2DX4: Microprocessor Systems Project

Final Project

Instructors: Dr.Bruce, Dr. Haddara, Dr. Hranilovic, Dr Shirani

Tianze Zhang – L01 – zhant22 - 400208135

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 [tianze zhang,zhant22,400208135]

Video link:

Project demo:
https://drive.google.com/file/d/1Hf4dpb1rhyuog8ImYML8uDSY08wQUnyZ/view?usp=sharing

Q1:
https://drive.google.com/file/d/1ruJF7OtLNRUqPImeiFSY1ZVPb4HsbDq9/view?usp=sharing

Q2:
https://drive.google.com/file/d/17rF8nAS-UId9gpuSQLt3HLD5zB1rgLOf/view?usp=sharing

Q3:
https://drive.google.com/file/d/1RoXVn78SuDMsa3d68mSN-pWrmmHg8tUj/view?usp=sharing

Device Overview:

Features:

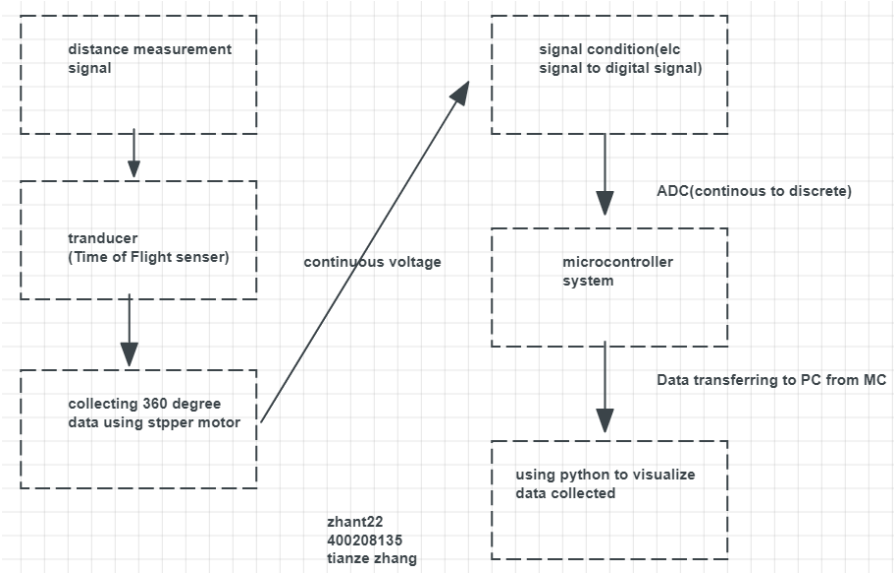
The Time of Flight distance measurement system I design has below features

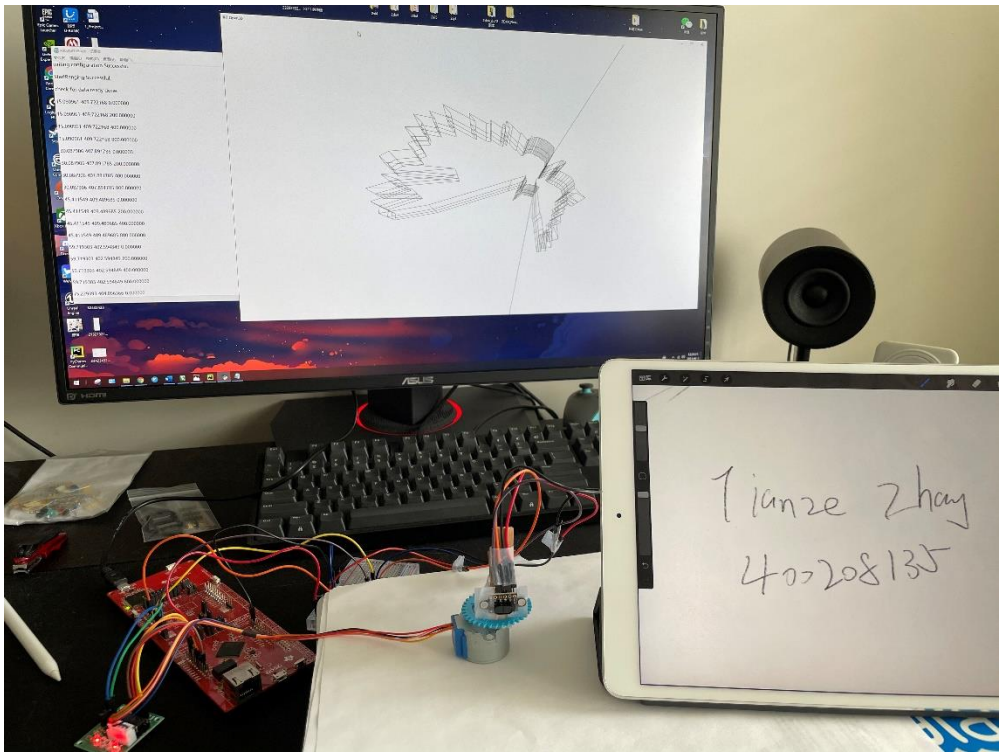
- collect the depth of distance in 360 degrees
- plot a interactive visualized graph of the depth of distance in three-dimensional.
- Bus speed : 24MHz
- Operating voltage: 5.0V
- ADC sampling rate: 48MHz
- Cost: 200 dollar estimated
- Language used: C and Python

