Spatial Mapping Project

Datasheet

Created By:
Abaan Khan

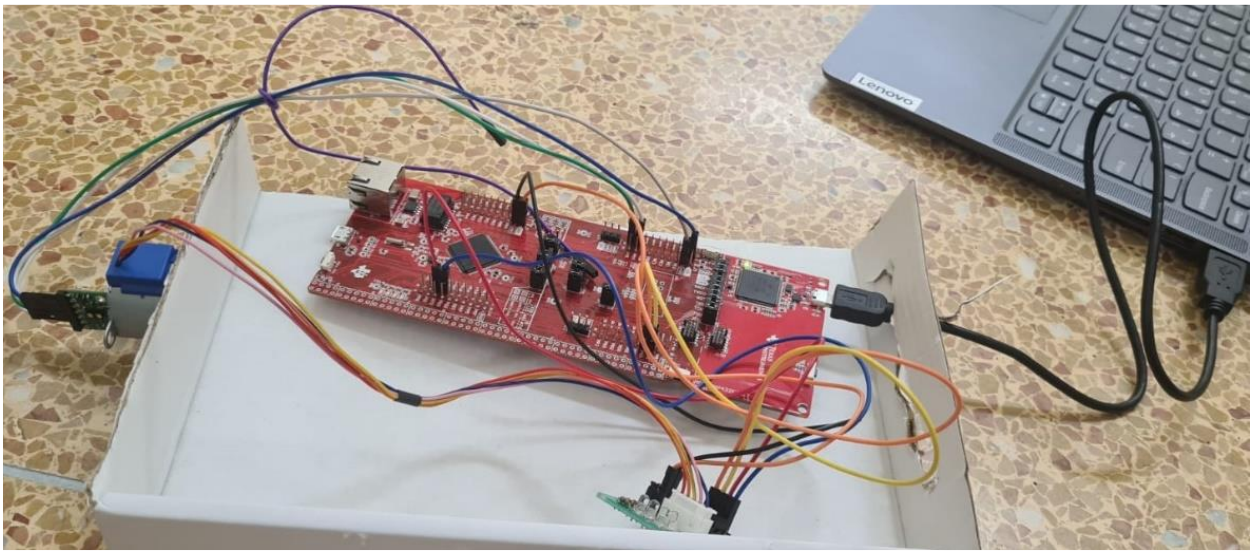


Table of Contents

Device Overview

Features

Texas Instruments MSP432E401Y Microcontroller

- ARM Cortex M4F Processor
- Operating at 120MHz bus speed by default (changed to 96 MHz)
- Range of operating voltage: 2.5-5.5 V

VL53L1X Time-of-Flight Sensor

- Maximum range of 400 centimeters, up to 50Hz ranging frequency for fast and accurate ranging
- Distance measurements in millimeters(mm)
- Range of operating voltage: 2.6-3.5 V

28BYJ-48 Stepper Motor with ULN2003 Driver board

- One rotation (360 degrees) in 512 steps
- Used in Full Step Mode (step angle = 11.25 degrees)
- The driver board has LEDs to indicate different phases
- Range of operating voltage: 5-12 V

User Interface

- Microcontroller push button PJ1 (configured active low), reset button
- Main status microcontroller LED PN1 (D1), additional LED PN0 (D2)

Communication

- MSP432E401Y and VL53L1X – I2C protocol (serial)
- MSP432E401Y and Computer – UART protocol (serial) [via pyserial on Computer]
- Baud rate = 115200 bps

Software and Tools:

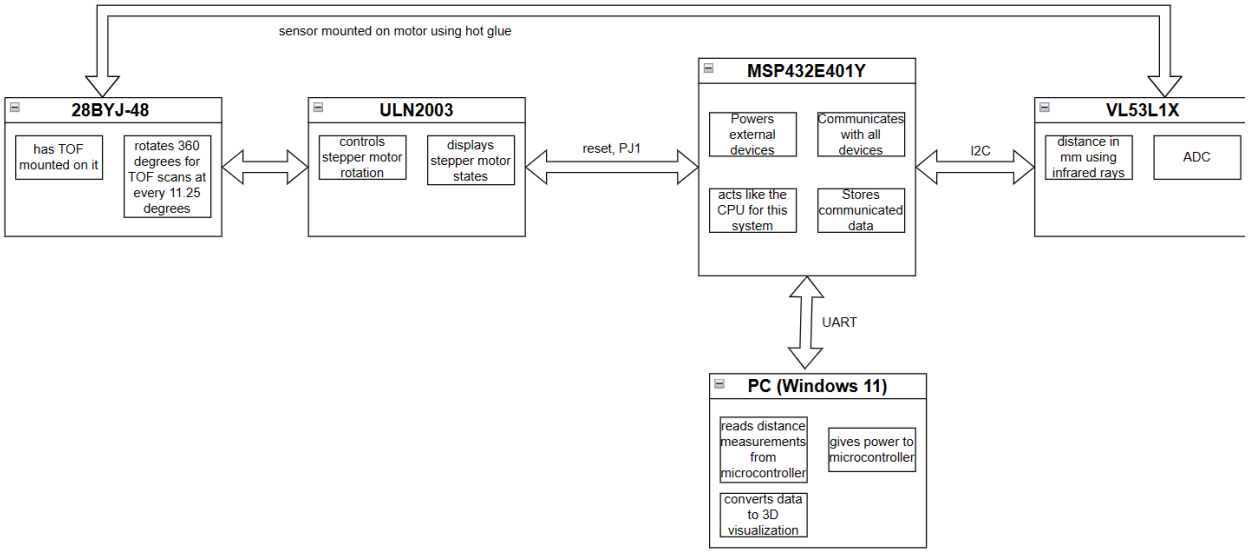
- Keil uvision software using C programming language
- Python 3.8 [pyserial module – communication, open3D module – data visualization]

General Description

This Spatial Mapping System offers a versatile and cost-effective solution for 3D visualization of indoor environments. At its core, the system comprises a combination of components, including a microcontroller, stepper motor, ToF sensor and other user interface elements. The microcontroller orchestrates the operation and communication among these components. Through protocols like I2C and UART, communication is facilitated. The ToF sensor uses infrared rays to accurately measure distances and uses the I2C protocol to send these measurements to the microcontroller. Coupled with the stepper motor, which rotates the sensor

through a complete 360-degree scan, comprehensive spatial data mapping is achieved. User interaction is facilitated through microcontroller pushbuttons (reset and PJ1), allowing for easy initiation, and pausing for scans respectively.
Upon pressing PJ1 the initiation of scans begins; the system collects data points in the xy plane at predefined step angle (11.25 degrees) for a total of 32 scans. Once a scan is done the system returns to home position, and when the user has moved to the next position on the z axis they can once again press PJ1 to begin scan initiation. Data transmission to the PC is facilitated via UART communication, enabling interaction with Python by the power of Python libraries such as pyserial and Open3D. The different slices, along z axis, of the data points collected on xy plane are transformed into 3D visualizations using the open3d python library, providing viewers with visual insights about the mapped environment.

Block Diagram



Device Characteristics Table

MSP432E401Y		VL53L1X		ULN2003	
Bus frequency	96 MHz	Vin	3.3 V	V+	5 V
Onboard LEDs	PN1, PN0	GND	GND	V-	GND
Buttons	PJ1, reset	SCL	PB2	IN1	PH0
Port	COM 7	SDA	PB3	IN2	PH1
Baud Rate	115200 bps			IN3	PH2
				IN4	PH3

Note: Every computer may have a different USB port in use. Change the port in the Python script to match the one being used for UART communication by checking the port in, Windows Device Manager > Ports.

Detailed Description

Distance Measurement

The system component which is responsible for gathering and transmitting distance measurements is the VL53L1X TOF sensor, which has a 27-degree field of view and a maximum measurement range of four meters. Infrared light is emitted by the emitter, reflected off a nearby object, and then detected by the receiver. The distance of the object in millimeters is determined by the TOF by timing the interval between emission and reception. The distance can be calculated using the equation below.

$$distance = \frac{travel\ time}{2} \times speed\ of\ light$$

The travel time is divided by two because it is the total time for photons of light to reach the object and then bounce back to the receiver.

