Carmen Diner Robot

Short Circuit - Grilled Cheese 5

Paul Bossley, Jonah Carter, Isaac Clemens, Mia Saurette

ENGR 1282.01H (6602)

Dr. Kathleen Harper

04/21/2020

Executive Summary

Faced with increased demand, Carmen's Diner has commissioned the Ohio State Research and Development team to select from prototypes for a robotic assistant developed by competing companies. This report details the development process for the Short Circuit team and the prototype that the team has produced.

At the time that development was halted due to the spring 2020 COVID-19 outbreak, the prototype included hardware capable of addressing each of the client's needs within the diner. A chassis with two driven wheels and two omnidirectional wheels allowed for good maneuverability around the diner, and an arm articulated in both the horizontal and vertical directions provided versatile interaction with elements of the diner. The software for this prototype included functions for navigating the diner using encoders and the Robot Positioning System (RPS) and for positioning the arm. One hardware weakness that would have been addressed if development had continued was instability when driving up the ramp. This could be resolved by adding weight over the driven wheel to increase traction on the ramp. In software, the functions for RPS navigation did not perform reliably, so this would be another area for continued development.

Within the context of COVID-19 interruptions, Short Circuit's prototype has demonstrated proficiency as a hardware and software platform that could continue to be developed, with strengths of mechanical simplicity and modular control functions. This prototype would be a strong candidate for continued development if Carmen's Diner chooses to resume the project in the future.

Table of Contents

List of Figures

List of Tables

1. Introduction

Robotic automation of tasks can save time and money for businesses in a variety of industries. In the food industry, the popularity of online ordering has increased demand for many restaurants. Carmen's Diner, upon opening a new online ordering system called FEHpingo, has been faced with increased demand. The business has commissioned the Ohio State Research and Development (OSURED) team to select a prototype for a robotic Food Establishment Helper from the variety presented by competing companies. As one of those companies, Short Circuit has utilized the scale model of Carmen's Diner provided by OSURED to develop an effective prototype to address the client's needs. This prototype must hold up to rigorous performance standards in a variety of restaurant service tasks because the best design, as selected by OSURED, will be implemented at full scale in an upcoming renovation of Carmen's Diner [1].

Paul Bossley, Jonah Carter, Isaac Clemens, and Mia Saurette comprised the Short Circuit team. The prototype development was completed between January 30th, 2020 and March 6th, 2020, before the documentation was finalized. The remainder of this report follows the team's progress throughout the project. Section 2 outlines the preliminary brainstorming and modeling process. Section 3 discusses the development and testing of the prototype robot and its software. Section 4 details the current status of this prototype, while section 5 suggests improvements that could be made in the future. Finally, section 6 presents a summary of the information presented in this document.

2. Preliminary Concepts

Before constructing the robot, the team worked through a collaborative design process to develop ideas to address the needs of Carmen's Diner. The best of these ideas were selected and synthesized into a physical mockup and a CAD model so that the design could be analyzed before spending time or money on components of the final robot.

2.1 Design Constraints

Short Circuit began developing ideas by detailing the assigned tasks and design constraints for the prototype. These tasks included beginning the run by reading a red light on the welcome mat indicated by Location 1 in Figure 1 on the next page, and returning an empty tray to the trash can (Location 2) or sink (Location 3). The robot would also be indicating to other workers that an order is in progress by sliding the order ticket across the rack (Location 4), flipping a burger using a hot plate and returning the hot plate to the burner (Location 5), and selecting the correct song that a customer has requested on the jukebox by reading a red or blue light and pressing the corresponding button (Location 6). Additionally, the robot received an ice cream order from the RPS (robot positioning system), determining which flavor ice cream to serve, and flipping the correct ice cream lever, returning it after seven seconds (Location 7). Locations 3, 5, and 7 are on the upper level of the diner and Locations 1, 2, and 6 are on the lower level. Location 4 can be reached on both levels. The levels are connected by a ramp [1].

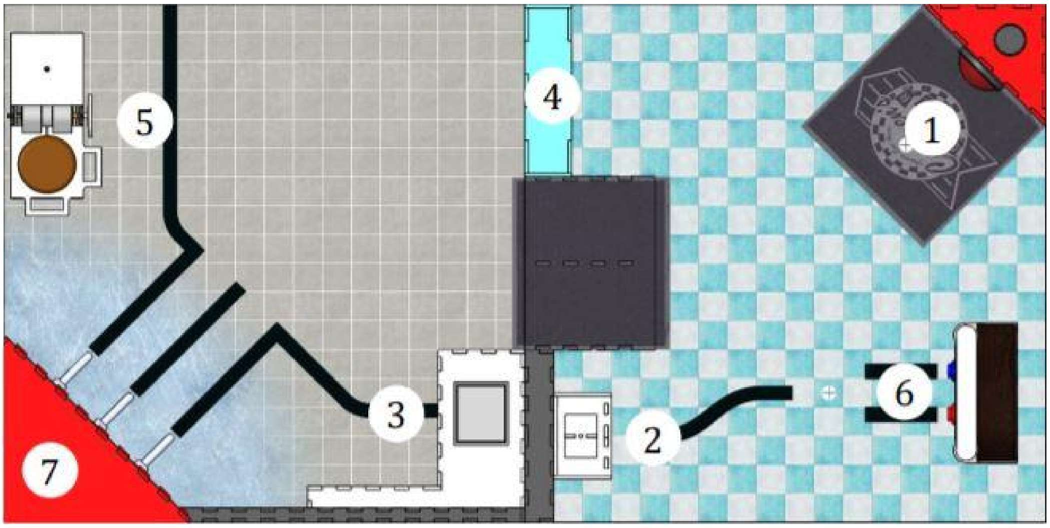


Figure 1: Robot course locations.

To navigate between levels of the diner, the robot needed to have motors strong enough to drive up the ramp. The prototype was constrained to a 9” x 9” footprint with a height 12” or shorter. A QR code was to be mounted 9” above the floor of the diner for communication with the RPS, and the robot was required to be controlled by the provided Proteus microcontroller. The team was restricted to a budget of $160 to spend at the FEH Company Store [1].

2.2 Brainstorming

The brainstorming process began by developing a list of four different ideas for each aspect of the robot and strategy. These ideas were rated using decision matrices to rate their merits in categories determined by the team (Table A1 in appendix A) and the best ideas for each aspect were combined into three different full design concepts. These concepts are included as figures A2, A3, and A4 in appendix A and were also assessed using decision matrices. The design that was selected used two rubber front wheels and two rear omni wheels. The chassis would be made with aluminum sheet metal, bent for wheel attachment. An arm with 90-degree up-and-down range and 180-degree side-to-side range would be able to interact with all of the diner elements. IGWAN motors for the drivetrains were chosen for their integrated encoding, and FITEC servo motors for the arm were selected based on their high torque compared to other servos [4]. The order of task completion was also strategically determined, as shown in figure A1.

