Room-Scale 3D LiDAR Solution

Team 7

Jamie Xu (chengx2)

Xizheng Fang (xizheng2)

Terence Lee (tklee3)

ECE 445 Design Document – Spring 2021

TA: Andrew Chen (andrew6)

March 4, 2021

Table of Contents

1. Introduction

1.1 Problem and Solution Overview

Since the introduction of the autopilot feature by Tesla in 2016, there has been an explosive interest in autonomous driving. Many companies have since spent large amounts of resources and manpower in this profitable field, especially in the research of 3D Light Detection and Ranging (LiDAR) sensors, one of the more vital aspects of self-driving vehicles [1] (along with countless applications in surveying, movie special effects, sim racing, etc.) This device is utilized to collect the data of a vehicle's surroundings and output an accurate 3D point cloud map so the vehicle can avoid obstacles accordingly [2]. Because of the growth in popularity of autonomous vehicles, the interest in 3D LiDAR sensors has seen a similar increase among enthusiasts and hobbyists. However, unlike big corporations that could obtain industry-grade LiDAR technology for years, individuals typically cannot afford the high cost of entry for a 3D LiDAR sensor. To make it easier for hobbyists to explore the self-driving vehicle field, we are developing a system that will provide similar functionalities as the current 3D LiDAR industry solutions utilizing 1D time-of-flight (ToF) sensors while also remaining both affordable and easily obtainable.

The system we build is aimed towards the hobbyist market which has been largely neglected by the market up until now. We plan for the solution's friendly cost of entry to be our main selling point. Instead of using typical single-phase or solid-state 3D LiDAR sensors which cost upwards of thousands of dollars, we will build the system based on a ToF sensor (also called 1D LiDAR) that can be a hundredth of the cost. The LiDAR will scan the environment and perform distance measurements on a 2-axis platform with a stepper motor helping to rotate the LiDAR 360-degree on the platform. Meanwhile, another motor will pitch the sensor up and down so the sensor can scan its environments at various height levels. The recorded spherical coordinates of the measured area will then be converted into Cartesian coordinates and eventually exported as a point cloud file that can be visualized to show the scanned room.

1.2 Visual Aid

Our design chooses the tried-and-true industry design of a revolving sensor. We have striven to keep the physical footprint as small as possible for ease of integration into our clients' projects. As such, we plan to integrate a 1/4-20 UNC thread at the bottom of the housing, on-axis rotation for easy mounting on standard tripods. Though not pictured in this render, the sensor will need a USB connection to the computer as part of the data transfer process.

Figure 1: Render of Assembled Device and Some Mock Use-cases

1.3 High-level requirements list

- The system can scan the room and output successfully within a time frame of 20 seconds
- The point cloud output by the system will have an accuracy of ±0.15 m (meters) at a distance of 3 m and ±0.06 m at a distance of 0.2 meters.
- The output file will use the industry-standard LASer (LAS) file format specified by the American Society for Photogrammetry and Remote Sensing (ASPRS) and be readable by most external software.

2. Design

2.1 Block Diagram

The 3D LiDAR system consists of five major subsystems: primary sensor, vertical rotation, horizontal rotation, control system, and computer. The primary sensor subsystem is responsible for ensuring the ToF sensor readings are accurate. The vertical and horizontal rotation subsystems control the rotational movement of the ToF sensor in vertical and horizontal planes respectively. They work in tandem with the primary sensor subsystem to emulate a scanning effect similar to its industrial 3D LiDAR counterparts. Additionally, these three subsystems are managed by the control system allowing the sensor to scan an indoor room with an approximate dimension of 10x10x5 m in 20 seconds with an accuracy of ±0.1 m at a distance of 0.2 to 3 m. After the control system receives the measurement data from the sensor under the I^2C communication protocol, it will convert the 3D spherical coordinates to Cartesian coordinates then pass the processed data to the computer subsystem from which the python program will output a 3D point cloud file of the scanned environment.

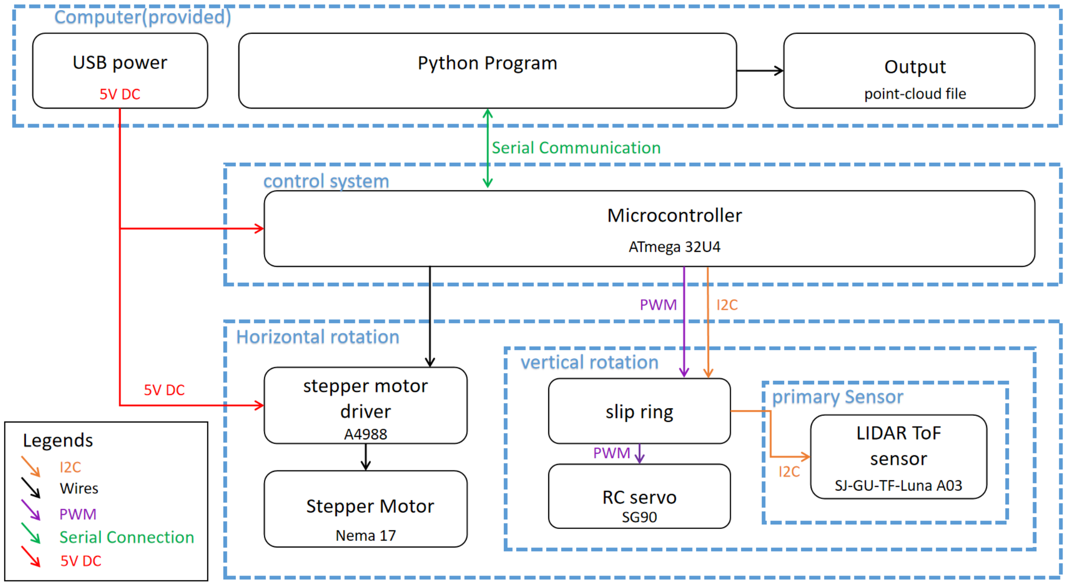

Figure 2: Block Diagram

2.2 Physical Design

Since our product is designed to be integrated into our clients' projects, we focused most of our effort on minimizing the physical footprint and simplifying everything down to the essentials. As shown in Figure 3, our final design consists of only a base platform and a rotating sensor, with a total dimension of 113×102×64 mm. From the outside, the base platform has a micro USB port for powering the device and communication between the device and the computer, a DC barrel jack for redundancy, and a 1/4-20 UNC thread at the bottom of the housing, on-axis to the rotation for easy mounting on standard tripods.

