

NELSON MANDELA
UNIVERSITY

Change the World

Faculty of Engineering, the Built Environment & IT

Technology for tomorrow

Department of Mechatronics

North Campus

Development of an indoor guidance system for an AGV

Mechatronics Project 4(EMP4110)

Compiled by: Mayur Ramjee 210050039

ii. Contents

iii. List of Figures	v
iv. List of Tables	viii
v. Synopsis	ix
vi. Nomenclature and Abbreviations	x
1. Introduction	1
1.1 Aim of Project	1
1.2 System Background and Significance	1
2. Literature Review	4
2.1 Methods of range measurement	4
2.1.1 LIDAR	4
2.1.2 Ultrasonic	5
2.1.3 Laser pointer and web camera	5
2.1.4 Microsoft Kinect (RGB-D Sensor)	6
2.2 Advancements in Robotic Navigation	7
2.2.1 SLAM	7
2.2.1 Hector SLAM	7
2.2.2 Landmark extraction	8
2.2.3 Extended Kalman Filter	8
2.2.4 RANSAC	8
2.3 Methods of Triangulation	9
2.3.1 Triangulation	9
2.3.2 Trilateration	9
2.4 Data processing to form a point cloud	9
2.4.1 Processing	9
2.4.2 ROS	10
3. Problem Analysis and System Requirements	11
3.1.1 System Functionality	12
3.1.2 Technical Performance Measures	12
4. Design Description	13
4.1 Description of the design development	13
4.1.1 Mechanical Design	13
4.1.2 Electrical System design	34

4.1.3 Information Technology design	52
4.1.3.3 Processing Sketch	60
4.1.4 System Integration	62
4.2 Design Specification	64
4.2.1 Component Specification	64
4.2.2 Device build methodology	67
5. Performance Evaluation Experiments and Data analysis	75
5.1.1 Design stage uncertainty of Instruments	75
5.1.2.1 Uncertainty of Garmin Lidar Lite v3	76
5.1.2.1 Uncertainty of Digital data acquisition	76
5.2.1 Experiment Method of identifying reflective material	77
References	84
Appendix A Working Drawings	87
Base Drawing	87
Appendix B Force Calculations	90
Appendix C Torque Calculation for actuation system	91
Appendix D Thyristor Calculations	93
Appendix E Spherical Coordinates Calculation	95
Appendix F Arduino Code	96
Appendix H Matlab Code	111
Appendix I Trilateration Code	112
Appendix J Specification sheets of equipment used	113

iii. List of Figures

Figure 1: Optical Line Navigation AGVS (2017)	1
Figure 2: Wire Navigation AGVS (2017).....	1
Figure 3 : Magnetic Tape Navigation AGVS (2017)	1
Figure 4:Fixed track AGV A-vt.be (2017)	3
Figure 5: Laser navigation SICK (2017)	3
Figure 6: Operation principle of rangefinder Todddanko (2017).....	6
Figure 7: Conceptual Design Rotational Aspect.....	14
Figure 8: Conceptual Design.....	14
Figure 9: Wanhao Duplicator i3.....	15
Figure 10 : Sensor mount.....	16
Figure 11: Mount FEA Loading	17
Figure 12 : Mount Stress analysis	17
Figure 13: Mount Strain analysis	18
Figure 14: Mount Displacement analysis.....	18
Figure 15: Bracket design	19
Figure 16: Loads applied to bracket	20
Figure 17: Moment about x axis	20
Figure 18: Moments applied to the system.....	20
Figure 19: Forces and Moments applied to bracket	21
Figure 20: Bracket Stress Analysis	21
Figure 21: Bracket Strain Analysis	22
Figure 22: Bracket Displacement Analysis	22
Figure 23: Base Loading Diagram.....	23
Figure 24: Forces and moment applied to Base.....	24
Figure 25: Base Stress Analysis	25
Figure 26: Base Strain Analysis	25
Figure 27:Base Displacement Analysis	26
Figure 28: Synchronous Pulley Design	28
Figure 29: Synchronous Pulley design	30
Figure 30 : Synchronous Pulleys and Belt.....	30
Figure 31: Designed Flange	31
Figure 32 Miniature Slip Ring	31

Figure 33: Slip Ring assembly.....	31
Figure 34: Final System Design	32
Figure 35: GEWISS Enclosure.....	33
Figure 36: Thyristor Characteristics.....	35
Figure 37: Single phase Thyristor converter circuit	36
Figure 38 : Thyristor Rectifier Circuit.....	37
Figure 39: Oscilloscope XSC2 output.....	37
Figure 40: Oscilloscope XSC1 output.....	38
Figure 41: 220v AC to DC module (Micro robotics)	38
Figure 42: LM7805	39
Figure 43: Buck Converter Simplified Circuit.....	40
Figure 44: Garmin Lidar lite v3 Robotshop (2017)	43
Figure 45: SICK NAV-350 Sensortrade (2017)	43
Figure 46: HCSR04 sensor MACFOS (2017).....	43
Figure 47: Kinect sensor Engadget (2017).....	44
Figure 48: Laser webcam Codeproject (2017)	44
Figure 49: I2C operation.....	45
Figure 50: SDA and SCL Operations	46
Figure 51: Proposed Sensor Operation.....	46
Figure 52: Stepper motor windings	47
Figure 53: Stepper Motor Control.....	48
Figure 54: Output Gear Home position.....	49
Figure 55: LDR.....	49
Figure 56: Servo Motor.....	49
Figure 57: PWM for a Servo Motor.....	50
Figure 58: Power Supply Summary.....	51
Figure 59: Main Flow Diagram	53
Figure 60: 2-D Mode	54
Figure 61: 3-D Mode	55
Figure 62: Spherical Co-Ordinates.....	57
Figure 63: Spherical Trilateration	58
Figure 64: 2-D Trilateration	59
Figure 65: Program Flow.....	61

Figure 66: System Integration	63
Figure 68: ABS base	67
Figure 69: Example of a X, Y Mab produced by MATLAB.....	72
Figure 70: Example of point cloud output by the system	72
Figure 71: Final Designed Device	73
Figure 72: Device Final Design	73
Figure 73: Sensor Test at different lengths	77
Figure 74:Sensor Baseline experiment	77
Figure 75: Reflective Material Calibration.....	78

iv. List of Tables

Table 1 : Mount FEA details	17
Table 2 : Mount FEA results	17
Table 3: Bracket FEA details	21
Table 4: Bracket FEA results.....	21
Table 5: Base FEA details	24
Table 6: Base FEA results.....	25
Table 7 : Battery vs Power Supply	34
Table 8 : Voltage Regulator Vs Buck Converter	41
Table 9: Device Properties	43

v. Synopsis

The following report is focused on the development of an Indoor guidance system for an AGV (Automated Guided Vehicle). AGVs are used for industrial applications for moving materials around warehouses without the need for human interaction.

The necessity for this system within industry was investigated, with emphasis of existing systems and their advancements/shortfalls determined via a literature survey. Upon the analysis of the problem at hand, TPMs (Technical Performance Measures) are determined. TPMs would then dictate the requirements of the system which were classified into mechanical, electrical, control system and I.T (Information Technology) requirements. These requirements were used as the basis of a design stage system architecture which provided solutions to each TPM.

From the design stage architecture, a concept for the navigation system was developed. The operations of many range finding devices were compared prior to selection for the system. Once a device had been chosen CAD drawings were then developed. These drawings integrated as many readily available parts as possible with only mountings and plates to be machined/3-D printed. Once the CAD drawings were completed a theoretical background of the system was investigated.

The design stage functionality of this system would be tested via theoretical background calculations. These calculations entailed the movement of the range finding device relative to its refresh rate and applied loads to the motor.

A prototype system was built that could meet the TPMs set out by the student with emphasis being put on possible future improvement's that could be applied. The subsequent built system could map out its surroundings but was not able to localise itself in its surroundings.

vi. Nomenclature and Abbreviations

TPM – Technical Performance Measures

CAD – Computer Aided Drawings

AGV – Automated Guided Vehicle

LIDAR – Light Detection and Ranging

TOF – Time of Flight

SLAM – Simultaneous Localization and Mapping

EKF – Extended Kalman Filter

RANSAC – Random Sampling Consensus

UAV - Unmanned Aerial Vehicles

MCU – Microcontroller

PWM – Pulse Width Modulation

FEA – Finite Element Analysis

1. Introduction

1.1 Aim of Project

The aim of the following project is to design and build a cost-effective navigation system for an Automated Guided Vehicle (AGV). This system should be a viable alternative to the fixed track method of navigation by employing the usage of efficient triangulation and trilateration methods based on land marks and laser ranging.

1.2 System Background and Significance

Currently in the interest of competition, production and storage industries are moving towards a new standard called Industry 4.0. This forth industrial “revolution” entails systems and machines connecting digitally with an increase in predictability, reproducibility, and flexibility for activities. A major part of this standard involves the use AGVs defined as “a programable mobile vehicle for goods handling and transport” by Kumar Das (2016). AGVs are multifunctional vehicles with the ability to be modified towards their own specific task. Currently companies are already moving towards the deployment of AGVs within their facilities, whereas this is regarded as relatively new technology in South Africa. Most AGVs currently in use within South Africa are guided throughout facilities on a fixed track basis. The fixed guidance system is that of using systems such as that of inertial (odometry), optical line following (figure1), magnetic tape (figure3), magnetic grid, or wire navigation (figure 2) seen below.

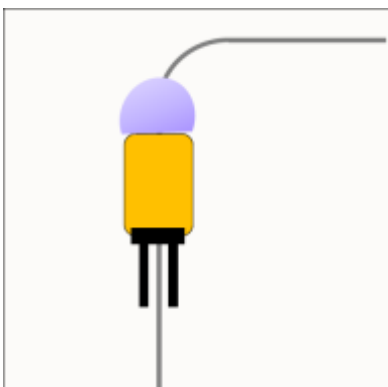


Figure 1: Optical Line Navigation AGVS (2017)

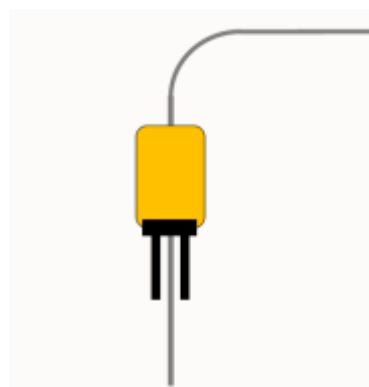


Figure 3 : Magnetic Tape Navigation AGVS (2017)

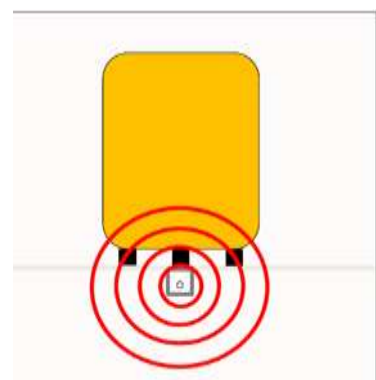


Figure 2: Wire Navigation AGVS (2017)

Optical Line Navigation is the method of navigating an AGV by means of following a black line set out on a contrasting floor as a track. The AGV has an optical sensor mounted on the front that is able to emit a signal and determine whether the system itself is following the line via the signal received. Kumar (2012,26) states that in application, the use of three or more IR (Infra-red) sensors are common with a corresponding instruction sent to a MCU to control the AGV's motors. This system is dependent on the quality of signal received from track laid down thus could be susceptible to problems such as an obscured track.

