ECE 4950 - Project 4
The Big Lebowski: Autonomous Bowling Robot

Team 19

Corey Golec, John Phillips, Jacob Criswell, Avi Patel, Josue Morales-Sanchez

Date: 12/7/2025

Executive Summary

This project involves integrating several elements to create a functional robotic system that determines the location of multiple pins and drops a metal ball to knock them over. The project must be able to demonstrate bowling a strike (knocking over all the pins) as well as a spare (knocking over a subset of pins in random locations). Named 'The Big Lebowski, a complex camera-based (Logitech Brio) image processing system, a MATLAB App GUI, and a MATLAB/Simulink control stack running on an Arduino Mega 2560 were used. Using the provided game board, DC motor, servo, and camera, we created a low-cost (<$300) system with closed-loop position control of a motorized ramp, 3D-printed carriage and guard rails, and a makeshift ball catch that keeps the moving ball and pins safely separated from electrical components. A wood plank was used to hold all the components together, and everything was screwed onto this plank in order to prevent any accidental movement of parts. The GUI which was built using the MATLAB App Designer allows the user to initialize the lane, select individual or automatically chosen pin combinations, and then triggers the control loop, which computes the required ramp angle and commands the motors so that the full cycle from selection to launch and return to idle completes in roughly 10 seconds, with interface response times under 2 seconds. Further testing showed that the final prototype operates reliably, safely, and within the specified performance and cost requirements.

Engineering Requirements

For our final implementation of the robot bowling system, we start by remaking some of the engineering requirements that we came up with at the beginning of our project. With these engineering requirements, we can better understand why and how we will implement certain parts of our system. The engineering requirements, with quantifications, are as follows:

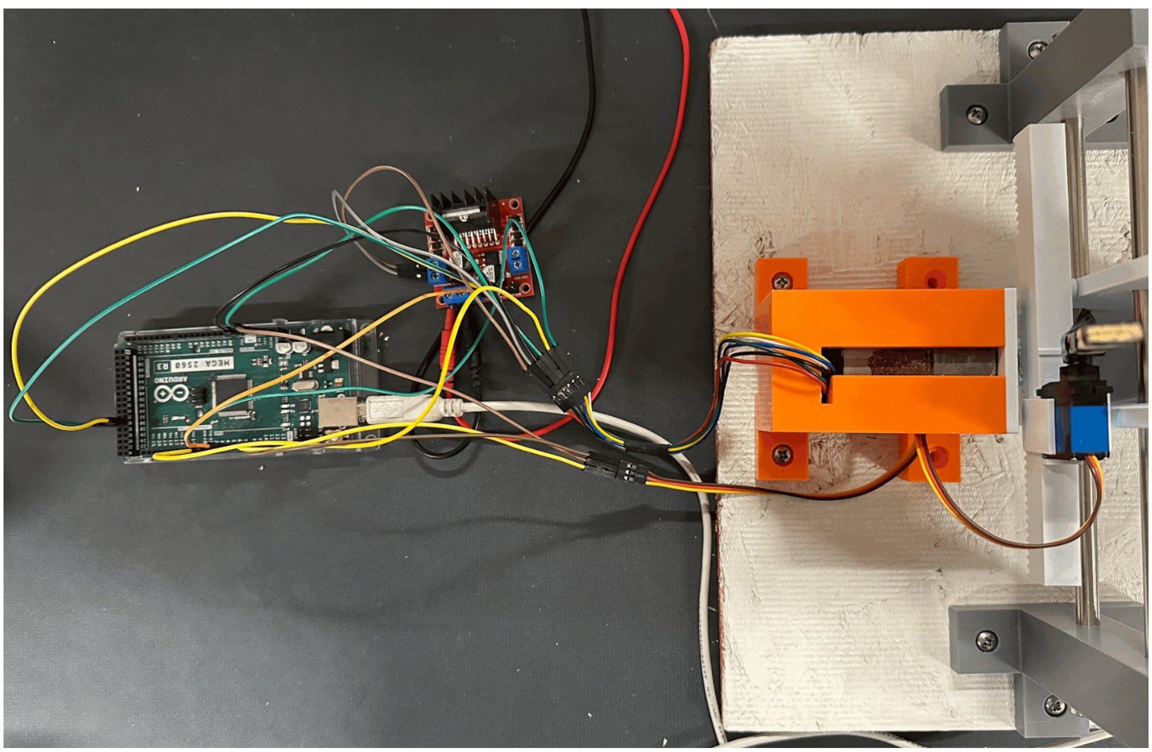
- Ensure system is 100% rule compliant during runtime
- Construct system based with provided game board and electronics, ensuring other components used total less than $300
- Design system such that electronics need no batteries, and are powered via external 12 V power supply or wall outlet
- Ensure system reliability by verifying 95% success rate on bowling attempts
- Ensure system durability by verifying the system can run over 10 separate frames back-to-back without any intervention
- Design system without pinch points, sharp edges, or exposed wiring to protect user
- Implement system rails and ball catch to guard user from ball during system runtime
- Avoid loud materials, components, or system configurations to prevent maximum system noise level from reaching above 70 dB (barrier for hearing damage)
- Implement a clean and simple graphical interface with low latency and quick system response times of less than 2 seconds at user interaction
- On user selection, quickly and efficiently target and bowl on target pins, with less than 10 seconds passing between user selection and the system's return to idle

With these requirements established, we can move on to a detailed analysis of the design choices made to achieve system functionality and adherence to the above requirements.

