McMaster University

COMPENG 2DX4

Microprocessor Systems Project

Observe, Act, Reason
2DX4 Course Project 2020

Author:
Elston Almeida

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

April 9, 2020

Contents

Video Links

Answer to Question 1:
https://drive.google.com/file/d/1mhXzmYS6zf-gbSuowlDp_ge_3_qCbLS7/view?usp=sharing

Answer to Question 2:
https://drive.google.com/file/d/1qeWXAxKIjS8tua6epDcme5S2XcgD7wgZ/view?usp=sharing

Answer to Question 3:
https://drive.google.com/file/d/1qf02WmTbPKA6HvMeRYWu7ighH9fgucOg/view?usp=sharing

Device Overview

Features

- Up to 400 cm distance measurement
- Up to 512 samples per revolution (0.703° per sample)
- Millimeter resolution for distance measurements
- Up to 27° FOV available on the distance sensor
- Open source solution for data process and visualization
- Complete 360°coverage in a plane

General Description

This is a 2D LIDAR module composed of the MSP432E401Y microcontroller, VL53L1X time of flight sensor, and a 28BYJ-48 servo driven by a ULN2003. The MSP432E401Y is a 120 MHz, 32-bit ARM Cortex M4F microcontroller capable of floating point arithmetic and uses I2C and UART to communicate with the sensors and the user[1]. The 28BYJ-48 is a 4 phase uni-polar stepper motor that provides accurate rotation data (within 5%) and each motor step is independent as to provide non-cumulative error[2]. The VL53L1X provides accurate distance measurements through utilization of a 940nm class A laser and a SPAD(Single Photon Avalanche Diode) receiver with infrared filters for the best results in typical environments[3].The user is able to connect to the 2D LIDAR module through UART over USB to run the python program getData.py. This python program will generate a point cloud from the data being recorded during each measurement. The end user can utilize their own surface construction algorithms using the point cloud data or use the the python program render.py to generate a clean 'ply' file.

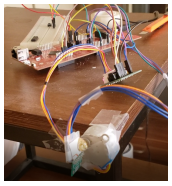


Figure 2.1: 2D LIDAR Module

Component Block Diagram

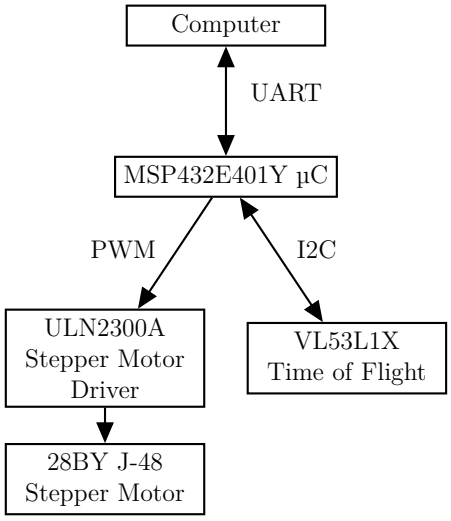


Figure 2.2: Overview of components and relations

Data Flow Graph

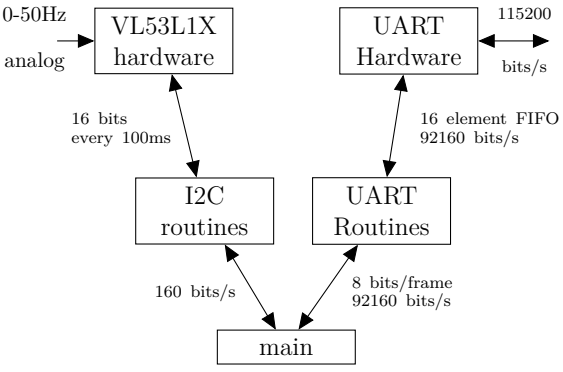


Figure 2.3: MSP432E401Y Data Flow

The MSP432E401Y communicates to the uni-polar stepper motor through PWM where GPIO outputs are momentarily set high in sequence to accurately control the position of the rotor.

