ToF Spatial Mapping Device for Indoor Exploration: Datasheet

Nathan Electronics

April 9, 2025

Table of Contents

Device Overview

General Description

The spatial mapping ToF device provides quality performance and accuracy. As an alternative to the Light Detection and Ranging sensor, the ToF spatial mapping device provides a cost effective and less bulky design suitable for indoor navigation. This device revolves around the main computing system of the MSP-EXP432E401Y microcontroller, integrating the VL53LIX time of flight (ToF) sensor, the 28BYJ 4-phase stepper motor, along with the ULN2003 motor driver board to accompany the stepper motor. The device comes initialized with appropriate default firmware to interface the microcontrollers on board push buttons and LEDs upon power up.

The VL5L1X ToF sensor achieves distance measurements for each scan through signal acquisition, preprocessing and analog-to-digital conversion (ADC). The ToF sensors principle works by measuring the time it takes for the light to reflect from the emitter back to the sensor, the value is then divided by two and multiplied by the speed of light. After acquiring the raw value, the sensor must conduct some internal pre-processing, such as ambient light rejection or signal filtering to prepare the signal for ADC. The ADC enables the sensor to convert the reflected analog light pulses into digital timing data used to process the digital distance data.

Given this raw digital distance data, the VL53L1X communicates this information to the microcontroller via I2C protocol. Once the data is received to the microcontroller, it works to send data to the personal computer (PC) via the UART serial communication protocol. The PC uses the specific COM port to access the values and manipulate the data with trigonometric functions to transform the data into a y-z coordinate system. The entirety of the serial communication protocols driven by the set 18MHz bus speed which facilitates data transfer. The PC can then take multiple layers of the y-z coordinate, and piece them together along the manually displaced x axis to render a 3D spatial map of the interior scan.

Features

Texas Instruments MSPEXP432E401Y	ToF Sensor VL53L1X	Motor Driver Board ULN2003	4-Phase Stepper Motor 28BYJ-48
Bus Speed: 120MHz **Operating Voltage:** 2.97V to 3.63V **Cost:** $39.99 USD	**Bus Speed:** ≤ 400kHz **Operating Voltage:** 2.6V to 3.5V **Cost:** $23.79 CAD **Serial Communication:** I2C protocol	**Operating Voltage:** 5-12V DC **Cost:** $2.05 CAD **Maximum Current per Output:** 500mA	**Frequency:** 100Hz **Operating Voltage:** 5-12V DC **Cost:** $5.50 CAD **Step Angle:** 5.625°/64

Serial Communication: UART, I2C, SSI/QSSI, USB, CAN protocols **Baud Rate:** 115200 bps **Flash Memory:** 1024kB **SAR:** 12-bits	**Ranging Frequency:** 50Hz **Distance Measurement:** ≤ 400cm	**Indicators:** 4 On-Board LEDs	

