2DX3 Project Report
Spatial Scanner Datasheet

Michael Mota – 400449867 – motam1

Date: April 17, 2024

As a future member of the engineering profession, the student is responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is our own and adheres to the Academic Integrity Policy of McMaster University and the Code of Conduct of the Professional Engineers of Ontario. **[Michael Mota, 400449867]**

Device Overview

Features:

- The system utilises a VL53L1X ToF Sensor and a Stepper Motor to acquire 360-degree spatial scans along a chosen geometric plane.
- 16 coordinate datapoints collected at equal intervals for each 360-degree scan (22.5 degree difference)
- The system takes three scans of the chosen plane per program execution, each orthogonally displaced 1cm from the previous scan.
- Three distance settings able to measure up to a maximum of 360cm in dark condition and 135cm with the presence of strong ambient light.
- Digital IO port PM1 is connected to a push button and used to start the execution of each scan.
- Digital IO LEDs on the microcontroller are used to as status indicators
  - Onboard LED D4 blinks when a 360-degree scan is completed.
  - Onboard LED D3 blinks each time a distance measurement is taken.
- The microcontroller facilitates coordinate computation and collection of distance coordinates.
- Serial communication between the TOF sensor and microcontroller is implemented using I2C. Communication between the microcontroller and PC is done using UART protocols.
- Provided Python script gathers collected coordinate data and displays a visual mapping of the data using Open3D.

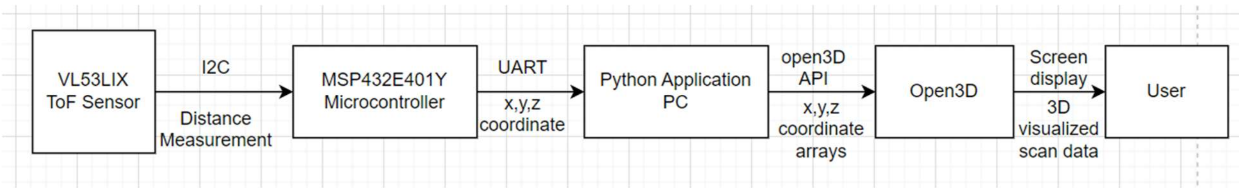
General Description:

The spatial scanner is designed to provide a simple and cost-effective 3D mapping device for indoor uses. The system is built around a MSP432E401Y microcontroller which orchestrates the operation of a stepper motor and ToF sensor.

The stepper motor rotates in 360-degree cycles, stopping every 22.5 degrees to allow the ToF sensor the collect a distance measurement. The scanner is fixed along a point on a chosen measurement plane. Each 360-degree scan should be done with a 1cm orthogonal displacement from the original point. Distance measurements are collected by the microcontroller using I2C protocols. The measurement are converted to x,y,z coordinates and sent to a connected PC using a UART COM port (set to COM3 by default). The coordinates are then visualised using open3D API within a python script.

LED indicators onboard the microcontroller are used to indicate each instance of distance collection and the completion of a 360-degree scan. Each scan must be started by pressing an external push button connected to GPIO PM1.

Block Diagram:



Device Characteristic Table

Table 1: Device Characteristic Table

MSP432E401Y		**VL53L1X**		**Stepper Motor**	
Characteristic	Value	Characteristic	Value	Characteristic	Value
Bus Speed	16MHz	Vin	3.3V	V+	5V
LED D4 GPIO	PF0	Gnd	Gnd	V-	Gnd
LED D3 GPIO	PF4	SDA	PB3	IN1	PH0
Button GPIO	PM1	SCL	PB2	IN2	PH1
Baud Rate	115200bps	Max Range (dark)	360cm	IN3	PH2
PC		Max Range (light)	135cm	IN4	PH3
Characteristic	Value				
UART COM Port (default)	COM3				
Baud Rate	115200bps				

Detailed Description

Distance Measurement

The distance measurements are taken from the VL53L1X time of flight sensor. The ToF sensor uses Light Detection and Ranging (LIDAR) to measure distances. It works by emitting a pulse of light (940nm) from an emitter and measuring the time it takes for that pulse of light to bounce off a surface and return to a detector. Multiplying the measured time with the known speed of light, the ToF sensor can determine the distance travelled by the light. However, the distance travelled must be divided by two, because the light travels from the sensor to the object and then back again. This leaves the equation for distance as.

$$Distance = (Photon\ Travel\ Time * \ Speed\ of\ Light)/2$$

We must also consider the VL53L1X limitations. The ToF sensor has three different distance modes (Short, Medium, Long) which vary in performance depending on the conditions. When the surroundings are dark, long-range mode performs much better (longer maximum distance). However, when there is strong ambient light, short-range mode performs better (longer maximum distance). The mode can be changed within the Keil code.

