2DX4: Microprocessor Systems Project
Final Project

Instructors: Dr. Bruce, Dr. Haddara, Dr. Hranilovic, Dr. Shirani

Mohamed Negm – L07 – negmm

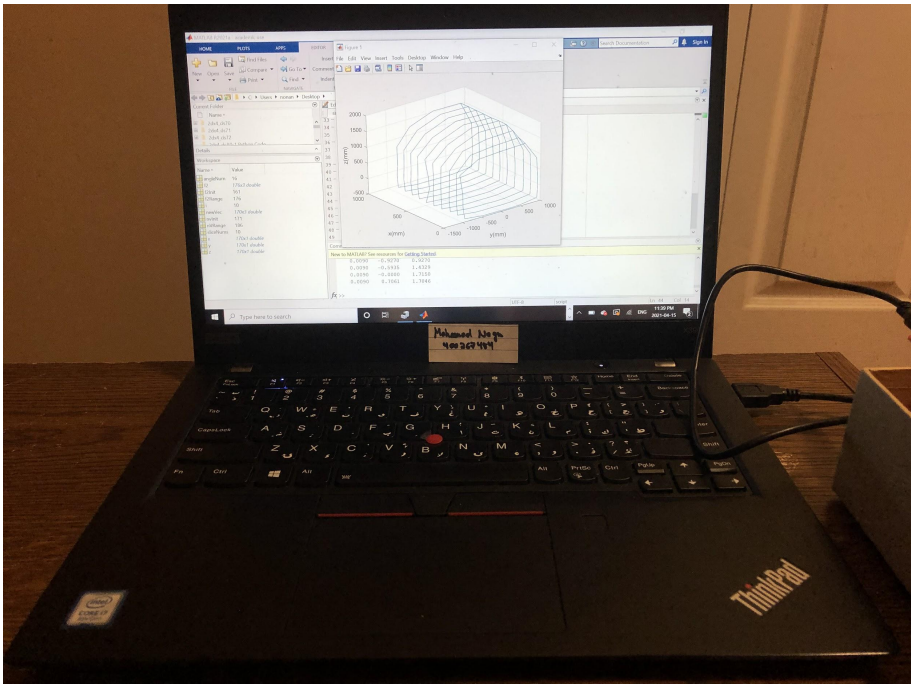

Figure 1: PC Element of Device

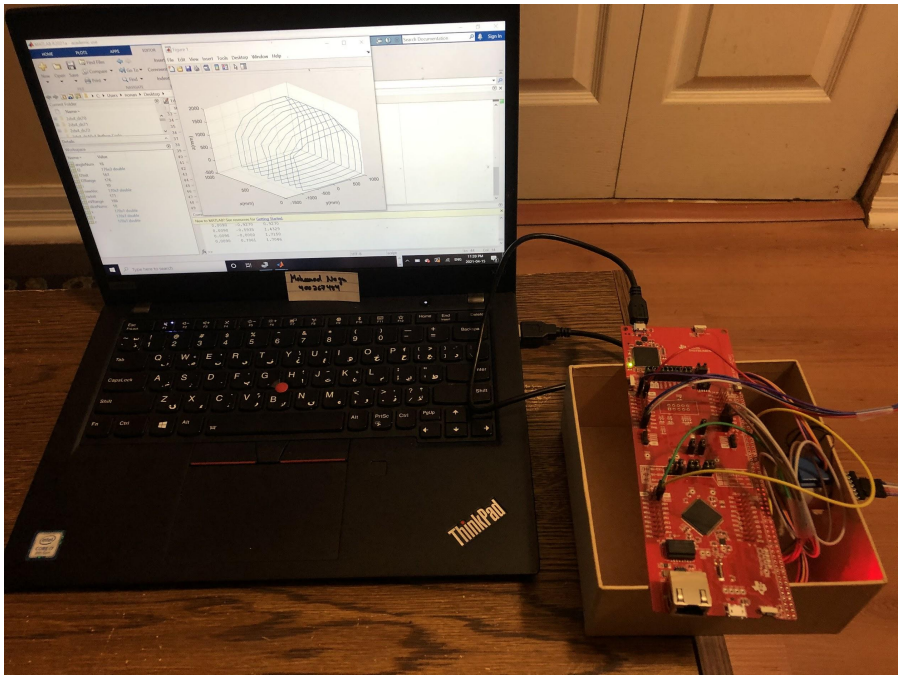

Figure 2: Complete Device

1.0 Device overview

1.1 Features

- This device ranges and scans the outline of its surroundings using a time of flight sensor, then visually displays the data on a 3D plane.
- Fast and accurate long-distance measurement ranging up to 4m
- Adjustable measurement angle
- Operating voltage of 5V
- Components
    - MSP432E401Y Microcontroller
    - VL53L1X Time of Flight(ToF) Sensor
    - 28BYJ-48 Stepper Motor
- Bus frequency: 24MHz
- Memory
    - 1024KB of Flash Memory
    - 256KB of SRAM
    - 6KB EEPROM
- Two 12-bit ADC modules
- Infrared Emitter of 940nm
- Serial communication
    - Programmable Baud Rate generator up to 15 Mbps
- Cost
    - $320 MSP432E401Y Microcontroller
    - $7.20 Stepper Motor
    - $15 Time of Flight(ToF) Sensor
- Integrated Programming Languages
    - 3.8.8 Python
    - C
    - R2020b Matlab