Device Characteristics Table

Table 3.1: Component Power Requirements

Component	Voltage
MSP432E401Y Operating Voltage	+5V, from USB supply
ULN2003 Operating Voltage	+5V, from MSP432E401Y
VL53L1X Operating Voltage	+2.8 to +5V, on V_{IN} from MSP432E401Y

Table 3.2: Pin Descriptions

Pin Name	Signal Type	Signal Description
GPIO [M0:M3]	PWM	Output for Stepper Motor
GPIO B2	SCL	I2C Clock
GPIO B3	SDA	I2C Data
GPIO J0	Input	Sample collection control
MicroUSB (Debugger)	UART	UART output to PC for data output

Table 3.3: Communication Parameters

Parameter	Value
I2C Clock	100Kbps
I2C Address VL53L1X hline UART BUS speed	0x29 16MHz
UART Baud rate	115200(default) can be increased up to 921,600 in standard mode
UART Frame	1 start bit, 8 data bits, 1 stop bit (No parity)
UART Transmission	Half-duplex

Table 3.4: Timing Specifications

Parameter	Description	Value
MSP432E401Y BUS Speed	Microcontroller Clock Speed	96MHz
Inter-Measurement Delay	VL53L1X Time to measure a distance	100ms
Measurement Time budget	VL53L1X Time in between measurements	100ms

Table 3.5: Thermal Characteristics

Parameter	Value
MSP423E401Y Operating Temperature	-45° C to 105° C (Range set on application)
VL53L1X Operating Temperature	-25° C to 85° C
28BYJ-48 Rise in Temperature	$<$ 40K(120)

Detailed Description

Distance Measurement

The VL53L1X is the module for measuring distance. It emits a 940 nm laser that is invisible and eye-safe and then measures the time until the receiver can sense a returning photon[3]. The receiver is a single photon avalanche diode(SPAD) which is a p-n junction that is reverse biased exceeding the breakdown voltage. At the bias, the electric field is high enough such that a charge carrier that enters the depletion region creates a very large current. The depletion layer of the SPAD has an opening the size of the expected returning wavelength. When a single photon is able to hit the sensor, it creates an instantaneous surge in current[4]. From the time the beam was emitted to the time it was received, the distance can be calculated as $d = \Delta t/(2 \cdot c)$, where d is the distance being measured, Δt is the time difference, and c is the speed of light.

The 28BYJ-48 is a stepper motor that is tightly controlled as to provide accurate angle information for data points measured. The stepper motor works by energizing the coils in the correct sequence as to turn the rotor. The 28BYJ-48 provides a reduction ratio 1/64 and so it follows the step angle would be 5.625°/64 or in other words, there are 360°/(5.625°/64) = 4096 steps per revolution [5]. Since there are 4 phases and its rotation is controlled utilizing half-steps, the smallest angle between data points of a sample is 0.703°.

The distance sensor and the stepper motor provide data as to how much open space exists between the sensor and an object at a specific angle in the plane; the plane is the current sample. The data being recorded could be considered to be in polar coordinates. The first data point measured in each sample must have the sensor pointed parallel and facing away from the floor. The first point will be our y-axis and as the stepper motor turns 90°it will be the z-axis. From the coordinate system being used, for any point we have $y = d\cos(\theta)$ and $z = d\sin(\theta)$, where θ is angle given by the stepper motor state. The user will be moving according to a defined Δx per sample to complete the $\mathbb{R}^3$ coordinate system (Note: Δx must be either positive or negative depending on the right handed coordinate system).

The flowchart is for the data processing program is shown in Figure 8.1 on page 12. The parameters for data collection such as the number of data points per sample, the number of samples, the Δx, and the COM port for data collection will be input on startup of the data collection python script getData.py. As the user takes a sample, the data is sent to the PC where the point is converted into $\mathbb{R}^3$ and then stored into an 'xyz' file.

The 'xyz' file format is such that each line an x, y, and z coordinate in order, space separated and followed by a newline character. This file format can be easily read by open3D which will be used in the visualization process.

Displacement

The current implementation of displacement measurement between samples is set to a constant parameter Δx. The MPU 9250 is a 9 Degrees of Freedom (9DOF) sensor including 3 axis gyroscope, 3 axis accelerometer, and a 3 axis magnetometer. During the time between samples, the MSP432E401Y would receive data from the MPU9250 as to keep track of the 2D LIDAR sensor's displacement from the current position. Due to the inher-

ent error in such sensors, reading displacement would yield to larger errors over time therefore the outputted data must be processed through a Kalman Filter before being numerically integrated to find displacement. Each time the next sample reading is requested, the displacement is calculated and it would be added onto the old position coordinate.

Visualization

The visualization process is performed through the usage of the Open3D library. Once the xyz file is completed as outlined in the Displacement section, the user must now import the data, process the data, and produce a mesh. The flowchart for this process is shown in Figure 8.2. To import an existing point cloud in the 'xyz 'file format, we can use the function open3d.io.read_point_cloud, and this will automatically load the data points of the 'xyz' file into a 3D space as shown in Figure 4.1.

