ECE 1000 Final Report: Remote Controlled Robotic Arm

Ethan Adams and Silas Garmon

Tennessee Technological University

Department of Electrical and Computer Engineering

Cookeville, TN, USA

emadams42@tntech.edu, shgarmon42@tntech.edu

***Abstract* – The Robotic Arm discussed further in the report combines designing of machinery parts using 3D printing, programming using a microcontroller, and electrical components to create an efficient Robotic Arm. Operated by a Raspberry Pi Pico, the Microcontroller interprets inputs from a Joystick and will use the data to operate three Servos. This report describes the functionality, the design, the components used, potential flaws, and the events that occurred during the production of our Robotic Arm.**

Introduction

Motivated by seeing Robotic Arms in the past and having experience in creating Robotic Arms using kits and having worked with various projects in robotics clubs, we wanted to improve on previous attempts of constructing Robotic Arms to be able to gain experience that can be brought into engineering clubs and career applications. This project also allows us to have experience with programming microcontrollers along with designing and constructing parts of a device. Consisting of Ethan Adams and Silas Garmon, the group is determined to construct a robotic arm that has ease of function and use. With this report, our goal is to elucidate our processes and results with this project.

I. Background

In developing our design for the Robotic Arm, we used a mixture of our own experience with similar designs and programming software, along with that of others online in forums and similar webpages. Altogether, we were both solidly experienced in development and programming of Robotic Arms in the past, and as such, we used few sources in developing the CAD files & programming the servos, and the sources used have been properly cited under the References section.

II. Project Description & Formulation

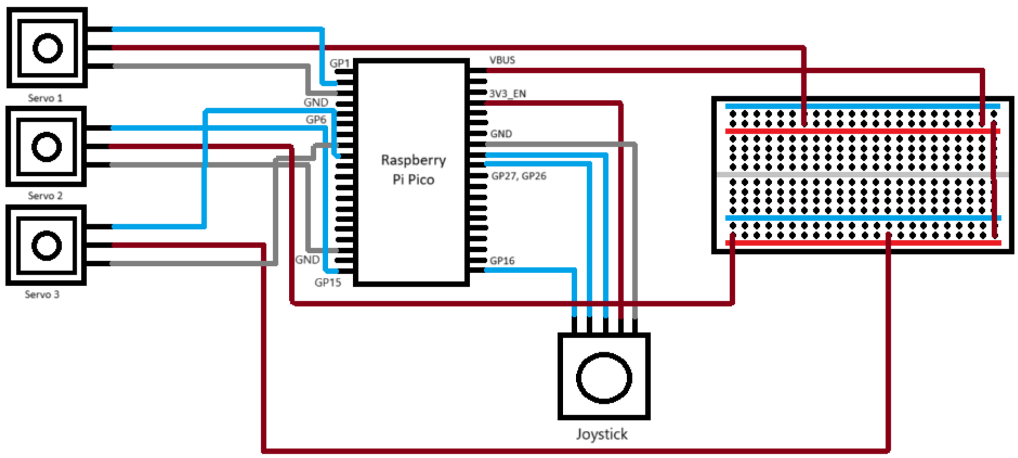
Materials:

1. Raspberry Pi Pico (RPP) Kit – The kit containing the microcontroller used for this project which serves as the unit that processes inputs from the joystick and uses the information control the servomotors, along with the wires used in connecting each electronic component.
2. Arduino 2-Axis Joystick – The joystick is used to control the up, down, left, and right motions of the arm, as well as the opening and closing of the claw by pressing in on the joystick.
3. SG90 Servomotors – The servomotors were the components used to control the position of the joints of the arm.
4. 2d Roofing Nails (x14) – We did not have the self-threading screws necessary to mount the servos, so we opted to use nails to attach the servos to the 3d-printed parts and the makeshift base. Though we intended to use six nails – two per servo – we ended up using 14 due to the extra 8 needed to craft the base.
5. ELEGOO PLA 3D 1.75mm Printing Filament – This is the specific filament used to print each component used to craft the arm.
6. Plywood Board – Though unintended, we ended up using pieces of a plywood board to craft the base since the intended base failed to print during the allotted time frame.

Diagram:

The circuit diagram shown below shows how each of the components connect. While simple to make, the circuit proved effective, and its simplicity turned to be a pleasantly surprising benefactor. The joystick controls all inputs, with left and right motions powering one servo, and up and down motions powering another servo. The final servo, which controls the claw, was controlled by pressing in on the joystick.

Figure 1: The Circuit Diagram of the Raspberry Pi Pico Breadboard used to program the Robotic Arm.