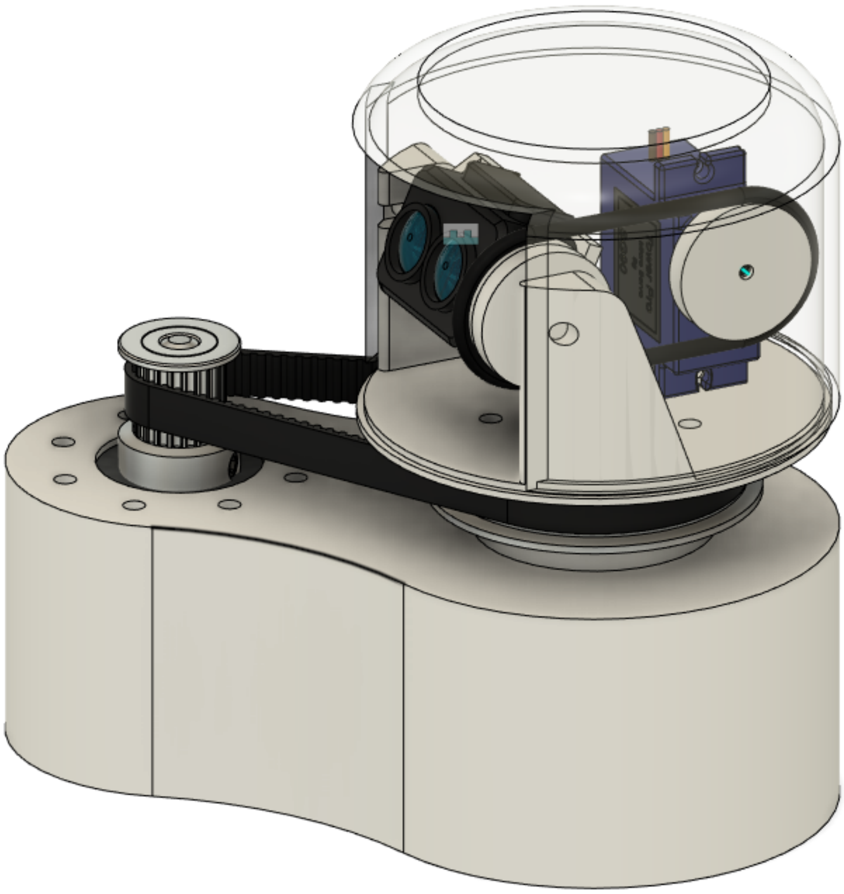

Figure 3: Physical design of sensor and motors

A lot of thought has gone into minimizing the overall dimensions of the product without compromising on the mechanical rigidity of the assembly. We have conveniently divided the mechanical subassemblies in the same way as the block diagrams and kept every block separate and modular.

For the vertical rotation block from the block diagram, this part is going to be rotating continuously, and therefore the footprint of this subassembly directly dictates the final dimension of our product. As you can see from Figure 4, the LiDAR sensor pivots vertically while being driven by the micro-servo. We chose to “fold-up” the servo-sensor assembly using an o-ring as a short belt, cutting the diameter of the bounding cylinder in half while also minimizing the rotational inertia.

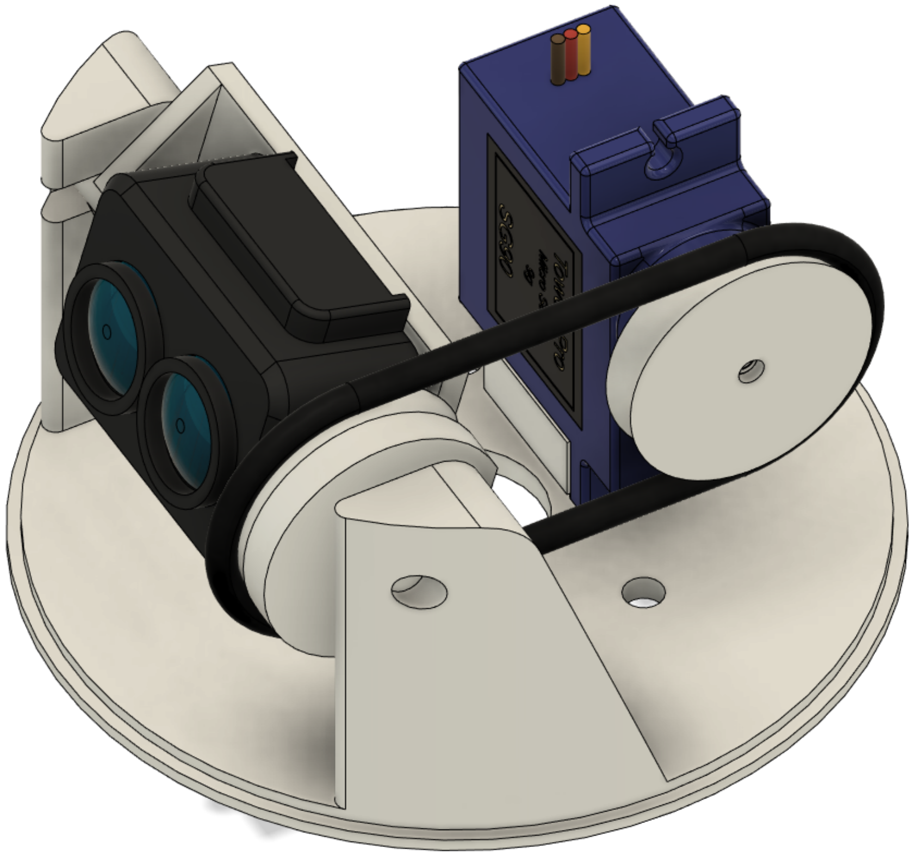

Figure 4: Physical design of the sensor and vertical servo

Since our design revolved around a continuously rotating subassembly, a lot of elements needed to be fitted onto the axis of rotation, including a slip ring, the motor, the mechanical restraint, and the subassembly itself. We have determined that it is impossible to fit everything mentioned directly onto the axis of rotation while maintaining affordability and a small footprint. As you can see in the exploded view of our physical design (Figure 5), we chose to use a GT2 timing belt along with two pulleys to offset the NEMA 11 stepper motor. The slip ring remains in the center of the rotation, with a larger bearing ensuring the accuracy and rigidity of the rotating assembly.

