Project Proposal

Adrik shamlonian

Big Picture

Mapping systems are being used in mobile robots in order to alter their localization and path planning features. In this research, we are trying to come up with an inexpensive and compact hardware which can be used in mapping systems. The main hardware component used in this system is an IR time of flight VL53L1X sensor which is mountable on small mobile robots such as quadrotors. One of the usages of this sensor is in autonomous rescue robots for single distance range sensing. However, in this research project, we are trying to use the sensor as a LIDAR for mapping purposes. It can be used in a way to help the robot to estimate the obstacles and the objects' shape and their locations with respect to itself. Using the captured information in form of a point cloud can help the robot to map the environment and as a result, plan its path using the generated mapping data. The general approach to solve the image capturing problem is by writing functions for the VL53L1X's API (Application Programming Interface) to get access to its 256 SPAD (Single Photon Avalanche Diode) values and generate a 16×16 array of ranges. The ranges will help us to capture the shape and the location of the object/s located in front of the sensor in the distances of 40mm to 4m. The data generated from the VL53L1X sensor can be used to create a point cloud of the objects located in front of it and be used in mapping algorithms. Generating a point cloud using this sensor can lead to inexpensive mapping and navigating devices.

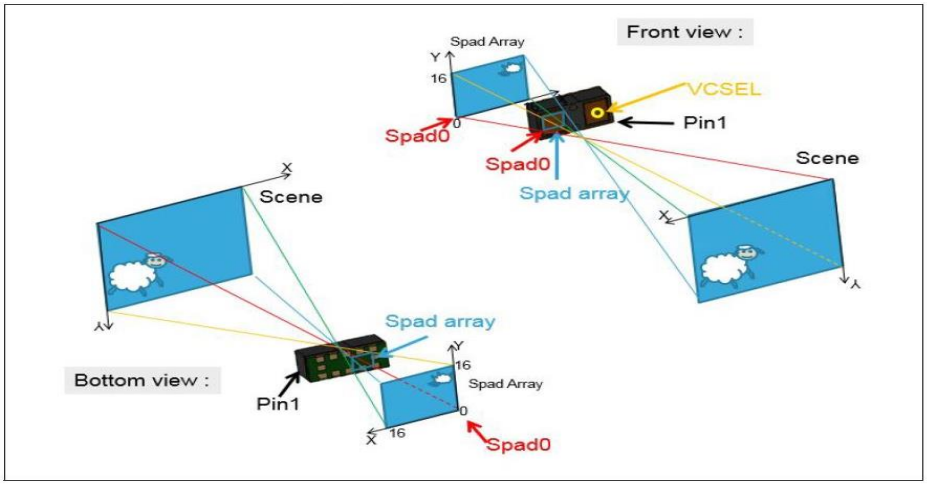

Figure 1 Estimation of Location of objects based on the selected ROI[1]

[1] VL53L1X API_User_Manual_UM2356_rev2

Specific project scope

The VL53L1X sensor is primarily used for detecting the distances an object using a single distance range sensing method. However, this sensor uses the average value of its SPAD's located in an ROI (Region of Interest) selected by the user to measure the distance of the object located at 4mm to 4m of it. The problem is that the sensor's API only allows changing its ROI from 16×16 to 4×4 SPADs while our goal is to write a function to have access to its each individual SPAD values (1×1 ROI). If we get access to each SPAD value, we will be able to generate a 16×16 3D point cloud of the objects in front of the sensor which is mounted on the robot. Therefore, by saving and stitching those point clouds using mapping algorithms we will be able to generate a map of the environment while the robot is navigating around and as a result, find the right path which leads to its goal's position. The approach is to get access to the register values of each individual SPAD through the I^2C data communication line (SDA) between the sensor and the microcontroller. The data received can be processed, stored and/or published using a ROS node. Therefore, this sensory system can be used as a modular component in robotic systems. Once we are able to receive a valid 16×16 matrix of individual distances corresponding to the objects located within the 4mm to 4m ranges of the sensor we will be able to claim that we have accomplished our goal. The solution can be used as a part of any navigation system or merely for visualization proposes to give an estimated shape of the objects located in front of the sensor.

Background / related work / references

The foundation and fundamentals needed to be known in order to understand and solve this problem are C programming, Data analysis, I^2C, familiarity with LIDAR technologies. Similar technologies have been developed or are under development by companies such as TeraRangerEvo 64px [2], REAL3™ image sensor IRS238xC [3], Structure Core[4], etc. but majority require high computational power for image processing due to the high-resolution image they generate and can create a heavy reliance on an external computational power and might not be a suitable choice for our small and low weight/cost autonomous systems. For this particular sensor (VL53L1X) no attempts have been done so far based on our researches to access the individual SPAD values for mapping purposes.

---

[2] https://www.terabee.com/shop/3d-tof-cameras/teraranger-evo-64px/

[3] https://www.infineon.com/cms/en/product/sensor/3d-image-sensor-real3/

[4] https://structure.io/structure-core

Goals / Deliverables / Tasks

Week 2

Goals:

1. Finish reading the data sheets of the VL53L1X and fully understand its architecture
2. Research other possible technologies and compare their pros and cons and costs. (come up with a table)
3. How much doable/feasible is changing the ROI of the VL53L1X sensor and can it be used as an on-board lidar on the Clazyflie and only rely on its own computational power.

