Microprocessor Systems Project 2DX3
Final Project Report

Instructors: Dr. Athar, Dr. Doyle & Dr. Haddara

Ashviya Jeyaseelan

As a future member of the engineering profession, the student is responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is my own and adheres to the Academic Integrity Policy of McMaster University and the Code of Conduct of the Professional Engineers of Ontario.

Table of Contents

1. Device Overview

a. Features

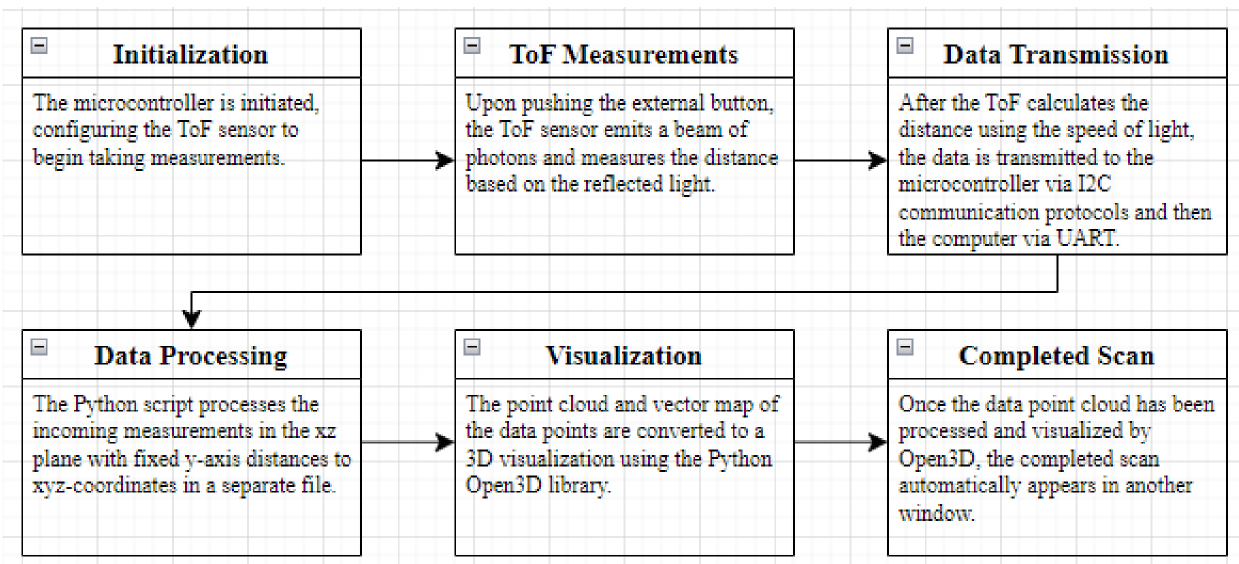
The embedded spatial mapping system incorporates the following devices: Texas Instruments MSP432E401Y microcontroller, VL53L1X Time of Flight Sensor (ToF), 28BYJ-48 stepper motor with ULN2003 driver board, external push button, and computer software for data collection and visualization.

Device	Description
Texas Instruments MSP432E401Y Microcontroller	The MSP432E401Y microcontroller contains the following features [1]: • 32-bit Cortex M4 processor core • 256 kB of SRAM, 1024 kB of flash, and 6 kB of EEPROM • Default bus speed of 120 MHz, but configured to 48 MHz • Consists of 90 GPIO pins, 8 UARTs and 10 I^2Cs • UART communicated with the computer
VL53L1X Time-of-Flight Sensor (ToF)	The ToF sensor contains the following features [2]: • Accurate distance ranging up to 4 m • Communicates with the microcontroller using I^2C • Operating voltage ranges from 2.6 V to 5.5 V • Operating frequency of 50 Hz
28BYJ-48 Stepper Motor with ULN2003 Driver Board	The stepper motor contains the following features: • Requires four phases for every step • Operating voltage ranges from 5 V to 12 V • 512 steps are required for a 360° rotation • Four LEDs to indicate each phase
External Push Button	The push button contains the following features: • Supply voltage of 3.3 V from the microcontroller • Operates on an active low configuration with a pull-up resistor
Computer Software	The embedded system collects and visualizes the data with software contains the following features: • Connects to computer using UART protocols with COM port • Python is used to collect data from the COM port, store data in a .xyz file, and visualize the points with Open3D

b. General Description

The spatial mapping system aims to scan and three-dimensionally visualize enclosed spaces in an efficient, inexpensive manner, using Commerical Light Detecting and Ranging (LIDAR) with a Time-of-Flight ToF sensor. Initiating the system with an external button, the ToF sensor mounted onto a stepper motor rotates every 11.25° as indicated by the flashing of an onboard LED to measure the distances of the space in the xz-plane with fixed distance samples along the y-axis. Once the stepper motor completes a 360° rotation in the clockwise direction, it rotates counterclockwise to its home position to prevent tangling. Employing the I^2C protocol, the data interpreted by the ToF is communicated to the microcontroller, which then transmits the data to the designated computer through a USB COM port using UART communication. The Python script converts the incoming data into the xyz-coordinates to be stored in a separate file that is visualized into a 3D vector map using the Python Open3D library.