Design Details

We can begin our design details by first defining and analyzing the functional blocks that make up our system. The first one we examine and analyze is the electronics. As a whole, the electronics that make up this system are fairly simple. In total, the electronics consist of the Arduino 2560 Mega, which is used to talk to a servo motor, a 12 V DC motor, and a camera. We also use a motor driver and a plug-in light, as well as a laptop to connect to the Arduino. The Arduino is then plugged into the laptop and the camera is as well. The light is plugged into a standard outlet. Our efficient use of the materials given to us with little components used ultimately simplifies the system greatly, and contributes towards the customer's requirement that the system cost less than 300 dollars. Overall, the only electronics we paid for in this project were the light, which cost around $10. Excluding the laptop, the total cost of the electronics in this system would be around $130. Figures 1 and 2 below show the Arduino with the servo and DC motors connected, and also the camera system with the light configured.

Figure 1. Arduino motor connections



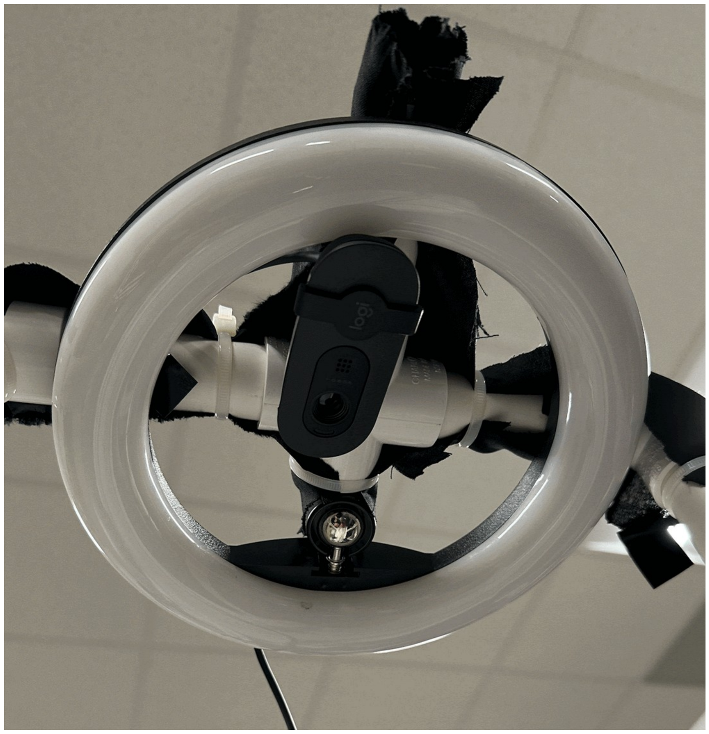


Figure 2. Ring light and camera setup

With our circuits established, we also performed some tests to ensure that our Arduino motor connections were working properly. Using Simulink, we sent signals to the DC motor to rotate it a given amount. Figure 3 below shows the oscilloscope output for the motor as it rotates. With these tests, we were able to confirm the motor does in fact spin the amount of degrees that it is sent. We also tested the servo motor, and were able to confirm that the signal sent to the servo also spins it the correct amount of degrees.

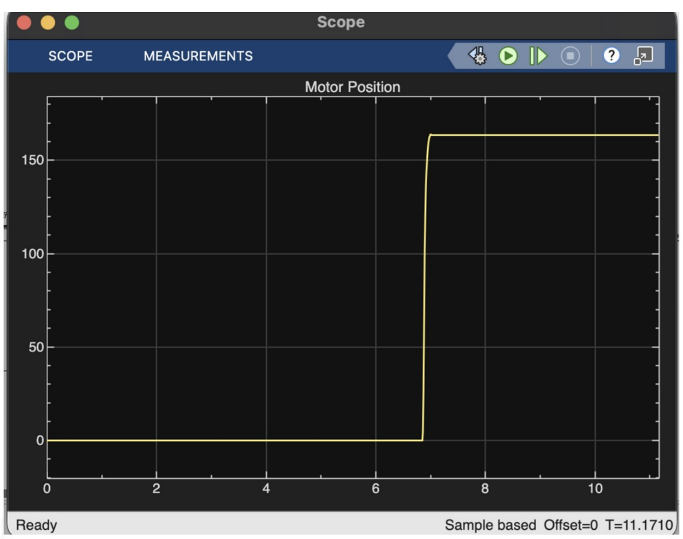


Figure 3. Motor position oscilloscope output

With our functional block for the electronics established, we can now move on to analyze our functional block for the mechanical parts of our system. The mechanical block is responsible for physically aiming and launching the metal ball once the control system chooses a pin combination. As shown in Figure 4, the main moving element is a 3D-printed curved ramp that rides on a pair of 8 mm steel guide rods. The ramp has a V-shaped channel that keeps the 5/8-inch steel ball centered as it rolls, and a tapered exit lip that straightens the trajectory as the ball leaves the ramp and enters the acrylic lane. The ramp body, rack, spur gear, motor housing, and the two vertical support stands that hold the guide rods were all modeled in Fusion 360 and printed in PLA. The hollow ramp and ribbed side walls keep the part light enough for the DC motor to move quickly while still being stiff enough to avoid visible flex during motion.

