SYSC 4805 B

Lab Section – L3

Lab 2

Ultrasonic, ToF Distance, and IMU Sensors

Group – 1

Group Members:

Raneem Abouseta (101180464)

Aryan Huq Khan (101205385)

Ibrahim Faisal (101209598)

Ali Zaid (101223823)

Date: 20-09-2024

Due Date: 27-09-2024

Task 1: Testing the Ultrasonic Distance Sensor

An ultrasonic distance sensor helps a system understand its surroundings by measuring how far objects are. It works by sending out sound waves at 40 KHz and timing how long they take to bounce back. Using this information, the system calculates the distance.

When the microcontroller sends a brief signal (10 µs) to the Trig pin, the sensor sends out sound waves. As soon as the waves bounce back, the Echo pin signals the return. The time it takes for this process tells the system how far away the object is, based on the speed of sound. [3]

Experiment 1.1: Generating the trigger signal using the Arduino Due Board

Exp. 1.1.1: Using Delay Functions

1.1.1.7)

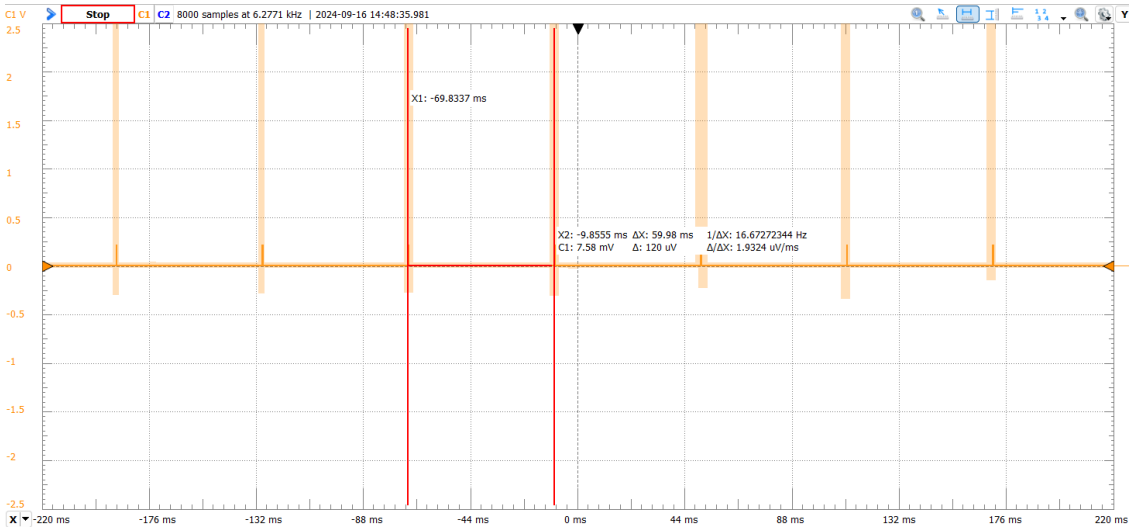

Figure 1: Generated Signal using Delay [1]

1.1.1.8)

Measured Period = 59.98ms

The small difference in the period (59.98 ms instead of 60 ms) happens because the Arduino's timing functions aren't perfectly precise, and other tasks running on the board can cause slight delays. The oscilloscope also has its own tiny measurement variations. Plus, the time it takes for the Arduino to run each line of code adds a bit of overhead.

Exp. 1.1.2: Using the Analogwrite() Function

1.1.2.5)

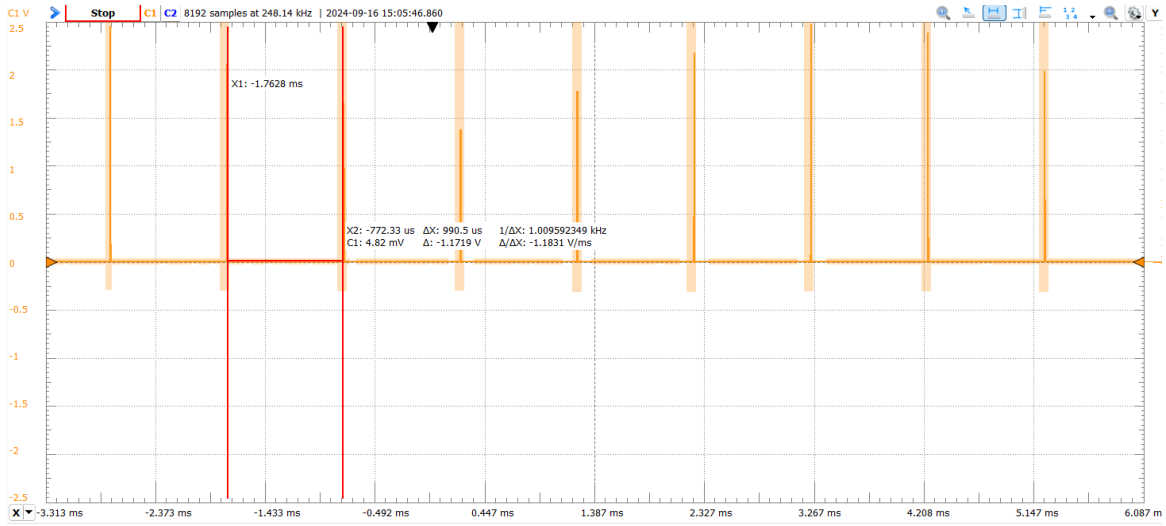

Figure 2: Generated Signal using AnalogWrite [1]