2.3 Preliminary Mechanical Design

Following the design brainstorming and the selection process, Grilled Cheese 5 collectively settled on a complete initial design concept, stemming from the design sketches in Appendix A. Before purchasing or using physical components, a computer-aided design (CAD) model was created with CAD models of individual robot parts provided by the FEH department in combination with new parts for the chassis and arm of the robot, shown in Figure 2.

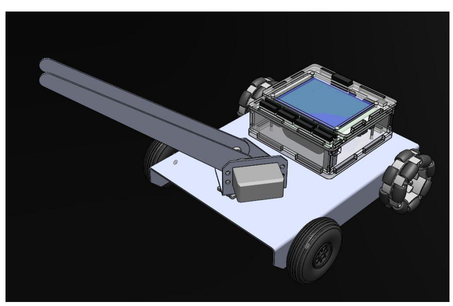


Figure 2: Preliminary CAD model.

The CAD model was used as a basis for a robot mockup, which was assembled to better visualize the robot size and its maneuverability on the course itself. The mockup was constructed out of cardboard and duct tape, providing a solid yet inexpensive physical model to aid the design process and formulate expectations for later robot development. The mockup is pictured in Figure 3.

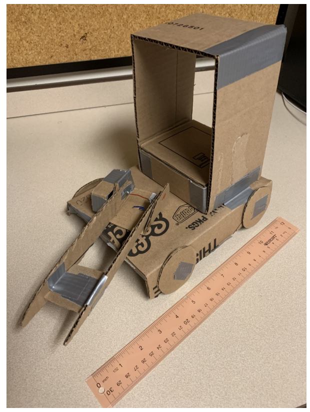


Figure 3: Robot design mock-up.

With the CAD model completed, drivetrain calculations were made to assess the robot's performance on the ramp. First, the minimum wheel RPM was calculated using an estimated travel distance divided by the remaining travel time between tasks. Then, the SolidWorks mass tool was used on the CAD model to calculate an estimated weight of the robot. The angle of elevation of the ramp and the estimated robot weight with a provided average friction force gave an estimated force applied on the robot as it travels up the ramp. The minimum torque required was calculated from the wheel radius and this force [4].

Based on these minimum RPM and torque requirements, a list of acceptable motors was generated. All of the available motors met these minimum requirements, but the deciding factor was the inclusion of encoders on the IGWAN motors, which offered a precise measurement of wheel rotation. Although it would be possible to add encoding to other motors, the IGWAN encoders offered significantly more transitions per revolution than other options [4].

While considering other sensor usage, the characteristics of the robot CAD were taken into account. The robot's front wheel drive layout would limit the placement of an array of opto sensors, making line following more difficult. Therefore, the only additional sensor included in the design was the CdS cell required to detect the color of lights on the course floor. The accuracy of the encoders and the RPS system was considered sufficient to make further navigational tools unnecessary.

2.4 Preliminary Software Design

The preliminary structure of the code included functions to allow for modular control of the robot's drivetrain and arm within the main function. These functions included driving straight a certain distance, turning a certain angle, and positioning the arm with inputs for the angles of its two servos. A detailed description of the drive straight function is included in Appendix C. In addition to these functions, interaction with the RPS was planned as a future improvement to verify the robot's positioning in the diner.

3. Analysis, Testing, and Refinements

Building on preliminary design concepts, the team assembled a functional version of the robot as a platform for further testing. Through multiple performance tests, the robot hardware and software were further refined to better address the client's specifications.

3.1 Movement from Preliminary Concepts to Final Design

The assembly of the mockup design gave the team some insight into how the arm should be built. A tab added between the two strips of metal would allow for the arm robot to flip the ice cream levers more easily as well as add stability to the arm. The team began assembling the robot based on the preliminary concept: aluminum chassis, IGWAN motors, a single arm with two servos, and omni wheels. A majority of the assembly was done exactly as the initial design had depicted. However, in order to begin testing, some items were initially left off, such as the second strip of metal on the arm, the RPS QR code, and its holder. The initial build of the robot contained the basic chassis and drivetrain. Later on, after some drivetrain testing, the arm and QR code holder were built. After each of these construction steps the team was able and ready to test.

3.2 Analysis and Calculations

After assembling the robot, measurements of physical components were used to refine initial hardware and software concepts. One simple analysis step was checking that the chassis could clear the ramp, particularly after the addition of the CdS cell that was not represented in the initial CAD model. Another significant analysis of the drivetrain was verifying that the motors had enough torque to climb the ramp and enough speed to complete tasks within the time limit. The multiple calculation steps to check for torque are included in Appendix B. The speed requirement of the motors was determined using estimates of distance and time parameters and Equation 1 below. Based on these torque and speed requirements, the Igwan motors were an appropriate choice for the drivetrain [4]

$$RPM = 60 \cdot \frac{\frac{inches}{seconds}}{2\pi \cdot radius} = 22.9 \tag{1}$$

The most significant calculation used in software was a more accurate constant for relating encoder counts to distance traveled. Although a theoretical constant was used in the initial version of the code, the physical dimensions of the wheels did not perfectly match the dimensions of the provided CAD model so practical measurements provided a more accurate starting point for testing.

3.3 Testing

Individual parts of the robot were tested as their assembly was completed. The drivetrain was completed and tested first. Straight line motion was tested first by comparing the theoretical distance of encoder-counted motion to the actual motion of the robot. Next, turning was tested by running the robot in a large circle and comparing its observed performance to the expected radius.

The next part of the robot completed and tested was the color sensor. The CdS cell was covered in a red filter. By testing different colors, it was determined that a red filter would give the largest difference in voltage between the red and blue lights. This component was tested by placing the robot in the starting position on the course and triggering the starting light. Later testing with the jukebox lights confirmed that the sensor could reliably differentiate between red and blue light.

The final completed component of the robot was the arm. After calibration, test routines moved the arm through tasks such as sliding the ticket on the course. The arm’s load bearing capacity was also tested by hooking the tray on the end of the arm to apply a load.

After individual systems were tested, the design concept as a whole was tested in multiple scheduled performance tests. Each of these tests scored the robot's performance in a subset of the tasks from the final competition. The first performance test involved pressing a jukebox button and driving up the ramp. The second test required that the robot deposit the tray and slide the ticket. The third and final test included driving up the ramp and flipping the burger.

