Design and Implementation of a Time-of-Flight–Based 3D Environment Mapper

COMPENG 2DX3, 2024

McMaster University

Ian Cho

(1) Device Overview

1.1 Part A — Hardware Summary

This embedded system uses two external push buttons and two assigned on-board LEDs. The core components are:

Microcontroller (TI MSP-EXP432E401Y)

- Bus speed: 60 MHz
- Operating voltage: 3.3 V I/O (5 V tolerant where applicable)
- Memory: 256 KB SRAM, 1024 KB Flash
- Core: 32-bit Arm® Cortex-M4F with single-precision FPU

Time-of-Flight Sensor (VL53L1X)

- Range: up to 4 m
- Sampling rate: ~50 Hz (typical)
- Operating voltage: 2.6–3.5 V
- I^2C address: 0x29 (7-bit)
- I^2C speed used: 400 kHz (Fast-mode)

Stepper motor (28BYJ-48 + driver)

- 512 steps / 360° (0.703125°/step)
- 4 phases with phase LEDs
- Supply used: 5 V (module accepts 5–12 V)

External push buttons

- 3.3 V logic; start/stop scan

Communications

- Sensor ↔ MCU: I^2C @ 400 kHz
- MCU ↔ PC: UART 115200 bps, 8-N-1

Receiver-side (PC) software

- Programming language: Python 3.8

- Libraries: pyserial, numpy, math, open3d

Visualization

- Open3D renders a 3D point cloud/line-set of the scene.

Approx. cost (CAD or USD, typical online prices)

- MSP-EXP432E401Y board: $25–35
- VL53L1X module: $12–18
- 28BYJ-48 + driver: $5–10
- Buttons/jumpers/misc: $3–5
- Total: ~$45–65

1.2 Part B — Functional Summary

The device builds a 3D simulation of an indoor scene. The ToF sensor captures distance along the *y–z* planes each time the stepper motor advances 11.25°. When the start button is pressed, the system enables: LED PN1 flashes, the motor rotates clockwise, and the sensor begins sampling. At each sample, the motor briefly pauses for accuracy; LED PN0 flashes to indicate a measurement. To keep cabling clear, the motor reverses before proceeding to the next position. Timing/delays are governed by the MCU bus speed.

Where the conversion happens: Distance/angle data are sent from the sensor to the MCU over I^2C, forwarded to the PC via UART, and converted from polar → Cartesian on the PC in Python. y,z are computed via trigonometry using the motor angle; x is incremented after each sweep to emulate translation. Open3D then visualizes the point cloud.

1.3 Part C — Block Diagram

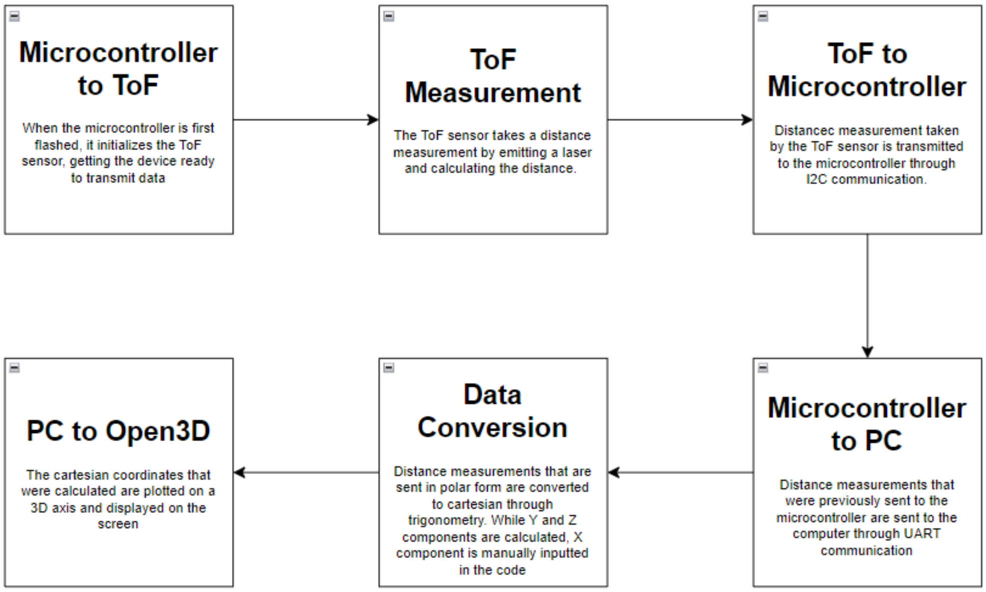

Figure 1. Data-flow block diagram of the system.