Exp. 1.1.3: Using the Timer Counter Unit

1.1.3.7)

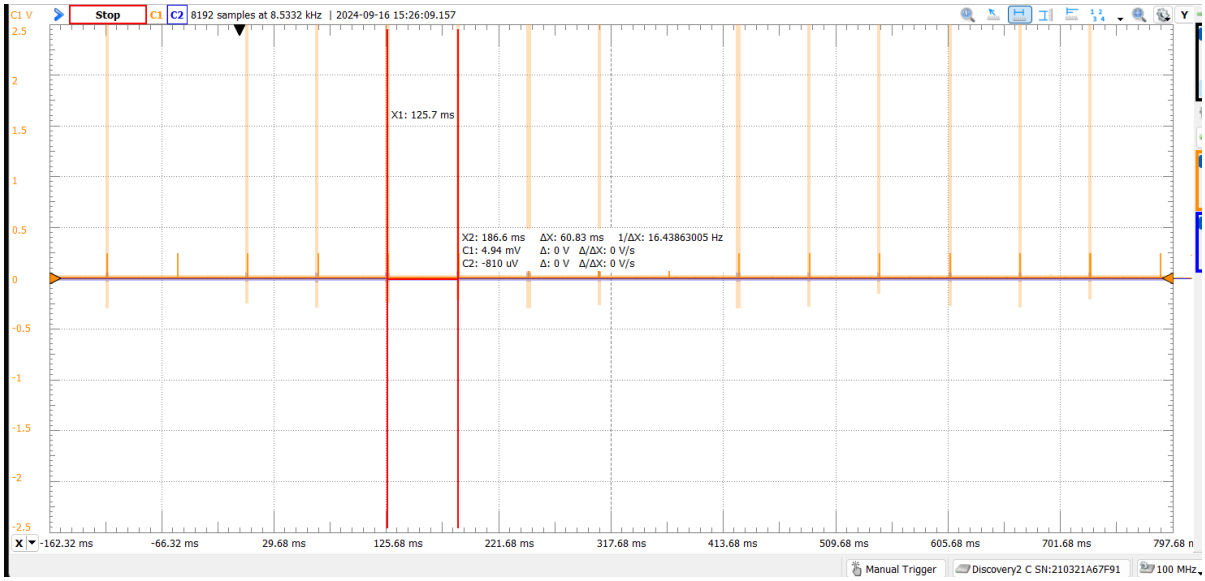

Figure 3: Generated Signal using Timer Counter [1]

Exp. 1.1.4: Using the PWM Macrocell Unit

1.1.4.7)

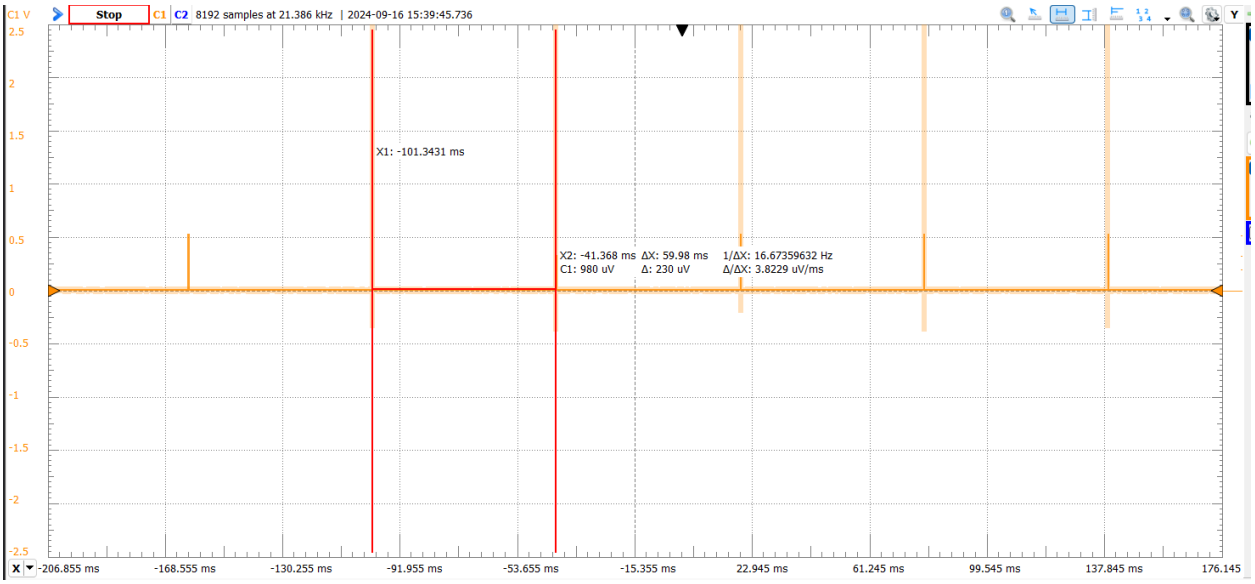

Figure 4: Generated Signal using PWM Macrocell [1]