c. Block Diagram



2. Device Characteristics

Device	Feature	Configuration
Texas Instruments MSP432E401Y Microcontroller	Default COM Port	COM 5
	UART Baud Rate	115200 bits/s
	Bus Speed	48 MHz
	Measurement Status LED	PF4 on MSP432E401Y
	Additional Status LED	PN1 on MSP432E401Y
VL53L1X Time-of-Flight Sensor (ToF)	SDA	PB3 on MSP432E401Y
	SCL	PB2 on MSP432E401Y
	V_{IN}	3.3 V
28BYJ-48 Stepper Motor with ULN2003 Driver Board	IN1	PH0 on MSP432E401Y
	IN2	PH1 on MSP432E401Y
	IN3	PH2 on MSP432E401Y
	IN4	PH3 on MSP432E401Y
	V_{IN}	5 V
External Push Button	Input	PM0 on MSP432E401Y
	V_{IN}	5 V

3. Detailed Description

a. Distance Measurement

The embedded spatial mapping system consists of the VL53L1X ToF sensor, which takes measurements of enclosed spaces based on Commerical Light Detecting and Ranging (LIDAR) technology. The ToF releases a beam of photons in the form of infrared light through its emitter and measures the amount of time that is taken for the reflected light to return through its receiver, calculating the average distance in millimeters to the object using the speed of light with the following formula: $d = \frac{c\,\Delta t}{2}$, where d is the distance, Δt is the difference in time, and c is the speed of light constant.

The mapping system is initiated by the external push button, which rotates the motor and momentarily pauses every 11.25° to allow the ToF sensor to collect data as indicated by the flashing of the onboard measurement status LED (PF4) each time a measurement is taken. After the ToF sensor's board performs transduction, signal conditioning, and analog to digital conversion, the data is then transmitted to the microcontroller using the 1^2C protocol through the serial data line (SDA) and serial clock line (SCL). To construct a visual map of the collected points, the data must be transmitted to the designated computer through the USB connection with the microcontroller using the UART communication protocol.

b. Visualization

While the microcontroller is run by a C program on Keil uVision5, the data collection and visualization are performed by a Python program. The program for storing the data receives the measurements from the microcontroller through the COM 5 port on the computer through UART. The measurements then need to be converted to xyz-coordinates as the mapping system is configured to scan the xz-plane with fixed distance samples along the orthogonal y-axis. The x-axis is along the width of the space, whereas the z-axis is along the height of space, both up to a maximum of 4 m due to the ranging capabilities of the ToF sensor. The y-axis is configured along the length of the space, which is offset by a fixed value that can be set by the user in between each scan. In the demo, the number of scans was configured to three with a 10 cm offset in between each scan. To calculate the coordinates for each scan, trigonometric relationships are used, where $x = d\,cos\theta$ and $z = d \sin\theta$. Once the conversions are complete for all the scans, the data is stored in a .xyz file.

An additional Python script completes the 3D visualization of the scanned space using the previously converted xyz coordinates. The .xyz file is read into an array using the numpy library, allowing the points to be connected with lines, and then transformed into a 3D visualization using the Open3D library. The final image is then illustrated in a pop-up Open3D window, displaying the spatial map of the scanned space.

4. Spatial Mapping System Information

a. Application Note

Commerical Light Detection and Ranging (LIDAR) equipment is an advanced technology that can be employed to generate a 3D map of a location and offers several real-world applications. LIDAR involves emitting a light and measuring distances based on the reflection, serving as an inexpensive, efficient method of mapping a space. The spatial mapping project in this course provides a thorough understanding of commercial data acquisition systems in the industry.

With the rise in intelligent systems such as autonomous vehicles, LIDAR technology serves as a powerful tool in the transportation industry that can improve traffic congestion, reduce the number of accidents, and enhance autonomous vehicle management [3]. LIDAR sensors provide mappings of intersections, providing insights on traffic patterns and user interactions on the road, which can be used to improve safety and lead to significant advancements in crash monitoring and smart road infrastructure.