The accuracy of a distance measurement can be judged by considering the range status value. This built in functionary returns a status of either 0, 1, 2, 4 or 7. A value of 0 means that there is no error in the distance reading. A value of 1 or 2 represents a warning that there might be some error in the distance reading. A value of 4 or 7 indicates that the distance readings are operating outside the maximum ranges.

Visualization

Once the ToF sensor communicates the distance data to the microcontroller using I2C. The data must then be converted to x and y coordinate values. These values map the scanned plane. This is done by keeping an angle variable stored on the micro. This variable keeps track of the angle of the stepper motor throughout the 360-degree scan. X and Y values are calculated using the following formulas.

$$X = Distance * \sin(angle)$$

$$Y = Distance * \cos(angle)$$

The Z value begins at 0 and is iterated by 1cm when each 360-degree scan is completed.

These x,y,z values are then sent to a connected PC over UART (Port COM3 by default). They are sent in the form “x, y, z”. A python script is implemented to read the data and strip it at the commas. The stripped data is then appended to arrays for x,y and z respectively. This process repeats for all coordinate data received. Once all the data has been received, the open3D API is used the connect the lines of the data points along the x,y plane. This creates a 3D visualization of the distance data.

Application Note

To use the spatial scanner, you must have access to Keil, Python and Python libraries such as “serial”, “numpy” and “open3d”. The default UART COM port is set to COM3. This value can easily be changed at the top of the Python script. The spatial scanner is meant to be used in enclosed locations. It performs best with low lighting. The distance mode is set to long by default. This mode can be changed on line 146 of the main Keil code. If you believe there are errors in the distance readings (perhaps the scan is not correct), consider uncommenting lines 206-208 within the Keil code to print the range status.

Instructions

1. Connect the microcontroller to a PC using USB cable.
2. Connect the stepper motor to the associated pins on the microcontroller.
    - V+ to 5V

- V- to Gnd
- IN 1-4 to PH 0-3 respectively

3. Connect the ToF sensor to the associated pins on the microcontroller.
   - Vin to 3.3V
   - Gnd to Gnd
   - SDA to PB3
   - SCL to PB2
4. Connect an active low push button to PM1 on the micro. Don't forget to connect a source (3.3V) and ground (Gnd). A pullup resistor is already set for PM1.
5. Open and run the Python script.
6. Open Keil and Translate, Build, Download and Load the project onto the micro.
7. Press enter to commence the program
8. Press the push button to start each scan (even the 1st one).

Expected Output

The expected output is a 3d scan of the measured plane to appear on open3d.