Experiment 1.2: Test the Ultrasonic Distance Module using Oscilloscope

1.2.5)

Table 1.2: Experiment 1 Measurements (Using Oscilloscope)

Actual Distance (cm)	Pulse Width	Measured Distance (cm)	Distance Error (cm)
Min	32ms	0.4	N/A
5	348.5us	5.977	0.977
10	610us	10.461	0.461
15	1.017ms	17.442	2.442
20	1.317ms	22.587	2.587
25	1.681ms	28.829	3.829
30	2.133ms	36.581	6.581
40	2.354ms	40.371	0.371
50	2.547ms	43.681	6.319
75	5.348ms	91.718	16.718
100	10.300ms	176.645	76.645
200	12.241ms	209.933	9.933
Max	32.99ms	565.933	N/A

Experiment 1.3: Testing the Ultrasonic Distance Module using Arduino

Exp.1.3.1: Using PulseLn() with trigger using Delay(), Timer Counter and PWM

1.3.1.2)

```
Section L3, Group 1, Distance: 24.51 cm
Section L3, Group 1, Distance: 25.18 cm
Section L3, Group 1, Distance: 26.02 cm
Section L3, Group 1, Distance: 23.05 cm
Section L3, Group 1, Distance: 254.30 cm
Section L3, Group 1, Distance: 254.88 cm
Section L3, Group 1, Distance: 254.88 cm
Section L3, Group 1, Distance: 254.88 cm
Section L3, Group 1, Distance: 254.87 cm
Section L3, Group 1, Distance: 254.30 cm
Section L3, Group 1, Distance: 254.30 cm
```

Figure 5: Distance(cm) on Serial Monitor of Ultrasonic Sensor using Arduino [2]

```
Section L3, Group 1, Pulse Width: 694 us
Section L3, Group 1, Pulse Width: 664 us
Section L3, Group 1, Pulse Width: 659 us
Section L3, Group 1, Pulse Width: 679 us
Section L3, Group 1, Pulse Width: 644 us
Section L3, Group 1, Pulse Width: 652 us
Section L3, Group 1, Pulse Width: 691 us
Section L3, Group 1, Pulse Width: 9713 us
Section L3, Group 1, Pulse Width: 9813 us
Section L3, Group 1, Pulse Width: 9714 us
Section L3, Group 1, Pulse Width: 9814 us
```

Figure 6: Pulse Width(us) on Serial Monitor of Ultrasonic Sensor using Arduino [2]

1.3.1.3)

Table 1.2: Experiment 1 Measurements (Using Arduino Due)

Actual Distance (cm)	Pulse Width	Measured Distance (cm)	Distance Error (cm)
Min	29728us	0.2	N/A
5	493us	8.455	3.455
10	723us	12.399	2.399
15	964us	16.533	1.533
20	1121us	19.225	0.775
25	1366us	23.427	1.573
30	1744us	29.91	0.09
40	2189us	37.541	2.549
50	3247us	55.686	5.686
75	4631us	79.422	4.422
100	5704us	97.824	2.176
200	11261us	193.126	6.874
Max	31087us	533.142	N/A

Exp.1.3.2: Using Timer Unit for both triggering and measuring the Echo

1.3.2.7)

```
Section L3, Group 1:  6.712167 cm
Section L3, Group 1:  8.997452 cm
Section L3, Group 1:  9.068690 cm
Section L3, Group 1:  65.388881 cm
Section L3, Group 1:  64.343786 cm
Section L3, Group 1:  64.908024 cm
Section L3, Group 1:  65.106357 cm
Section L3, Group 1:  65.968095 cm
Section L3, Group 1:  66.650119 cm
```

Figure 7: Serial Monitor showing the Echo measurements using Timer [2]

Exp. 1.3.3: Enable and use of Interrupt on the Echo signal

1.3.3.5)

```
Section L3, Group 1:   63.733000 cm
Section L3, Group 1:   63.649214 cm
Section L3, Group 1:   63.722881 cm
Section L3, Group 1:   64.172571 cm
Section L3, Group 1:   63.831357 cm
Section L3, Group 1:   63.784000 cm
Section L3, Group 1:   64.203333 cm
Section L3, Group 1:   63.831357 cm
Section L3, Group 1:   64.219119 cm
Section L3, Group 1:   64.226000 cm
Section L3, Group 1:   63.838238 cm
```

