Open Source LIDAR

Critical Design Review

Light Seekers

Gaurav Bhalla

Yile Chen

Aamhish Rao

Paul Roy

Department of Computer Science

Texas A&M University

03/29/2022

Table of Contents

1 Introduction (1 page; 5 points)

Generally a Critical Design Review is meant to define the transition from design to implementation. Due to the nature of our specific project, being heavy on the integration of the project, the general progress we gained during this period is in clarification and progress of our understanding of what we are going to do going forward. After creating a list of parts to order and identifying different sellers we have been waiting for these parts to arrive. In the meantime we made several advances in our understanding of the software components, testing procedures and in several other key parts of the open source repository.

A lot of the effort was focused on generally trying to prepare ourselves for the integration process to not have any major issues or surprises. For instance we examined testing procedures for displaying the data created from the open source lidar repository. This was done by using the hector slam software to create a mapping of the area around it. We have become somewhat familiar with the ROSS software and hector slam as well.

Another area where we have been working on was trying to understand the codebase as a whole. We believe it is in the interest of ourselves to get a feel for how the various files work in unison to be able to later modify the code to fit our own purposes. In order to do this we went ahead and annotated each file and have discussed it to better go forward.

Finally we have made several changes to our plan of constructing the hardware going forward. Through our conversations with the owner of the code repository, as well as other individuals familiar with the open source project we have gotten a fuller understanding of how we will construct the system.

Another major portion which we have been working on was dealing with supply chain issues. Due to external factors some parts may take longer to arrive than initially anticipated. Because of this we have begun to research and discuss several contingency plans to try and anticipate any parts that we may not receive.

1.1 Overview

LIDAR stands for Light imaging detection and ranging. The technology uses varying light waves and the times it takes for reflected light to come back to the sender to create an image of the surrounding areas. There are various use cases for this technology; creating high resolution maps, navigating spaces for self-driving vehicles and even measuring the dimensions of large objects.

The most common use case for LIDAR is creating large high resolution maps of large space. One instance of this is using bathymetric systems to map out sea floors. This is done by shooting two different rays which map both the ocean surface and also piercing the water to map out the seafloor. Using various wavelengths of light we can map out agricultural fields in order to find specifics about the field; including insect activity, estimated yields and weed control. This is especially useful because we are able to take metrics of large spaces and directly measure soil metrics without having to only sample subspaces and also to not damage specific landforms. LIDAR can also be extended to the atmosphere and determine information about cloud cover, humidity, ice crystals and so on.

Another common use case for LIDAR is for autonomous vehicles. For instance, LIDAR sensors can be used in tandem with the conventional camera sensors in order to detect obstacles during driving. This can be especially effective as the use of two sensors covers the “blind” spots of either individually. This method of using both 2d and 3d imaging together has been proven to work better than when only the 3d score is used.

Finally the technology can be used for measuring or mapping large objects. This can be used to measure and study older growth forests and has several future applications in preservation of older forestry. Another use case is in archeology. LIDAR can be used to create digital maps of archeological sites revealing measurements that other models such as aerial photography cannot measure.

LIDAR is generally measured from a variety of crafts. This can be a sensor strapped to a drone in order to measure large swaths of land, or a set of sensors on a moving car. Another example is using a stationary scanner to create point clouds of the area around. This can be paired with conventional cameras to create a full mapping of surroundings.

1.2 Needs statement

We are given a repository with a schematic for how to create the LIDAR scanner. Our first goal is to recreate this system first resulting in a working model of the open source LIDAR. The research group uses a MEMS mirror which is difficult to use. Yet the MEMS mirror is useful as it can move in a manner which creates low and high sampling areas. We will try to follow the MEMS high scanning and low scanning regions through having a non-uniform sampling rate LIDAR product that mimics these patterns. Moreover, after this, we would aim to improve the sampling speed and the scanning range.

1.3 Goal and objectives

The primary goal of this project is to work off of an open source repository to create a working LIDAR scanner and to use the data collected to map out the surroundings. Our secondary goal is to improve upon the working model on sampling and access. In terms of sampling we aim to mimic the MEMS mirror using a rotating mirror of some sort. This will create a non-uniform sampling of the environment. On the other hand we also aim to better access. This can be done by creating a reconfigurable system; in which we can easily swap out the MEMS mirror for a spinning mirror.

1.4 Design constraints and feasibility

We have around four to five months to finish this project. The largest bottleneck that we can foresee at this point is the one brought on by the need to ship for parts from various suppliers due to the supply chain being backed up. This can further contribute to issues as if we have an error in the parts we order we will need to reship before we can begin work again. Financially we have allocated a budget of 700 dollars, which we can adjust if necessary. We don’t foresee needing to update this budget.

1.5 Validation & Testing Procedures

Validation is a very important part of our process in order to make sure that our sensors are properly functioning. Our validation approach will be to check the performance of our system against another LIDAR sensor. We will let two LIDAR systems measure the same room and cross check the dimensions created. Another potential way to validate our system is to directly measure an object and validate the LIDAR system using the base truth physical model. This way of validation is best for our initial prototype rather than whatever we create later on. We can directly measure the length of a wall and the crosslist against the measurements the LIDAR sensor decided.

2 Proposed design (5-10 pages; 60 points)

2.1 Updates to the proposal design

Due to a COVID outbreak in Shenzhen, China, where the PCB manufacturing company, PCBway is, the OpenTOFLidar PCB assembly has been delayed. Since it may not be possible to get the PCB on time for the testing and integration of the hardware and the software, our advisor Mr. Di has recommended that we purchase a SF30 LightWare LIDAR to replace the OpenTOFLidar PCB. Mr. Di recommended the SF30 LIDAR as his research group has used it in the past for trying to work with the MEMS mirror. However, due to the commercial privacy of the SF30 LIDAR, there are no PCB layout files or documentation available for this LIDAR. Therefore, if the OpenTOFLidar PCB does not arrive, our group will have to use the SF30 instead of the PCB with the motorized mirror spinning platform. If the OpenTOFLidar PCB does arrive, we will use the SF30 LIDAR for comparing results with the PCB.