3.4 Insight Gained from Tests

The testing of individual components ensured that small issues would be resolved before preparing the entire robot for a performance test. For example, preliminary testing of the drivetrain showed that the original concept for a solid rear axle did not allow the rear wheels independent motion and restricted turning. Testing was also essential to fine-tune the values of constants in software that allowed encoder inputs to accurately represent the robot's motion. For the color sensor and arm, testing did not demonstrate any major issues but did confirm the functionality of these components.

Performance tests were an important way to keep development of the robot on schedule and address problems incrementally. The first two performance tests were completed early because the previous testing of individual systems resulted in reliable performance. The main issue revealed in these tests was that the robot drove inconsistently on the ramp because the position of the proteus put most of its weight over the non-driven rear wheels. This issue became more prominent in performance test three, where the robot had to reliably navigate after climbing the ramp in order to position itself to flip the burger. Performance tests two and three demonstrated that the positioning of the arm using servo motors was reliable, but depended heavily on the drivetrain positioning the entire robot accurately. This meant that the arm was effective in tasks such as sliding the ticket or delivering the tray, but only when the drivetrain worked reliably. With the addition of the RPS, the robot was able to correct itself enough to get in position for flipping the burger in performance test 3. Some temporary extra weight, in the form of a wrench and some pliers, was added to the front, driven wheels of the robot to also help with the inconsistency of driving up the ramp. A more permanent solution for adding weight would need to be determined later.

3.5 Refinement as a Result of Testing

The most significant refinement during individual component testing was a change in the drivetrain construction. To allow the rear wheels independent rotation, the original square axle was replaced with a round one and the wheels were drilled out to spin on the axle. This change allowed the omni wheels to function correctly and deliver low-friction turning performance. Another design change was the removal of the second strip of metal on the arm. After completing the second performance tests without this second strip, it was determined that it would not be necessary to add. The single strip of metal made the ticket sliding easier, as the arm could fit between the ticket and the wall easily.

The repeated inconsistency of the robot on the ramp during performance tests resulted in a change to the navigational sensors used in software. Several attempts were made to calculate a better conversion from the encoder counts to inches, however the inconsistency was still apparent, which made the encoders unreliable. Therefore when the wheels slipped on the ramp, the team began using the RPS to attempt to better account for the inconsistencies, however this proved to be just about as inaccurate and presented several new issues such as variation in the heading of the robot. Because of this, a bump switch was added to the front of the robot during performance test three. This bump switch allowed the robot to reference its position through contact with walls. By reducing the length of movements that were guided by the encoders alone, the robot was able to reliably position itself to flip the burger.

4. Current Status of the Prototype

Due to the spring 2020 outbreak of COVID-19, development of the robot stopped following the third performance test. This section details the current status of the prototype and the team's projections of how the robot might have performed in competition if this interruption to the project had not occured.

4.1 Progress Toward Success

The current prototype is highly functional and can complete many of the tasks on the course, including initiating movement by the starting light, pressing the correct jukebox button, depositing the tray into the sink, sliding the ticket, navigating up the ramp, and flipping the burger and returning the hot plate to its initial position. The team successfully completed three performance tests as the prototype was developed. The first featured the aluminum chassis and wheels with a temporary tab mounted on the front of the robot to press the jukebox button. The second and third tests required the attachment of the robot's arm, which is single-pronged rather than double-pronged as initially designed. This arm was used to deposit the tray and flip the burger.

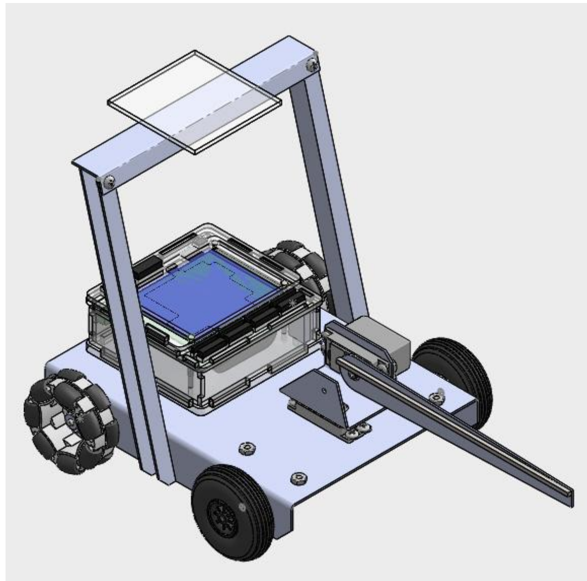


Figure 4: Final robot CAD.

The code is a solid foundation for the rest of the tasks and includes many general functions. It currently includes segments to start moving based on light, press the correct jukebox button, deposit the tray in the sink, slide the ticket, flip the burger and return the hot plate, and move up and down the ramp. Testing would still be required to piece these segments into competition code, but there is currently a strong baseline.

The strengths of the current robot prototype lie in its simplicity and sturdiness. The arm completes all of the required tasks while the aluminum chassis offers stability and requires little maintenance. The weaknesses of the prototype include its inconsistencies in navigation and heavy reliance on its few mechanisms. As a result of the front rubber wheels slipping, most noticeably on the ramp, the robot can move unpredictably about the course, making it difficult to finalize code. Rubber bands are used to minimize the slipping as much as possible. Although having a single arm is simplistic and efficient, any malfunctions would result in the robot's inability to complete every task on the course.

4.2 Resources

The team has spent a collective ninety hours working on this project, approximately twenty of which were dedicated to testing this prototype. The robot was making appropriate progress toward the fourth performance test and competition. Isaac worked primarily on design and code. Mia focused on documentation and project management. Jonah's contributions were mainly in building the robot and keeping the budget and testing logs. Paul concentrated on coding and testing the prototype.

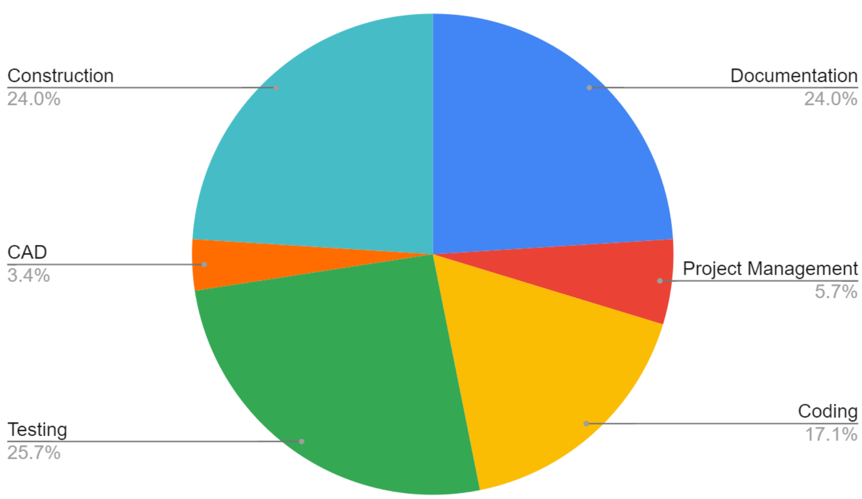


Figure 5: Time allocation.

