EEL 7170 : Introduction to IoT

Lab Report

॥ त्वं ज्ञानमयो विज्ञानमयोऽसि ॥

Name:	**Anushkaa Ambuj**
Roll Number:	**B21ES006**
Program:	**B.Tech in ES**

Lab 3: Setting up Raspberry Pi 4 and enable GPIO pin interfaces

23 Sep, 2024

1 InLab

1.1 Objective

1. To interface `VL53L0X Lidar sensor` and `OLED display` with the `Raspberry Pi 4`.
2. To learn how to utilize the Raspberry Pi GPIO pins.
3. To perform basic sensor interfacing using I2C communication.
4. To create a Python virtual environment for IoT applications.

1.2 Components Used

- Raspberry Pi 4
- VL53L0X Lidar Sensor
- OLED Display (128x64 resolution)
- Jumper wires (female to female)
- Breadboard (optional)
- MicroSD Card with Raspberry Pi OS
- USB Power Supply for Raspberry Pi
- Monitor, Keyboard, and Mouse for Raspberry Pi setup
- Python with required libraries (specified in `requirements.txt`)

1.3 Procedure & Observations

Part 1: Setting Up Raspberry Pi (Enable GPIO Interfaces)

- Open the terminal and run the command `sudo raspi-config`.
- Select "Interface Options" and then enable the I2C interface.
- Check if I2C is enabled by running:

  ```
  sudo i2cdetect -y 1
  ```

  This should list all I2C devices, including the Lidar sensor.

Part 2: Set up Python Virtual Environment:

- Navigate to the `Documents` directory using the command `cd Documents/`.
- Create a virtual environment with the command `python -m venv IOT_LAB`.
- Activate the virtual environment by running `source IOT_LAB/bin/activate`.

Part 3: Using Lidar Sensor with Raspberry Pi 4

- **Hardware Connections of VL53L0X sensor:**
  - `VCC` pin to `5V` on Raspberry Pi
  - `GND` pin to `Ground (GND)`
  - `SDA` pin to `GPIO 2 (SDA1)`
  - `SCL` pin to `GPIO 3 (SCL1)`
- **Run the Lidar Program:**
  - Ensure the Python virtual environment is active and execute `python lidar.py`.
  - The expected output includes distance measurements in millimeters, e.g., `Range: 8190mm`, `Range: 23mm`.

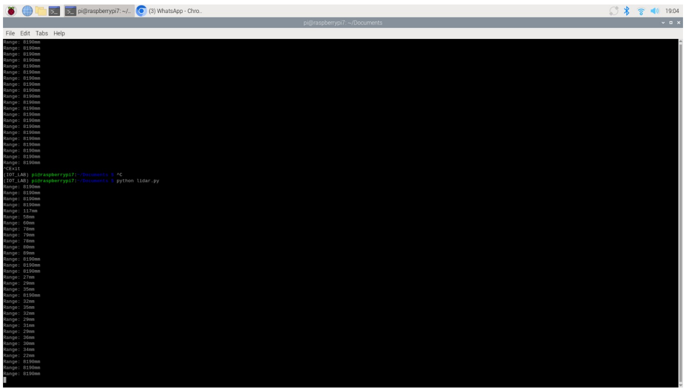

Figure 1: LIDAR Sensor data displayed on the serial monitor

1.3.1 Part 3: Using OLED Display with Raspberry Pi 4

- **Hardware Connections:**
  - `VCC` pin of OLED to `3.3V` on Raspberry Pi
  - `GND` pin to `Ground (GND)`
  - `SDA` pin to `GPIO 2 (SDA1)`
  - `SCL` pin to `GPIO 3 (SCL1)`
- **Run the OLED Program:**
  - Make I2C connections for the OLED display.
  - Run `oled.py` to display data on the OLED screen.
  - The display should show text output.

