting, and driving of the wheels. The wheels were run in all configurations we considered necessary for our robot, driving forward and backward, left and right, and pivoting both clockwise and counterclockwise. To ensure proper functionality of the rotary encoders were used a timer to debounce the encoders and read their values from the serial monitor in the Arduino IDE. Drawing a black line in sharpie on the wheels, we tracked their rotations and, knowing that 20 pulses of the rotary encoders translated to 360 degrees of rotations, we confirmed their use.

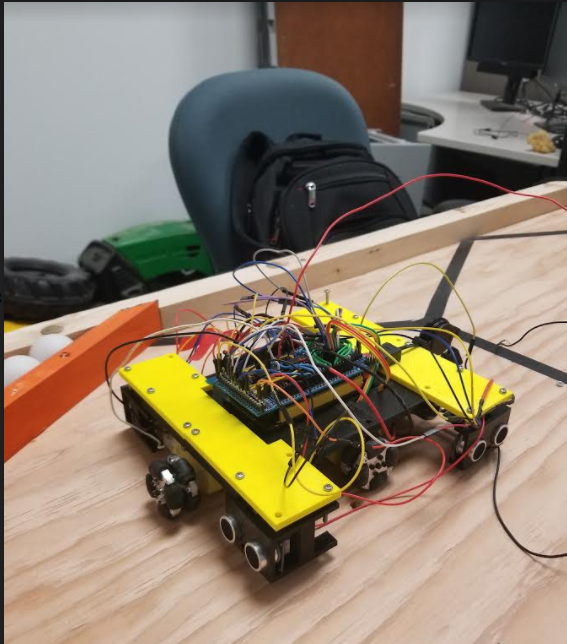
We extended this testing by running the robot on the competition board, using the honeycomb lines drawn in electric tape to see how straight the robot drove. Implementing a feedback system that synchronized the motors, or corrected in the event of errant operation, we measured the robot’s deviation from its intended path.

**7.2 Navigation Testing**

The Navigation subsystem was tested using in conjunction with a functioning locomotion system. Other than what was mentioned above, the ultrasonic sensor array was prototyped (figure 22) using a large solderless breadboard and two smaller solderless breadboards. The distance values from the ultrasonic sensors were sent to the serial monitor in the Arduino IDE and each sensor was tested using a ruler and a flat object to ensure proper operation and proper algorithm for converting ping time to distance in the ultrasonic distance library.

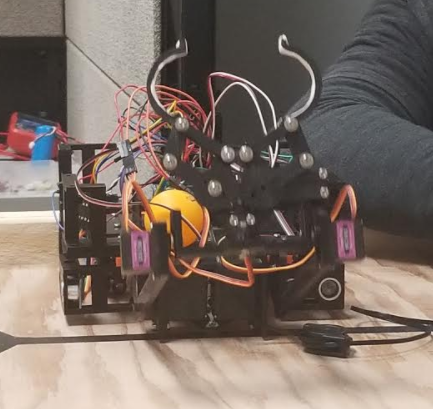
  
Figure 22: Prototype ultrasonic sensor array

Following the prototype sensor array, the ultrasonic sensors were attached to the base of robot and run with the locomotion system (figure 23). We used the ultrasonic sensors to properly induce braking at certain distance as well as change motor directions to correct for motor overshoot. Testing occurred with printing to the serial monitor of a laptop, so the ultrasonic reads could be seen. Once wireless movement with battery power started we needed a way to communicate values over the air. This was done with the temporary installment of an HC-05 bluetooth module that was transmitting data over one of the three serial communication lines in the Arduino. Another Arduino with its own Bluetooth module was printing the values to a monitor on a computer.

  
Figure 23: Testing the Navigation System

**7.3 Retrieval Testing**

The Retrieval Subsystem was tested both as a stand-alone mechanism and attached to the full robot base with wheels (figure 24). The robot has a set of motions between the five servo motors that control the retrieval system to pick up a ball in the most consistent manner. First it opens the gripper, lowers the lever, rotates the gripper towards the ball, closes the gripper, then rotates the gripper up first to ensure the ball has the best chance to not fall out, last it raises the lever for deposit and opens the gripper to drop the ball in the sorter. This particular order was found to be the safest way to transport the ball because it allows gravity the least opportunity to pull the ball out of the gripper. Calibration was needed to make sure the lever lowered to the right angle and to grip the ball at the widest diameter, so that when it closed it didn’t lose control of the ball.