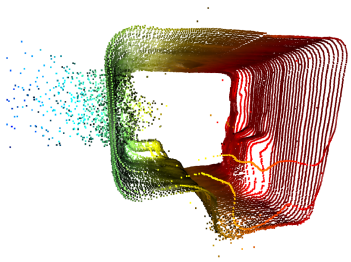


Figure 4.1: Raw point cloud data

Filtering is the next step as there is a lot of noise due to a window in this room. Filtering the data down first by utilizing vowels helps simplify the data to remove clumping of data points. The process of down sampling the data as voxels is accomplished through a method of point cloud objects called voxel_down_sample(voxel_size) which returns a cleaner point cloud. With the new point cloud outliers can be removed through statistical filtering which is another method available to point cloud objects called remove_statistical_outliers.

This method requires two parameters as follows: the number of neighbors to check around the point being considered, and the standard deviation of distance allowed for the point to exist. Running the dataset with statistical filtering returns Figure 4.2. It is clear that Figure 4.2 removes most of the noise apparent in Figure 4.1.

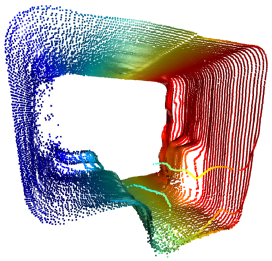


Figure 4.2: Filtered point cloud data

Creation of the mesh utilizes the ball pivoting algorithm in Open3D which outputs a triangle mesh. The ball pivoting algorithm utilizes a ball of a specific radius and runs it through the point cloud. If there are three points in contact with the ball, a face is created connecting those three points together. To utilize the ball pivoting mesh algorithm we require the normals and the average distances to get a good estimate for the radius of the ball. If the ball is too small then one too many points are left with holes, but if the ball is too large then it has predictable surface reconstruction. The output of this process is shown in Figure 4.3.

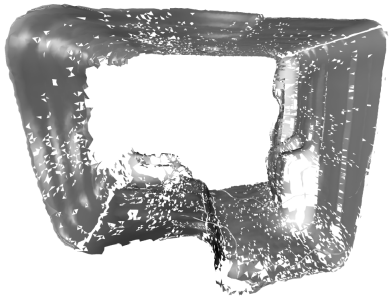


Figure 4.3: Mesh from filtered point cloud

Application Example

Note: It is assumed that Python has been setup correctly and works with the packages 'serial' and 'open3d' in the following steps. For this example we utilize Python 3.7.0 for getData.py and Python 3.6.4 for render.py. The following example will be performed with default settings: 512 data points per sample, 10 samples, and a constant Δx of 10mm.

1. Follow the application schematic digram according to the Figure 7.1 on page 11
2. Connect the microcontroller to the PC using the micro-USB port located on the XDS 110 debugger unit (Outlined in red in Figure 5.1).

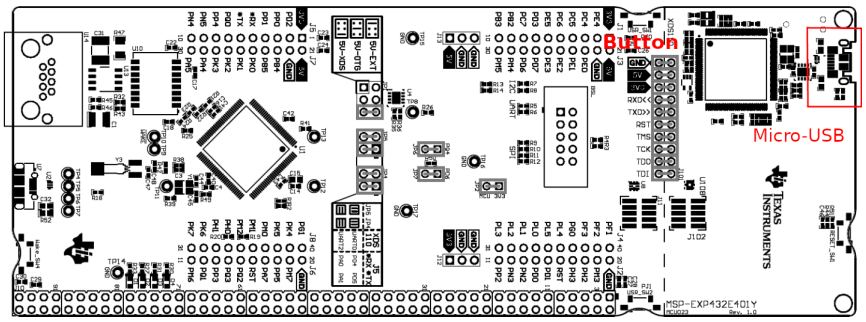


Figure 5.1: MSP432E401Y Board

3. Once the device is connected, the on-board LEDS will start blinking indicating the initialization process and they will stop once the initialization is completed.
4. After the initialization process has completed, check the COM port the PC has assigned to the device. Typically the COM port is COM5, but can also depend on other factors. The communication parameters for the device are described in Table 3.3 found on page 4.
5. Run the python file getData.py and fill in the prompts for data points per sample (integer < 512), number of samples in total (integer), the constant x change in mm (integer), and the COM port found in the previous step (integer). The python program should now run and stall with the text "Waiting for data..." as shown in Figure 5.2.

```
2DX4 - Final Project
Data Collection
Enter the # of data points per 360 degrees (up to 512): 512
Enter the # of samples in total: 5
Enter the constant change in X per sample (X is in mm): 10
Enter the COM PORT NUMBER of MSP432E401Y (eg. Enter 5 for COM5): 5
Opening: COM5
Current Settings:
deltaX: 10
Samples: 5
--------------------------
Waiting for Data...
```