After taking the distance measurement, the TOF sensor takes care of several processes, which include transduction, conditioning, and ADC, producing a digital version of the analog distance measurement. First the sensor goes through the transducer process in which the non-electrical signal is converted into an electrical signal. The signal is then conditioned, whereby the continuous signal is converted into a discrete signal. Lastly, this analog signal is converted into a digital signal. This is all done by the ToF sensor for us. Once the data is converted to a digital signal, it is sent to the microcontroller using the I2C protocol.

To activate the ToF sensor and initiate ranging, the ToF sensor boot function is invoked in the C program. Although the sensor is now active and primed to take distance readings, this process does not occur immediately. Instead, the microcontroller waits for a trigger from its peripherals through an interrupt method. This occurs with the use of an onboard push button PJ1, which when pressed rotates the stepper motor for a full rotation (360 degrees) while stopping it at every 11.25 degrees (32 scans). Distance measurements are retrieved from the TOF's API functions. These 32 distance data measurements are sent to the microcontroller using I2C and are stored in the microcontroller's onboard memory.

Visualization

An HP Elitebook computer (Windows 11) with 11th gen Intel core i-7 processor @2.8 GHz with 32GB RAM is used for the visualization process. This is not essential to the system's functioning and can be altered. Keil and Python v.3.8 are used to compute and process all the data received from the microcontroller and ToF using some python modules mentioned above. A USB-A port is used by the computer. The microcontroller communicates with the computer using the UART communication protocol and transfers the distance measurement data. The Baud rate is 115200 bps, the port used for UART communication is COM7(different for each computer) and there is a timeout of value 10 specified.

The data sent by the UART is broken down and obtained via specified Python code, and it is then saved into a text file by using an array. From there, the distance measurements may be retrieved directly from the text file and used to evaluate trigonometric formulas to find the x-y coordinates on the x-y plane. This can be done by understanding that we know the distance

measurements (magnitude) and the angle between each measurement, and these two things can be used to obtain the coordinates by using sin and cos functions of the math module in python.

$$x = distance \times \cos(11.25°)$$

$$y = distance \times \sin(11.25°)$$

Note: The angle 11.25 degrees can be changed to any other angle deemed appropriate to get an informative 360-degree scan of the chosen environment.

The z coordinate is manually set in the python code. The xyz coordinates then need to be moved into a xyz file, which can be mapped in a 3D plane by using the python open3d module features. This is how each piece of a scan is created, then all these pieces along the z axis are packed together by open3D to give us a complete scan.

Application Note, Instructions and Expected Output

Application Note

Lidar scanning, like the TOF sensor, is a robust imaging technique with a broad range of applications that uses infrared light to measure distances and produce 3D visualizations of objects and environments. This technology is also being used more and more in the design of autonomous cars, where it helps in obstacle identification and evasion by providing instantaneous information on the vehicle's surrounding. Precise depth measurements are possible with lidar systems installed on marine or aerial platforms. These systems send laser pulses towards the ocean's surface and time how long it takes for the pulses to return.
For engineers and researchers looking for an affordable and adaptable way to visualize indoor surroundings in three dimensions, we have our Spatial Mapping System. This system offers an efficient framework for data collecting and visualization by combining innovative components such the Texas Instruments MSP432E401Y microcontroller, VL53L1X Time-of-Flight sensor, and a 28BYJ-48 stepper motor with ULN2003 driver board.

Instructions

This section assumes all required software (Keil uvision, Python (including pyserial, open3d modules)) have been installed.

1) Make sure all the connections displayed in the circuit schematic have been made following the device characteristics table.

2) First Step for any user is to determine the UART port for their computer from Device Manager > Ports after plugging in the microcontroller to the computer. Once the port has been identified (eg: COM7), change the port name in line 15 of the python code.

3) Open the Keil C code and then, Translate, Load and Build.

4) Click the reset button of the MSP432E401Y.