2. Device Characteristics

Subsystem	Parameter/Signal	Value/Mapping
Microcontroller	Clock speed	30MHz
	Assigned LEDs	PN0, PN1
	Baud rate	115200 BPS
	Serial Port	COM4
	Push button	PM0
Time of Flight (ToF) sensor	Vin	3.3V
	GND	GND
	SDA	PB3
	SCL	PB2
	I²C addr	0x29
	I²C speed	400 kHz
	Supply	3.3V
Stepper Motor	In1	PH0
	In2	PH1
	In3	PH2
	In4	PH3
	V+	3.3V

	V-	GND
PC (receiver)	Language	Python 3.8
	Libraries	Pyserial, numpy, math, open3d

Table 1: Key settings, characteristics, and connections

3. Detailed Description

3.1 Part A — Sensing & MCU Interface

The VL53L1X ToF sensor computes distance using an infrared laser to detect nearby obstacles, as it emits the beam of light, and waits for the light to be reflected by a nearby obstacle and into the receiver:

$$D = \frac{in\,flight\,time}{2} \times speed\,of\,light$$

Measurements are digitized and sent to the MCU (microcontroller) over I²C. The system is idle until PM0 is pressed. Once active, the stepper rotates in 11.25° increments (512 steps per 360° ⇒ 0.703125°/step ⇒ 16 steps per 11.25°). After each 11.25° advance, PN0 flashes and a reading is taken. In firmware, step increments each motor step; depth increments each full revolution to emulate forward motion down the hallway. Data flows from the sensor (VL53L1X) → MCU (I²C) → PC (UART).

3.2 Part B — Host Processing & Visualization

When & where conversion occurs: Upon receiving each UART frame, the Python program analyzes the values of the variables (r, θ, i) immediately converts to Cartesian:

$$x = i \cdot \Delta x$$

$$y = r \cdot \sin(\theta)$$

$$z = r \cdot \cos(\theta)$$

Example calculation:
Let r =1200 mm, θ = 33.75°, Δx = 50 mm, sweep index I = 7.

$$x = 7 \times 50 = 350\,mm$$

$$y = 1200 \cdot \sin(33.75°) \approx 1200 \cdot 0.557 \approx 668\,mm$$

$$z = 1200 \cdot \cos(33.75°) \approx 1200 \cdot 0.831 \approx 997\,mm$$

The simulation was run on Windows 11 with Python 3.8. Required libraries:

- Pyserial —UART communication with the MCU

- Python math library — trigonometric conversion from polar to Cartesian
- NumPy — point-cloud arrays
- Open3D — point-cloud visualization

Pipeline: the PC receives distance/angle data over UART via PySerial, converts polar → Cartesian (x advanced per sweep; y,z from angle & distance), writes an .xyz point cloud, and renders a line set in Open3D by connecting points within each plane and then between planes. For N sweeps (e.g., 15), the result is N y–z planes × 32 points each, combined into a 3D model.

4. Application Note

Capabilities

- Start/stop acquisition via physical buttons
- LED status for sampling/debug
- Controlled stepper motion synchronized with sensing
- ToF-based y–z measurements and host-side x-axis translation
- Real-time 3D visualization on PC

Operating Summary

1. Power the system and connect to the PC.
2. Use buttons to start/stop scanning and motor rotation.
3. Monitor LEDs for status.
4. View the generated 3D model with Open3D.

Considerations

- MCU floating-point and timing can affect accuracy.
- FPU + trigonometry: MSP432 uses single-precision floats (~7 sig. digits). Repeated sin/cos and cumulative summations can introduce rounding error in y, z, especially at large r.
- ToF quantization and environmental factors introduce error.
- Ensure UART settings (COM port, 115200 bps) match on both ends.
- Validate end-to-end throughput to avoid bottlenecks.

5. Instructions (Quick Start)

1. Connect the MCU to the PC. In Device Manager → Ports (COM & LPT), note the XDS110 Class Application/User UART (COM#).
2. In the Python script, navigate to line 27 to find variable s and set to:

    s = serial.Serial('COM#', 115200, timeout=10)

Save the file.

3. Wire the stepper and ToF sensor per Table 1.
4. In Keil, click Translate → Build → Load.
5. Reset the board to flash the image.
6. Press the start button to begin scanning (PN0 on). Ensure the sensor has a clear line of sight.
7. After each full revolution, move forward one step.
8. Repeat until the desired number of sweeps is captured.
9. Open3D will render the 3D scene from the collected .xyz data.