The team set a goal budget of $120 out of the $160 alloted for the project and is currently a little over this goal having spent $122.42, the large majority of which was committed to the drivetrain. This budget was on track for the final product.

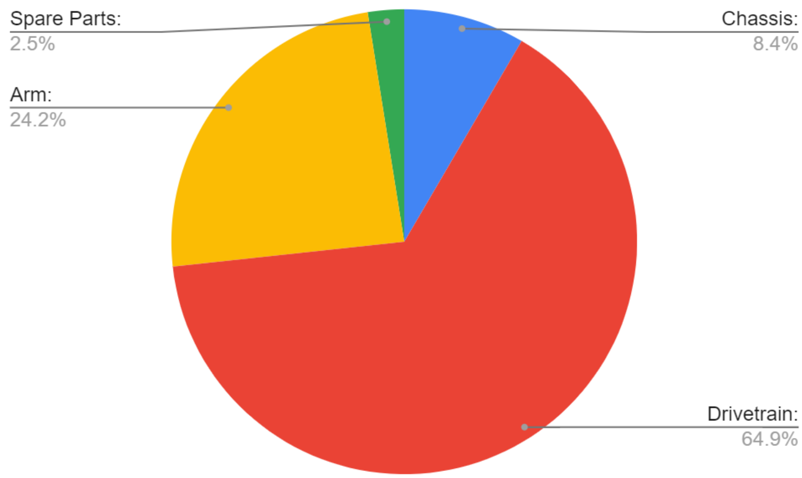


Figure 6: Spending breakdown.

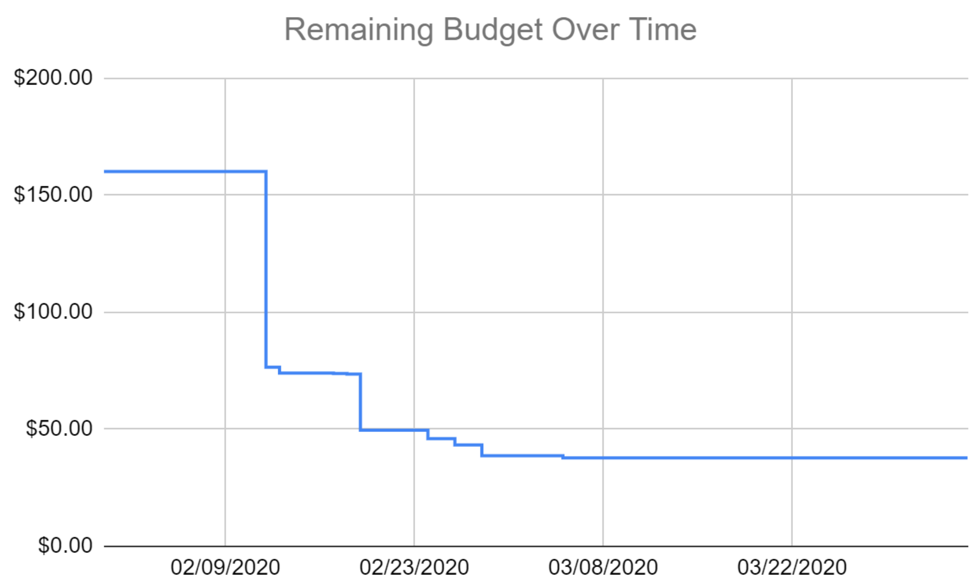


Figure 7: Remaining budget over time.

The resources necessary to complete this project would be access to the FEH Robot Store for any final or replacement parts the robot might need, access to the courses to conduct test runs to finalize code, consultation with the FEH Robot teaching staff to ensure the robot meets all requirements, and in-person team meetings to finish building the prototype and participate in competition.

4.3 Projected Strengths, Weakness, and Challenges

The team anticipates that combining the completion of all course tasks will be a challenge. Handling tasks in small groups has minimized the effect of navigational errors in test runs and performance tests, but completing the entire course in a single run may require a lot of trial and error due to the small inconsistencies that have built up. The robot will remain sturdy and simple, which will continue to be strengths in the team's design. The issue of wheel slipping would be remedied with new rubber bands when necessary.

5. Future Development

While COVID-19 has halted the project for the foreseeable future, the team's original design objectives included aspects of the client's needs that have not yet been tested. If the team were given the opportunity to resume development, these ideas could be implemented to further develop the robot's performance.

5.1. Resources Required for Completion

The previous section of this report mentioned that the chosen design of the robot was simple, utilizing a single arm to complete all of the necessary tasks. That simplicity has allowed the team to have already completed the planned construction of the robot prototype, significantly reducing the amount of resources necessary to complete this project. As a result, no additional budget allocation is planned other than occasionally replacing the rubber bands around the wheels whenever they lose traction. In terms of time and personnel required, the only area in which the team would be working on the robot is the actual programming and testing of the robot to complete all the necessary tasks at once. It is very likely that the amount of testing that would undergo would double the current number of testing hours; however, the amount of personnel that testing requires could easily be reduced to 2 people, allowing the hours to be distributed more efficiently among the team.

5.2. Plans For Completion

Since the construction of the robot is already completed, the future plans are concentrated on preparation for the competition. The anticipated challenges include the robot compounding inaccuracies in movement and the slipping of the tires, which can be reduced with new rubber bands. Something that the team had previously planned on testing was the accuracy of the simple drive functions and turn functions over extended distances so that some code alterations could be made to improve overall accuracy of the functions themselves; this would help reduce the amount of potential error after each function called in the code. Another potential solution to the error was an RPS-based movement function, which was started before break began. Due to the restrictions placed on the robot program, the function was never fully completed, but it would help account for the inaccuracy by using the consistency of the RPS as a guide for movement. The large majority of time remaining, however, would be spent creating and refining the algorithm that the robot would execute on the course to complete each task successively.

6. Summary and Conclusion

The team designed a robot prototype that emphasized great simplicity and efficiency, utilizing a single arm to accomplish all the necessary tasks. This allowed resources to be focused into the drivetrain of the prototype. Even with the greater attention to the drivetrain, there are clear flaws in the robot's ability to move accurately due to slipping or inaccurate encoding. This problem was encountered many times throughout testing and during performance tests, but these inaccuracies were temporarily accounted for in the tests. The algorithms for the tests were adjusted respectively, but it required application throughout the steps to be most effective. As is, the robot prototype was able to exceed expectations for each performance test, but extra movements and effort were required to account for the compounding errors. This report documents the brainstorming and design processes, the analysis, testing, and refinement phases, the current prototype status, and plans for future development.

