2DX3: Microprocessor Systems Final Project

Instructors: Drs. Boursalie, Doyle, and Haddara

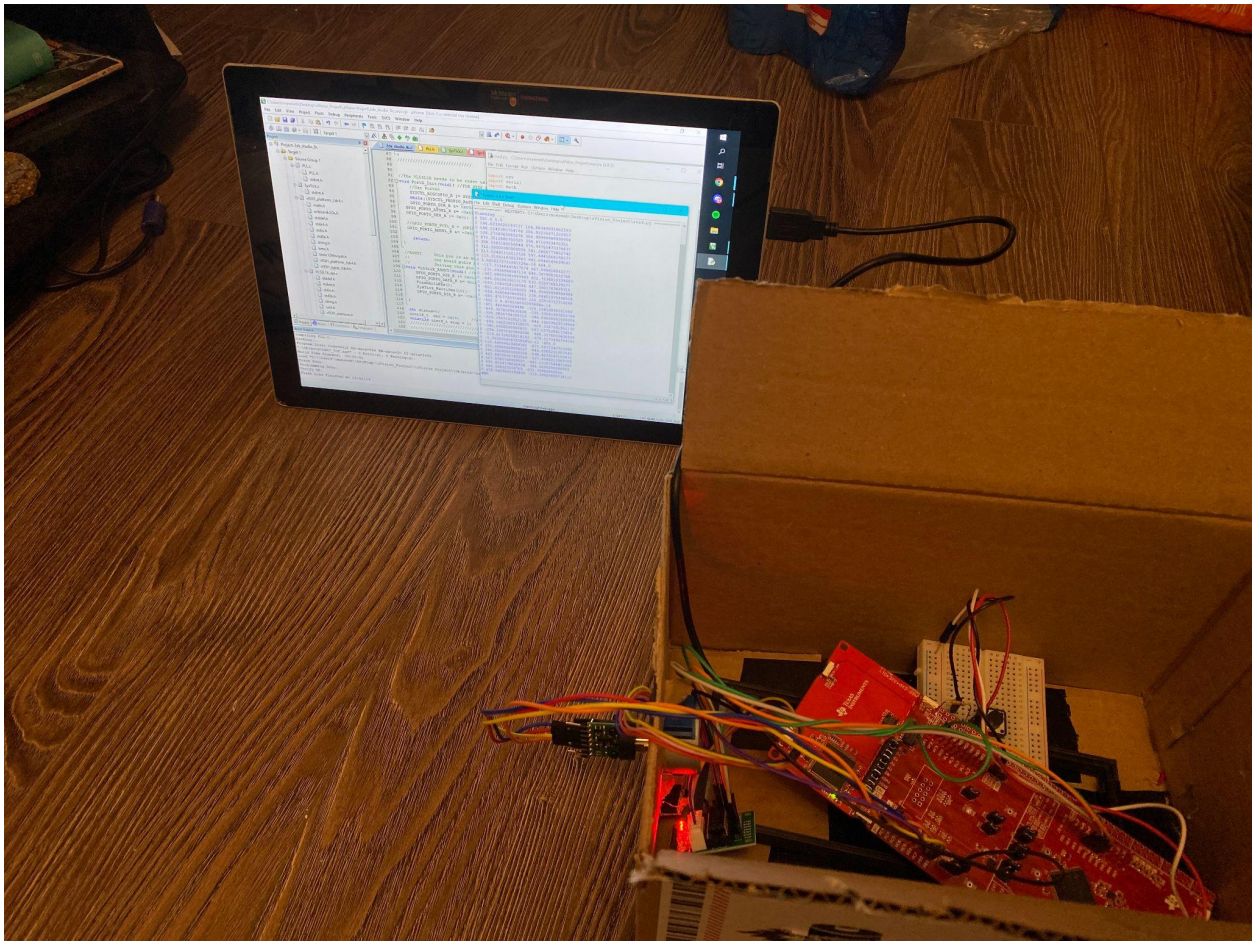
Muneeb Shah – shahm58– 400307005– COMPENG 2DX3

2DX3 – Tuesday Afternoon – L08

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. Submitted by [Muneeb Shah, shahm58, 400307005]

Report Format

Device Overview



Features

- Texas instruments MSP432E401Y Microcontroller
    - Student specific bus speed and distance status
        - Least significant digit is 5, so assigned bus speed is 48 MHz
        - Second least significant digit is 0, so distance status is PN0
    - Approximately $75.00
- VL53L1X Time of flight distance sensor
    - Up to 400 cm range
    - Approximately $20.00
- ULN2003A-PCB stepper motor
    - 512 steps for a 360 degree rotation
    - LEDS to indicate steps of the motor
    - Requires minimum 5V to operate, with a maximum of 12V
- ToF sensor and microcontroller communicate with each other using I2C serial communication