  
Figure 24: Retrieval Subsystem test

**7.4 Sort Testing**

The Sorting Subsystem was tested independently (figure 25) and RGB values were read through the serial monitor in the Arduino IDE. All 13 ball tests were conducted as well as infinite loop tests where balls were placed in the mechanism and compared to the output from the serial monitor. With our current implementation, the sorting mechanism works with 100% accuracy in the light conditions tested.

  
Figure 25: Sorting Subsystem independent test

**7.5 Deposit Testing**

The Deposit Subsystem has received the least amount of testing, but we have observed the deposit system functioning as intended and dropping balls into the appropriate receptacle. Currently we are waiting for a full robot build test to get conclusive results for this apparatus because the robot will have to approach all four corners to deposit the ping pong balls, and the robot’s position at the time of deposit is critical to the success of this subsystem.

The Deposit subsystem required simple tests to properly deposits balls out of the back of the robot by setting the servo angles. The difficulty of the subsystem was in figuring out innovative ways to get the balls in the sorting module to continue to track down the pipe into the deposit position. The first thing we did to achieve this was to set the entire sorting system on a slight pitch so that gravity would help the balls down towards the the exit. The last few balls had difficulty getting all the way down once the weight of the balls behind them was not as great, so we implemented an assistance from the Locomotion subsystem to quickly jerk forward and back into position so that the jostling around would shift the balls down towards the exit pocket. This in addition with a delay on the timing of the deposit servos would allow for the balls to be properly deposited into the pockets.

**7.6 Full Robot Testing**

For full robot testing we began by implementing a single process of moving to a ball, retrieving the ball and sorting it. We did this exhaustively looking for the percentage failure over time. Once we got that down to a consistently acceptable level we then moved on to pivoting to retrieve the next set of balls, and repeating the process for each of those balls. Once this was done we applied the process similarly to the rest of the balls. For navigation to the pockets for deposit we had to spend some time testing values to find the right spots to drive to because the sensors go blind when the robot has to move to the angle to deposit. We could only depend on the south sensor that was facing the pocket as a valid read and had to use the accelerometer to keep track on the angle of motion as we backed into the pocket.

**7.7 Testing Results**

All separate Subsystems function as intended, but can still have slight variations that make integrating everything an arduous and constantly changing and improving process. At the time of demonstrations we were able to traverse the whole board and touch all 13 balls, some would be just so slightly off that we would knock the ball out of the way with the gripper. Some ball positions would be read with a distance that made the gripper just far enough away to not be able to reach the ball and it would stay in its position. Since our robot is blind to the balls on the board it would have no way of knowing that it missed it and correct for it. We need to go back and double check our sensor reads at these positions for the balls. We could also deposit all the balls we collected into the appropriate pocket; however, it took about 5 minutes which was too slow for competition. We would need to look for ways to speed up our robots process of error correcting in order to get our time down.

**7.8 Testing Analysis**

The exhaustive hardware testing that occured while building the robot made it so that by the time demonstrations came we could reliably run our robot without having to worry about hardware failure, but because of the time it took to get the system functioning properly, we had little time to do as extensive software testing and found that we could improve our navigation algorithms. As of now, the robot takes a lot of time settling into the correct position although when it gets there it’s very close to its intended position. In order to fix this the best solutions are probably tied into modifications to the hardware and the sensors such as more consistently accurate ultrasonics that are guaranteed to always read the same values. There was also difficulty in dealing with the physical characteristics of the board we used. The wood was warped in some places and we found areas where this would cause the sensors to get invalid reads and we had to deal with these issues in our code. With the current design that we have gone with we are probably pretty close to maximizing the effectiveness of our design and barring some optimizations to the code for speed any further improvements would probably be made by taking large steps backwards in the design process and making significant changes at early stages by implementing different techniques or processes for retrieving the balls.