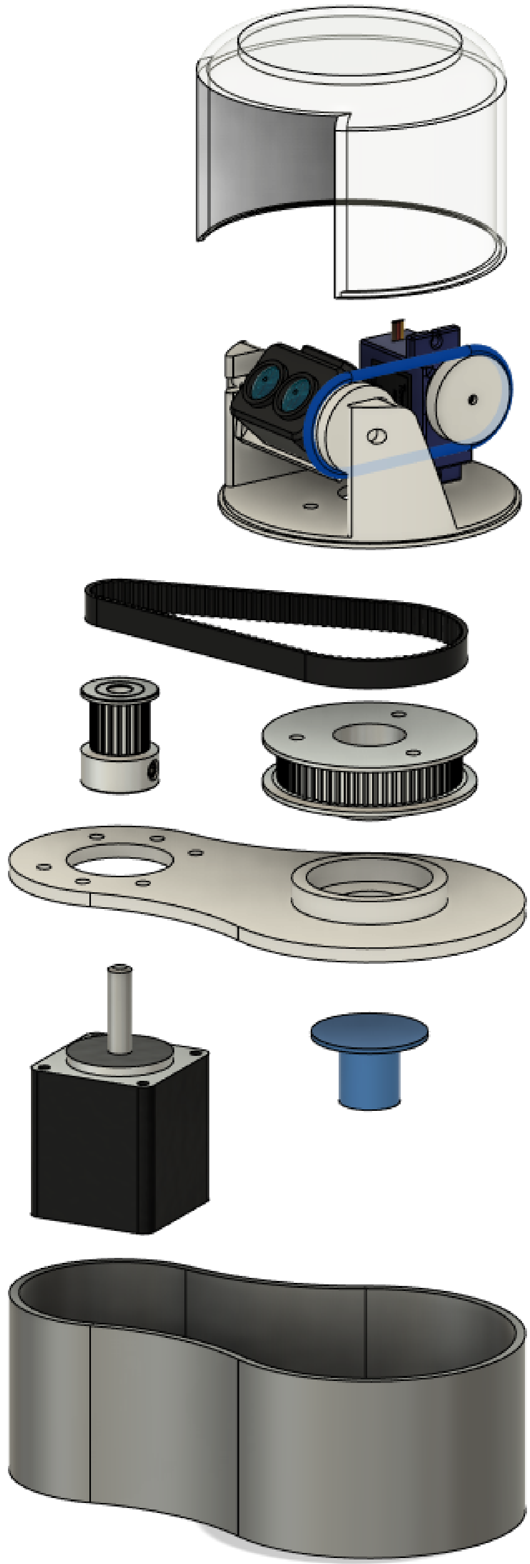

Figure 5: Dissembled physical design

We have also determined that it is possible to fit both the microcontroller and the A4988 driver inside the base housing using a double-layered PCB with custom dimensions without increasing the footprint of our product.

2.3 Functional Overview & Block Diagram Requirements

Subsystem 1: Primary Sensor

The primary sensor subsystem consists of a ToF sensor. The subsystem references the environment through the sensor by taking 1D point-to-point distance measurements from the environment. It receives DC power and sends digital signals from and to the microcontroller subsystem through the slip ring. The supply voltage needs to be kept between 3.7 to 5.2 V to ensure the sensor can function reliably, while the logic voltage needs to be kept at around 3.3 V. In addition, the power supply will be adequate to ensure the sensor can work for at least 20 seconds to take a full scan of the environment.

The sensor communicates with the microcontroller under the I^2C protocol and is responsible for sending the collected data to the control subsystem to be processed with the maximum transmission rate of 400 kbps. The connected wire starts from the sensor, passes through the slip ring, and finally goes to the microcontroller. With the help from both the vertical and horizontal rotation subsystems, the combined system can achieve a 3D scan of an indoor room (such as living rooms, classrooms, etc.)

Primary ToF Sensor (SJ-GU-TF-Luna A03)

We will use the SJ-GU-TF-Luna A03 LiDAR module as our ToF sensor. This specific sensor was chosen after researching the options available due to it having the smallest footprint and the lowest cost of entry while maintaining a maximum range of 8 meters and a maximum refresh rate of 250 Hertz [3]. The sensor will have a 360-degree continuous rotation and take distance measurements in every direction in a 2D plane with the help of the stepper motor to the precision of ±0.10 m at a 3 m distance and ±0.02 m at a 0.2

meter distance. In addition, the RC servo will pitch the ToF sensor up and down to ensure the sensor can reach various height levels allowing for a 3D environment of dimension 10x10x5 m to be scanned. The ToF sensor's frame rate will be adjusted to within 5% of 200 Hz so it can fully scan the room in 20 seconds to leave enough time for the point cloud file generation. In addition, the sensor has a 2-degree field of view, it can be operated to take a 360-degree scan in a horizontal plane in 200 steps in 1 second. This step number is in accordance with the step number that the Nema 11 stepper motor can take to have a horizontal rotation in 1 second. Thus we can ensure that the sensor can work synchronously with the horizontal motor.

Requirement	Verification
The LiDAR sensor can receive enough power ranging from 3.7 to 5.2 V to ensure the sensor can work in normal conditions for at least 20 seconds.	1. Connect the LiDAR sensor to a computer through the USB port. 2. Use an oscilloscope to measure the voltage across the power pins of the LiDAR sensor to make sure that the voltage is within the range from 3.7 V to 5.2 V while the platform is rotating at its typical speed of 60 RPM
The LiDAR sensor sends the data smoothly to the control subsystem through a wire under the I²C protocol.	1. Fix LiDAR sensor in a fixed position so it can take a distance measurement in a certain direction. 2. Connect the LiDAR sensor to the microcontroller through a wire that is under I²C protocol. 3. Write a program that streams the reading from the sensor over serial.

	4. check if any of the readings are lost or irregular.
The LiDAR sensor polls at a frequency of 200 Hz ±5% to ensure adequate scanning efficiency outlined in the high-level requirements.	1. Connect the LiDAR sensor to the microcontroller through a wire that is under I²C protocol. 2. Write a program that streams the reading from the sensor over serial at 200 Hz. 3. Check if the LiDAR sensor readings are consistent. 4. Push the frequency even higher to see the tolerance range.
The sensor has a max operating range of 6 m that enables it to scan a room with a dimension of 10x10x5 m.	1. Fix LiDAR sensor in a fixed position so it can take a distance measurement in a certain direction. 2. Set an object approximately 6 meters away from the LiDAR sensor and connect it to the microcontroller under I²C protocol. 3. Start the operation of the sensor and check the controller for a clear distance output. 4. Use a tape measure to measure the actual distance and ensure the error is within 0.10 m of the actual distance.

