



Inspiring Excellence

Lab report 3 of CSE461

Submitted By:

Group 4

Name	ID
Tahmid Chowdhury	19201115
Tabassum Nusrat Jahan	19201027
Sadman Sakib Nahid	19201029
Mahfuza Sultana Mim	18101703
Fairuz Anika	20301464

1.1 Name of the experiment : Introducing servo motor with Raspberry Pi

1.2 Objective: Control the position accurately and efficiently of a servo motor using Raspberry Pi, duty cycle and working mechanism.

1.3 Equipment:

- Raspberry Pi 4
- Connecting wires (female to male or male to male)
- Breadboard
- Servo motor

1.4 Experimental Setup: In this experimental setup firstly I turned on our raspberry pi and with the help of jumper wires I took 5V power and ground from the raspberry pi and connected it to our bread board. Then with the help of jumper wires again I connected the 5V with the red wire of our servo motor and also connected ground with the brown one of Servo motor. Then we connected the signal that means yellow wire of the servo motor with the GPIO 18 pin of our Raspberry-pi and after running the code our servo motor started to rotate.

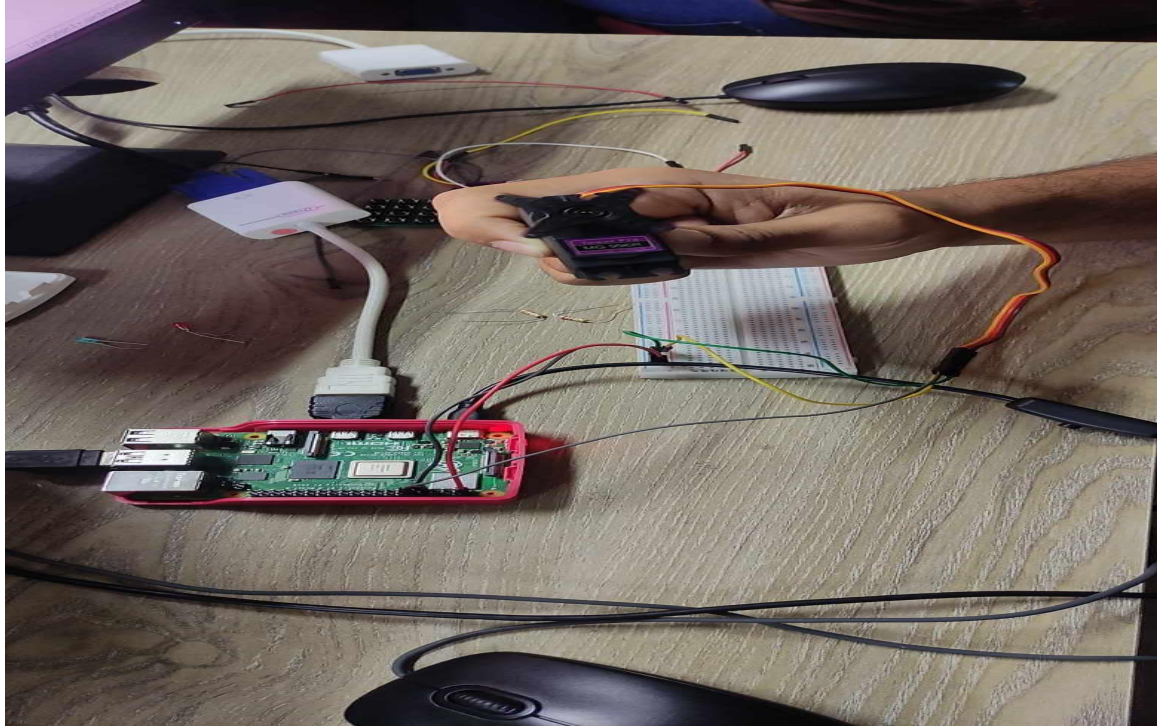


Fig1: Rotating Servo motor

1.5 Result : By employing Python's RPi.GPIO library the experiment managed to effectively link a servo motor to the Raspberry Pi. Through this connection accurate control over the motors' position was achieved using PWM signals. This integration presents numerous possibilities for projects involving motorized movements and automations.

1.6 Code:

```
from gpiozero.pins.pigpio import PiGPIOFactory
from gpiozero import Device, Servo, AngularServo
from time import sleep

Device.pin_factory = PiGPIOFactory()
s = AngularServo(18, min_angle = 0, max_angle =
180, min_pulse_width=0.5/1000, max_pulse_width = 25/10000)
while True:
    s.angle=120# (120 degree to the left)
    sleep(1)
    #right
    s.angle=60 # 60 degree to the right
    sleep(1)
```

2.1 Name of the experiment : Introducing servo motor using push buttons and LEDs with Raspberry Pi.

2.2 Objective: Create an experiment using a servo motor, push buttons, and LEDs with Raspberry Pi to control the motor's position and indicate its status.