To use the SF30 LIDAR, our group would need to 3D print something to properly place the lenses we got for the PCB laser and diode onto the SF30 photodiode and laser due to the different laser/photodiode sizes. Moreover, we would need to allow communication with the TDC and the microcontroller through SPI to mimic the MEMS mirror non-uniform sampling regions through our motorized spinning mirror platform. Mr. Di has advised that we should attempt to understand the hardware of the PCB through the schematic. Specifically, we would need to try to figure out how to change the settings on the microcontroller and TDC for non-uniform sampling.

2.2 System description (1-2 pages; 30 points)

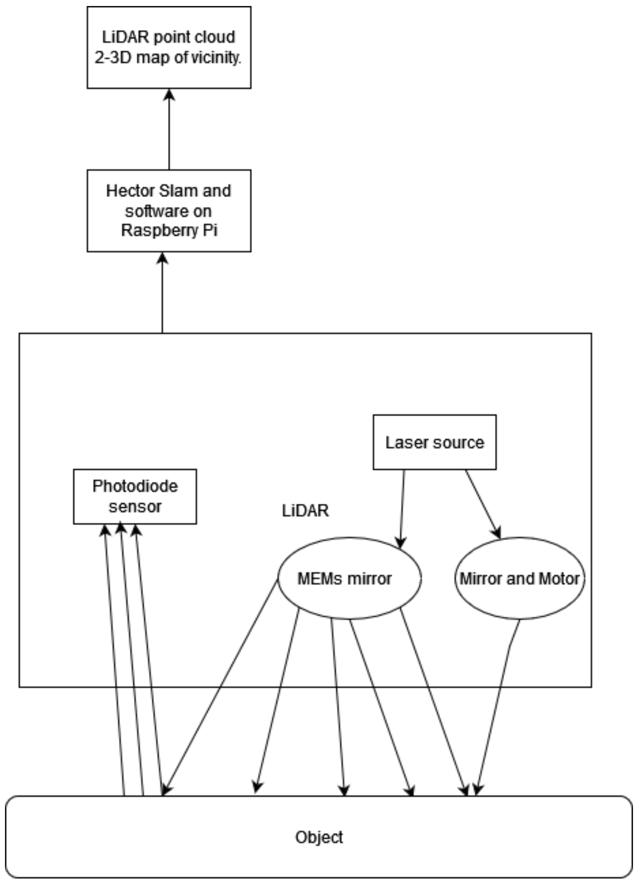
The following is a list of components that are on the block diagram below:

LIDAR Point Cloud: The LIDAR Point Cloud would essentially be the final product of our project, this would be a point cloud that would show the objects surrounding our LIDAR system and could then be used for systems such as autonomous driving or other autonomous motion.

HectorSlam/ROS Software: This layer would take the data from the LIDAR system and convert it to the LIDAR Point Cloud mentioned above using the HectorSlam software.

Laser Source: The laser source is the main component of the LIDAR and is used to track the object.

MEMs Mirror: The MEMs Mirror is an alternate to the traditional mirror and motor method of LIDAR and is electric and has the motor built into the mirror.


LiDAR point cloud 2-3D map of vicinity.
Hector Slam and software on Raspberry Pi
Laser source
Photodiode sensor
LiDAR
MEMs mirror
Mirror and Motor
Object

Mirror and Motor: The Mirror and motor are used to move the module around so that the laser can capture all of the objects in the environment. The motor will continuously be moving around so that new objects will be detected in real time with little to no blind spots.

Object: The objects we are trying to track in the environment. The system will come together as one in order to track the objects in the environment.

Photodiode sensor: The photodiode sensor is used to detect the light being reflected from the laser upon contact with the objects that are being detected in the environment.

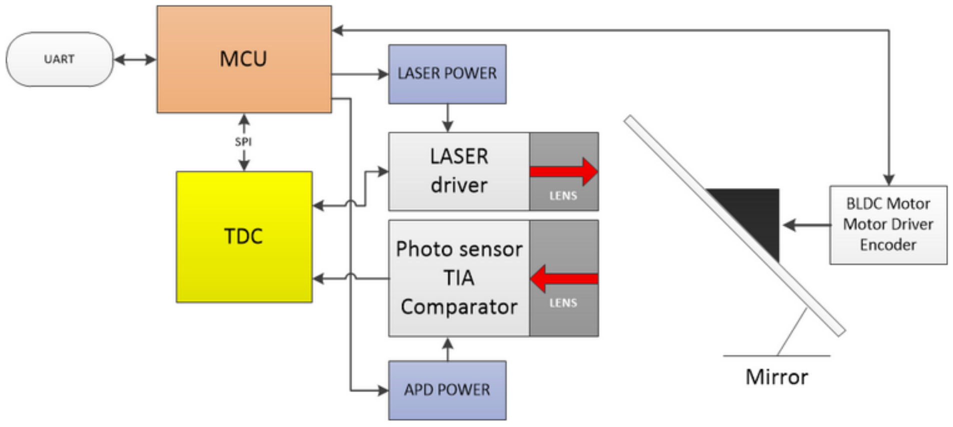
The electronic components such as sensors and power components will be soldered together using a service, while the physical components will be 3D printed and constructed together. We also plan on using a Raspberry Pi and have the LIDAR module that we are constructing to be wired to that. The software will all be running on the Raspberry Pi and we will have the data readable on a screen or through a laptop with VNC enabled to see the Raspberry Pi display.

We have two alternate methods for the mirror and motor aspect to the project, we can either use a MEMs mirror which would essentially have the two components together or use a traditional motor and have a 3d printed mount which would mount the mirror to the motor and spin around. We will be using both methods and testing out which one would be better.

