Project 1 Report

Team 21
Andrew Berkeland, Andrew (AJ) Eubanks, Steven Johnson, Eric Mitchell, Alvi Topuzi
09/18/2024

Executive Summary

In this project, we had two main goals. Our first goal was to create a 3D model of a 4 in x 2 in x 1 in design of our choice. We made a model in Autodesk Inventor of a box with slanted top edges. We etched our team number on top of the box and printed our team members' names inside the box. Our second goal involved spinning a motor when an input device sent a value that passed a limit we set ourselves. We broke that goal into two subgoals: Take input values from a sensor and enable a servo motor on command. We used a potentiometer sensor as a dial. To take input from the dial, we attached it to two reference voltages, and we connected its second lug to an Arduino board. As we turned the dial clockwise, the output voltage increased. The opposite was true for counterclockwise rotation. For the second subgoal, we created a switch to enable or disable the motor. When we turned the switch to the on position, the motor slowly spun from zero to 180 degrees and back again. To meet the primary goal, we combined both subgoals into one system. We set a limit halfway between fully on and fully off for the dial. We attached the dial and the motor to the Arduino device. We then programmed the device with the halfway threshold and connected code blocks telling the Arduino what behavior the motor should have. When the input voltage met the threshold, the motor continuously swept from zero to 180 degrees. During testing, we were able to change the threshold to enable the servo to about two-fifths of the reference voltage during operation. We were able to meet both major goals for this project.

Materials and Methods

For this project, we used the Arduino Mega 2560 R3, a 47 uF capacitor, a 100 uF capacitor, a 2 kΩ potentiometer, and an MG92B standard servo motor. We used Autodesk Inventor Professional 2024 for 3D modeling and MATLAB version R2024a for Simulink.

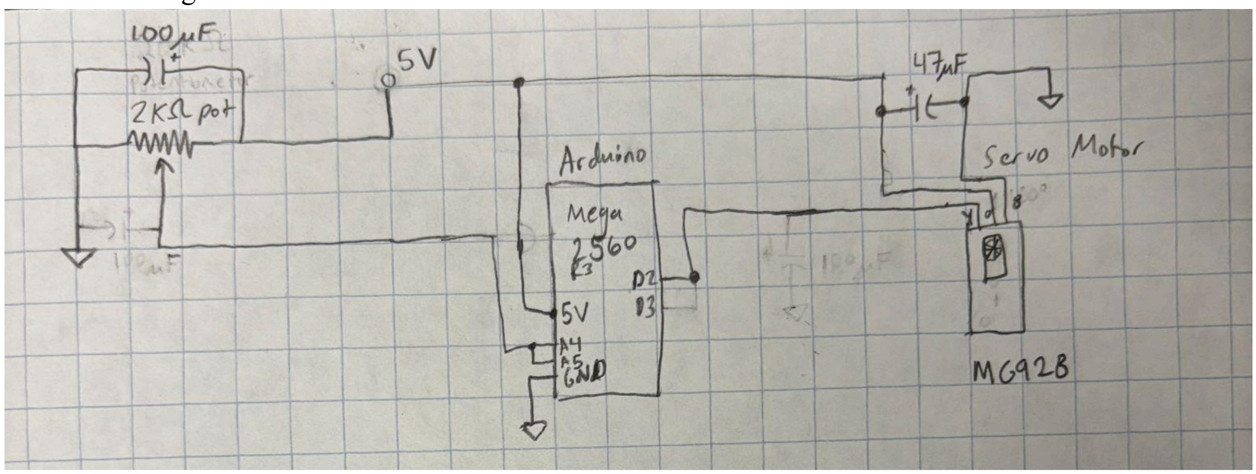

Figure 1: Circuit Diagram for Project 1

We began by assembling the circuit shown in Figure 1. The Arduino board's 5V and ground are our circuit power and ground. We wired our potentiometer to 5V and ground and used the wiper as our analog signal source. We placed a 100 uF electrolytic capacitor between the potentiometer's 5V connection and ground. The anode of the capacitor was connected to the 5V signal. The capacitor mitigates voltage spikes since the voltage across a capacitor cannot change instantaneously. Using a capacitor as described protects the Arduino from voltage spikes and smooths the voltage response. We used analog pins 4 and 5 (A4 and A5 in the diagram) as our analog inputs. We then connected our servo motor to 5V and ground. We placed a 47 uF capacitor between the servo's power and ground connections to prevent spikes in the operating voltage. Then, we attached the servo's signal connector to digital pin 2. We would later connect it to digital pin 3 in the final programming task.