5) Run the python script, it will ask you to specify number of scans and enter an ‘s’ as a signal to begin reading scans.

6) When in appropriate position to start scanning, enter an ‘s’ on the python script and press PJ1 (interrupt trigger) on the MSP432E401Y to begin scanning.
7) After each 360 degrees scan and while the motor rotates back to the home position, move the setup forward by the specified z axis distance which is 10 cm by default (can be changed in line 93 of python script). `z_coor = i*100` .
**Note: The user has approximately 9 seconds to complete step 7, this is to maintain the synchronization between taking and reading distance measurements.

8) Once the specified number of scans are done, a mesh visualization of the data will pop up. To view a 3D visualization of the data, close the mesh visualization.
9) Close all opened software and unplug the microcontroller from the computer.

Expected Output

By comparing the 3D visualization with the designated scanning location’s shape, we can see the expected output of the device as shown below. The actual image of the hallway is shown below, along with a side-by-side comparison with the 3D scan created by the finalized circuit design. The completed scan demonstrates that the system can record the specifics of the scanned hallway to a good accuracy, the main reason for this is the chosen small step angle of 11.25 degrees. As seen below, the general layout of the hallways was recorded. The hallway is uniform in most places, however at the beginning there is extra space on top and doors to lecture halls on the right and it is irregular on the left side too because of benches, doors, and windows.