- Microcontroller and PC communicates using python
    - Python version 3.8.5
    - pySerial installation required through command prompt
    - 115200 baud rate
- Open3D python library used to map data to create a 3D reconstruction
    - open3D installation required through command prompt

General Description

With the use of the Texas Instrument MSP432E401Y microcontroller, a VL53L1X time of flight sensor, and the assistance of a stepper motor with the sensor attached to it, users are able to develop a 3D reconstruction of a hallway. The microcontroller uses I2C serial communication to communicate with the ToF sensor which allows the system to collect data and output it to the PC. The sensor is connected to the PB2 and PB3 pins on the microcontroller from the SCL and SDA pins respectively. The microcontroller and laptop communicate using UART protocols with a baud rate of 115200 bits per second. When the PJ1 button on the microcontroller is pressed, the ToF sensor is activated along with the stepper motor to collectively rotate 360 degrees and gather distances to create a map of the surroundings, or in this case, a map of the hallway. The sensor collects data every 10 degrees, while the user moves up every 300mm. Obtaining the x, y, and z coordinates can be done using some math and conversions. The reason for this is because the stepper motor only operates in degrees, whereas python operates only in degrees. The chart below shows how the coordinates are obtained.

x coordinate	300 * (sample number)
y coordinate	r * math.sin(math.radians(theta))
z coordinate	r * math.cos(math.radians(theta))

Table 1: calculations for coordinates

The MSP432E401Y contains a Cortex-M4 CPU. Within the project, the microcontroller is considered the heart, and all the other components are tied together by it. The VL53L1X ToF uses infrared light that measures the time the light is taking flight for until it reaches a surface. It uses I2C (inter-integrated circuit) serial communication with the microcontroller. The stepper motor uses a minimum of 5V (the microcontroller supplies either 3.3V or 5V) and has the sensor attached to it via a 3D printed component.

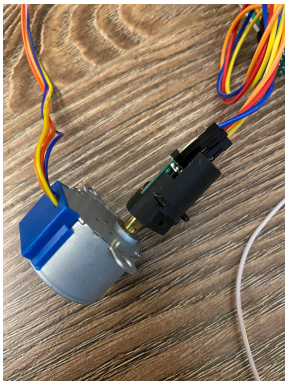
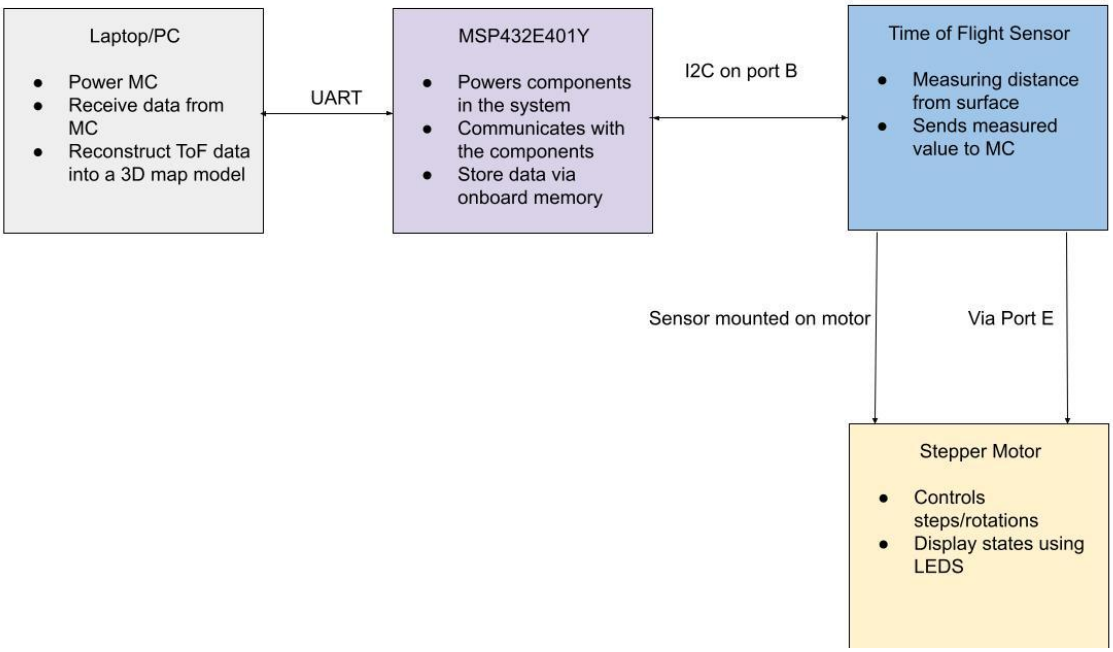