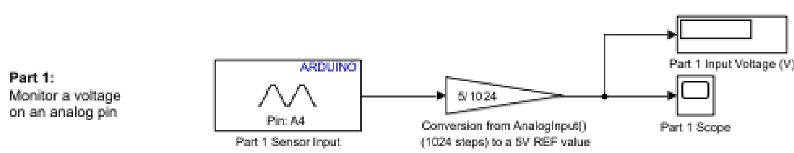

Figure 2: Simulink model for programming task 1

Our first programming task was to read from an analog sensor using the Arduino. Our analog sensor is a potentiometer, so we read the voltage from its wiper to ground. This voltage was available on analog pin 4 (A4). The gain factor in the model is $\frac{5}{1024}$, which is a conversion factor. The analog input pins on this Arduino model give a value in the range 0-1023. We must multiply the sensor input value by the gain factor to convert the analog input reading into a voltage value. Our final steps in this model were to display the voltage in a display and to graph it using a scope. To run the code, we pressed "Monitor and Tune" in the "Hardware" > "Run on Hardware" tab. This is an essential step for running any code because it allows Simulink to obtain and display information from the Arduino.

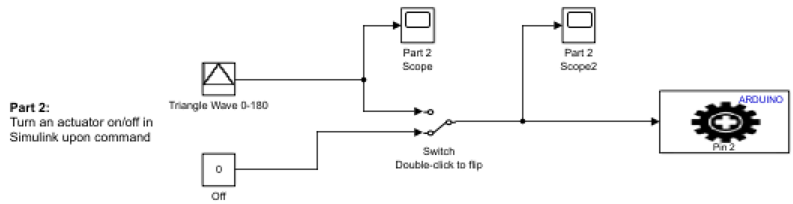

Figure 3: Programming task 2 model

Our second programming task required us to be able to turn a servo on and off in Simulink. We chose to use a switch to toggle the servo's movement. Figure 3 shows the model we used. We used a triangle wave that sweeps from 0-180 and back in reverse. We chose these values because the standard servo write takes angles between 0 and 180 degrees to send the servo to that position. When the switch is off, the servo goes to position 0. To test our code, we ran the program as previously described. We expected the servo to rotate from 0 to 180 degrees and back in a smooth motion, and it did.

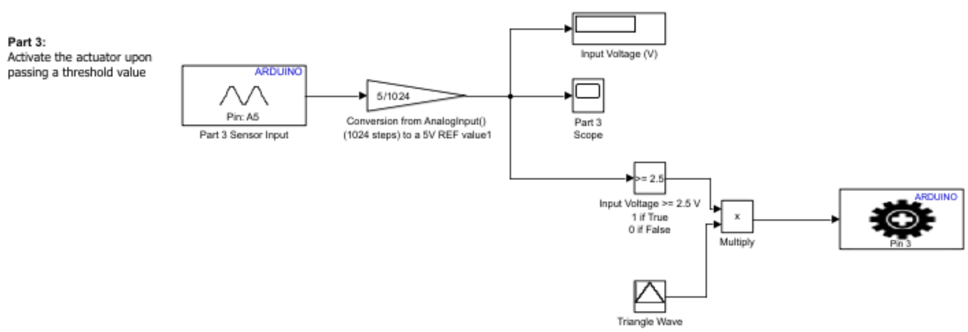

Figure 4: Programming Task 3 Model

Our last programming task was a combination of the prior two. The goal was to have the servo start moving when the voltage read from the potentiometer reached a threshold value. We had one writing change: we moved the analog input to analog pin five and the digital output to digital pin three. From there, we combined the previous models and added two new blocks. We set a threshold using a compare-to-constant block, shown in Figure 4. If the input voltage passes the threshold, the block outputs a one. Otherwise, it outputs a zero. We multiplied the result of this block with the triangle wave from task 2. In effect, the servo moves when the threshold is passed. Our first threshold was 2.5 V, and the second threshold was 2 V. We ran the code the same way as before, and our design worked with both threshold values.

The final stage in our project was to design a part in CAD with the team number embossed or debossed on it. Since one of our members was familiar with Inventor, we used it as our modeling software. We chose to create a box to house components in. On top of the box, we debossed the team number. The top edges are chamfered. We embossed our names inside the box as well.

