Microprocessor Systems Project
COMPENG 2DX3

Final Report

Instructors: Dr. Haddara / Dr. Doyle / Dr. Athar

Apr 9th, 2025

Aaron Ghosh - L10 - Group 129 - ghosha20 - 400512786

1 Device Overview

1.1 Features:

Texas Instrument MSP-EXP432E401Y microcontroller:

- Bus speed of 14MHz (assigned)
- Operates at voltages between 2.5 - 5.5V
- 1024Kb of flash memory
- 256Kb of SRAM
- 6Kb of EEPROM
- 32-bit Cortex M4F CPU
- GPIO Status LEDs
    - PF0(Measurement)
    - PF4 (UART Tx)
    - PN1 (Additional Status)
- Data transmission to PC via UART
- UART communication at 115200 BPS baud rate
- $60 CAD

VL53L1X time-of-flight (ToF) sensor:

- Maximum detection range of 400cm (4m)
- Operates at frequencies up to 50Hz
- Operates at voltages between 2.6 - 3.5 V
- Data transmission to microcontroller via I2C
- $35 CAD

28BYJ-48 Stepper Motor & ULN2003 Driver:

- A full 360° rotation is 512 steps
- 4 LEDs to indicate current phase
- Operates at a voltage between 5 - 12V
- $8 CAD

Scanning:

- Indoor 3D mapping using Open3D
- Adjustable length based on Z-axis
- Python version 10 - 12 (compatible with Open 3D)
- Visualization with point cloud using a .xyz file for

1.2 General Description

This project is a compact embedded system that is designed to scan enclosed spaces using a rotating ToF sensor that is mounted onto a stepper motor. Both the ToF sensor and stepper motor are controlled by the MSP432E401Y microcontroller. The system is configured to capture radial distance data every 11.25° (32 positions), completing one full rotation per scan (within the XY plane). After each scan, the user can move the system along with the sensor forward along a linear path (Z direction), simulating 3D space with multiple scans.

Upon start, the system will be waiting for the onboard button PJ0 to be pushed (polling), this will start the scanning process, allowing the sensor to scan distance data and move the stepper motor accordingly. Throughout each scan, data is sent from the sensor to the microcontroller via I2C communication. This information is then sent to the PC via UART communication where the data is then read by a python program using pyserial. This program processes the data and uses the Open3D library to create a visual representation of the scans. The sensor will transmit X and Y values, whereas the Z values will be incremented separately to create depth. This is because the user will move forward and take another scan/slice of the 3D space. The python program handles all this separately from the microcontroller to create the 3D visualization of the scans for the user to view.