Magnetic tape systems operate similarly to that of optical systems however instead of using a black line, magnetic tape is set out as a track. Wu (2011) states that a magnetic sensor is able to detect the intensity of the magnetic field of the tape. The deviations of the AGV and the central line of the magnetic tape can then be controlled by adjusting velocities of the controlling wheels. RFID tags can send information to the AGV for specific operations. This system can only follow the field generated from this magnetic tape track, thus maintenance of this track will be a necessity.

Wire navigation systems are one of the oldest forms of navigation with in the production industry and use inductive wires embedded into the floor of the facility. These wires will have an electric current flowing through at an appropriate frequency, which causes a magnetic field to be detected by guide path sensors aboard the AGV. These systems are accurate but are prone to interference caused by metal used in the floor of a facility, Proxaut (2017). The greatest disadvantages of this method of navigation would be the intrusive nature of laying a track down and the need for power supplied to the wire.

In the interests of flexibility, the use of multiple AGVs, concerns of collision and obstacle avoidance, the integration of fixed track systems are becoming obsolete. Currently systems that employ the technique of fixed track navigation are limited by their designated route seen in figure 4. The route set-out for the AGV will dictate its speed, efficiency and reach whereas the use of a non-fixed track system would overcome these limitations.



Figure 4: Fixed track AGV A-vt.be (2017)

The latest technique of indoor navigation is that of Laser navigation seen in figure 5 and triangulation. This process involves the use of a signal emitter and receiver combination that can triangulate the AGV's position based on relative distance to landmarks. The added advantages of the non-fixed track system would be that of multiple AGV deployment whereby many sub system AGVs could communicate with one another with information related to route, speed, and current location.

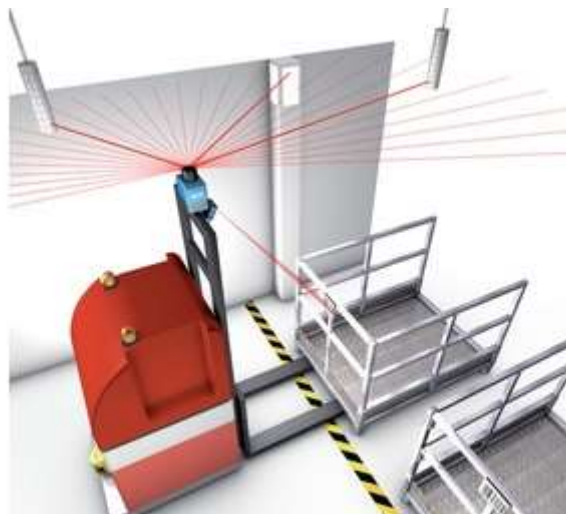


Figure 5: Laser navigation SICK (2017)

2.Literature Review

For a better understanding of these requirements and existing systems, an in-depth literature review of each concept was conducted. The concepts deemed essential for the student to better understand the project in its entirety are as follows:

- Methods of range measurement.
- Current advancements within robotic navigation.
- Methods of triangulation.
- Processing of data to form a point cloud.

2.1 Methods of range measurement

To understand the location of the AGV whilst it is in motion a device that can interpret range is necessary.

2.1.1 LIDAR

Light Detection and Ranging (LIDAR) is used to describe the technique of using light for determining the distance to an object. LIDAR is a remote sensing methodology which is similar to radar but uses laser light pulses instead of radio waves.

International LIDAR Mapping Forum (2008) states that LIDAR can acquire data to produce accurate digital evaluation models by allowing positioning of a laser beam as it hits an object. LIDAR follows the principles of Time of Flight (TOF), which states that the beam of a laser is traveling at the speed of light thus the distance to an object can be calculated by the following equation.

$$Distance\ to\ object = \frac{(Time\ taken\ for\ pulse\ return \times c)}{2}$$

c is the speed of light (3×10^8 m/s).

LIDAR sensors are used in a wide variety of applications such as micro-topography, mapping, agriculture and, more recently, in vehicle automation. The use of LIDAR for range measurement is a potential solution for this project based on previous and current applications.

2.1.2 Ultrasonic

Ultrasonic sensors use the properties of ultrasound as a specific frequency for operation. These sensors comprise of two main categories piezoelectric and electromagnetic, with piezoelectric being the more cost-effective option.

Carullo and Parvis, (2001) state that ultrasonic sensors are reasonably cheap and work for ranges of up to a few metres. Problems do arise when working in open air noisy conditions. Ultrasonic sensors emit an Ultrasonic sound wave and measure the time taken for an echo to be received. These sensors work on the same TOF principles as that of LIDAR but, in comparison, use the speed of sound to calculate distance away from an object. However, Yanbina, Hong and Xue, (2012) states that the temperature of **the** air will influence the speed of sound thus this is a key factor to the accuracy of measurements when using Ultrasonic sensors.

The use of ultrasonic sensors will be compared to other options, the susceptibility of operating conditions regarding noise and temperature will be considered.

2.1.3 Laser pointer and web camera

An inexpensive option for designing a basic rangefinder would be that of using a combination of a laser pointer and a web camera. Shojaeipour et al., (2010) state that using a simple setup of a laser pointer mounted a known distance from a web camera allows for an algorithm to be used to determine the distance away from the web camera itself.

This algorithm identifies the brightest pixel seen in the web camera image and identifies it as the Laser pointer. Using trigonometry, the distance from the centre pixel of the web camera to that of the laser pointer allows for a distance estimate to be made as demonstrated in figure 6.

This technique may be applied with the following formula:

$$D = \frac{h}{\tan\theta}$$

$$\theta = pfc \times rpc \times ro.$$

pfc = number of pixels from the focal plane.

rpc = the radians per pixel pitch

ro = the radian offset (compensates for alignment errors)

h = the height at which the laser pointer is away from the camera system.

the above may be seen in figure 6.

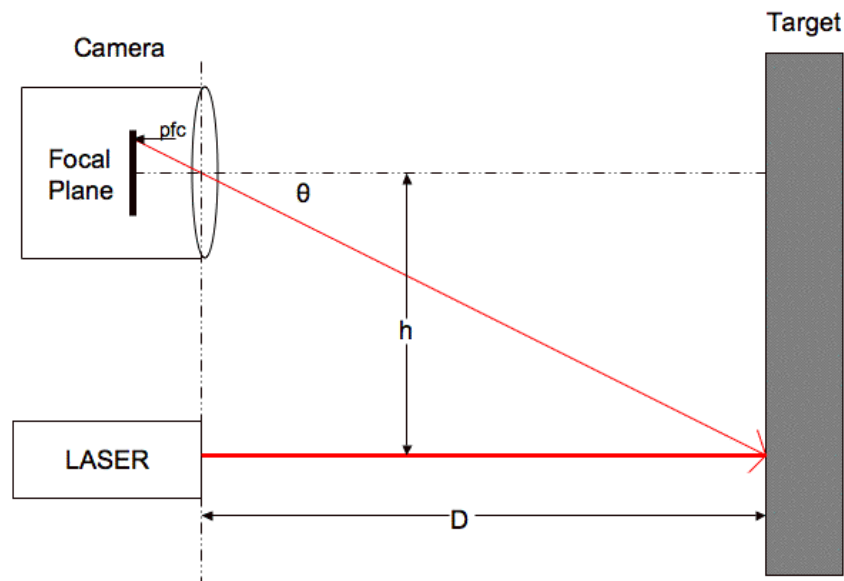


Figure 6: Operation principle of rangefinder Todddanko (2017)

This system can be seen to be very inexpensive but limitations include that of an initial calibration which is always necessary for a baseline and that system performance is dependent on the resolution of the webcam used. Low light situations may produce inaccurate results, as the webcam would be unable to find the laser pixel.

2.1.4 Microsoft Kinect (RGB-D Sensor)

The Microsoft Kinect was launched initially as a vision-based controller for the Microsoft Xbox 360 video game console. It consists of several sensors including a RGB sensor, a 3D depth sensor and an array of microphones and accelerometers. The Kinects depth sensor is deemed useful due to its ability to produce depth images at a resolution of 640 x 480 pixels or VGA size. Kamarudin et al., (2014) state that the sensor has a field of view of 57° horizontally and 43° vertically. The Kinect's depth sensor is able to provide a resolution of 1cm with a maximum stable transfer frame rate stated to be up to 30Hz.

The Kinect has been the subject of many scientific papers such as that of it's use in Unmanned Aerial Vehicles (UAVs) as the depth sensor may be used as an altitude

sensor, Christopher M. (2011). According to Zhi-Ling Hu (2011) the Kinect has been used in the medical field with the use of information from the depth sensor relating to tracking limb movement and prosthetics. Additionally, the fact that the Kinect sensor is relatively affordable makes it very popular in the robotic research community Schmidt et al., (2013).

2.2 Advancements in Robotic Navigation

2.2.1 SLAM

Simultaneous Localization and Mapping (SLAM) is a term which has been commonly used in recent years as it seems to contain answers for problems such as that of autonomous and automatic vehicles. SLAM allows for the construction or updating of a map whilst simultaneously understanding a device's current location in it. Since the system intended to be built will have its own set of calibrated landmarks some of the relevant techniques from SLAM will be utilised. Riisgaard & Blas (2005) state that SLAM is made up of the following components viz; Landmark extraction, data association, state estimation, state update and landmark update.

The ability of to understand surroundings from apparent motion and imaging from on-board cameras is a prerequisite for truly autonomous vehicles. The need for simultaneous localization and accurate mapping is necessary in GPS-denied environments such as that of Martian and Lunar surfaces. Vision-based SLAM has been successfully applied to planetary exploration missions, such as NASA's Mars Exploration Rover 2003 (MER) mission and China's Chang'E-3 mission Di et al., (2016).

2.2.1 Hector SLAM

Hector SLAM is an implementation SLAM that does not require wheel odometry information. Orientation of a system that employs this technique relies solely on information from scan matching, Kamarudin et al., (2014). Scan matching involves an initial scan that is used as a baseline for the subject's movement in its environment. From a change of orientation and distance the position of the subject is estimated in real time. Since this system only has input from its surroundings and not of the subject's movement, a High rate of scanning is necessary.

2.2.2 Landmark extraction

Merriam-webster (2017) defines a Landmark as a conspicuous object on land that marks a locality. Landmark extraction is a process where as a robot moves within its environment it can keep track of where it is on the way to its destination. This is achieved by a robot associating new landmarks and comparing them to previous ones. Landmarks that have not been observed are added to an Extended Kalman Filter (EKF).

2.2.3 Extended Kalman Filter

The Extended Kalman Filter or EKF is an algorithm used to estimate information for the locations current state from previous states measured. Khan et al., (2014) state that the EKF has the capability to work in non-linear systems and that its working is based on the linearization model around the mean of its current state. An advantage to the EKF is that it has a small computational complexity, thus, it requires low memory when compared to other algorithms therefore it is well suited for low powered mobile devices. Using an EKF for efficient determination of landmarks passed and their current orientation with respect to the system, a system must be used to recognise straight segments - this is determined via RANSAC.

