2DX: Microprocessor Systems Final Project Report

Instructors: Drs. Boursalie, Doyle, and Haddara

Dayyan Hashmi – hashmid – 400317085 – 2DX3 – Monday Afternoon – L01

Device Overview

Features:

This device includes these features:

- Enclosed in a box as one package
  - Size: 30x30x6 cm
  - Material: Cardboard
- Simple Operation
  - One button completes the operation of the device
  - Button to reset the operation of the device.
- Uses a Time-Of-Flight sensor (VL53L1X) to take distance measurements up to a full 360-degree rotation. [1]
  - Takes 10 distance measurements across the 360
  - Up to 400 cm distance ranging
  - Up to 50 Hz ranging frequency
  - I²C interface
- Open Source:
  - Code is fully available to edit and configure written in C and Python
  - Parts used are non-proprietary and easily repairable.
- Uses a MSP432E401Y Microcontroller to process [2]
  - Bus Speed: 48 MHZ down from 120-MHZ
  - Operating Voltage: 3V – 5V
  - Memory Resources: 1024KB of Flash Memory and 256KB of SRAM
  - Baud rate: 7.5 Mbps
- Approximate Cost of System: 30 USD (Estimate based on cost of microcontroller, sensor, and wires)

General Description:

This device is an encompassed and complete embedded spatial measurement system that uses a Time-Of-Flight sensor (VL53L1X) to acquire distance measurements and processes the data to model a 3-dimensional area. The main components of this device are:

- Digital I/O: A push button to activate the data acquisition of the device, an LED to indicate the operation of the device and the recording of each measurement.
- Transducer: Time-Of-Flight sensor that measures the distance values across the y-z plane.
- Data Processing: Collects coordinate values using the ToF sensor, computes them into their Y-Z values.
- Polling based device
- Control: Stepper motor rotation controlled for 360 degree rotation to support the ToF sensor
- Communicate: Communication between the ToF sensor and Microcontroller. Communication of data to the PC application. Communication of the data to graphically view the data as a 3D model in Python and Excel.

This device goes through several stages in its operation. First is the quantification of the analog signal. The signal is taken with a transducer. Then it is processed from an analog to digital signal after it has been preconditioned. After receiving the measurement from the TOF sensor, is it send to the PC program to be processed into an excel file to be visualized. The data is transmitted to the program using UART serial communication one byte at a time. The program is operating in Polling mode and send the data to a Python script file that allows the values to be stored in Excel to be processed into a 3D graph. Below (Figure 1) is a figure that showcases the entire device. Also below (Figure 2) is a depiction of the process described in a block diagram format to better visualize the devices operation.

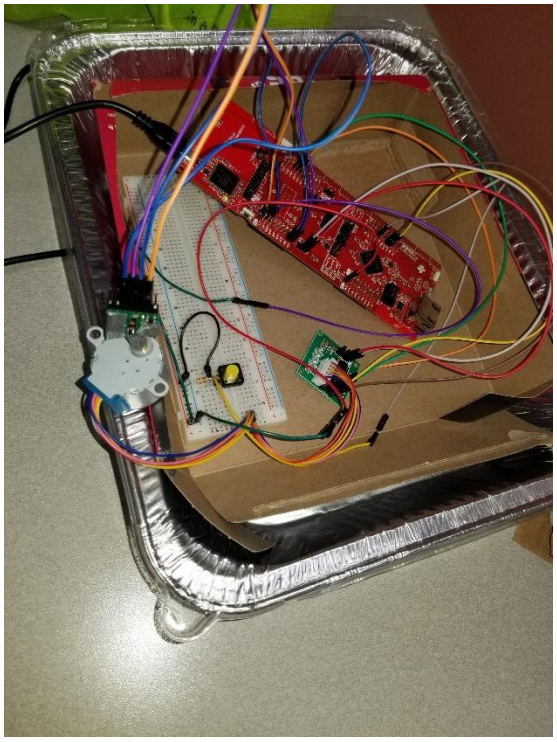

Figure 1: Showcase of the entire device

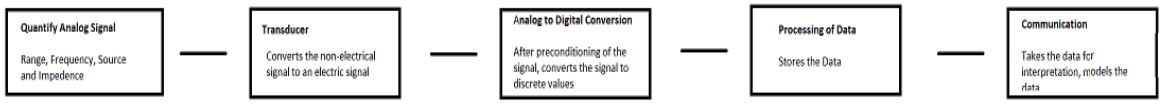

Figure 2: Block Diagram Visualization of the Devices Operation

Device Characteristics

Below is a table (Figure 3) featuring the device characteristics:

Characteristic	Defined Value/Detail
Stepper Motor Pins	PH0 – PH3
Push Button Pins	PE0, PM0
ToF Sensor Pins	PB2 (SCL), PB3 (SDA)

Bus Speed	48 MHZ
Communication Port	COM3 (Dependent on PC)
Communication Speed (Baud Rate)	115200 bps
Libraries	<stdint.h>

Figure 3: Table of Device Characteristics

Detailed Description of Device

Distance Measurement:

While collecting measurements of distance, it is imperative that one knows the specific steps involved with the process of acquiring a applicable output value. The first step requires the device to have a transducer which is a component that collects a non-electrical signal converts it to an electrical signal. This electric signal gets passed onto a pre-conditioning circuit which is able to filter the signal which is able to be processed by the analog to digital converter. The (ADC) is what converts the actual analog signal to a digital signal. This can be done on the microcontroller provided above however it is already done in the VL53L1X sensor [1]. The sensor fires a light signal upon it returning to the sensor calculates the distance from the amount of time it takes for it to return (returned value in mm) [1]. In terms of the device, the stepper motor after a button has been pressed to initialize the device (in Polling mode) will begin to rotate 360 degrees where the sensor has been programmed to take 10 measurements across the rotation. The data is sent to the PC via UART communication and will receive the data through a Python script. This python script will receive the data through the Communication Port 3 (however it will be different depending on the PC) and will transmit at a baud rate of 115200 bps. The data will be written to a XLSX format file using the Python library xlsxwriter. After receiving the data, the excel file will automatically calculate the y and z values from the raw measurements. A trigonometric based formula is used to convert the values of y and z based on the ToF value stored in excel and the angle value that was also transmitted previously. The formula for the conversion factor is listed below. Note that r is the value from the ToF transmission.

$$Y = r * \sin(angle)$$

$$Z = r * \cos(angle)$$

After converting these values into their respective y and z values. The excel file will automatically generate a 3D graph to visualize the given results.

Visualization:

The computer that the visualization was conducted on is a Acer Aspire 3 (AMD Ryzen 5 3500U with Vega Mobile Graphics) running Windows 10 and Python 3.8. The following Python libraries were used:

- Math Library for trigonometric calculations of converting y-z values
- Xlsxwriter Library to write to Excel

To store the required data acquired from the sensor, an Excel file (xlsx) is used with the xlsxwriter library to store the values in it. This distance data includes the measurement value and the angle that it was taken at. The data is then converted like mentioned previously into

its y-z form and then visualized in a 3D graph in Excel. This can be repeated for each x value that one would like to capture at the increments of their choosing. Below (Figure 4) is a dimensioned sketch that is an example hallway of the space that was recreated. In addition to that Section 7 (Figure 7) is a flowchart that describes the Python scripts operation process discussed in this paragraph.