6. Expected Output

The output should be a 3D outline of the hallway (Hallway II in this project). The simulated model (Figure 2) shows overall geometry, depth, height, and width. The right wall appears longer—and gradually inclines upward—consistent with the real hallway and adjacent stairwell (Figure 3).

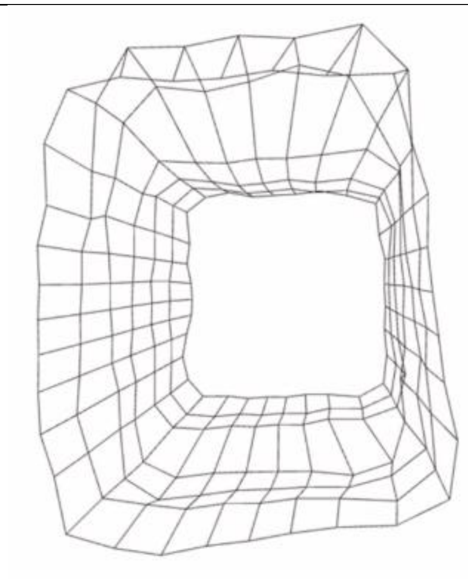

Figure 2. Front view of the 3D simulation

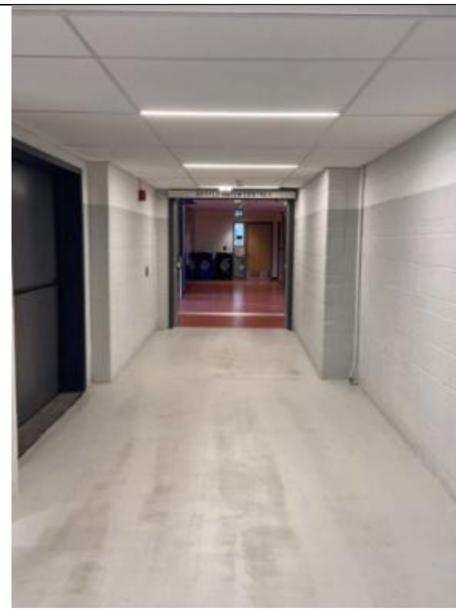

Figure 3. Front view of the physical hallway

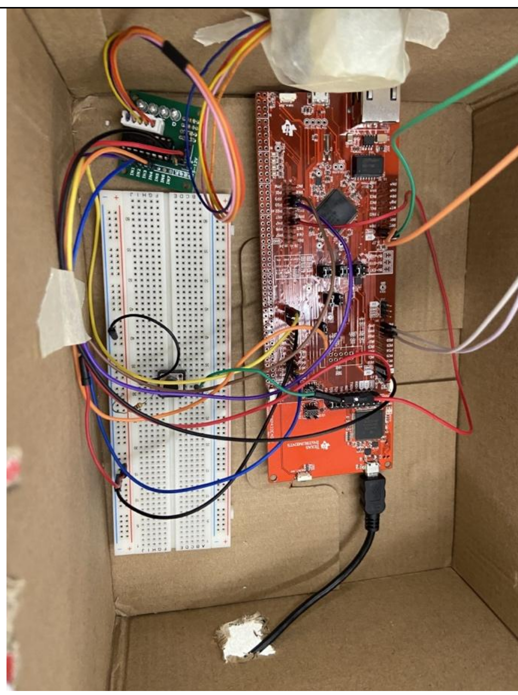

Figure 4: Wiring of the physical circuit

Figure 5: Front view of the box where the ToF sensor scans from

7. Limitations

- Floating-point behavior on the MCU can influence numerical precision.
- Quantization error could occur in ToF during the process of converting data from analog to digital as digital systems can only represent the data to a certain level of detail:

$$Max\ Quantization\ Error\ = \frac{(Max\ reading)}{2^{ADC\ bits}}$$

In this case, it would be equal to:

$$\frac{4000mm}{2^{16}} = 6.10\ \times 10^{-2}mm$$

- Speed limits of both the stepper and ToF sensor depend on delays, bus speed, and mechanical constraints.
- PC serial port may cap standard baud rates (e.g., 128000 bps); ensure settings are compatible.