Figure 5.2: getData.py waiting for data

6. Point the VL53L1X module parallel, facing away from the floor and ensure there is enough room for the module to move as the stepper motor rotates. The coordinate system of the module is outlined in the Detailed Description under Distance Measurement.

7. Press the on-board button found on the MSP432E401Y shown in Figure 5.1 to activate the data collection for the current sample. The MSP432E401Y should now start sending data and the each data point recorded will show on the terminal.

Figure 5.3: getData.py recieveing data

8. When the sample is complete, move the module in the x direction by Δx and then repeat step 7 for the number of samples desired (right handed coordinate system).
9. All the data collected will be stored between samples in a file called plotFile.xyz local to getData.py script. A sample output of the file is shown in Figure 5.4. Verify the data is correctly stored and remove/modify data points if needed.

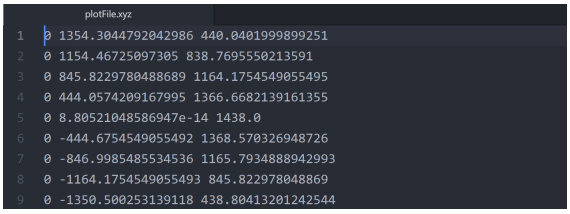


Figure 5.4: plotFile.xyz Sample Data

10. Ensure the file render.py is local to the plotFile.xyz and run python script. Upon completion, it should show a render of the mesh created. An example of the rendered mesh is shown in

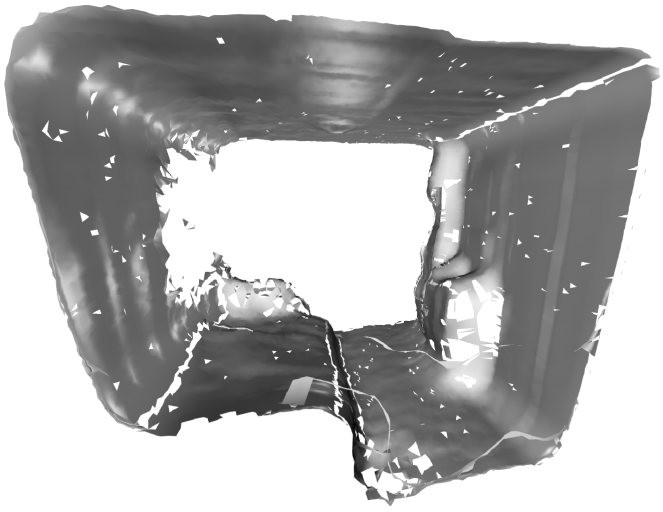


Figure 5.5: render.py mesh created

11. The user can also choose to use a different renderer or import the object elsewhere. The python script render.py produces the output of the mesh as a .ply format which can be used in other 3D software such as MeshLab or 3D builder.

