Device Overview

Microcontroller – TI MSP-EXP432E401Y LaunchPad
• 32-bit ARM Cortex-M4F with floating-point unit (FPU)
• 256 KB SRAM, 1024 KB Flash memory
• System clock: 60 MHz | **Bus speed: 18 MHz** (LSD = 5)
• Interfaces: I2C (sensor) + UART (PC, 115200 baud)
• Operating voltage: 3.3V
• Acts as central controller for motor control and data transmission

Distance Sensor – VL53L1X ToF
• Max range: 4 m | Operating frequency: 50 Hz
• I2C digital communication
• Operating voltage: 2.6V – 3.5V
• Integrated ADC: 16-bit resolution
• Eye-safe, compact, accurate laser-based ranging

Stepper Motor – 28BYJ-48 + ULN2003 Driver
• 4-phase unipolar stepper | 512 steps per 360°
• Step angle: 11.25° | Scan: 32 measurements
• Supply voltage: 5–12V
• Onboard LEDs indicate active phase

Estimated Hardware Cost
• MCU board: ~$35 | VL53L1X: ~$20 | Stepper + Driver: ~$10
• **Total estimated cost: ~$65 CAD**

General Description

This system is carefully developed to perform 360° indoor space distance scanning using a VL53L1X Time-of-Flight (ToF) sensor mounted on a stepper motor. When triggered by an onboard push button (PJ0), the microcontroller (TM4C1294) initiates the series of scans in steps of 11.25°, resulting in a total of 32 measurements for a full circle. At each stepping step, the VL53L1X sends out an infrared pulse and then digitizes the time-of-flight value as digital distance via its on-chip ADC with 16-bit resolution. The sensor is communicated with using the I2C protocol, and motor stepping is controlled using a ULN2003 driver based on GPIO. After reading a distance value, it is sent via UART at a baud rate of 115200 as an angle-distance pair to be processed on the PC side using Python code that translates polar coordinates to Cartesian coordinates (x, y) and visualizes the resulting point cloud using Open3D. Real-time visual indications are given by LED indicators: PF4 stays lit constantly during the scan period, and PN1 blinks after each step to indicate successful data reading. An overview of the whole sequence of data flow is given in the block diagram, which clarifies how the sensor input is interfaced with microcontroller control schemes, UART transmission, and visualization on the PC side. The PC-

side receiver code is written in Python, which handles serial parsing, angle-distance conversion, and rendering. Additionally, the **XSHUT** pin is toggled at the start of each scan to reset the VL53L1X sensor, ensuring consistent measurement reliability.

