*ECE 1000 Final Report: Robotic Servo Arm*

Evan Morse and Weston LaRue  
ECE 1000  
Tennessee TechCookeville TN, US  
 [emorse42@tntech.edu](mailto:emorse42@tntech.edu) [wlarue42@tntech.edu](mailto:wlarue42@tntech.edu)

***Abstract*—The use of controllable servos brings forth many advantages within many different fields. Whether it is extreme precision for medical operations, or the ability to lift substantial weights, servos and control systems can greatly increase the safety and effectiveness in jobs around the world. The creation of devices that use servos and servo control requires knowledge about .**

***Keywords—Servo, Servo Control, Mechatronics,***

# Introduction

For the ECE 1000 final project, we (Weston LaRue and Evan Morse) chose to create a servo robot arm to showcase the possibilities within the world of electrical mechatronics. Weston LaRue, freshman, and Evan Morse, Junior, are both electrical engineering majors. The importance of this project is in having a better understanding of the uses of servos, and how they can be applied to different aspects of the world. The prime reasons for choosing this project over the other options were the opportunities to learn about the programming and assembly processes required in the creation of such systems.

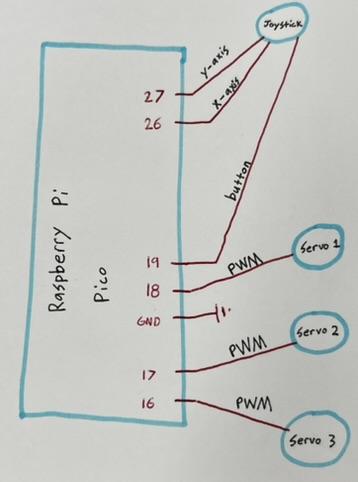
# Background

Before starting our project, we first decided that we needed to break the coding down into four sections. The joystick input, the conversion of the input to pulse width modulation (PWM), and then outputting this PWM to control the servo motors. One of the largest problems that we faced was converting the analog joystick input to a digital signal that the Pico could read. After conducting some research, we used the website “ How to connect an analog joystick to Raspberry Pi Pico” [1] to assist our coding. At this point, we realized that we needed to use a function other than GPIO in to read the joystick input.

# Project Description and formulation

The main goal of this project was to build a controllable robot arm using servo motors and a joystick. The servo motors were controlled using a Raspberry Pi Pico W. The pins that we used on the Raspberry Pi Pico are listed below:

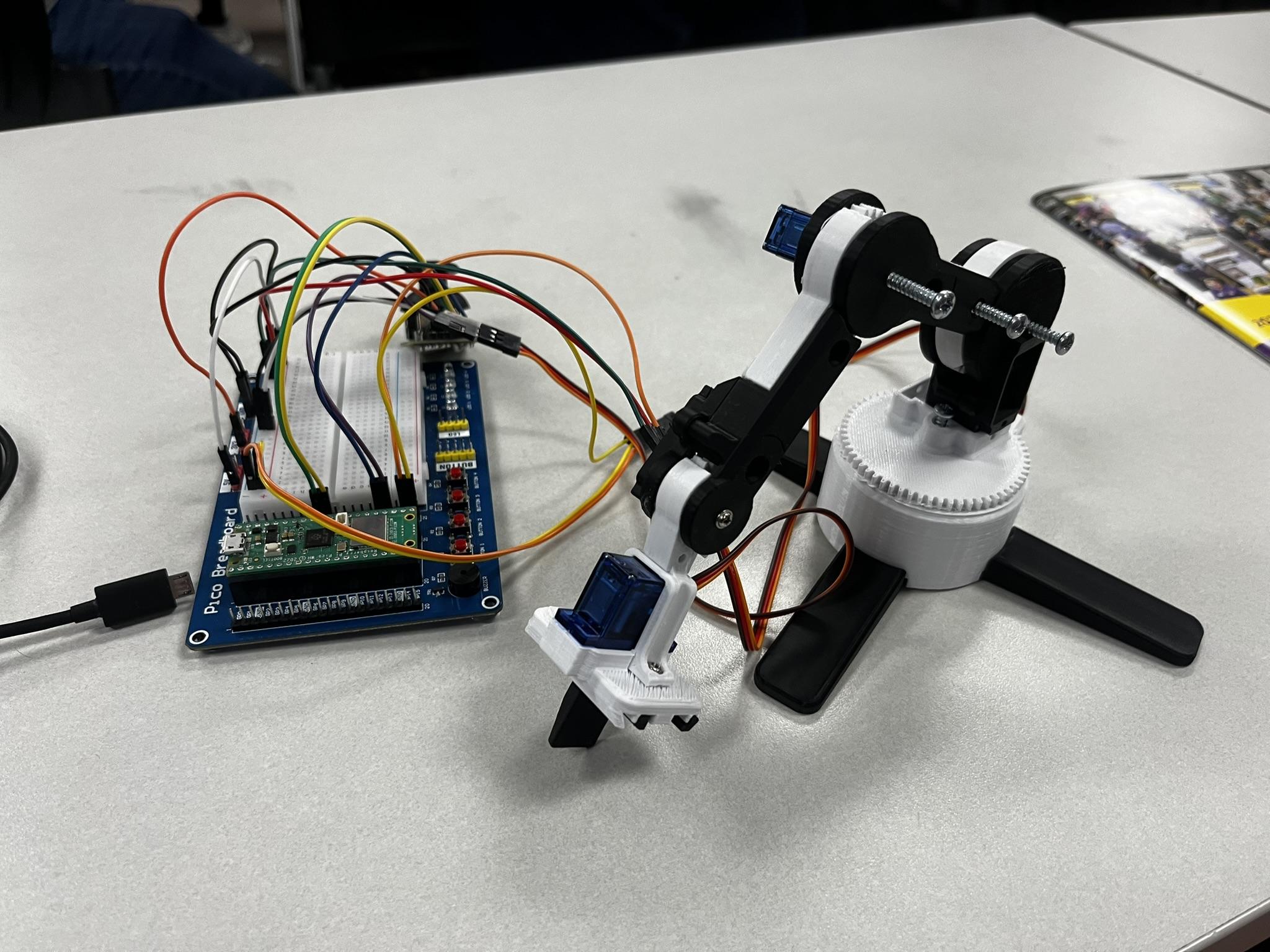
19, 26, 27: Inputs from the joystick  
 16, 17, 18: PWM outputs for the servo motors



Our code can be broken down into a few steps:

First, our code initializes the pins listed above to be Analog-to-digital inputs, GPIO inputs, or GPIO outputs. Next, the code specifies the minimum and maximum PWM values which are the limits of our servo motors’ rotations. Next our code reads the values of the joystick’s x-axis, y-axis, and whether it is being clicked or not. With these values, our code uses some math to take the inputted axis values, and convert these into PWM values.

The design we used for the robot arm came from Francis128 on Thingiverse [2]. The arm was broken into 5 segments being the base, arm L, arm M, arm S, and hand. Since we only had 3 servos to work with, we locked two of the joints in place. The other three segments were completely free to control. The segments were secured using screws to hold the joints to the gears.



# IV. Discussion and results

The results of our project were not entirely as expected. We had found a problem with our joystick not sending the proper input when commanded, sending multiple input changes seemingly at random. We also had problems with the arm design we chose to use. The servos, being 9g, were too weak to lift the weight of the arm, and had been separated from the gears. Our team would change the design of the arm, and would change the method of securing the arm together. Changing out the 9g servos for a stronger version would also be a change that we would make.

There were many enjoyable parts to this final project, primarily having to learn the programming skills required to properly use the servos and joystick. There was great satisfaction from the code working successfully, and seeing the physical outputs from the servos. The work was split evenly among our team, tackling each section of the project together.

# V. Conclusion

Understanding the importance, and functionality of servos and servo controls can be a very useful skill within a world that is increasing the applications of such technology in more aspects of daily life. Skills such as circuitry design, programming, and basic assembly are required when creating systems that utilize servos. Even with an incomplete print for our build, and lack of response, we now have a more in depth understanding of servos and their relationship with controllers and the software needed to properly utilize servos to their best. We also better understand the requirements needed to create a properly functioning servo arm.

##### References

1. [1] A. Piltch, “How to connect an analog joystick to Raspberry Pi Pico,” Tom’s Hardware, https://www.tomshardware.com/how-to/raspberry-pi-pico-joystick (accessed Apr. 26, 2024).
2. Thingiverse.com, “SG90 mini robot arm S9 by FrN (Robot Arm for everyone!) by Francis128,” *Thingiverse*. https://www.thingiverse.com/thing:2954381