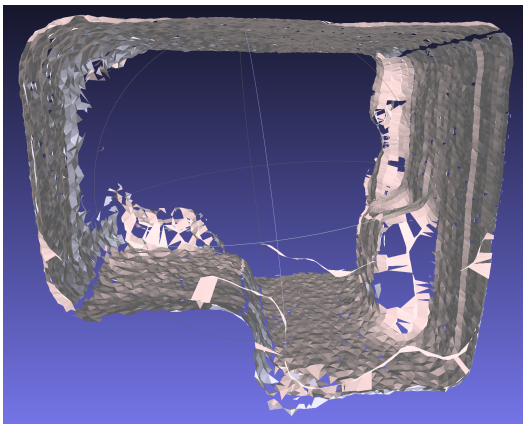


Figure 5.6: MeshLab View 1

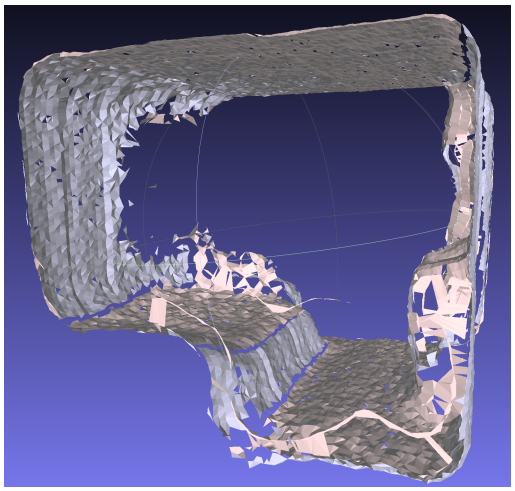


Figure 5.7: MeshLab View 2

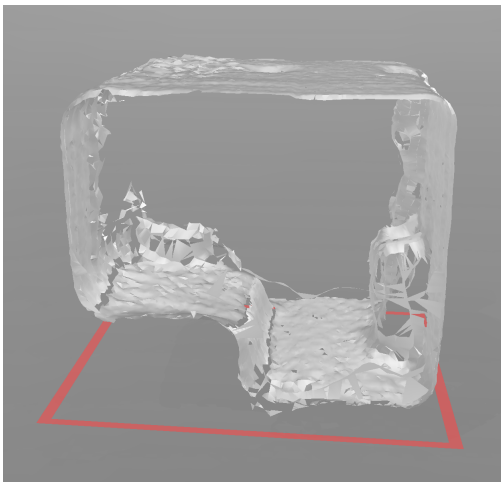


Figure 5.8: 3D Builder View

Application Note: The hole on the left side of the wall in Figure 5.7 is due to sunlight passing through a window. In this region without processing there would be a lot of noise, but the points are removed in order to produce clean, usable results.

On the right side of the wall in Figure 5.6 it is apparent there are a few holes in the wall, this was due to the immediate increase in distance from when one sample was taken in the plane with a large object near to the sensor, but the following sample did not include the large object. Due to the large distance seen by the sensor between these two samples, the algorithm used to create the mesh is unable to verify that it is not a hole.

Limitations

Stepper Motor - 28BY-J48

- Overheating issues under continuous use so the user must consider a cool down delay.
- Rated with 5% error so small steps would not be ideal for angle accuracy
- Smallest angle permitted per step is approximately 0.7 degrees so not ideal for long range sensors
- Due to the 1/64 speed reduction ratio and the high operating temperatures under continuous use, this component is the limiting factor of the speed and operational time of the device. This can be tested by timing the rotor as it moves through steps.

Distance Sensor - VL53L1X

- Unable to obtain readings on areas of reflective surfaces with high curvature due to reflections not seen by the receiver.
- Unable provide a continuous stream of distance data due to the time budget and the inter-measurement delay.
- Direct or partial sunlight (and other sources of IR) would lead to unpredictable measurements due to interference creating noise.
- The resolution is 1 millimeter and the maximum quantization error is $2.6/2^{16} = 3.97\text{e-}5$