Block Diagram

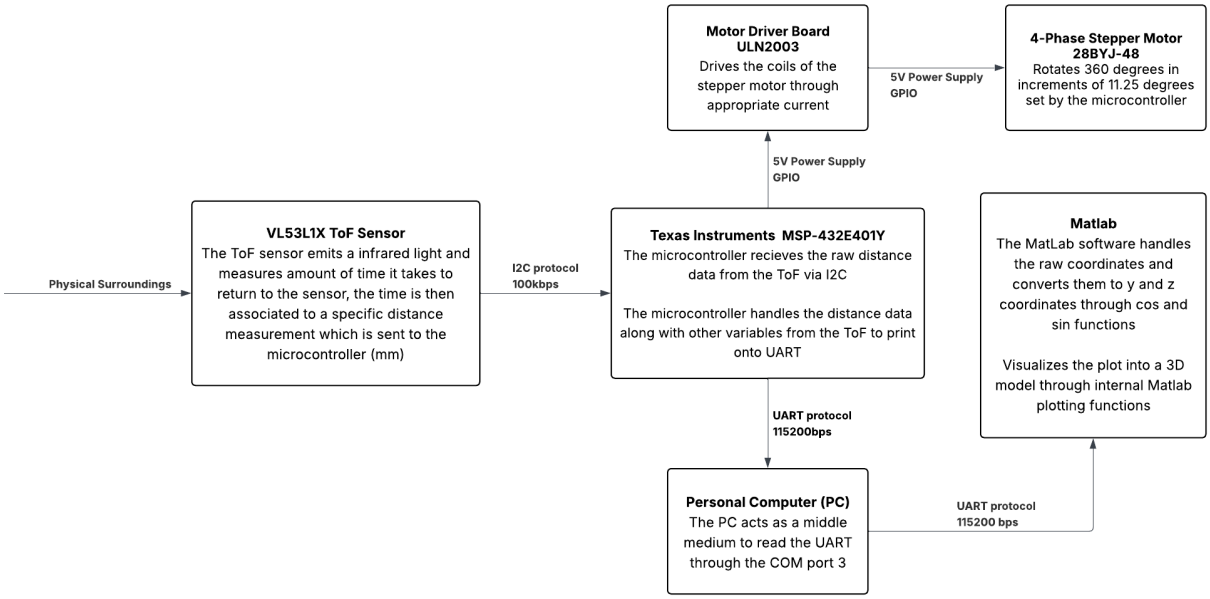


Figure 1: Block Diagram representation of the Data Flow through the Device

Device Characteristics

MSP-EXP432E401Y Microcontroller

Device Specifications	Assigned Pins & Values
Clock Frequency	18MHz
Baud Rate	115200 bps
Measurement Status (LED)	PF0
UART Bit Transmission Status (LED)	PN0
360° Rotation Completion (LED)	PN1
Start/Stop Button (Push Button)	PJ1
Communication Port	COM3

VL53L1X ToF Sensor

Device Specifications	Assigned Pins & Values
Vin	+3.3V
GND	GND
Serial Data Line (SDA)	PB3
Serial Clock Line (SCL)	PB2

ULN2003 Motor Driver

Device Specifications	Assigned Pins & Values
V (+5V)	+5V
GND (-)	GND
Input 1 (IN1)	PH0
Input 2 (IN2)	PH1
Input 3 (IN3)	PH2
Input 4 (IN4)	PH3

Detailed Description

Distance Measurement

The measurement aspect of the spatial mapping process of this device is based on the VL53L1X ToF sensor. The sensor consists of an emitter and a sensor, in principle it works by emitting pulses of infrared laser light at a 940nm wavelength, this laser then reflects on the nearest object such as a wall, which is collected by the receiver sensor. The time delay between the emitted and received periodic signal is measured by the given equation.

$$Measured\ Distance = \frac{Photon\ Travel\ Time\ (Time\ Delay)}{2} \times Speed\ of\ Light$$

Equation 1: Time of Flight Principal Calculation

The measured distance value is measured internally within the VL53L1X sensor, for example if the time delay was 6.67ns and the speed of light is 3×10^8m/s then the resultant measured distance would be 1005 mm. Once appropriately filtered and calculated, the value measurement value is sent over I2C protocol from the VL53L1X to the microcontroller in units of millimeters (mm).

The spatial mapping involves a multitude of components that work in tandem to effectively join the data together. Using the pin assignments for wiring in the device characteristics section for the VL53L1X sensor, the ToF transmits data using I2C protocol across the SDA and SCL serial bus lines. Additionally, following the pin assignments for the ULN2003 4 phase motor driver to setup the stepper motor across to operate the inner magnetic coils. These two devices work in unison, with the ToF mounted on the stepper to create a 360° apparatus for

each layer. Once the appropriate wiring has been configured, powering on the microcontroller will enable the appropriate initializations of the sensor and serial communication interfaces, these initializations are shown by PN0 flashing. The microcontroller will then immediately sit in a low power mode due to the implementations of interrupts for the integrated push button PJ1.

Once initialized and ready to scan PJ1 activates the start and stop functionalities of the scan. Initial press of PJ1 activates the scanning by putting the system in measurement mode indicated by PF0 being on. The internal logic enables the stepper motor to spin 360° per slice, while taking measurements 32 times per rotation, effectively setting the measurement angle at 11.25°. The distance measurements and sensor initializations are all printed onto UART for the PC to read and interpret. Additionally, indication of each successful UART transmission bit is noted by the flashing of PN0, thus every scanned slice PN0 will flash 32 times.