For example,

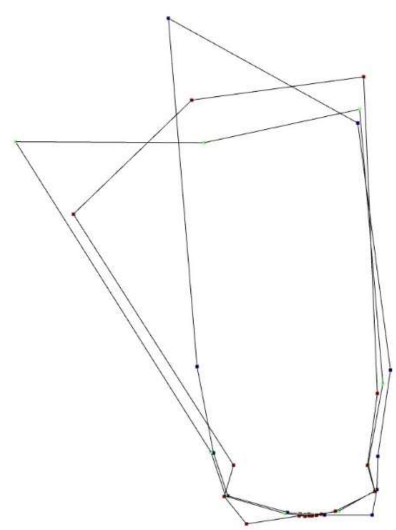


Figure 1: 3d Scan of Given Hallway

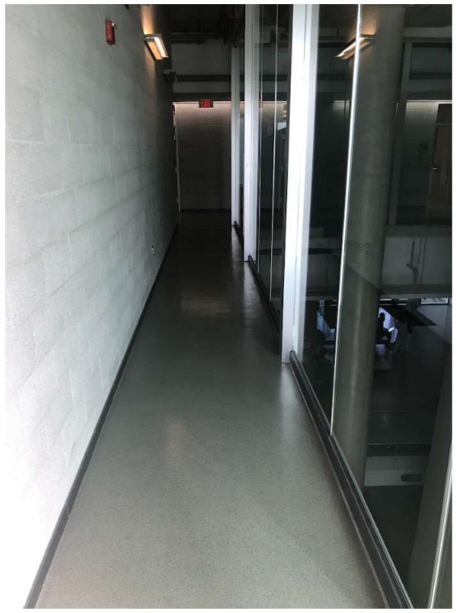


Figure 2: Image of Given Hallway

Limitations

A main limitation of the design is its range. Even in dark conditions, it only has a maximum range of 360cm. This limits it's use to small indoor spaces.

Time is also a limitation. Most of the scan time is spent rotating the stepper motor. This is very apparent by just observing the scan execution. However, if I was unsure, it could be easily verified with the AD2. While the stepper motor spin code block is running, I would set a GPIO port to high. When the code block ends, I would set the port to low. I could then connect the port to my AD2 and observe the Thigh and Tlow values and compare.

Another limitation is the quantization error of the ToF distance readings. The distance readings are stored as a 16-bit ADC value. Using the max distance value of 360cm, we can calculate the maximum quantization error.

$$Max\ Quantization\ Error = \frac{360}{2^{16}} = 5.49 * 10^{-3} cm$$

My given bus speed was 16MHz. To set this bus speed, I changed the PSYDIV value within PLL.h to 29. I determined the value for PSYDIV by using the following equation.