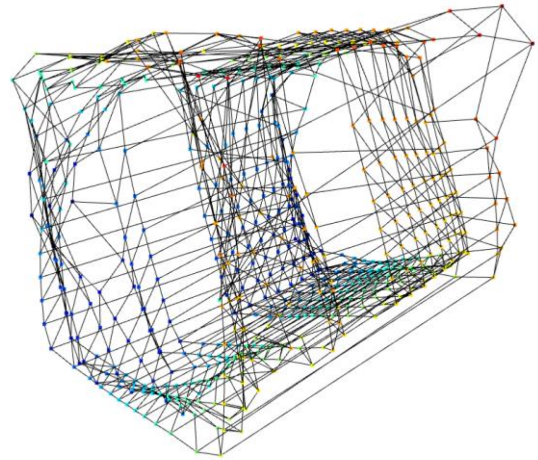
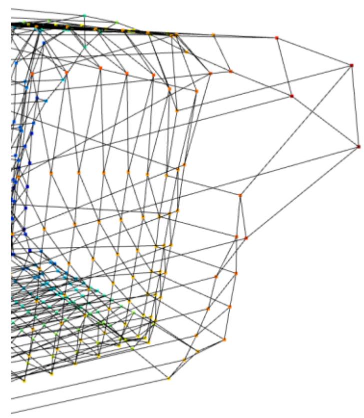
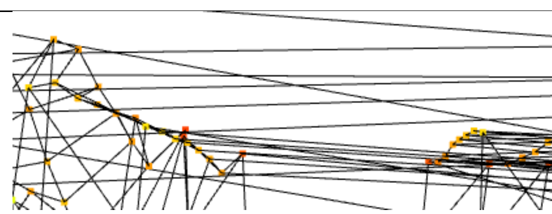
b. Instructions

The spatial mapping system requires the following software to be installer on the user's computer: Python 3.11, Keil uVision5, the necessary Python libraries, and optionally VSCode.

1. Connect the MSP432E401Y microcontroller into the computer and locate the device's COM port in "Device Manager" to initiate UART communication and data transmission.
2. Change the 'MYPORT" variable in the "DataCollection_Demo.py" Python script to the COM port number from the previous step to allow for data transmission between the microcontroller and the computer. Modify the number of scans in the "SCANS" variable and the y-axis offset in the "OFFSET" variable.
3. Open the Keil uVision project file and compile the code onto the microcontroller by clicking the following in order: translate, build, and load.
4. Press the onboard "RESET" on the microcontroller, which initiates the configuration and readies the system. (Note: This step is not necessary if the program has already been flashed.)
5. Open the "DataCollection_Demo.py" file and run the Python script in the terminal of an IDE such as VSCode.
6. Press the external push button to initiate a scan, which rotates the stepper motor clockwise. At the end of the scan, the stepper motor turns counterclockwise back to its initial position.
7. Move the device at the indicated y-axis offset and push the button to conduct another scan. Repeat until all the scans have been completed.
8. Open the "DataVisualization_Demo.py" file and run the Python script in the terminal. An Open3D window automatically pops up and generates the 3D map of the scanned space.

c. Expected

Hallway Features	Open3D Visualization

The hallway segment is located at zone C on the second floor of ITB. 	
There is an extension in the visualization is due to another hallway at the end of the target hallway. 	
There is an indent inward for the ridge of the door in the hallway segment.	

As demonstrated in the table, it is evident that the hallway is accurately map due to the ridges, indents, and bulges in the 3D visualization. The mapping is accurate except over areas where there is significant reflection due to ceiling lights, the reflective floor, and the hallway with a depth beyond the ToF's maximum ranging distance of 4 m.

5. Limitations

1. Limitations of the Microcontroller Floating Point Capability

The MSP432E401Y microcontroller consists of a floating-point unit (FPU) in its arithmetic logic unit (ALU), which allows for calculations with floating-point values. This results in a maximum of 32 bits, which may contribute to minor inaccuracies in calculations when the bits exceed this threshold. The use of trigonometric functions in the spatial mapping system may yield results with several decimals, reducing the overall accuracy of the distance calculations. Upon evaluating the microcontroller's floating-point capabilities, the design choice was made to have the Python script perform the trigonometric calculations as a computer's processing and floating-point capability is much higher, yielding more accurate results.

2. Maximum Quantization Error for each of the ToF Module

$$Maximum\ Quantization\ Error\ = \frac{Max\ dis\tan c\ e}{2^{ADC\ Bits}} = \frac{4000\ mm}{2^{16}} = 0.061\ mm$$

$$Maximum\ Quantization\ Voltage\ Error\ = \frac{3.3}{2^{8}} = 0.013$$

3. Maximum Standard Serial Communication Rate in PC

The maximum standard serial communication rate that can be implemented is dependent on the hardware of the designated computer. The computer that implements the spatial mapping system has a maximum baud rate of 128000 bits per second, which can be verified in the "Device Manager" settings under the specified COM port. In the system, a baud rate of 115200 bits per second is chosen for quicker results.