1.3 Block Diagram

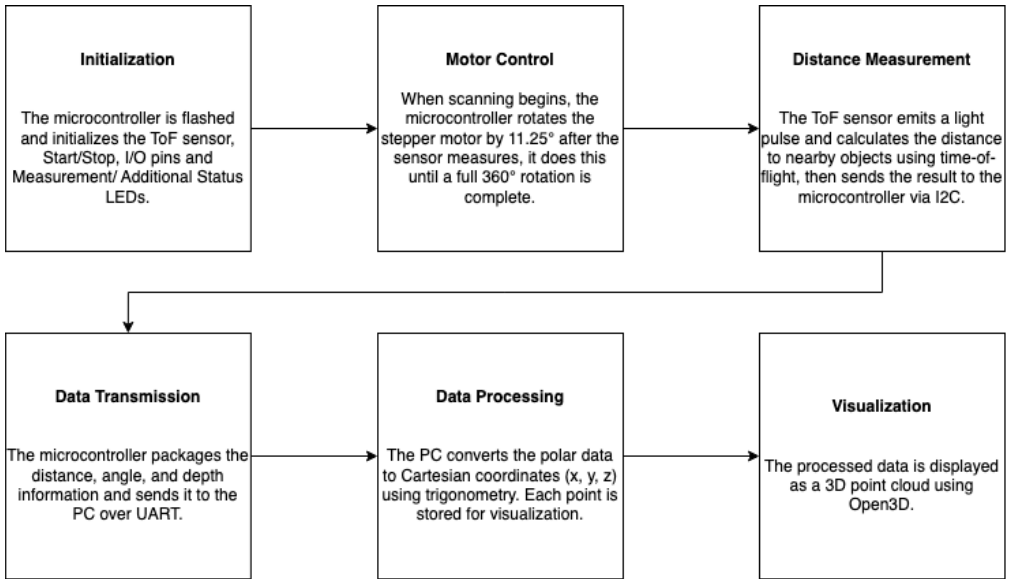

Figure 1: Data flow graph

2 Device Characteristics Table

Microcontroller (MSP432E401Y)	Stepper Motor / Driver	ToF Sensor (VL53L1X)
Clock Speed: 60 MHz	Pins: IN1-IN4 → PH0-PH3	Vin → 3.3V
Bus Speed: 14 MHz	V+ → 5V	GND → GND
Baud Rate: 115200 bps	V− → GND	SDA → PB3
Serial Port: COM3	The motor has 32 positions	SCL → PB2
LED Status Pins: PN1, PF1, & PF4	Rotates every 11.25° (16 steps)	Frequency 50Hz
UART Communication to PC	Full rotation is 512 steps	I2C Communication with MC

Table 1: Device Characteristics Table

3 Detailed Description

3.1 Distance Measurement

The VL53L1X Time-of-Flight (ToF) sensor plays a crucial role in measuring distances. It works by emitting infrared light pulses from its emitter. This light then travels to an object within the sensors path and reflects back into to the sensor's receiver. The time delay between the light being emitted and received is used to calculate the objects distance. This is done using the time-of-flight formula:

$$Distance = \frac{Time\ of\ Flight \times Speed\ of\ Light}{2}$$

After receiving the data for the distance, the ToF sensor will automatically start processing the signal including transduction, conditioning, Analog to Digital conversion, and outputting a digital value representing the distance in millimeters. This data is then sent to the microcontroller via I2C communication.

To activate the ToF, the microcontroller will invoke its boot function. However, the system will not start scanning as the microcontroller waits for a trigger from the onboard button (PJ0) through a polling method. As soon as the user presses PJ0, the system will enter a loop and wait until the button is released. This is called debouncing and ensure that no accidental button presses are inputted. After this the system will begin capturing data for the scan.

As the ToF sensor starts capturing distance data, the stepper motor will start rotating, allowing the sensor to scan at a different position. The stepper motor has a total of 512 steps to cover 360°. The measurements are taken every 11.25°, this corresponds to 16 steps, meaning the motor must rotate 32 times in in intervals of 16 steps. The 32 positions each cover 11.25°, allowing the ToF sensor to capture 360°. This angular spacing divide the whole scan into 32 segments per rotation. To ensure that the scans capture a clear reading, the system pauses every time the motor reaches a new measurement angle. The system checks if the motor has moved 16 steps (11.25°) and, when it does, takes a distance measurement and logs it along with angle of the motor.
A status LED flashes with each measurement to give visual feedback and indicated that the system is working as intended PF0 for Measurement, PF4 for UART and PN1 for Additional Status)

To prevent wire twisting and avoid cables tangling and unplugging during the scanning process, the motor alternates its rotation direction after each layer starting with a clockwise turn. This allows the sensor to operate continuously without intervention. Since the system itself doesn't track its physical location, the Z-coordinate for each scanned point is calculated on the computer by multiplying the current scan layer number by the step size. The X and Y coordinates are computed using trigonometric formulas that convert the radial distance and motor angle from polar to Cartesian form.

3.2 Visualization

At the end of every scan (one full motor rotation), the microcontroller sends a special, informing the PC that the current layer's data transmission is finished. The Python program will then pause after each scan until this signal is received, ensuring each layer is fully captured before the next begins

The VL53L1X ToF sensor operates in Long Distance mode, allowing for measurements up to 4 meters. While this mode increases the usable range it also increases the noise sensitivity, reducing accuracy on dark or reflective surfaces.