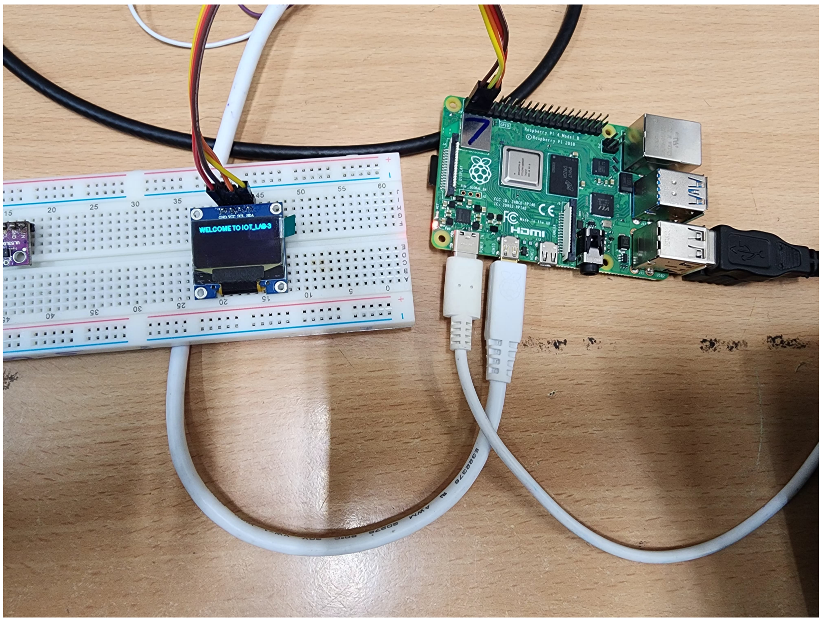

Figure 2: OLED Output

1.4 Code

1.4.1 Lidar Sensor Program

```
# SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
# SPDX-License-Identifier: MIT

# Simple demo of the VL53L0X distance sensor.
# Will print the sensed range/distance every second.
import time

import board
import busio

import adafruit_vl53l0x

# Initialize I2C bus and sensor.
i2c = busio.I2C(board.SCL, board.SDA)
vl53 = adafruit_vl53l0x.VL53L0X(i2c)

# Optionally adjust the measurement timing budget to change speed and accuracy.
# See the example here for more details:
#   https://github.com/pololu/vl53l0x-arduino/blob/master/examples/Single/Single.ino
# For example a higher speed but less accurate timing budget of 20ms:
# vl53.measurement_timing_budget = 20000
# Or a slower but more accurate timing budget of 200ms:
# vl53.measurement_timing_budget = 200000
# The default timing budget is 33ms, a good compromise of speed and accuracy.

try:
```

```python
    # Main loop will read the range and print it every second.
    while True:
        print("Range:␣{0}mm".format(vl53.range))
        time.sleep(0.2)
except KeyboardInterrupt:
    print("Exit")  # Exit on CTRL+C
```

[language:python]

1.4.2 OLED Display Program

```python
import time

import board
import busio

import digitalio
import adafruit_ssd1306
from PIL import Image, ImageDraw, ImageFont

# Initialize I2C bus and sensor.
i2c = busio.I2C(board.SCL, board.SDA)

# Define the Reset Pin
oled_reset = digitalio.DigitalInOut(board.D4)

oled = adafruit_ssd1306.SSD1306_I2C(128, 64, i2c, addr=0x3C)

# Clear display.
oled.fill(0)
oled.show()
def disp(data):

    # Create blank image for drawing.
    image = Image.new("1", (oled.width, oled.height))

    # Get drawing object to draw on image.
    draw = ImageDraw.Draw(image)

    # Load default font.
    font = ImageFont.load_default()

    # Define text position
    (x, y) = (0, 0)

    # Draw the text
    draw.text((x, y), data, font=font, fill=255)

    # Display the image
    oled.image(image)
    oled.show()


try:
    while True:
```

```
        data = input("Enter␣text␣to␣display␣:-␣")
        disp(data)
        time.sleep(0.2)
except KeyboardInterrupt:
    print("Exit")  # Exit on CTRL+C
```

[language:python]

2 Assignment

2.1 Objective

This assignment aims to use the VL53L0X Lidar sensor to measure distance and display the results on the OLED screen. The display should also show the student's roll number followed by the measured distance.

2.2 Code

```
import time
import board
import busio
import digitalio
import adafruit_ssd1306
from PIL import Image, ImageDraw, ImageFont
import adafruit_vl53l0x

# Initialize I2C bus and sensor.
i2c = busio.I2C(board.SCL, board.SDA)

# Define the Reset Pin
oled_reset = digitalio.DigitalInOut(board.D4)

# Initialize OLED display
oled = adafruit_ssd1306.SSD1306_I2C(128, 64, i2c, addr=0x3C)

# Initialize LIDAR sensor
vl53 = adafruit_vl53l0x.VL53L0X(i2c)

# Clear display.
oled.fill(0)
oled.show()

def disp(data):
    # Create blank image for drawing.
    image = Image.new("1", (oled.width, oled.height))

    # Get drawing object to draw on image.
    draw = ImageDraw.Draw(image)

    # Load default font.
    font = ImageFont.load_default()

    # Clear the display before drawing new text
    draw.rectangle((0, 0, oled.width, oled.height), outline=0, fill=0)
```

```python

    # Define text position
    (x, y) = (0, 0)

    # Draw the text
    draw.text((x, y), data, font=font, fill=255)

    # Display the image
    oled.image(image)
    oled.show()

try:
    while True:
        distance = vl53.range
        disp("M24EEV020_Dist: {} mm".format(distance))
        time.sleep(1)  # Update every second
except KeyboardInterrupt:
    print("Exit")  # Exit on CTRL+C
```

[language:python]

2.3 Explanation of the Program

The program initializes the I2C bus to communicate with both the VL53L0X Lidar sensor and the OLED display. It continuously retrieves the distance measured by the Lidar sensor and displays this value on the OLED display alongside the roll number. The screen updates every second, allowing real-time measurement of the distance between the sensor and any object in its path.