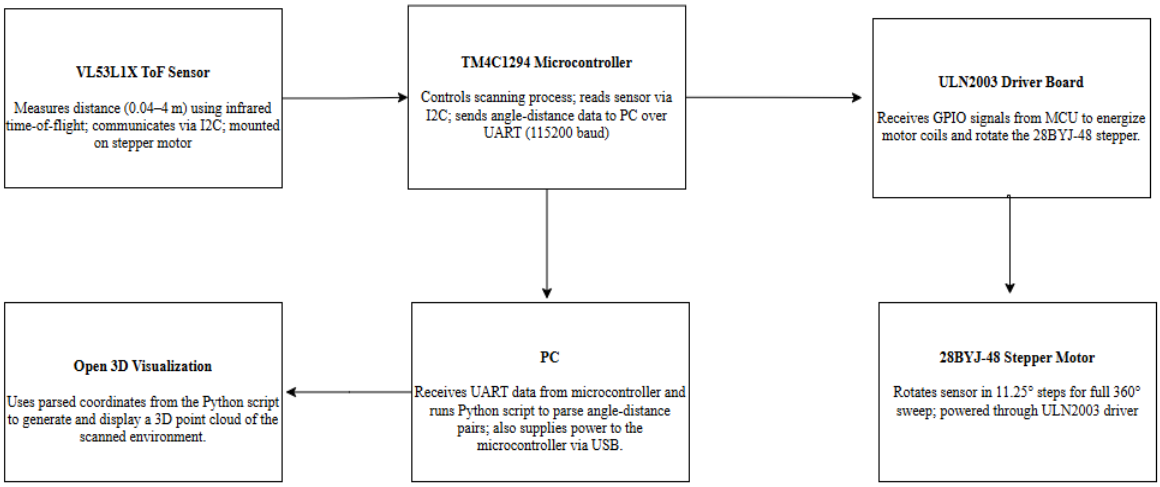

Figure 1: Block Diagram

Device Characteristics

Table 1: Microcontroller Characteristics

Microcontroller – TM4C1294	
Feature	**Value / Pin**
Clock Speed	60 MHz
Bus Speed	18 MHz
Supply Voltage	3.3 V
UART TX Pin	COM3 (to PC)
Baud Rate	115200 bps
I2C SDA / SCL Pins	PB3 (SDA), PB2 (SCL)
Push Button (Start)	PJ0
LED 1 (Stepper Blink)	PN1
LED 2 (Capture Status)	PF4
XSHUT (ToF Reset)	PG0

Table 2: Stepper Driver Characteristics

Stepper Driver – ULN2003	
Feature	**Value / Pin**
Input Voltage	5 V
Ground	GND
IN1 – IN4 (Motor Control)	PH0, PH1, PH2, PH3
Output	Connected to Stepper Motor

Table 3: Stepper Motor Characteristics

Stepper Motor – 28BYJ-48	
Feature	**Value**
Operating Voltage	5 V
Total Steps per Revolution	512
Step Angle Used	11.25°

Table 4: ToF Sensor Characteristics

ToF Sensor – VL53L1X	
Feature	**Value / Pin**
Operating Voltage	3.3 V
Ground	GND
SDA (I2C Data)	PB3 (TM4C1294)
SCL (I2C Clock)	PB2 (TM4C1294)
XSHUT / Interrupt	PE1

Detailed Description

Distance measurement

The board utilizes the VL53L1X Time-of-Flight (ToF) sensor to determine distance by first emitting a burst of laser and then calculating the time taken for the light that is reflected back. It converts this information into a digital measurement in millimeters, and its effective ranging distance reaches from 40 mm to 4000 mm.

To this end, a 28BYJ-48 stepper motor is used, which moves in steps of 11.25° via a ULN2003 driver that is controlled by the GPIO pins of the TM4C1294 microcontroller. For every angle value read, the microcontroller sends an I2C command to the ToF sensor to take a measurement. After reading that data, the microcontroller associates the measured distance with the respective angle value and then sends this combined data in UART form at a baud rate of 115200 to the PC.

A repetition of the above is made across the full circle of 360°, giving a total of 32 distance readings per scan. Pressing PJ0 initiates the scan procedure, in which LED PF4 glows as a data logging indicator, whereas PN1 blinks after each single reading.

Visualization

When the 32 angle-distance pairs are received by the PC, data parsing, transformation, and rendering are handled by a Python script. The script opens a serial connection at a baud rate of 115200, reads line by line in ASCII, and performs a polar-to-Cartesian conversion

- $y = r \times \sin(\theta)$
- $.x = r \times \cos(\theta)$
- $z = 0$ (since scanning is planar)

For example, if $r = 2000$ mm and $\theta = 90°$, then $x = 0$, $y = 2000$

Computed (x, y, z) values are combined in a list and then passed on to Open3D in the PointCloud format. The script allows for real-time visualization using the functions of Open3D so that the environment may be viewed either as a two-dimensional map or as a full three-dimensional structure.

The visualization window allows for instant feedback, thus enabling the checking of scan accuracy. In addition, the scan data can be saved in standard formats like .xyz or .pcd for archival purposes or additional analysis.

Application example

The scanning system can be used to assess the indoor accessibility of public spaces. For instance, a robot that is outfitted with this sensor package can scan the interior of a library or community center, detecting possible accessibility obstacles like slim corridors or blocked routes. [2] Information collected in this method can be later utilized in the creation of in-depth accessibility maps, making easier the implementation of the standards of accessibility as well as infrastructure growth.