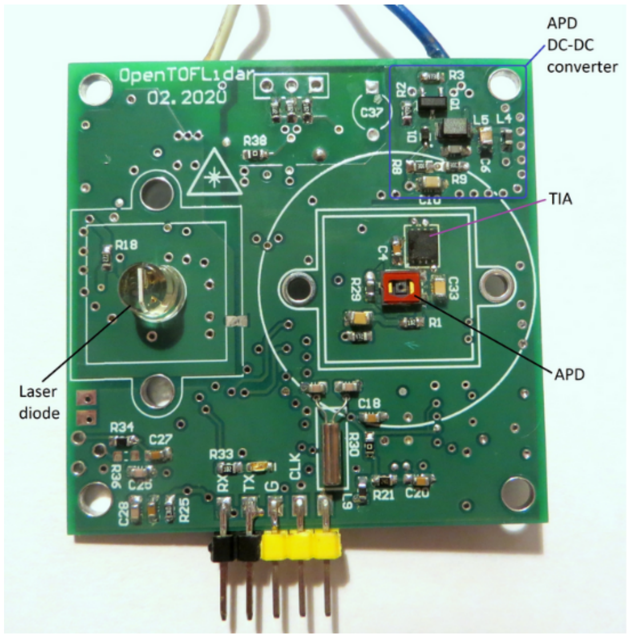
The overall system will have a fair amount of room for flexibility as we can add components as needed to the LIDAR module and additional motors if needed to the 3D printed design. The only issue is the processing time for getting components and having them assembled through the service as that is done in China and so due to this we are not planning on making many alterations upon the initial send off.

2.3 Complete module-wise specifications

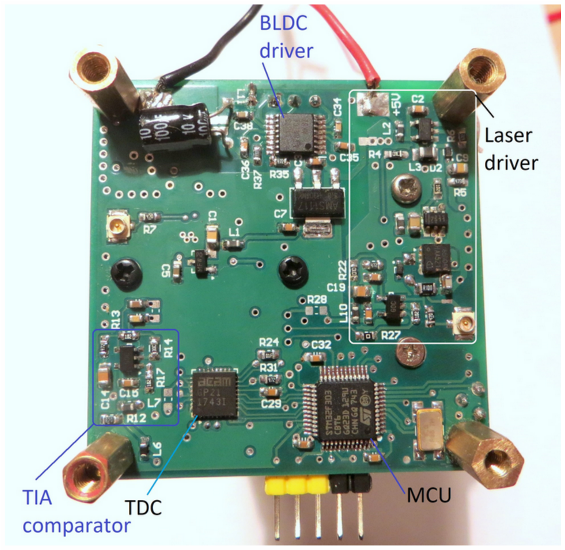
- **Circuit and logic diagrams**



Shown above is the low-level design for the OpenTOFLidar system [1]. The MCU communicates with and controls the laser and the photodiode. Specifically, the code executed on the MCU specifies the input voltage for the laser and the photodiode. Moreover the code that will run on the MCU also controls the motorized spinning mirror platform through the BLDC motor driver encoder chip. The MCU samples the distance (Time-of-Flight) data from the TDC through SPI through polling. However, the method to read data from the TDC can be improved through using interrupts instead of polling.

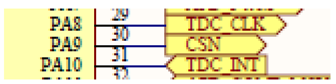


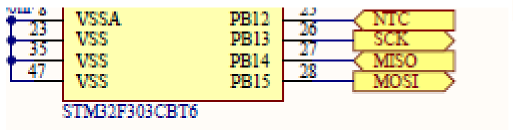
Shown above, is the OpenTOFLidar Author's annotated image of the top part of the PCB [1]. The main components on the top part of the PCB are the TIA, the DC-DC converter, the APD, and the laser diode. The transimpedance amplifier (TIA) amplifies the current received from the photodiode into a voltage.
Since this PCB is a rangefinder, the TIA accommodates a large signal bandwidth to amplify the signal produced from the APD. The Avalanche photodiode (APD) is a special photodiode component that has higher gain and signal-to-noise ratio which allows for greater sensitivity of the diode compared to normal PN diodes which is very useful for LIDAR. APDs require a larger reverse bias voltage for operation to occur compared to normal diodes. Therefore, a DC-DC converter was used to form the high enough reverse bias voltage to use the APD. The laser diode produces a 905 nm wavelength of light but requires a large current of about 30A .The lens for the laser diode focuses the light produced by the laser diode allowing for stronger concentration of the laser beam, allowing the 905 nm wavelength light produced from the laser to reach farther.

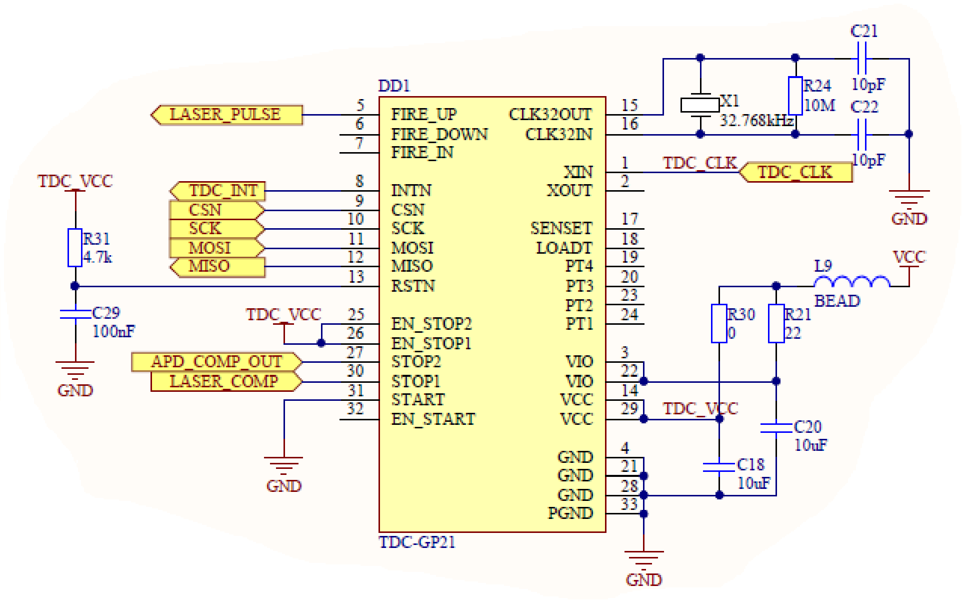


Shown above, is the OpenTOFLidar Author's annotated image of the bottom part of the PCB [1]. The main components on the bottom part of the PCB are the MCU, the laser driver, the TDC, the TIA comparator and the BLDC motor driver. The brushless direct current (BLDC) motor driver receives a PWM signal through UART from the MCU code which controls the motor speed. The laser driver creates a high energy pulsed signal for the laser diode to output. The TIA comparator distinguishes the signal from the noise received by the APD through the reflected light. The Time-to-digital-converter (TDC) measures the time it takes to receive the reflected 905 nm light (TOF) from the start of one of the laser's pulses, allowing for the MCU to accurately sample distance values to objects in the surrounding area. As explained with the low level block diagram, the MCU is responsible for controlling and communicating with all the main components mounted on the PCB. The code executed on the MCU generates the pulse for the laser, reads in data from the TDC through polling to sample distance values, controls the motor speed, controls the power provided to the laser diode and the APD, and receives/sends the distance values to the Orange Pi PC through UART.