2.3 Equipment:

- Raspberry Pi 4
- Servo Motor
- LED
- 2 resistor of 220 ohms
- Push-button
- Connecting wires (female to male)
- Bread Board

Circuit diagram of the setup:

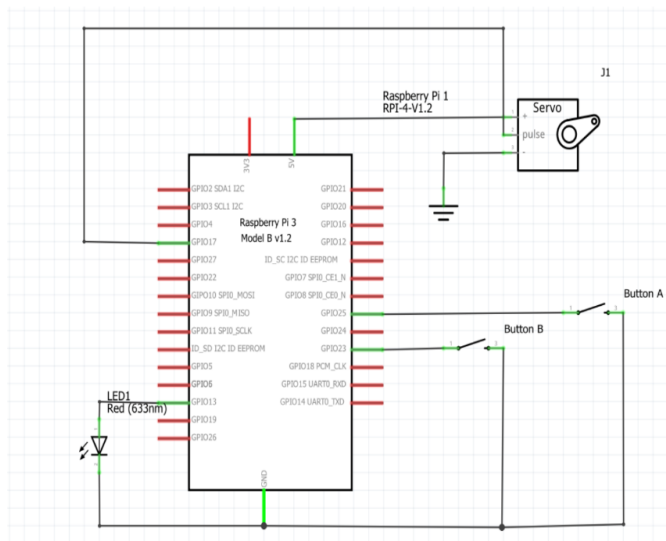


Fig: Circuit diagram of the setup (task 2)

2.4 Experimental Setup: In this 2nd experiment we tried to rotate the servo motor in different angles with the 2 switches of a push button and also light a LED. To start the experiment firstly

we give the 5V power, Ground and Signal to the servo motor with the help of jumper wires from Raspberry-pi. After that, with GPIO 27 of Raspberry-pi we connected R1 and with ground we connected the C1 of a switch of Push button. Then we also connected GPIO 22 of Raspberry-pi with R4 and with ground we connected the C4 of a switch of Push button. Moreover, with GPIO 4 of Raspberry-pi we connected a 220 ohm resistor to a LED and connected the other side with ground also. After doing all of this work, when we pressed the 1st switch of the push button the LED lit up and the servo motor rotated 120 degrees to the left. Then we pressed the 16th switch of a push button and again the LED lit up, also the servo motor rotated 60 degrees to the right.