Subsystem 2: Vertical Rotation

The vertical rotation block is in charge of pitching the sensor up and down so the sensor can scan the various horizontal planes at different heights. It consists of a slip ring and an RC servo. When the scan process begins, the RC servo will pitch the LiDAR sensor up and down under the PWM control from the control block. Meanwhile, the control signal and the data collected by the sensor will be transferred past the slip ring to the microcontroller in the control block. In addition, the servo will receive a DC voltage ranging from 5 to 5.5 V from the USB in the computer block passing through the slip ring to keep it operating at the voltage within 10% of 4.8 V.

Slip Ring

The slip ring inside the block is connected to the ToF sensor and the RC servo in the primary sensor block on one side, and the microcontroller on the other. It makes sure the wire does not get twisted when the sensor is rotating so that the stepper motor in the horizontal rotation block can rotate infinitely to ensure the sensor's continuous scan. The slip ring acts to maintain the signal integrity of three components. One of the leads will supply 5 to 5.5 V DC power to the electronics within the vertical rotation and primary sensor block. The communication between the ToF sensor and the control system is run through the slip ring and is under the I^2C protocol. The PWM signal coming from the microcontroller into the RC servo also passes the slip ring.

Requirement	Verification
The slip ring ensures the PWM and I^2C signals are not compromised under constant rotation. Also, the noise induced by the slip ring must not interfere with the I^2C signal sent from the primary ToF sensor or the PWM signal going to the RC servo.	1. Attach the LiDAR sensor on the vertical rotation block that consists of the slip ring and the Micro Servo. 2. Connect the LiDAR sensor to the microcontroller using a wire that runs under I^2C protocol and passes through the slip ring. Also, connect

	the servo to the controller using a wire that transmits the PWM control signal from the controller to the servo and passes through the slip ring. 3. Start the operation of this system and check if the system works normally to make sure that the I^2C and PWM signals are transmitted correctly. Also, check if the sensor sends out a clear output and the servo is rotating constantly and continuously to ensure that the noise from the slip ring does not interfere with the I^2C and the PWM signals.
The 5 to 5.5 V DC power coming out of the slip ring must be smoothed out enough to fit the 3.7 to 5.2 V requirement of the ToF sensor.	1. Connect the LiDAR sensor to a 5 to 5.5 V DC power from a computer USB port through a wire that passes through the slip ring. 2. Use an oscilloscope to measure the voltage across the sensor while the slip ring is rotating to make sure that the voltage is still within the range from 3.7 to 5.2 V.

RC Servo

We will use an SG90 9G Micro Servo as the RC servo in this subsystem. The servo, which is regulated by the control system through PWM, can pitch the sensor up to 180 degrees so that the ToF sensor can aim up and down. However, due to the limitation of the height of the sensor and the structure of the scanned environment, the servo usually will only pitch a 60-degree range in the vertical plane to help the sensor to collect the required distance measurements. In addition, we decided to use 10 steps in the vertical rotation, which means we will have to angle across 6 degrees between each rotation for vertical rotations. Because the angular resolution of the LiDAR ToF sensor is 2 degrees, the accuracy of the servo needs to be kept around 1 degree to guarantee accuracy and repeatability. The servo receives a 5 to 5.5 V DC power from the computer block through the USB cable and the power needs to be within 10% of 4.8 V to ensure the servo functions normally.

Requirement	Verification
The RC servo must receive a DC power supply ranging from 5 to 5.5 V from the computer block to make it operate at the voltage of 4.8 V ±10%.	1. Connect the micro-servo to a 5 to 5.5 V power from the USB through the slip ring. 2. Write a program that sweeps the servo across its range of motion. 3. Start the operation of the servo and use an oscilloscope to measure the voltage across the servo to ensure that the voltage remains within 10% of 4.8 V.
The servo must be able to rotate across a range of 6 degrees in 1 second under a continuous 5 to 5.5 V DC power supply.	1. Connect the Servo to the microcontroller as mentioned before. 2. Write a program that sweeps the RC servo in 6-degree increments. 3. Use a stopwatch to time the duration of each movement. Ensure the recorded time is within 1 second.

The servo must be able to repeatedly go to a certain angle with an accuracy of ±1 degree.	1. Connect the Servo to the microcontroller as mentioned before. 2. Write a program that sweeps the RC servo across its full range of motion and stops at a predetermined angle. 3. Print an indicator arm that fits around the output shaft of the servo. 4. Use a protractor to ensure that for each cycle the stopped position is within 1 degree of each other.

Subsystem 3: Horizontal Rotation

The Horizontal Rotation block, which consists of a stepper motor driver and a stepper motor, rotates the sensor in the horizontal plane so the sensor can scan 360 degrees around its surroundings. It works with a vertical rotation block and the primary sensor block to make sure we can have a 360-degree 3D scan of an indoor room with an approximate size of 10x10x5 m in 20 seconds. The stepper motor driver operates at a voltage between 3.3 to 5 V. It receives DC power ranging from 5 to 5.5 V from the computer block through the USB cable and receives the control signal from the microcontroller in the control block. When the scanning process begins, it will control the stepper motor to rotate the system horizontally to ensure that the sensor can have 360-degree scanning of the horizontal plane.

Stepper Motor Driver & Stepper Motor

We will use a Nema 11 stepper motor as our stepper motor and an A4988 micro-stepping motor driver as our motor driver. The motor driver will receive commands through wires from the microcontroller in the control system block. After it receives the commands, it will operate the motor so the motor will rotate the sensor in the primary sensor block horizontally with a minimum step angle of 1.8 degrees. In addition, the motor is connected with the platform that holds an LiDAR sensor by timing belts and the sensor has a field of view of 2 degrees so it needs around 200 steps to have a scan with a reasonable resolution. Because the pulley for the platform has 60 teeth and the pulley for the stepper motor has only 20 teeth, in order to coordinate the platform RPM with the required steps for the sensor, the motor driver operates the motor at a step angle within 5% of 1.8 degrees and step 3 steps for each measurement. Therefore, the stepper motor can maintain enough accuracy to ensure the integrity of the data. The whole block is powered by the computer through a USB cable that carries a DC voltage of 5 to 5.5 V. At the same time, we have to make sure that the driver operates under a voltage between 3.3 to 5 V and the motor operates under a voltage of 3.75 V ±10% so they can work in normal conditions.