- **Interfaces and pin-outs**

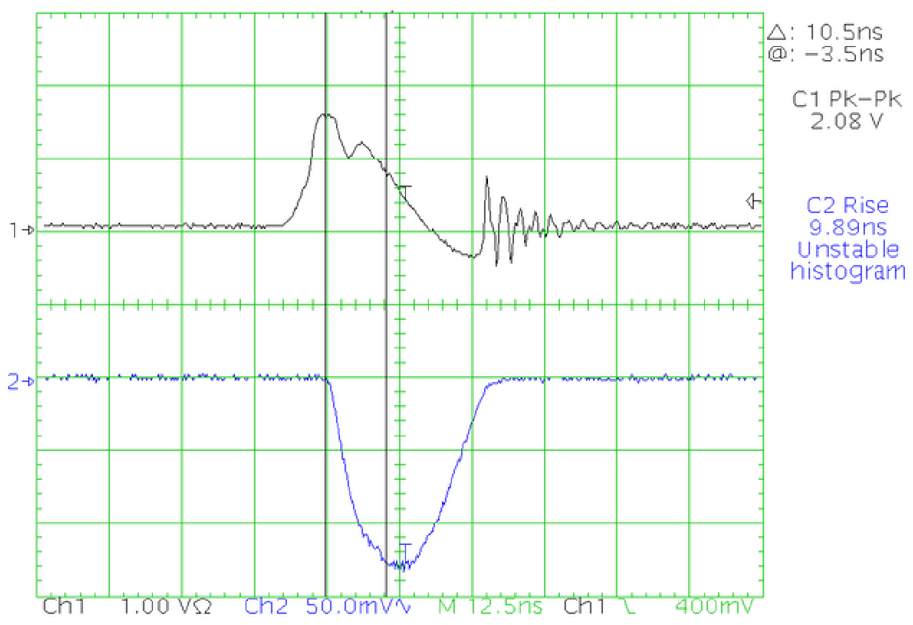






A Serial Peripheral Interface (SPI) is used to communicate between the TDC and the MCU through polling. SPI is used for short distance communication between modules in an Embedded system. In this case, SPI is used for communication between the TDC (slave) and the MCU (master). In the parts of the schematic shown above, the MCU pins used in the SPI communication and TDC chip pins are shown through their corresponding schematic diagram. There is only one master and slave chip in this system. The master (MCU) pinout is shown on the 2 pictures on the top left and the slave (TDC) pinout is shown on the right. There is the TDC_CLK pin for the clock signal, the Chip Select Not pin (CSN) which is active when the signal is low to activate the SPI connection between the TDC and the MCU; Master in slave out (MISO) which is the channel for sending data from the TDC (slave) to the MCU (master); and Master out slave in (MOSI) which is the channel used for sending data from the MCU to the TDC for configuration purposes. Outside of SPI communication with the TDC, STOP2 - STOP1 is calculated via an ALU to determine the time of flight. From there, after multiplying by the speed of light, distance measurements are recorded and sent to the ALU.

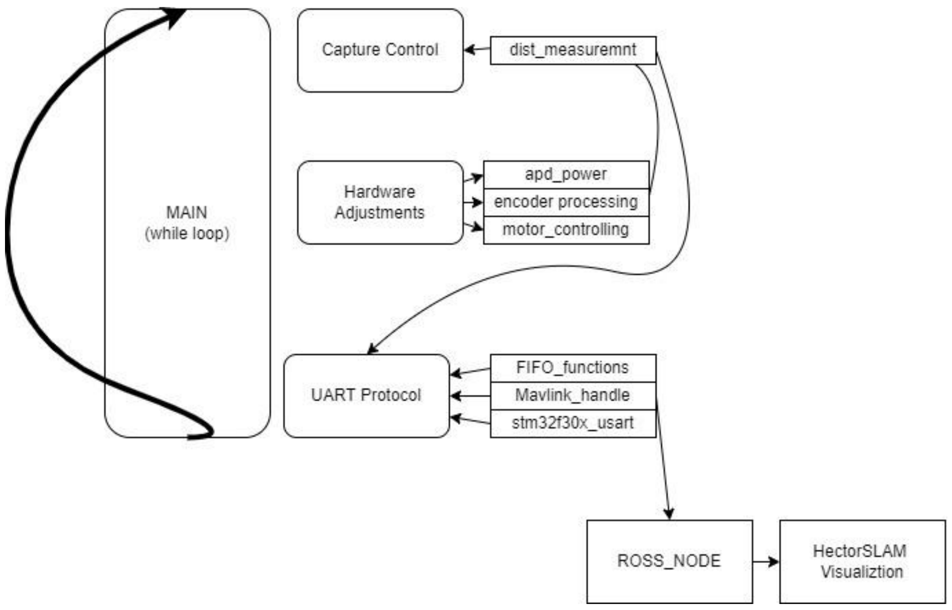
- **Timing diagrams and waveforms**



In the image above, channel 1 is the laser pulse, and channel 2 is the signal received by the Avalanche Photodiode (APD). The signal in channel 2 is inverted because APDs are reverse biased [2]. By subtracting STOP1, which is the time the laser pulse is emitted by the laser diode from STOP2, which is the rising edge of the APD signal, the Time-of-Flight (TOF) is calculated. Then, the distance measurements to objects in the surrounding area is calculated and sent to the MCU by SPI.

- **Software processes with their inputs and outputs**

Software dependency chart



Main.C
The main file purely consists of a infinite loop which on preset increments of 1ms, 10ms, 100ms, perform certain actions on a loop. For the 1ms data processing and sending is done. For the 100ms power is adjusted for the photodiode and encoder. Finally for the 100ms loop the laser and adp voltage are controlled. In addition the distance measurement is handled.