On the UART serial communication within the PC, the external PC software Matlab is used to process the distance measurement sent over the UART line by the microcontroller. The Matlab script handles the data conversion of the distance data to appropriately display them in the y-z plane. Using the following calculations with angles incrementing at 11.25° from 0 to 348.75°:

$$y = distance\ (mm) \times \cos\ (angle)$$

$$z = distance\ (mm) \times \sin\ (angle)$$

Equation 2: Y and Z trigonometric coordinate calculations

The y and z planes are achieved through dynamic updates based on the distance values from the sensor, while the x plane is manually incremented at 500mm steps to approximately simulate each human step through a room while accounting for high resolution, effectively piecing together multiple y-z plane scans along the x plane.

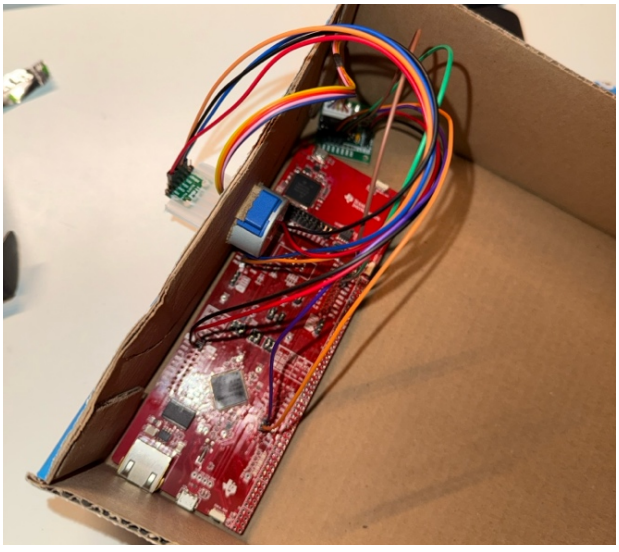


Figure 2: Embedded Hardware Setup for Spatial Mapping Device

Visualization

Throughout the measurement phase, the microcontroller continuously sends distance data at each measurement point from the ToF across UART. On the PC, the external software Matlab is used to open the COM3 port to receive and process the data within the UART serial line. Additionally, the Matlab script opens a xyz formatted file to write the coordinate data that has been processed trigonometrically (Equation 2). As an intermediate step, the software plots each layer individually after each scan while temporarily storing the data in a 32x3 matrix, giving the user an option to select whether to write the data to the xyz or restart that slice of the scan. This is a precautionary step to ensure that should transmission failures happen halfway through a scan; the user does not need to restart the spatial mapping from the beginning. Once all 10 layers of the scan have been completed, the program will be terminated effectively closing COM3 port, and closing the xyz file with all saved coordinate information. Following the termination, the script runs a new section that involves reading the xyz matrix file to individually extract each coordinate, the program then indexes through the coordinates appropriately. Using the built in plotting and meshing functions in Matlab, the points are all connected through a series of lines effectively showing a 3D visualization of the indoor scan.

Application

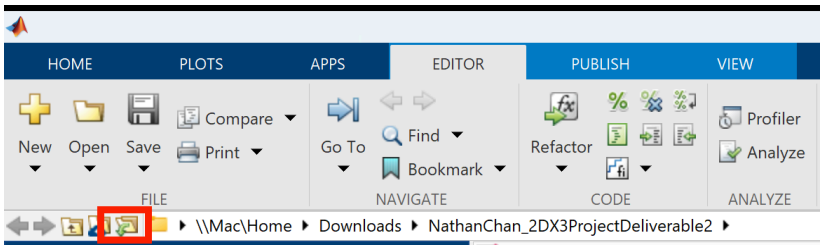
Application Note

The use of spatial mapping is to create high resolution 3D digital elevation models for different applications such as archeology and forestry. The following section focuses on an instructional guide on how to operate this spatial mapping device. Providing an application example of an expected output of an indoor hallway scan done in the John Hodgins Engineering (JHE) building at McMaster University; all found below for reference. This instruction set assumes the Keil Developer and MatLab software's is installed and ready for use.