The team had planned on finishing an RPS-based movement function to account for the inaccuracies along with possible plans to do a thorough drivetrain testing session to adjust the workings of the drive functions themselves so that the inaccuracies would be permanently reduced. In terms of physical additions, the robot design had been completed, so other than the occasional rubber band replacement to maintain friction on the ramp, no other chassis alterations were planned.

The current state of the robot is ideal for future development as it would be focused on testing the process to complete the tasks required by OSURED. The ample amount of time left for refinement of the entire process would allow for a final result of a robot prototype that could efficiently satisfy the performance standards set by OSURED. Beyond designing a prototype for

Carmen’s Diner, the team hopes to work toward the implementation of robotics for the improvement of customer service across the industry.

References

[1] Robot Scenario. 2020, February 23rd. www.carmen.osu.edu.

[2] FEH Robot Store. 2020, February 23rd. feh.osu.edu/store.

[3] Pre R09 - Final Reports. 2020, February 23rd. www.carmen.osu.edu.

[4] FEH Motors Graph. 2020, March 24th. www.carmen.osu.edu.

Appendix A

Brainstorming and Decision Matrices

R03

Grilled Cheese 5

KAH 8:00

2/4/2020

Chassis

1. Modular chassis using erector set rails for versatile bolt holes.
2. Fabricated chassis with bolt holes.
3. Proteus mounted low, with higher frame to mount other mechanisms.
4. Inclined surface for sliding tray downwards and mounting other mechanisms higher.

Drivetrain

1. Triangular omni wheel layout (satisfies wheel restriction).
2. Treads to avoid slipping on ramp.
3. One normal wheel with two omni wheels.
4. Four regular wheels.

Ticket Sliding

1. Extending an arm beside the ticket and driving the robot.
2. Two arms extend to hold both sides of the ticket.
3. Extending arm slides on the robot so that the robot remains stationary.
4. Press rubber wheel against ticket and spin wheel to slide ticket.

Jukebox Buttons

1. Fixed prong on chassis.
2. Prong extends straight out from robot.

3. Multi-purpose arm hits button.
4. Robot chassis hits button directly.

Ice Cream Lever

1. Flipping paddle mounted at correct height on chassis.
2. Arm rotates up and down to bump the lever.
3. Arm with cupped end grabs the lever.
4. Fixed inclined plane on chassis pushed switch with no additional motors.

Tray

1. Tray slides down an inclined plane into sink.
2. Tray starts resting on an arm and slides off into the sink.
3. Magnets hold tray by its base and the sink scrapes it off.
4. Non-motorized arm flips down under deceleration to deliver the tray.

Burger Flip

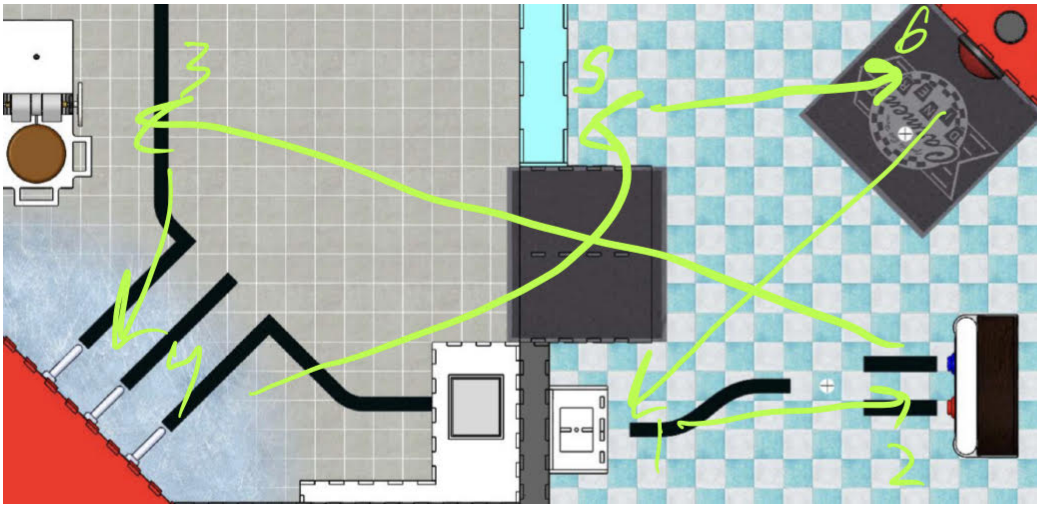
1. Rubber wheel interfaces with wheel on hot plate to flip burger.
2. Arm lifts straight up on hotplate.
3. Pronged adapter interfaces with spokes in hot plate wheel.
4. Crane lifts hot plate from above.

Final Button Press

1. Drive into the button.
2. Push the button with an arm.
3. Launch something at the button from a distance.
4. Linear extending prong.

Strategy

1.



2.

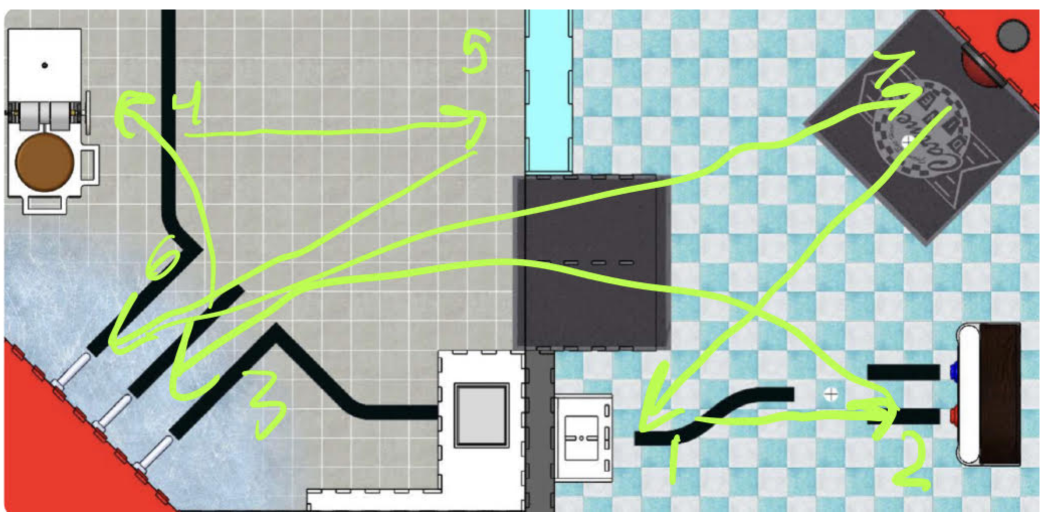


Figure A1: Task Order Options

Rank: Strategy 2 was decided as the better of the two strategies, based off of the efficiency and difficulty of maneuvering around the course. The team decided that it would be more efficient to flip one of the ice cream levers down then move to flip the burger and move the ticket before returning to flip the lever back up. The maneuverability around this option would not be too difficult, as there are a lot of straight line paths for the robot to move on rather than moving at an angle along the course as the first strategy would give. Depending on what the team decides to make the robot design to be, this strategy may change, based on the price of parts, ease of programming, and the sturdiness of the design.