Setup Instructions

1. Connect the System:
    - Plug the TM4C1294 microcontroller into your laptop via USB.
    - Confirm all wiring is correct (VL53L1X → I2C, stepper → ULN2003 → MCU).
2. Start the Scan:
    - Flash the C code provided onto the microcontroller
    - Press the onboard Push Button (PJ0) to begin scanning.
3. LED Indicators:
    - LED1 (PN1) blinks at each step (shows motor is moving).

- LED3 (PF4) turns ON during the full scan (indicates data is being captured and transmitted).

4. Run the Python Script on Your Laptop:
    - Open and run visualizer.py.
    - It connects over UART (115200 baud), reads scan data in real time, and renders a 3D point cloud using Open3D. The file will automatically save as a .xyz file
    - Run the tof_radar.xyz file to view the 3D image of the scan

Expected Output

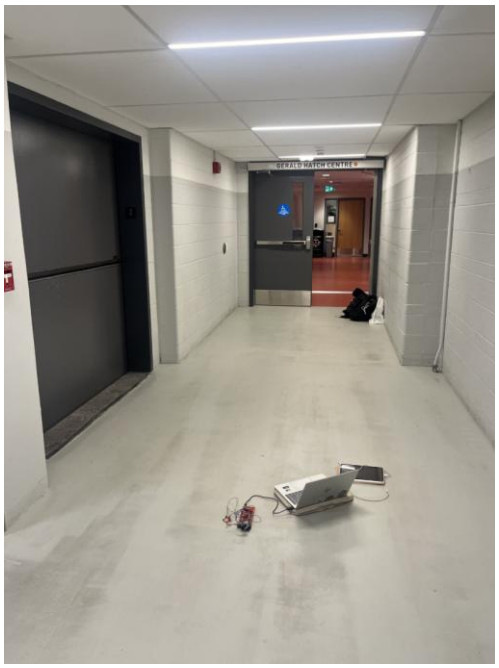

Figure 2: Hallway being scanned

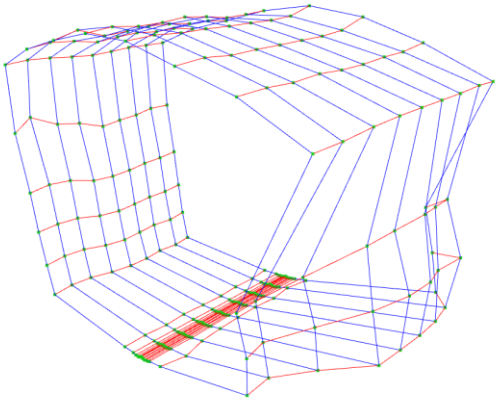

Figure 3: 3D Visualization Output

The above photographs show a comparison between the real hallway and the three-dimensional point cloud scanned by the system. In the middle of the hallway, the device carried out an extensive scan of 360° in steps of 11.25° using the VL53L1X Time-of-Flight sensor, leading to a total of 32 measurements of angle-distance.

The scans were processed and visualized using Open3D to create a two-dimensional top-down point cloud map of the area. The resulting output successfully encodes important spatial features like:

- Straight wall borders take the form of parallel, vertical stripes. The floor is represented by the lower points at the base

- The dip in coordinates at the door which is not in line with the starting wall

Here, x and y represent the floor plane, whose center is located at the position of the scanner, whereas z is 0 because there is no movement in the z-direction. Results verify that this cost-effective embedded system can reconstruct simple indoor environments at satisfactory resolution for either mapping or navigation functions.