User Instructions

** The following instructions assumes all hardware and software has been set up **

1. **Determining the device COM Port**
    i. Plug in the MSP-EXP432E401Y microcontroller into your PC
    ii. Open Device Manager > Ports (COM & LPT) > 'XDS110 Class Application/User UART (COM #)'. (Make note of the COM number)
2. **Open and Edit the MatLab software**



i. Navigate to the appropriate .m file through the folder using the circled button: NathanChan_2DX3ProjectDeliverable2 > ProjectD2_MatLab.m

ii. In line 2 of the code : s = serialport('COM3', 115200, 'Timeout', 10);, change the COM3 value to your specified COM# found from the Device Manager

iii. In line 17 of the code: angles = 0:11.25:348.75;, change the values ’11.25’ and ‘348.75’, given the nature of the 360° rotation, the ’11.25’ is your incrementing angle (ex. 360/11.25 = 32 measurements), and the ‘348.75’ value represents the last value of indexing (ex. 360-11.25 = 348.75)

iv. In line 19 of the code: numLayers = 10;, change the value ‘10’ for the number of layers you would like to have/number of slices along x plane.

v. points_per_layer = 32;, change the value ‘32’ to your specified number of measurements per 360° rotation on the y-z plane.

3. Open and Edit the Keil Development Software

i. Navigate to the appropriate file by clicking Project > Open Project > NathanChan_2DX3ProjectDeliverable2 > 2DX_2024_Studio8 > 2dx_studio_8c (project file)

ii. Navigate to line 220: if ((numSteps % 64) == 0) {, edit the number shown as ‘128’ to adjust number of measurements taken in a 360° rotation of 2048 steps total. (ex. 64 is from 2048/32 measurements per rotation = 64)

4. Setting up the Microcontroller

i. After configuring the number of measurements per rotation: Click options for target to open it > Open the Debug Window > Use: CMSIS-DAP Debugger > Click Settings next to the drop down > Select XDS110 and Close Window

ii. Click ‘Translate’ > ‘Build’ > ‘Load’ to upload firmware to Microcontroller

iii. Once Flashed, hit the Reset Button as indicated in the image below

iv. All LEDs will flash once, following 2 flashes of the PN0 to indicate proper initialization of all parameters

5. **Operating the Spatial Mapping Device**
    i. Place the device on a sturdy surface like a chair with the sensor initially facing down, the sensor should be scanning in the y-z plane assuming you will be taking steps forward (in the x plane direction)
    ii. In the MatLab software, click Run, follow the prompts by clicking enter to start receiving data from UART for that layer
    iii. Once ready, commence the scan by hitting the onboard PJ1 push button
        - Should you encounter any issues during your scan (i.e. someone walks through your scan), you can simply press the PJ1 button once more to temporarily terminate/pause the run, when you press the button again it will resume where you last pressed it.
    iv. Following each completed layer, the software will print onto the Command Window: 'Keep this layer? (y/n):' along with a plotted slice of that layer, follow the prompt and enter your response. Should you have any transmission errors like loose wire connections during you scan, you can redo the scan individually. Follow the prompt and enter your answer.
    v. Once you are satisfied with your slice, proceed to follow the prompts printed on the command window: 'Take a step forward for next layer and press Enter to continue...'
6. **Visualizing the Results:**
    i. After the scan has been completed, the software will close the tof.xyz file along with the COM port, then proceed to generate the 3D visualization plot.
    ii. The visualization is an interactive plot allowing you to move your point of view on the 3D spatial mapping.

Expected Output

The following section relays an example of what the expected output of the device should look like. The example is taken in a hallway in the JHE building at McMaster University, indicated in Figures 3 and 4 showing the orientation of the scan and width of the hallway.

Given the 32 measurements per 360-degree rotation in the y-z plane, along with 500mm step displacements in the x plane. This enabled high accuracy and resolution in the completed scan. Evident in Figures 5 and 6, the hallway initially starts at a given width, then expands outwards slightly before enclosing in again for a single slice, the scan then expands outwards into a long hallway intersection on both sides. The successful scan was able to accurately replicate the change in the physical surroundings.