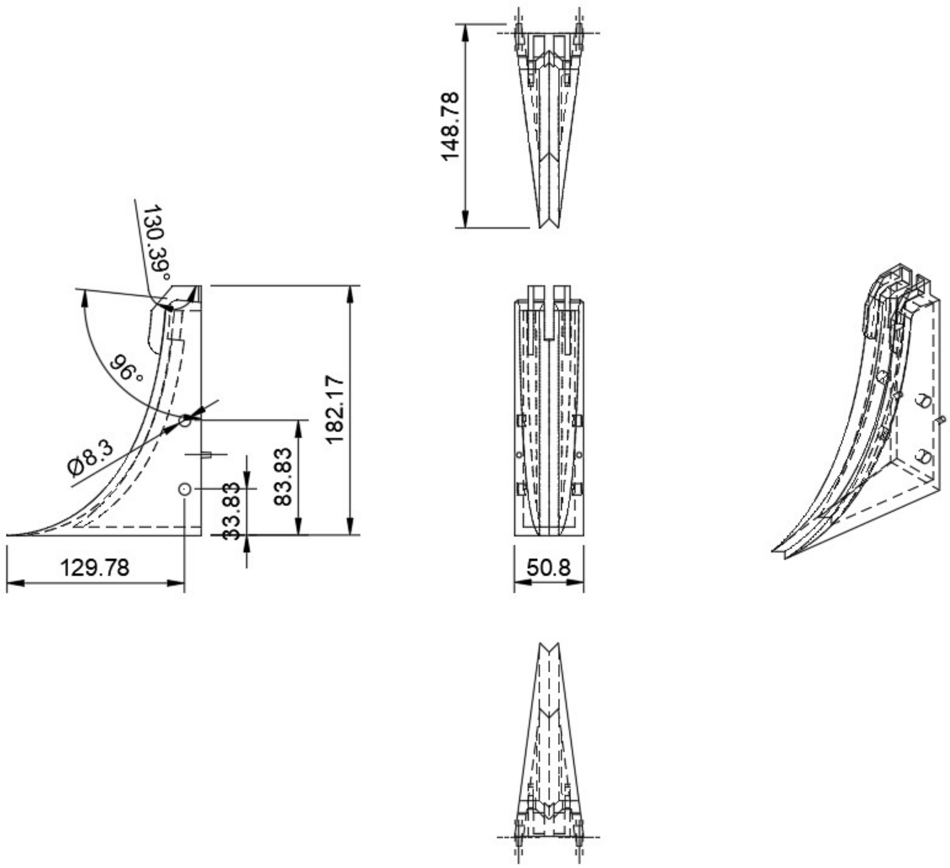


Figure 4. Modeled ramp [mm]

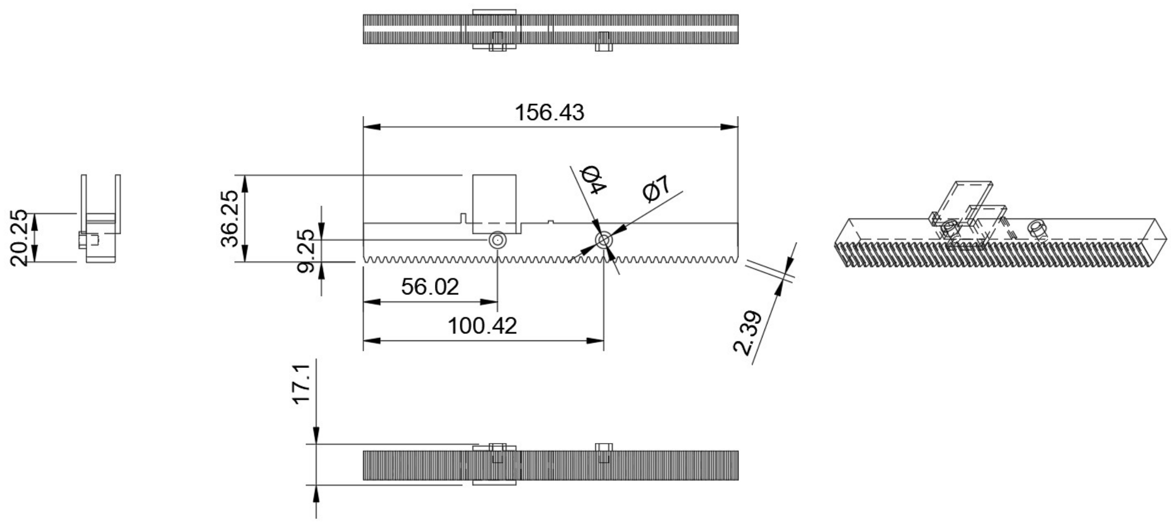


Figure 5. Modeled rack with servo housing [mm]