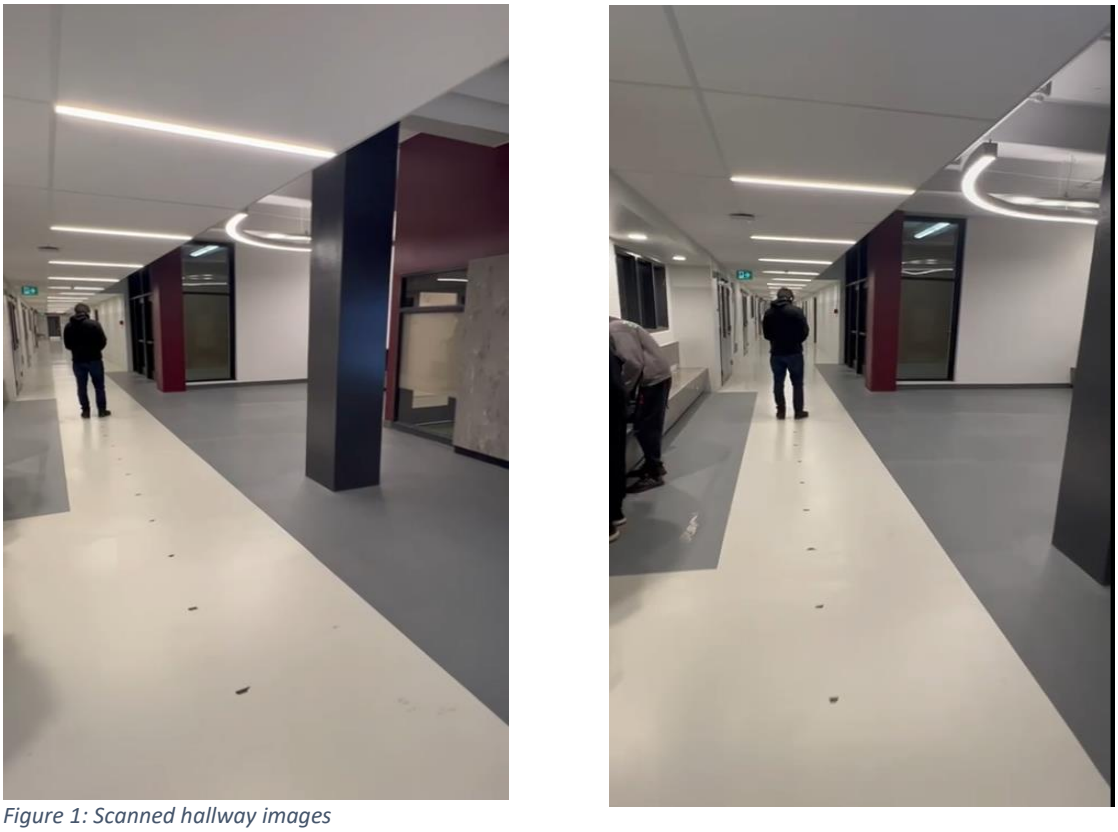


Figure 1: Scanned hallway images

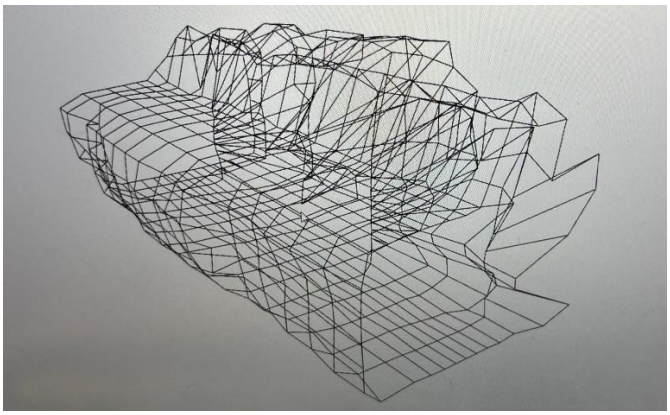


Figure 2: Front view

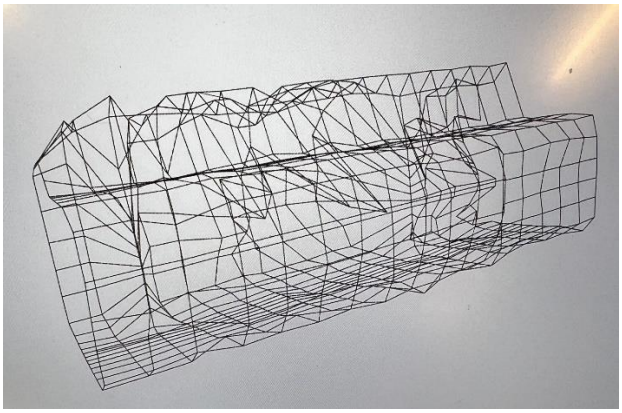


Figure 3: Top view

The above is the mesh 3D visualization , generated by open3D after connecting all the distance points in one slice of distance measurement data and then connecting all the slices along z axis together. The general outline of the assigned hallway can be compared to the scan, and it shows us that a good scan was captured. We can say this because, in Figure 2 the scan captures the lecture door hallways to the right and there are outgrowths on the left which represent the benches.

Before producing the mesh visualization, the open3D software produces a point cloud visualization which gives us brief description of the way distance points have been measured and arranged in the 3D plane. However, this is not as helpful as the mesh visualization.

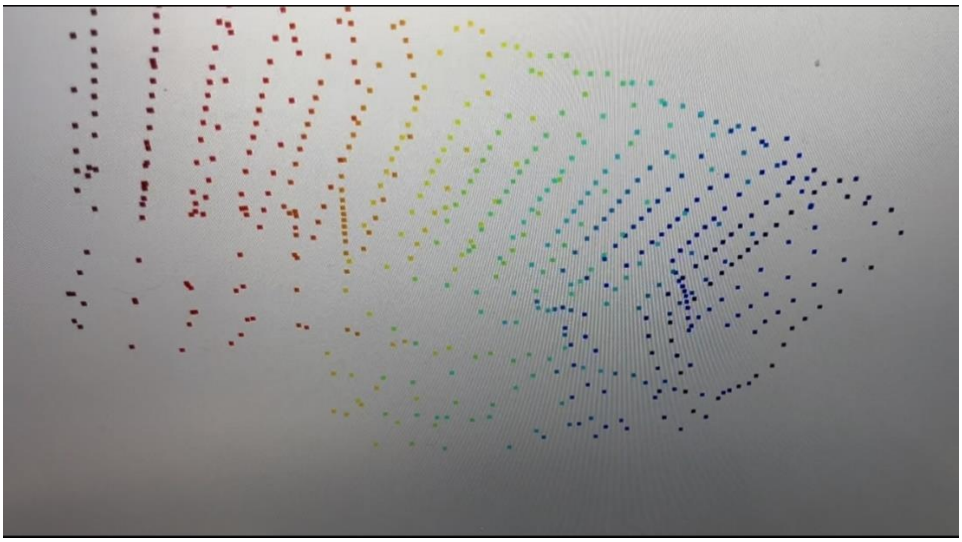


Figure 4: Point cloud visualization (bottom view)