2.2.4 RANSAC

RANSAC (Random Sampling Consensus) is a method which can be used to extract lines from a laser scan. These lines can in turn be used as landmarks. In indoor environments, straight lines are often observed by laser scans as these are characteristic of straight walls in most cases. Khan et al., 2014, states that RANSAC finds these line landmarks by randomly taking a sample of the laser readings and then using a least squares approximation to find the best fit line that runs through these readings. Once this is done RANSAC checks how many laser readings lie close to this best fit line. If the number is above some threshold we can safely assume that we have seen a line (and thus seen a wall segment). This threshold is called *the consensus*.

Methods that utilize algorithms of both the EKF and RANSAC methodology will be investigated by the student as to increase the efficiency of the system to be built. Systems which can apply these methodologies must be built upon, and set up, for programming capabilities of the student.

2.3 Methods of Triangulation

2.3.1 Triangulation

Triangulation is a method of surveying based upon the trigonometric proposition that if one side and three angles of a triangle are known the remaining sides may be computed ("Fundamentals of Mapping", 2017). In addition if one of the directions of the triangle's sides are known the others may be determined. For a triangulation method to work effectively the workspace or navigational space must be envisioned as a series of joined or overlapping triangles. A known side of one of the triangles must be known to determine the characteristics of other triangles such as that of angles measured at vertices.

2.3.2 Trilateration

Trilateration is a process used by GPS receivers to determine their relative position on the surface of the earth (Mio.com). Position is calculated via a process of timing signals from three satellites that form part of the Global Positioning System. Each of the satellites that forms part of the GPS constellation transmits a periodic signal along with a time signal, the receiver module receives these signals. The receiver is then able to determine its own distance away from the 3 satellites and from using these three distances as radii form 3 spheres each centred at the satellites positions. Algebraically the point where these three spheres intersect can be determined which results in the relative position of the receiver.

("Fundamentals of Mapping", 2017), state that Trilateration has become a feasible with the development of EDM or Electronic Distance Measurement equipment as measurements of all lengths with a high order of accuracy under almost any field conditions.

The student would need to investigate the implementation of trilateration for navigation. A prerequisite to successfully employ trilateration is that of a minimum of three known landmarks in the subjects' environment.

2.4 Data processing to form a point cloud

2.4.1 Processing

Processing is a flexible open source programming IDE (Integrated Development Environment) that may be used to "sketch" 2D or 3D graphics (Processing.org, 2017). The processing development environment (PDE) contains a simple text

editor, with a run button to execute sketch files, capabilities of these sketch files are dependent on the libraries and tools loaded. Libraries are being constantly updated by the Processing community with abilities from playing simple audio sounds to complex computer vision.

The PDE is highly configurable with options from the preferred operating system to that of programming language selection. Most resources available such as that of examples and libraries are programmed in Java.

2.4.2 ROS

The mathematics and programming necessary for these applications of systems such as the EKF or RANSAC can be seen to be extremely time consuming and labour intensive to implement. Since these issues could become problematic due to the amount of data updated from the sensor, a system called ROS or (Robotic Operating System) was investigated. (Quigley et al.,2009) states that ROS is an open source robot operating system which has many benefits to both commercial and non-commercial applications. ROS is multi lingual thus supporting many languages which include C++, Python, Octave, and LISP, thus allowing for more platforms to use this system in a broader sense. Further-more (Quigley et al.,2009) states that ROS can re-use code from numerous open source software applications of which SLAM is included.

ROS is open source and free, which allows for developments to be shared and allow for rapid prototyping of systems in a much quicker fashion since existing systems do not need to be recoded.

3. Problem Analysis and System Requirements

The construction of a point cloud and localization of an AGV within its environment requires information of the surroundings, such as distances from objects and distinct landmarks. The student would need to identify an appropriate laser ranging sensor to receive information about the device's surroundings. The laser ranging sensor would need to be portable, have a rapid update rate and be able to be moved in 2 axes as to gather information of features at varying levels of the devices' surroundings. Data gathered from the laser ranging sensor would need to be processed to build a point cloud that contains information of surroundings and landmarks.

To localize via SLAM techniques the student would need to employ a Hector SLAM program to analyse data, since odometry from a AGV will not be available. From research conducted the student had found that methods that employ SLAM are highly dependent on both processing power and that of dynamic memory. The demands required for this operation were deemed to be high for the use of a MCU (microprocessor) such as an Arduino, to compute, thus information would be sent to a computer in real time to perform the necessary processing.

Initially the student had proposed that information could be processed on a computer system using ROS which has SLAM functionalities inbuilt. After further research into the use of ROS the student had found that ROS is incompatible with Windows and would need to be implemented via a LINUX operating system. Due to the students in-experience with using command-line LINUX, an alternative option was necessary to localize the sensing system.

The final proposed system the student had decided to prototype would use the program "Processing" to output the display of information from the sensing system in real time with localization not being done simultaneously, but rather post-scanning. Localization would be done via analysis of feedback from the laser ranging sensor namely that of determining relative distances from reflective material.

After the relative distances were computed the student had determined that the use of trilateration in 2 axes would allow for rapid localization if specific Landmarks could be identified.

3.1.1 System Functionality

The required system functionality would be the following:

- Utilize an appropriate Laser ranging sensor.
- Rotate along the z-axis to gather information of features within surroundings at the same level as the sensor.
- Rotate along the x-axis to gather information of the features within the surroundings at a different altitude to that of the sensor.
- Send data from the sensor to a computer application to build the Point-cloud.
- Perform localization post-scanning by determining relative distances away from predetermined Land-marks identified.
- Apply trilateration to identify X and Y Co-ordinates of the device.
- Be stable as to not skew the recorded data found from the sensor.
- Project build should stay within the R5 000 allocated budget
- Have appropriate safety measures.

3.1.2 Technical Performance Measures

Technical Performance of this system would be rated on the following factors:

1. Lightweight design – The system should be lightweight as it is to be fitted onto and AGV. An overweight system may infringe on the performance of the AGV thus it must be kept as compact and size appropriate as possible.
2. Accurate – The system should be accurate such that its operation within a facility would not lead to collision or damage.
3. Cost effective – The system should be relatively cheaper than that of industrial systems currently in use, this will be highlighted in the component specifications.
4. Efficient – Localization of the system should occur as fast as possible as the system will be moving simultaneously.
5. Point cloud display – The point cloud generated of which will comprise of points scanned by the system. From the point cloud a map will be generated in the appropriate planes of operation.
6. Safe and Ergonomic design – The design itself should not leave components exposed, as contact with the sensor whilst the AGV is mobile could be

hazardous. The system should be safe to operate and be able to power down in the event of an emergency.

4. Design Description

4.1 Description of the design development

Within the following section the description of the final design will be discussed in detail. These descriptions will be separated into sub-systems and their integration into a single working device will be explained.

The sub-systems designs include that of:

- Mechanical design
- Electrical Design
- IT design (Information Technology inclusive of microcontroller programming)

4.1.1 Mechanical Design

To produce a system that can perform navigation and localization both accurately and rapidly, stability is a necessity. The system will need to maintain stability whilst rotating about two axes. During the system operation the rotation of the laser ranging sensor should not be interrupted. Thus, the deliverables decided upon for the mechanical design would be the following:

- Rotation about two axes.
- Stability provided to the Laser ranging sensor.
- Uninterrupted communication between the Laser ranging sensor and the microcontroller during all operations.

To design for these deliverables, the mechanical system would first need to be conceptually designed such that it is able to perform in the correct manner theoretically.

4.1.1.1 Mechanical Concept

Within this concept it can be seen in figures 7 and 8 that the sensor chosen for operation has been selected to be a Garmin Lidar Lite v3. The reasoning behind this choice will be explained within section 4.1.2.4 The selection of this sensor proved to be vital to the mechanical design of the system thus the concept was generated around it.

The concept can be seen to contain the following:

- A NEMA 17 stepper motor for rotation about the z axis.
- A Servo motor for rotation about the x axis.
- A Slipring used to transmit the signal feedback from the Laser ranging sensor.
- A base used to mount all components.
- A bracket used to transfer the movement from the stepper motor to that of the sensor.
- A mount used to secure the sensor and transfer the movement from the servo motor to that of the sensor.
- Two synchronous pulleys with a 1:1 ratio driven by a single synchronous timing belt.
- System is secured with M4 and M2 screws and nuts.



Figure 8: Conceptual Design

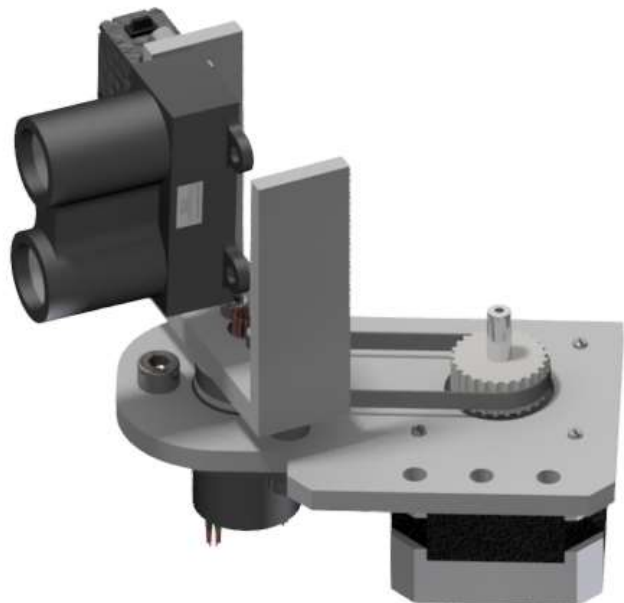


Figure 7: Conceptual Design Rotational Aspect

This initial concept would thus serve as the basis of the student's design with alterations made for availability of parts, real world constraints, and engineering principles.

4.1.1.2 Part Production

Parts would be 3-D printed as to save both weight and time needed to prototype the system. The 3-D printers used for this design would be a combination of one provided by the Mechatronics department namely that of the Wanhao Duplicator i3 seen in figure 9 for student use and that of one in ENTSA's facilities namely that of the Wanhao duplicator 6 dependent on availability.



Figure 9: Wanhao Duplicator i3

The use of a 3-D printer for part manufacturing was deemed as a viable solution to the student as parts to be manufactured are relatively small since the conceptual base dimensions were (100 x 50 x 6). The loads applied to the system would be very small mainly being the weight of the sensor and mounting system thus the risk of delamination would be highly decreased. Delamination is the process of layers of a 3-D printed part separating itself and is dependent not only the application but the process of part printing. (Guy ,2017) states that delamination can be the result of the type of printing filament used such as ABS, the temperature of the print , the speed of the print to the limitation of the polymers used and the temperature of the room in which the printing occurs

4.1.1.2 Mount design

In the design process of a bracket to transmit motion to the sensor the student would need to initially produce a mount to secure the laser ranging sensor securely. The mount needed for the sensor would need to do the following:

- Stabilise the laser ranging sensor during movement.
- Allow for motion to be transmitted via the servo motor.
- Be fitted onto the existing servo horn such as to allow for precise meshing with the servo gear.

The mount would need to sustain the torque applied from the servo motor as well as not deform under the weight of the sensor.

From these requirements a part was designed - this can be seen in figure 10. In the design an appropriate indentation is formed to secure the horn of the servo motor. Mounting to the ranging sensor would be done via 4 M4 screws and nuts.

The weights applied to this mount would fully depend on the laser-ranging sensor. This weight was found to be 22g. Maximum torque that would be applied would be dependent on the servo motor. Applied torque would be that of the stall torque from the selected motor which is 2.7kg.cm. The mounting would be made from PLA (Polylactic Acid), a 3-D printer filament.



