



Inspiring Excellence

## CSE461

Introduction to Robotics Lab

Lab No. : 05

Group : 03

Section : 08

Semester : Summer\_2025

Group members :

MD.Sohanur Rahman Shimul	22299079
Fayez Ahmad Protik	23101474
Mahibi Islam	22201828
Tithi Halder	22101406
Sharmin Jahan Ananna	22101850

Submitted Date: 23-08-2025

Submitted to -

Utsha Kumar Roy & Sadman Sharif

- 1. Objectives:** Learn to interface a servo motor with a Raspberry Pi, control its position using PWM and GPIO programming, and build a responsive automated system that reacts to real-time sensor data like ultrasonic input.

- 2. Equipments:**

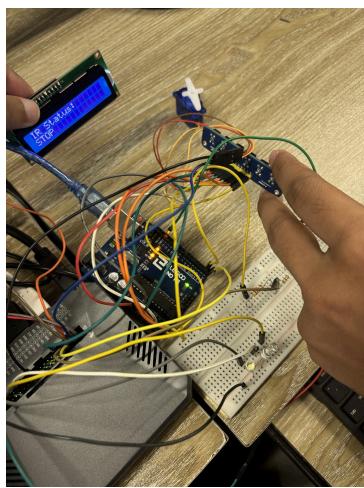
- Raspberry PI development board
- Servo Motor
- Ultrasonic Sensor(HC-SR04)
- Resistor (appropriate value for current limiting, typically around 220 ohms)
- Jumper wires
- Breadboard
- USB pen drive containing OS

- 3. Experimental Setup:**

**Explanation:**

Here used an ultrasonic sensor to detect the distance of an approaching object. The Raspberry Pi reads the sensor data and sends a PWM signal to a servo motor depending on how close the object is. If the object is less than 5 cm away, the servo rotates to 180° to open the gate; otherwise, it stays at 90° to keep the gate closed. For the connections, I wired the servo motor with its red wire to the 5V pin, brown to GND, and orange to GPIO17 for the PWM signal. The ultrasonic sensor was connected with VCC to 5V, GND to ground, TRIG to a GPIO pin defined in the code, and ECHO to another GPIO pin with a voltage divider if needed.

**Picture:**



**4. Code: (If Applicable)**

```
# Enter Code Here

import serial
from rpi_lcd import LCD
from time import sleep
import RPi.GPIO as GPIO
from gpiozero.pins.pigpio import PiGPIOFactory
from gpiozero import Device, Servo, AngularServo
from time import sleep

Device.pin_factory = PiGPIOFactory()

s = AngularServo(17, min_angle = 0, max_angle = 180, min_pulse_width=0.5/1000,
max_pulse_width = 25/10000)

GPIO.setmode(GPIO.BCM)
#Initialize LCD

# Set up serial connection
ser = serial.Serial('/dev/ttyACM1', 9600, timeout=1)

try:
    while True:

        if ser.in_waiting > 0:
            # Read and decode the line from Arduino
            line = ser.readline().decode('utf-8', errors='ignore').strip()
            if line == "FORWARD":
```

```
s.angle=120# (120 degree to the left)
elif line == "RIGHT":
    s.angle=60 # (60 degree to the right)
elif line == "LEFT":
    s.angle = 140
elif line == "STOP" or line == "UNCERTAIN":
    s.angle = 40
except KeyboardInterrupt:
    lcd.clear()
    ser.close()
    print("Program terminated")
```

## 5. Results (Output of the experiment):

Whenever an object came closer than 5 cm, the servo motor rotated to 180°, which represented the gate opening. If no object was detected within that range, the servo remained at 90°, keeping the gate closed. This setup effectively showed how an automated parking gate system can work based on real-time distance sensing.

## 6. Discussions/Answers:

Learned how to control a servo motor using PWM and adjust its duty cycle to get accurate angles. Also, explored how to measure distance with the HC-SR04 ultrasonic sensor. By combining both components, it is able to create a system that responds automatically to changes in its surroundings. At one point, it was noticeable that the sensor was giving inconsistent readings, likely due to interference, but fixed it by adding short delays between measurements and making sure all the wiring was secure. Overall, this experiment helped to understand the basics of building automated systems that react to real-time environmental input.