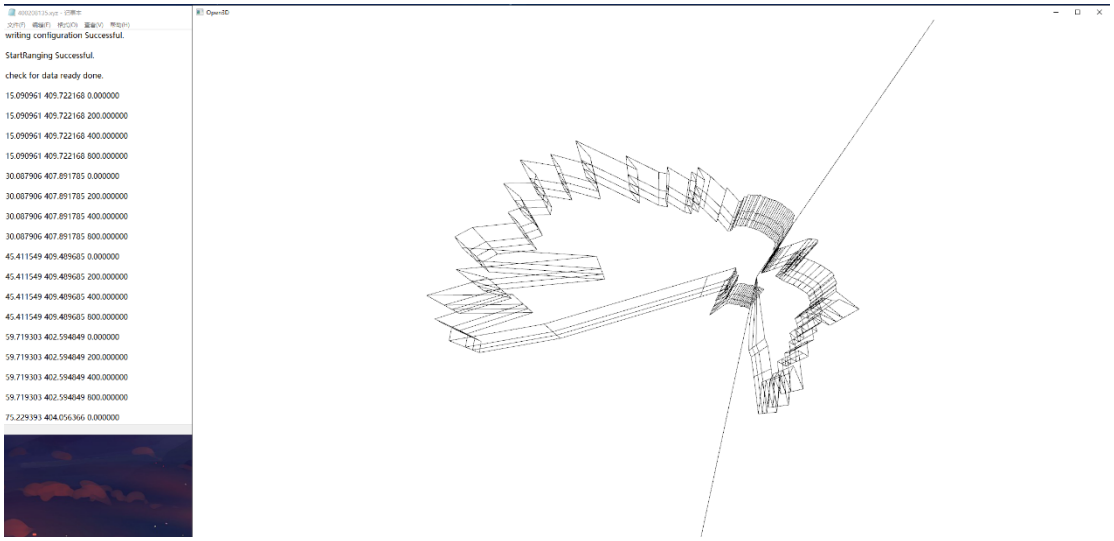
General Description:

When the start button is pressed, the interrupt start; the stepper motor connect to the micro controller will perform a 360 degree uniform speed clock-wise rotation. The time of flight sensor is attached to the stepper motor top in order to rotate 360 degree along with the stepper motor. After collecting 451 data of depth of distance information, the stepper motor and time of flight sensor will stop at exactly 360 degree from the starting point; as this time the open3d module from python libraries will process the data collected by ToF sensor and plot a interactive visualized graph of the depth of distance in three-dimensional.

Block Diagram:







Device Characteristic Table:

ToF module	ST VL53L1X with 940 nm invisible laser
Microcontroller	Texas Instruments SimpleLink™ Ethernet MSP432E401Y MCU Launchpad™
Stepper motor	28BYJ-48
Stepper motor driver board	ULN2003
Software	Python 3.7 with pyserial, numpy,and open3d/ realterm
Operating voltage	5.0V
Max ranging distance	400cm

Operating Temperature	-20°C ~ 85°C
Avg time needed for a single measurement	21s
Total measurement points	451
Bus speed	30MHz
Baud rate	115200
Max ranging frequency	50Hz
ADC sampling rate	48Hz
Serial COM port	COM 4
Pin for Stepper motor	PK0 - 3
Pin for ToF	PB2-3
Visualization	Open3d

Detailed Description:

Distance Measurement:

The Time of flight sensor measure the distance between its target and itself by calculation the time difference from the laser ToF shoot and reflection bounce back.