Figure 10 : Sensor mount

Using the formula:

$$F = mg$$

The subsequent forces applied to the part could be calculated, may be found in Appendix B.

Using this information of material specification and that of applied force/torque an FEA was conducted.

Table 1 : Mount FEA details

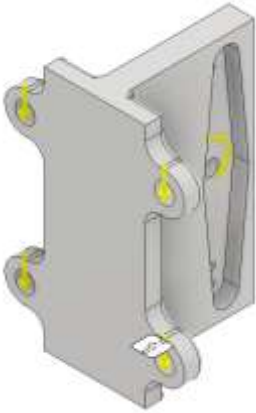
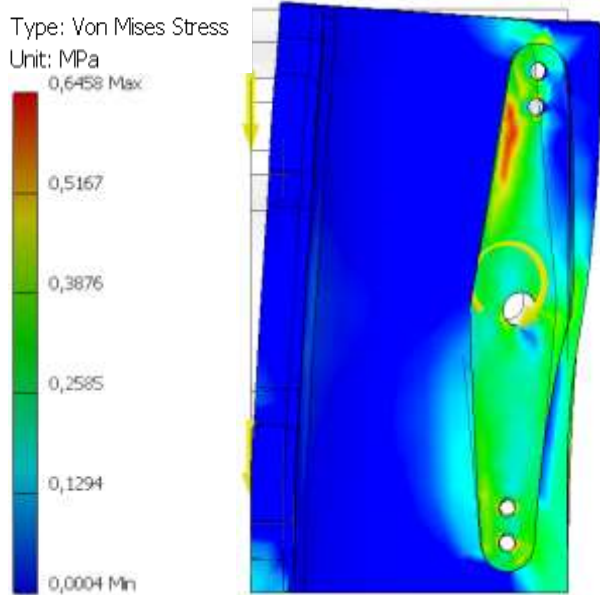
Mount FEA	Details
 <p>Figure 11: Mount FEA Loading</p>	<ul style="list-style-type: none"> • Fixture type: Fixed • Material: PLA • Density: 1400 kg/m³ • Yield Strength: 3e+007 N/m² • Tensile Strength: 6.1e+007 N/m²

Table 2 : Mount FEA results

FEA analysis	Type	Max	Min
Stress analysis	Von Mises	0.6458 MPa	0.0004 MPa
 <p>Type: Von Mises Stress Unit: MPa 0,6458 Max 0,5167 0,3876 0,2585 0,1294 0,0004 Min</p> <p>Figure 12 : Mount Stress analysis</p>			

FEA analysis	Type	Max	Min
Strain Analysis	Equivalent Strain	1.461e-07 ul	2.707 ul

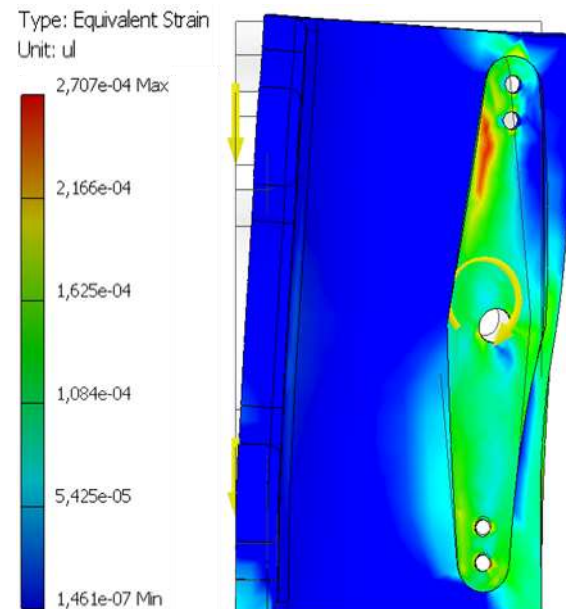


Figure 13: Mount Strain analysis

FEA analysis	Type	Max	Min
Displacement	Total Displacement	0.01184 mm	0 mm

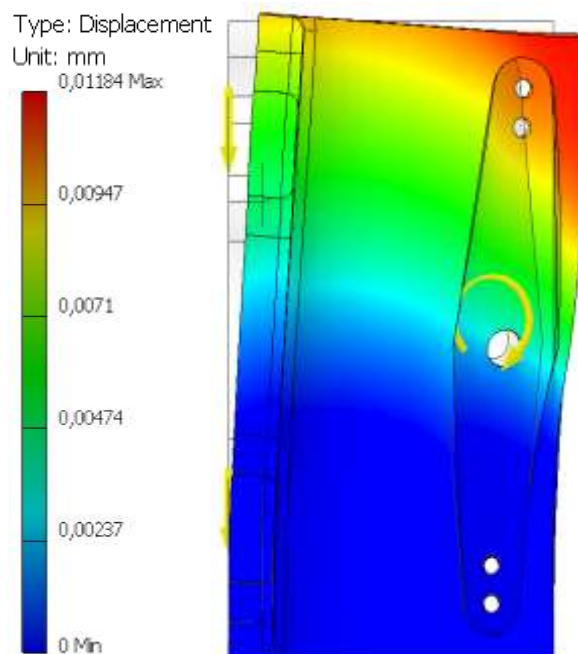


Figure 14: Mount Displacement analysis

Design Factor of Safety : 15

4.1.1.3 Bracket Design

A bracket used to mount the Laser ranging sensor and the servo motor would need to be designed and produced. The outcome of the bracket design should ensure the following:

- Secure mounting points for the transfer of movement **from** the servo motor to the mount with the laser ranging sensor attached.
- Stability under loads applied during the movement of the system.
- Consideration of attachment to a synchronous pulley.

An appropriate design was generated which allowed for the bracket to be attached to a synchronous pulley seen in figure. Accommodation for the attachment of a servo motor was made such that 0° to 90° actuation of the sensor would be allowed.

The forces applied to this bracket would be that of the weight of both the servo motor, the mounting bracket, and that of the laser ranging sensor seen in figure 16.

Mass of the mount was found to be 10g, mass of the servo motor was found to be 16g with the assumption of mass of each screw and nut to be approximately 4 grams. Using these weights applied to the equivalent forces could be applied to the part with the subsequent moments resulting about both the y axis and x axis seen in figure 17. Eccentricity from the centre of the bracket would determine the size of the moment applied to either axis of the bracket.



Figure 15: Bracket design

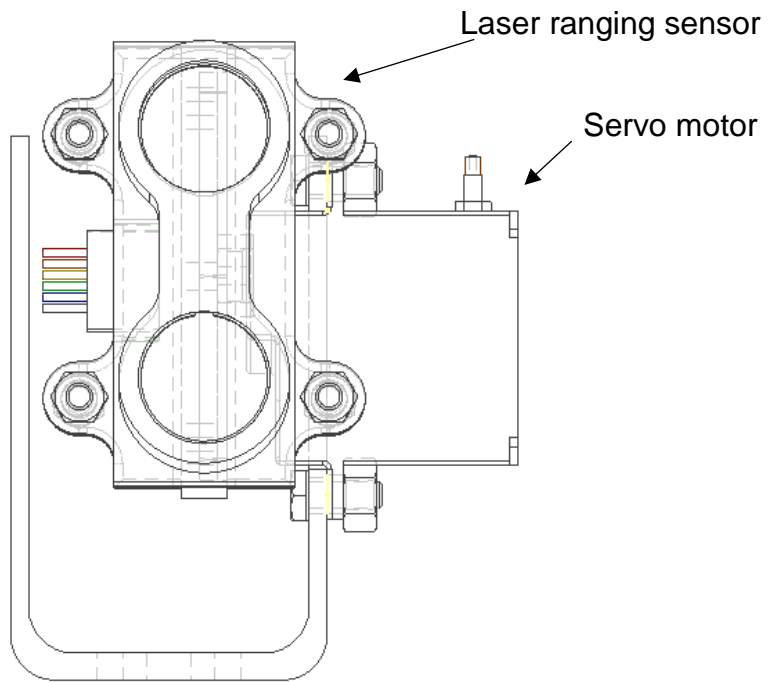


Figure 16: Loads applied to bracket

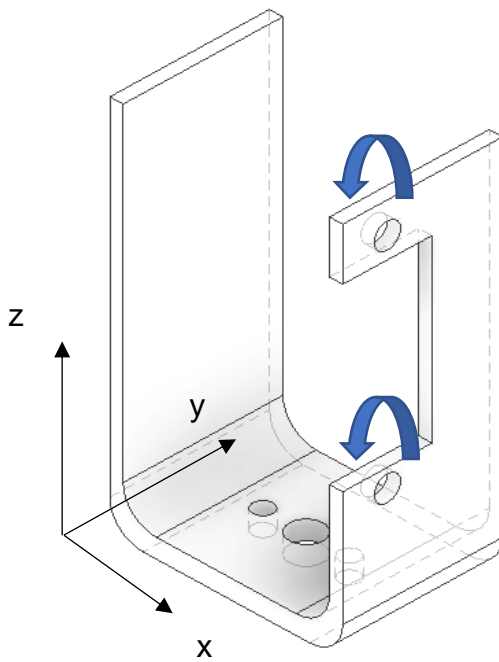


Figure 18: Moments applied to the system

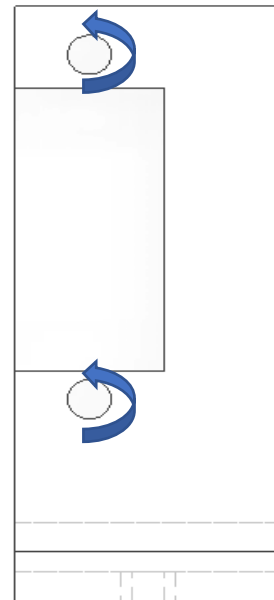


Figure 17: Moment about x axis

The resultant moment about the y axis was found to be 10.045 N.mm and that about the x axis to be 4.3948 N.mm seen in Appendix B. The bracket would be 3-D printed from the printing filament PLA. These forces and moments were applied to the part and an FEA was then generated.

