VL53L0X Time-of-Flight (ToF) Sensor Project

Objective

The goal of this project was to connect multiple VL53L0X Time-of-Flight distance sensors to the Raspberry Pi using the I²C bus and implement address reassignment to enable simultaneous operation of four sensors for autonomous robot navigation.

Requirements

Component	Quantity
VL53L0X ToF sensor modules	4
Raspberry Pi 4 Model B	1
Breadboard	1
Jumper wires	>20
Power source (3.3V from Pi)	1

Note: No voltage dividers or resistors are required as the VL53L0X operates at 3.3V, which is directly compatible with Raspberry Pi GPIO pins.

Wiring Setup

I²C Bus Connections (Shared by All Sensors)

- **VCC** → 3.3V (Pi)
- **GND** → GND (Pi)
- **SDA** → GPIO 2 (I²C Data)
- **SCL** → GPIO 3 (I²C Clock)

Individual XSHUT Pin Connections

Each sensor requires a unique GPIO pin connected to its XSHUT (shutdown) pin for address reassignment:

Sensor Position	XSHUT Pin	GPIO Pin
Right sensor	XSHUT	GPIO 22
Left sensor	XSHUT	GPIO 27
Back sensor	XSHUT	GPIO 17
Front sensor	XSHUT	GPIO 26

All sensors were mounted on a breadboard with their I^{2}C lines connected in parallel, while each XSHUT pin was individually controlled to enable sequential initialization and address assignment.

Code

Sensor Initialization with Address Reassignment

python

```
import time
import board
from digitalio import DigitalInOut
from adafruit_vl53l0x import VL53L0X

# Initialize I2C bus
i2c = board.I2C()

# Define XSHUT pins for each sensor
xshut = [
    DigitalInOut(board.D22),  # Right sensor
    DigitalInOut(board.D27),  # Left sensor
    DigitalInOut(board.D17),  # Back sensor
    DigitalInOut(board.D26)   # Front sensor
]

# Configure all XSHUT pins as outputs and shut down all sensors
for power_pin in xshut:
    power_pin.switch_to_output(value=False)
    print(f"Shut down sensor: {power_pin}")

time.sleep(0.1)

# Initialize sensor array
vl53 = []
SENSOR_NAMES = ['Right', 'Left', 'Back', 'Front']

# Power on and configure sensors one by one
for i, power_pin in enumerate(xshut):
    # Power on this sensor
    power_pin.value = True
    print(f"Turned ON {SENSOR_NAMES[i]} sensor")
    time.sleep(0.1)

    # Create sensor object
    sensor = VL53L0X(i2c)

    # Change I2C address (except for the last sensor)
    if i < len(xshut) - 1:
        new_address = i + 0x30
        sensor.set_address(new_address)
        print(f"  → Address changed to: 0x{new_address:02X}")
    else:
        print(f"  → Kept default address: 0x29")

    vl53.append(sensor)
    time.sleep(0.1)

print("All sensors initialized successfully!")
```

Distance Measurement Loop

python

```
def read_sensors():
    """Read distance from all four sensors"""
    distances = {}

    for i, sensor in enumerate(vl53):
        name = SENSOR_NAMES[i]
        try:
            distance = sensor.range  # Distance in mm
            if 10 <= distance <= 2000:
                distances[name] = distance
            else:
                distances[name] = 2000  # Out of range
        except OSError:
            distances[name] = 2000  # Sensor error

        time.sleep(0.02)  # Small delay between reads

    return distances

# Main measurement loop
try:
    while True:
        distances = read_sensors()
        print(f"Right: {distances['Right']:4}mm | "
              f"Left: {distances['Left']:4}mm | "
              f"Back: {distances['Back']:4}mm | "
              f"Front: {distances['Front']:4}mm")
        time.sleep(0.1)

except KeyboardInterrupt:
    print("\nMeasurement stopped")
```

Discussion

Sensor Technology