Figure 8: Serial Monitor showing the Echo measurements using Interrupts [2]

Discussion

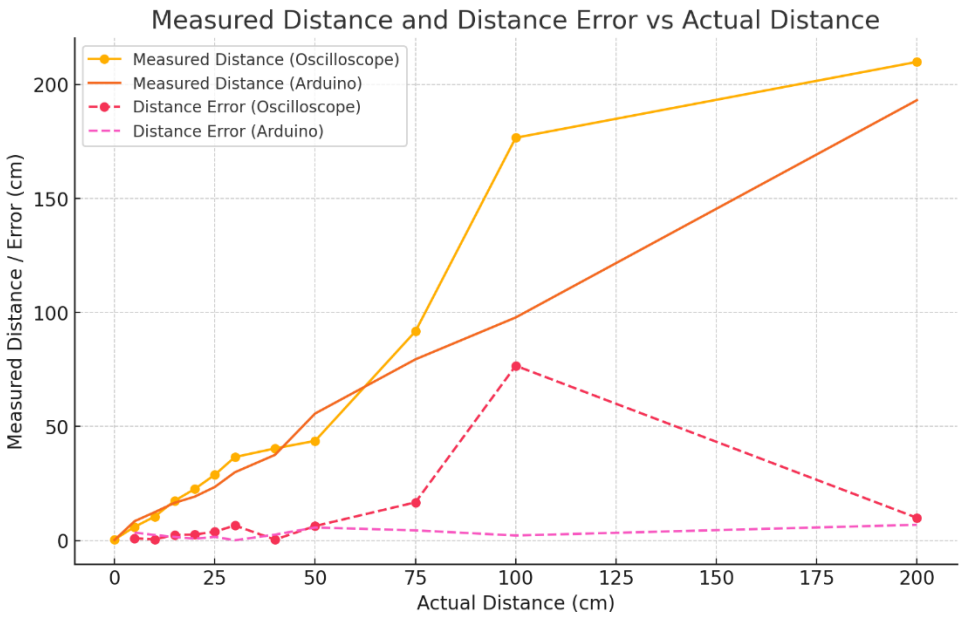

Figure 9: Graph of MDO v MDA v DEO v DEA [4]

- What are the measurement conditions that correspond to minimum error?

There is the least error at 0.4cm for the oscilloscope and 0.1cm for the Arduino Due board.

- For the Arduino measured curve, what do you suggest to compensate the error?

In software, we can implement an algorithm that uses our data on measurement error to enhance the accuracy of measurements.

Task 2: Testing the VL53L1X Time-of-Flight Distance Sensor

The VL53L1X is a compact infrared (IR) laser distance sensor capable of measuring distances up to 4 meters. It operates by transmitting IR pulses and calculating the Time-of-Flight (the time it takes for the pulse to reflect from an object) to determine distance. Though small (less than 0.5 cm), it comes on a Pololu carrier board for easier integration into projects.

Key features include fast, accurate measurements at up to 50 Hz and a customizable 27° Field-of-View (FoV), allowing for multizone operation through a programmable region-of-interest (ROI). The sensor uses I2C for communication and has shutdown and interrupt pins for additional control. It offers three distance modes (short, medium, long), optimized for different ranges and light conditions, and allows users to adjust settings like timing budget and ROI for optimal performance. [3]

Experiment 2.1: Using one VL53L1X Sensor over I2C

```
Start, Stop
Start,
Start,
Start,
Start, Stop
Start,
Start, Stop
Start,
Start,
Start, Stop
Start,
Start,
Start, Stop
Start, h07 [ h03 | RD ] NACK, Error
Restart,
Restart,
Restart, h00 [ h00 | WR ], h06, h00, h00, h00, h00, h00, Error
Start, h00 [ h00 | WR ], h00, h00, h00, h00, h00, h00, h00, h00, h00, h00, h00, Error
Restart, h00 [ h00 | WR ], h00, h00, h00, h00, h00, h00, h00, Error
Restart, h00 [ h00 | WR ], h00, h00, h00, h00, h00, h00, h00, Error
Start, h00 [ h00 | WR ], h00, h00, h00, h00, Error
Restart, h00 [ h00 | WR ], h00, h00, h00, Stop
Start, h00 [ h00 | WR ], Error
Start, h00 [ h00 | WR ], h00, h00, Error
Start, h00 [ h00 | WR ], h00, h00, h00, h00, h00, h00, h00, h00, Error
Restart,
Start, Stop
Start,
Start,
Start,
Start, h0A [ h05 | WR ], h80, h41 NACK, Error
Start, h53 [ h29 | RD ], hEA, hCC NACK, Stop
Start, h52 [ h29 | WR ], h00, h00, h00, Stop
Start, h52 [ h29 | WR ], h00, h00, h01, Stop
Start, h52 [ h29 | WR ], h00, hE5, Stop
Start, h53 [ h29 | RD ], h03 NACK, Stop
Start, h52 [ h29 | WR ], h00, h2E, Stop
Start, h53 [ h29 | RD ], h00 NACK, Stop
Start, h52 [ h29 | WR ], h00, h2E, h01, Stop
Start, h52 [ h29 | WR ], h00 NACK, Stop
Start, h53 [ h29 | RD ] NACK, Stop
Start, h52 [ h29 | WR ] NACK, Stop
Start, h53 [ h29 | RD ] NACK, Stop
Start, Error
Restart,
Restart,
```