Table A1: Decision Matrices

Drivetrain							
Criteria:	Torque	Speed	Traction	Low Price	Maneuverability	Lightweight	Weighted Score
Criteria Weight:	3	2	2	2	3	2	
Triangular omni wheel layout	1	2	1	1	3	1	22
Treads to avoid slipping on ramp	3	2	3	2	1	2	30
One normal wheel with two omni wheels	2	3	2	1	2	3	30
Four regular wheels	3	3	3	3	1	2	34

Ticket Sliding						
Criteria:	Sturdiness	Lightweight	Low Price	Versatility	Space Efficiency	Weighted Score
Criteria Weight:	2	1	2	3	1	
Extending an arm beside the ticket and driving the robot	2	3	2	1	3	17
Two arms extend to hold both sides of the ticket	3	2	1	1	2	15
Extending arm slides on the robot so that the robot remains stationary	2	1	1	3	2	18
Press rubber wheel against ticket and spin wheel to slide ticket	1	3	3	2	3	20

Juke Box Buttons						
Criteria:	Sturdiness	Lightweight	Low Price	Versatility	Space Efficiency	Weighted Score
Criteria Weight:	2	1	2	3	1	
Fixed prongs on chassis	3	3	3	1	3	21
Prong extends straight out from robot	2	2	2	2	2	18
Multi-purpose arm hits button	2	1	1	3	2	18

Robot chassis hits button directly	3	3	3	1	3	21

Ice Cream Lever						
Criteria:	Sturdiness	Lightweight	Low Price	Versatility	Space Efficiency	Weighted Score
Criteria Weight:	2	1	2	3	1	
Flipping paddle mounted at correct height on chassis	2	2	2	1	2	15
Arm rotates up and down to bump the lever	1	1	1	3	3	17
Arm with cupped end grabs the lever	1	1	1	2	3	14
Fixed inclined plane on chassis pushed switch with no additional motors	3	3	3	0	1	16

Tray						
Criteria:	Tray Security	Lightweight	Low Price	Versatility	Space Efficiency	Weighted Score
Criteria Weight:	2	1	2	3	1	
Tray Rests on Arm	1	2	2	3	2	19
Magnet holds tray and use sink to detach	3	3	2	0	3	16
Lowered on non-motorized arm	2	2	2	0	1	11
Inclined surface for sliding tray.	1	3	3	1	2	16

Burger Flip						
Criteria:	Sturdiness	Lightweight	Low Price	Versatility	Space Efficiency	Weighted Score
Criteria Weight:	2	1	2	3	1	
Rubber wheel interfaces with wheel on hot plate to flip burger	3	2	2	1	2	17
Arm lifts straight up on hotplate	2	2	1	3	2	19
Pronged adapter interfaces with spokes in hot plate wheel	3	2	1	1	2	15

Crane lifts hot plate from above	1	1	1	2	2	13

Chassis						
Criteria:	Sturdiness	Lightweight	Low Price	Versatility	Space Efficiency	Weighted Score
Criteria Weight:	2	1	2	3	1	
Modular with erector set rails.	1	3	1	3	2	18
Custom fabricated with bolt holes.	3	2	3	3	2	25
Higher frame over proteus for mounting systems.	3	2	2	1	3	18
Inclined surface for sliding tray.	3	2	3	2	2	22

Final Button Press						
Criteria:	Sturdiness	Lightweight	Low Price	Versatility	Space Efficiency	Weighted Score
Criteria Weight:	2	1	2	3	1	
Drive into the button	3	3	3	1	3	21
Push the button with an arm	2	2	1	3	2	19
Launch something at the button from a distance	1	1	1	0	1	6
Linear extending prong	2	2	2	1	2	15

Overall Design						
Criteria:	turdiness	ightweight	ow Price	Versatility	Space Efficiency	Weighted Score
Criteria Weight:				3	1	
custom chassis, 4 regular wheels, articulated arm slides ticket, chassis hits jukebox buttons, arm rotates up and down to hit icecream lever, arm delivers tray, arm flips burger, drive into final button				3	2	24
erector chassis, treads, rubber wheel moves ticket, prong hits jukebox, fixed paddle for ice cream lever, magnets deliver tray, rubber wheel rotates burger, drive into final button				3	1	17
incline TRIANGULAR slope chassis, one				2	3	1

regular wheel with 2 omniwheels, 2 prong on end of arm, arm hits jukebox button, arm bumps lever, arm moves tray, pronged adapter interfaces with spokes, drive into final button						6