Capture_Controlling
This file initializes some functions that are related to the encoder, distances measurement and TDC. It keeps track of the encoder periods and cycles, and captures timer and encoder interrupt. It contains a function that is called periodically to capture data from the lidar, which is distance measurement during one encoder period, and stored inside some data buffers.

Dist_measurement.C
This file performs distance measurement and calculation from the lidar data, returns real distance in millimeters. There is code to communicate with the microcontroller by sending data through mavlink. It also contains code to calibrate the distance measurement.

Motor.C

The motor classes effectively just use the cycles and addresses outlined in the specification sheet and control it accordingly. The motor class also has a P loop that it uses to control the speed with feedback from the encoder.

Encoder.C

The encoder class effectively just gets data from the encoder using the cycles and addresses outlined in the specification sheet and gets the position accordingly. The encoder classes interact directly with the motor classes as that is how the motor classes get feedback for the P loops.

APD

The APD class initializes direct memory access to capture light data from ADC through fluctuation in voltage to configure APD. Moreover, it sets the input voltage for the photodiode to be appropriate so that it can function properly.Moreover, the APD class initializes the ADCs in the photodiode to capture light data voltage digitally.

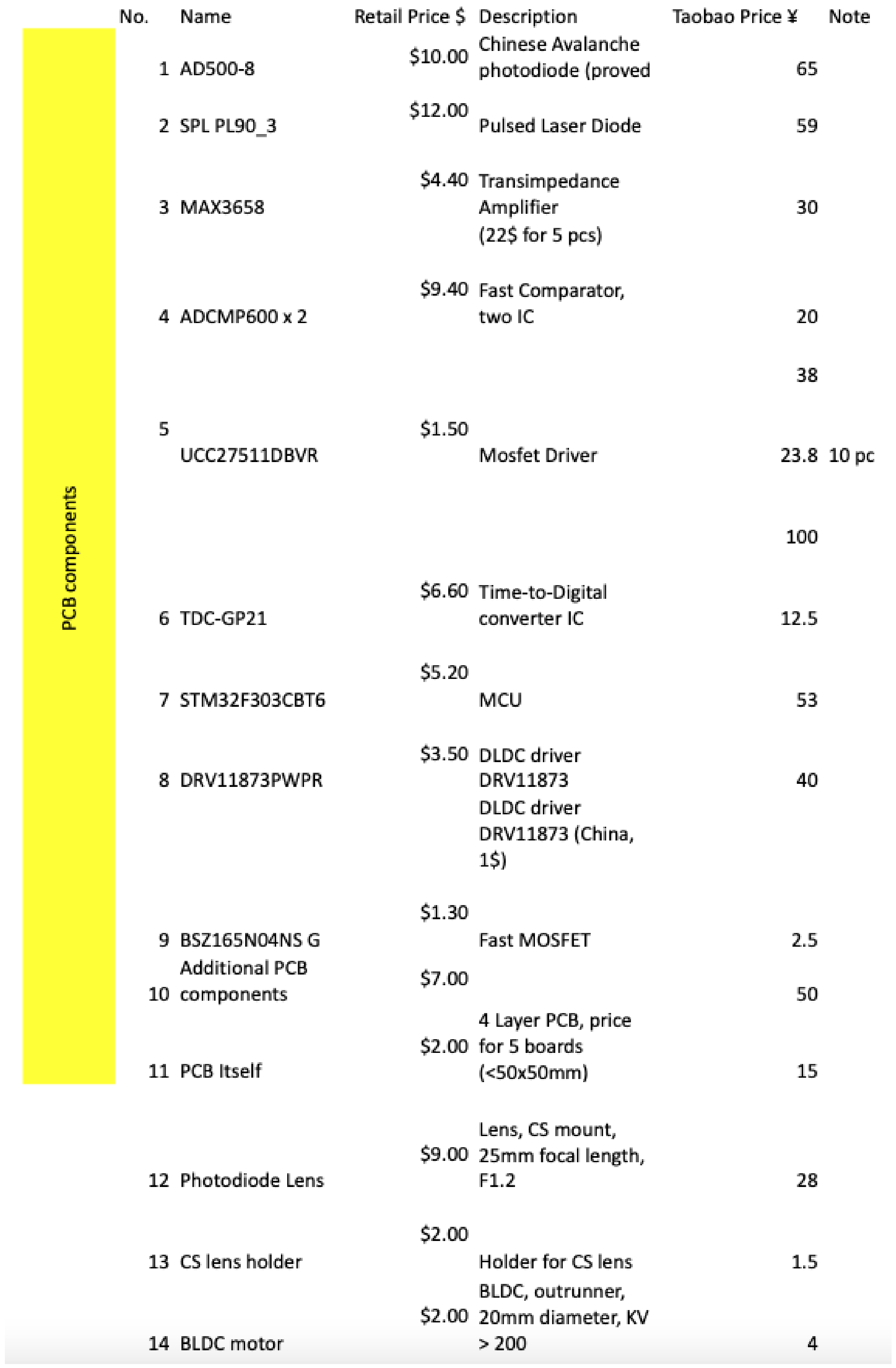
Hardware.C

As the name implies, the hardware class initializes the laser driver, the TDC, the TIA comparator, and several data watchpoint and trace unit (dwt) functions. The laser driver is used to set an input voltage given to the laser diode. A Digital-to-Analog (DAC) converter is initialized to set the 10V analog input signal for the laser diode. The DWT functions allow for delaying by several milliseconds or microseconds and is used for code to generate the PWM input signals for the TDC and other hardware components on the OpenTOFLidar PCB.

