ECE 1000 Final Project

Robotic Arm

Alek Presley, Alex Eisenbeck, Erick Ortiz, Jesse Mofield

Exploration in ECE

ECE-1000-001

Experiment Date:12/4/25

Due Date: 12/9/25

Lab Instructor: Dr Charles Van Neste

ECE 1000 Final Project

Robotic Arm

Introduction

For this project, we were tasked with designing a low cost, 3D printed robotic arm to be powered using servo motors. The three servo motors control the horizontal, vertical, and claw movements of the robotic arm. The servo motors will be controlled with a joystick from the user. The robot arm is able to rotate to the left and right, move up and down, and open and close the claw.

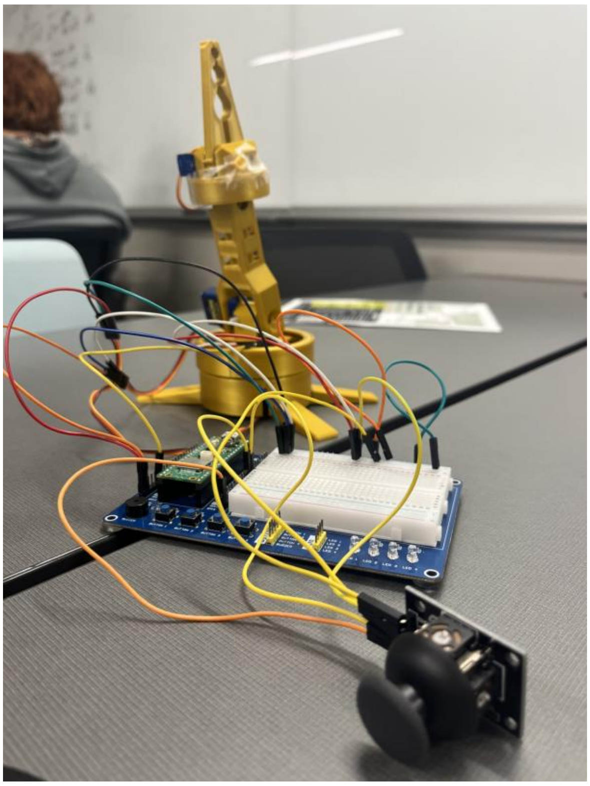
Description

Our Robotic Arm has four major components, Source Code, Circuit, Analog Joystick, and Output (Servo Motors).

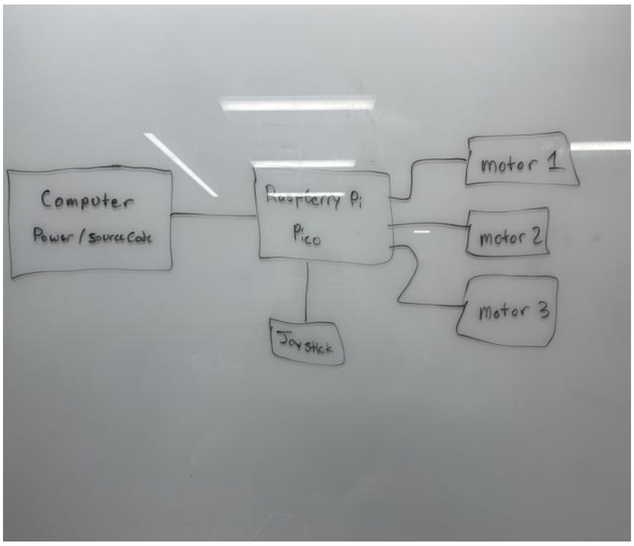
In our design, we have 3 servos that work unilaterally based on the input from the joystick. The joystick has 3 information pins, the x axis pin, the y axis pin, and the SW pin (joystick button). The x axis pin and the y axis pin input information based on location of the joystick. If the joystick moves horizontally, it changes the information output for the x axis pin. If it moves vertically, it changes the information output for the y axis pin. All of this information is implemented into the circuit, which is read by the source code.