Requirement	Verification
This subsystem must be repeatable enough to maintain spatial awareness over 30 rotations.	1. Connect the stepper motor to the stepper motor driver. 2. Connect the motor driver to the microcontroller.

	3. Write a program that will rotate the stepper motor continuously at the speed of 60 RPM until given an angle from the computer. When given the angle, the stepper motor will rotate to that angle. 4. Wait for 30 revolutions and then input a random angle. 5. Observe whether the stepper moves to the correct angle using a protractor.
The motor driver receives a voltage between 3.3 to 5 V and the motor receives a voltage within 10% of 3.7 V from a 5 to 5.5 V power supply that comes from a computer through USB cables to make sure the subsystem can help the sensor to finish scanning an indoor (10x10x5m) in 20 seconds.	1. Connect the stepper motor driver to the microcontroller and motor driver via wires. 2. Connect a 5 to 5.5 V DC power from a computer to the motor driver through a USB cable. 3. Start the operation of the motor drive and the motor; use an oscilloscope to measure the voltages across the motor driver and the motor to ensure that the motor drive has a voltage between 3.3 to 5 V and the motor has a voltage within 5% of 3.7 V.
The stepper motor has a step angle within 5% of 1.8 degrees to make sure that the scanning process can have enough resolution and precision of ±0.10 m at a distance of 3 m and ± 0.02 m at a distance of 0.2 meters.	1. Connect the stepper motor to the stepper motor driver. 2. Connect the motor driver to the microcontroller. 3. Write a program that will step the stepper motor 1 full step

	4. Measure with a protractor whether the rotation is 1.8 ± 5% degrees.

Subsystem 4: Control System

The Microcontroller coordinates all the other subsystems so that the rotation of the vertical and horizontal axis is coherent, systematic, and repeatable enough for the ToF sensor to obtain meaningful measurements. It is also responsible for instructing the stepper motor driver to move horizontally to its next position (current position + 1.8°) after the ToF sensor receives a reading and pitching the sensor to the next vertical angle (current position + 6°) after a full 360° horizontal revolution. At the same time, the sensor is gathering data, the microcontroller sends converted data via a serial connection to the computer subsystem for LAS file generation.

Microcontroller (ATmega32U4)

We decided to go with this specific microcontroller as it has been very well documented and supports both I^2C and PWM communication protocols needed for the system. The microcontroller sends pulses to the A4988 chip that controls the stepper motor, maintains I^2C communication with the ToF sensor (400 kbps maximum transmission rate) for distance readings, outputs PWM signals to the RC servo setting the sensor angle, and establishes a serial connection to the client's computer streaming the coordinates of different points all simultaneously. Before sending the sensor data to the computer, it is first converted from the LiDAR system's spherical coordinate space into the Cartesian coordinate space. This is accomplished via an Arduino C program that utilizes a spherical to cartesian formula further discussed below in the Tolerance Analysis. In addition, the controller is powered by a 5 to 5.5 V DC power supply from the computer's USB port and operates under a voltage of 4.5 V ±10%.

Requirement	Verification
Converts the spherical coordinates into Cartesian coordinates at a rate of 200 Hz ±5% (refresh rate of ToF sensor).	1. Connect the microcontroller to the computer 2. Load the Arduino with the C program that converts the input coordinates 3. Artificially generate 200 tuples of measurements for the distance, vertical and horizontal angle. 4. Time the spherical to cartesian conversion rate to confirm the entire process is at or under 1 second. 5. Collect and verify that the conversion between spherical and cartesian coordinates was successful. 6. Repeat steps 3 to 5 multiple times to ensure that we can have a successful conversion at a rate within 5% of 200 Hz.
The controller receives a 5 to 5.5 V DC power from a computer to operate at a voltage within 10% of 4.5 V.	1. Connect the microcontroller to a computer via USB. 2. Run the microcontroller on some arbitrary code and use an oscilloscope to measure the voltage across the controller to ensure that it operates at a voltage within 10% of 4.5 V.

PCB

As shown in Figure 7, the microcontroller powers the A4988 chip by a 5 V power source. Simultaneously, it controls the pulses of the A4988 chip by sending signals to the STEP pin in the A4988 chip in Figure 7. After receiving the pulse limit from the microcontroller, the A4988 chip can control the Nema 11 stepper motor by the four output pins on the right of the A4988 chip in Figure 7 so we can ensure the motor can be operated at a step angle within 5% of 5.4 degrees. Therefore, the stepper motor can work synchronously with the sensor and the servo to ensure a scan cycle for at most 20 seconds. The PCB footprint will fit perfectly inside the base housing, saving physical space.