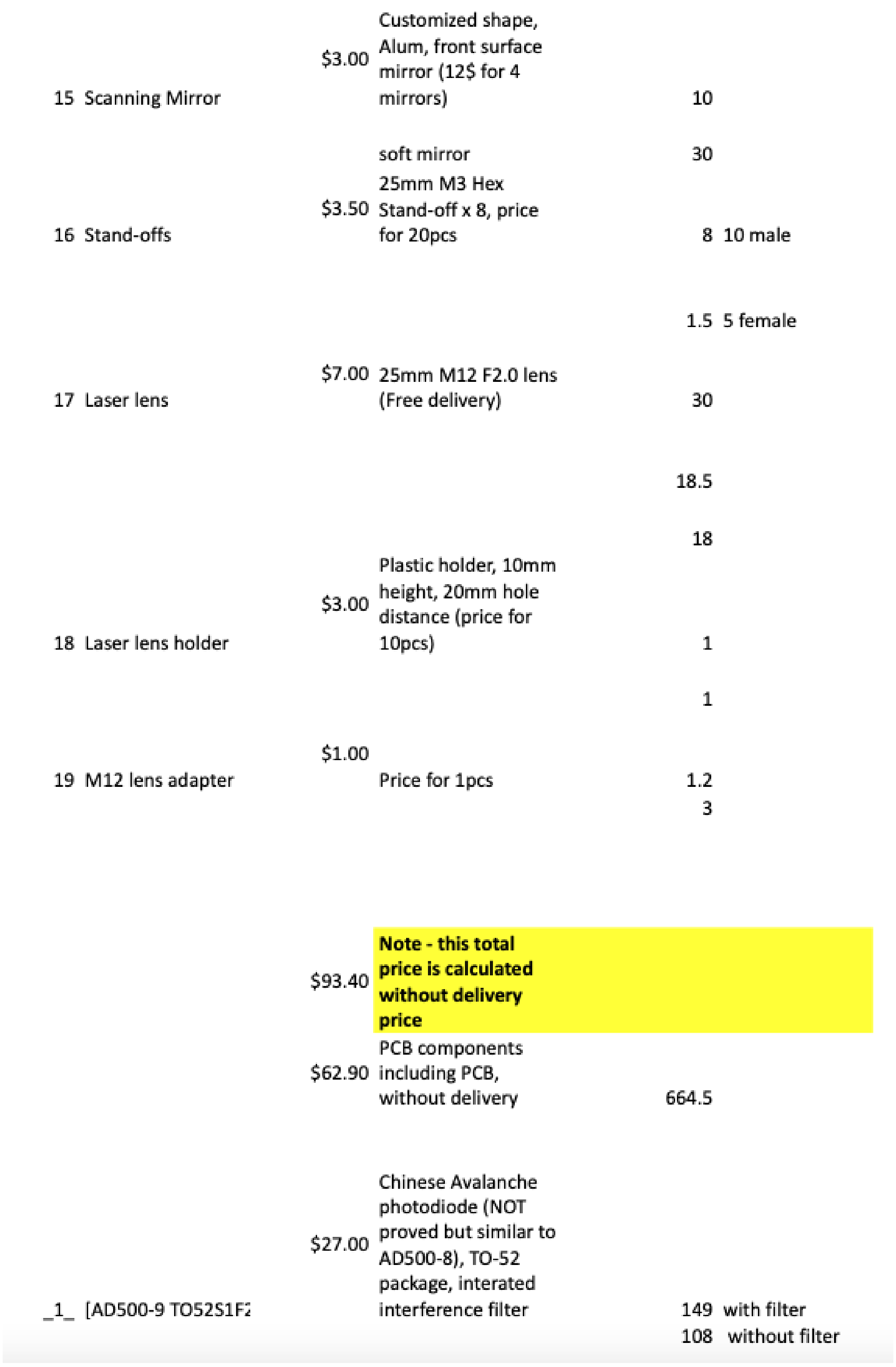
Uart_driver.C

The main purpose of the UART program is to send and receive bytes in a custom FIFO buffer between the MCU and the Orange pi PC through UART. The UART.c file itself does not have too many lines. Specifically, there are functions to initialize the UART protocol between the MCU and the send and receive bytes. Since the UART class is used as a basis for communication between the MCU and the orange pi, it is very important. It is important for the distance data to be sent to the Orange Pi PC for creating the 2d point cloud map of the surrounding area.

- **Complete parts list**



	No.	Name	Retail Price $	Description	Taobao Price ¥	Note
PCB components	1	AD500-8	$10.00	Chinese Avalanche photodiode (proved	65	
	2	SPL PL90_3	$12.00	Pulsed Laser Diode	59	
	3	MAX3658	$4.40	Transimpedance Amplifier (22$ for 5 pcs)	30	
	4	ADCMP600 x 2	$9.40	Fast Comparator, two IC	20	
					38	
	5	UCC27511DBVR	$1.50	Mosfet Driver	23.8	10 pc
					100	
	6	TDC-GP21	$6.60	Time-to-Digital converter IC	12.5	
	7	STM32F303CBT6	$5.20	MCU	53	
	8	DRV11873PWPR	$3.50	DLDC driver DRV11873 DLDC driver DRV11873 (China, 1$)	40	
	9	BSZ165N04NS G	$1.30	Fast MOSFET	2.5	
	10	Additional PCB components	$7.00		50	
	11	PCB Itself	$2.00	4 Layer PCB, price for 5 boards (<50x50mm)	15	
	12	Photodiode Lens	$9.00	Lens, CS mount, 25mm focal length, F1.2	28	
	13	CS lens holder	$2.00	Holder for CS lens	1.5	
	14	BLDC motor	$2.00	BLDC, outrunner, 20mm diameter, KV > 200	4	