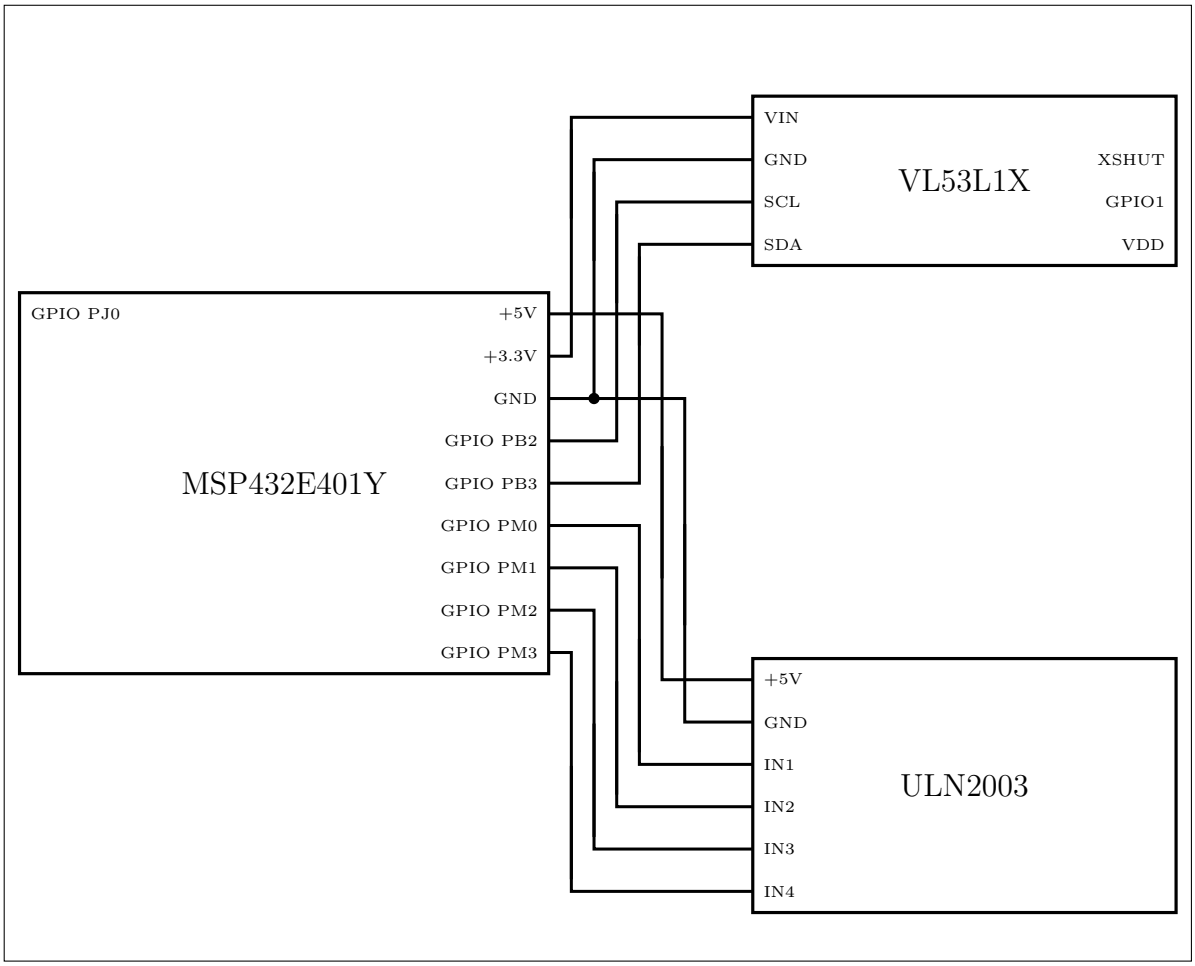
Microcontroller - MSP432E401Y

- Maximum standard serial communication rate supported is 921,600bps and with high speed mode it is able to reach 1,843,200bps for this application. The tests were performed by changing the IBRD and FBRD registers for UART0.
- The MSP432EY401Y dedicates 32 registers capable of 32 bit single precision floating point data processing based on the IEEE 754 standard. The 32 dedicated 32 bit registers can be combined to support 16 total 64 bit precision registers. On a hardware level with the FPU enabled, it supports addition, subtraction, multiplication, division, and square root[1].
- The math library is not utilized for this application, but can be implemented to support floating point numbers and trigonometric functions. The trigonometric functions found in math.h utilizes double precision arguments and returns a double precision output. Therefore the system utilizing trigonometric functions found in the math library are limited to 64 bit precision leading to error if higher precision is required.

Answer to Question 6 on Documentation - The Nyquist sampling frequency for the displacement module is double the input frequency. The Nyquist frequency should be approximately 200Hz as recommended by the sensor manufacturer[6]. If the input signal exceeds this frequency then the measurements of the signal are would be an inaccurate representation of the true signal.

Application Schematic

Figure 7.1: Wiring Diagram for 2D LIDAR Module



Application Note:
The VL53L1X sensor requires 2.6V supply to operate, but the carrier board has a voltage regulator on V_{IN} that accepts 2.6V to 5V as supply. The voltage regular may be bypassed by providing 2.6V to V_{DD} instead. If V_{IN} is used as the source voltage, then V_{DD} will be able to supply +2.8V from the output voltage of the regulator to other devices. Do **NOT** connect V_{DD} and V_{IN} at the same time. The XSHUT pin is the active-low shutdown input that turns the sensor off when not in use. GPIO1 is utilized for interrupts to the microcontroller. The carrier board has in-built pull up resistors for XSHUT and GPIO1 so they do not need to be connected when not in use[7].