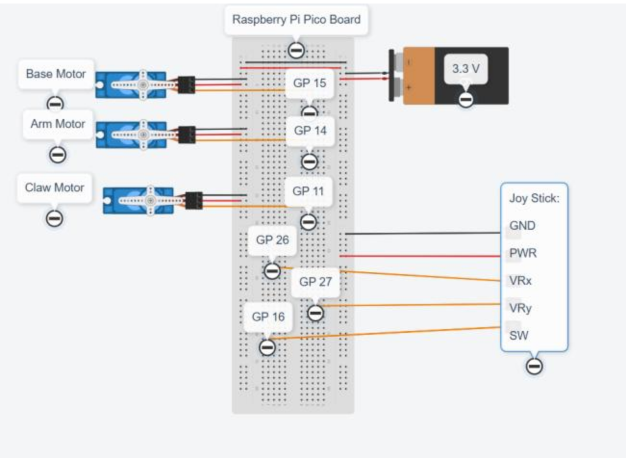
The project has 3 servo motors. Each of the motors are controlled by the designated GP pin from the analog joystick. The base motor that turns the whole arm is controlled by the x axis pin. Motor 1 which makes the arm go up and down is controlled by the y axis pin of the joystick. The Claw motor which opens and closes the claw is controlled by the SW pin or the button clicking pin.

Final Version



Circuit Diagrams





Source Code

```
# File: FinalProjectECE1000
#  Description:
#    Robot claw arm controlled by a joystick
#    - X-axis of the joystick moves the base servo moving horizontally
#    - Y-axis of the joystick moves the arm joint vertically
#    - Joystick button opens/closes the claw
# Authors: Alek Presley, Alex Eisenbeck, Erick Ortiz, Jesse Mofield
# Due Date: 12/9/2025

import machine
import time

# ==========================================================================
# Joystick axes
# ==========================================================================

x_Joystick = machine.ADC(machine.Pin(26))   # horizontal
y_Joystick = machine.ADC(machine.Pin(27))   # vertical

# Joystick button
#   Not pressed  = reads 1
#   Pressed      = reads 0

JSW = machine.Pin(16, machine.Pin.IN, machine.Pin.PULL_UP)

# Servos:
#   1) Base servo  = rotates arm left/right
#   2) Y1  =  arm joints that move up/down
#   3) Claw servo  = open/close robot claw

servoBase = machine.PWM(machine.Pin(15))   # base rotation
servoY1   = machine.PWM(machine.Pin(14))   # arm joint 1
servoClaw = machine.PWM(machine.Pin(11))   # claw

servos = [servoBase, servoY1, servoClaw]
for s in servos:
    s.freq(50)


MIN_DUTY = 1638    # 0°
MAX_DUTY = 8191    # 180°

# def angle_to_duty(angle_degrees) converts a servo angle in degrees to a 16-bit
# duty value. The function assums a linear relationshit between angle and duty
# cycles over the range 0°-180°.

# angle_degrees (float or int):
#    Desired servo angle in degrees
# returns:
#   int: corresponding 16-bit duty value for duty_u16

def angle_to_duty(angle_degrees):

    # Clamp angle to protect the servo
    if angle_degrees < 0:
        angle_degrees = 0
    elif angle_degrees > 180:
        angle_degrees = 180

    duty_span = MAX_DUTY - MIN_DUTY

    # Linear map:
    #   0°   = MIN_DUTY
    #   180° = MAX_DUTY
    duty_value = MIN_DUTY + duty_span * angle_degrees / 180.0

    return int(duty_value)

# Claw positions definitions:
CLAW_OPEN_ANGLE  = 360   # angle where claw is open
CLAW_CLOSE_ANGLE = 0     # angle where claw is closed

# Initial bas and arm joint angles (in degrees)
base_angle = 90
y_angle    = 120

# Move servos to initial positions
servoBase.duty_u16(angle_to_duty(base_angle))
servoY1.duty_u16(angle_to_duty(y_angle))
servoClaw.duty_u16(angle_to_duty(CLAW_OPEN_ANGLE))


time.sleep(0.5)  # allow servos to reach initial position

# ==========================================================================
# Main Control Loop
# ==========================================================================
while True:

    # Read Joystick Values
    x_value = x_Joystick.read_u16()
    y_value = y_Joystick.read_u16()

    # Map joystick readings to angles using a linear calibration model.
    # The constants were obtained by hand
    x_angle_raw = 0.00276 * x_value - 0.75
    y_angle_raw = 0.00276 * y_value - 0.75

    # Convert to integers
    x_angle = int(x_angle_raw)
    y_angle = int(y_angle_raw)


    # Clamp to [0, 180] range
    if x_angle < 0:
        x_angle = 0
    elif x_angle > 180:
        x_angle = 180

    if y_angle < 0:
        y_angle = 0
    elif y_angle > 180:
        y_angle = 180

# ==========================================================================
# Update Servo Positions
# ==========================================================================

    # Base rotation controlled by joystick X-axis.
    servoBase.duty_u16(angle_to_duty(x_angle))

    # Base rotation controlled by joystick Y-axis.
    servoY1.duty_u16(angle_to_duty(y_angle))

# ==========================================================================
# Claw control via joystick button
# ==========================================================================
    # JSW is active low:
    #   JSW.value() == 0  => button pressed  = close claw
    #   JSW.value() == 1  => button released = open claw
    if JSW.value() == 0:
        # Button pressed = CLOSE claw
        servoClaw.duty_u16(angle_to_duty(CLAW_CLOSE_ANGLE))
    else:
        # Button released = OPEN claw
        servoClaw.duty_u16(angle_to_duty(CLAW_OPEN_ANGLE))

# ==========================================================================
```