1.2 General Description

The functionality of this device can be separated into 3 segments of data acquisition, transmission, then processing, and visualization. Data acquisition begins at the VL53LIX time of flight sensor(ToF) which takes distance measurements. This sensor is mounted onto a 28BYJ-48 stepper motor which rotates clockwise and stops at a specified angle to take distance measurements. Then the measurements are transferred from the ToF to the MCU using I2C then to the PC using UART. The python code processes the data and calculates the x, y, and z dimensions for MATLAB to visualize.

1.3 Block Diagram

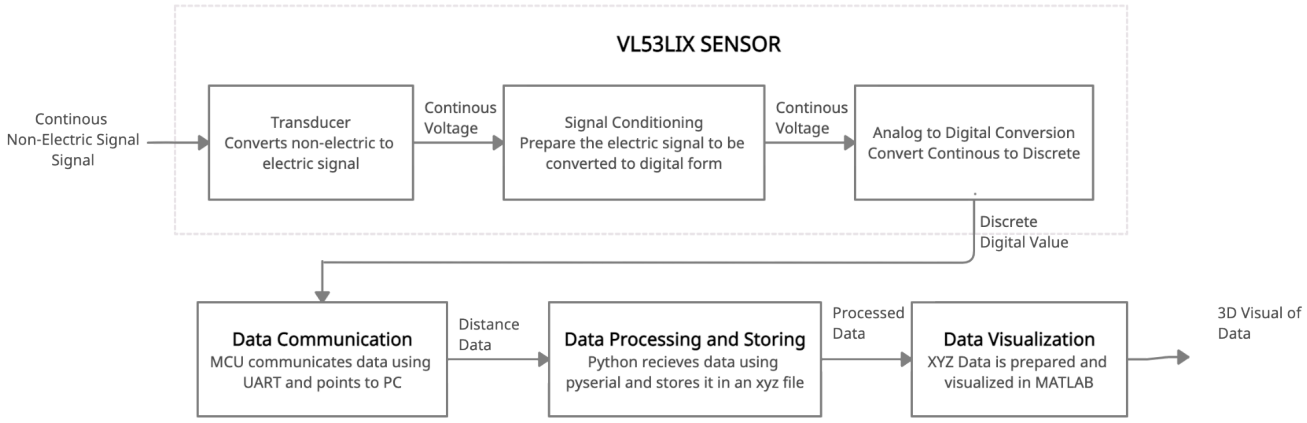

Figure 1.1: Block Diagram

2.0 Device Characteristics Table

Table: Device Characteristics Table

Characteristic	Detail
Major Components	- MSP432E401Y Microcontroller - VL53L1X Time of Flight(ToF) Sensor - 28BYJ-48 Stepper Motor - ULN2003 Driver
Pins	- PL0-4 for motor bit sequence - 3.3V, 5V, grd - SCL(PB2), SDA(PB3) Serial communication
Bus speed	24MHz
Serial Communication	- UART serial communication protocol baud rate of 115200Bps - I2C