When ToF is measuring distance, the sensor will first shoot a invisible laser that have higher frequency than human eye can see; when the laser reach the target, there will be a reflection bounce back from the target, the sensor will then collected the bounced back reflection. Because there will be a time difference between the first laser shoot out and reflection coming back, the sensor is able to calculate the distance by the time difference by using formula $\Delta d = v*t$. In this case, because the speed of laser is the speed of light, so it is always a constant number, the distance is depends on the time only. The farther distance, the longer it takes the laser to bounce back. Figure1 shows blow:

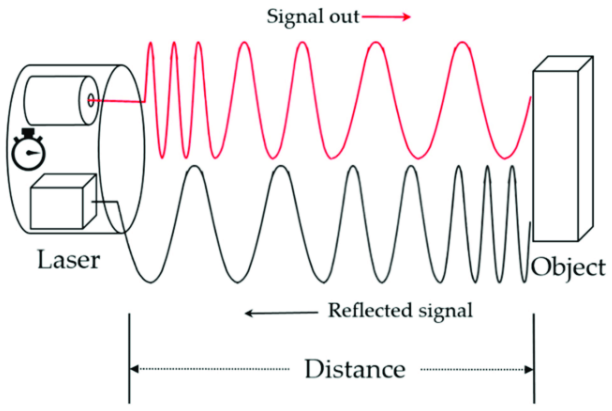


figure 1.

The maximum measuring distance VL53L1X time of flight sensor is 400cm(performance could be impacted by ambient light). The distance data collected by the sensor will be obtained through I2C bus; SCL and SDA ports on the sensor will then allows I2C bus to communicate data with microcontroller MSP432E401Y.

In order to construct a point cloud system for open3d to generate a 3d graph, the ToF sensor will do a 360 degree clock-wise rotation from the power of stepper motor(model 28BYJ-48). Open3d will generate a 3D graph from xyz format of file, so that the distance data needs to be convert into xyz coordinate. In order to do this, trigonometric is needed. From the starting position of the ToF sensor to the current position the ToF is pointing at, there is a angle between this two position; defined as angle (a). from the initial position to the current position, the stepper motor has moves n step, define the number of current step processed as (n). the total step the stepper motor needs to process is 512. Distance defined as (D), X component of the coordinate defined as (X), Y component defined as (Y). we can find the X,Y,Z component of the distance data using formula as below:

$A = (n/512) * 2\pi$

$X = D * \sin(A)$
$Y = D * \cos(A)$
$Z = 0 + (I * 200)$

Since the Z coordinate is just the height of the plane when open3d generate the 3d graph, I is defined as the number of print function for UART.

Example:

If the ToF sensor just process 180 degree, so that the current step n is 256 the distance measured is 100cm, and UART is print it to PC in first time:

$A = (256/512) * 2\pi = \pi$

$X = D * \sin(A) = 100 * \sin(\pi) = 0$
$Y = D * \cos(A) = 100 * \cos(\pi) = -100$
$Z = 0 + (I * 200) = 0+ (0*200) = 0$

So the XYZ file will has the data as:
0 -100 0

Microcontroller flowchart:

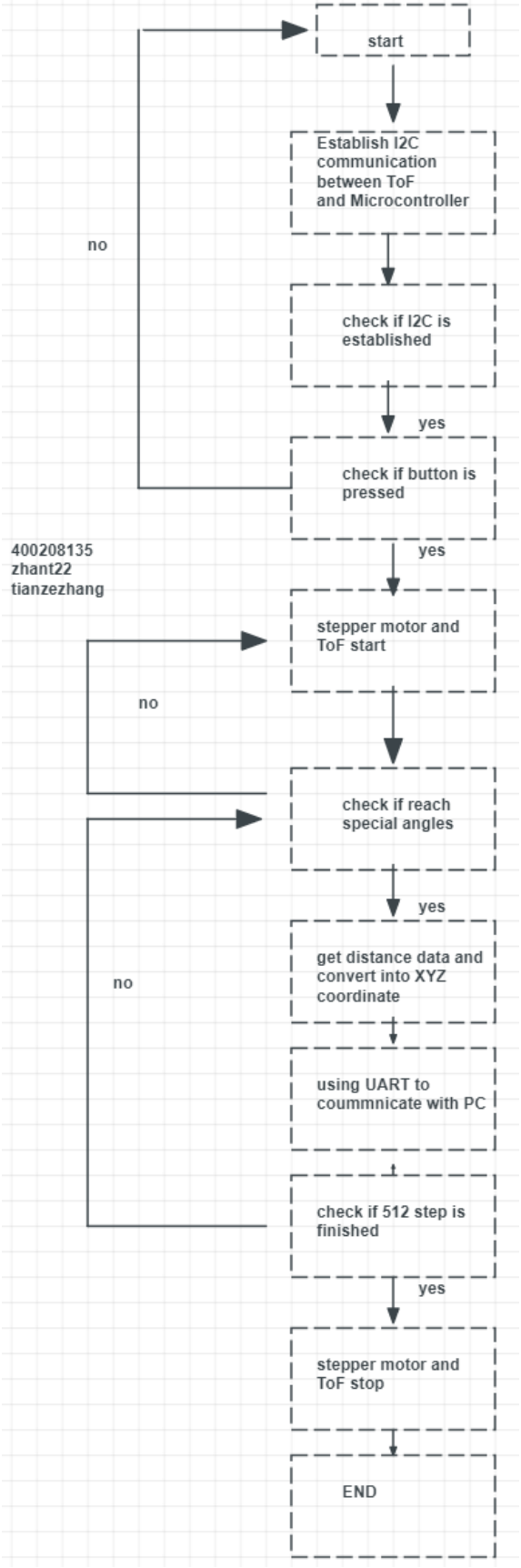


Figure2.

Visualization:

The software used to visualize the XYZ file is python 3.7 with the help of open3d from python library and keil on window 10 platform. The desktop used to run this task is equipped with nVIDIA RTX 2080S GPU with AMD 3800X CPU, 16 Gigabyte RAM.

- The python library I used is pyserial, numpy, and open3d.

- The data is store in the I2C and transferred to microcontroller via the port SCL and SDA on ToF, after that, the pyserial will read the data in COM4 and transfer the data form microcontroller to the PC.

- It visualize by using open3d to process the point cloud system. In microcontroller, the distance data collected by ToF will be translated into XYZ coordinate using formulas mentioned above, then the XYZ data will be transfer to PC, as the same time, python code running on PC will read and decode those XYZ data and print them into a XYZ format file; open3d program will then process the XYZ format file as a point cloud system. Finally, python code will connect each XYZ point will lines and call the open3d program to generate the final result.