**8. Conclusion**

**8.1 ASEE Robotics Competition Conclusion**

Currently, our ASEE Robotics Competition Senior Design has not concluded. We are still in the midst of testing. Even when the class portion of this Senior Design is finished, should we at that time have a team to mentor we will be walking them through successful design processes and teaching them what worked well for us and how they can (hopefully) build a better robot than us. The lesson plans we created can be viewed in their entirety in Appendix B. While creating the lesson plans, we strived for generality, so as not to stifle the mentee student’s creativity and so we could adapt to them and teach them in a way that they would find effective. The lesson cover: an overview of the competition, principles of design, chip selection, common robotics parts, prototyping, and five case studies covering each of our Senior Design robot’s Subsystems. With these ten lesson plans and a hands-on approach, we hope to educate that group of students and instruct them on how to build a functional robot capable of winning the ASEE Robotics Competition in June of 2018.

**8.2 Lessons Learned**

Overall, we were very frugal with how we spent our $400 budget, so much so that we ended up drastically under budget, spending just over $200. Utilizing 3D printing saved us a significant amount of money, but we did not spend our surplus on expensive parts that potentially would have worked more effectively and had less risk of failure. If we had the opportunity to complete this project again, something we would have considered would have been a lidar sensing system or an onboard camera using image recognition. Those kinds of sensors are very expensive but the resolution would have been much better than what we used. In general, we could have purchased much better parts, though the ones we did purchase served their purpose.

Likewise, as well as the omni-directional wheels worked, something with more traction might perform better on the uneven surface of the competition board. Though we sanded the board over and over, the grain of the wood and filled cracks still presents problems for the robot. It would be prudent to test several different wheels on an already assembled competition board before settling on a design, something we were unable to do due to a bottleneck in purchasing the required wood to build the competition board.

**8.3 Future Work**

There is still much that can be done to improve the ASEE Competition robot. While we hope that the group we mentor significantly outperform us, there are several things that they can do to improve performance.

Obviously minimizing the amount of time on the competition board (with a perfect score) will net a higher score in the competition. This can ideally always be improved. More clever algorithms, faster DC motors (if accuracy can be maintained), more powerful servos (if accuracy can be maintained), and different robot designs are all potential solutions to the faster robot problem; that is why this is a competition. All of these factors (and more) should be kept in mind while improve this project for the future.

**9. Appendices**

**Appendix A: References**

Arduino IDE (Version 1.8.5). (2011).

*http://www.ti.com/lit/ds/symlink/l293.pdf*. (1986). Retrieved from http://www.ti.com/lit/ds/symlink/l293.pdf

*Language Reference*. (2018). *Arduino.cc*. Retrieved 17 October 2017, from https://www.arduino.cc/reference/en/

Rowberg, J. (2011). MPU 6050 Library (Version 2.0).

*Ultrasonic Ranging Module HC - SR04*. (2018). Retrieved from https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf

**Appendix B: Lesson Plans**

**Lesson 1: ASEE Robotics Competition Overview  
Lesson Objectives:** Introduce students to the Robotics Competition, curriculum, and outcomes of this mentorship.

**Hook:** “What do you already know about robotics?”

* Have students introduce themselves and volunteer their prior experience with robotics
* Gauge experience levels (**IMPORTANT** for further lessons, which might require modification if students are highly proficient in robotics, or have no experience at all)

**Topic 1:** The ASEE Robotics Competition

* The competition will be held in **MAY** of **2018** in **SALT LAKE CITY, UTAH**
* The competition is called **WORKERBOT**
* The goal of the competition is to build an autonomous robot that will collect 13 balls (orange and white), placed on a track, sort them and place them into their appropriate corners of the track.
* The robot has 120 seconds to complete its tasks.
* Robots will be scored on a 130+ point basis. 10 points are awarded for each ball placed in the correct corner, 5 points are awarded for each ball incorrectly placed.
* Bonus points are awarded if the robot scores 130 points under the time limit. Bonus points are awarded using the equation: **Points earned = (120 – Time in seconds to complete the run)**
* The robot must meet other constraints as well. It shall be no bigger than 8 x 12 x 10 when it starts. The robot shall not cost over $400. The robot shall not emit any gaseous/liquid/solid emissions. Etc.
* **Full list of rules located at:** <http://faculty.tcc.edu/PGordy/ASEE/ASEE2018/ASEE%20TYCD%202018%20Rules%20%20(6-21-17).pdf>