3.0 Detailed Description

3.1 Distance Measurement

The distance measurement process is split into three segments, data acquisition, transmission, then processing, and visualization. Data acquisition begins at the time of flight sensor. The VL53LIX ToF sensor measures distance by transmitting a class 1 pulsed laser at a wavelength of 940nm. Once the pulse is transmitted, it reflects off of objects back to the sensor. The time taken between the transmission is used to measure the distance between the object and the sensor using the equation$distance\ =\ speed\ of\ light(\frac{travel\ time}{2})(mm)$. For instance, if the travel time was measured to be 340ns, then the distance comes out to be $d\ =\ 3\ *\ 10^{8}(\frac{340ns}{2})\ =\ 51mm$. Furthermore, the VL53LIX sensor offers 3 modes of operation that differentiate in distance limit and resilience to ambient light. The modes of operation have distance limits of 150cm, 290cm, and 400cm, respectively. But, as the distance limit increases, the resistance to ambient light decreases. For this device, the ToF sensor was set to the default mode of 400cm maximum distance. However, before any measurement can be taken, a connection between the microcontroller and the sensor must be established. This connection is built when the MCU receives the model ID and module type of the sensor using I2C. Now that the connection is established, the final step before measuring distance is to configure the stepper motor.

The stepper motor consists of four coils, a center tap, and eight teeth per coil. The center tap is connected to all four coils and 5V. To move the rotor inside, a coil is pulled to ground causing a magnetic field that moves the magnetic rotor to one adjacent tooth. Thus for a single rotation, the coils must be pulled to ground 32 times. Furthermore, to drive the rotor in a specific direction, the coils are pulled to ground in a specific order. Programming-wise, this means outputting the appropriate sequence of bits to the motor driver. For example, to rotate counterclockwise in full-step mode the bit sequence is 1001, 0011, 0110, 1100. Moreover, The outer shaft and the inner rotor are interconnected through a set of gears that have a gear ratio of 1:64. Thus, for one complete rotation of the shaft, the rotor must spin 64 times and the motor driver must output the 4-bit sequence 2048 times in the correct order. This concept assists in determining the angle of rotation reached by the motor which is governed by the equation $\theta\ =\ \frac{number\ of\ steps\ reached}{2048}\ *\ 360$. For example, if the number of steps reached is 256, then the angle reached is $\theta\ =\ \frac{256}{2048}\ *\ 360\ =\ 45\ \circ$. Finally, the angular speed of the motor is determined by the delay between each sequence. The motor in this device rotates both clockwise and counterclockwise at an angular speed of 187.5RPM.

Now all the components for measurements are ready. First, the VL53L1X_StartRanging() function is called for the sensor to start ranging. The motor begins to spin and 0x2 is outputted to

port N data to light the PN1 LED. Then, using polling, the program continuously checks if the specified angle has been reached. Once the angle is reached, the range status is determined and the distance is measured and transmitted from ToF to MCU, then from MCU to PC. This sequence of events recurs until a complete cycle has occurred, at which the measuring stops and the motor rotates back to its original position.

The data transmission process is separated into two segments. The ToF to MCU and MCU to PC data transmission. First, the data is transmitted between the sensor and the MCU using the I2C protocol. The I2C interface uses two signal wires, the SDA and SCL, to exchange information. The Serial Data Line(SDA) is used for sending and receiving data and the Serial Clock Line(SCL) sends the clock signal for synchronous data transfer. For a single distance measurement, 16 bits are transmitted from the ToF to the MCU. The data transmission from MCU to PC is done through the Universal Asynchronous Receiver/Transmitter(UART). For asynchronous transfer, there's no clock signal to synchronize the output of the bits. Instead, the two systems communicate on an agreed baud rate of bits per second. Also, each PC has specific communication lines for the transfer of data. For this device, the baud rate is 115200 and the communication line is COM3.

The data receiving, processing, and storing occur in python. As stated before, the communication line used in data transfer is COM3 at a baud rate of 115200. Thus the program opens the communication line COM3 at the specified baud rate of 115200. Once the button is pressed and the distance measurement is taken, the data is received to be processed and stored in an xyz data file. When one byte of data is received, the python serial library functions read() and decode() are used to read and convert the byte of data into a string. Then to separate the distance measurement from the rest of the input, the string is split at space characters using the function split(). To separate printed messages from actual distance measurements, the isdigit() function is called. Now that the distance measurement is separated, it's type casted into an integer using the function int(). Using this integer and the angle of measurement, the y and z measurement are calculated by calling sin() and cos() functions. Finally, the x, y, and z measurements are stored in the file in the form "x y z\n".