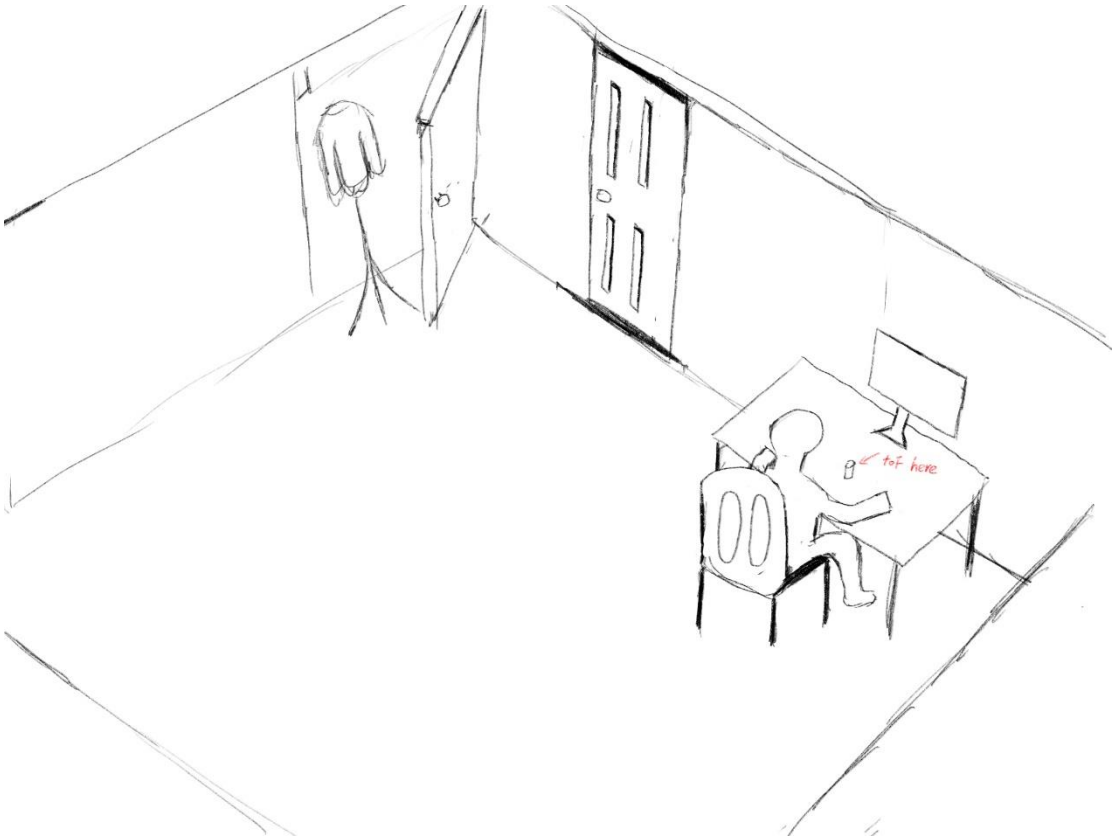


Figure3.

Application example:

In order to using the device, there are some wires that we need to connect:

1. For stepper motor, connect 1N1 to PK0, 1N2 to PK1, 1N3 to PK2, and 1N4 to PK3. The blue + wire to 5V; green _ wire to GND on microcontroller.

2. For time of flight sensor, connect Vin to 5V, GND to GND, SCL to PB2, SDA to PB3.

3. For the start button, connect 5V to a 330 ohms resistor and PE0 in series with the push button, on the other side of the push button, connect to the GND.

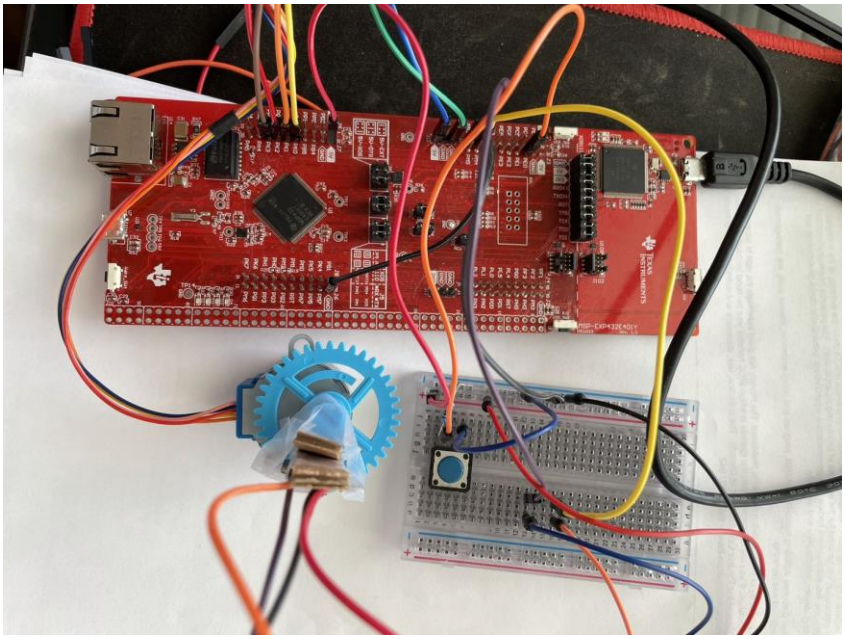


Figure4.

After connecting physical circuit, software set up is the next step:

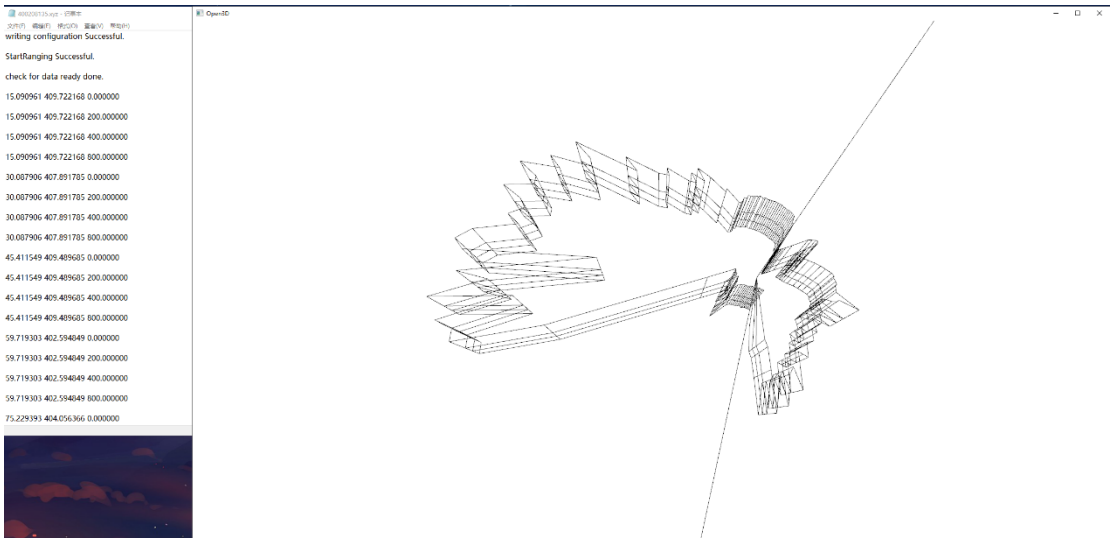
1. Open up the python file and install pyserial, numpy, and open3d from the python library using python -m pip install XXXX in comment window.

2. Check the COM port number in the device manager window, it can be found searching in windows starting menu. The baud rate needs to be set to 11520 in the python code.

3. Click run button to start running the python code, and build the keil code into the microcontroller, then press reset button on microcontroller. After the onboard LED stop blinking, press the push button; XYZ data can present in comment window, after a full rotation, open3d will plot the 3d graph.

In the design, the X and Y is defined as the horizontal plane of the axis. With ToF placed as the origin, Z axis is the height of the target. The expected result could be

like below:



The program will generate a interactive 3d graph will a XYZ format file with all the XYZ coordinate of the distance collected by time of flight sensor.

Limitation:

1. The limitation of the microcontroller floating point capability, because the value is represented in floating point, the value is not that exactly, for example, the value in decimal is 1, and in floating point it is 0.99999. in order to use the trigonometric functions; there is a angle between this two position; defined as angle (a). from the initial position to the current position, the stepper motor has moves n step, define the number of current step processed as (n). the total step the stepper motor needs to process is 512. Distance defined as (D), X component of the coordinate defined as (X), Y component defined as (Y).

2. The maximum quantization error for each of the ToF module is

3. maximum standard serial communication rate I can implement with the PC is limited by the UART hardware of the microcontroller. For Texas Instruments SimpleLink™ Ethernet MSP432E401Y MCU Launchpad™, the UART speed is limited at 11520, we can verify it by checking the data sheet of the microcontroller.

4. The communication method(s) and speed used between the microcontroller and the ToF modules is via I2C. the distance data collected by the ToF is obtained by the I2C buses, then transfer to microcontroller via SCL and SDA which connect to PB2 and PB3 respectively.

Circuit Schematic:

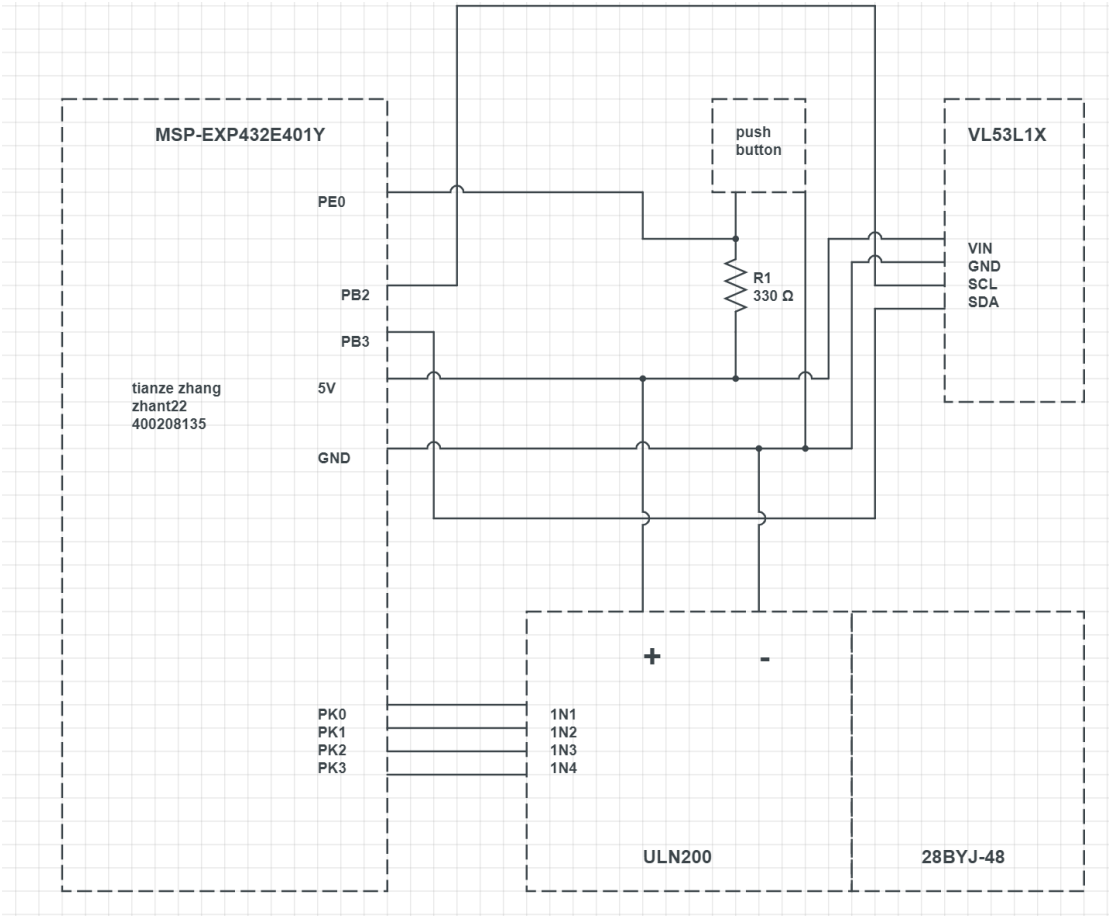
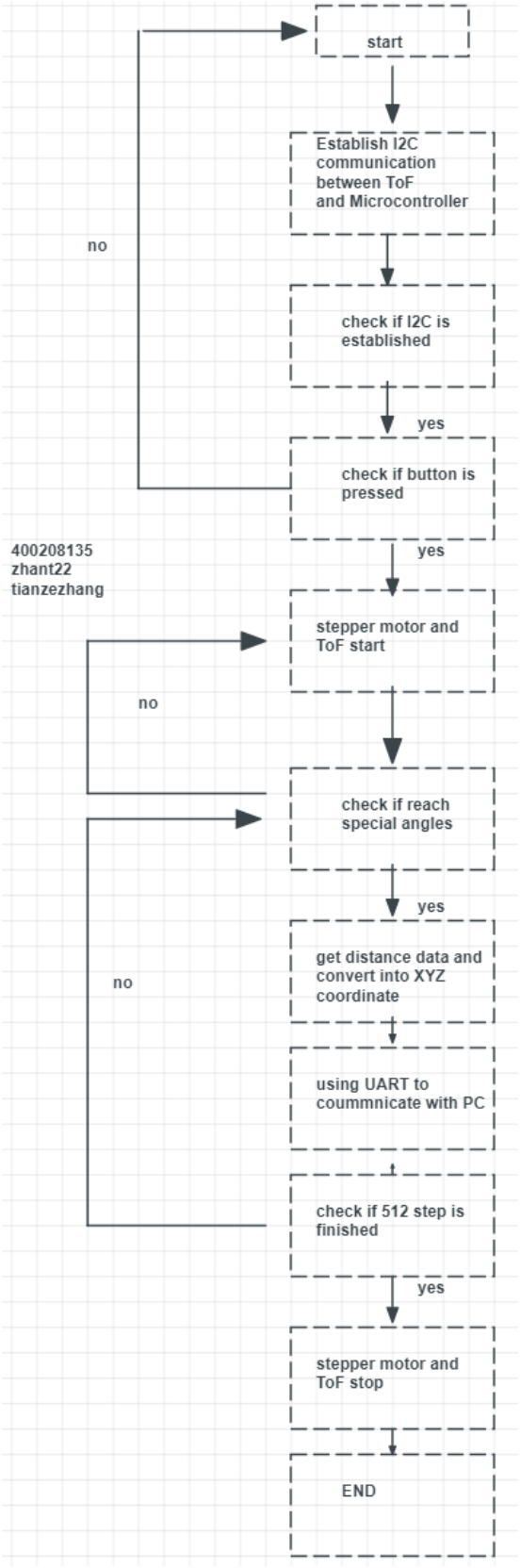