Conclusion

Our group successfully designed a robotic arm with the specified requirements. The robotic arm we designed had three working servo motors to control horizontal, vertical, and claw movement. We faced some unexpected setbacks with our arm because of the chassis design that we chose for the 3D print. We initially tried using all five motors and using the full chassis design, but we faced some difficulties writing code for five servo motors as opposed to three. In addition, the base servo motor could not handle the weight of the full arm moving vertically up and down. We downsized our design and took off an arm segment and two servo motors, leaving our design with the base, arm, and claw components. This project took a group effort from start to finish. Once the chassis was 3D printed, Jesse took charge by connecting all the printed parts together and screwing in the servo motors. Alex contributed towards the circuit design by making the block diagram and the tinker cad schematic. Alek contributed by implementing the circuit and handling the Raspberry Pico board during testing. Erick was the backbone of this group because he handled all the coding aspects of this project. Everyone in our group made contributions towards every aspect of this project, however these were the main contributions from each member.

Estimated Cost

- PicoBoard: $20
- Male/Female Wires: $3 to $7
- Joystick: $4 to $6
- Servo Motors: $7
- 3D printing filament: $4
- Screws: $2

References

[1]“Pico Breadboard Kit,” *PiShop.us*, 2025. https://www.pishop.us/product/pico-breadboard-kit/?srsltid=AfmBOopmSGh5aoyNRMxuZOEEk81bw5hHybpTZ6vO7bA7uj8nL0cX1d9sQaQ (accessed Dec. 08, 2025).

[2]“120Pcs Connect Components 10cm Male to Female Jumper Wires for Arduino Breadboard Cord,” *Walmart.com*, 2025. https://www.walmart.com/ip/120Pcs-Connect-Components-10cm-Male-to-Female-Jumper-Wires-for-Arduino-Breadboard-Cord/5915628952?wmlspartner=wlpa&selectedSellerId=101654882 (accessed Dec. 08, 2025).

[3]“Thumb Joystick,” *Sparkfun.com*, 2025. https://www.sparkfun.com/thumb-joystick.html (accessed Dec. 08, 2025).

[4]“Amazon.com: WWZMDiB SG90 Micro Servo Motor for Arduino Raspberry Pi DIY (3 Pcs) : Toys & Games,” *Amazon.com*, 2025. https://www.amazon.com/WWZMDiB-SG90-Control-Servos-Arduino/dp/B0BKPL2Y21?th=1 (accessed Dec. 08, 2025).

[5]OpenAI, “ChatGPT,” *ChatGPT*, 2025. https://chatgpt.com/