Figure 10: Output from VL53L1X module using I2C protocol analyzer [1]

2.1.6)

Table for Short Mode (Timing Budget = 50ms)

Actual Distance (cm)	Measured Distance (cm)	Distance Error (cm)
Min	0	N/A
10	10	0
20	20.1	0.1
30	29.9	0.1
40	40.2	0.2
50	50	0
60	60	0
70	69.9	0.1
80	80	0
90	90.2	0.2
100	100	0
Max	128.3	N/A

Table for Medium Mode (Timing Budget = 70ms)

Actual Distance (cm)	Measured Distance (cm)	Distance Error (cm)
Min	0	N/A
10	10.3	0.3
20	20.2	0.2
30	30.1	0.1
40	39.9	0.1
50	50.2	0.2
60	60	0
70	70.2	0.2
80	79.7	0.3
90	90.5	0.5
100	100.2	0.2
Max	202.7	N/A

Table for Long Mode (Timing Budget = 100ms)

Actual Distance (cm)	Measured Distance (cm)	Distance Error (cm)
Min	0	N/A
10	10.5	0.5
20	19.7	0.3
30	30.1	0.1
40	40.6	0.6
50	50.4	0.4
60	60.6	0.6
70	69.5	0.5
80	80.3	0.3
90	90.1	0.1
100	100.2	0.2
Max	388.6	N/A

2.1.7)

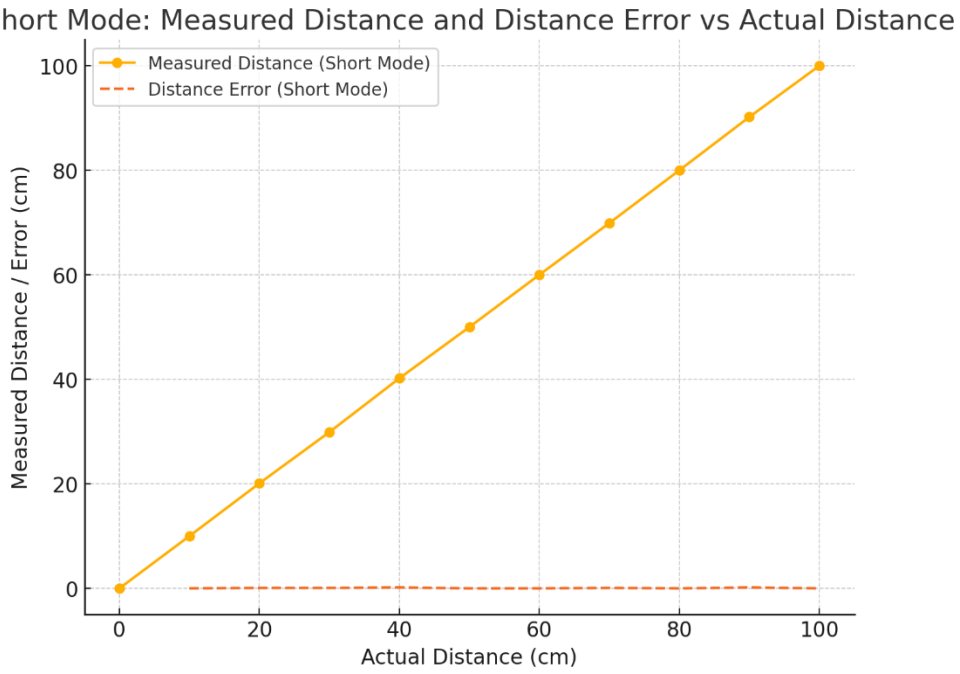

Figure 11: Graph for Measured distance and Error Distance(cm) v Actual Distance(cm) for Short Mode