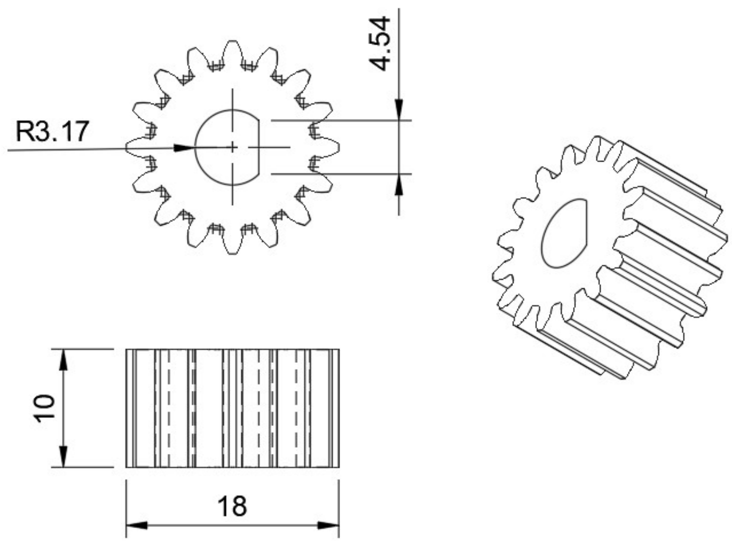


Figure 6. Modeled spur gear [mm]

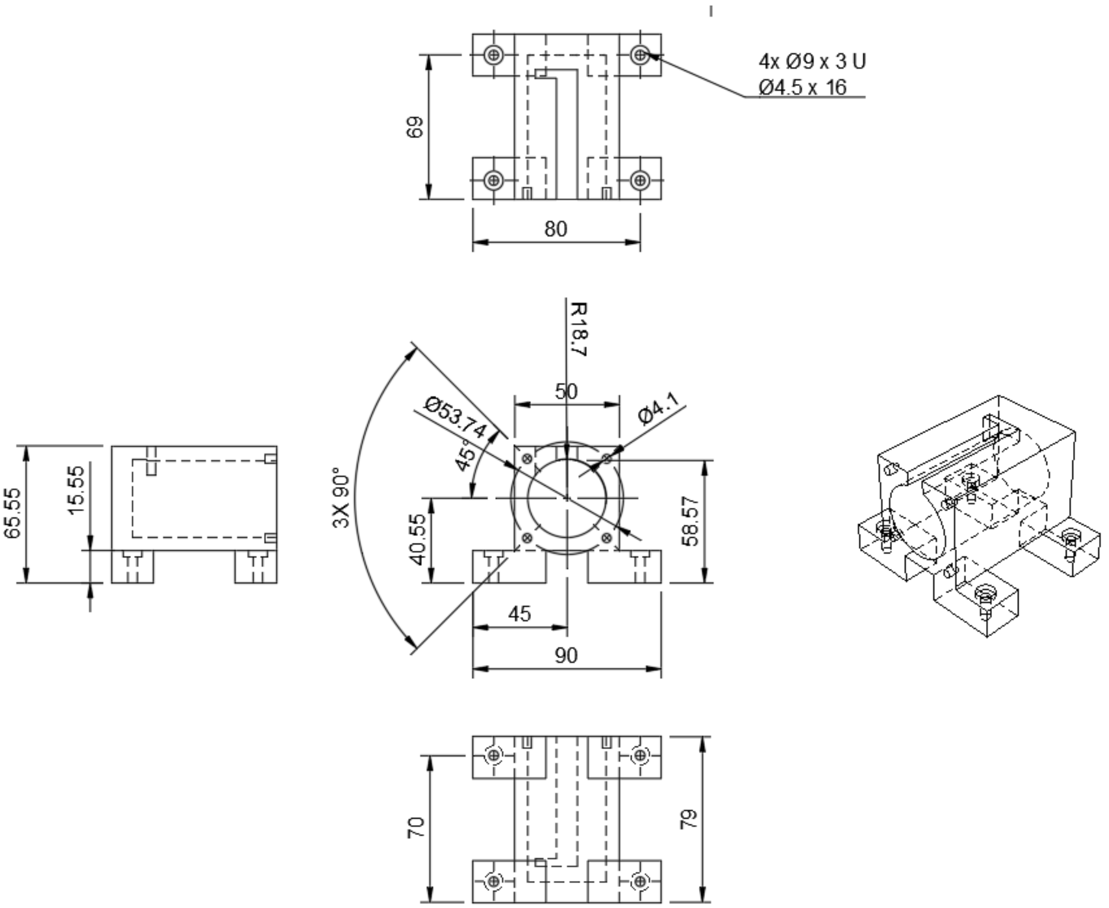


Figure 7. Modeled motor housing [mm]