Results and Discussion

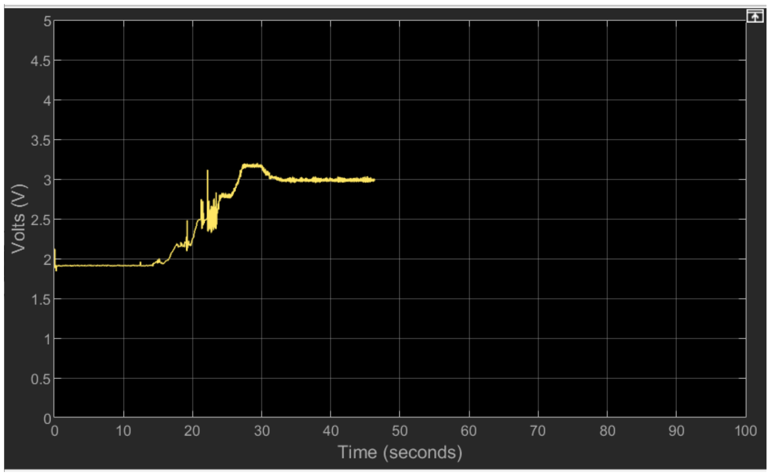

Figure 5: Programming task 1 scope output and voltage reading

We took screen captures of the scope figures for the three experiments. Figure 5 shows the scope output and voltage reading. The display in the top left corner shows an instantaneous voltage reading, which is 2.969 V at the time of the screenshot. The scope shows the voltage output from the potentiometer as we tuned the resistance to get a 2 V output up to 3 V. Our graph and display prove that we were able to read analog values from our potentiometer, completing the first task.

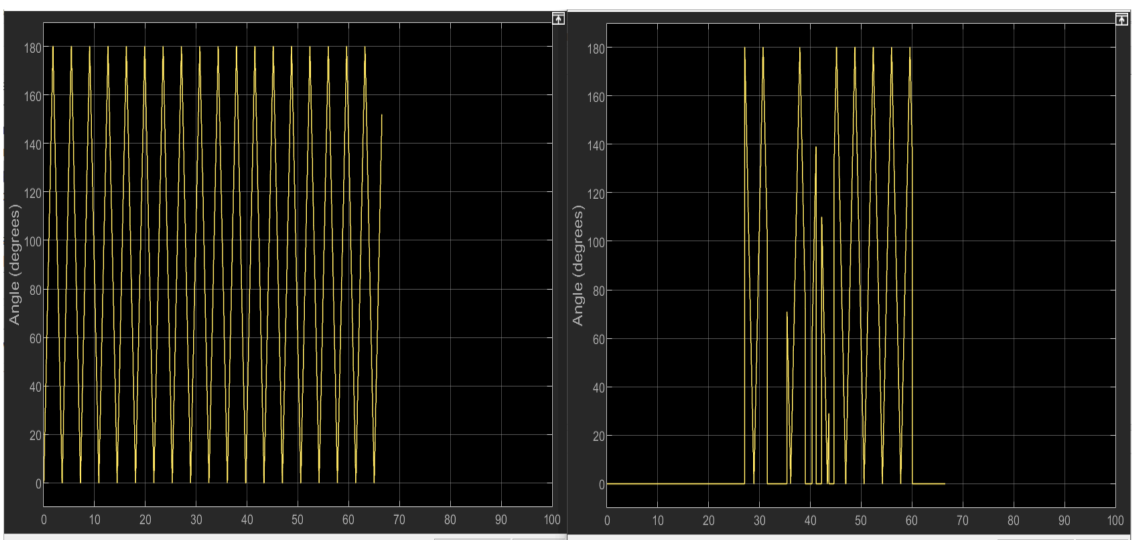

Figure 6: Programming task 2 input waveform (left) and output where we toggled the switch many times (right)

The second task's scope readings are shown in Figure 6. The scope output on the left shows a triangle wave from 0-180 degrees. The scope on the right displays the same waveform with one exception: we toggled the switch in Figure 3. In Figure 6, the incomplete triangles on the scope on the right show when the switch was toggled. This waveform is the one that the servo motor used to determine which position to go to. These graphs show that we completed the toggle operation as well.