The Figures below show CAD models of each component of our Robotic Arm, including an Assembly of each component to show how it would move & fit together. Each component is mechanically intuitive, using gears or the capability PLA has to flex and bend with slight force to fit into certain spaces, allowing adhesion of certain parts without use of adhesive.

Figure 2: The CAD model of the Robotic Arm's Arm. There is a slot that corresponds to an extrusion in the Rotating Base Piece which the Arm rotates about, and a slot on the other side which a Servo's included gear attaches to.

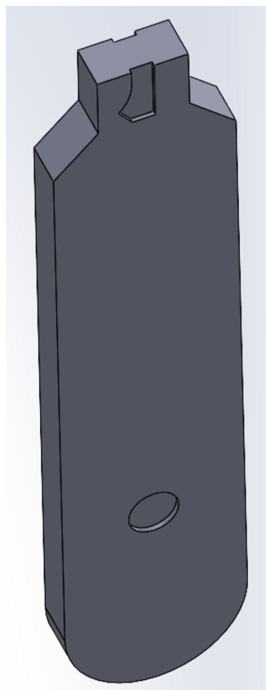


Figure 3: The Base intended to be used on the Robotic Arm. Though it didn't come to fruition, there would have been a slot for a Servo to rest, and a port for its wire to escape.

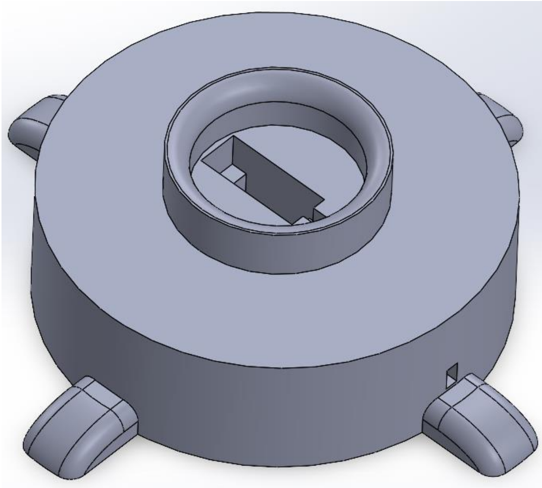


Figure 4: The Claw Baseplate for the Robotic Arm. It has a point which slots perfectly onto the cuts in the Arm, locking the Baseplate in place on the Arm. It also has various points where gears can be attached without the use of adhesive.

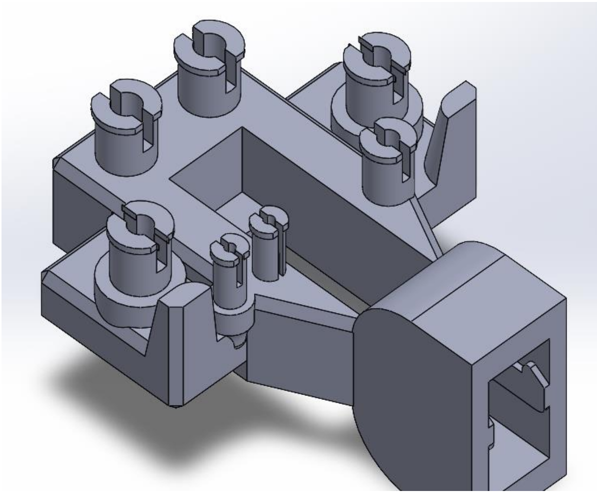


Figure 5: A Claw Pincher for the Robotic Arm. There are two points which the Pincher Gear & Pincher Stabilizer can be attached to this piece without adhesive.

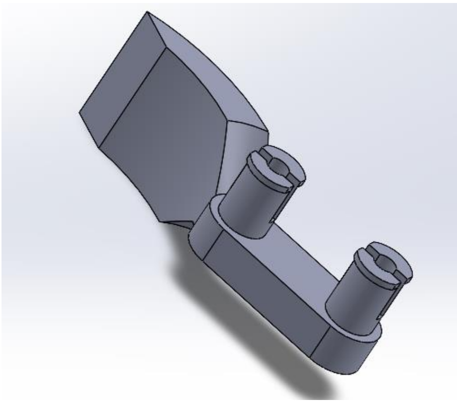


Figure 6: the Pincher Gear for the Robotic Arm. This is designed to be moved by a gear in which the Servo on the Claw Baseplate with turn, moving the Claw Pincher with this piece.