GPIO PJ0 is the on-board button that internally connected to a pull up resistor. The UNL2003 connects to the 28BYJ-48 stepper motor through the on-board 5 wire connection(not visualized)

Programming Logic Flowchart

Data Extraction

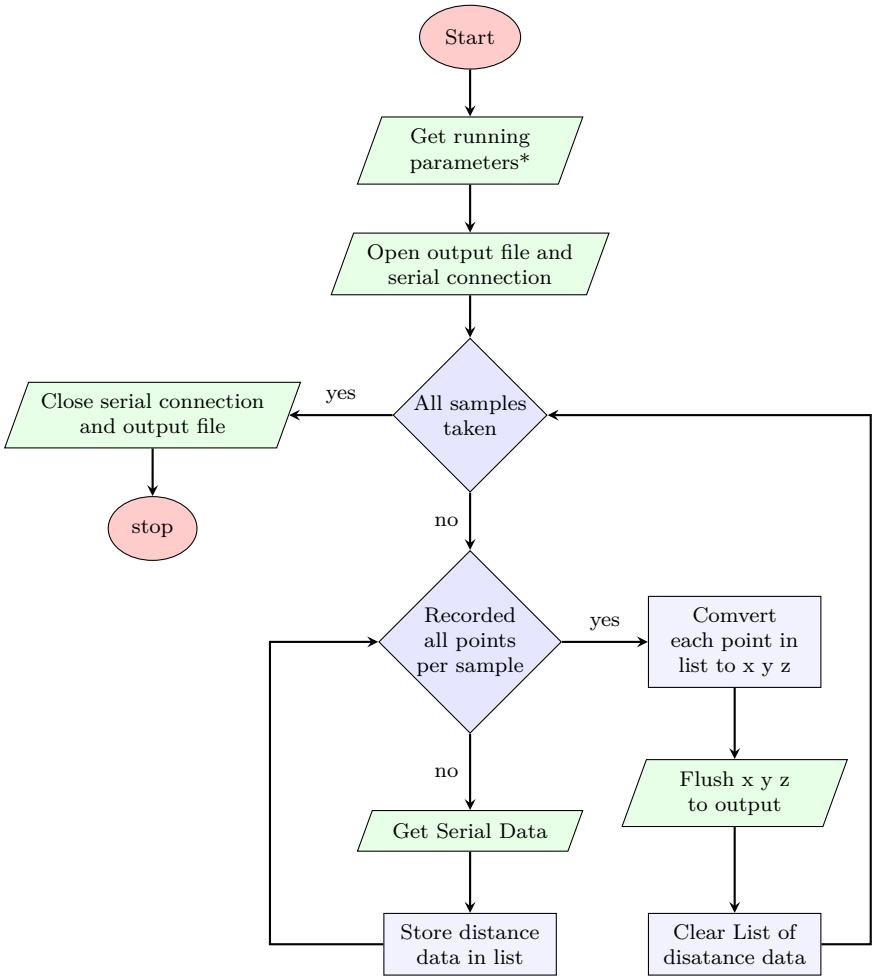


Figure 8.1: Programming logic flowchart for the data collection

* = running parameters include the number of sample total, number of points per sample, constant Δx between samples, and COM port