Three Design Concepts

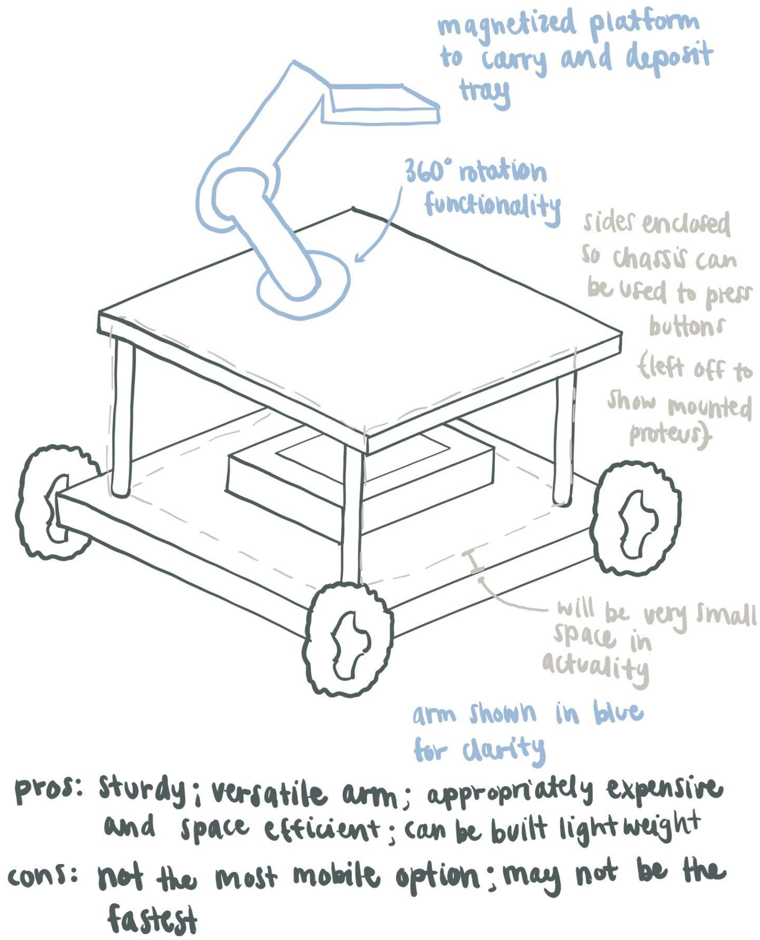
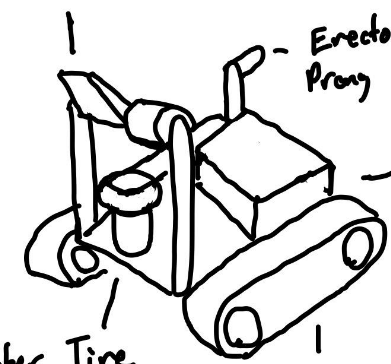


Figure A2: Design Concept 1



Pros

- Specialized devices perform well at each task
- Erector chassis allows for changes

Cons

- High cost and complexity
- Treads decrease maneuverability

Figure A3: Design Concept 2

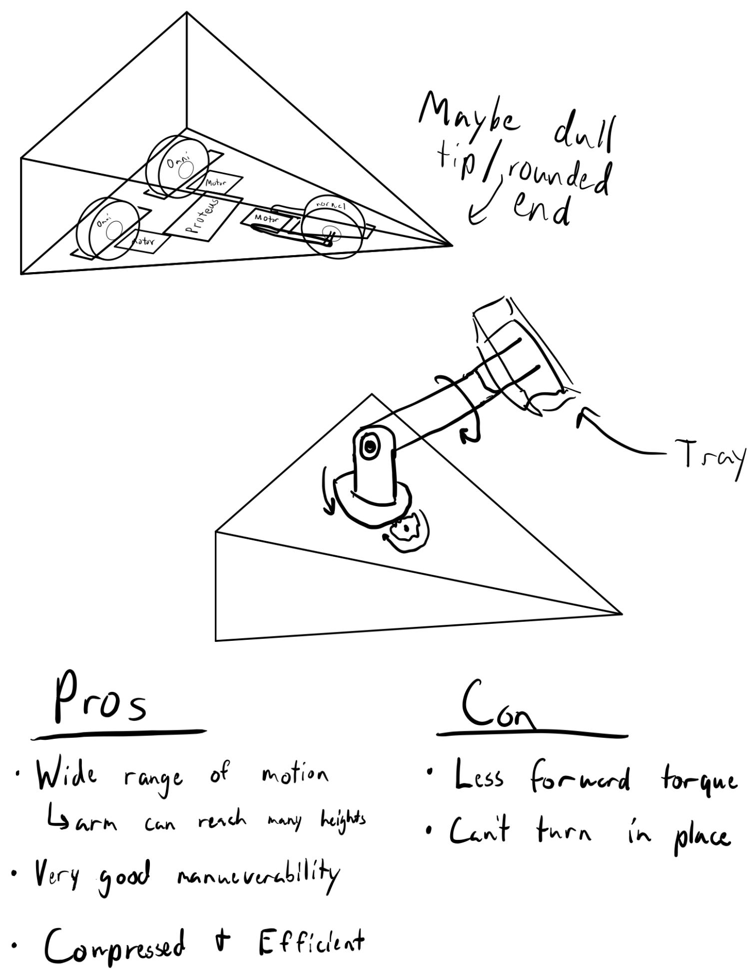


Figure A4: Design Concept 3

Appendix B

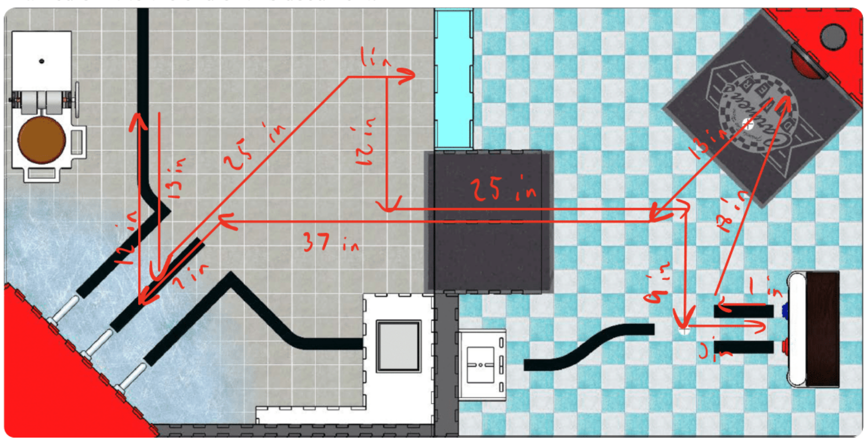
Drivetrain Calculations

This worksheet will guide you through a detailed drivetrain analysis, culminating with the selection of the motor(s) for your robot's drivetrain. Make sure to show your work. If you feel that a figure or table will aid in your explanation or answer to a question, be sure to include it.

Part I: Speed

- Use your course strategy and measurements of the course to determine how many inches your robot will travel on the course. Present the total distance traveled, as well as detail on each segment of the travel. Attach a diagram of the course (available on Carmen) with the path marked on it to the end of this document.

13
37
2
12
13
25
1
12
25
9
3
1
+ 18
171 in
~180 in



- Estimate the total number of seconds it will take you to accomplish each task, not including time for navigating between tasks. Again, present the total time for completing tasks as well as reasoning behind each task time estimate.

Tray = 3 sec
↳ quick motion

Jukebox = 5 sec
↳ quick, time for line up + color sensing

Lever = 13 sec
↳ time to flip up, wait 7, and flip down

Burger = 17 sec
↳ lift burger, move + flip back

Ticket = 12 sec
↳ positioning to fit arm in gap

over-lap of 5 sec

- Consider adding to the total task time some additional seconds as a buffer. Put the resulting amount of task time below.