Table 3: Bracket FEA details

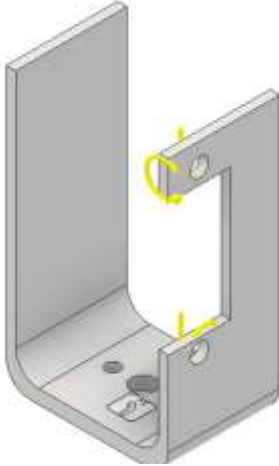
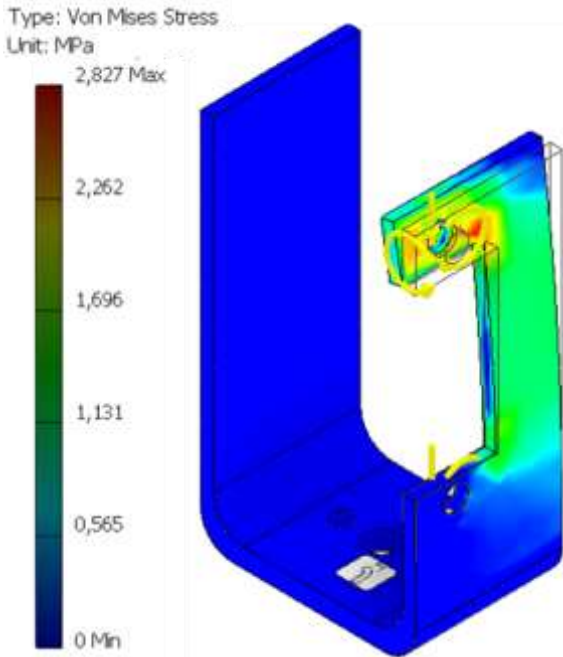
Mount FEA	Details
 <p>Figure 19: Forces and Moments applied to bracket</p>	<ul style="list-style-type: none"> • Fixture type: Fixed • Material: PLA • Density: 1400 kg/m³ • Yield Strength: 3e+007 N/m² • Tensile Strength: 6.1e+007 N/m²

Table 4: Bracket FEA results

FEA analysis	Type	Max	Min
Stress analysis	Von Mises	2.827 MPa	0 MPa
 <p>Type: Von Mises Stress Unit: MPa 2,827 Max 2,262 1,696 1,131 0,565 0 Min</p> <p>Figure 20: Bracket Stress Analysis</p>			

FEA analysis	Type	Max	Min
Strain Analysis	Equivalent Strain	0.001172 ul	0 ul

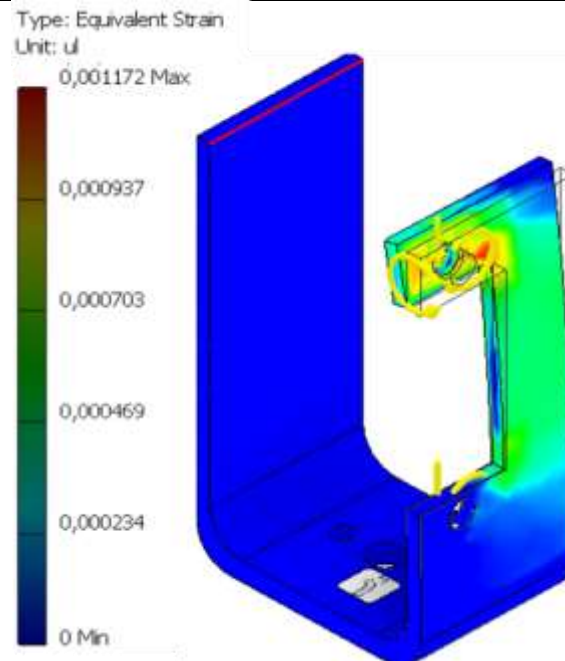


Figure 21: Bracket Strain Analysis

FEA analysis	Type	Max	Min
Displacement	Total Displacement	0.8123 mm	0 mm

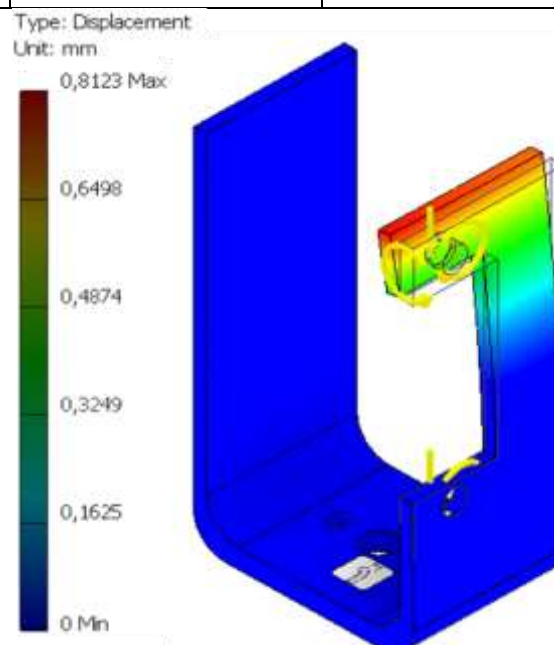


Figure 22: Bracket Displacement Analysis

Design Factor of Safety : 10

4.1.1.3 Base Design

To maintain stability to all components during operation of the system a base would need to be designed. The base would serve as the main mounting point for the slipping, stepper motor and the driven pulley. In addition to the weights of components being exerted on the base torque from the stepper motor operation must be considered seen in figure 23

The following considerations were to be made during the design process of the base:

- Stability for all components during operation.
- Provision for mounting on an enclosure.
- Provision for input buttons for the system.
- A “tensioning” system for the synchronous belt system.
- Indentations where possible to allow for the protrusion of the slip ring rotor for synchronous pulley mounting.

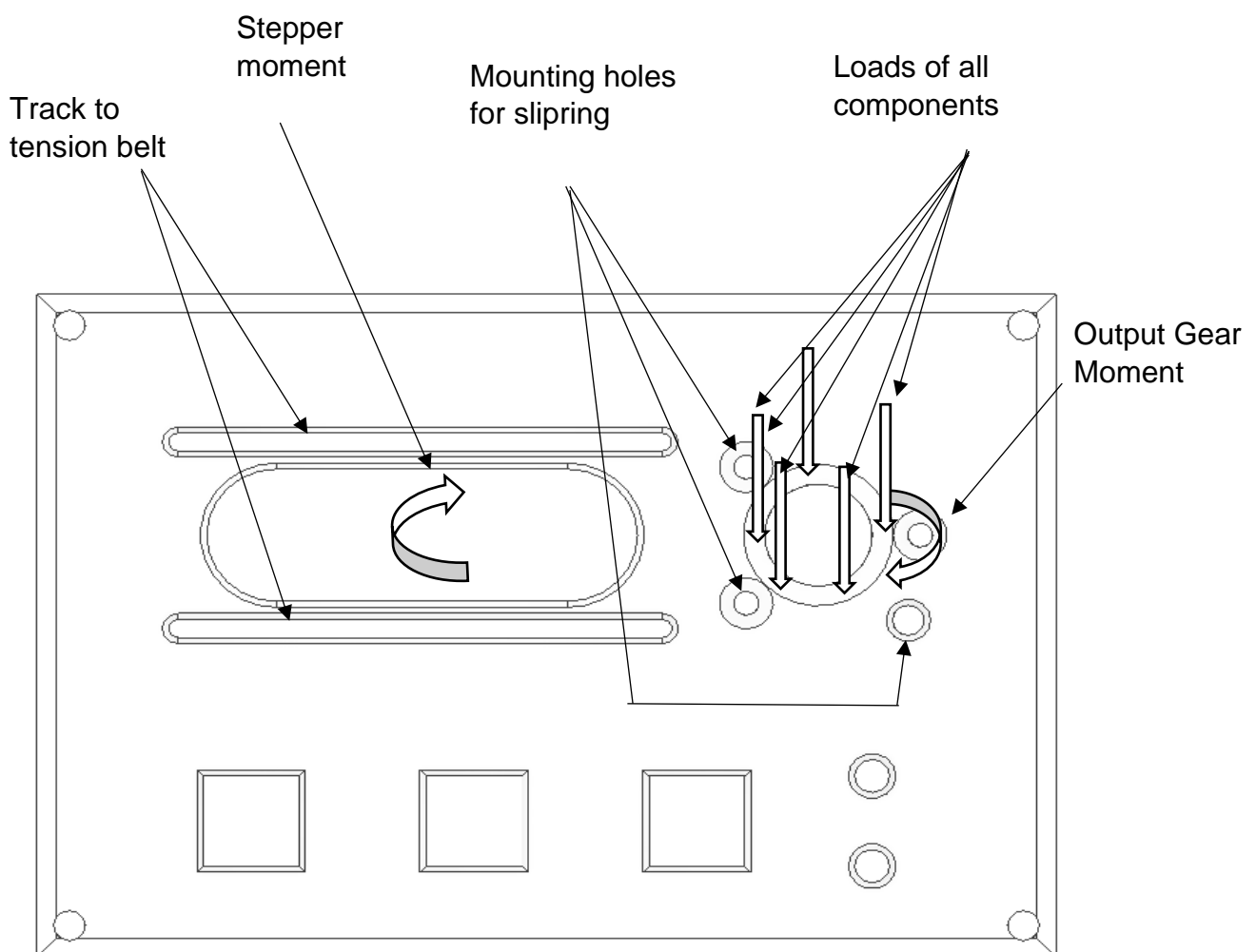


Figure 23: Base Loading Diagram

Due to the unavailability of PLA filament the base would be printed from ABS (Acrylonitrile Butadiene Styrene). The masses of components acting upon the base to be considered included the following:

- Mass of sensor = 22g
- Mass of sensor mount = 10g
- Mass of bracket = 12g
- Mass of input pulley = 20g
- Mass of output pulley = 14g
- Mass of fasteners = 24g
- Mass of the Stepper motor = 200g

Moment applied to the base about the z axis would be the holding torque of the stepper motor (2800g.cm). The Resultant torque developed with the inclusion of the 1:3 ratio output gear would be added on to the output side of the base seen in Appendix B. With the use of these details a FEA was setup to analyse the base behaviour under loading.

Table 5: Base FEA details

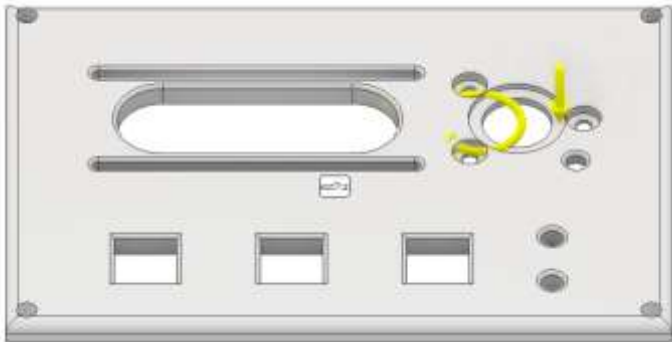
Base FEA	Details
 <p>Figure 24: Forces and moment applied to Base</p>	<ul style="list-style-type: none"> • Fixture type: Fixed • Material: ABS • Density: 1080 kg/m³ • Yield Strength: 3.4e+009 N/m² • Tensile Strength: 7e+006 N/m²