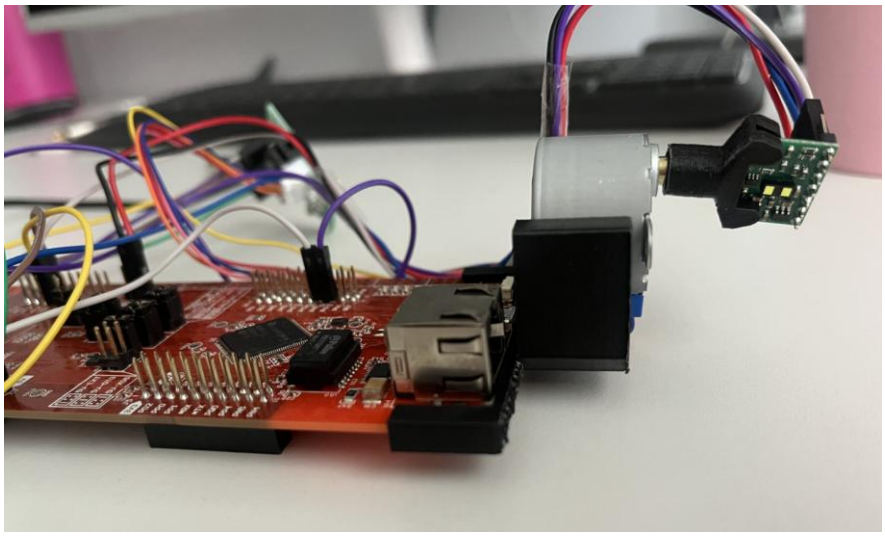

Figure 4: Physical Scanner Setup

Limitations

Several limitations were encountered in the development and integration of this system, stemming from both hardware and communication constraints.

1. Floating-Point Precision and Trigonometric Accuracy
The TM4C1294 microcontroller is based on a 32-bit Cortex-M4F processor with an embedded Floating-Point Unit (FPU). While this setup simplifies the use of floating-point arithmetic, it can lead to rounding errors when performing trigonometric functions, e.g., when converting polar coordinates to Cartesian coordinates using sine and cosine functions. Such rounding errors will become increasingly noticeable at higher distance values or when working with a higher angular resolution, potentially affecting the accuracy of the resulting three-dimensional point cloud.

2. Quantization Error of the ToF Sensor
The VL53L1X Time-of-Flight sensor has millimeter resolution for ranging measurements and works at 16-bit resolution. Since its output is pre-processed in digital form, its minimum step is 1

mm, which implies its worst-case quantization error equals ±1 mm. Its resolution is fixed and adequate for most indoor ranging use cases, as defined by sensor firmware.

Though this is a small margin, it may impact applications that require highly precise measurements or consistent surface detection.

3. Maximum Serial Communication Rate

In an effort to determine the optimal speed supported by the UART, the Device Manager of the PC was opened and accessed, and the XDS110 UART interface, which is located in the folder called Ports, was examined. It was found out that this interface can support an optimum baud rate of 128000 bps. Based on those details, a steadier and slower rate of 115200 bps was chosen, thus allowing consistent data exchange between the microcontroller and PC.

4. Communication Protocols and Speeds

The VL53L1X communicates to the microcontroller using an I2C bus at its regular speed of 100 kHz, sufficient for single sensor integration. UART at 115200 bps was used in microcontroller-PC communication. These interfaces were used for ease in development and simplicity of use in development environments, even if they were less than ideal high-speed solutions.

5. Primary Limiting Factor on System Speed

After analyzing the overall system, it became clear that the major bottle-neck of the system in its initial form was the timing delay of the VL53L1X sensor, closely followed by the stepper motor speed. Even with the high-speed motor capabilities, the system has to wait for the completion of measurement on the Time-of-Flight (ToF) sensor before progressing to the next angle. Considering this project's operation within an imposed timing budget for measurements, namely, 140 milliseconds for a single measurement, the system has a limit upon the time taken for one measurement in long-distance mode.

To determine if this was an operational limiting factor, the motor delay was reduced, which caused the data returned from the sensor to become unreliable in some cases. We therefore tried to ensure that the sensor had not completed its ranging sequence. Through modification of the delay and careful monitoring of the point at which data became valid again, we concluded that the scan rate was limited by the sensor timing rather than by mechanical stepping or serial communications.

Bus Speed Configuration and Relevance

As required by the student-specific parameters, I configured the system bus speed to 18 MHz by changing the PSYSDIV value in the PLL initialization to 26 by using this formula:

$$\text{Bus Speed} = \frac{480\ \text{MHz}}{\text{PSYSDIV} + 1} = \frac{480\ \text{MHz}}{26 + 1} = 17.78\ \text{MHz}$$

The bus speed has a significant impact on the execution speed of the microcontroller in executing instructions, such as I2C communications, GPIO bit toggling, and delays. Therefore, after reducing the system clock frequency from 120 MHz to around 18 MHz, I recalibrated the SysTick delay constants to maintain the integrity of motor stepping and sensor timing.