**Topic 2:** Breakdown of Curriculum

* Week 1: Overview of the ASEE Competition
* Week 2: Design Elements
* Week 3: Chip Selection
* Week 4: Robotic Components
* Week 5: Prototyping
* Week 6: Subsystem 1: Locomotion
* Week 7: Subsystem 2: Navigation
* Week 8: Subsystem 3: Collection
* Week 9: Subsystem 4: Sorting
* Week 10: Subsystem 5: Depositing

**Topic 3:** Goals and Expectations

* The goal of this mentorship is to introduce you (**the mentees**) to robotics and help educate you on how to effectively design, program, prototype, and build a robot using specifications provided to you.
* It will be a fun and rewarding course, but expect to put a lot of work into it. We will have deliverables for you once the project kicks off and we begin designing and prototyping the WorkerBot.

**Lesson 2: Principles of Design  
Lesson Objectives:** Introduce students to different design elements and tools that will help them in the design process.

**Hook:** “Example: You’re all familiar with the robotics competition. How many different ways could the robot propel itself around the track.?”

**Breakout:** Let the students brainstorm and volunteer suggestions for robot locomotion. Possible answers may include:

* Walking
* Driving (4 wheel)
* Driving (3 wheel)
* Omnidirectional driving (4 wheel)
* Omnidirectional driving (3 wheel)

**Write suggestions on the board as they name them.**

**Secondary Hook:** “All of these are good ideas, but how to we choose between them?”

**Topic 1:** Choosing a Design

* Explain each of the following in detail, provide examples:
  + Pugh Matrices
  + House of Quality
  + Pros vs Cons list

**Hook:** “What makes a design good?”

**Topic 2:** Design Elements

* Explain each of the following in detail, emphasize their importance:
  + Cost
  + Robustness
  + Efficiency
  + Accuracy vs Precision
  + Feasibility
  + Reliability

**Topic 3:** Subsystems of the Robot

* Ask the students if they can identify subsystems that would allow our robot to complete its task. **Write them on the board.**
* Case Study: Our WorkerBot
  + Locomotion
  + Navigation
  + Collection
  + Sorting
  + Depositing
* These five subsystems are the most basic elements of a successful WorkerBot, in our opinion.
* Explain how these subsystems fit together to provide a robust, effective design. Show how we chose each element using our Pugh Matrices.

**Lesson 3: Chip Selection  
Lesson Objectives:** Introduce students to different chip possibilities and things to consider when choosing a microchip/controller.

**Hook:** “What programming languages have you all used in the past?”

* Have students volunteer programming experience, write them on the board. Make special note of experience with microcontrollers/Arduino/Raspberry Pi.

**Topic 1:** How to choose a microchip

* Explain importance of each topic:
  + Number of pins
  + Power requirement
  + Processing speed/clock frequency
  + Cost
  + Physical size/space requirement
  + Cost
  + Programming languages supported (Embedded C in microprocessors vs variety in Raspberry Pi, etc)
    - Discuss tradeoffs between learning a new language vs using one they already know
    - Discuss using libraries and open source software.

**Topic 2:** What chips can we choose?

* Review ASEE guidelines, No LEGO Mindstorm, etc.
* Ask students to name possible chips if possible. Probable responses will include:
  + Arduino
  + Raspberry Pi
* Discuss viability of:
  + Arduino
  + Raspberry Pi
  + Different microprocessors like PIC chips
* “Stand on the backs of giants” – Don’t build the system from the ground up if you don’t have to. Use resources available to you, libraries, open source, etc.
* Case Study: Arduino
  + We used an Arduino Mega in our design. Reasoning:
    - Pinout: 70 pins. (Arduino Uno only has 20 pins)
    - Cost: $38.50
    - Programmable using modified C language
    - Plenty of available libraries
    - Can be powered by a 9V battery
    - Relatively small and already has an integrated board attached.