Table 6: Base FEA results

FEA analysis	Type	Max	Min
Stress analysis	Von Mises	3.882 MPa	0 MPa

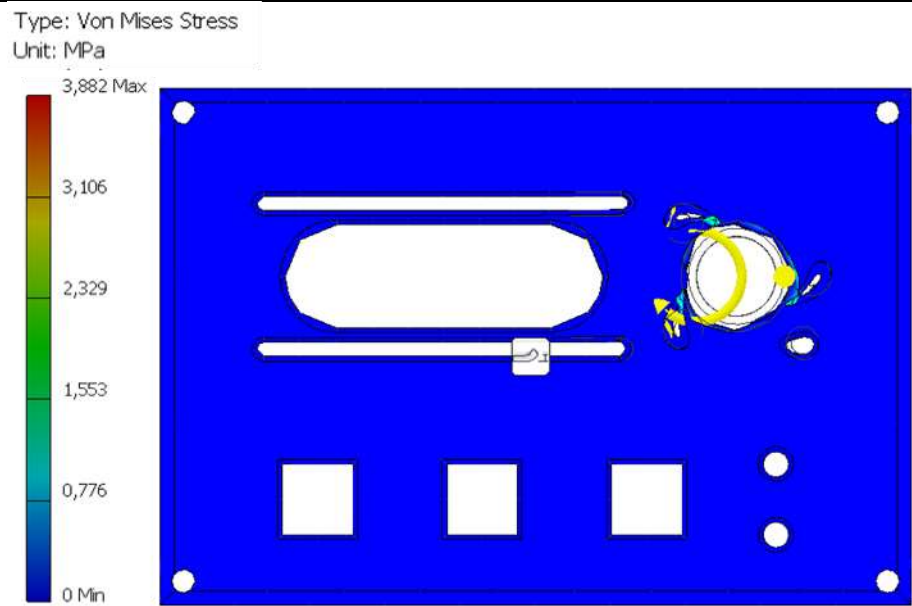


Figure 25: Base Stress Analysis

FEA analysis	Type	Max	Min
Strain analysis	Equivalent Strain	0.001594 ul	0 ul

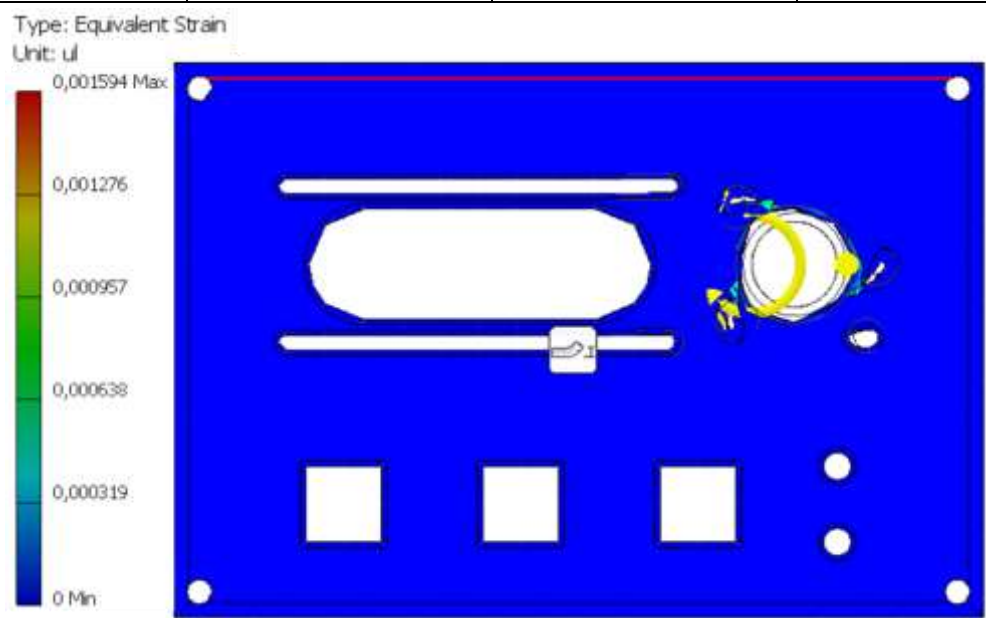


Figure 26: Base Strain Analysis

FEA analysis	Type	Max	Min
Displacement	Total Displacement	0.005091 mm	0 mm

Type: Displacement
Unit: mm



0.005091 Max
0.004073
0.003054
0.002036
0.001018
0 Min

Figure 27:Base Displacement Analysis

Design Factor of Safety = 12

4.1.1.4 FEA analysis

From the FEA analysis of critical components to the design each component has a high design Factor of safety (FOS) which one could expect since loads are small. The lowest design FOS is offered by the design of the bracket (10) which should ensure the part will not fail. The design of the bracket would allow for a relatively “high” displacement of 0.8123 mm which is brought about by the overturning moment caused by the eccentricity of the servo motor, this would need to be considered when designing the rotational system. From the subsequent FEA results the structural designs of the system is successful.

4.1.1.5 Design of an actuation system:

For the system to rotate about both the z and x axes, an appropriate method of actuation would need to be designed. Within the conceptual design the method of actuation about the z axis can be seen to be a synchronous belt driven system. A drive pulley would be attached to the Stepper motor and a driven pulley would be attached to the bracket housing the Laser ranging sensor.

The student had decided to enquire as to the availability of small synchronous timing pulleys and their appropriate timing belts. These synchronous pulleys would need to attach to a bipolar NEMA 17 stepper motor the student was able to salvage from a previous project. The shaft diameter of the stepper motor is 5mm thus the appropriate inner bore diameter pulley would need to be attached. The student had decided to use one aluminium pulley which would attach to the stepper motor shaft and to 3-d print the driven pulley. The student would have the ability to modify the design of the driven pulley to attach to the system to the bracket.

From the understanding of the maximum power output from the NEMA 17 stepper motor the student would design the actuation system. The maximum speed that the driven pulley may operate at is dependent on the refresh rate of the laser ranging sensor. The laser ranging sensor has a refresh rate of 270 Hz seen in Appendix J thus, movement from the driven pulley should not exceed this.

Since:

$$1\text{Hz} = 2\pi \text{ rad. s}^{-1}$$

$$\text{maximum speed (driven pulley)} = 1696.46 \text{ rad. s}^{-1}$$

This maximum speed would be used as a design constrain in the development of the rotational system about the z axis.

Gears and belts selected for this design would be of the specification of GT2 which specifies pitch diameter as 2mm with the use of a belt width of 6mm. The viability of this design can be seen in Appendix C.

Initially a design with a gear ratio of 1:1 had been decided upon which was then changed to a ratio of 1:3. After a prototype had been constructed this will be discussed in Section 4.2.

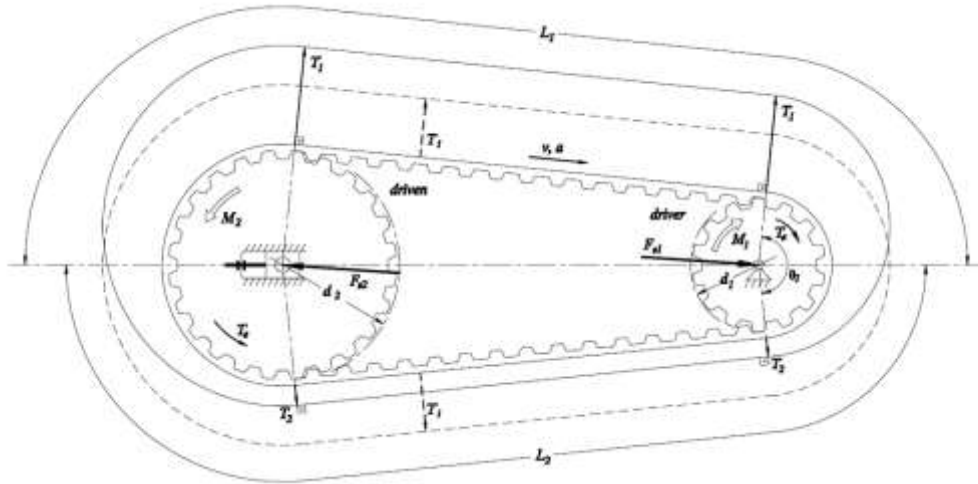


Figure 28: Synchronous Pulley Design

Formulas used include:

$$T_e = T_1 - T_2$$

Where:

T_e = effective tension of the belt (N)

T_1 = tight side tension of the belt (N)

T_2 = slack side tension in the belt (N)

$$M = T_e \cdot \frac{d1}{2}$$

M = driving torque of the driven pulley (N.m)

$d1$ = driver pulley pitch diameter (m)

$$\omega = \frac{\pi \cdot n}{30}$$

ω = angular velocity (rad.s⁻¹)

n = rotational speed (rpm)

$$\omega_2 = \omega_1 \cdot \frac{d1}{d2}$$

ω_2 = angular velocity of the driven pulley in (rad.s⁻¹)

ω_1 = angular velocity of the drive pulley in (rad.s⁻¹)

$d1$ = diameter of the drive pulley

$d2$ = diameter of the driven pulley

$$M_1 = T_e \cdot \frac{d1}{2} = \frac{M_2}{\eta} \cdot \frac{d1}{d2}$$

M_1 = driving torque of the driven pulley (N.m)

M_2 = torque requirement

η = efficiency of belt drive (usually 0.94 – 0.96)

From the calculations found within Appendix C the Maximum angular speed that the system may travel is limited by the slip ring as max operation speed is 300rpm.

Max angular velocity

$$\omega = 31.4159 \text{ rad.s}^{-1}$$

The output moment achieved by the system is:

$$M_2 = 0.089 \text{ N.m}$$

For actuation about the x axis the main concern would be that of torque required to lift the sensor and mount precisely through 0° to 90°. Mass to be lifted had been found to be 48g with a distance from the horn of the servo to the centroid of the sensor being 30mm.

Thus, torque required from the servo motor is a minimum of:

$$T = Fl$$

$$T = (0.048 \times 9.81) \times (0.03)$$

$$T = 0.01412 \text{ N.m}$$

The torque supplied by the Power HD – 1160A servo motor which had been selected is rated to a stall torque of 2.7kg.cm which can be translated to 0.26477 N.m thus the performance is satisfactory. With this information the student could then implement this design for actuation seen in figure 29 and 30.

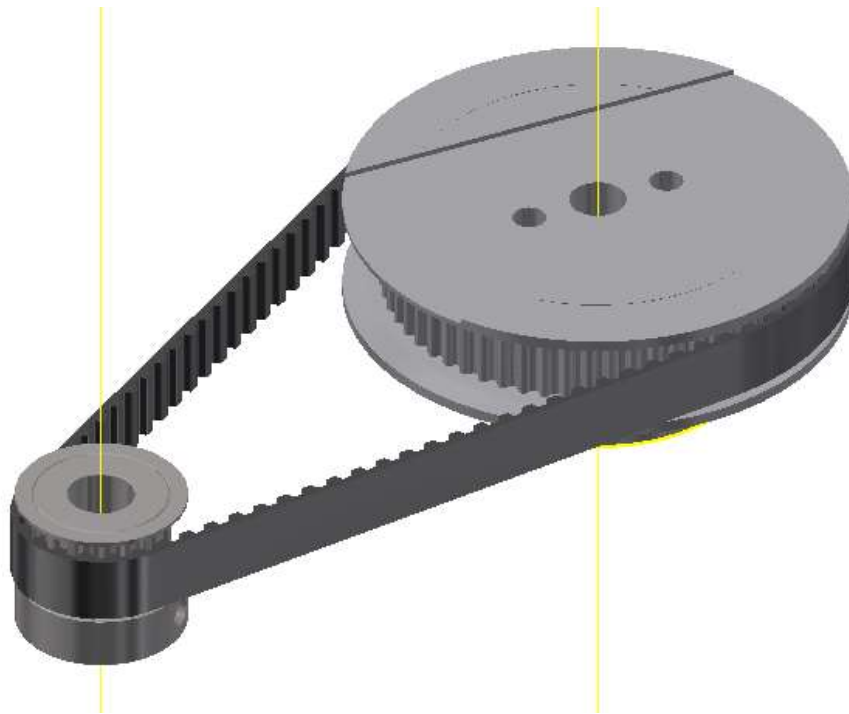


Figure 29: Synchronous Pulley design

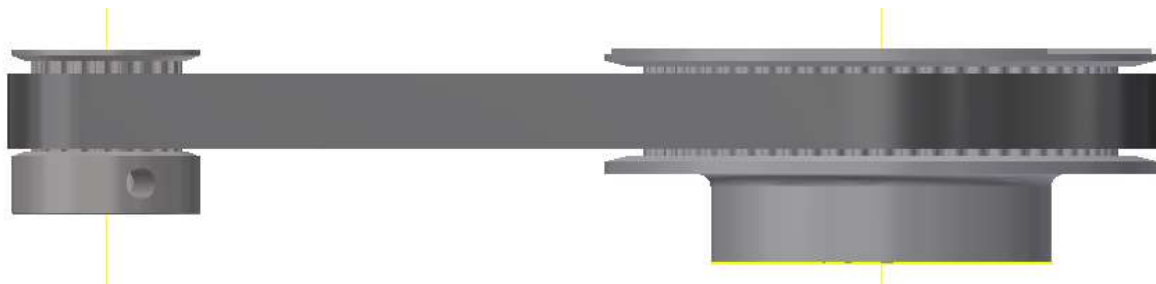


Figure 30 : Synchronous Pulleys and Belt

4.1.1.6 Slip ring

To successfully transfer signal from the laser ranging sensor whilst the system is in motion, an appropriate design would need to be included. From the concept in figures 7 and 8 a slipring had been included for this specific operation. A slip ring can carry signal from a stationary wire to a rotating wire, by transferring signal through a stationary metal contact (brush). Signal to be transferred includes that of the power and control connections from the Power HD – 1160A servo motor.