Once all scans have been fully completed, the measurement data gets sent over UART to the PC, where it is received by the Python program with pyserial. This program takes the distance and angles values in the format (distance, angle) with 32 (0 - 31) unique scans. This is then parsed and used to be converted into 3D coordinates. For each point, the radial distance and motor angle will be used to convert the polar form into cartisan form using the equation below:

$$x = r \times \cos(\theta) \quad y = r \times \sin(\theta) \quad z = \text{num\_slices} \times \text{z\_step}$$

Each (x, y, z) coordinate is appended and used to build the 3D model of the scanned environment. In addition to generating a point cloud, the program will also create the structural links between data points using the Open3D library. Within a single scan, adjacent points are connected in sequence to form a circular loop. Matching points between are linked vertically. To account for the alternating motor direction between scans

Once all layers are collected and converted, the full set of coordinates is saved to a .xyz file. This file can be reopened later for rendering, analysis, or comparison.

4 Detailed Description

4.1 Instructions

To run this project, you'll need Keil µVision to compile and flash the firmware to the microcontroller, Python 3.10 - 3.12 (compatible with Open 3D) and the libraries, pyserial, Open3d, and numpy installed on your PC, and any Python IDE like Eclipse or VS Code to run the visualization script.

Part 1: Hardware Setup: *(Refer to device characteristics for wiring)*

- Connect the MSP-EXP432E401Y microcontroller to your PC with a USB cable
- Make sure the VL53L1X Time-of-Flight sensor is connected to the I2C interface
- Wire the 28BYJ-48 stepper motor to GPIO ports PH0 - PH3 using the ULN2003 driver module

Part 2: Programming the Microcontroller

- Launch Keil µVision and open the project file named Final_Project.uvprojx
- Click on the "Translate" option to compile the code, then "Build" to generate the output, and finally "Download" to flash the code onto the microcontroller.

Part 3: Running the Scanning & Visualization Program

- On your PC, open Device Manager and identify the COM port that the microcontroller is plugged into
- Open the Python file 3dvisual.py in your preferred Python IDE.
- Modify the configuration variables (commented as CONFIG)
- Run the program and wait for the terminal prompts to appear
- Press the PJ0 button on the board to begin the scanning sequence
- After each scan, physically move the entire system forward along the z-axis to prepare for the next scan slice
- The program will continue running until the process ends when all set scan layers are completed

Part 4: Accessing the Visuals

- After the last scan, all the 3D points will be saved to a file called "3Dvisual.xyz". this file can be used in Open3D or other third-party tools for further analysis

4.2 Expected Output

After a successful scan, the system will provide several visual cues that will confirmation the system is operating correctly at different stages of the scanning process. These include flashing LEDs that indicate when a scan has been triggered, UART data is being transmitted, or a button has been pressed. Once a full scan cycle is finished, the Python program will use the terminal to display a message to indicate that a scan is done " scan_done", this confirms that data collection for that specific layer has ended. At the end of the program, the user can see that there is a new Open3D viewer that launched. This viewer creates an interactive 3D point cloud generated from the scanned data, it will also be saved locally as an .xyz file. This can be used by the user to further examine the scan for analysis purposes.

My assigned scan location was location G in ETB. Below is a picture of the hallway as well as a side by side comparison with my scan.

As shown in Figures 3 - 5, my scan was able to pick up specific key details of the hallway, highlighting the general shape as well as specific features such as other hallway parallel to the edge, the gap in the ceiling and the narrow parts with the tables.