Sensor Range and Surface Limitations

VL53L1X features an effective range of four meters, thus limiting its use in indoor as well as mid-range scenarios. Glossy, absorbent, or dark surfaces can potentially interfere with infrared reflections, leading in turn to inaccurate or noisy readings. It is important to consider these aspects while interpreting outputs or choosing suitable scan environments.

Circuit Schematic

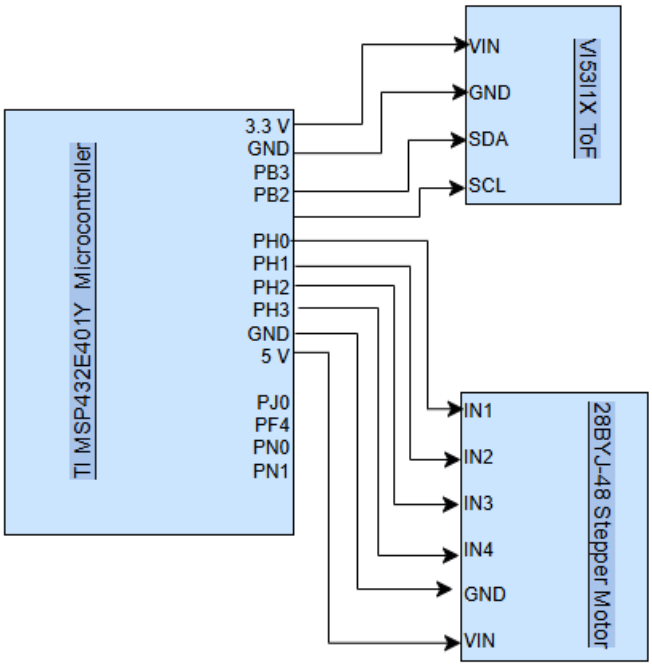

Figure 5: Circuit Schematic

Programming Logic Flowcharts

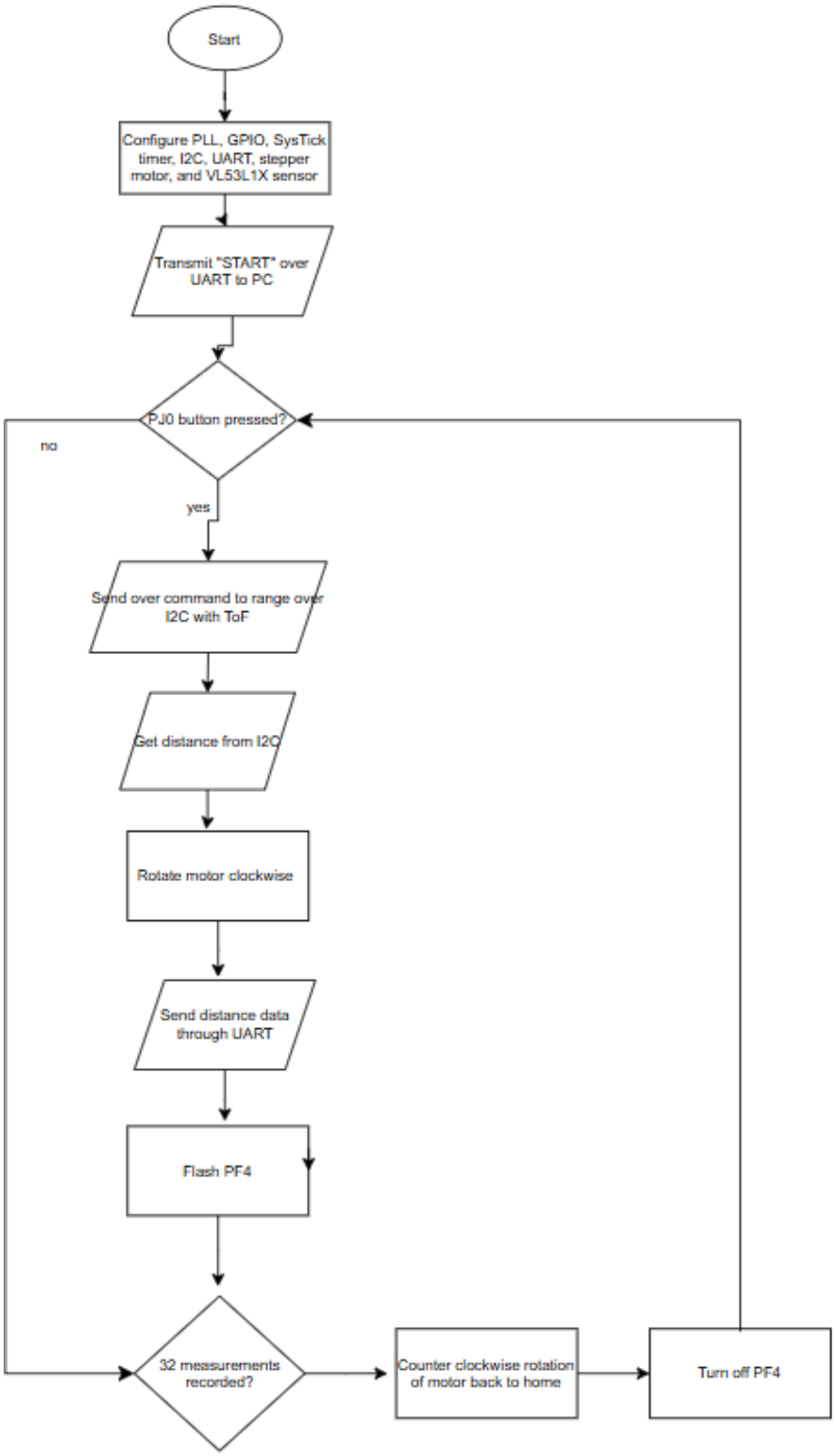

Figure 6: Flowchart For Microcontroller

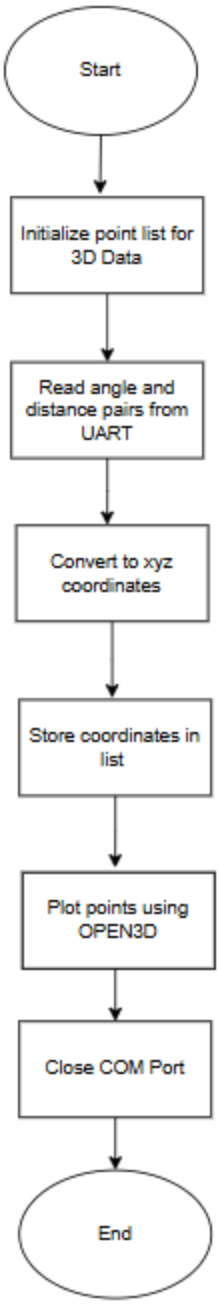

Figure 7: Flowchart for PC Code

References

[1] Texas Instruments, *"MSP432 SimpleLink™ Microcontrollers – Technical Reference Manual"*, 2018, Ch. 4, 17, pp. 326–327, 1201–1252.

[2] X. Su, D. Campos Zamora, and J. E. Froehlich, "RAIS: Towards A Robotic Mapping and Assessment Tool for Indoor Accessibility Using Commodity Hardware," *The 26th International ACM SIGACCESS Conference on Computers and Accessibility*, pp. 1–5, Oct. 2024, doi: https://doi.org/10.1145/3663548.3688512. Available: https://makeabilitylab.cs.washington.edu/media/publications/Su_RaisTowardsARoboticMappingAndAssessmentToolForIndoorAccessibilityUsingCommodityHardware_ASSETS2024.pdf?utm_source=chatgpt.com. [Accessed: Apr. 10, 2025]

[3] Adafruit Industries, "Adafruit VL53L0X Time of Flight Distance Sensor - ~30 to 1000mm," *Adafruit.com*, 2019. Available: https://www.adafruit.com/product/3317