Limitations

1) Although the MSP432E401Y features a Floating-Point Unit (FPU) capable of handling single-precision 32-bit operations, we opted to perform trigonometric calculations in Python using the math library rather than utilizing the microcontroller's capabilities. While this decision hastened the development process, it disregarded potential optimizations offered by onboard processing.

2) The maximum quantization error for the Time-of-Flight (ToF) sensor was determined to be,

$$Mazimum\ quantization\ error = \frac{Maximum\ reading}{2^{ADC\ bits}} = \frac{4000\ mm}{2^{16}} = 0.061\ mm$$

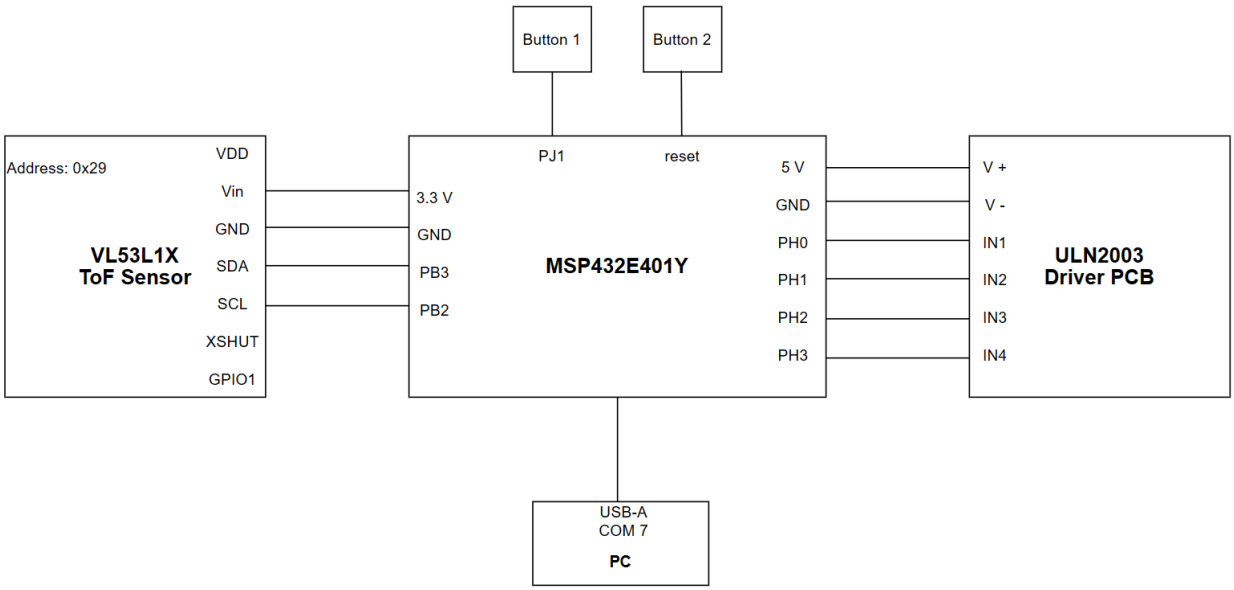
It was calculated based on the 16-bit resolution of the ADC and a maximum distance measurement range of 4000 mm. This quantization error, although relatively small, introduces a level of uncertainty into distance measurements that may impact precision-critical applications.

3) Serial communication between the microcontroller and PC is constrained by the PC's maximum standard serial communication rate, set at 128,000 bits per second. This can be checked for any PC from [Device manager > Ports > Right click on port > Properties > Port settings]. Despite the microcontroller's capability of supporting speeds up to 15 Mbps, the bottleneck is dictated by the PC's limitations. This constraint underscores the importance of compatibility considerations in system design. The baud rate that we used was 115200 bps.