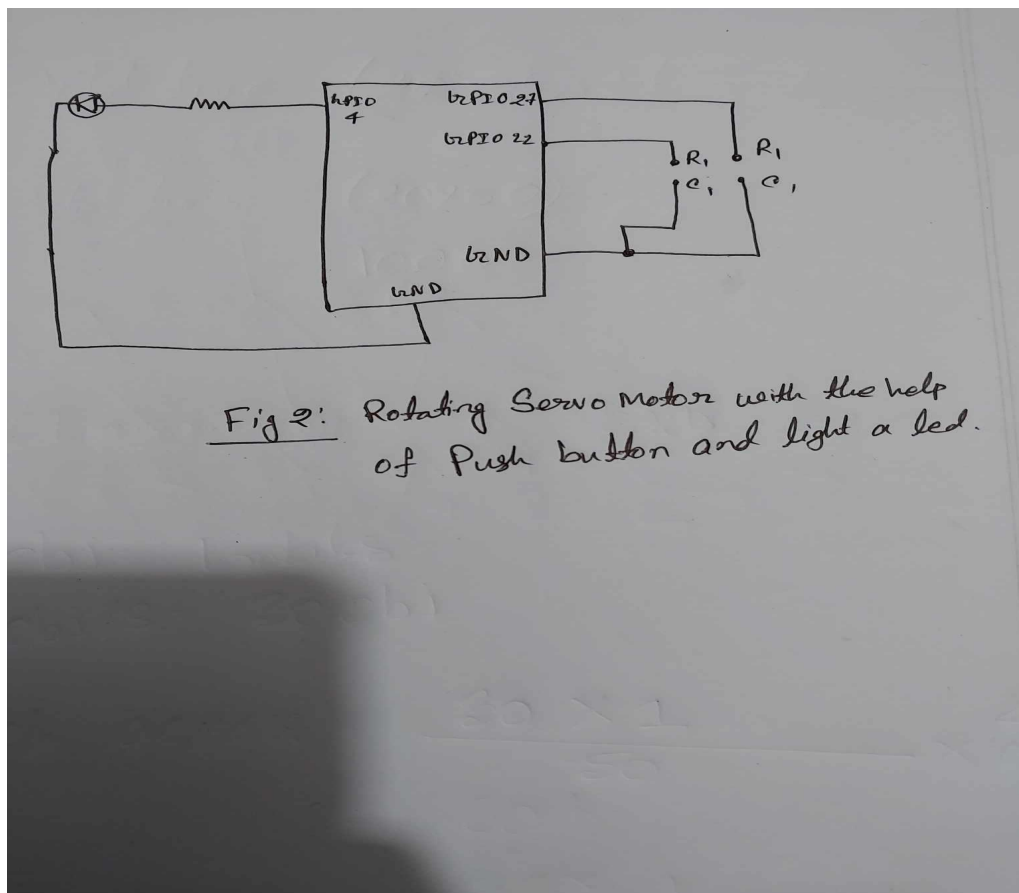


Fig 2: Circuit diagram of Rotating Servo Motor with the help of Push Button and light a LED

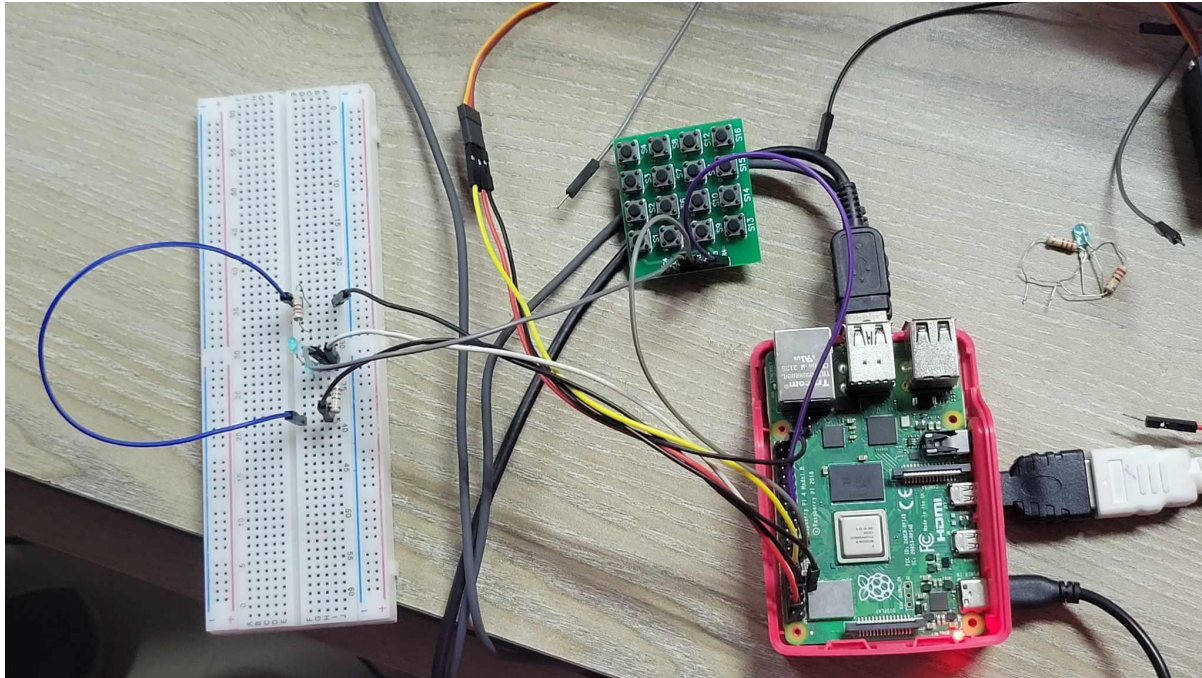


Fig 3: Rotating Servo Motor with the help of Push Button and light a LED

2.5 Result : The experiment successfully integrated a servo motor, push buttons, and LEDs with the Raspberry Pi. Python's RPi.GPIO library played a pivotal role in controlling button inputs for determining servo motor positions as well as providing visual feedback via LED lights. By exemplifying this interactive setup with utmost clarity and efficiency, we emphasized how Raspberry Pi has heightened capabilities in controlling servos alongside driving interaction through well-designed features like push buttons and LED indicators.