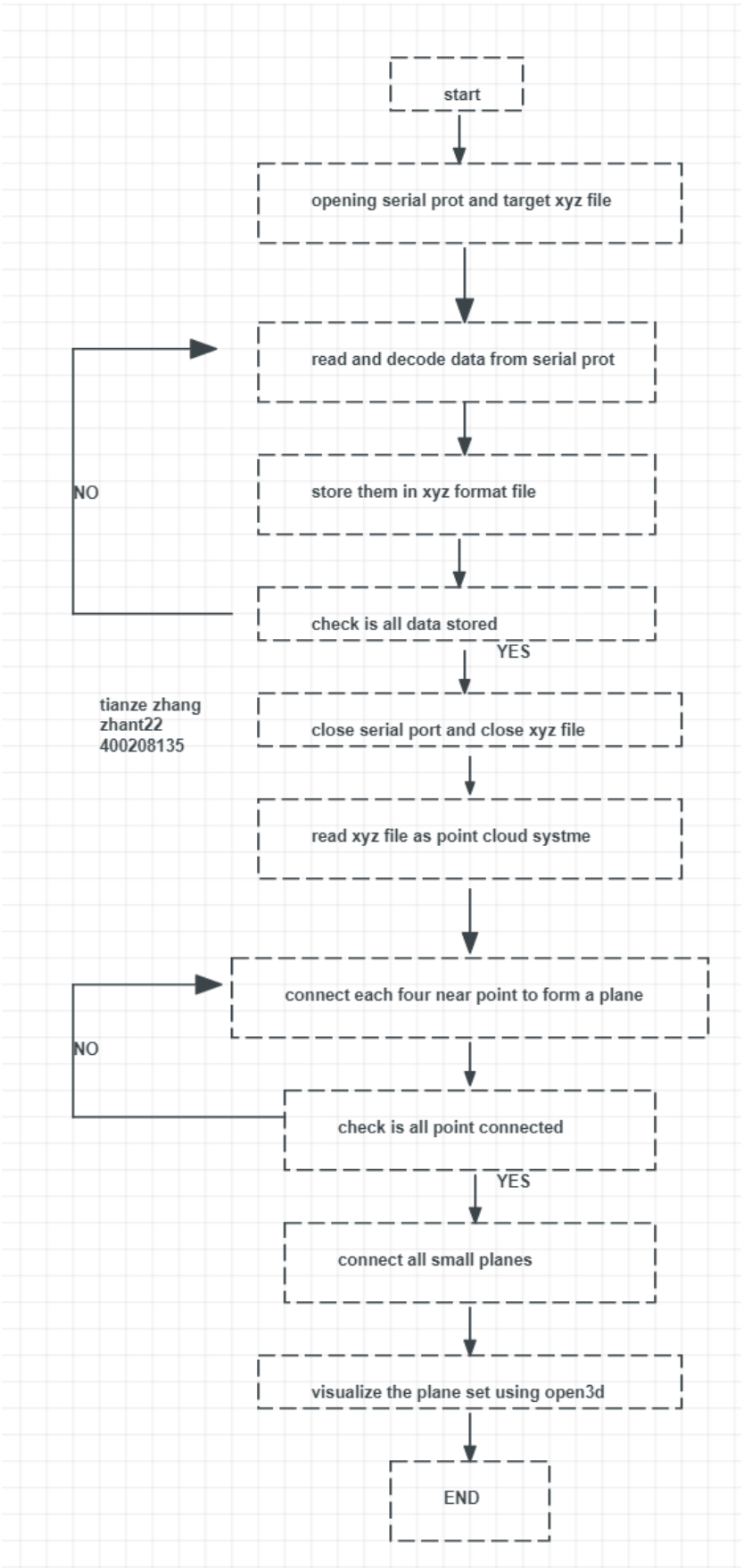
Figure5.

Logic flowcharts:

Microcontroller:



Python code:


start
opening serial prot and target xyz file
read and decode data from serial prot
store them in xyz format file
NO
check is all data stored
YES
tianze zhang
zhant22
400208135
close serial port and close xyz file
read xyz file as point cloud systme
connect each four near point to form a plane
NO
check is all point connected
YES
connect all small planes
visualize the plane set using open3d
END