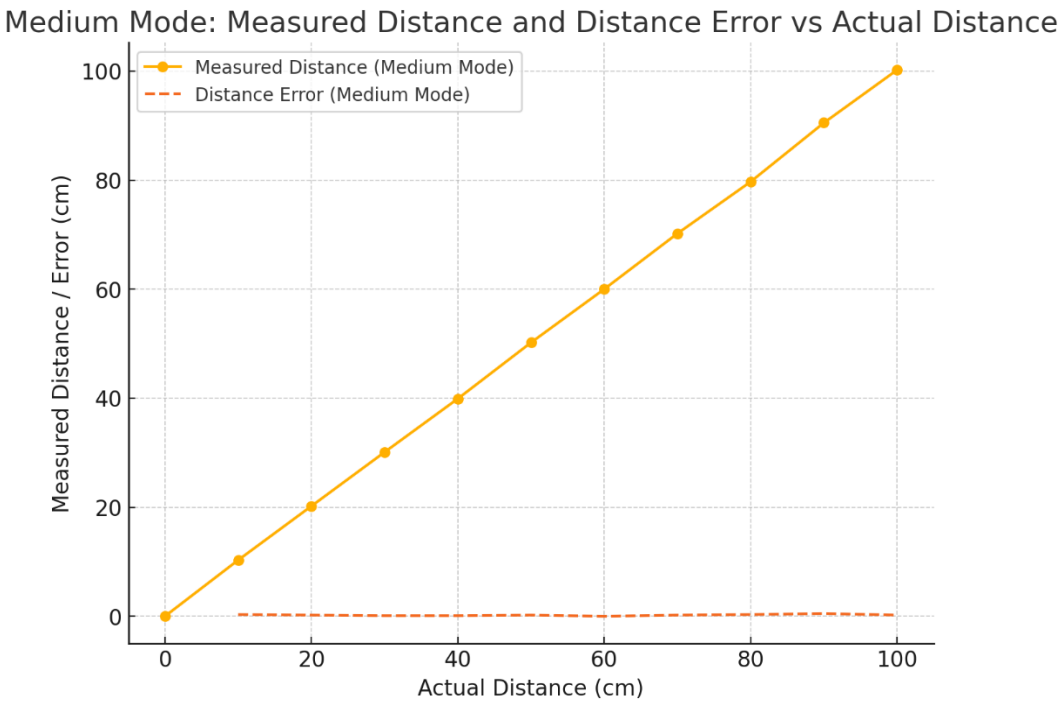

Figure 12: Graph for Measured distance and Error Distance(cm) v Actual Distance(cm) for Medium Mode

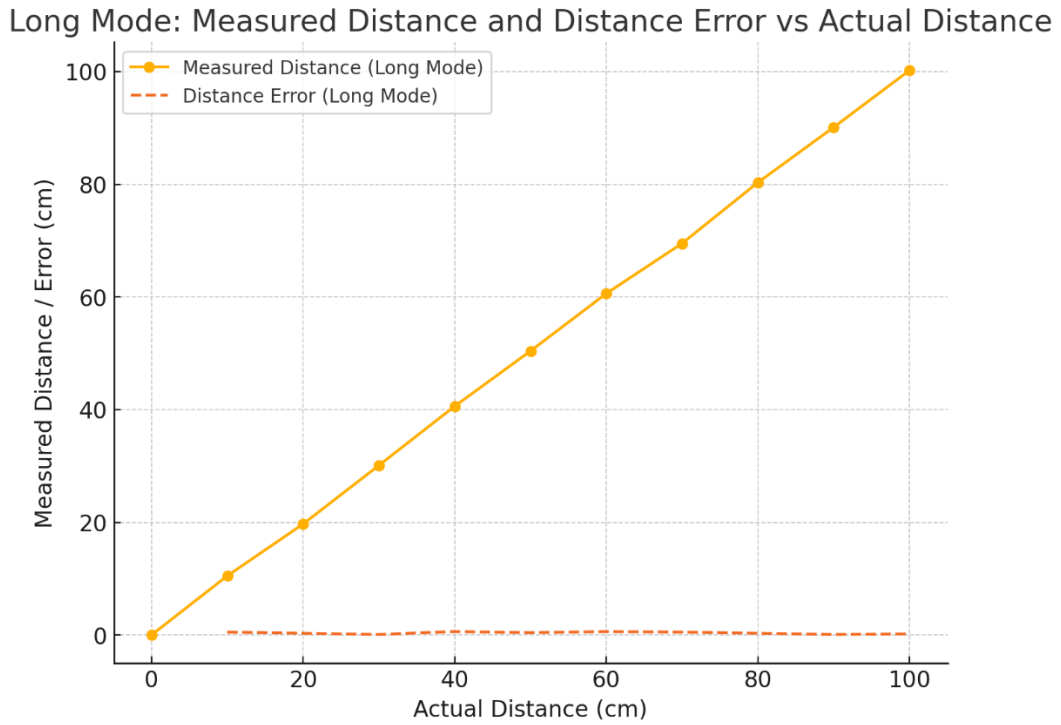

Figure 13: Graph for Measured distance and Error Distance(cm) v Actual Distance(cm) for Long Mode

Experiment 2.2: Using two VL53L1X Sensors over I2C

2.2.3) (c)