Figure 2: Hallway location G in ETB

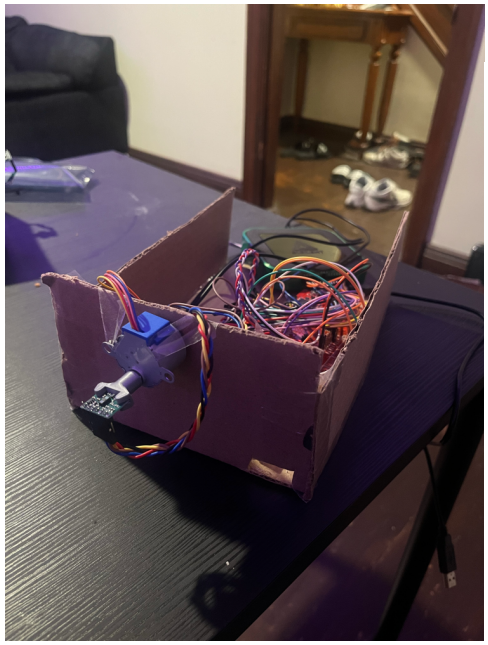

Figure 2: Physical Circuit

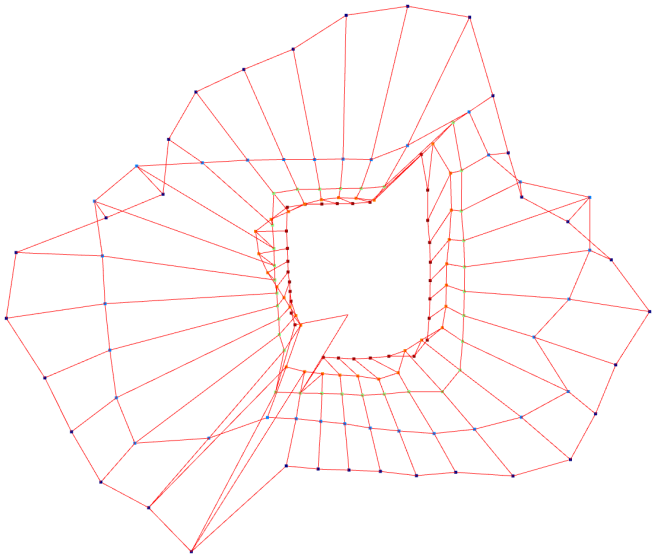

Figure 3: 3D visualization of with front view

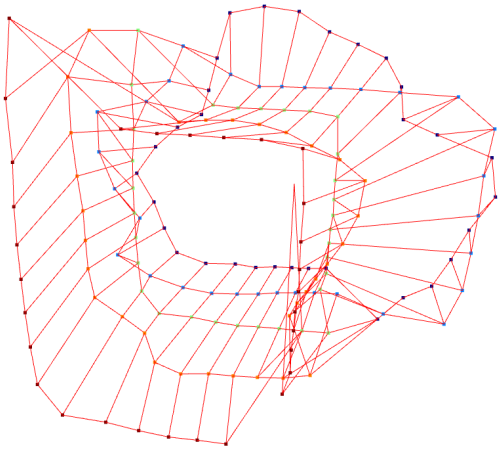

Figure 4: 3D visualization of with back view

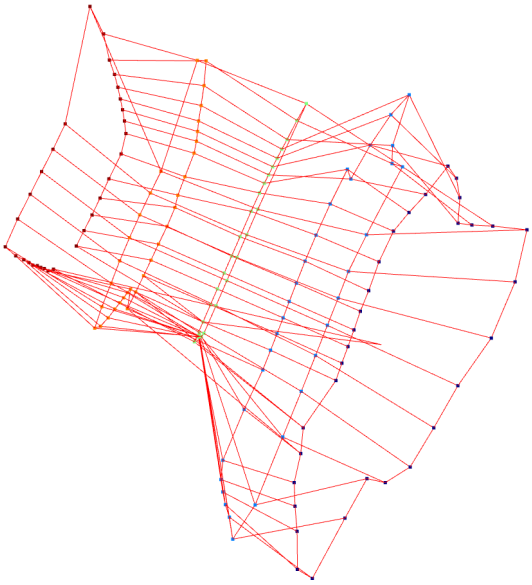

Figure 5: 3D visualization with top view

5 Limitations

5.1 Floating Point & Trigonometric Calculations

The MSP-EXP432E401Y microcontroller is has a built-in Floating Point Unit (FPU) that enables efficient 32-bit single-precision calculations like addition, subtraction, multiplication, and division. It also allows quick and easy conversion between different types of data types such as integer and floats. In this project, we used functions like sin and cos which almost always use float values with the standard math.h library. Because of the FPU, these calculations can be performed quickly and do not introduce any noticeable processing delays. Although the FPU enables fast trigonometry operations, floating point data types face rounding which may introduce slight inaccuracies in position when processing many layers, especially if angles are computed in radians repeatedly.

5.2 Quantization Error

The VL53L1X Time-of-Flight sensor provides digital distance readings with a resolution of 1 millimeter, this means that it can only detect changes in distance within 1 mm intervals. These values are generated internally and transmitted directly to the micro via I2C. All signal processing is handled internally by the ToF sensor. This means that no ADC conversion is necessary since the sensor cannot differentiate between two values that fall within the same 1 mm range (1.3 vs 1.4). This means that the theoretical maximum quantization error is ± 0.5 mm. This accounts for the largest possible deviation between the actual physical distance and the digitized output. For practical purposes, the quantization error for this project is considered to be 1 mm.