**Lesson 4: Robotic Components  
Lesson Objectives:** Familiarize students with common robotics parts.

**Hook:** “What robotic components can you all name?”

* Write suggestions on the board. Suggestions might include:
  + Servos
  + Stepper motors
  + Sensors (various)
  + LEDs

**Topic 1:** Common Robot Parts

* Explain that these parts we’re about to discuss are not an exhaustive list, they should be encouraged to research additional parts and branch outside their comfort zone.
  + Servos motors vs stepper motors (explain the difference)
  + DC motors (and their relation to servo motors)
  + Infrared sensors
  + Ultrasonic sensors
  + Motor drivers
  + Omnidirectional wheels
  + Encoders (like the optical shaft encoder)
  + Frequency converters (like the TCS3200)
  + Batteries (and voltage regulators)

**Breakout:** “How might we use these different parts to construct different robotic subsystems?”

* Let the class engage in a discussion that encourages ingenuity. Write suggestions on the board. Ideally guide the discussion toward the five subsystems mentioned in previous lesson plan:
  + Locomotion
  + Navigation
  + Collection
  + Sorting
  + Depositing

**Lesson 5: Prototyping  
Lesson Objectives:** Introduce students to prototyping concepts that they will use in the initial design and construction of their prototype robot/subsystems.

**Hook:** “Why do we prototype designs before we begin construction?”

* Write suggestions on the board. Break off if one suggestion is particularly relevant to the construction of their robot.

**Topic 1:** Solderless breadboards vs soldered breadboard

* Solderless breadboards provide an easy way to prototype designs.
  + Wires can be attached and detached easily.
  + Multiple boards can be used if you have different power requirements
  + **BUT** solderless breadboards allow wires to be pulled out easily so we cannot use it for a final design!
* Soldered breadboards
  + Sturdy, robust, but creates a semi-permanent design
  + Should be used only after the design has been prototyped on a solderless breadboard. (Ensure that the design functions before soldering)

**Topic 2:** Soldering

* It is very likely that you will need to solder for this project. Many students will not have any experience, so here is a short tutorial. This is not comprehensive, tutorials can be found online that go into more depth if they need it.
* Materials required:
  + Soldering iron
  + Stand for the iron
  + Solder with a flux core
  + A damp sponge
  + Prototype board
  + Solder braid
* Soldering tutorial overview:
  + Tin the soldering iron
  + Heat the connection for a few seconds before applying solder
  + Apply solder (not too much!)
  + Remove solder, then soldering iron from connection. Let it cool.
  + If you used too much solder, or the connection isn’t good, remove solder by heating the connection again and touching the braid to it. The braid will suck up the solder.
  + Ideally the soldered connection will look shiny. Good job!
  + Clean the tip of the iron on the damp sponge. Re-tin if necessary.
  + Repeat until all components are attached.
  + Unplug iron and let it cool before storage.

**Topic 3:** 3D printing

* 3D printing is a powerful (and cheap) way to create prototype parts as well as final product parts.
* Experiment with CREO or other 3D modelling software to create parts.
* There are several 3D print labs around campus including in the ET and ITCT buildings.
* Many of the parts we made for our robot were 3D printed.
* Show students some of the parts we created and potentially demo CREO if we have enough time/if they have never used 3D modelling software. Any mechanical engineers in the group will likely be better at it than us and will not need a demo.

**Lesson 6: Subsystem: Locomotion  
Lesson Objectives:** Delve into the first, and arguably most important, subsystem of the robot: locomotion.

**Hook:** “What do you think the most important (foundational) subsystem is for our robot?”

* Allow students to discuss and debate this question.
* **ANSWER:** In our opinion, locomotion is the foundation of this robot. Without movement, no other subsystems can function for this competition. Locomotion serves as a literal base for the robot and allows for movement.