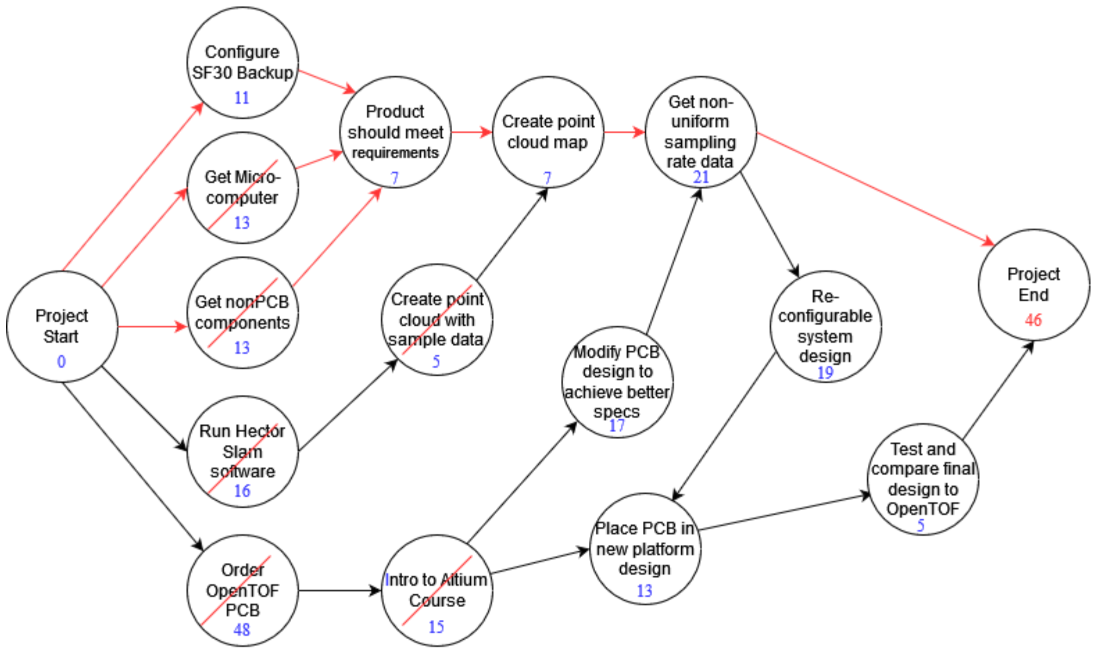


15	Scanning Mirror	$3.00	Customized shape, Alum, front surface mirror (12$ for 4 mirrors)	10	
			soft mirror	30	
16	Stand-offs	$3.50	25mm M3 Hex Stand-off x 8, price for 20pcs	8	10 male
				1.5	5 female
17	Laser lens	$7.00	25mm M12 F2.0 lens (Free delivery)	30	
				18.5	
				18	
18	Laser lens holder	$3.00	Plastic holder, 10mm height, 20mm hole distance (price for 10pcs)	1	
				1	
19	M12 lens adapter	$1.00	Price for 1pcs	1.2	
				3	
		$93.40	**Note - this total price is calculated without delivery price**		
		$62.90	PCB components including PCB, without delivery	664.5	
_1_	[AD500-9 TO52S1F2	$27.00	Chinese Avalanche photodiode (NOT proved but similar to AD500-8), TO-52 package, interated interference filter	149	with filter
				108	without filter

3 Project management

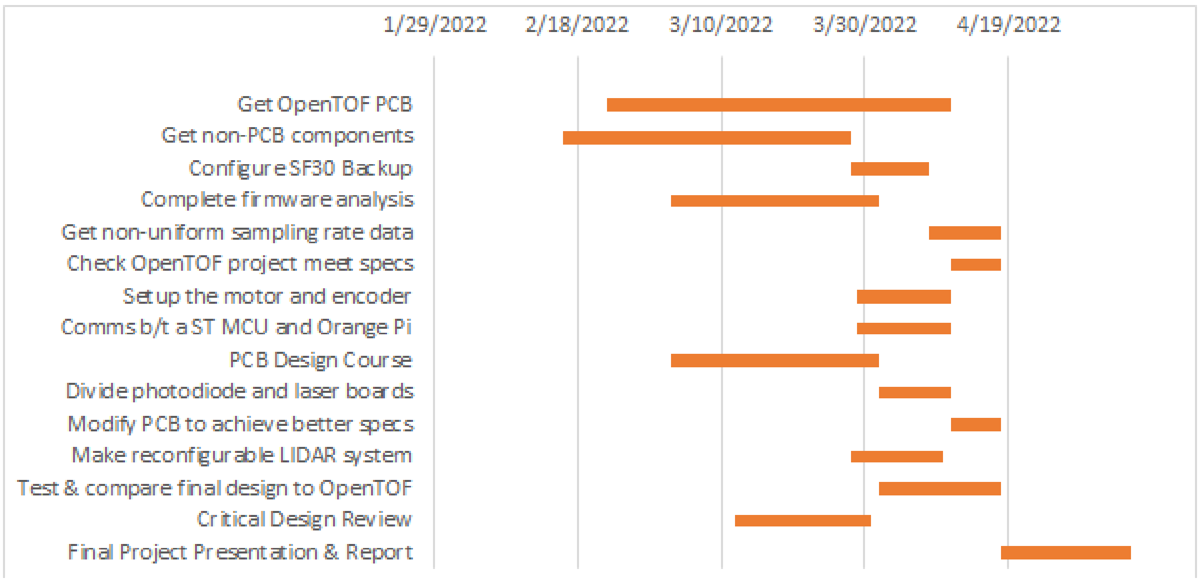
3.1 Updated implementation schedule

Task Dependency Chart



Configure SF30 Backup 11
Get Micro-computer 13
Get nonPCB components 13
Run Hector Slam software 16
Order OpenTOF PCB 48
Project Start 0
Product should meet requirements 7
Create point cloud map 7
Create point cloud with sample data 5
Intro to Altium Course 15
Get non-uniform sampling rate data 21
Modify PCB design to achieve better specs 17
Place PCB in new platform design 13
Re-configurable system design 19
Test and compare final design to OpenTOF 5
Project End 46

Gantt Chart



3.2 Updated validation and testing procedures

1) Start understanding the SF30C Lightware LIDAR and how to use it as a backup solution. Specifically, understand how to interface with the Atmel MCU on the SF30 PCB and have the SF30C TDC communicate with the Atmel MCU through SPI.

Assuming PCB arrives on April 1st:

2) If PCB arrives within the next week, flash the scanning firmware.
3) Adjust code after flashing the code once through the IAR Embedded workbench.
4) Adjust the configuration values in config files and programs such as the motor and encoder programs to account for different technology used in the project.

Assuming PCB does not arrive on April 1st:

2) From my conversation with Mr. Di, if we use the SF30 lidar, we can replace the PCB with this Lidar. The sampling rate is constant due to being commercial. So, we need to find a way to modify the programming on the microcontroller to make the sampling rate non-constant to mimic the MEMs mirror though changing the TDC and microcontroller settings.

3) 3D print something to properly place the lenses we got for the PCB laser and diode onto the SF30 photodiode and laser due to the different laser/photodiode sizes.