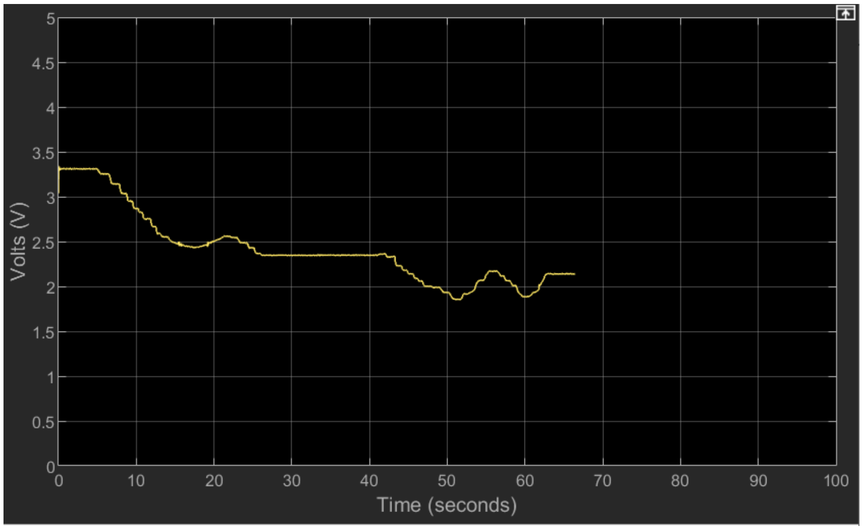

Figure 7: Programming task 3 with the threshold at 2.5 V (left of t=40s) and 2 V (right of t=40s)

The third task's voltage output is displayed in Figure 7. The waveform to the left of t=40s shows the voltage when the threshold was set to 2.5 V. Originally, we had rapid changes in voltage on the graph occur when the servo transitions from on to off. The Arduino's voltage output is not stable enough to prevent this when the servo changes from on to off. We remedied this by placing a 100 uF capacitor across the potentiometer, as described in a previous section. This helped stabilize the response when the servo turned on and off. We tested our thresholds by dipping above and below each threshold. In Figure 7, the sinusoidal-looking areas show where we straddled the threshold. This did toggle our servo on and off as expected.

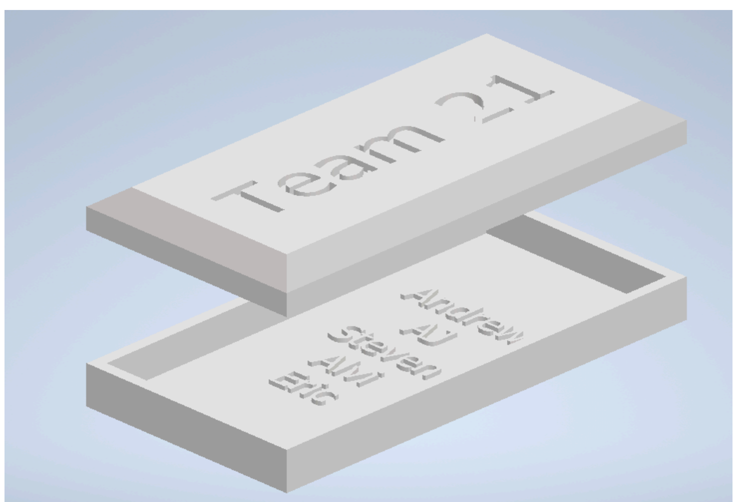

Figure 8: Box 3D Model

Figure 8 shows the 3D model described in the last paragraph of the Materials and Methods section. It is an Inventor assembly with the team name debossed and the members' names embossed. This design shows that our team is able to work with CAD software.

Conclusions and References

Based on the results of Project 1, we recommend the equipment used in this experiment for a robot control project. The MG92B is not high-quality, but other servo motors with continuous motion and precise angle movements could be helpful in robotics tasks. We could integrate a potentiometer into our system as an angle detector to control a servo motor's position. Closed-loop control coming from a sensor and feeding back to the controller is a fundamental controls concept. We could duplicate this setup to create a larger system. With additional sensors and outputs, it could be suitable for robotics tasks.

One limitation we observed was the stability of the voltage output, especially during transitions between the servo's on and off states. The rapid voltage fluctuations interfered with our system. They would interfere with a more sensitive system as well. For this reason, we added capacitors connecting signal sources to ground connections to relieve noise and reduce the transient. Unfortunately, this did not completely solve our issue. We had loose wires, so when we turned the potentiometer, we accidentally bumped the wires, causing voltage spikes. We would recommend using capacitors and stable wiring in a larger-scale application.