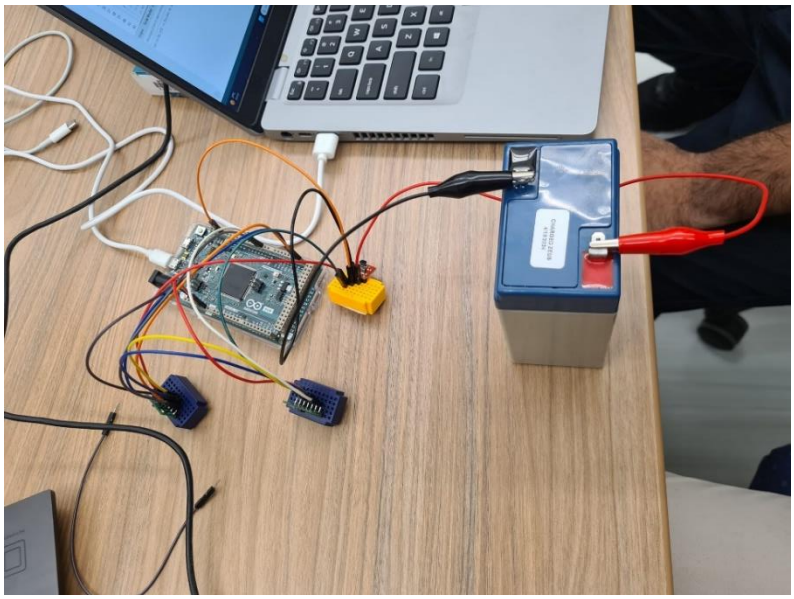

Figure 14: Two VL53L1X Sensor Area Setup

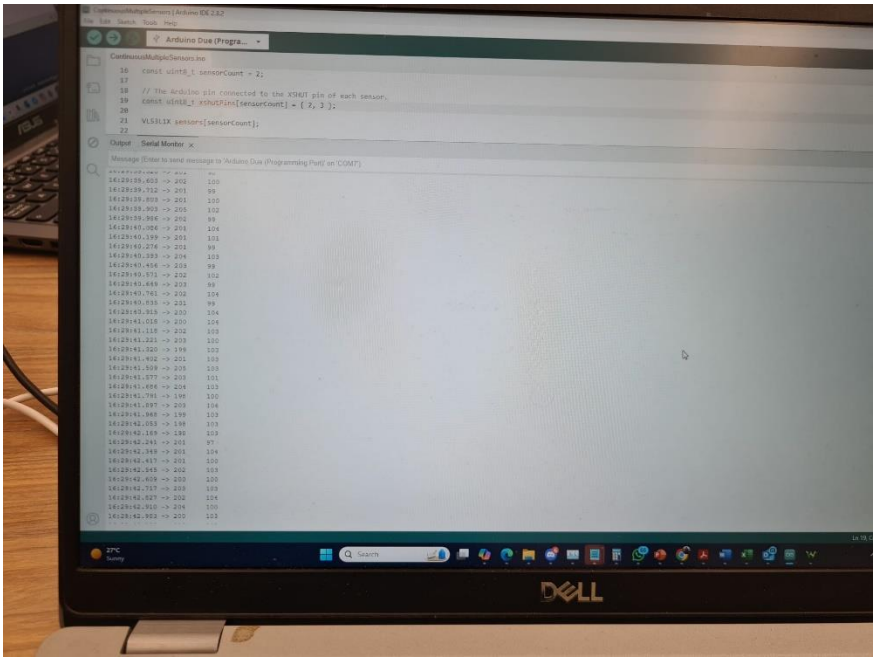

Figure 15: Serial Monitor shows 2 VL53L1X Sensor Readings [2]

Task 3: Testing the MinIMU-9 v5 IMU Unit

An Inertial Measurement Unit (IMU) is an electronic sensor that tracks an object's orientation and movement by measuring inertial quantities like acceleration, angular velocity, and magnetic fields. The MiniIMU-9 v5 module combines two chips: the LSM6DS33 (accelerometer and gyroscope) and the LIS3MDL (magnetometer) to provide detailed 3D orientation and movement data.

Key Components:

1. Orientation Angles: Changes in orientation are described by three Euler angles: Yaw, Roll, and Pitch. IMUs track these angular changes to determine the object's position in space.

2. Accelerometer: The accelerometer measures linear acceleration in 3 axes. It detects the force acting on the object and can determine the object's orientation relative to the Roll and Pitch axes, but Yaw changes don't affect its readings.

3. Gyroscope: The gyroscope measures the angular velocity of the object in 3 axes. It works with the accelerometer to calculate Roll and Pitch changes and helps the magnetometer track Yaw, but it measures only the rate of change in angles, not absolute angles.

4. Magnetometer: The magnetometer measures the Earth's magnetic field in 3 axes to determine Yaw and provide a reference for orientation. While it can be affected by magnetic interference, it works with the gyroscope for more accurate Yaw measurements.