Figure 2: 3D Printed Component

Block Diagram



Device Characteristics Table

Feature	MSP432E401Y Microcontroller	VL53L1X ToF sensor	Stepper Motor
Description	• 48 MHz bus speed • COM6 serial port • UART communication using python • 115200 baud rate • PJ1 inbuilt button • Stop button via PK0	• SDA to PB3 pin • SCL to PB2 pin • Vin to 3.3V pin • Gnd to Gnd pin	• In1 to PM0 • In2 to PM1 • In3 to PM2 • In4 to PM3 • Vin to 5V pin • Gnd to Gnd pin

Detailed Description

Distance Measurement

The distance is measured using the VL53L1X time of flight sensor which emits infrared light. The light is measured using the time it takes for it to be interrupted via the surface. The time is used to measure distance away from the surface by obtaining the average travel time and multiplying it by the speed of

light, and then dividing the answer by 2. The data is saved onto the laptop onto an xyz file, and is transferred via I2C serial communication.

Visualization

The visualization of the data is done on a 2014 Microsoft Surface Pro 3, with specifications shown in Figure 3 on the right. Version 3.8.5 of python is used, as shown in Figure 4 of the command prompt below when typing "py". Python version 3.9 was being used originally, however, it was not compatible with the libraries used in this project like Open3D.

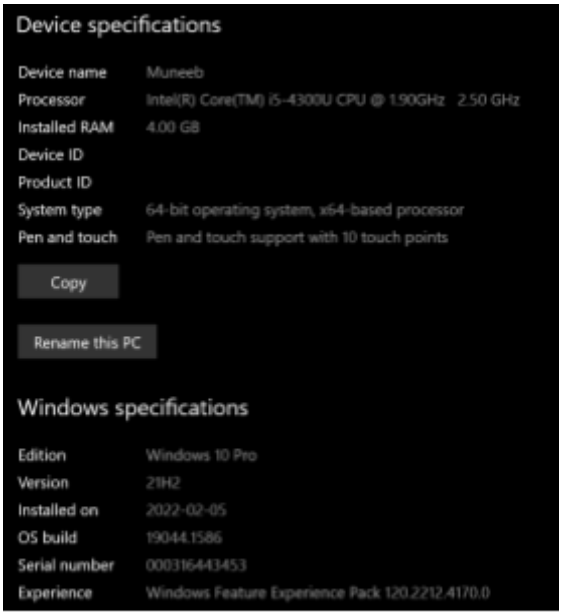


Figure 3: Device Specifications

The data visualization is created by flashing the embedded C code on to the microcontroller, hitting the reset switch on the MC, and then running the python script. The script reads the ToF data and exports it to a .xyz file. Python libraries like open3D, serial, math, and numpy were used to create the python code. Additionally, serial port COM6 was used, and can be confirmed from the device manager of the laptop. Using a while loop with embedded try excepts, the code strips the commas and creates a space in between the coordinates. It converts x y and z by using the formulas from *Table 1* and appends it to the xyz file with commas split between each part of the coordinate. The code also has a constant step distance, but each time the code is run, it will double it to account for each step. There will be 36 samples in the file because every 36 degrees, the ToF sensor records a sample (36*10 = 360).

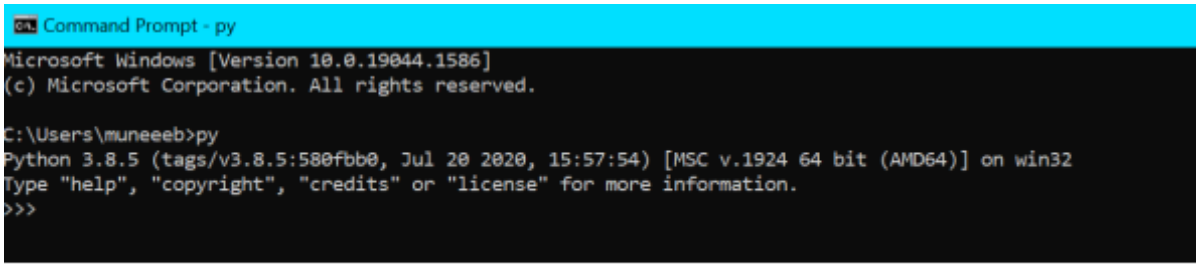


Figure 4: Python Version

Application Example

1. Connect the MSP432E401Y to the laptop with the provided micro-USB cable
2. Have the python file ready to be used in a later step
3. Build, translate, and load the main .c file. This will load the code onto the microcontroller.
4. Verify the terminal that the code was successfully loaded without any errors.
5. Press the reset switch on the microcontroller
6. Run the python read script within the IDLE shell (or dedicated python IDE)

7. Press PJ1 on the microcontroller to begin the sample collection
8. The x,y, and z distances will appear on screen.
9. Repeat until desired x distance is collected (until the end of hallway is reached)
10. Close read file and open the open3D file
11. Run the open3D file
12. Close the first window to show the final map.