2.4 Observations

- The Lidar sensor provided accurate distance measurements, with typical outputs such as `Range: 8190mm`, indicating successful interfacing with the Raspberry Pi.

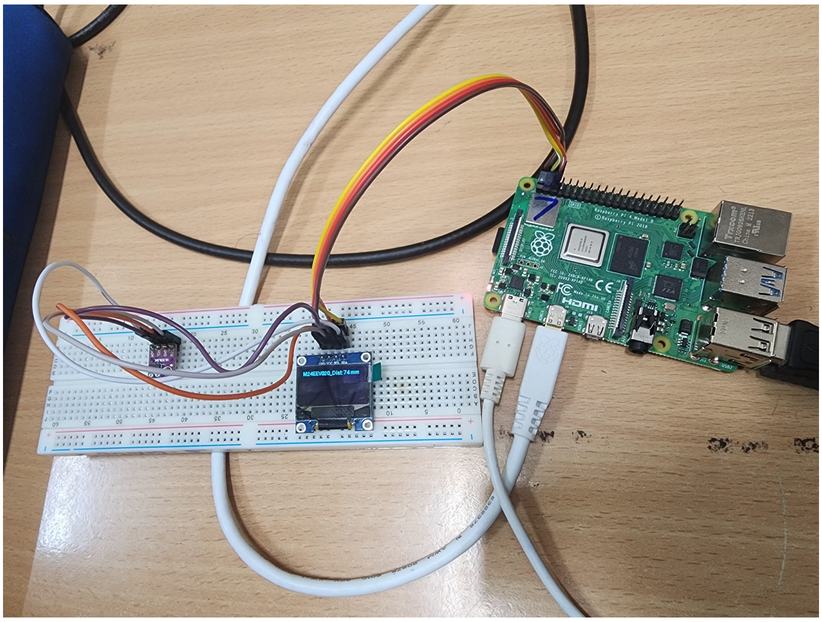

Figure 3: Setup of Raspberry Pi with LIDAR Sensor and OLED Display

- The OLED display successfully visualized the measured distance alongside the roll number, e.g., `RollNo: B21ES006, Distance: 23mm`.

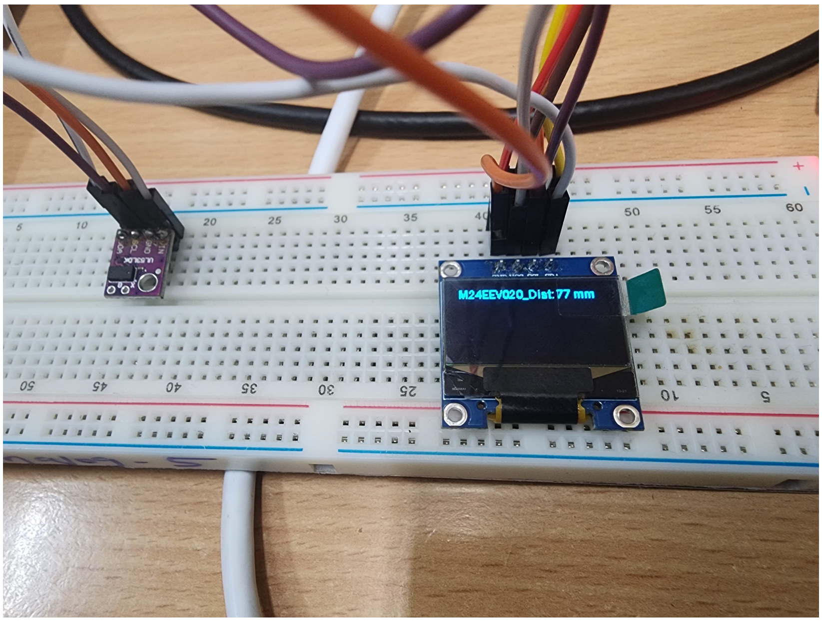

Figure 4: Data of LIDAR Sensor displayed on OLED with Roll no.

2.5 Results

- **Part 1:** The Raspberry Pi was successfully set up with I2C communication enabled, allowing proper interfacing with external sensors.
- **Part 2:** The Lidar sensor was successfully interfaced, and distance measurements were accurate.
- **Part 3:** The OLED display worked as expected, showing both the roll number and the distance measured by the Lidar sensor.