3.3 Updated division of labor and responsibilities

Scanning Firmware Analysis:

Apd_power.c/h	Gaurav
Capture_controlling.c/h	Allen
Config.h	Gaurav
dist_measurement.c/h	Allen
Encoder_processing.c/h	Aamhish
fifo_functions.c/h	Allen
hardware.c/h	Gaurav
main.c/h	Paul
mavlink_handling.c/h	Paul Roy
motor_controlling.c/h	Aamhish
stm32f30x_conf.h	Gaurav
stm32f30x_it.c/h	Aamhish
uart_driver.c/h	Gaurav
lidar_publisher.c/h	Paul Roy
open_tof_lidar.c/h	Paul Roy
EWARM	Aamhish
System	Paul
TDC	Gaurav
Nvram	Allen

Hardware Sub-Team

Gaurav -

1. Solder breadboard wires onto the motor and give that and raspberry pi to Aamhish.
2. Modify PCB design to have separate laser and photodiode boards.
3. Troubleshoot the backup solution - the SF30 lightware hardware to eventually change sampling rate to be non-uniform.

Aamhish -

1. Complete motor and encoder setup.
2. Setup and correct motor speed correction (PID) loop.
3. 3d print overhang.
4. Set up 2 separate holders for APD and laser PCBs.

Software Sub-Team

Allen -

1. Setup Orange Pi to work with Keil Uvision to flash code onto the PCB ST MCU.
2. Ensure distance data is sent from MCU to ROS nodes on orange pi properly.
3. Run "ROS across multiple machines" to view Hector slam results from Orange Pi on laptop [4].

Paul -

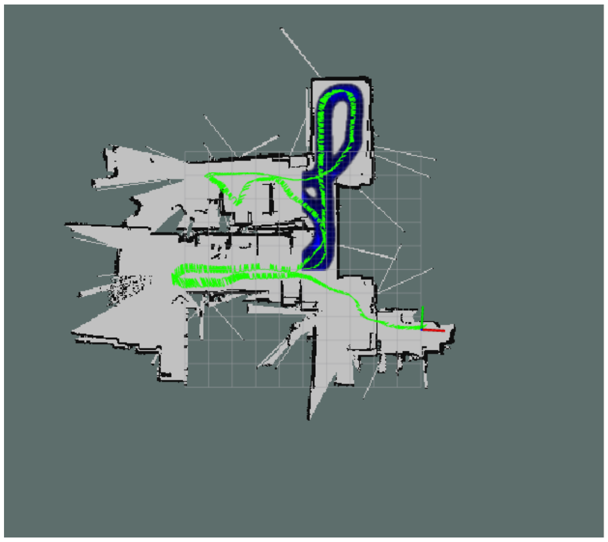
1. Ros nodes should work on Orange Pi to create a 2d point cloud map with Hector Slam software.
2. Also export data to bag file data for other people to analyze.
3. Help Allen to flash the code from the Orange Pi.

Our team plans to complete all of the deliverables for this project whether through the SF30 backup solution or through the PCB by April 18th, 2022.

4 Preliminary results

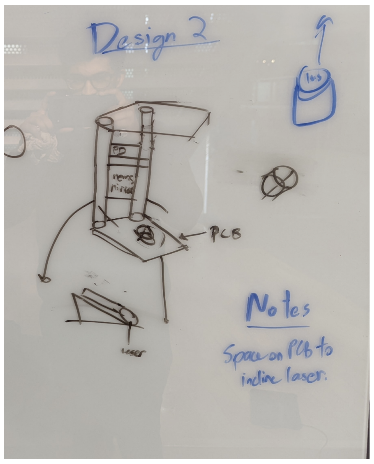
Provide any test results and demo of completed parts of the system at the time of the CDR.

We were able to thoroughly analyze the software and moderately analyze the hardware used in the OpenTOFLidar project.



The picture above is a screenshot of the output of the hector slam software. The graph vector icon represents the current position and orientation of the robot's position. The green icons represent places the robot has traveled and the general path it has taken. The gray represents places the lidar system has viewed through; that is the light beam has traveled through this area. The black line represents the wall that the system has found.

In order to generate the image we used a bag file. The bag file contains the position of the robot, the timestamp, the orientation, and where the flight of time laser senses the walls. The hector slam software puts all of these together and plays out how the robot moves and views the world around through sensors.



Gaurav and Aamhish PCB design start and reconfigurable system design as seen to the left. However, Mr. Di corrected that our design was not accurate since the MEMS mirror should be on the overhang in the same place the motorized spinning mirror platform is.

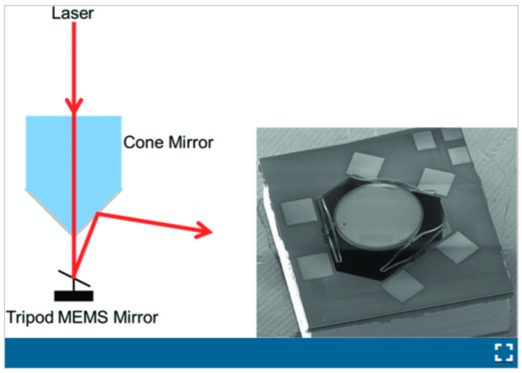


Figure 1:
(a) The concept of the omnidirectional scanner. (b) The tripod MEMS mirror.

Mr. Di corrected us and told us that the laser and the photodiode should be separated such that the laser is in line with the MEMS mirror vertically and that the photodiode can be placed anywhere as shown in the image above [3].

5 References

[1] iliasim, “OpenTOFLidar.”https://github.com/iliasam/OpenTOFLIDAR. (accessed Feb 02, 2022).

[2] “Avalanche Photodiodes.” https://lambdageeks.com/avalanche-photodiodes/

[3] D. Wang, C. Watkins, S. Koppal, M. Li, Y. Ding and H. Xie, "A compact omnidirectional laser scanner based on an electrothermal tripod mems mirror for lidar," *2019 20th International Conference on Solid-State Sensors, Actuators and Microsystems & Eurosensors XXXIII (TRANSDUCERS & EUROSENSORS XXXIII)*, 2019, pp. 1526-1529, doi: 10.1109/TRANSDUCERS.2019.8808659.

[4] “Running ROS Across Multiple Machines.” http://wiki.ros.org/ROS/Tutorials/MultipleMachines