8. Circuit Schematic

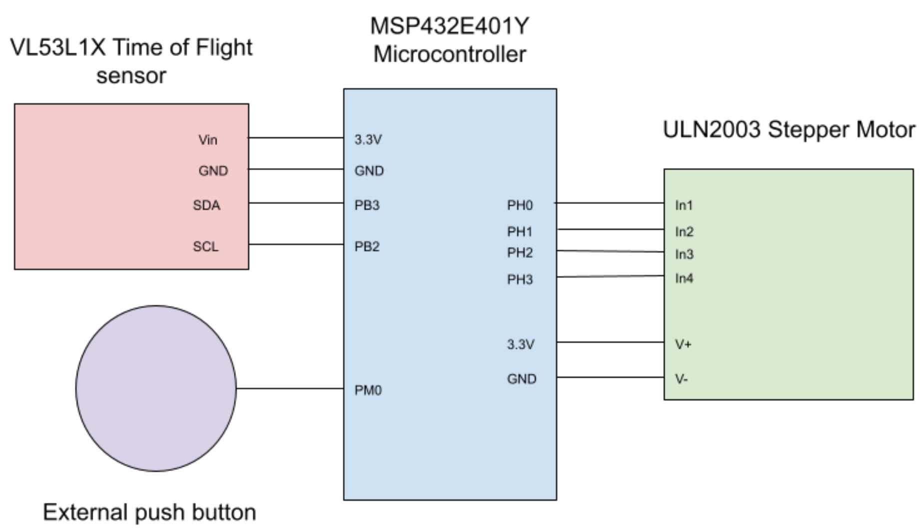

Figure 6. Complete circuit schematic.

9. Programming Logic Flowchart

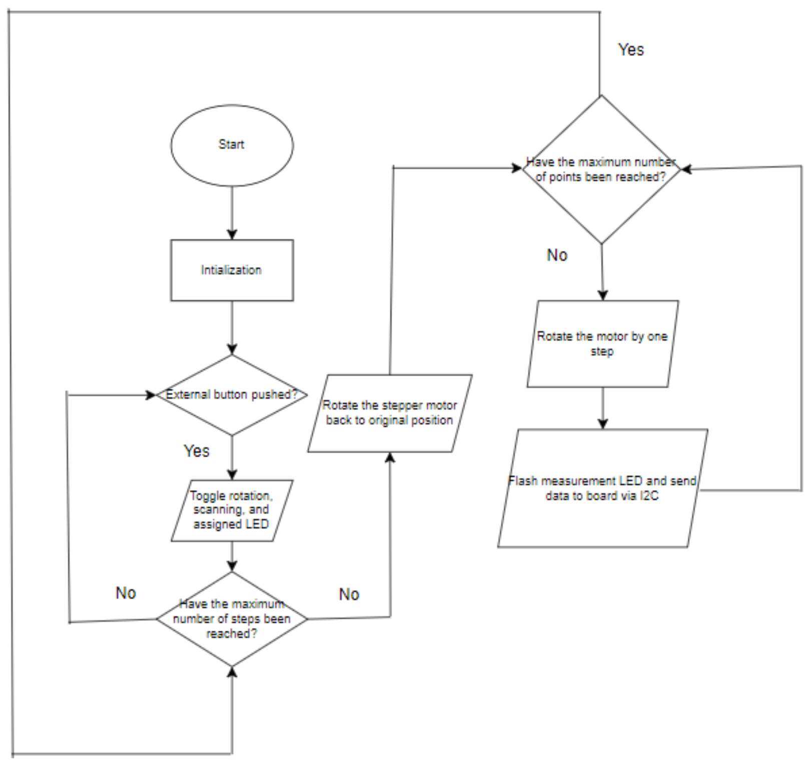

Figure 7. Program flowchart

10. References

[1] “MSP-EXP432E401Y Development kit | TI.com.” https://www.ti.com/tool/MSP-EXP432E401Y [accessed Apr. 17, 2024].

[2] “vl53l1x - COMPENG 2DX3: Microprocessor Systems Project.” https://avenue.cllmcmaster.ca/d2l/le/content /512368/viewContent/3979106/View [accessed Apr. 17, 2024].

[3] “Pololu - VL53L1X Time-of-Flight Distance Sensor Carrier with Voltage Regulator, 400cm Max.” https://www.pololu.com/product/3415/specs [accessed Apr. 17, 2024].

[4] “Analog Discovery 2 - USB Oscilloscope and Logic Analyzer.” https://digilent.com/shop/analog-discovery-2-100ms-s-usb-oscilloscope-logic-analyzer-and-variable-power-supply/ [accessed Apr. 17, 2024].