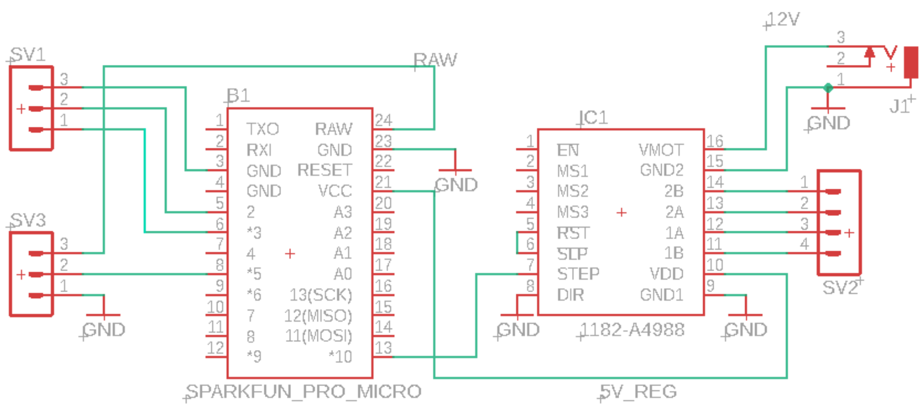

Figure 6: Schematic for the microcontroller and A4988 stepper motor driver

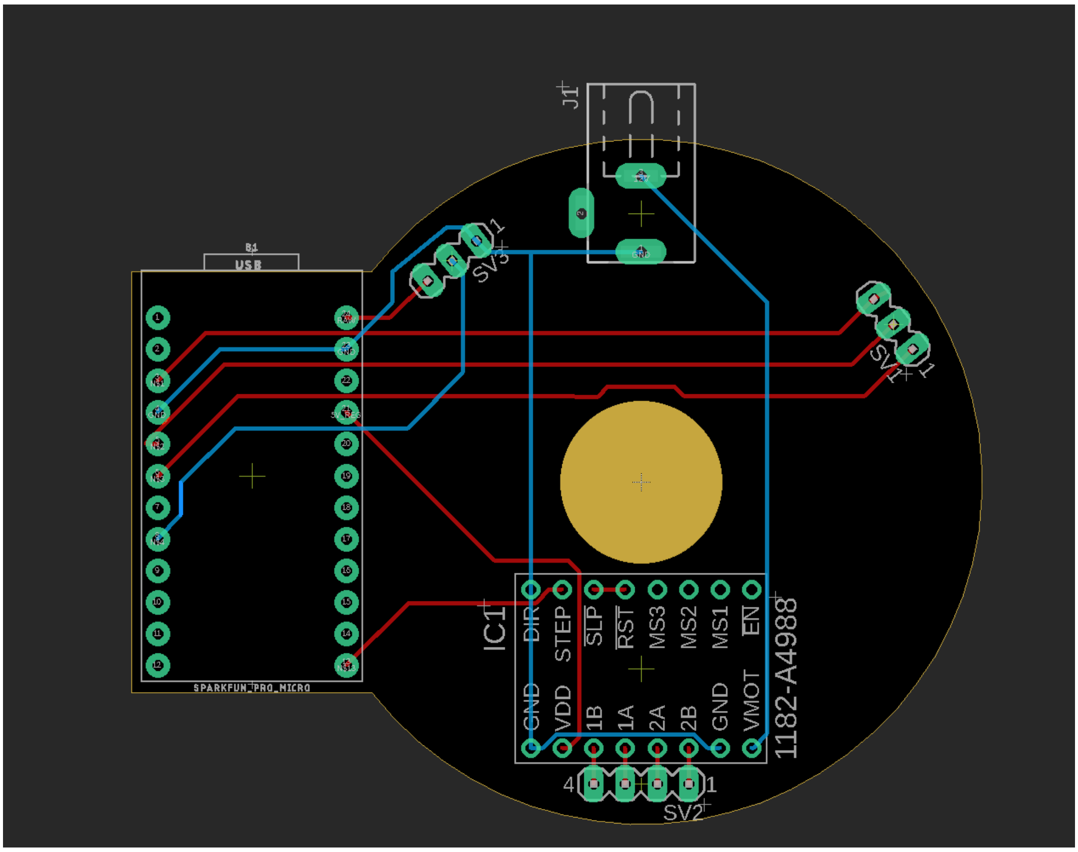

Figure 7: Eagle PCB Layout for the microcontroller and A4988 stepper motor driver

Requirement	Verification
Each trace is uncompromised.	1. Use a multimeter in continuity mode to test each trace to ensure the PCB manufacturing is without fault.
Components will work with the PCB incorporated.	1. Solder every component onto the PCB.

	2. Write a program that will briefly unit test the functionality of each component.

Subsystem 5: Computer

The Computer subsystem is fed in converted cartesian coordinates via a continuous serial data stream from the microcontroller. The data will then be converted into the LAS point cloud file format through a python program that utilizes the laspy and NumPy libraries. Simultaneously, the computer supplies power to the entire system in the form of 5 V/500 mA via its USB port.

USB Port

5 to 5.5 V DC are output from the computer via a USB Type-A port, which will power the microcontroller, micro-servo motor, a LiDAR sensor, and a stepper motor to handle rotational movement. Since the entire system is powered via a singular port, the scan/conversion process is heavily reliant on consistent power from the port.

Requirement	Verification
The USB port can continuously output a voltage from 5 to 5.5 V and a maximum current draw of 500 mA ±5% to the rest of the system.	1. Connect the computer, microcontroller, and the sensor-servo subsystem that consists of the servo motor, LiDAR sensor, stepper motor, and micro-stepping motor driver.

	2. Attach a multimeter in between the Computer and the cable providing power to the system. 3. Ensure that the voltage ranges from 5 to 5.5 V and current do not exceed 500 mA.

Python Program

The python3 program is fed a stream of 1D measurements that have been converted from spherical to cartesian coordinates by the microcontroller. First, the program utilizes the NumPy python library to classify the data into a readable format as well as stores key traits in the data such as maximum/minimum data points, offsets, scale factors, etc. The sorted data and stored values are then used to construct a LAS file and header respectively using the laspy python library. The LAS file type is an open format specified by the American Society for Photogrammetry and Remote Sensing (ASPRS) and the standard for the "interchange of 3-dimensional point cloud data" [4]. Once in this format, the point cloud can be imported and visualized into numerous different software.

Requirement	Verification
The program successfully converts the raw data into a LAS file within the 20 seconds specified as a high-level requirement.	1. Feed-in pre-generated spherical coordinates into the python program and obtain the LAS output file.

	2. Run a file hash checksum on the output LAS file to ensure the file has been encoded properly.

2.4 Tolerance Analysis

2.4.1 Overall accuracy

After reviewing the different subsystems, our team believes that the overall accuracy of each individual point in the point cloud warrants further tolerance analysis. Due to the nature of our design, the ToF sensor, the horizontal rotation, and vertical movement systems are nested within each other. Therefore, errors from the inner block in the block diagram are carried over and compounded through to the outer block.

For each subsystem, we get data in the form of distance(r), theta (θ), and phi (φ) respectively (See Figure 8.)