Together, these sensors provide accurate orientation and motion tracking, with the accelerometer and gyroscope measuring changes in Roll and Pitch, and the magnetometer offering Yaw angle data. [3]

Experiment 3.1: Using the LSM6 Accelerometer and Gyroscope

3.1.3)

Exp. 3.1.1: Test the Accelerometer

(a)

```
Scaled Values of Acceleration (+-2g))
X-axis:0.092 Y-axis:0.011 Z-axis:-0.959
Pitch and Roll Rotation Angles
Roll(x): -0.685 - Pitch: -5.476
```

(b)

```
Scaled Values of Acceleration (+-2g))
X-axis:-0.998 Y-axis:-0.029 Z-axis:-0.053
Pitch and Roll Rotation Angles
Roll(x): 28.367 - Pitch: 86.517
```

(c)

```
Scaled Values of Acceleration (+-2g))
X-axis:0.017 Y-axis:-1.015 Z-axis:0.061
Pitch and Roll Rotation Angles
Roll(x): -86.554 - Pitch: -0.938
```

(d)

```
Scaled Values of Acceleration (+-2g))
X-axis:-0.191 Y-axis:0.671 Z-axis:-0.688
Pitch and Roll Rotation Angles
Roll(x): -44.306 - Pitch: 11.218
```

(e)

```
Scaled Values of Acceleration (+-2g))
X-axis:0.033 Y-axis:-0.008 Z-axis:1.016
Pitch and Roll Rotation Angles
Roll(x): -0.471 - Pitch: -1.881
```

Figure 16: Serial Monitor showing LSM6 Accelerometer Measurement [2]

3.1.4)

Exp.3.1.2: Test the Gyroscope

(a)

```
Scaled Values of Gyroscope (+- 245dps)
X-axis:7.577 Y-axis:-5.031 Z-axis:0.079
Scaled Values of Gyroscope (+- 245dps)
X-axis:-17.561 Y-axis:17.938 Z-axis:-168.516
Scaled Values of Gyroscope (+- 245dps)
X-axis:16.870 Y-axis:-15.286 Z-axis:34.597
Scaled Values of Gyroscope (+- 245dps)
X-axis:3.220 Y-axis:-8.094 Z-axis:66.115
Scaled Values of Gyroscope (+- 245dps)
X-axis:1.505 Y-axis:-2.266 Z-axis:-19.530
```

(b)

```
Scaled Values of Gyroscope (+- 245dps)
X-axis:286.703 Y-axis:-34.624 Z-axis:-28.192
Scaled Values of Gyroscope (+- 245dps)
X-axis:121.835 Y-axis:-20.965 Z-axis:-13.151
Scaled Values of Gyroscope (+- 245dps)
X-axis:-286.703 Y-axis:31.894 Z-axis:127.986
Scaled Values of Gyroscope (+- 245dps)
X-axis:-46.673 Y-axis:-1.531 Z-axis:1.207
```

(c)

```
Scaled Values of Gyroscope (+- 245dps)
X-axis:-0.079 Y-axis:-11.515 Z-axis:-5.180
Scaled Values of Gyroscope (+- 245dps)
X-axis:-4.209 Y-axis:-0.577 Z-axis:16.608
Scaled Values of Gyroscope (+- 245dps)
X-axis:-24.972 Y-axis:14.525 Z-axis:-18.944
Scaled Values of Gyroscope (+- 245dps)
X-axis:44.748 Y-axis:-12.749 Z-axis:0.630
```

Figure 17: Serial Monitor showing LSM6 Gyroscope Measurement [2]

Experiment 3.2: Using the LIS3MDL Magnetometer

3.2.3) Exp. 3.2.1: Check operation (absolute)

(a) Max X value

```
Scaled Values of Magnetometer (+- 4 gauss)
X-axis:-0.182 Y-axis:0.175 Z-axis:0.083
```

Max Y value

```
Scaled Values of Magnetometer (+- 4 gauss)
X-axis:-0.139 Y-axis:0.229 Z-axis:0.111
```

(b) Max Y value

```
Scaled Values of Magnetometer (+- 4 gauss)
X-axis:-0.084 Y-axis:-0.093 Z-axis:-0.017
```

Max Z value

```
Scaled Values of Magnetometer (+- 4 gauss)
X-axis:-0.029 Y-axis:-0.082 Z-axis:-0.181
```

(c) Max X value

```
Scaled Values of Magnetometer (+- 4 gauss)
X-axis:-0.329 Y-axis:0.102 Z-axis:-0.077
```

Max Z value

```
Scaled Values of Magnetometer (+- 4 gauss)
X-axis:-0.335 Y-axis:0.122 Z-axis:-0.134
```

Figure 18: Serial Monitor showing LIS3MDL Magnetometer Measurement [2]

References

[1] - “Digilent waveforms,” Digilent, https://digilent.com/shop/software/digilent-waveforms/.

[2] - Docs.arduino.cc, https://docs.arduino.cc/software/ide/#ide-v2.

[3] - https://brightspace.carleton.ca/d2l/le/content/292069/viewContent/3933819/View (Lab Manual)

Glossary

[4] – MDO = Measured Distance using Oscilloscope

MDA = Measured Distance using Arduino

DEO = Distance Error using Oscilloscope

DEA = Distance Error using Arduino