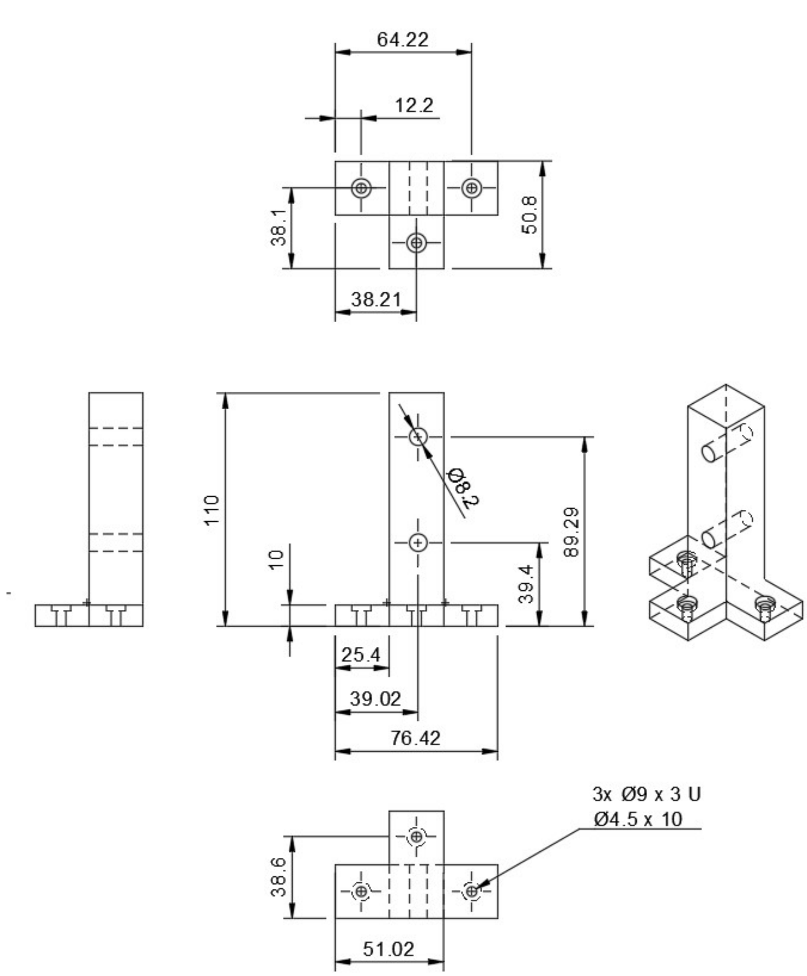


Figure 8. Modeled rod stand [mm]

Lateral positioning of the ramp is handled by a rack-and-pinion stage (figure 5). A 16-tooth, module-1 spur gear (figure 6) is press-fit onto the shaft of the 12 V DC gearmotor and runs inside a printed motor housing (figure 7) that bolts to the wooden base. The matching straight rack is press fit to the back of the ramp, so any rotation of the gear converts directly into sideways translation of the ramp along the guide rods. For a spur gear with module m (in mm/tooth) and number of teeth z, the pitch diameter is

$$d = mz$$

and the linear rack travel per motor revolution is just the pitch circumference,

$$\Delta x = \pi d = \pi mz$$

In our case m=1 mm/tooth and z =16, so

$$\Delta x = \pi(1)(16) = 50.3\, mm = 1.98 \in / rev$$

If the lane is 6 in wide, the ramp only needs to move 3 in from the center to either gutter, which corresponds to roughly

$$N = 3 \in \frac{¿}{1.98 \in / rev} = 1.5 \; motor \;\; revolutions ¿$$

This relationship is what the Simulink controller uses to map a desired lateral ramp offset into a target motor rotation.

Once the ramp is positioned over the desired release point, a second actuator handles the actual launch. A TowerPro MG92B metal-gear micro servo is mounted directly on the rack and drives a short wooden dowel through a gap at the top of the ball pocket. In the idle state, the ball sits in a pocket at the top of the ramp. When the control loop reaches the commanded position, the servo is commanded to rotate about 35◦, swinging the dowel into the groove just enough for the ball to drop into the ramp. The servo then returns to its home angle, and the ramp returns to its starting position. Mounting the servo on the rack means it always moves with the ramp, so the release geometry stays consistent regardless of where the ramp is aimed.

Structurally, all printed parts are fastened to the wooden plank using #8 screws and the motor housing features M3 screws and heat set inserts to keep it in place while the shaft is rotating. The two rod stands (figure 8) stay parallel when the ramp is accelerated back and forth. Overall, this mechanical design gives the control system a simple and repeatable mapping from motor rotation to ball landing position, while keeping the hardware light, low-cost, and easy to service.

With our mechanical block analyzed, we can finally move on to analyzing the software block of our system. All the code for the graphical interface, image processing, and movement calculations are performed in a MATLAB app. The app sends values to a Simulink model, which in turn sends movements to the DC motor and the servo motor. Figure 9 below shows a flow diagram for the software loop.