$$Bus\ Speed = \frac{480MHz}{PSYDIV + 1}$$

$$PSYDIV = \frac{480MHz}{Bus\ Speed} - 1 = \frac{480MHz}{16MHz} - 1 = 29$$

Circuit Schematic

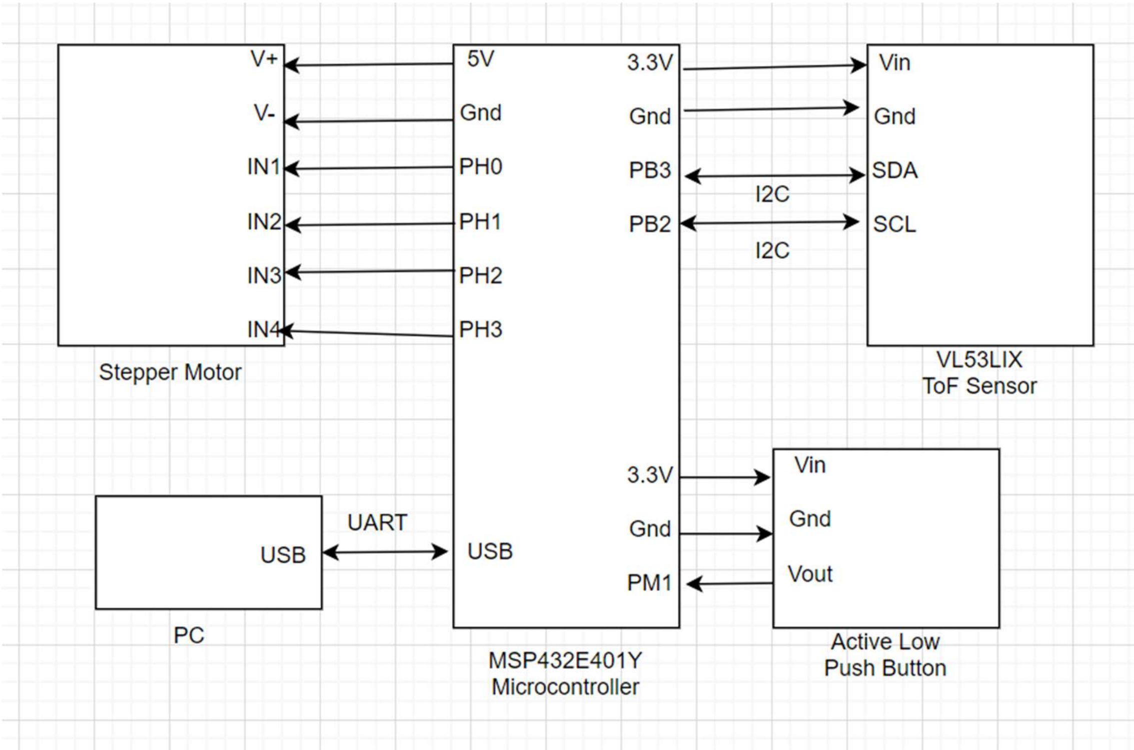


Figure 3: Circuit Schematic

Programming Logic Flowchart

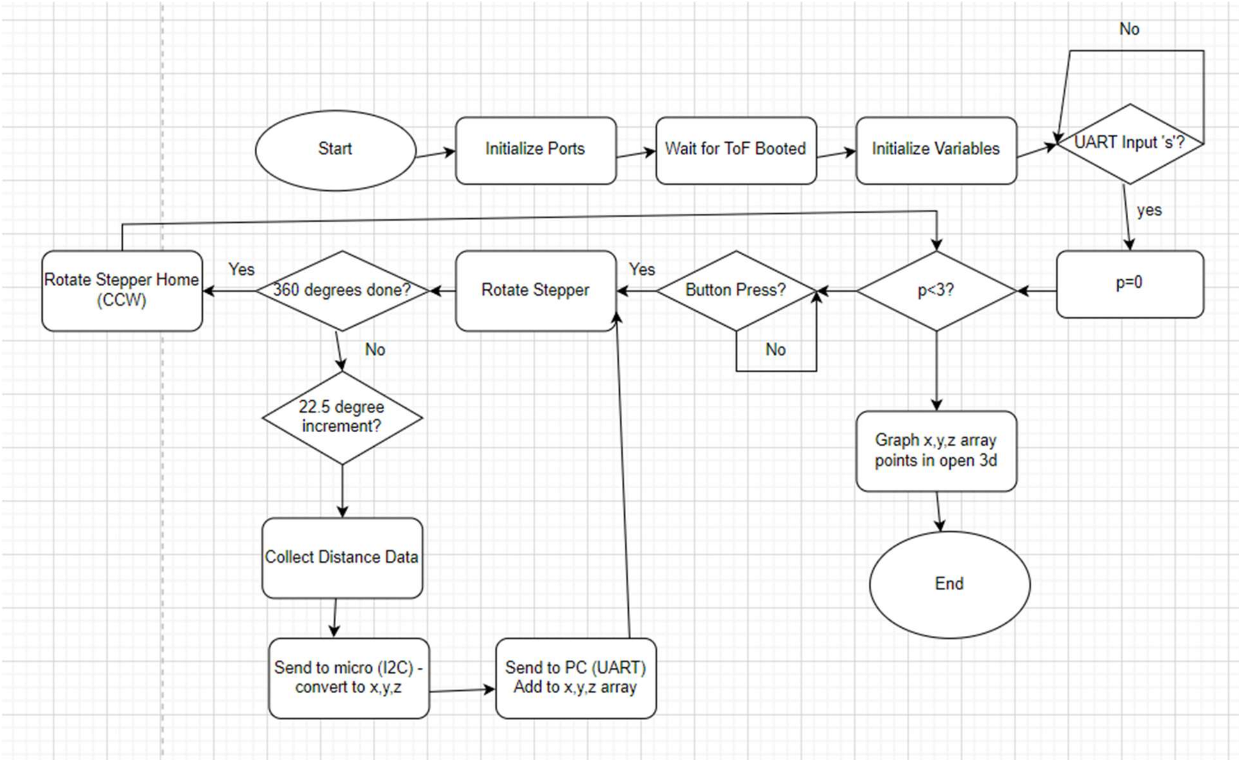


Figure 4: Programming Logic Flowchart

References

[1] “Time-of-Flight (ToF) ranging sensor based on ST’s FlightSense technology.” ST. https://www.st.com/en/imaging-and-photonics-solutions/vl53l1x.html [accessed Apr. 17, 2024].