4. Microcontroller and ToF Modules Communication

The microcontroller and the ToF modules communicate using the I^2C protocol at a speed of 100 kbps.

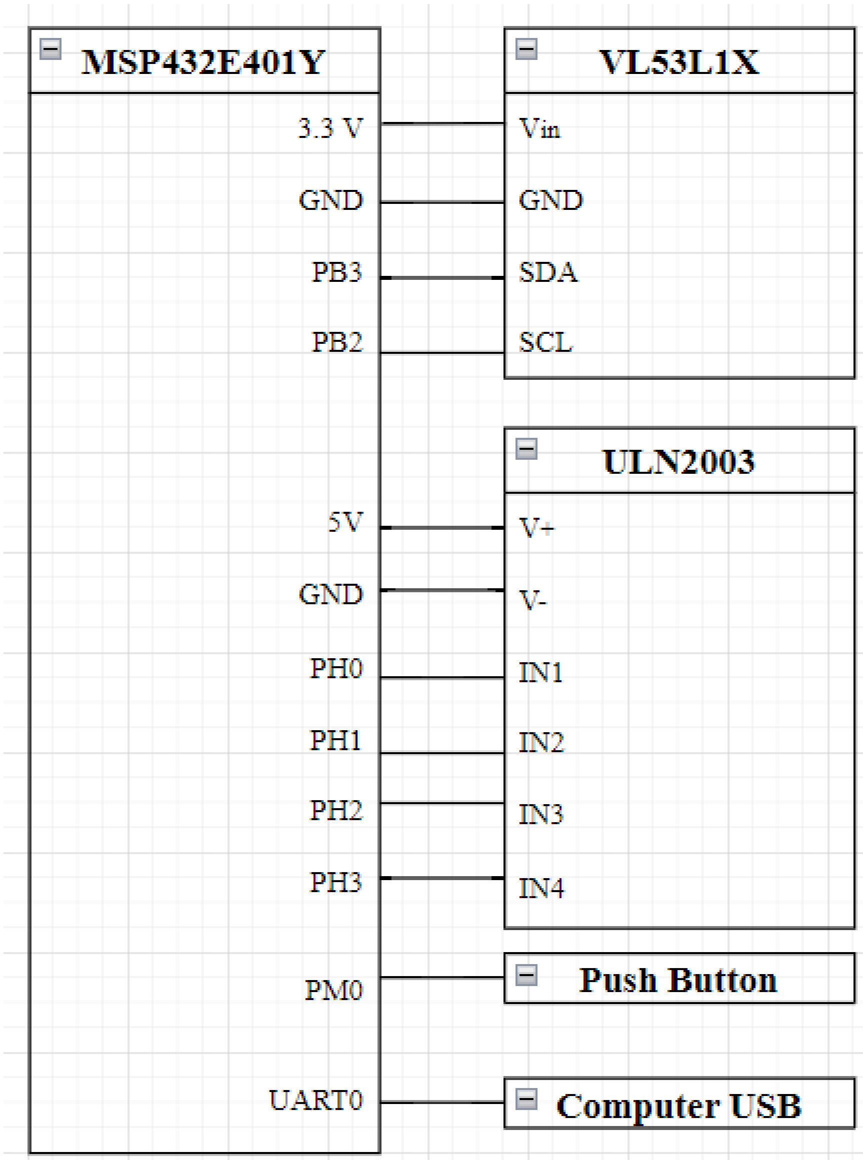
5. Limitations on System Speed

The primary limitation on the system speed is the stepper motor, as it employs the full step method and requires four phases for each step with a maximum speed of 15 RPM. To prevent tangling of wires, the motor performs a scan clockwise and then returns to its initial position in the counterclockwise direction, further slowing down the system. To test the limitations of the motor, the wait time was modified to change the delay in between steps. A reduced delay resulted in quicker runtimes, but the motor eventually stopped rotating when the delay became too low.