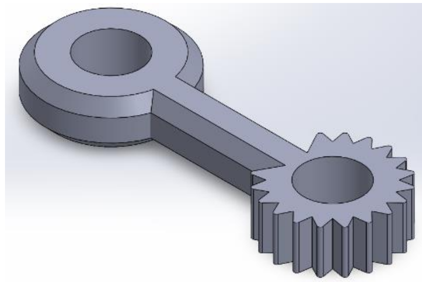


Figure 7: The Pincher Stabilizer for the Robotic Arm. This piece stabilizes the Claw Pincher onto the Claw Baseplate, ensuring that the Claw Pincher is consistently locked into place.

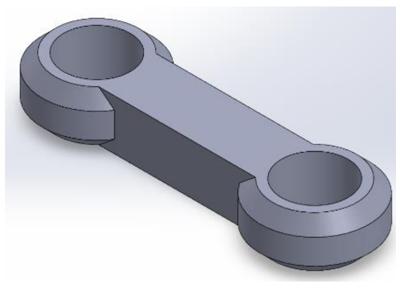


Figure 8: The Servo Gear. This piece attaches directly onto the Servo, turning with it without the use of any adhesive. Designed to be place on the Servo attached to the Claw Baseplate.

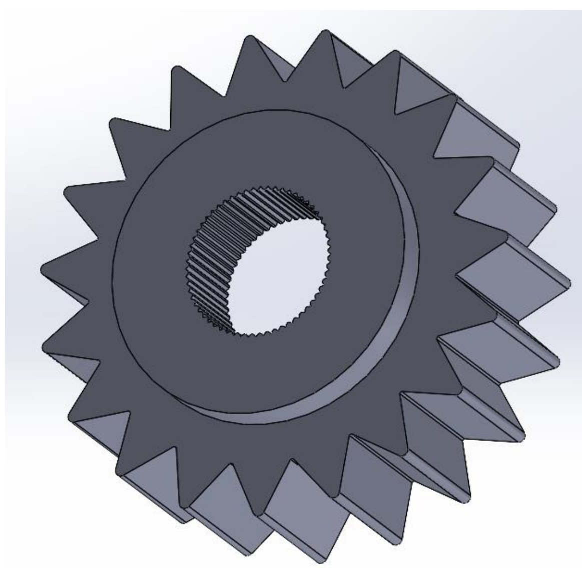


Figure 9: The Rotating Base Piece for the Robotic Arm. This piece was specially designed to house the & to be placed on the Base, swiveling left & right with a Servo attached directly to the bottom in a similar fashion as seen in Figure 8.

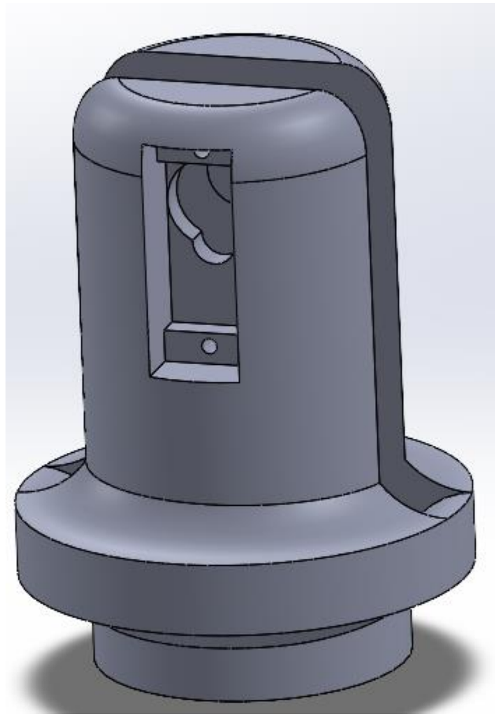
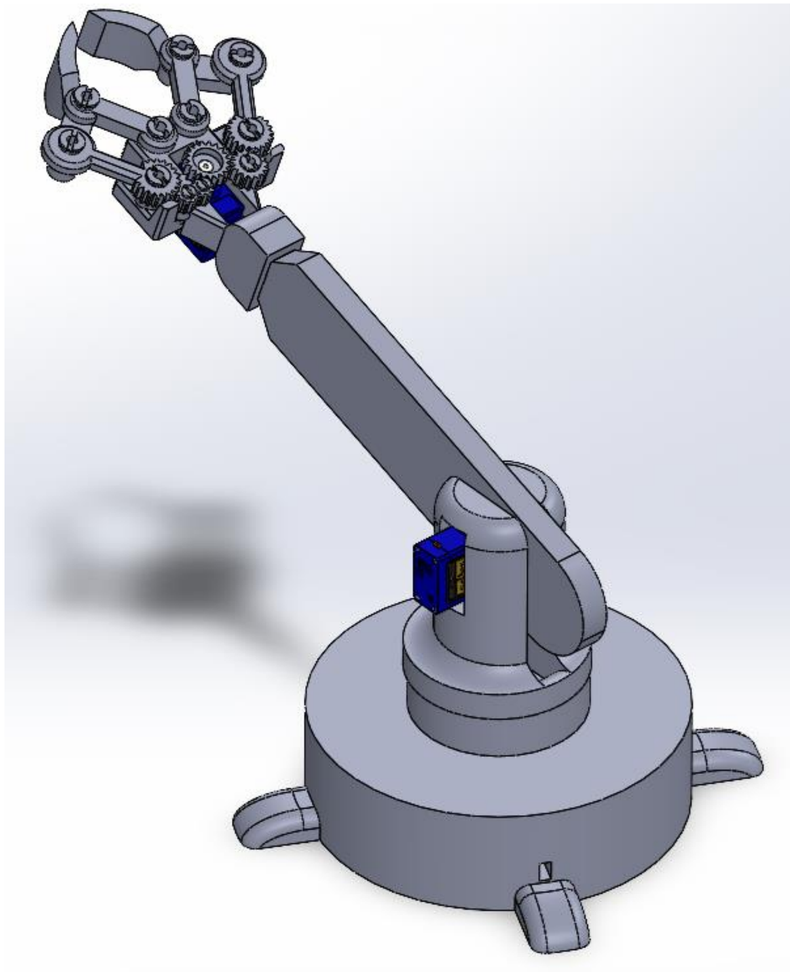