**Topic 1:** Locomotion Activity

* Have students brainstorm locomotion possibilities.
* Have students brainstorm their own criteria for successful locomotion, insert helpful criteria like COST, POWER CONSUMPTION, ETC if they struggle to come up with adequate criteria.
* Have students make a Pugh Matrix to determine which mode of locomotion they will use to build the robot.

**Topic 2:** Case Study: Our Locomotion Subsystem

* Give an overview of our design choices and decisions using our Design Document as a reference.
* Give tips for building and prototyping.
* Explain challenges encountered and how they were overcome.
* Features of the Subsystem:
  + Omnidirectional wheels (4)
  + Optical rotary encoder
  + Motor drivers
  + Powered by 9V battery
  + Controlled by Arduino Mega

**Deliverable:** Next week have a Locomotion Subsystem that can move forward and backward 1 meter with a tolerance of <0.5 inches. The robot should also be able to turn itself 90 degrees with a tolerance of <10°.

**Lesson 7: Subsystem: Navigation  
Lesson Objectives:** Delve into the creation of the robot’s navigational system.

**Hook:** “Looking at the competition requirements and structure of the board, what do you all think will be the best way for the robot to navigate?”

* Allow students to discuss and debate this question.
* Write their answers on the board.
* Common answer may include:
  + Memory Mapping (Hardcoded navigation)
  + Infrared sensor navigation (using the black tape)
  + Ultrasonic navigation

**Topic 1:** Navigation Activity

* Have students brainstorm navigation possibilities.
* Encourage creativity, but if some of the above answers were not named, write them on the board and explain them.
* Have students brainstorm their own criteria for successful navigation, insert helpful criteria like COST, FAILURE RATE, SPEED, ACCURACY, ETC if they struggle to come up with adequate criteria.
* Have students make a Pugh Matrix to determine which mode of navigation they will use to build the robot.

**Topic 2:** Case Study: Our Navigation Subsystem

* Give an overview of our design choices and decisions using our Design Document as a reference.
* Give tips for building and prototyping.
* Explain challenges encountered and how they were overcome.
* Features of the Subsystem:
  + Memory Mapped Navigation
  + Sensor backup to ensure accuracy
  + Controlled by Arduino Mega
  + Makes extensive use of the locomotion system’s capabilities.

**Deliverable:** Next week have a Navigation subsystem that can traverse a predefined track. The track can be modified to accommodate their choice, such as a black line made of electrical tape for infrared navigation. The track should be relatively simple, no reason to have them start on competition board to learn the basics of this complex and important subsystem.

**Lesson 8: Subsystem: Collection  
Lesson Objectives:** Delve into the robot’s collection subsystem.

**Hook:** “What would be the most effective way for a robot to pick up ping pong balls?”

* Allow students to discuss and debate this question.
* Write their answers on the board.
* Common answer may include:
  + A crane-like system
  + A claw
  + A scoop or bucket
  + A vacuum
  + A robotic arm

**Topic 1:** Collection Activity

* Have students brainstorm navigation possibilities.
* Encourage creativity, but if some of the above answers were not named, write them on the board and explain them.
* Have students brainstorm their own criteria for successful navigation, insert helpful criteria like COST, SPEED, ACCURACY, POTENTIAL TO DAMAGE BALL, ETC if they struggle to come up with adequate criteria.
* Have students make a Pugh Matrix to determine which mode of collection they will use to build the robot.

**Topic 2:** Case Study: Our Collection Subsystem

* Give an overview of our design choices and decisions using our Design Document as a reference.
  + For Collection, note our change from a scoop to a claw/robotic arm and highlight why these design changes were made.
  + Highlight how space and speed is a HUGE consideration for the method of collection.
* Give tips for building and prototyping.
* Explain challenges encountered and how they were overcome.
* Features of the Subsystem:
  + Robotic arm
  + Padding
  + Several servo motors
  + Powered by 9V battery (5V port on Arduino Mega cannot support more than a few servos)
  + Controlled by Arduino Mega
  + Ultrasonic sensor for ensured accuracy

**Deliverable:** Next week have a prototype robot that can advance 2 meters forward to a ping pong ball. Pick the ping pong ball up without damaging it.