2.6 Code:

```
from gpiozero.pins.pigpio import PiGPIOFactory
from gpiozero import Device, LED, Button, Servo, AngularServo
from time import sleep
button1= Button(27) #pin 13
button2= Button(22) #pin 15
led=LED(4) #pin 7

Device.pin_factory = PiGPIOFactory()

s = AngularServo(18,min_angle = 0, max_angle =
180,min_pulse_width=0.5/1000,max_pulse_width = 25/10000)

while True:
    button1.wait_for_press()
    s.angle=120# (120 degree to the left)
    led.on()
    sleep(1)
    led.off()
    #right
    button2.wait_for_press()
    s.angle=60 # 60 degree to the right
    led.on()
    sleep(1)
    led.off()
```


3.1 Name of the experiment: Introducing servo motor using push buttons and LEDs with Raspberry Pi where task is to update the circuit and modify the code in such a way that we can implement following task:

- Implement 2 LED
- Press Button_A and one LED will turn on and servo motor will rotate for a specific angle
- Press Button_B and another LED will turn on and servo motor will rotate for a specific angle

3.2 Objective: Create a project using a servo motor, push buttons, and LEDs with Raspberry Pi to control the motor's position and indicate its status. Specifically, pressing Button A will activate one LED and rotate the servo motor to a specific angle, and pressing Button B will activate another LED and rotate the servo motor to a different specific angle.

3.3 Equipment:

- Raspberry Pi 4
- Servo Motor
- 2 LED
- 2 resistor of 220 ohms
- Push-button
- Connecting wires (female to male)
- Bread Board

3.4 Experimental Setup: In this 3rd experiment we tried to rotate the servo motor in different angles with the 2 switches of a push button and also light two LEDs (Red and Blue). To start the

experiment firstly we give the 5V power, Ground and Signal to the servo motor with the help of jumper wires from Raspberry-pi. After that, with GPIO 27 of Raspberry-pi we connected R1 and with ground we connected the C1 of a switch of Push button. Then we also connected GPIO 22 of Raspberry-pi with R4 and with ground we connected the C4 of a switch of Push button. In addition, with GPIO 12 of Raspberry-pi we connected a 220 ohm resistor to a LED(Red) and connected the other side with ground also. Moreover, with GPIO 4 of Raspberry-pi we connected a 220 ohm resistor to a LED(Blue) and connected the other side with ground also. After doing all of this work, when we pressed the 1st switch of the push button the Red LED lit up and the servo motor rotated to the left. Then we pressed the 16th switch of a push button and again the Blue LED lit up, also the servo motor rotated to the right.

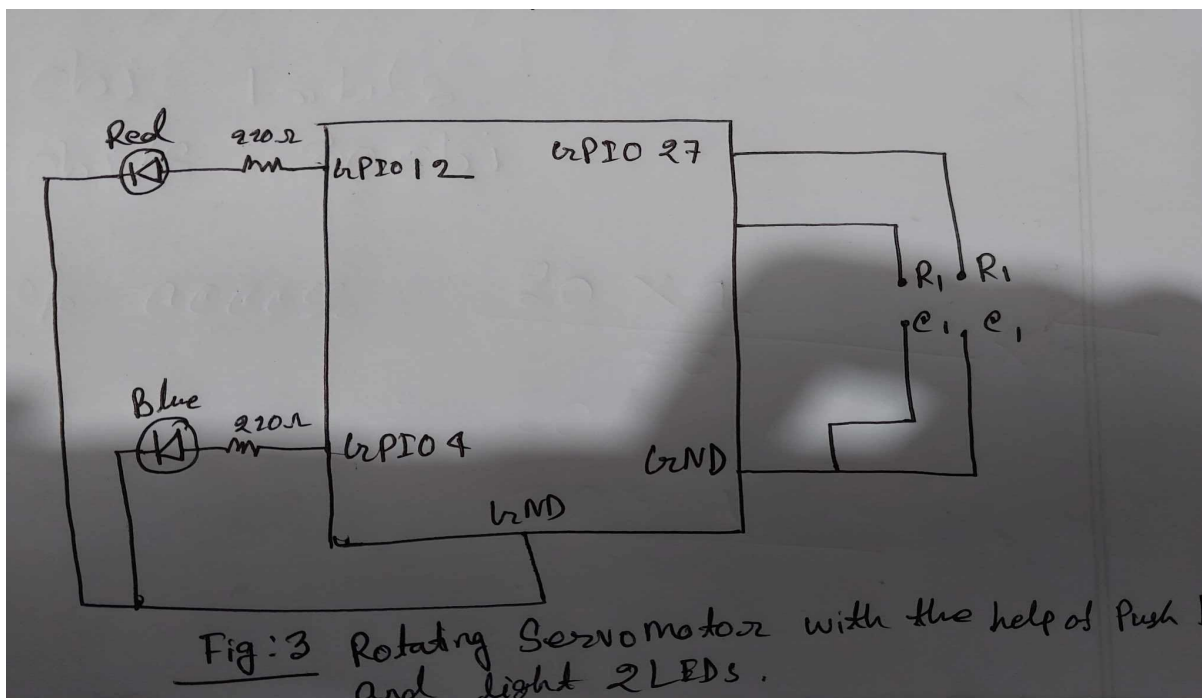


Fig 4: Circuit diagram Rotating Servo Motor with the help of Push Button and Light 2 LEDs(Red and Blue)