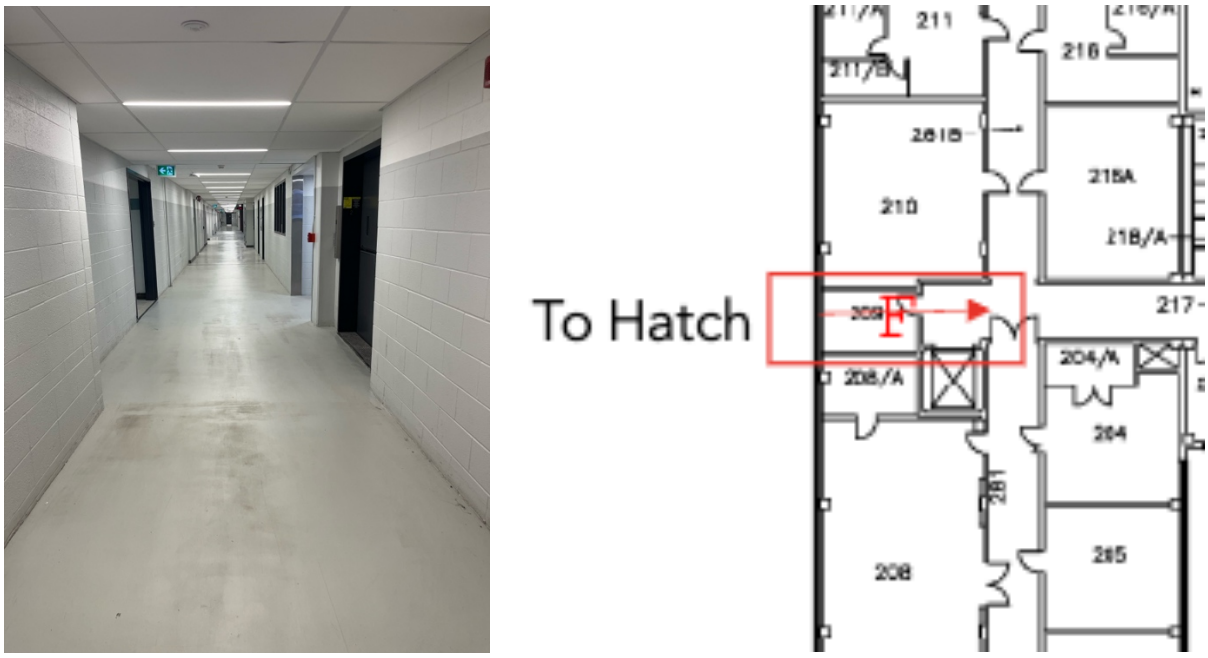


Figure 3 & 4: Scanning Location and Direction of Scan

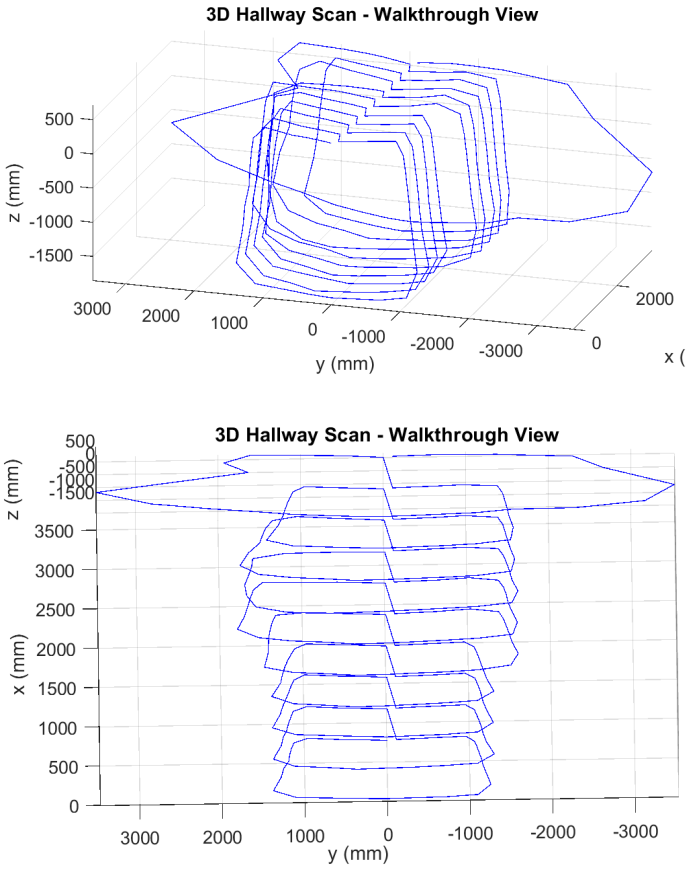


Figure 5 & 6: Complete Spatial Map Scans of JHE (F)