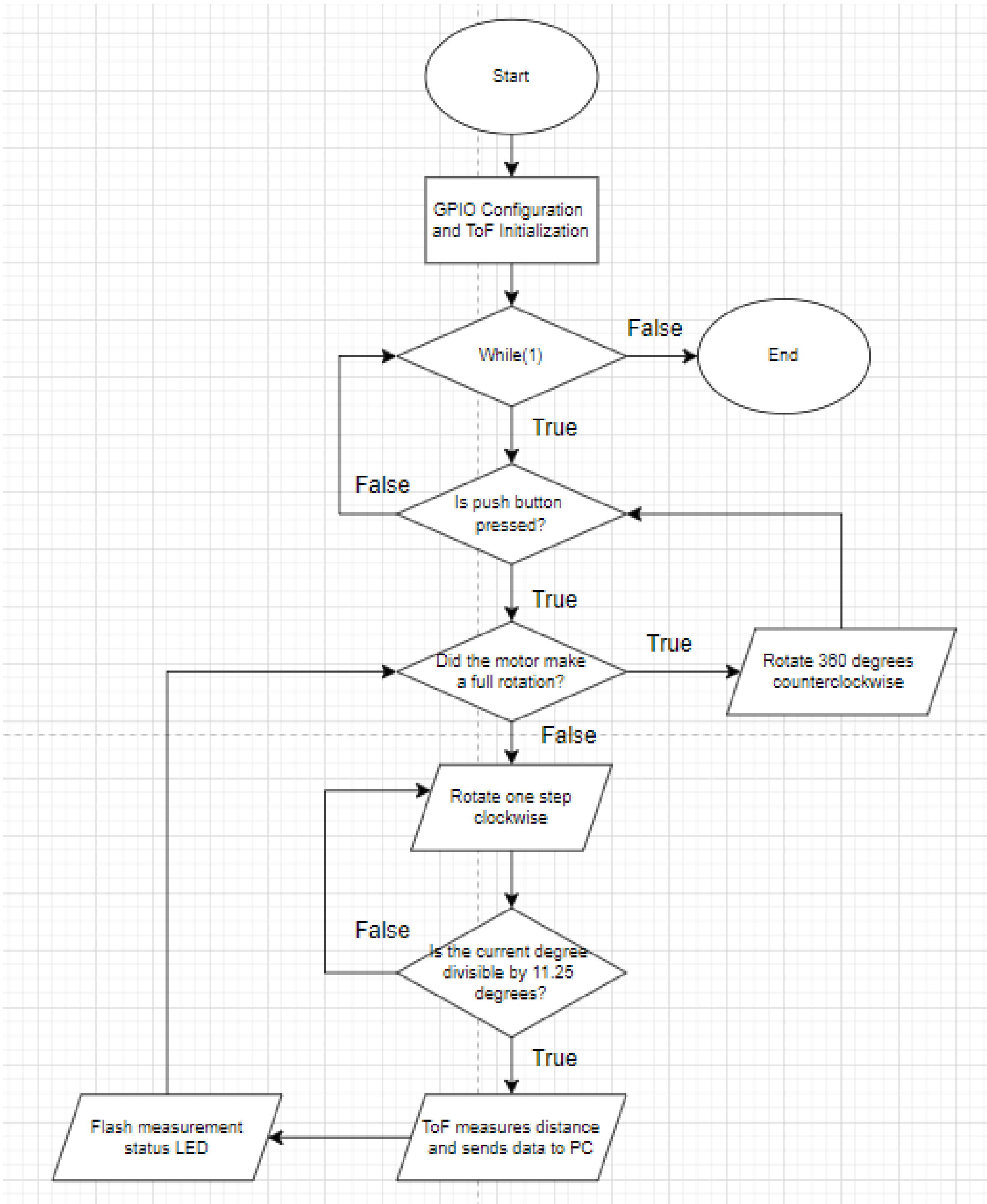
6. Assigned System Bus Speed Calculations

The assigned bus speed is 48 MHz, which was configured by changing the PSYSDIV variable in the PLL.h using the following formula: 480 MHz / (PSYSDIV + 1) = 48 MHz. Upon rearranging the formula, the PSYSDIV variable is calculated to be 9.

6. Circuit Schematics



7. Programming Logic Flowchart



References

[1] "MSP432E4 SimpleLink Microntrollers – Technical Reference Manual," *Texas Instruments* [Online]. Available: https://avenue.cllmcmaster.ca/d2l/le/content/557632/viewContent/44841118/View. [Accessed April 6, 2024].

[2] "VL53L1X Time-of-Flight Distance Sensor Carrier with Voltage Regulator, 400cm Max," *Pololu*. [Online]. Available: https://www.pololu.com/product/3415. [Accessed April 6, 2024].

[3] C. Vickers, "New roadside LiDAR sensors help build a safer transportation infrastructure," *University of Nevada*. [Online]. Available: https://www.unr.edu/nevada-today/news/2020/hao-xu-roadside-lidar. [Accessed April 6, 2024].