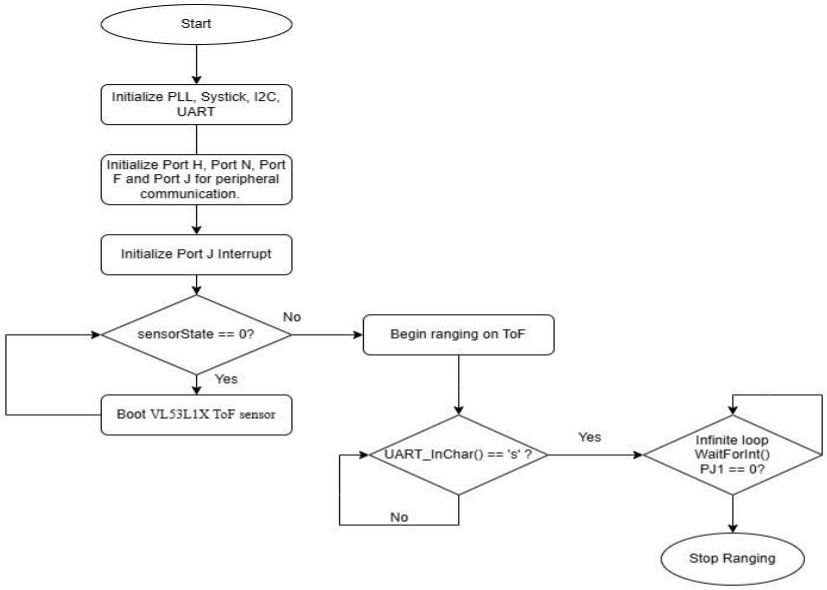
4) The device employs the I2C protocol for communication between the microcontroller and ToF sensor, offering a maximum transmission speed of 3.33 Mbps. However, the UART communication between the microcontroller and PC, utilizing COM7 (different for all PC’s) with a baud rate of 115,200 and imposed bus speed of 96MHz, imposes further limitations on data transfer rates.

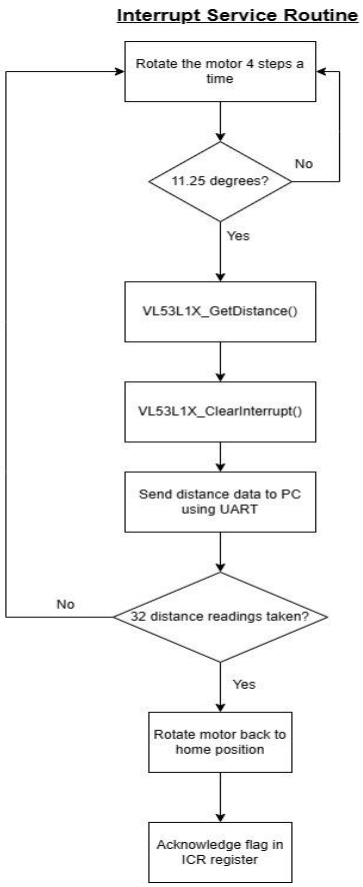
5) The primary roadblock in system speed is attributed to the ranging time required by the ToF sensor since the TOF has a maximum ranging frequency of 50Hz. While the motor's rotation speed contributes to minor delays, it remains relatively fast in comparison to the ranging process. The ToF needs a considerable amount of time to calculate the distance as well as to send and receive the infrared rays. The results showed too many errors while testing with reduced delays for this. As a result, before completing the full 360-degree turn, the motor must temporarily stop rotating. This need is the main restriction that slows down the system's overall speed.

Circuit Schematic



Programming Logic Flowchart





References

1) Texas Instruments, "MSP432 SimpleLinkTM Microcontrollers – Technical Reference Manual", 2018, Ch. 4, 17, pp. 326 – 327, 1201 – 1252.

2) "Vl53l1x.pdf," vl53l1x.pdf, https://www.st.com/resource/en/datasheet/vl53l1x.pdf , 2022 (accessed Apr. 16, 2024).