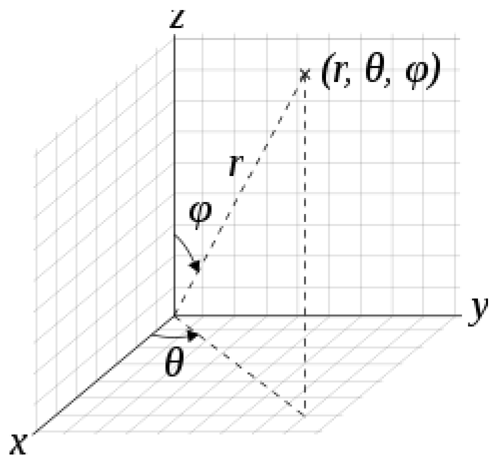

Figure 8: Error range illustration

To convert the obtained spherical coordinates into cartesian coordinates that the point cloud uses, we follow the formulas below.

$$
\begin{aligned}
x &= \rho\cos(90^\circ - \varphi)\cos(\theta) \\
y &= \rho\cos(90^\circ - \varphi)\sin(\theta) \\
z &= \rho\sin(90^\circ - \varphi)
\end{aligned}
$$

Equation 1: Polar to Cartesian Conversion

To simulate the error that our systems might introduce, we can introduce an error term for each variable and calculate a new coordinate:

$$
\begin{aligned}
x_e &= \rho\cos(90^\circ - \varphi + e_\varphi)\cos(\theta + e_\theta) \\
y_e &= \rho\cos(90^\circ - \varphi + e_\varphi)\sin(\theta + e_\theta) \\
z_e &= \rho\sin(90^\circ - \varphi + e_\varphi)
\end{aligned}
$$

Equation 2: Polar to Cartesian Conversion with error

We can then obtain the overall deviation resulted from all the erroring being propagated from the beginning by calculating the Euclidean distance between the actual coordinate and the coordinate with the estimated error.

$$
E = \left\| \begin{bmatrix} x \\ y \\ z \end{bmatrix} - \begin{bmatrix} x_e \\ y_e \\ z_e \end{bmatrix} \right\|
$$

Equation 3: Error Derivation via Euclidean distance

Based on the component datasheets and our calculations from block descriptions earlier, we establish the error for the sensor, vertical, and horizontal to be ±5 cm, ±1°, and ±0.2° respectively. By substituting the three error margin into the three variables e_ρ, e_φ, and e_θ, and sweeping the distance ϱ over the intended range of 20 cm to 300 cm in python using Matplotlib, we obtain the error plots shown in Figure 9.

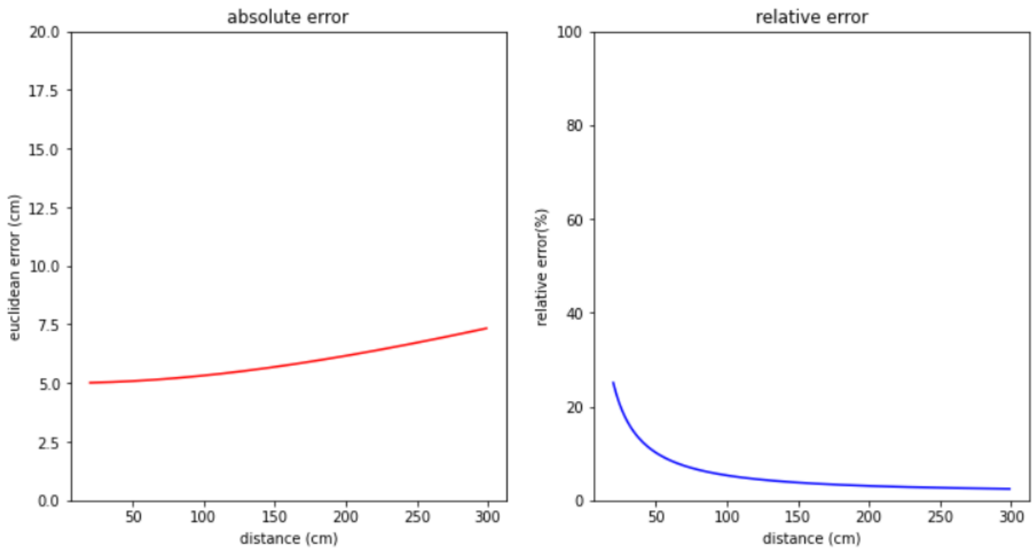

Figure 9: Error range simulation results derived from python by using Matplotlib

As we can see, with the calculated errors presented before, the absolute error starts at 5 cm and peaks at the max distance of 300 cm at 7.33 cm. While not shown in the graph, from the simulation we also know that the error of 10 cm listed in our high-level diagram was not met until 500 cm, while the percent error continues to decrease even at 500 cm. Looking at our high-level requirements, the error value falls within the ±15 cm for 300 cm and ±6 cm for the 20 cm acceptable range for our system. With this in mind, we believe that under the assumptions all of the system's components are within the error specified in their datasheets, the data collected will be able to fulfill our accuracy-based requirements.

3. Cost and Schedule

3.1 Cost Analysis

Cost of labor

Using the hourly labor rate of $50 as provided in the instructions with a 250% overhead multiplier, each senior design business labor hour translates to

$125. On average, 8 hours per week is spent each for three people over a period of 8 weeks from the proposal deadline to the demo date, the total labor cost is calculated below:

$$\$50/hr \;*\; 2.5\,(overhead) \;*\; 8\,hr/week \;*\; 3\,people \;*\; 8\,weeks \;=\; \$28,800$$

Equation 4: Labor Cost Calculations

Bill of Materials

Part	Part No.	Price
6 wires x2A flanged slip ring	SRW-12-06A	$ 5.49
Micro RC Servo	SG90	$ 1.70
Deep-Groove Bearing	6803-2RS	$ 0.44
NEMA 11 stepper	11HS12-0674S	$ 18.79
GT2 60T timing pulley	KINGPRINT-WCL000015	$ 8.99
GT2 60T timing pulley	M-GT2-PULLEY-20T	
GT2 200mm belt	200-2GT-6	
single-point LiDAR module	SJ-GU-TF-Luna A03	$ 22.89
microcontroller development board	Atmega32U4	$ 4.99
Stepper motor driver	A4988	$ 1.59
Total		$ 64.88

Cost of Custom manufacturing:

Manufacturing Method	Hours	Rates Per Hour	Material Cost	Overall Cost
3D printing	20	$2.00	$19.99	$59.99
machine shop	1	$50.00	$0.00	$50.00
PCB	N.A	N.A	$4.99	$4.99
Total				$114.98

As shown above, the total development cost of this project is $28979.86 when labor is considered. It is worth mentioning that the material cost can be reduced when bulk ordering and the 3d printing cost can be reduced down to a few dollars per part when converting to injection molding. With all this in mind, we believe we can reduce the material and manufacturing cost down to close to $100, leaving us enough overhead to market to the enthusiast and hobbyist market.

