ECE 1000 Final Project Report: Joystick Controlled Robot Arm

Noble Carpenter, Skyler Raines, Teagan Tobias
Electrical Engineering Department
Tennessee Tech
Cookeville, Tennessee
jncarpente42@tntech.edu, sraines42@tntech.edu, ttobias42@tntech.edu

***Abstract*—This project report covers the Joystick Controlled Robot Arm project. This project was developed based on the need for more cost-effective devices. The goal of this project is to create a fully-functioning robotic arm controlled by a small joystick.**

Keywords— Raspberry Pi Pico, Joystick, Servo, 3D-printed, (Robotic) Arm

I. INTRODUCTION

Robot arms aren't necessarily cheap, especially with human-controlled aspects. To make the model cost-effective and easy-to-build, it could feature a 3D-printed structure and servo motors, which will be presented through this project. The 3D-printed base is lightweight, as are the servos, and the motors are very precise. The arm is designed for quick assembly and intuitive control. It is controlled via a joystick, contributing to the responsive aspect of the prototype. It is designed for simplicity, affordability, and versatility, making it ideal for anyone exploring robotics and software.

II. BACKGROUND

The team followed an Instructables to complete a lot of the project [3]. We used this for our 3D-printing and it also gave us an idea of how to code the arm itself. To code the joystick, we used a website titled “Tom’s Hardware,” which we were guided on how to initialize the joystick, how to connect it, and a few basic codes [5]. To code the servo motors, we referenced another website titled “Random Nerd Tutorials” and used it for wiring the servos and basic codes for those, too [2]. In order to understand the duty and angular positions of the servo, we had to reference ChatGPT and searched for how to convert angles to duties [4]. Lastly, as a general reference guide for coding, we used a Raspberry Pi datasheet [1].

III. PROJECT DESCRIPTION AND FORMULATION

The materials used in this project include a Raspberry Pi Pico Breadboard, four servo motors, numerous jumper wires, a joystick, and a 3D-printed model. Each motor has a signal, VCC, and ground wire to it. For all four motors, the VCC is connected to the 5V pins on the breadboard and the ground wires are connected to GND pins on the Raspberry Pi. The signal wires are connected to pins GP0-GP3, respectively. The joystick has five pins to connect to the board. The first two pins on the joystick, GND and +5V, connect to the ground pin on the breadboard and the 3V3 pin on the breadboard, respectively. The last three joystick pins are VRX, VRY, and SW, which correlate to the x-axis, y-axis, and button motions of the joystick. The VRX connects to ADC pin 26, or GP26, the VRY connects to ADC pin 27, or GP27, and the SW connects to pin 28.

The code begins with importing all the necessary packages, libraries, and other items so that each component works as needed. Inside the code, each new function, instantiation, and more are commented on to direct someone on how

each line works. In order to use the servos and joysticks later in the code, they are first declared and initialized in the beginning. The servo frequencies are set and so are their starting positions. The last main part of the code shows how each motor moves according to joystick motion.

IV. DISCUSSION AND RESULTS

In the end, the joystick-controlled robotic arm ended up being a successful prototype. There are a few small things with the arm that didn't work out or go well, such as an OLED screen displaying the angles, the joystick not being able to control partial positions, and the speed of the arm. The 3D-printed model took a few tries to make in its entirety but worked out eventually.

The team enjoyed putting it all together in the end and testing it out. Noble printed the 3D-printed model. Skyler and Noble put together all of the 3D-designed pieces with screws, nuts, or just clicking in. Teagan coded the arm and wired the breadboard to the components. Altogether, we wired the robot arm to ensure our wires were long enough and not going to jumble in the arm's movements.

V. CONCLUSION

For the robotic arm, the Raspberry Pi Pico plays a vital role in controlling the robotic arm. The code that Teagan wrote for the Raspberry Pi Pico makes the robotic arm move and with it the arm is useful. The code makes sure that the robotic arm can move up, down, left, and right, as well as the claw opening and closing. These movements are controlled by a joystick.

The robotic arm is a project that is made up of Electrical Engineering and a little bit of Mechatronics Engineering. The Electrical Engineering aspect of this would include Raspberry Pi Pico, code for the Raspberry Pi Pico, servo motors, and joystick controller. The Mechatronics Engineering aspect would include 3D printing design and assembling the 3D-printed robotic arm.

This Robotic Arm would be a good cost-efficient and easy-to-do project for anyone interested.

REFERENCES

[1] Admin. Raspberry Pi Pico-Series Python Sdk: A MicroPython Environment for Raspberry Pi Microcontrollers, Jan. 2021, datasheets.raspberrypi.com/pico/raspberry-pi-pico-python-sdk.pdf.

[2] Author Unknown. "Raspberry Pi Pico: Servo Motor (MicroPython)." Random Nerd Tutorials, 23 May 2024, randomnerdtutorials.com/raspberry-pi-pico-servo-motor-micropython/.

[3] by1ronin. "DIY Robotic Arm & Snake Game with Raspberry Pi Pico Proto Shield." Instructables, Instructables, 6 Dec. 2022, www.instructables.com/DIY-Robotic-Arm-Snake-Game-With-Raspberry-Pi-Pico-/.

[4] OpenAI, "ChatGPT response to 'how to convert angle to duty of a servo motor in micropython raspberry pi'," unpublished.

[5] Piltch, Avram. "How to Connect an Analog Joystick to Raspberry Pi Pico." Tom's Hardware, Tom's Hardware, 27 Jan. 2021, www.tomshardware.com/how-to/raspberry-pi-pico-joystick.