BRAC
UNIVERSITY

Inspiring Excellence

Group Number: 07

Name	ID
Safwat Sufia Raida	19101293
Shoeb Islam Hamim	20101337
F.M. Ashfaq	20101327
MD Fuad Hasan	20101345

Submitted by: *Safwat Sufia Raida*

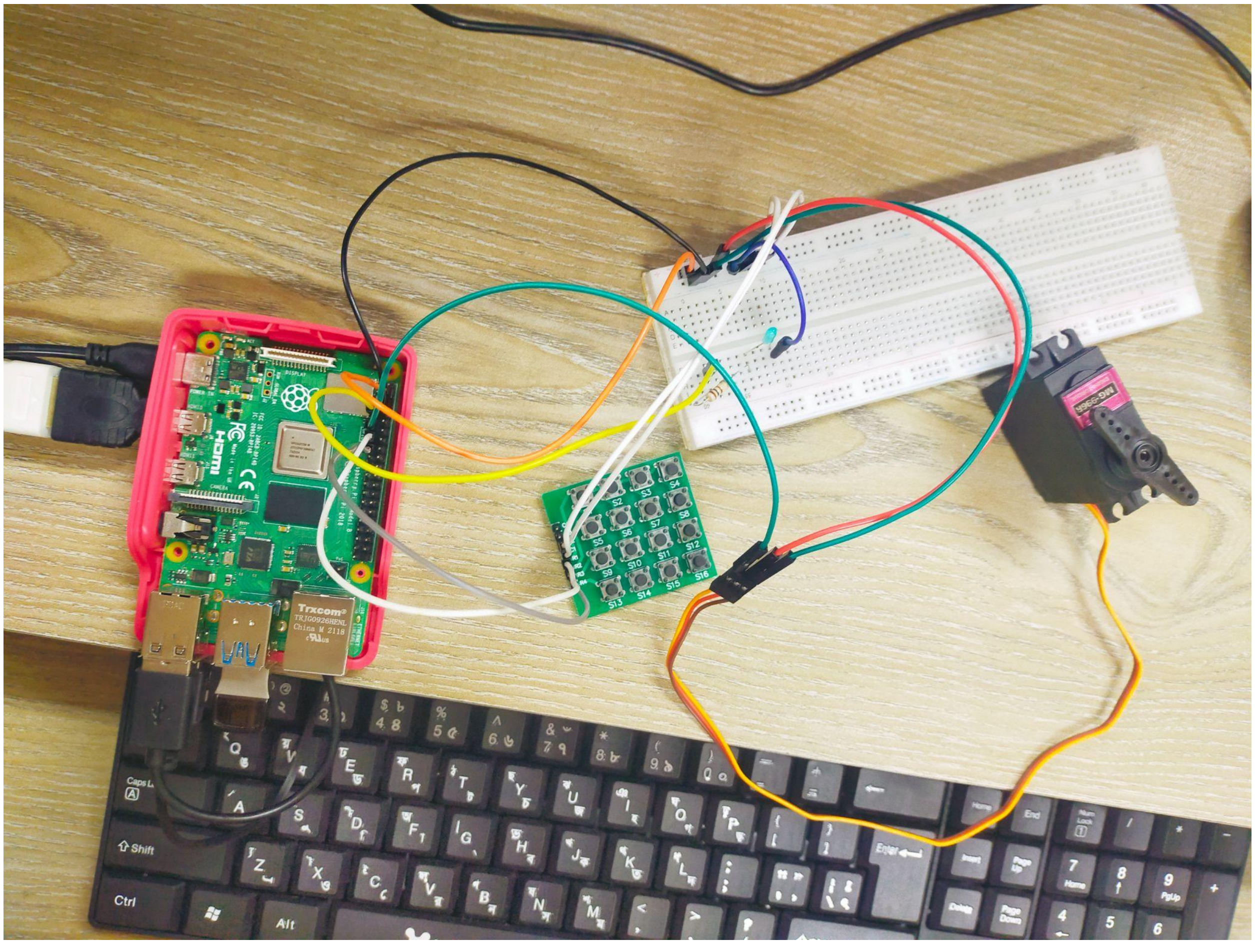
Name of experiment: Introducing servo motor using along with push buttons and LEDs
with Raspberry Pi.

Objective: The objective of this experiment is to to get the basic idea on servo motors about how they function and how they can be controlled using a Raspberry Pi. Moreover, this experiment gives us hands-on experience of working with servo motors and understanding their functionality, as well as how they can be integrated with other components such as push buttons and LEDs to create a simple interactive system.

Equipment: The following components were used for this experiment,

- Raspberry Pi 4
- LED
- Servo Motor
- Push-button
- Breadboard
- Connecting wires

Experimental Setup: The LED was first connected to the breadboard, and then a resistor was added in series with the LED. The GPIO4 pin was then connected to the resistor's other end. The LED's negative pin was then linked to the GPIO13 pin and the other side was connected to the GND. Next, we connected push-button A to GPIO25 and push-button B to GPIO23, the other end of both the push buttons was connected to the breadboard where there is connection from GND of Raspberry Pi. Finally, we connected the pulse of the servo motor to the GPIO17 pin, the power to the breadboard where there is 5V from Raspberry Pi as well as the ground of the servo motor to the breadboard where there is connection from GND of Raspberry Pi.



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(17,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()
```

Code with duty cycle control:

```
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(17,GPIO.OUT)
pwm=GPIO.PWM(17,50)
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()
```

Results: After connecting the wires, the servo motor, the resistor, the breadboard, the led, the switch and Raspberry Pi, when the code is executed, and then when we press the first button the servo motor moves to the left and the LED turns on and after some time the LED turns off. Next, when we press the second button, the motor moves itself to the right. The program also turns on the LED when the second button is pressed and after some time it turns the LED off. Thus, it demonstrates that the experiment was successful.

Discussion: We encountered a number of challenges in our experiment, which prevented us from finishing it on time. After spending some time troubleshooting it, we found that the issue was caused by a defective servo motor and several damaged wiring. But, given the time constraints, we were unable to achieve the expected results.

In conclusion, this experiment demonstrated how to use a Raspberry Pi to control a servo motor, push buttons, and LEDs. By connecting the servo motor to the Raspberry Pi, and the python program we wrote, we were able to control its position using push buttons, as well as turning on and off LEDs in response to button presses. This experiment helped to demonstrate the basic principles of servo motors and how they can be controlled using a microcontroller like the Raspberry Pi. It also demonstrated the importance of proper wiring and the potential for faulty components to cause issues in the system. Despite facing some difficulties during the experiment, such as a faulty servo motor and faulty wires, we were able to troubleshoot the issues and broaden our understanding.