In addition to the control features, the simplicity of the interface between the Arduino and Simulink made programming and troubleshooting more straightforward. We were able to rapidly prototype as well. The Simulink interface and the Arduino enabled us to visualize data in real time, ensuring our system's smooth operation. Rather than creating code and algorithms, Simulink allowed us to assemble a system at a higher level, saving time and effort. This combination of hardware and software allowed us to design and perform our given task quickly.

Additionally, the use of 3D modeling and printing is beneficial for the design of our project. Creating custom components for our final project would allow us to create a more precise design rather than solely using the woodshop and wooden materials. We could use 3D printing to create custom game components such as the pieces or the game board. This will allow us to control the pieces' size, shape, and weight to work better with our electronic manipulator design. Our prototype may require custom components to create an arm or housing for electronic components.

Another important thing to consider is the ergonomics of the pieces and components used for our game. The ability to 3D model and print parts allows us to model our parts to comply with the best ease of use for the end user. Another benefit of using a 3D printer throughout our design process is that we can iteratively go through and change things as needed. If parts do not fit together just right, we are able to print a different version to fit our needs better. With woodworking or machining, it is much more difficult to go back and change things as frequently as our project may require. Additionally, with 3D printing, we can have a more consistent look across our project. If we 3-D print most of our non-electrical components, our prototype will look more professional than if we created parts using varying methods. This process demonstrates how 3D printing can provide consistency within our physical parts, offering a useful alternative to woodworking. Producing a 3D-printed prototype can save time and money further down the road, and it requires less hands-on skills than woodworking requires.

From the design process, we learned that careful planning and iteration are crucial to achieving a successful final product. Combining the aspects of Simulink modeling with the physical assembly of electronic components provided a comprehensive understanding of how digital designs can translate into physical systems and how they can communicate with one another. We also learned the importance of testing and refining thresholds in control systems to ensure reliable performance.

ECE 4950 Project 1 – Research Report Rubric

Each group will create a report that will eventually become a section in the **“Research”** section of your final project website. Use the guidelines below to complete your report and add at the end of your report.

Group Number (Group Name optional):

Group Member Last Names:____

Score	Pts		
	15	**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.	**O3-SA1:1**
	15	<u>**Page 1: Title, Group Name, Group Members, and Date**</u> <u>**Executive Summary**</u> (~1/3 of the page) Provide a summary of the whole project. Use language that targets **a non-technical audience**. An important skill for an engineer is to communicate complex technical information to a general audience that may be involved in decision making, e.g. marketing. Important criteria: a) Can a non-technical audience (~ high-school degree) read this section and understand your goals, procedures, and conclusions? b) Use simple words and graphics to help explain	**O3-SA1:1**
	30	The next sections of the report follow the standard **laboratory report format**. <u>**Page 2: Materials and Methods for the Sensor/Actuator Demonstration**</u> (1 page) You are establishing the credibility and usefulness of your results by providing all the details so that <u>someone else could repeat your experiment</u>. As an example, MATLAB 2021a may behave differently than MATLAB 2021b – the software version information which would be required to reproduce your result should be included. This section should answer the following: a) What equipment is used, include software versions. **b)** How were the experiments conducted? How is the equipment connected and used? Describe the instrumentation, cables, connections, and experiments using diagrams and photos. You should have drawings (pin connection and connector part numbers) <u>**Pages 3-4: Results and Discussion for the Experiments**</u> (~1 page) Describe what you have done. Include plots (from MATLAB, not photos of the Target screen) for each of the three experiments and a brief discussion of how you interpret the result. Did you demonstrate (through your documentation) that the equipment has been configured and used correctly?	**O2-SA2:3** **O1-SA5:4**
	5	<u>**Page 4-5: Conclusions and References**</u> (~ 1 page) a) Based on this experiment, do you recommend this equipment for use in a robot control project? What are the possible limitations? Your results and observations should be the basis for your conclusions. (~1/2 page) b) What are the possible uses for the 3D Printer in your projects? (~1/4 page) c) What did you learn from the process of determining your final project design? <u>**Page 6: This Grading Sheet**</u>	**O3-SA1:1**
	5	<u>**3D Printed Part**</u> (turn in with printed report) Grading based on: a) Does the design meet the requirements? b) Originality and creativity	**O7-SA1:1**