Figure 10: The full Assembly of the Robotic Arm.



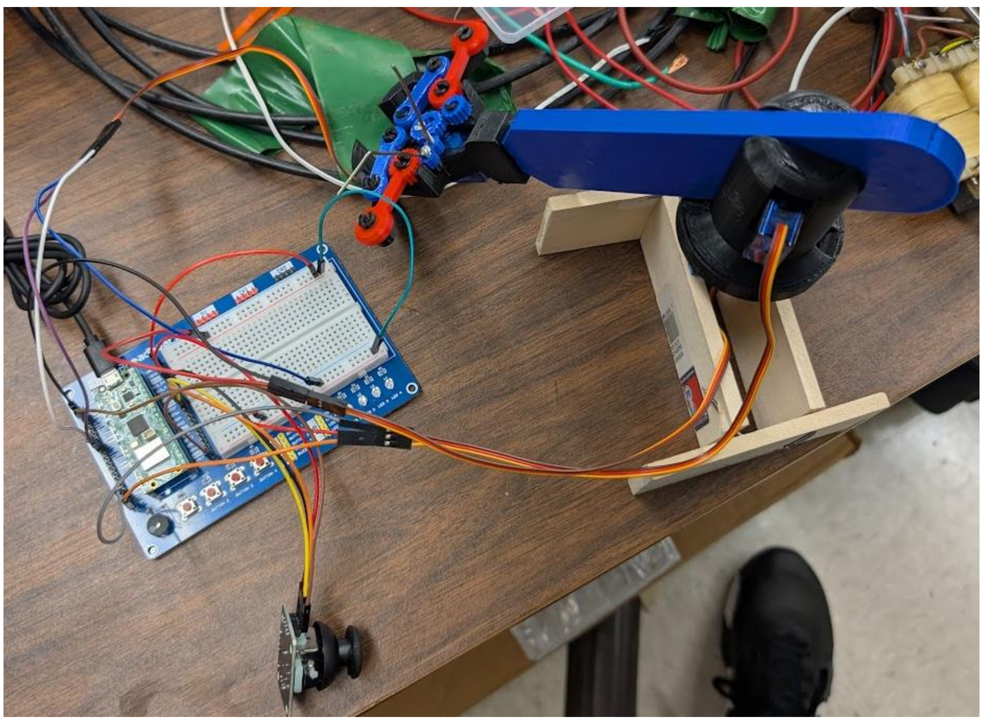
Full System:

Our programming process involved researching how these components worked within the MicroPython library & Raspberry Pi Pico Breadboard. Once researched, various test runs were completed to see how the Joystick and Servos worked independently. Using this knowledge, a basic code was written to test controlling the Servos using the Joystick, with one Servo designed to respond to left-right inputs, another to respond to up-down inputs, and the final Servo to respond to pressing in on the Joystick.

To create the Arm components, we modelled each component in SolidWorks, carefully referencing each part to ensure they fit together smoothly to minimize the screws and glue necessary to assemble the full design. Once each part was modelled, we created an Assembly to ensure that each piece fit together smoothly and adjusted each piece as necessary to reduce the possibility of a faulty part. Finally, each part was printed over the span of 3 days, and the design prototype was assembled and tested, succeeding in its results!