The slip ring that had been chosen for this operation is rated for voltage of 240V and a current consumption of 2A. Due to the unavailability of a flanged slip ring the student had sourced a miniature 12-way slipring SRC022C – 12 seen in figure 31. A flange to mount this slipring had to then be designed as to appropriately mount the slipring to the underside of the base. The flange to be designed would need to be the appropriate length as to allow for the rotor of the slipring to be protruding for actuation of the system.



Figure 32 Miniature Slip Ring



Figure 31: Designed Flange



Figure 33: Slip Ring assembly

4.1.1.7 Final Assembly Rendering

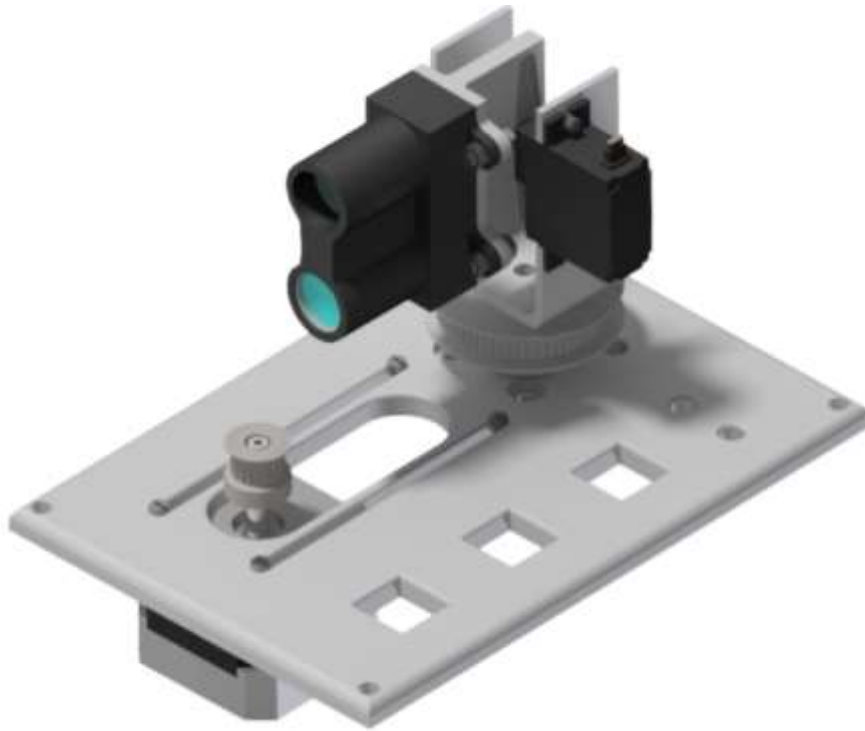


Figure 34: Final System Design

The final System Design can be seen in figure 34, The synchronous belt was not added into this assembly as it has been shown in figures 29 and 30 motors and sensor were added. Note the system is secured with M2, M3 and M4 screws. Provisions for peripherals such as buttons for inputs and Mode selection indicators are shown on the base.

4.1.1.8 System Housing

To protect the system and all subsequent electrical components the student had decided a housing would be necessary. The housing would need to have the required space for components and that of the MCU (Micro-controller) whilst keeping all circuitry hidden from the user.

For the laser ranging sensor kept on the exterior of the housing the required cut-outs would be needed:

- Mounting points for the base design.
- Holes for on/off switches and an E-Stop.
- Attachment of cable glands to allow for power in and communications out.
- Possible attachment of a cooling system to the design.

After doing sourcing for available electrical housings the student had found that the GEWISS G44209 was readily available seen in figure 35



Figure 35: GEWISS Enclosure

The size of the housing would suit the needs of application since it is sized at 300(H) x 220(W) x 120(D). The system would be secured in durable environment since the housing is made from PVC and has an IP65 waterproof rating. The housing will serve as the main structural base for all peripherals to be connected to.

4.1.2 Electrical System design

Electrical system design for the device maybe broken up into the following sub components, namely that of:

- Power supplied to the system.
- Selection of Laser ranging sensor.
- Motor drive operation.

Each of the following subsections will be described in detail with the following deliverables kept in mind:

- Isolated circuitry such as to not damage components.
- Efficient operation of the laser ranging sensor.
- Smooth stable movement of motors as to ensure accuracy of measurements made.

4.1.2.1 Power Supply vs Battery

Initial considerations for power supplied to the system was that of battery operation vs that of an AC to DC step down conversion. The units that would need power from the system would be that of the two motors namely that of the Wantai NEMA 17 stepper motor and that of the Power HD 1160A.

In consideration of a battery for the operation the pros and cons were weighed out in the Table 7.

Table 7 : Battery vs Power Supply

	Battery	Power Supply
Mobility	High mobility, small and can be mounted into a system. Needs Recharge when depleted.	Low mobility, needs a supply of 220V AC wall power.
Power delivery	Dependent on cut-off voltage of the battery, delivery diminishes until cut-off voltage reached.	Constant.
Peripherals needed	Charger	None
Price	R400 – R1000	R200-R300

From the advantages seen in table 7 with the main considerations being that of budget and engineering principle design the student had deemed that the best option would be a power supply.

4.1.2.2 Power Supply Design

The student had initially considered the benefits of creating their own 220V AC to 12V DC step down converter from their knowledge of Power electronics. The circuit needed to step down a 220 AC signal down to a 12 DC would need to comprise of a switching device as to only allow certain portions of the AC waveform to be passed through. From research the student had found that a thyristor may be used for this function. The current characteristics of a thyristor may be found in figure 37.

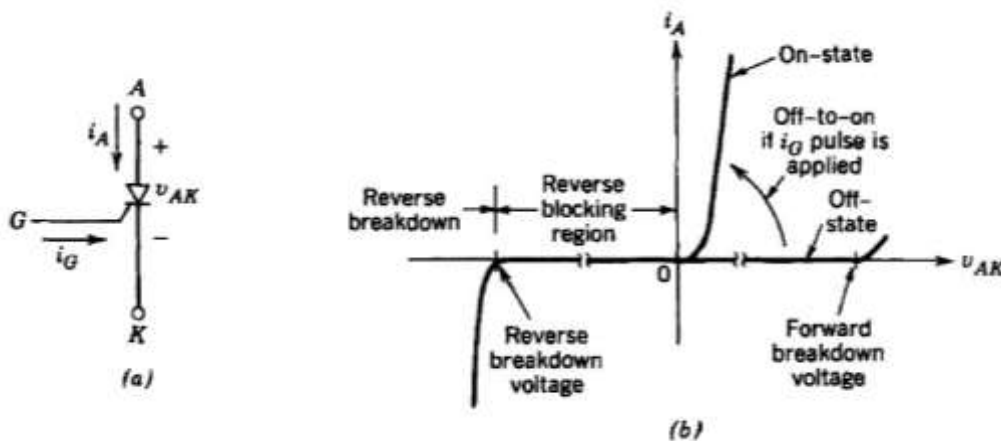


Figure 36: Thyristor Characteristics

From the understanding of a thyristors voltage characteristics from figure 36 a and b show that the thyristor cannot be turned off by its gate and may conduct in a similar way to that of a diode. The proposed circuit needed for this design with the use of thyristors would be a single-phase thyristor converter seen in figure 37

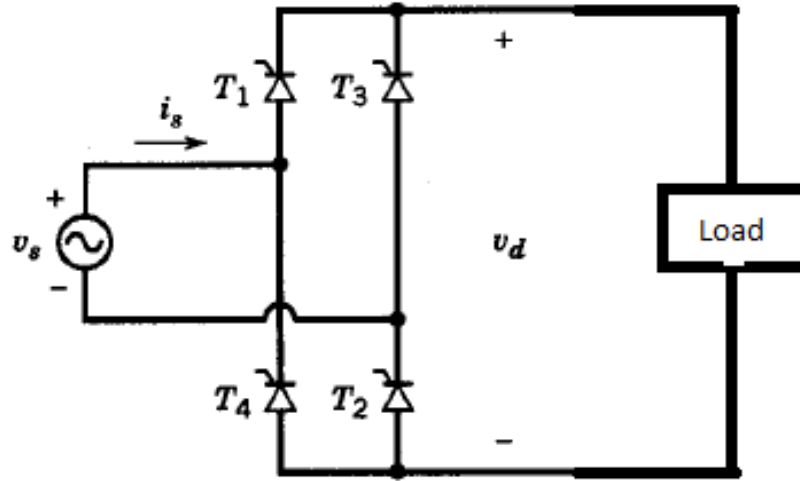


Figure 37: Single phase Thyristor converter circuit

The Firing angle required for this operation can be found via the following formula:

(Mohan, Robbins & Undeland, 2007)

$$\begin{aligned}
 V_d &= \frac{1}{\pi} \int_{\alpha}^{\pi+\alpha} \sqrt{2} V_s \sin \omega t d(\omega t) \\
 &= \frac{2\sqrt{2}}{\pi} V_s \cos \alpha \\
 &= 0.9 V_s \cos \alpha
 \end{aligned}$$

Where:

V_s = the supply voltage

V_d = rectified output from the system

α = firing angle of the thyristor

Circuit parameters are calculated in Appendix D. The simulated circuit can be seen in figure 38. Which comprises of 4 thyristors connected in a bridge configuration with voltage pulse simulators applying the required duty cycle as to rectify the AC waveform into a Stable DC output. A 15 mF Capacitor is added to the circuit to ensure the output produced is a smoothened waveform.