3.2 Visualization

Once the data is stored using the python program, it's now stored in the form " x y z\n". Using the load() function in MATLAB, the measured data is put into a Ax3 matrix, where A represents the number of measurements taken. Each of the 3 columns of this matrix represents a dimension where 1,2, and 3 represent x, y, and z. Now that the data is separated into individual dimensions, the function plot3() is used to visualize the data. The function plot3 creates a 3 dimensional space according to the given axis. Then plots each point on the graph while connecting each point with the previous point using a straight line. This means that for a single slice, all the point will be connected except for the first and last measurement. To overcome this

gap, the first measurement of each slice is appended after the last measurement. To implement this, a for loop iterates in the range of number of slices, in this case 10 slices, and copies a slice into a new matrix. Then, the initial measurement is copied to the next position, and the process repeats for all of the slices. At this point, the data is almost ready for visualization, the final step is to ensure the units of the measurements match. From the python processing, the y and z measurements are in millimeters, but the x dimension is in centimeters, thus it's multiplied by 100. The data is now ready for graphing. The axes are labelled as "dimension(mm)" and plot3() is called to visualize the data.

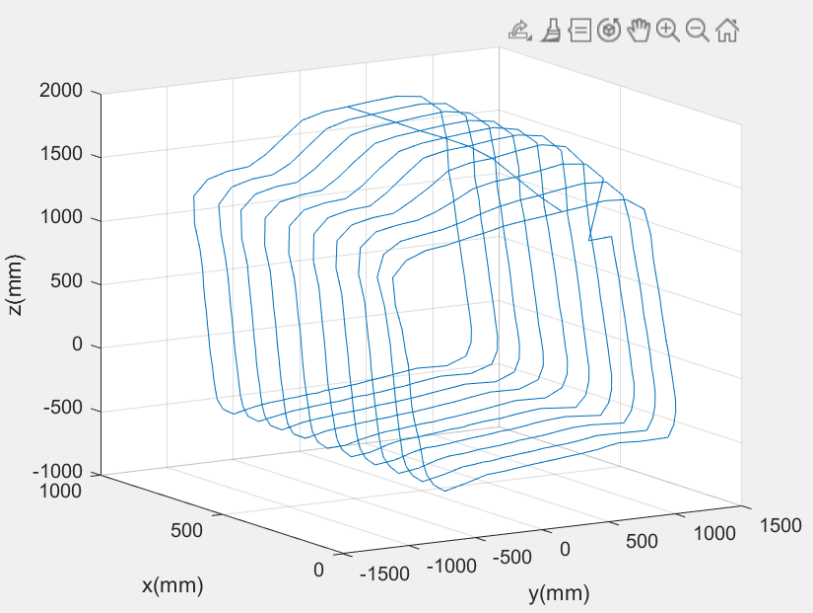

Figure 3.1: Visual of Sample Data at 5.625 degress

5.0 Limitations

1) The limitations of the microcontroller floating-point capabilities have minor effects on the data results. This limitation is not only specific to our microcontroller, but to all binary computers. Because computers operate in binary and natural floating-point numbers are in decimal, the floating-point numbers cannot be represented accurately in computers. For this device, this limitation is observed in the use of trigonometric functions to calculate the y and z values. Thus, the y and z measurements cannot be represented exactly. However, in comparing the theoretical value with the calculated one, it's evident that this limitation has minor effects on the results.
2) The quantization error for the ToF sensor is the data range divided by the bit rate of the sensor which is given by $error = \frac{range}{2^n} * 100$where n is the number of bits. For this

sensor, we have a range of 4000mm and 16 bits for single distance measurement. Thus the quantization error is equal to $error = \frac{4000}{2^{16}} * 100 \approx 6.1\%$.