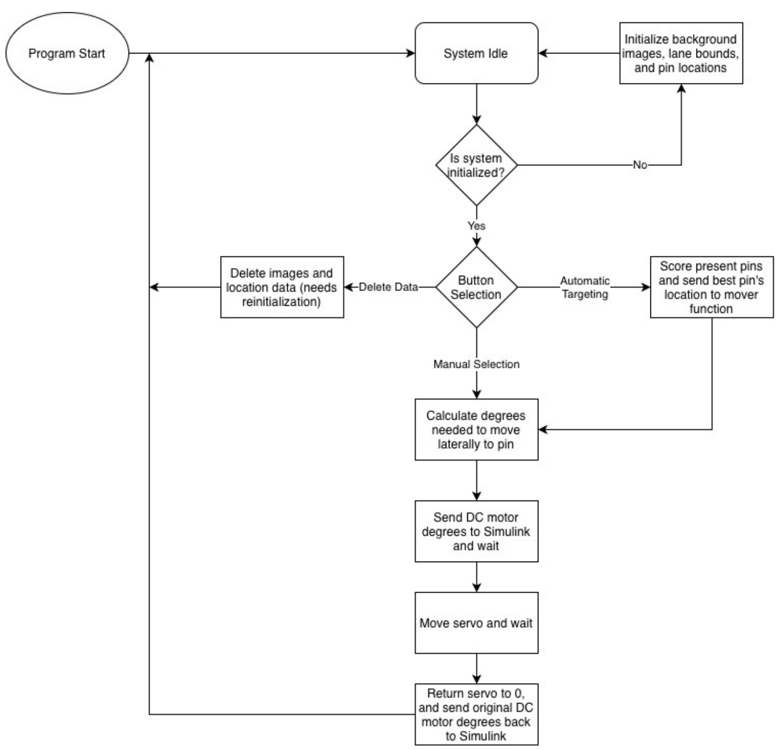


Figure 9. System software loop flow diagram

Overall, the software loop is robust, and allows the user to reset data within the app itself. On top of this, the buttons all call local functions, meaning the data stored is quite minimal. Everything stored is deleted and recaptured during runtime, so only a small and fixed amount of memory is needed for running this program. Several of the functions within the app interact with the Simulink model by assigning parameters within the model itself. In particular, targetAngle and servoAngle are the DC motor and servo motor angles that are sent to Simulink by the app. The simulink model can be seen below in Figure 10.

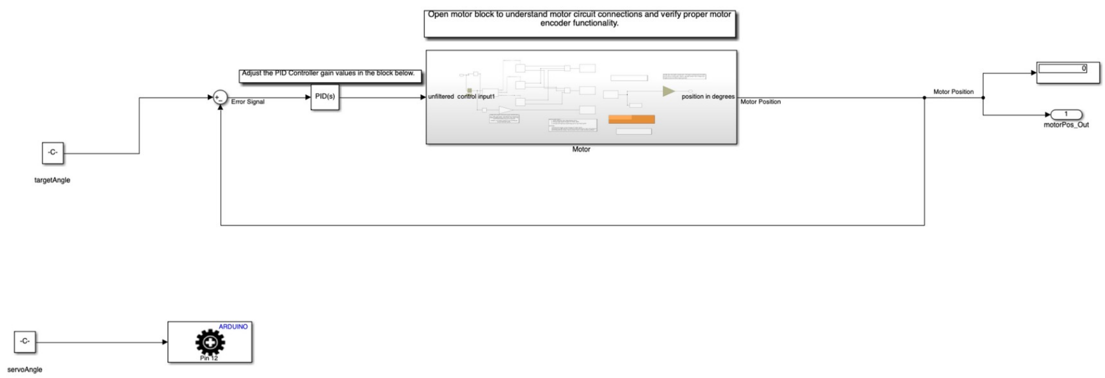


Figure 10. Simulink motor model for driving DC and servo motor

We do not store many structures within the app. There are two structures stored externally, called “laneCalibration” and “pinDotCenters.” The existence of these MATLAB files is used to confirm whether or not the program needs the user to initialize the lane bounds and pin locations for the system. Figure 11 below shows an example of the prompt displayed when the program needs lane and pin initialization.

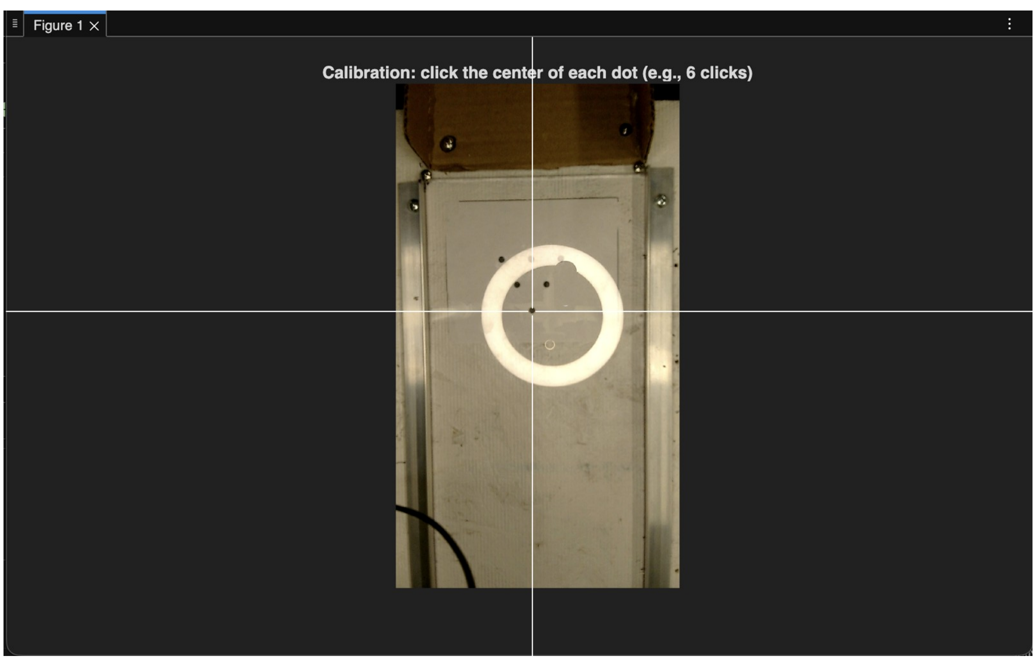


Figure 11. Pin configuration pop-up for system initialization

With our software designed, we first tested the Simulink for varying PID values, and settled on a proportional constant of 0.5 and a derivative constant of 0.01. With the motor now functioning, we tested our actual targeting functionality by placing pins and selecting them, which showed us that our system was in fact able to send the desired degrees required to move the ramp to the selected pin. We then tested the servo motor with different degree values to see the best movement value that would result in the ball being sent down the lane straight and with maximum velocity. Finally, we tested the automatic targeting functionality of our system by placing various combinations of pins, which showed us that the automatic targeting was consistently accurate. Overall, the software was shown to be an effective implementation that fulfilled our engineering requirements by being fast, robust, and accurate. On top of this, the cost of developing the software was completely free since MATLAB licenses are free to students.