5.3 UART Communication

Using the Windows Device Manager, it can be seen and verified that the maximum supported baud rate for the XDS110 UART interface on the PC is 128000 bps. However, in this project, the UART communication baud rate on the microcontroller was set to 115200 bps as it is the most commonly used transmission speed. This configuration ensured a consistent data exchange for every scan cycle. To validate the integrity of the communication, the number of lines received on the PC was matched against the expected transmission count to confirm that everything is correct.

5.4 Microcontroller ToF Communication Speed

The ToF sensor communicates with the microcontroller over the I2C bus, operating at a standard clock speed of 100 kHz. It is connected to the microcontroller though the GPIO pins PB2 and PB3. They are connected with the SCL and SDA pins on the ToF sensor. This speed aligns with the VL53L1X's specifications and is fully supported by the API used for sensor initialization. Even though the I2C protocol allows for faster data rates when operating in the Fast Mode, this project prioritizes accuracy and simplicity over speed. By maintaining the standard speed as opposed to the higher speeds capable with I2C, the lower standard speeds ensure stable communication and avoids extra unnecessary system complexity.

Another thing to note is that most of the delays occurs within the sensors internal processing. This means that a faster I2C communication speeds would not make a noticeable impact as the delays will take up a constant amount of time despite the speed.

5.5 Motor & ToF System Bottleneck

The system's overall performance limited by the time required for the VL53L1X sensor to complete each distance measurement is almost negligible. The performance of the sensor also varies depending on which mode it is operating at as well as the light level and reflectivity of the surface being scanned. A single range scan can take anywhere from 20 - 60 ms. The stepper motor speed may also influence the total performance of the system as it impacts the scan duration. However, the sensor will have to scan at 32 positions regardless of the motor speed and the time needed to move between positions is already extremely fast. During testing, reducing the delay between the VL53L1X_CheckForDataReady() function almost always caused the sensor to not return a proper value, effectively stalling the scan process, taking more time. It can also lead the motor skipping steps because it moves too fast and does not interrupt program execution. These results prove that the sensor's internal processing time is the main performance bottleneck and dictates the timing for each scan cycle to ensure data accuracy.

5.6 Assigned Bus Speed Configuration and Timing Adjustments

To configure the system for a 14 MHz bus speed based on assigned individual student numbers, the internal 480 MHz PLL was divided using the PSYSDIV register. By using the calculation below and always rounding up, a PSYSDIV value of 34 will always produce the desired clock rate.

$$BusSpeed = \frac{\text{SysClk}}{\text{P}SYSDIV + 1} = \frac{480}{34 + 1} = 14\,MHz$$

Alongside this, the SysTick timer was updated to reflect the new clock speed, enabling accurate software-based delays. The accuracy of this delay was experimentally confirmed using an oscilloscope, validating the timing precision through consistent delay