Mesh Generation

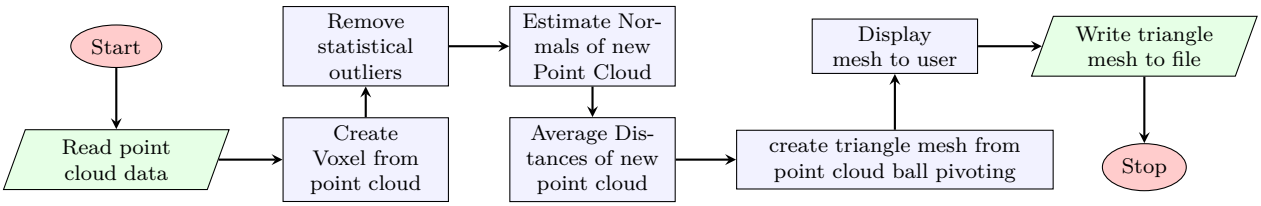


Figure 8.2: Programming logic flowchart for the mesh creation

MSP432E401Y Data Acquisition

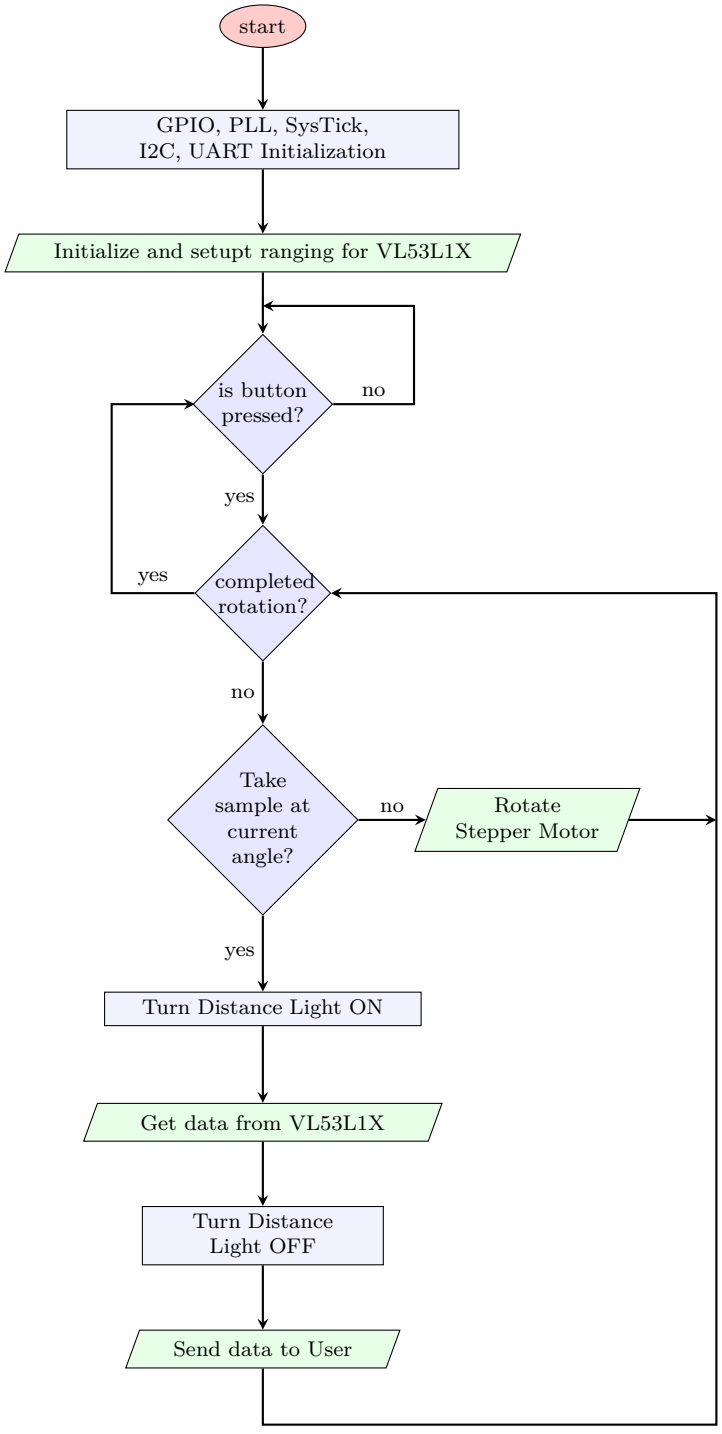


Figure 8.3: Programming logic flowchart of data aquisition

Bibliography

[1] 2017, *Msp432e401y simplelink™ ethernet microcontroller*, MSP432E401Y, Texas Instruments. [Online]. Available: https://www.ti.com/lit/ds/slasen5/slasen5.pdf.

[2] *Stepper motor 5v 4-phase 5-wire & uln2003 driver board for arduino.* [Online]. Available: http://www.geeetech.com/wiki/index.php/Stepper_Motor_5V_4-Phase_5-Wire_%26_ULN2003_Driver_Board_for_Arduino.

[3] 2018, *Vl53l1x - a new generation, long distance ranging time-of-flight sensor based on st's flightsense™ technology*, VL53L1X, STMicroelectronics. [Online]. Available: https://www.st.com/resource/en/datasheet/vl53l1x.pdf.

[4] S. Cova, M. Ghioni, A. L. Lacaita, C. Samori, and F. Zappa, "Avalanche photodiodes and quenching circuits for single-photon detection.," *Applied optics*, vol. 35 12, pp. 1956–76, 1996.

[5] *28byj-48 datasheet*, 28BYJ-48, MikroElektronika. [Online]. Available: https://download.mikroe.com/documents/datasheets/step-motor-5v-28byj48-datasheet.pdf.

[6] 2014, *Mpu-9250 product specification revision 1.0*, MPU-9250. [Online]. Available: https://cdn.sparkfun.com/assets/learn_tutorials/5/5/0/MPU9250REV1.0.pdf.

[7] *Pololu - vl53l1x time-of-flight distance sensor carrier with voltage regulator, 400cm max.* [Online]. Available: https://www.pololu.com/product/3415.