Limitations

1. A limitation of the MSP-EXP432E401Y microcontroller is its Floating Point Unit (FPU), the piece of hardware responsible for performing math operations with decimals such as division, cosine.etc. The microcontroller contains a 32-bit FPU precision, what this entails is that the decimal numbers are represented with limited range and accuracy. Due to the nature of this project needing to convert polar distance measurements into cartesian y and z coordinates, it requires the use of cos and sin trigonometric operands. However, in this specific implementation of the device, the trigonometric cos and sine operations are done externally on the PC using the MatLab software. Should the user choose to implement the operations internally in the microcontroller, then it will raise some inherent limitations in the accuracy and precision of the results.
2. The use of sensors using ADC always present themselves with quantization error, quantization refers to the actual analog voltage level and the digitized voltage level due to rounding in each step of the ADC. In this case the ToF module has a maximum reading of 4000 mm in the 16-bit format, thus from the resolution of the sensor, it has a maximum quantization error 1mm.
3. The maximum standard serial communication rate is limited by the performance of the individuals PC port. Using the application device manager on a PC can enable you to discover the maximum baud rate, being 128000 bps in the case of this PC. For this project, a baud rate of 115200 bps was implemented, this was the best option to fall under both limits of the PC as well as the microcontroller which has a maximum 15Mbps. To verify this, the software RealTerm can be used by setting the appropriate baud rate that matches both ends, if both ends did not match it would lead in frame errors, characters and letters you would not be able to comprehend. Properly setting the baud rate to 115200bps would enable proper understanding of distance content.
4. The communication method used between the microcontroller and the ToF module was the I2C serial communication protocol. In this project, the transmission rate was set to standard mode at 100kbps, however, if needed the transmission rate could be increased to fast mode at 400kbps.
5. One of the primary limitations of this project was the time consumed in taking each full scan of an interior. This was dependent on the speed of the stepper motor and the transmission rate of the ToF sensor. Using the Systick functions the minimum delay between each step was around 1.14ms, as a result given 64 incremented steps per rotation the fastest transmission rate is determined to be 72.96ms per rotation:

$$64 \times 1.14 = 72.96\text{ms per rotation}$$

Circuit Schematic

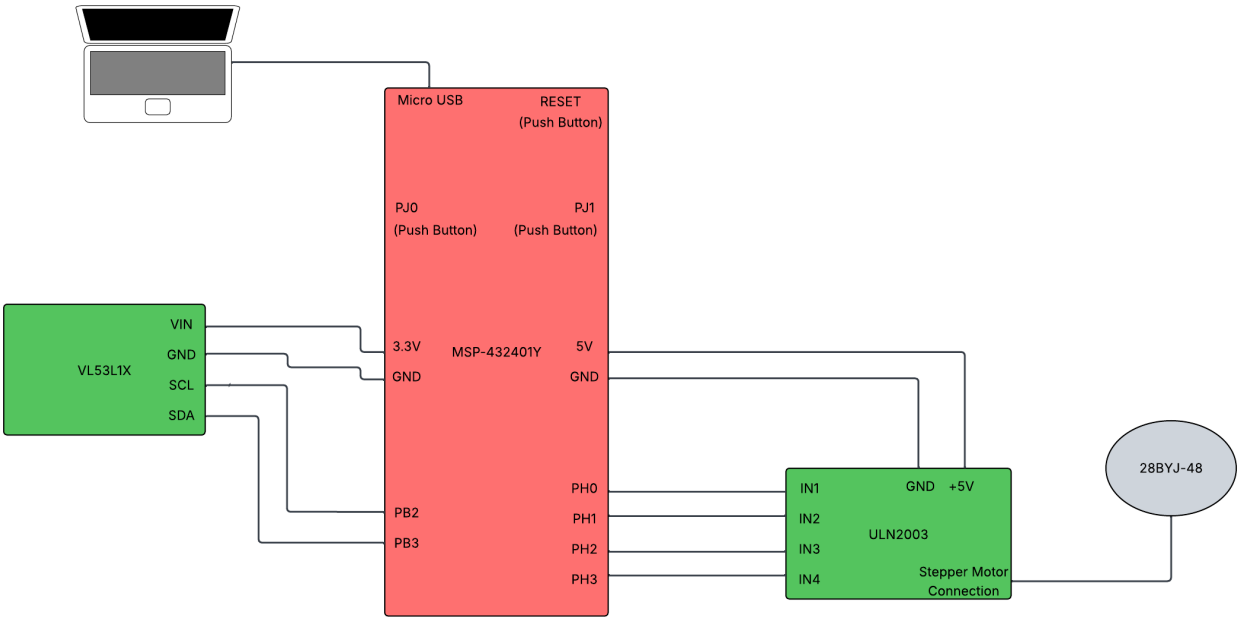


Figure 7: Circuit Schematic Pin Wiring

Programming Logic Flowchart