Brief User Guide

1. Right click on the read file and select “EDIT WITH IDLE”
2. Connect microcontroller to computer using a micro-usb cable. The port on the MC is opposite to the ethernet port.
3. Press the reset switch on the MC
4. Run the code by selecting “RUN” > “RUN MODULE” in the top bar.
5. Press the start button on the microcontroller. This button is PJ1 which is on the right side of the microcontroller when the ethernet port is facing down.
6. Allow the stepper motor to finish its rotation
7. Move up about 300mm and follow steps 5 to 7 until the end of the hallway is reached.
8. Open open3D application using IDLE by right clicking it (like step 1)
9. Exit out of the first window to view the final sample.

Limitations

1. The max quantization error can be calculated using the equation $V/2^m$. The ToF sensor needs an input voltage of 3.3 with an 8-bit ADC, which shows the max error to be 1.28×10^2V.

2. The MC’s floating point unit supports 32 bit math operations that are integrated within the Cortex-M4 processor. It accepts floating points but can reduce accuracy when trigonometry calculations are performed. This is because it is limited to 32 bits before any casting or truncation can occur. To overcome this, the conversion can be done in python.

3. The ToF sensor uses inter-integrated circuit communication protocols with the microcontroller. The microcontroller is the master and the ToF sensor is the slave. UART is used between the microcontroller and the laptop using COM6 with a baud rate 115.2kbps.

4. The max SSC (standard serial communication) rate used in the system is 115.2kbps. To verify this, using different baud rates in realterm were tested, however, all other rates were unsuccessful.

5. The stepper motor and bus speeds limit the speed. The speed of the motor is determined using the bus speed, which was student number specific. In this case, the bus speed was 48 MHz, limiting the speed of the stepper motor. The max speed of the stepper motor is 120MHz.

Circuit Schematic

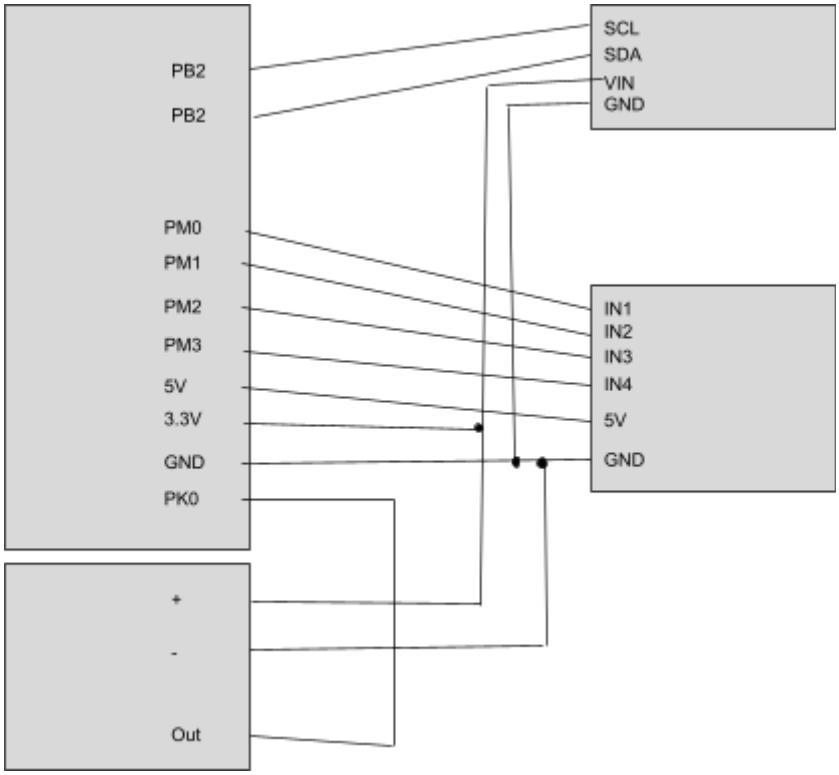


Figure 5: Circuit Schematic

Programming Logic Flowchart(s)