$$SysTick\ =\ 140\,000$$

6 Circuit Schematic

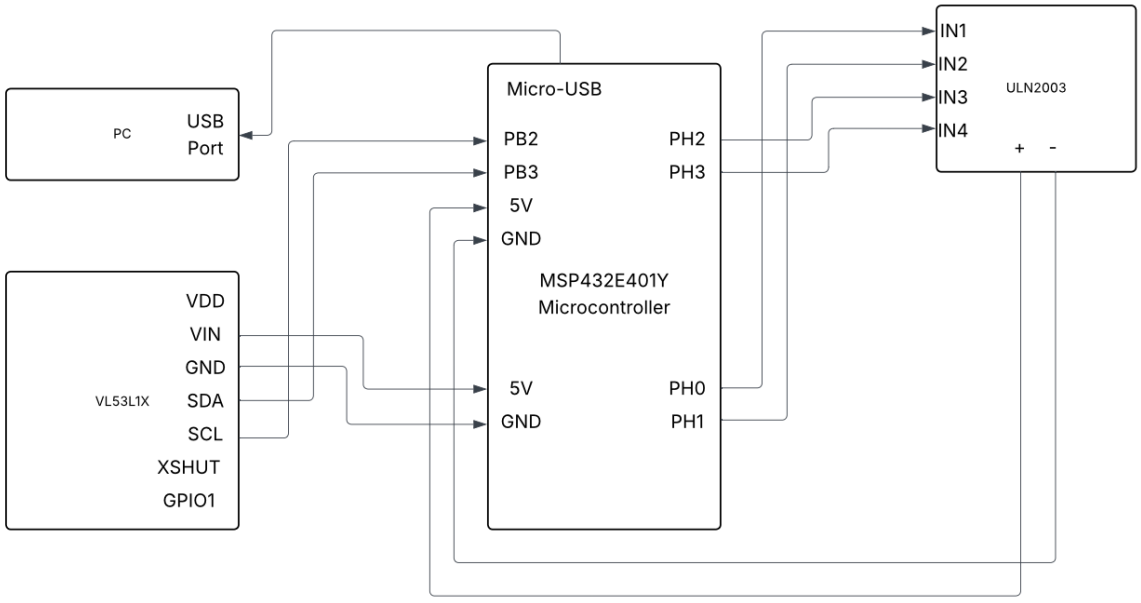

Figure 6: Circuit system schematic

7 Programming Logic Flowchart

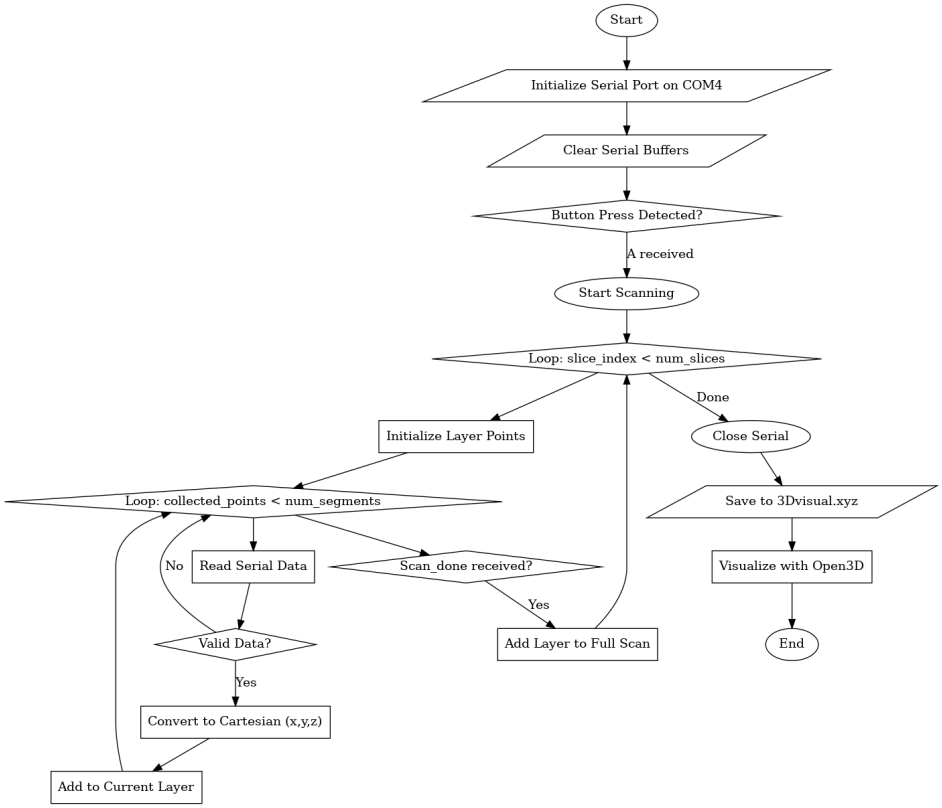

Figure 7: 3dvisual.py

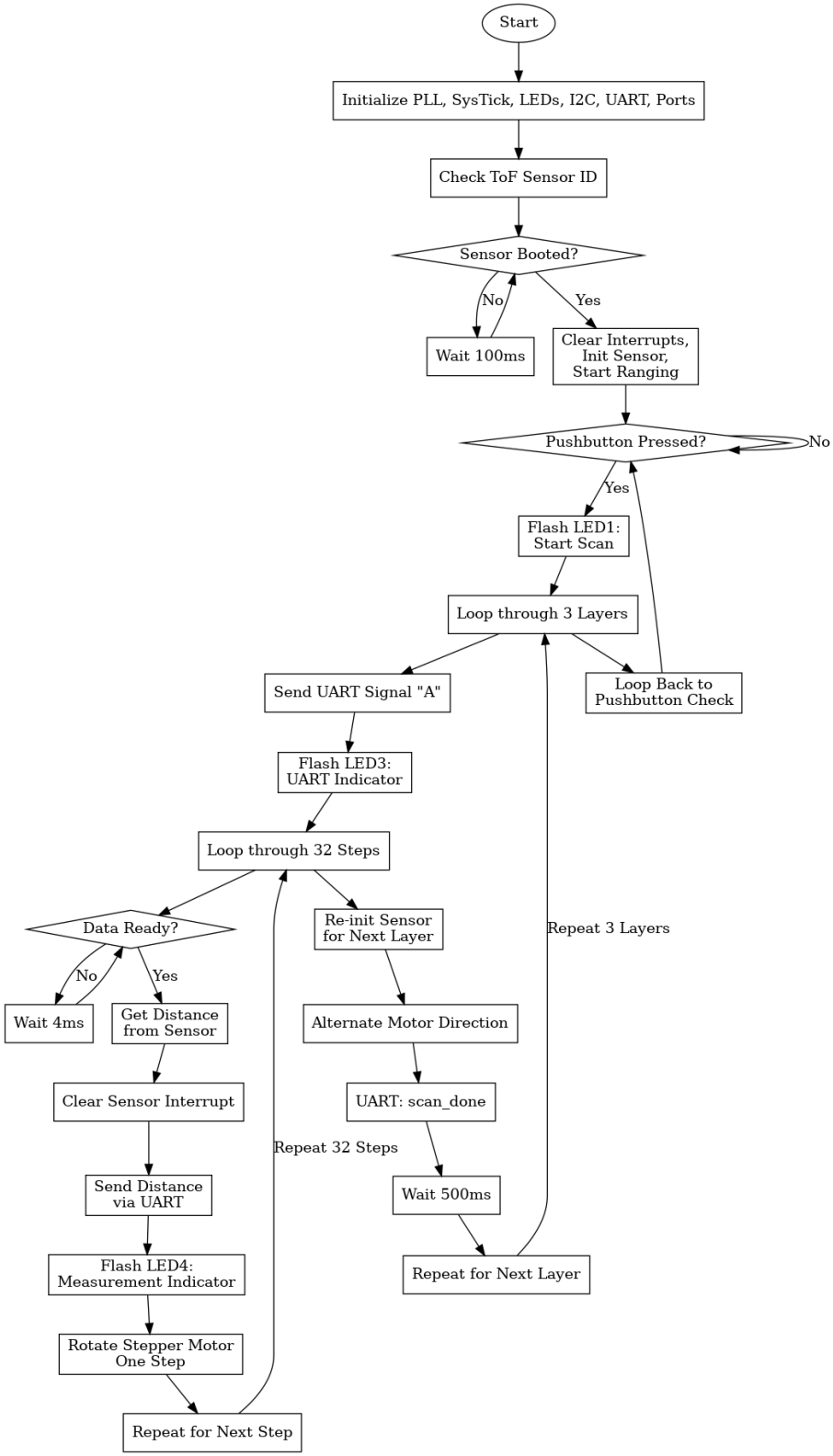

Figure 8: Flowchart for main.c

8 References

[1] "MSP-EXP432E401Y Development kit | TI.com" https://www.ti.com/tool/MSPEXP432E401Y [Accessed April 1st, 2025]

[2] "vl53l1x - COMPENG 2DX3: Microprocessor Systems Project" https://avenue.cllmcmaster.ca/d2l/le/content/638924/viewContent/5019689/View [Accessed April 1st, 2025]

[3] "Pololu - VL53L1X Time-of-Flight Distance Sensor Carrier with Voltage Regulator, 400cm Max" https://www.pololu.com/product/3415 [Accessed April 1st, 2025]