3) The maximum standard serial communication rate that can be implemented with the PC is 128kBps. This value can be verified by opening the task manager of the PC.
4) The communications methods are split into two segments. The ToF to MCU data communication which uses I2C and MCU to PC which utilizes UART for communication. The speed at which the I2C communicates with the MCU is the bus speed of 100kBps and the speed of the UART communication is 115200Bps
5) The element that acts as the primary limiter on speed is the time of flight sensor as its data communication rate is 100kBps compared to the 115200Bps of the UART communication protocol between the MCU and PC.

6.0 Circuit Schematic

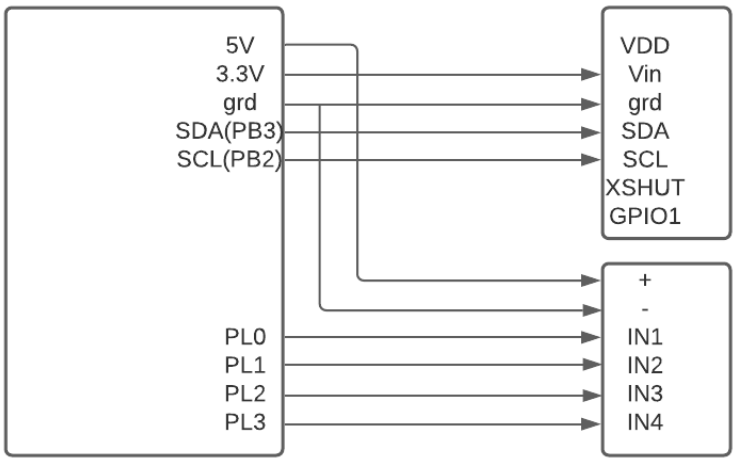

Figure 6.1: Circuit Schematic of Device

7.0 Logic Flow Charts

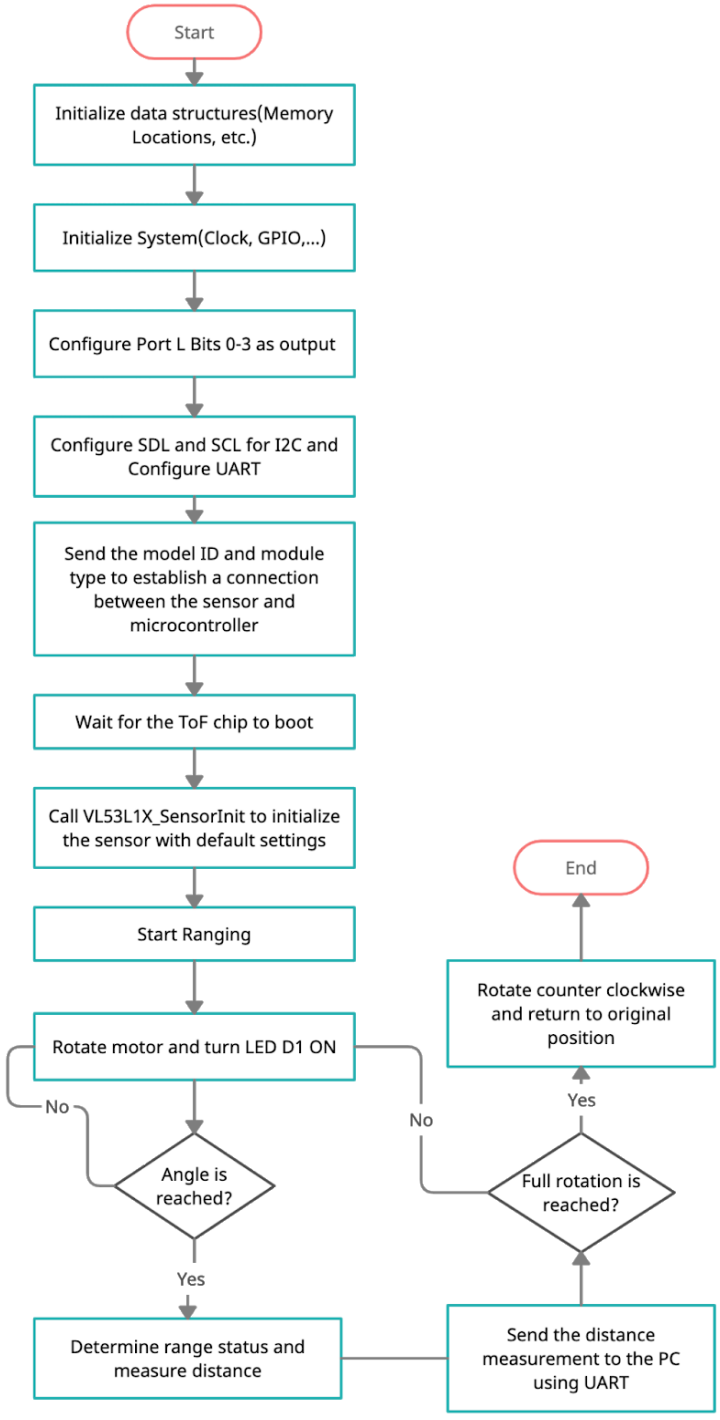

Figure 7.1: Microcontroller Flow Chart

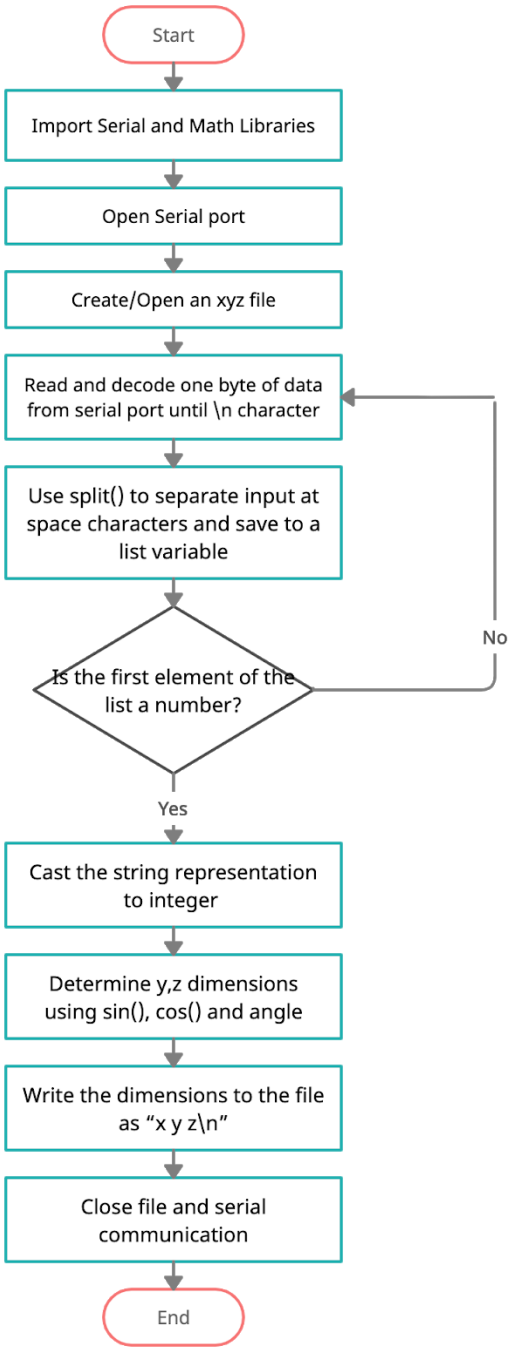

Figure 7.2: Python Code Flowchart

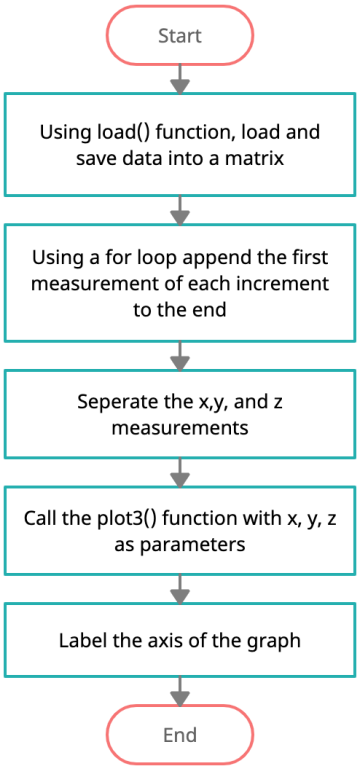

Figure 7.3: MATLAB code flowchart