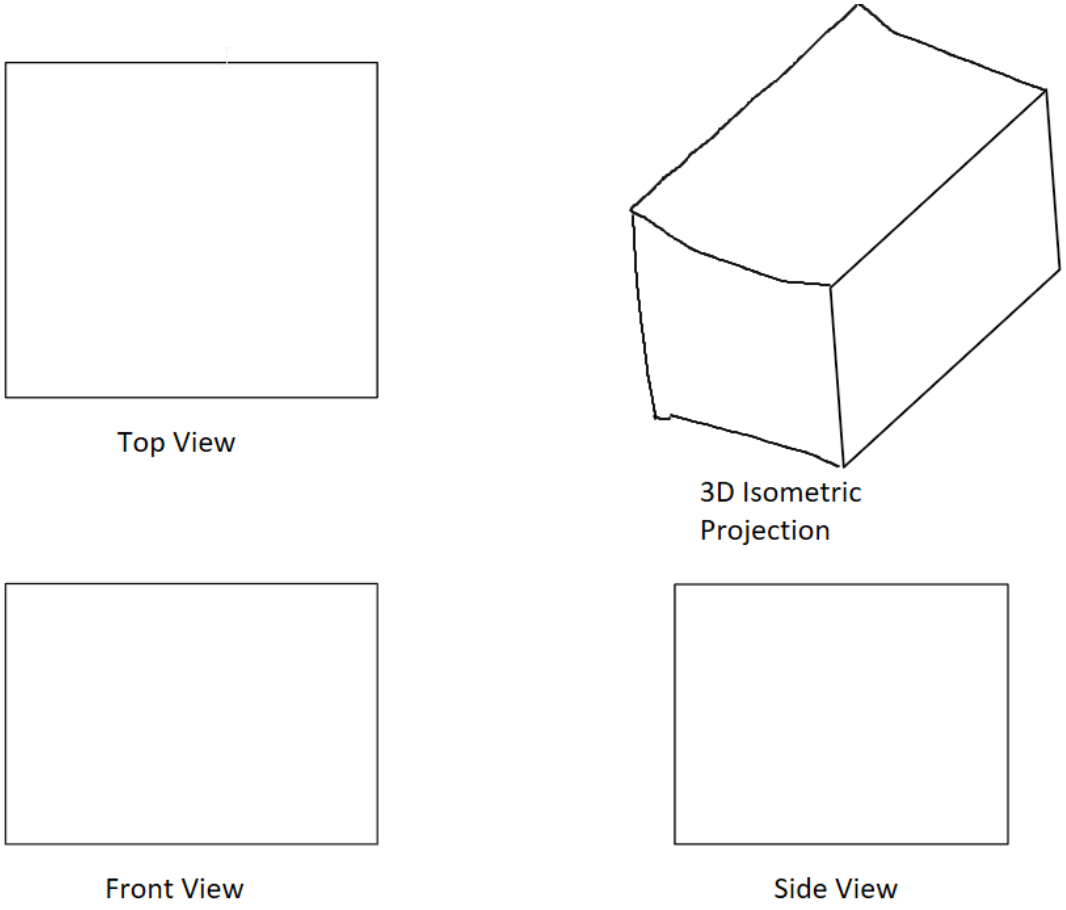

Figure 4: Orthographic and Geometric Drawing of Hallway

Application Example of Device

Example:

This is a detailed example of how to run the device and acquire a signal and map it to Excel.

1. Place the device at the beginning of the measurement plane (ie at one end of the hallway)
2. Run the Python script on the PC but do not press enter to begin communication.
3. Flash the C program onto the microcontroller
4. Reset the microcontroller with the button on the back
5. Press enter on the python script
6. Press the push button to begin the devices operation. It will spin 360 degrees clockwise, collect the data, and then spin 360 degrees in the counter clockwise direction.
7. Move the device by amount needed (example 0.5m)

8. Repeat steps from 2, and 4-7 for desired amount of measurements (for example, 15 measurements)
9. Finally open the excel file to see the final visualization in a 3D graph.

Setup Guide:

This is a guide on how to setup the device for use:

1. Assemble the circuit as shown in Section 6 Figure 5. The pin requirements are in Section 2 Figure 3.
2. Ensure the ToF sensor is attached correctly and double check wiring.
3. Flash the C code provided onto the microcontroller. A detailed flowchart can be found in Section 7 Figure 6
4. Find your serial communication port in device manager and change it in the Python script file.
5. Run the Python script file

XYZ Plane Definition:

The coordinates are X,Y and Z. The X direction is parallel to the users eyes and is what is the independent variable that is changed and incremented. The Y direction points up and the Z direction is right and left. The measurements are on the Y-Z plane.

Limitations of Device Documentation

Below are some limitations that were considered:

1. The MSP432 microcontroller has a 32-bit FPU integrated into the Cortex-M4F Core Processor. This means that it can support floating points but only up to 32 bits. The microcontroller can do trigonometric function calculations however it is much easier to implement this in Python and can then use the processing power of the PC.
2. To calculate the maximum quantization error of the ToF modules, it can first be noted that it is equal to the resolution. $\frac{VFs}{2^m}$ is the resolution formula where m is the number of bits and VFs is the full scale voltage. The ToF sensor has an 8-bit ADC and the VFs is 5 volts therefore: $ToF\ resolution = \frac{5}{2^8} = 19.53\ mV$
3. The maximum standard serial communication rate that one can implement with the PC is the one that is used in the project for its communication. This is 115200 bps. To verify, one can use RealTerm (a PC application) and brute force higher values then this. Errors and misvalues were the result of trying values above this.
4. The device uses I²C to communicate between the ToF sensor and the PC. The MSP432E401Y has eight UARTs, but only one was used for asynchronous transmission. The speed was 115200 bps on COM3 port of the PC sending one bit at a time.
5. The primary limitation on speed for the entire system is the speed of the microcontroller. This is tested by gradually upping the value of the Bus Speed on the microcontroller and recognizing the gradual speed improvements of the device.

Circuit Schematic

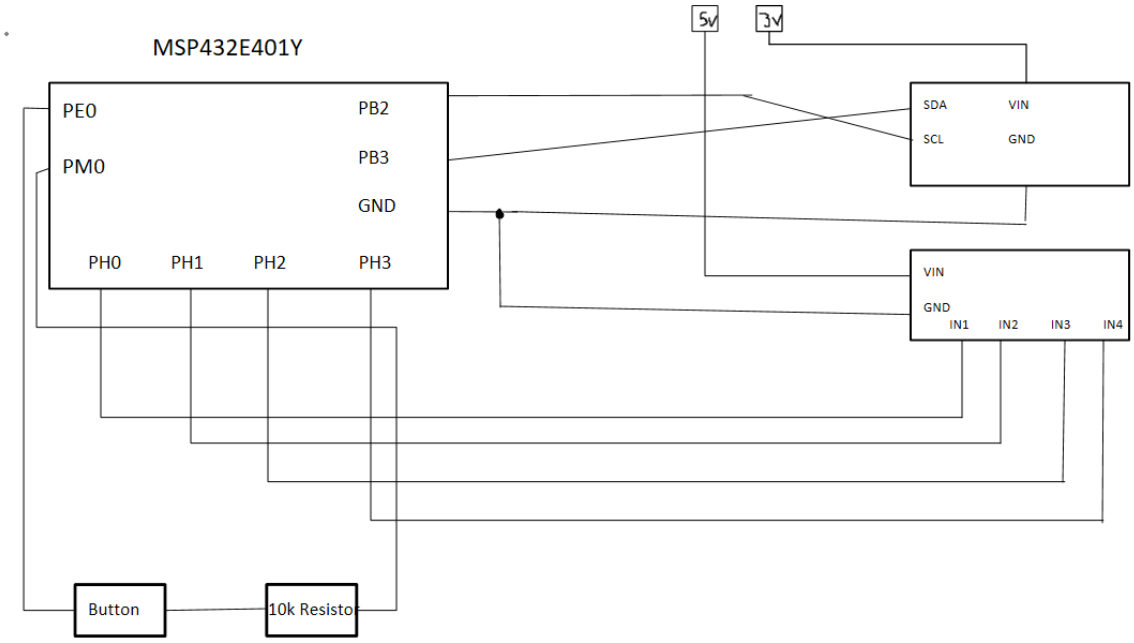

Figure 5: Circuit Schematic

Flowcharts of Programming Logic

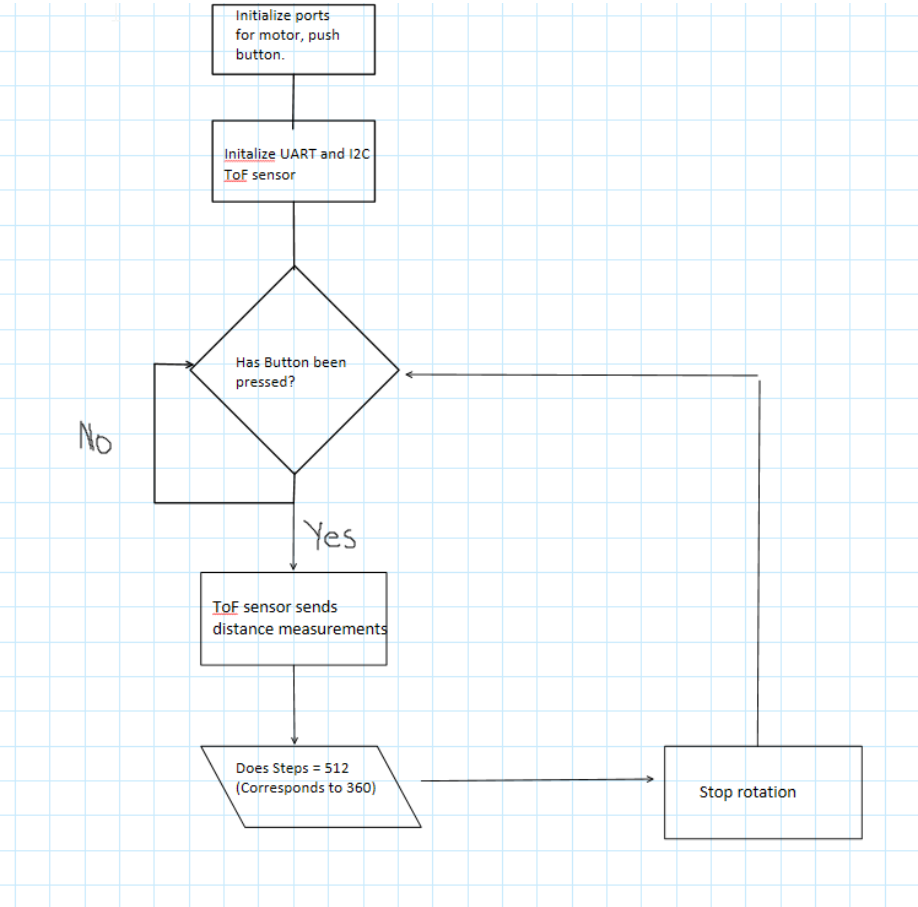

Figure 6: Microcontroller Flowchart

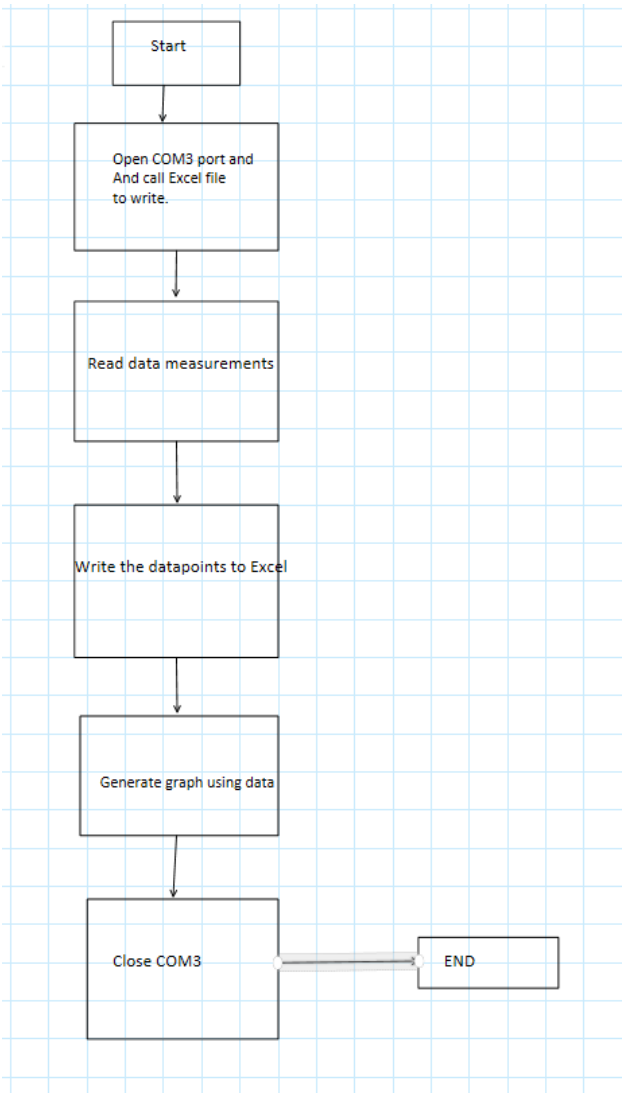

Figure 7: Python script and visualization flowchart

References:

[1] STMicroelectronics,"VL531LX", 5th ed, April 2022

[2] MSP432E401Y - Microcontroller Data Sheet, 1st ed., Texas Instruments, Dallas, TX, USA, 2017, pp.