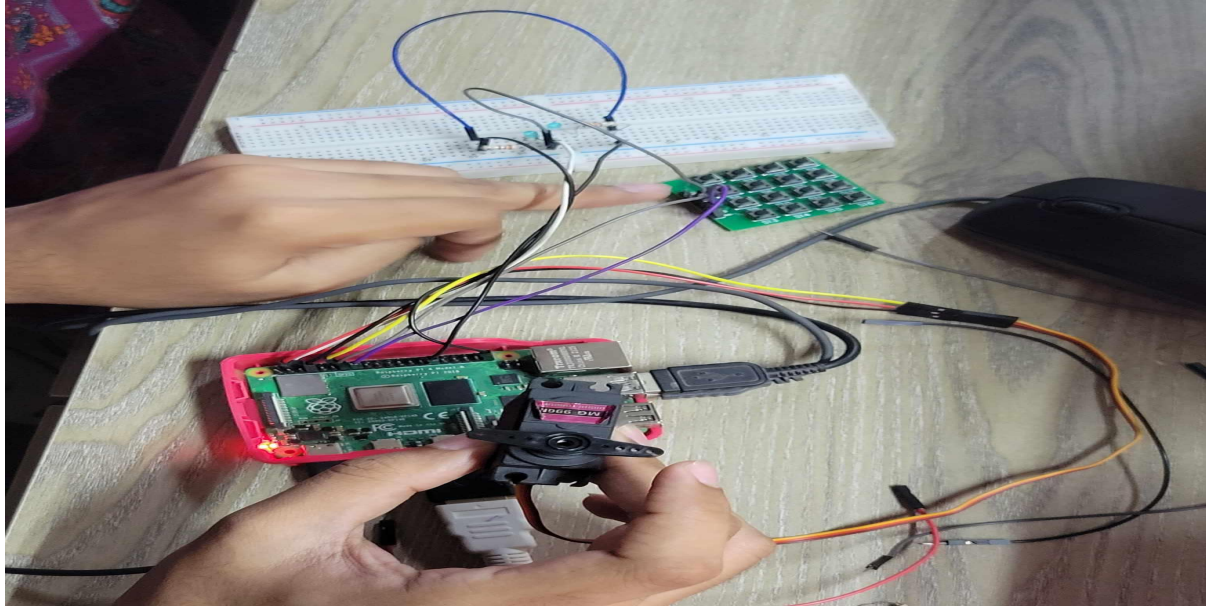


Fig 4:Rotating Servo Motor with the help of Push Button and Light 2 LEDs(Red and Blue)

3.5 Result : The circuit using Raspberry Pi push buttons, LEDs, and a servo motor was successfully implemented in this experiment. By pressing Button A one LED would turn on and the servo motor would rotate to a specific angle. Similarly pressing Button B would activate another LED and rotate the servo motor to a different specific angle. This project effectively accomplished its goal of controlling the position of the motor and providing indications of its status.

3.6 Code:

```
import RPi.GPIO as GPIO
from time import sleep
from gpiozero import Button
from gpiozero import LED
button1= Button(27)
button2= Button(22)
led1=LED(4)

GPIO.setmode(GPIO.BCM)
GPIO.setup(18,GPIO.OUT)
pwm=GPIO.PWM(18,50)
pwm.start(7) #center(90 degrees)
```

while True:

```
    button1.wait_for_press()
    pwm.ChangeDutyCycle(5.3) # (left)
    led1.on()
    sleep(1)
    led1.off()
```

```
    button2.wait_for_press()
    pwm.ChangeDutyCycle(8.7) # right
    led1.on()
    sleep(1)
    led1.off()
```

Discussion : Push buttons and LEDs were successfully used in the experiment to demonstrate how to control a servo motor. We were able to control the Raspberry Pi's position depending on the status of the push buttons by connecting it to the servo motor. The LEDs were utilized to deliver visual input, indicating the system's present condition. Push buttons and LEDs were incorporated into the project to give it an interactive component. To control the servo motor, we could push the buttons, and the LED indicators would show the corresponding movement. This improved the system's use and engagement by creating a user-friendly interface. The Raspberry Pi's adaptability and programming simplicity were two of its primary benefits in this experiment.