Deliverables:

1. The Flow deck v2 and VL53I1X are compatible with the Crazyflie drone and can be accessed through low level I2C
2. Compared three main types of indoor navigation systems
3. The VL53L1X integrates a SPAD receiving array of 16×16 with adjustable ROI with minimum array of 4×4

Tasks:

1. Install and run the VL53L1X software and debug any issues
2. Check the on-board processing power of the Crazyflie
3. Check the compatibility of SLAM algorithms

Week 3

Goals:

1. Finish learning I2C
2. Run the sensor either on an Arduino or STM32 and get some readings with different ROIs
3. Check the implementations of the ROI functions

Deliverables:

1. Finished reading the documentations of both hardware and API
2. Timing budget (two consecutive readings)
3. Ranging functions initializations
4. Calibration
5. Change in the coordinates of FoV in the sense of SPAD

Tasks:

1. Get readings from the sensor using an Arduino or STM32
2. Interpret the readings

Week 4

Goals:

1. Find the register addresses that will give me access to the SPAD values (are not included in the documentation file)
2. Test theVL53L1x using Arduino Uno
3. Fully understand the API's functions and work with them
4. Start implementing my own functions that will read the SPAD values through I2C

Deliverables

1. Found some similar technologies with higher resolution but require more computational power
2. Some of the examples are: TeraRanger, Evo 64px , REAL3™ image sensor IRS238xC, Structure Core

Week 5

Goals:

1. Write a code which takes measurements using 4×4 ROI on the sensor
2. Understand the variables used in "void VL53L1X::updateDSS()" function and other related functions
3. Do more research about mapping algorithms using multiple LIDARs

Deliverables

1. Connected the VL53L1X to the Arduino Uno
2. Plotted the single distance receiving from the sensor with 50 measurement time
3. The data gets noisy as the object moves further away

Week 6

Goals:

1. Create a map of the system (block diagram)
2. Do 13*13 data points next time
3. Plot and visualize data using a proper visualization software
4. Dig into the ST's libraries to eventually do 256 data points!
5. Improve the update rate

6. Why is the effective SPAD 207 instead of 256?

Deliverables

1. Wrote Python and Arduino scripts to generate and visualize 16 data points
2. Read and used the Arduino ST API this time to change the ROI values
3. Lots of error checking and 16 measurements each time which causes the code to slow down
4. If we get access to all SPADs it will become faster
5. Any proper python visualization tools?

Week 7

Goals:

1. Create a map of the system both for software and hardware (a block diagram)
2. Should debug the visualization code and do demoes with different RIOs
3. LookformoreI2C implementations within the library and add them to the script

Deliverables

1. Measured a single distance by only using I2C protocol within the main script
2. Were able to extract a single range value from the data packets received from the sensor
3. The sensor takes care of the measurements based on the prior setups and sends them through I2C
4. Were able to tweak the ROI values to 1×1 in the API and removed the 4×4 restriction with no errors!

Week 8

Goals:

1. Create a map of the functions in form of a block diagram and find the path which leads to the register values related to the ROI settings of the sensor

Deliverables:

1. Wrote Python scripts for data visualization
2. Searched for methods to create inheritance block diagrams which can lead us to functions that are implementing the I2C register values of the sensor

Week 9

Goals:

1. Going to use the register functions and structs and read and print the register data directly on Arduino
2. Also check what values do global_config_spad_enables_rtn return

Deliverables:

1. Converted the Arduino ST library functions, classes and structs into block diagrams using Doxygen + Graphviz
2. Visualizedtheclasseshierarchy and found out about how the functions are connected
3. Found the register functions and the structs which store the register values

Summer

Week 1
Announced the Summer plan

Summer Goals:

1. Get all the individual SPAD values which correspond to the distance of the objects located in front of the sensor.
2. Write a custom designed code to generate a 16×16 array of datapoints captured from the sensor in a stationary position
3. Mount the LIDAR sensor on the Crazyfiledrone and get point cloud data for different positions of the drone and visualize them on ROS Rviz or any other visualization tool.

Summer Deliverables:

1. Generate a point cloud of 256(16*16) points all at once in 50ms (20Hz).
2. A 3D point cloud of the objects that the drone has flied around them generated in ROS Rviz or other visualization software.

Week 2

Goals:

1. Finish up the list that is created in week one (Comment and understand the content of the registers)
2. Read resources which explain the terminologies regarding the characteristic of SPADs
3. Write an Arduino code that will go over each internal register address and read its values and print them using the serial port.
4. Write a python script to read the serial data and save the data on a file.

Deliverables

1. A classified list containing expected values which each point to a specific 8-bit internal register and their 16-bit indexes

2. All the internal registers with indexes from 0X0000 to 0x0FFF are taken by a variable. (total of 4096 registers)
3. 467 registers have name but are not included in any of the structs (Not classified)

Week 3

Goals:

1. Revise thetable1ifit needs any corrections
2. Find a more efficient way to extract the I2C commands from the code and put them in the table 1 format
3. Finish up the table and start implementing it in an actual data communication way between the microcontroller and the LIDAR sensor only using I2C commands

Week 4

Goals:

1. Write my own code which does the same behavior but only using I2C commands no other library used
2. Compare the generated results from my code with the code that uses the ST's library

Deliverables

1. Completed the single-distance I2C communication table between the VL53L1X sensor and Arduino