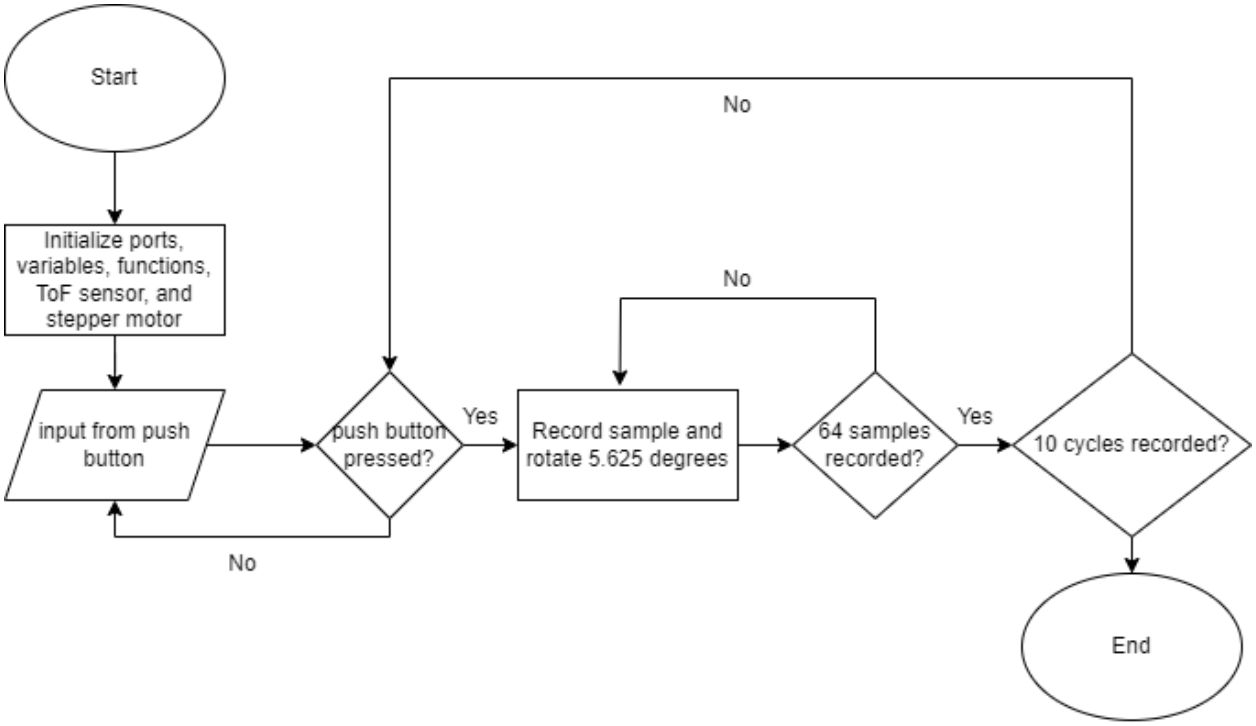


Figure 6: Flowchart of Microcontroller

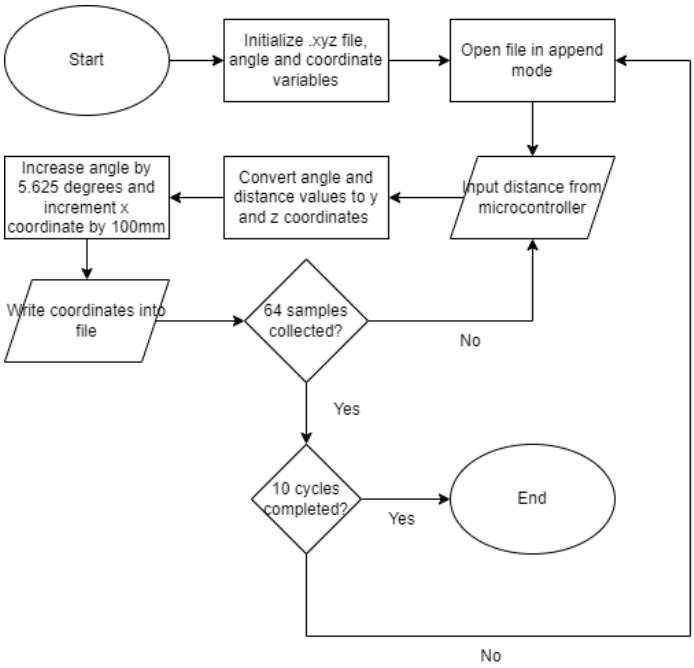


Figure 7: Flowchart of python code