CSE461

Introduction to Robotics Lab

Lab No. : 04
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:**

- Establish communication between Arduino and Raspberry Pi using the UART protocol.
- Create level-shifting circuits to ensure compatibility between 3.3V (Raspberry Pi) and 5V (Arduino) devices.
- Write Python and Arduino programs to send real-time sensor readings through UART.
- Present the transmitted sensor data on an LCD screen using the I2C interface.
- Perform testing and debugging of hardware and software to resolve issues in UART and I2C communication.

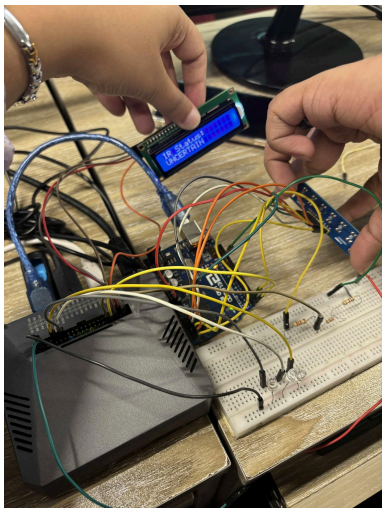
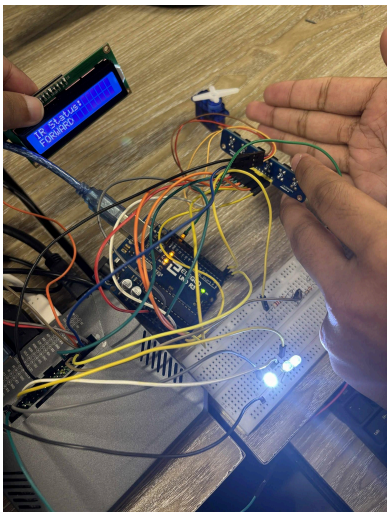
2. **Equipments:**
   - Arduino Uno R3
   - Raspberry PI 4
   - Analog 6-array IR sensor
   - I2C LCD Display (16x2)
   - Resistor (220Ω)
   - Breadboard
   - Jumper wires

3. **Experimental Setup:**

Explanation:

The experimental setup consists of an Arduino Uno connected to a 6-channel IR sensor via analog inputs (A0–A5). The Arduino transmits the sensor readings to a Raspberry Pi 4 through UART. A voltage divider is implemented on the RX line to safely match the 5V output of the Arduino with the 3.3V input of the Raspberry Pi. The Raspberry Pi interprets the received signals (such as LEFT, RIGHT, FORWARD, STOP, etc.) and displays them on a 16x2 I2C LCD. Each module is powered by its respective board, with all communication lines properly connected and a shared ground between the devices.

Picture:

4. Code: (If Applicable)

```
# Part 1
import serial
from rpi_lcd import LCD
from time import sleep
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
# Initialize LCD
lcd = LCD()
LEFT=17
FORWARD=27
RIGHT=22

GPIO.setup(LEFT,GPIO.OUT)
GPIO.setup(RIGHT,GPIO.OUT)
GPIO.setup(FORWARD,GPIO.OUT)
# Set up serial connection
ser = serial.Serial('/dev/ttyACM0', 9600, timeout=1)
def  all_off():
    GPIO.output(LEFT,GPIO.LOW)
    GPIO.output(RIGHT,GPIO.LOW)
    GPIO.output(FORWARD,GPIO.LOW)


try:
    while True:
        all_off()
        if ser.in_waiting > 0:
            # Read and decode the line from Arduino
            line = ser.readline().decode('utf-8', errors='ignore').strip()
            if line =="FORWARD":
                GPIO.output(FORWARD,GPIO.HIGH)
            elif line == "RIGHT":
```

```
                GPIO.output(RIGHT,GPIO.HIGH)
            elif line == "LEFT":
                GPIO.output(LEFT,GPIO.HIGH)
            elif line == "STOP" or line =="UNCERTAIN":
                GPIO.output(LEFT,GPIO.LOW)
                GPIO.output(RIGHT,GPIO.LOW)
                GPIO.output(FORWARD,GPIO.LOW)
            # Clear LCD and display the received message
            lcd.clear()
            lcd.text("IR Status: ", 1)
            lcd.text(line, 2)

except KeyboardInterrupt:
    lcd.clear()
    ser.close()
    print("Program terminated")
```

```
# Part 2
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")
```

Photo:

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

- The 6-channel IR sensor effectively identified surface differences (black/white).
- The Arduino interpreted the sensor readings and generated direction commands: FORWARD, LEFT, RIGHT, or STOP.
- These commands were transmitted to the Raspberry Pi, which displayed them in real time on a 16x2 I2C LCD.
- The system demonstrated real-time responsiveness with continuous updates on the display.
- UART communication was reliable, with no loss or misinterpretation of data.

6. **Discussions/Answers:**

● Learning : This lab demonstrated the importance of voltage compatibility when interfacing different microcontrollers. Students gained practical experience in UART serial communication, I2C-based peripheral handling, and real-time data processing.

● Challenges Faced: Raspberry Pi not detecting the LCD initially: resolved by checking wire connections and re-running i2cdetect.

● Solutions : Ensured both systems share a common ground.