Figure __ shows the full assembly of the 3D-printed components and the circuit built from the diagram shown in Figure __. The depicted assembly is the main prototype that was presented with this project, and while some components were rough to fit together, simple tweaking of the CAD models here would yield strong results. Additionally, the Base shown in Figure__, which was meant to strongly weigh the arm down to the surface, failed to print in the time allotted, so we created a makeshift base to stabilize and weigh down the design.

Figure 11: The full Prototype & Circuitry for the Robotic Arm.



Functionality:

The Raspberry Pi Pico Breadboard responds in real-time to inputs from the Arduino Joystick, moving the arm in its corresponding directions when inputted into the joystick.

The Servo attached at the arm's base is designed to swivel the Arm left and right in a 180° range when either left or right is inputted on the Joystick. In doing so, the Arm is effectively aimed left and right to provide an expanded range of motion for users to work with, allowing rotation about the y-axis.

The Servo attached at the side of the rotating base piece is placed & calibrated to rotate the Arm up and down in a 180° range when either up or down is inputted on the Joystick, allowing users to pick objects up and place them down lightly. As a result, the Arm is enabled to rotate about the x-axis (or z-axis, depending on one's perspective).

The final Servo attached to the Claw Baseplate is linked a chain of gears – two on the right side and 3 on the left side – allowing an opening and closing of the claw by powering the Servo to rotate approximately 90°, fully opening and closing the claw to allow users to pick and up and release objects that fit within the claw's grasp. The differing number of gears allows the claw to properly open by reversing the direction of the left Claw Pincher in relation to the right Claw Pincher.

III. Discussion & Results

The Robotic Arm as a whole was a success, with each component correctly responding to inputs from the Joystick and moving smoothly with the Servos. While some parts fit together extremely roughly, the arm as a whole succeeded in moving in three dimensions, and in opening and closing the claw to pick up (albeit light) loads. However, our tests were numbered due to an issue where both of our Raspberry Pi Pico Breadboards refused to connect for more than two minutes to either of our laptops, using either of our USB-A to USB-C Cables. This flaw negatively affects the longevity of any tasks completed by the arm and harmed our ability to test and improve upon our code.

Additionally, our team should consider increasing the size of the Claw Baseplate, as doing so would allow for larger gears which may print smoother and would fit together less roughly. In addition, doing so would strengthen the attachment points by which the gears revolve around, which can be seen to have cracks or even snapped, but still hold the gears to the plate without affecting their rotation, aside from the two smallest attachment points, which entirely broke off and were replaced with screws.

In terms of individual contributions:

Ethan Adams was responsible for modelling, printing and assembling the design using Solidworks and the 3D printers available at the Tennessee Technological University's Angelo Volpe Library iMakerSpace.

Silas Garmon was responsible for coding the servos & testing the code using Thonny & his Raspberry Pi Pico Breadboard Kit.

IV. Conclusion

Coding with MicroPython and a Raspberry Pi Pico Kit proves extremely versatile and can be executed to perform extremely complex and vastly different projects. Its use in our project allowed us to code and implement each component extremely easily, despite our technological errors during the process.

The creation of our Robotic Arm using simple Servos and the Raspberry Pi Pico Kit, along with parts created from SolidWorks demonstrates what creativity and effort can achieve from limited resources. By using the tools and knowledge at our disposal, we were able to combine our prior experience and knowledge, filling in the gaps with online resources to facilitate the creation and use of a compact Robotic Arm that would otherwise have been extremely difficult to create without any one of the tools we had at our disposal.

By using simple code, circuitry, and CADwork, we were able to create a design for a Robotic Arm akin to that of what might be seen at manufacturing plants or even just in robotics competitions seen in educational institutions & clubs.

V. References

[1] Piltch, Avram (25 June 2022). *How to Connect an Analog Joystick to Raspberry Pi Pico*. Tom's Hardware. https://www.tomshardware.com/how-to/raspberry-pi-pico-joystick.

[2] Author Unlisted (5 March 2022). *Servo Motor with Raspberry Pi Pico using MicroPython*. Microcontrollers Lab. https://microcontrollerslab.com/servo-motor-raspberry-pi-pico-micropython/.

[3] Author Unlisted (Date Accessed 4 December 2024). *1. Controlling hobby servo motors*. MicroPython. https://docs.micropython.org/en/latest/pyboard/tutorial/servo.html

[4] Frasson, Mattheus (11 November 2017). *SG90 - Micro Servo 9g - Tower Pro.* GrabCAD. https://grabcad.com/library/sg90-micro-servo-9g-tower-pro-1.