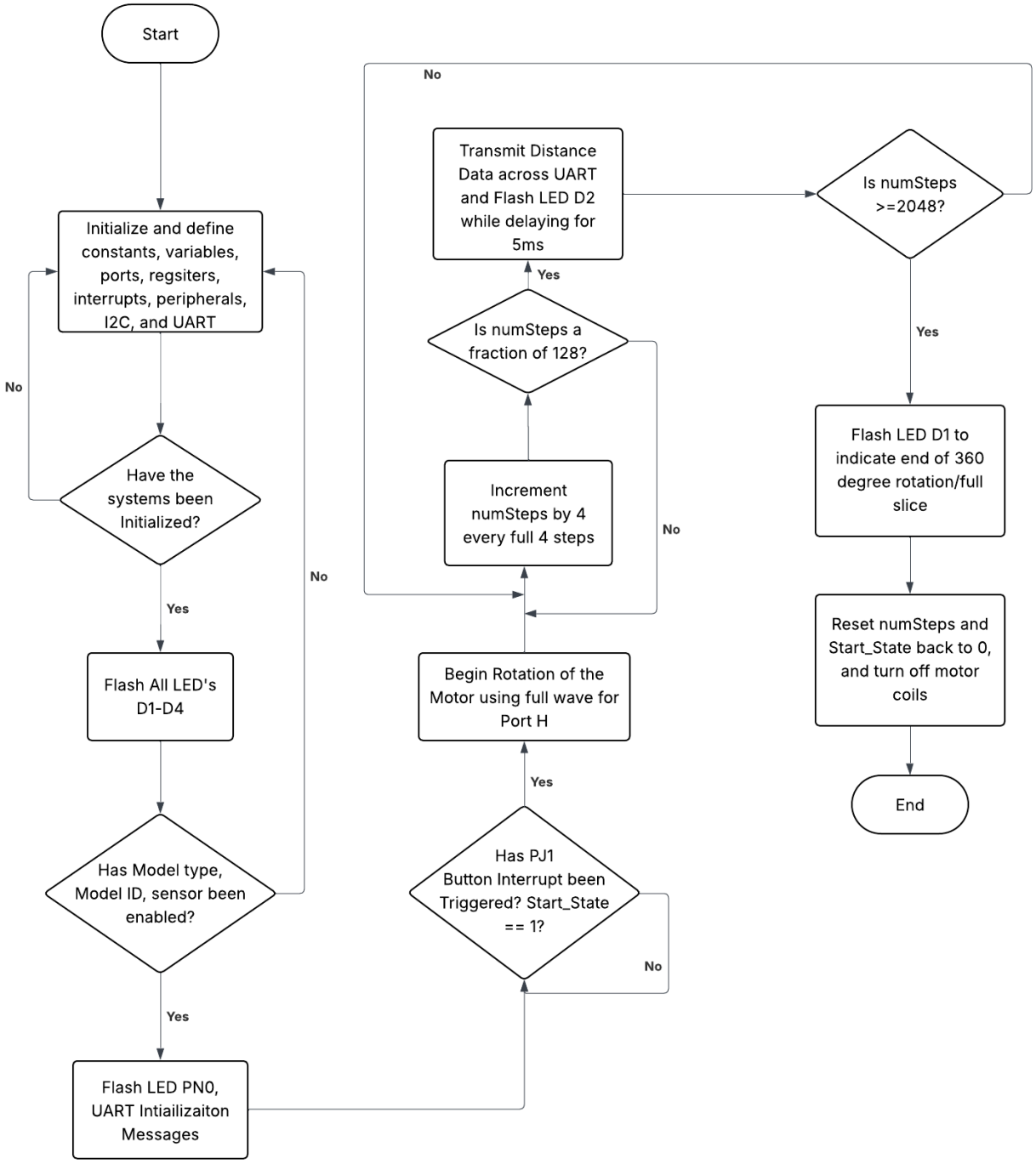


Figure 8: Programming Logic Flowchart for Microcontroller

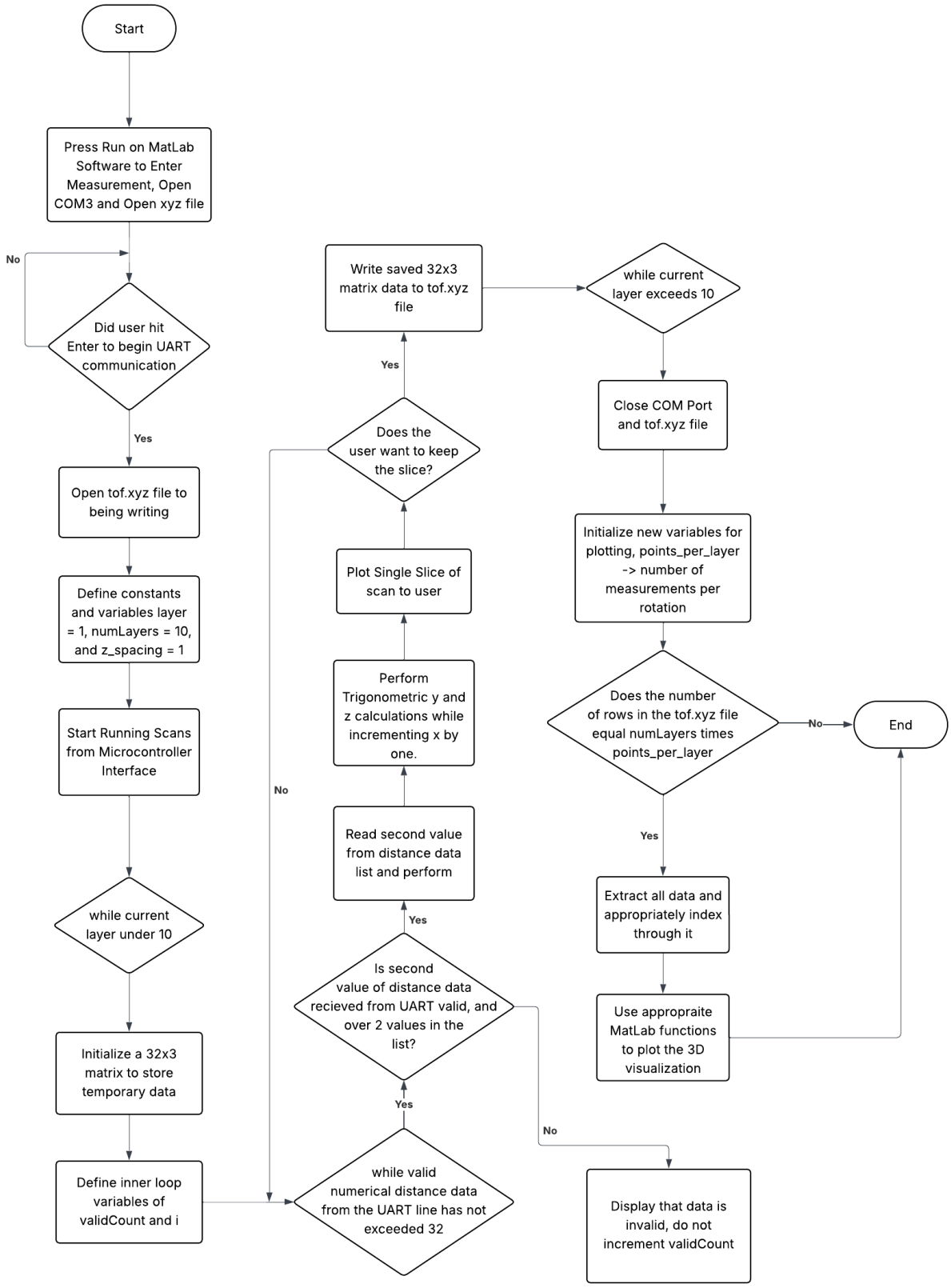


Figure 9: Programming Logic Flowchart for Personal Computer