PROGRAM 5

Board & Environment Setup (Raspberry Pi 4)

Board: Raspberry Pi 4
OS: Raspberry Pi OS
Language: Python 3 (ONLY)

Required Libraries / Packages (Install BEFORE running program)

```
# Update system (recommended)
sudo apt update

# Install GPIO library for Python 3 (hardware dependent)
sudo apt install python3-rpi.gpio -y
```

> **Note:** `RPi.GPIO` must be installed using **apt**, not pip, because it is hardware-dependent.

Program Title

Servo Motor Control using Raspberry Pi with User Input

Aim

To control the angular position of a servo motor using Raspberry Pi and Python based on user-provided angle input.

Components Used

- Raspberry Pi 4
- Servo motor (SG90 or equivalent)
- External 5V power supply (recommended)
- Jumper wires

Pin Configuration

Servo Motor Wiring

Servo Wire	Raspberry Pi Physical Pin	GPIO
Red (VCC)	External 5V supply	—
Brown / Black (GND)	Pin 6	GND
Yellow / Orange (Signal)	Pin 12	GPIO18

Important Notes: - GPIO18 is used because it supports PWM. - Ground of the external power supply and Raspberry Pi must be common.

Pre-Code Concept

A servo motor is controlled using a PWM (Pulse Width Modulation) signal. The angular position of the servo depends on the duty cycle of the PWM signal. By varying the duty cycle between minimum and maximum values, the servo can be rotated between 0° and 180°.

Working / Logic Explanation

1. Initialize GPIO in BCM mode.
2. Configure GPIO18 as PWM output at 50 Hz.
3. Accept angle input from the user.
4. Convert the angle into a corresponding duty cycle.
5. Rotate the servo to the desired angle.
6. Exit the program when the user enters 'q'.
7. Clean up GPIO resources before termination.

Program Code (Python 3 – Final Version)

```
# --------------------------------------------------
# PROGRAM 5: Servo Motor Control using Raspberry Pi
# User Input Based
# Platform : Raspberry Pi 4
# Language : Python 3
# Library  : RPi.GPIO
# PWM Pin  : GPIO18
# --------------------------------------------------

import RPi.GPIO as GPIO
import time

# Set GPIO mode to BCM
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)

# Define servo control pin
SERVO_PIN = 18

# Set the pin as output
GPIO.setup(SERVO_PIN, GPIO.OUT)

# Create PWM object at 50 Hz (standard for servo motors)
pwm = GPIO.PWM(SERVO_PIN, 50)

# Start PWM with 0 duty cycle
pwm.start(0)

# Function to rotate servo to a specific angle
def set_angle(angle):
    min_duty = 2.5
    max_duty = 12.5

    # Convert angle (0-180) to duty cycle
    duty = min_duty + (max_duty - min_duty) * (angle / 180)

    pwm.ChangeDutyCycle(duty)
    time.sleep(1)            # Allow servo to reach position
    pwm.ChangeDutyCycle(0)   # Stop jitter

try:
    while True:
        user_input = input("Enter angle (0-180) or 'q' to quit: ")

        # Exit condition
        if user_input.lower() == 'q':
            print("Exiting servo control program.")
            break

        try:
            angle = float(user_input)

            if angle < 0 or angle > 180:
                print("Invalid angle! Enter a value between 0 and 180.")
            else:
                print(f"Rotating servo to {angle} degrees")
                set_angle(angle)
        except ValueError:
            print("Invalid input! Enter a number or 'q'.")

finally:
    # Stop PWM and clean up GPIO
    pwm.stop()
    GPIO.cleanup()
    print("GPIO cleaned up. Program ended.")
```

Output / Result

- The servo motor rotates to the angle entered by the user.
- The program terminates safely when the user enters 'q'.

Possible Viva / Exam Questions

1. Why is PWM required for servo motor control?
2. What is duty cycle?
3. Why is GPIO18 preferred for servo control?
4. Why is external power recommended for the servo motor?
5. What is the purpose of GPIO cleanup?

Conclusion

The program successfully demonstrates user-controlled actuation using Raspberry Pi by generating PWM signals to control the angular position of a servo motor.