50 sec + 10 sec buffer = $\boxed{60 \text{ sec}}$

Rev. 01/21/20 KAH

- Determine the minimum linear speed your robot will need if it is to complete all the tasks within 2 minutes. To do this, subtract the total task time from 2 minutes, then divide the total distance the robot needs to drive by that driving time. Present this in inches per second. You may want to round this up slightly to give yourself a bit of margin for error.

$$\frac{\sim 180 \text{ in}}{60 \text{ sec}} = 3 \text{ in/sec}$$

- Determine a probable wheel radius for your robot (in inches).

1.25 in.

- Use the wheel radius to convert the linear speed you calculated above into a rotational speed for your motor. Do this by dividing the linear speed by the circumference of the wheel. Present this speed in revolutions per minute.

$$\frac{3 \text{ in/sec}}{(1.25 \text{ in})\, 2\pi} = 0.38 \text{ rev/sec} \cdot \frac{60 \text{ sec}}{1 \text{ min}} = 22.9 \text{ rpm}$$

Part II: Torque

- List the major components of your robot and estimate the mass of each. Note that SolidWorks can help you with some of this and that a scale is available in the store. Calculate the estimated total mass of your robot and convert it to ounces.

Chassis:	~ 0.16 lbs	= 2.56 oz
Front Drivetrain:	~ 0.68 lbs	= 10.88 oz
Back Wheels:	~ 0.20 lbs	= 3.20 oz
Proteus:	~ 0.65 lbs	= 10.40 oz
Arm Mechanism:	~ 0.26 lbs	= 4.16 oz
	1.95 lbs	= 31.20 oz
Other:	~ 0.25 lbs	= 4.00 oz
		35.20 oz

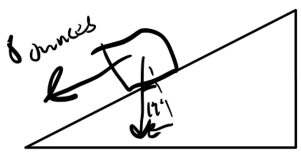
Rev. 01/21/20 KAH

- Measure the vertical and horizontal components of the ramp on the course and determine the angle of the ramp.

$$\theta \text{ of incline} = \arctan\left(\frac{3}{4}\right) \approx 18.4^\circ$$

$$\theta \approx 19^\circ$$

- Determine the force that you will need from the motors to get up the ramp according to the relationship in the class PowerPoint on drivetrain calculations. You may estimate the internal friction at about half a pound (or 8 ounces).



$$\text{Force} = 8\,oz + \text{weight} \cdot \sin(19°)$$

$$\text{Force} \approx 19.1\,oz$$

- Determine the torque required to get your robot up the ramp by multiplying the required force by your wheel radius.

$$\text{Torque} = \text{Force} \cdot \text{Radius} \qquad \text{radius} = 1.25\text{ in}$$

$$\text{Torque} = 19.1\,oz \cdot 1.25\text{ in}$$

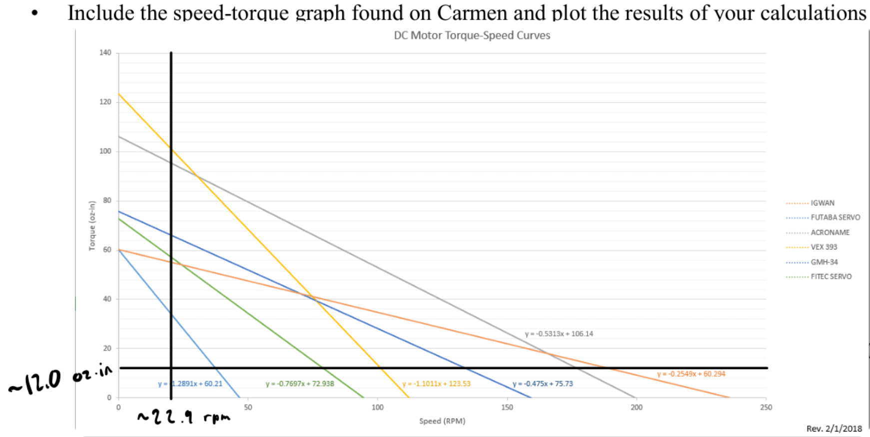
$$\text{Torque} \approx 24.0\text{ oz}\cdot\text{in}$$

- Divide this torque by the number of motors you intend to use to get the torque required for each motor.

$$\frac{\text{Torque}}{\text{motor}} = \frac{24.0\text{ oz}\cdot\text{in}}{2\text{ motors}} = 12\text{ oz}\cdot\text{in}/\text{motor}$$

Part III: Motor Selection

- Include the speed-torque graph found on Carmen and plot the results of your calculations on it.



) KAH

- List any motors that satisfy the speed and torque requirements for your robot.

FUTABA Servo
ACRONAME
VEX-393
GMH-34
FITEC Servo
IGWAN

- If there are multiple motors that meet your robot's requirements, write or list brief descriptions of the criteria that you will use in the selection of your motor.

Torque
Speed
Low Cost
Encoding

- List the motor that you have selected.

IGWAN

Submit electronically per the DAL.

"No one is going to hand me success. I must go out & get it myself. That's why I'm here. To dominate. To conquer. Both the world, and myself." - Unknown

Rev. 01/21/20 KAH

Appendix C

Code Description

Function Detail: Driving Straight

The function drive() controls motor speeds and monitors encoder counts to drive the robot a particular distance in a straight line.

The function declaration includes inputs of inches to travel and motor power, given as a percentage.

```
void drive(float inches, float power){
```

The first lines of the function convert inches to encoder counts using a global constant and then define variables for left and right power so that the two motors can be controlled separately.

```
int counts=inches*COUNTSPERINCH;

float leftPower=power, rightPower=power;
```

Next, the encoder counts are reset to provide a reference point.

```
leftEnc.ResetCounts();

rightEnc.ResetCounts();
```

With this reference point established, a while loop drives the motors until both motors' encoders have reached the correct number of counts.

```
while(leftEnc.Counts()<counts && rightEnc.Counts()<counts){
    leftMotor.SetPercent(leftPower);
    rightMotor.SetPercent(rightPower);
```

Inside the while loop, a series of if statements checks whether both encoders have turned the same number of counts. If not, the motor powers are adjusted slightly to correct the robot back to a straight path. The logic used when power is positive is shown below; the logic for negative power is identical except for the signs of the power adjustment statements.

```
if (power>0){
    if (leftEnc.Counts()<rightEnc.Counts()){
        leftPower+=0.0001;
        rightPower-=0.0001;
    }

    if (rightEnc.Counts()<leftEnc.Counts()){
        rightPower+=0.0001;
        leftPower-=0.0001;
    }
}
```

Once the encoders have reached the target counts and the while loop terminates, both motors are stopped and the function ends.

```
leftMotor.Stop();
rightMotor.Stop();
```

The function returns no output.