Bringing everything together, we can now take all of our blocks and examine the combined system as a whole. Figure 12 below shows the whole system combined.

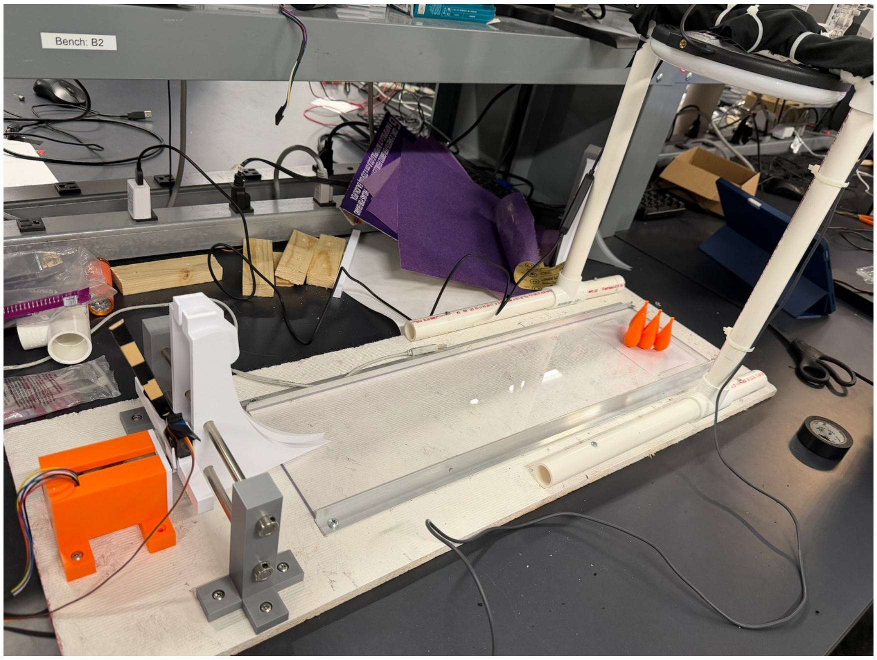


Figure 12: Picture of Implementation of Final Design

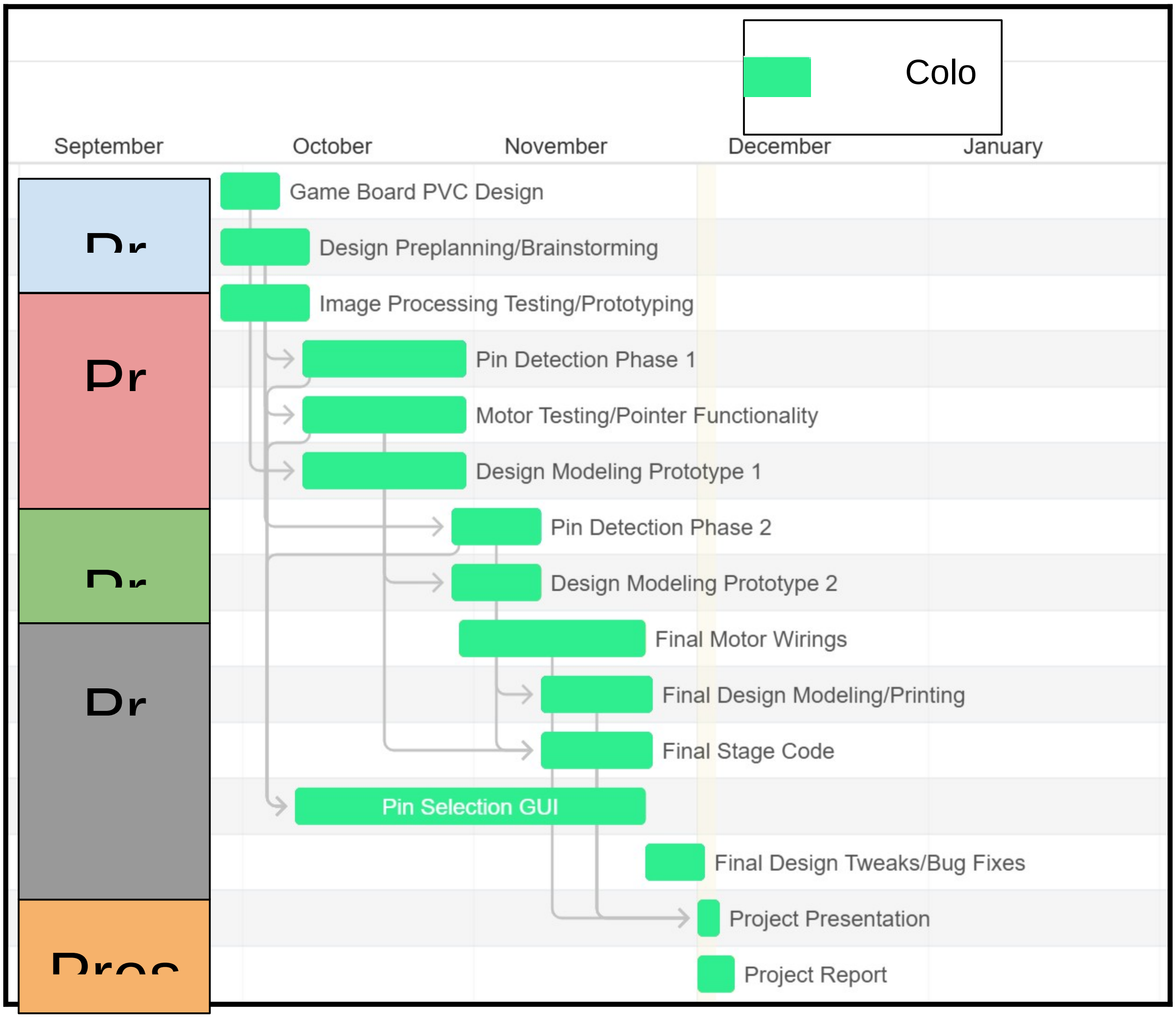
Overall, the system performs well, and within the engineering requirements we have set for ourselves. The software loop is robust and effective, and provides a simple and easy to understand interface for the user. The mechanical systems in place are safe and effective, and perform consistently without unwanted behavior. The electronics tie the software and mechanical components together and consistently actuate as intended, with motor movements occurring quickly and accurately after user interactions. On top of these successes, the total system cost paid out of pocket comes to be $67, which further fulfills our customer's requirements.