3.2 Schedule

Date	Jamie	Xizheng	Terence
Mar 8	Write the Arduino code to unit-test the ToF sensor and servo. assess the options for driving the stepper motors. Start designing the mechanical	Study documentation from data sheets and wire up the servo, stepper, and ToF sensor individually for	Establish a serial connection to and from the microcontroller, ensure the bandwidth is enough for the

	components.	unit testing.	intended data transfer.
Mar 15	Finalize the mechanical design and provide the drawing to the machine shop. Work with Terence to get the reading from the sensor to the python program.	Wire up all the components together according to the schematic with and without the slip ring to ensure the PCB is designed correctly	Visualize the received data using Matplotlib inside Jupyter Notebook.
Mar 22	Ensure the horizontal rotation is synced properly with the polling of the sensor. Work with Terence to produce a 2D point cloud.	PCB prototype/revisions sent to PCBway.	Explore documentation for the laspy library and familiarize himself with it.
Mar 29	Integrate the final vertical rotation and produce a 3D point cloud.	Wire all the components with the PCB and finalize the mechanical/electrical design.	Ensure the file outputs properly and within the time limit.
Apr 5	Optimize the code to tighten the timing of rotation and polling.	Finalize PCB design to be sent off to PCBway	Experiment with real-time visualization if time permits
Apr 12	Reserved for unexpected changes	Reserved for unexpected changes	Reserved for unexpected changes

4. Discussion of Safety and Ethics

Our project aims to lower the cost of entry into the light-based scanning field. By minimizing the construction cost and streamlining the assembly, setup, and maintenance processes, we hope to make it easier for everyone to gain a better "understanding...of conventional and emerging technolog[y]"[5] that is 3D LiDAR.

In the efforts to "protect the privacy of others" [5] following Section I-1 of the IEEE Code of Conduct, after heavy deliberation, we do not believe there are any privacy concerns from a user data perspective. This is based on the fact that all data that the system collects is stored offline and does not interact with the internet at any point throughout the scan process.

As stated in the IEEE Policies Section 7.6, the "safety, health, and welfare of the public" [6] is to be held paramount. With the ToF sensor utilized in our design, there is an 850 nm laser that has the potential to cause damage to the human eye. With this wavelength of light, humans cannot distinguish its intensity and can easily be injured as a result [7]. To account for this, our team will utilize special safety laser goggles that protect against the laser and will include a safety warning to any consumers.

Taking into account other possible safety concerns such as system voltage, we have concluded that there are no electrocution-related issues as the maximum voltage used by the system of 5.5 volts is not enough to cause any significant or lasting harm [8]. Additionally, the USB standard is designed such that the 2.6 watts of power it outputs does not pose a fire hazard for the system.

It is important to note that a cheaper cost of entry for 3D LiDAR systems could make it easier for an individual to conduct more precise attacks via an unmanned device. Already, higher complexity LiDAR systems are being utilized by the US military. "...military scientific research institutes in the United

States have developed...LiDAR seekers with specific algorithms have the ability of automatic target recognition [9].” Though our project does not exist to promote such uses, there remains a definite possibility of a third party utilizing our system alongside others to threaten public safety which also contradicts Section 7.6 mentioned above. There does not seem to be a good way to proactively counteract such utilization of our 3D LiDAR solution as the use case of general surrounding scanning is too broad and mostly innocuous to try and police. Any sort of countermeasures integrated into the system would most likely detriment the user experience significantly. With this in mind, we acknowledge the slight possibility of the use of our system for unintended purposes that could breach the IEEE code, however in the efforts to maintain device functionality and given the low probability of misuse, the system will not be designed to counteract this specific concern.

5. Citations

[1] R. Amadeo, “Google's Waymo invests in LIDAR technology, cuts costs by 90 percent,” Ars Technica, 10-Jan-2017. [Online]. Available: https://arstechnica.com/cars/2017/01/googles-waymo-invests-in-lidar-technology-cuts-costs-by-90-percent/. [Accessed: 18-Feb-2021].

[2] C. Domke and Q. Potts, “LiDARs for self-driving vehicles: a technological arms race,” Automotive World, 03-Aug-2020. [Online]. Available: https://www.automotiveworld.com/articles/lidars-for-self-driving-vehicles-a-technological-arms-race/. [Accessed: 18-Feb-2021].

[3] Benewake Co. Ltd, “TF-Luna LiDAR Module” SJ-GU-TF-Luna A03 datasheet, Available: https://github.com/budryerson/TFLuna-I2C/blob/master/documents/TF-luna-A03%20Datasheet.pdf [Accessed: 18-Feb-2021]

[4] “Sustainability of Digital Formats: Planning for Library of Congress Collections,” LAS (LASer) File Format, Version 1.4, 30-Jan-2020. [Online]. Available: https://www.loc.gov/preservation/digital/formats/fdd/fdd000418.shtml#:~:text=LAS%20is%20a%20file%20format,radar%2C%20RAdio%20Detection%20and%20Ranging. [Accessed: 18-Feb-2021].

[5] Ieee.org. 2021. IEEE Code Of Ethics. [online] Available at: <https://www.ieee.org/about/corporate/governance/p7-8.html> [Accessed 17 February 2021].

[6] Ieee.org. 2021. IEEE Policies. [online] Available at: <https://www.ieee.org/content/dam/ieee-org/ieee/web/org/about/corporate/ieee-policies.pdf> [Accessed 17 February 2021].

[7] P. D. Alex Kilpatrick, IR Illumination and Eye Safety, 14-Sep-2017. [Online]. Available: https://medium.com/@alex.kilpatrick/ir-illumination-and-eye-safety-f0804673ca7. [Accessed: 17-Feb-2021].

[8] “Electric Shock at 5V and 40 A,” Electrical Engineering Stack Exchange, 01-Apr-1966. [Online]. Available: https://electronics.stackexchange.com/questions/307105/electric-shock-at-5v-

and-40-a#:~:text=3%20Answers&text=Your%205%20V%20is%20far,contact%2C%20you're%20OK. [Accessed: 17-Feb-2021].

[9] M.-le Li, Y.-hua Hu, N.-xiang Zhao, and Q.-shu Qian, “Modeling and analyzing point cloud generation in missile-borne LIDAR,” Defence Technology, 18-Oct-2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214914719308062#bib4. [Accessed: 17-Feb-2021].