**Lesson 9: Subsystem: Sorting**

**Lesson Objectives:** Delve into the robot’s sorting subsystem and different sorting methodologies.

**Hook:** “How can we sort the ping pong balls in the competition to maximize our points?”

* Allow students to discuss and debate this question.
* Write their answers on the board.
* Common answer may include:
  + Sort ball, then deposit
  + Sort all balls, then deposit
  + Sort all one color (sorting before collection) then deposit
  + Collect all balls, then sort
  + Etc (there are many possibilities and methodologies. Their effectiveness’ vary and are debatable.)

**Topic 1:** Sorting Activity

* Have students brainstorm navigation possibilities.
* Encourage creativity, but if some of the above answers were not named, write them on the board and explain them.
* Have students brainstorm their own criteria for successful navigation, insert helpful criteria like SPEED, POINT POTENTIAL, ERROR TOLERANCE ETC if they struggle to come up with adequate criteria.
* Have students make a Pugh Matrix to determine which mode of sorting they will use to build the robot.

**Topic 2:** Case Study: Our Sorting Subsystem

* Give an overview of our design choices and decisions using our Design Document as a reference.
* Give tips for building and prototyping.
* Explain challenges encountered and how they were overcome.
* Features of the Subsystem:
  + TCS7200 Color Frequency Convertor
  + 3D Printed ramps and pivot
  + Servo motor
  + Powered by 9V battery (5V port on Arduino Mega cannot support more than a few servos)
  + Controlled by Arduino Mega
  + Balls are sorted upon being collected, held in containers to be deposited after all 13 balls are collected.

**Deliverable:** Next week have a sorting subsystem functional. 13 balls (orange and white) will be placed in the sorting subsystem randomly. The system should have some indication of what color the ball is, or deposit into an appropriately labeled container. Ideally all balls are sorted correctly, otherwise changes to the system will have to be made. Subsystem does not have to be mounted to robot (yet).

**Lesson 10: Subsystem: Depositing**

**Lesson Objectives:** Delve into the robot’s deposit subsystem and demonstrate our robot’s operation for the mentees.

**Hook:** “The points your team will score depend on depositing the ping pong balls accurately and correctly. How should we design our deposit subsystem?”

* Allow students to discuss and debate this question.
* Write their answers on the board.
* Common answer may include:
  + Deposit using the collection subsystem (depending on previous design choice)
  + Dump balls into the corners (2 corners)
  + Dump balls into the corners (4 corners)
  + Place a container into the corners (2 corners)
  + Place a container into the corners (4 corners)
  + Etc
* Discuss point potential and the (slight) viability of placing balls in the wrong pockets.
* Discuss speed and the importance of the speed bonus, it will play a large part in their design choice.

**Topic 1:** Deposit Activity

* Have students brainstorm navigation possibilities.
* Encourage creativity, but if some of the above answers were not named, write them on the board and explain them.
* Have students brainstorm their own criteria for successful navigation, insert helpful criteria like SPEED, POINT POTENTIAL, BALL BOUNCE, ETC if they struggle to come up with adequate criteria.
* Have students make a Pugh Matrix to determine which mode of depositing they will use to build the robot.

**Topic 2:** Case Study: Our Deposit Subsystem

* Give an overview of our design choices and decisions using our Design Document as a reference.
* Give tips for building and prototyping.
* Explain challenges encountered and how they were overcome.
* Features of the Subsystem:
  + 3D Printed storage for the balls
  + Servo motors to place containers into corners
  + Powered by 9V battery (5V port on Arduino Mega cannot support more than a few servos)
  + Controlled by Arduino Mega
  + All of the balls are dumped IN the container to minimize time and maximize efficiency.

**Topic 3:** Demonstration

* Give a demonstration of OUR robot, either in person or through video.
* Allow students to poke, prod, and touch the robot in person.
* Answer questions about the robot.
* Give helpful encouragement.
* Thank the students for their diligence and assure them that they’re almost done.

**Deliverable:** The fully functional robot. Give between 1 to 2 weeks for this, more if necessary. This is the homestretch. Refinement can be made between this deliverable date and the competition.