Analysis of Final Prototype Performance

Overall, the system we designed performs well, and meets all the requested requirements from the customer. Table 1 below contains the customer requirements, and how well our design fulfilled them.

Customer Requirement	Design Fulfillment
“The robot plays the game according to the rules”	Our system does as required, and targets pins selected by the user. The only user interaction comes when setting up the pins, setting the ball, and selecting the target pin.
“Uses the provided game stage, electrical and mechanical interfaces… costs <300… no batteries”	Our design utilizes all the electronics given to us, and we made minimal purchases for other components. Our mechanical system is composed of mostly 3D printed parts, and our light does not use batteries. Everything we purchased is cheap and available to anyone. As a whole, we spent only $67 which is well under the $300 mark. Even including the components given to us, the system still is only $180.
“Reliable… Durable… Safe… Low-noise”	After lots of testing, we have seen that our design works very well after many back-to-back tests, and that the system operates quietly and without any dangerous behavior.
“Fast solving times”	Our system is quite fast when it comes to pin targeting. After user input, it takes less than a second for the system to calculate and send the number of rotations for the ramp. After moving, the system quickly deploys the ball and moves the ramp back to an idle position.
“Easy to use… Runs autonomously”	We utilized MATLAB App Designer to create a clean and simple graphical interface for the user. The app makes it easy to do actions such as selecting pins manually or automatically selecting pins. The system also has varying levels of autonomous functionality, allowing the user to either select pins manually, or let the system choose the best option. The user only needs to place the ball and pins.
“Must have at least one degree-of-freedom that has closed loop position control using the provided DC motor where the loop is closed through the Arduino.”	Our design moves horizontally, being controlled by a gear and pinion using the DC motor. The motor itself is controlled through the Arduino, using the Simulink model to send angle values to the motor.

Project Schedule: Gantt Chart



Website Link:
https://sites.google.com/g.clemson.edu/team19-thebiglebowski/home

ECE 4950 Project 4 – Customer Requirements and Final Design Parameters

Use the guidelines below to complete your report and add at the end of your report.
Group Member Last Names:____

Score	Pts	
	5	**General Format - Professional Looking Document/Preparation (whole document)** a) Fonts, margins (11pt, times new roman, single spaced. 1" margins on all sides). b) Spelling and grammar are correct c) Layout of pictures – all figures need numbers and captions and must be referenced in the text d) Follows the page limitations below. e) References. Use IEEE reference format. f) This grading sheet is included as the final page.
	0	**Page 1: Title, Group Name, Group Members, and Date** **Executive Summary** (1 concise, well-written paragraph) Provide an overview of this project. Briefly describe what you did and what you learned.
	5	**Page 2: Engineering Requirements** (<1 page) Bulleted list of Final Design Engineering Requirements
	10	**Pages: 3-7: Design Details (<5 pages)** Describe a system that can be built including System Architecture and System Integration based on the Engineering Requirements. Do not include data sheets or software code.
	10	**Page 8: Analysis of Final Prototype Performance** (<1 page) Did it succeed or fail to meet customer requirements? What went wrong and what happened in the design process to allow this problem? Make a table of the customer requirements and address how well your design met these expectations.
	5	**Page 9: Project Schedule/Gantt Chart** (<1 page) Create a schedule (Gantt chart) that shows the tasks and schedule for your project. Start from the very beginning of your project and extend to the end (completing final report and presentation).
		Page 10 This grading sheet is included as the final page.
	50	Laboratory demonstration of your prototype (evaluated by instructor and TAs). Evaluator will manipulate the interface and evaluate how well the system provides the timing and display functions (i.e. how well does the closed loop control work). Is it well built? Neat wiring? (.6 * the prototype evaluation score)
	15	Rating by reviewers after competition
	15	Slideshow for Demo containing the following information: • Executive Summary • Demonstration Video • Customer and Engineering Requirements • Brief Software Explanation • Performance Analysis