The Circuit Output may be seen in figure 39 and figure 40

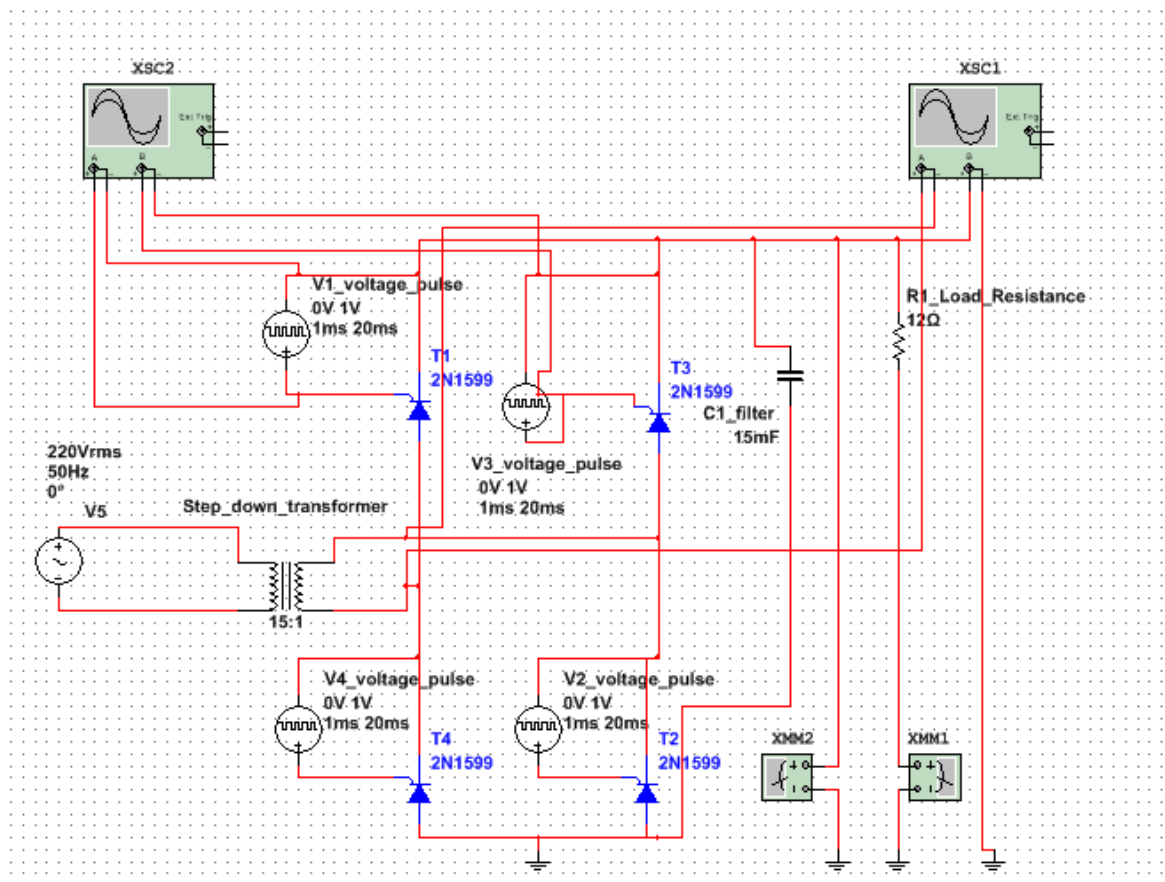


Figure 38 : Thyristor Rectifier Circuit

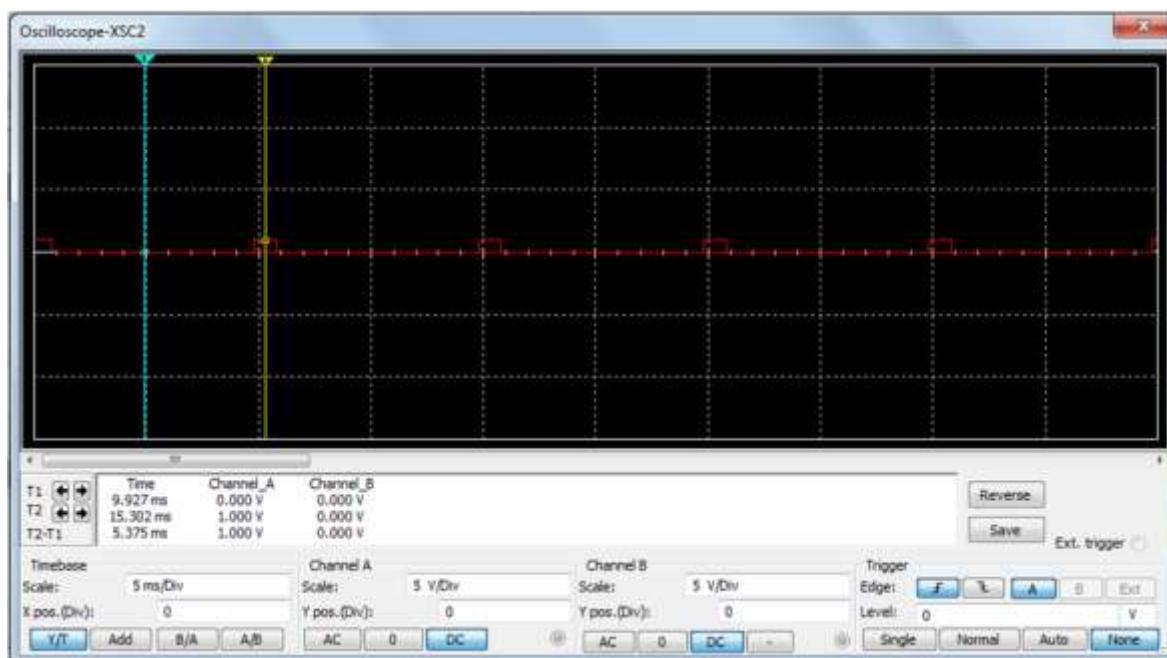


Figure 39: Oscilloscope XSC2 output

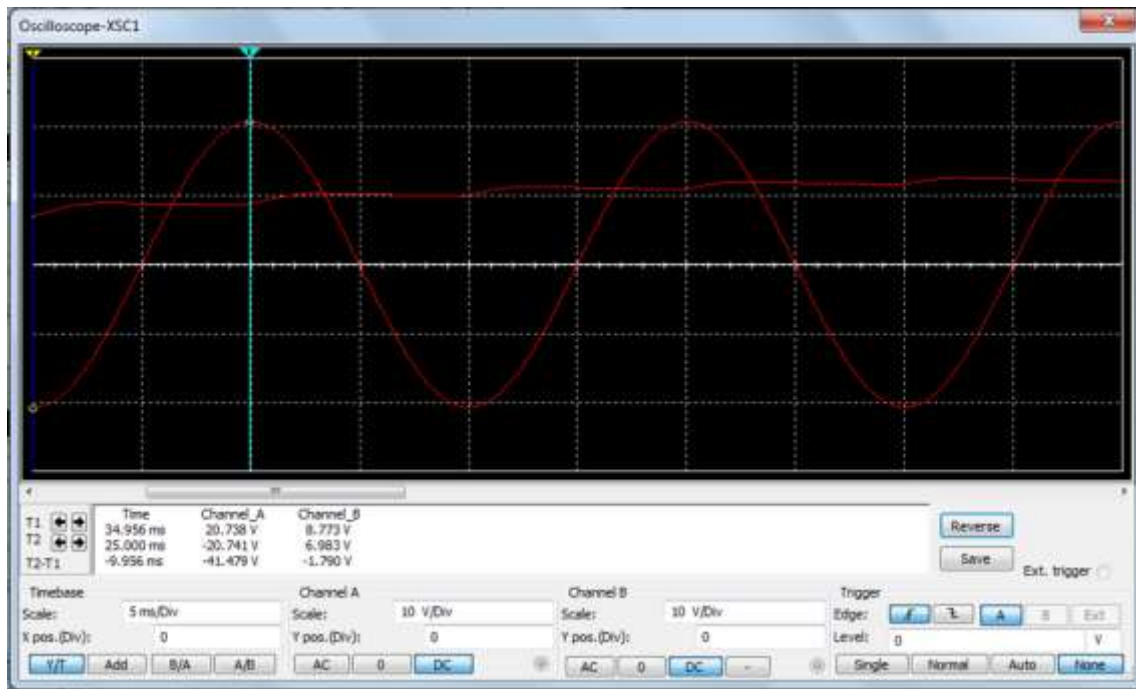


Figure 40: Oscilloscope XSC1 output

This circuit design could be successfully implemented by the student but was deemed too complex for its operation. The student then opted to search for readily available AC-DC converters and implement them into the system. The AC-DC converter module seen in figure 41 chosen by the student is that of an industrial grade power module that could convert 220V AC to 12V DC at 2A. This supply was deemed to be more than adequate for the needs required of the system.

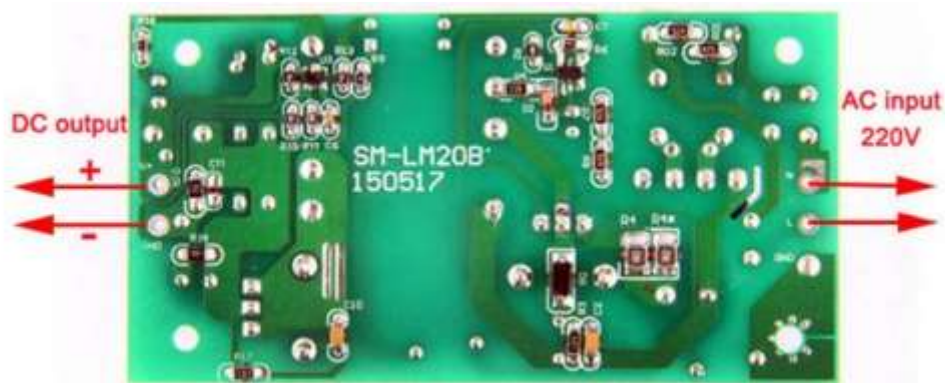


Figure 41: 220v AC to DC module (Micro robotics)

4.1.2.3 Step Down Regulation

The next issue of power to be supplied would be that of the supply required for the Power HD 1160A servo motor. Initially the student had powered this servo motor from the 5V output of the MCU, this resulted in the servo motor having a very slow response since it was under powered from the current supplied. The student had then decided to further step down the input voltage supplied into the system from 12v to that of 5.1V to 5.5V. Methods of achieving this were investigated by the student with two possibilities of operation.

Option 1: The use of a linear voltage regulator connected to the output of the AC to DC

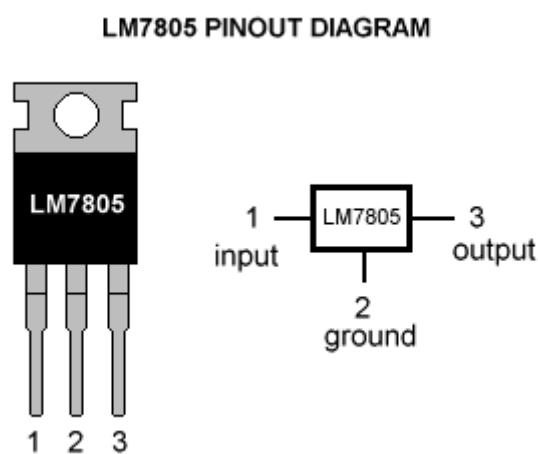


Figure 42: LM7805

A linear voltage regulator seen in figure 42 works on the principle of only allowing a regulated output voltage through IC and dissipates the unused power as heat.

Since the rated current consumption of the NEMA 17 Power dissipated by the LM7805 can be calculated by:

$$P = \Delta VI$$

$$P = (12 - 5.5) \times 1$$

$$P = 6.5 \text{ W}$$

6.5 watts being dissipated by the system would see a large increase in heat being generated, due to its nature of thermal protection the regulator will shut down after temperature threshold is passed which is 125°C. thus a heat sink would be necessary for continuous operation

Efficiency of a linear voltage regulator may be calculated by

$$\eta = \frac{P_{out}}{P_{in}} = \frac{V_{out} I_{out}}{V_{in} I_{in}} = \frac{5(1)}{12(2)} = 20.83\%$$

Option 2: The use of a Buck converter circuit to variably step down the 12V supply.

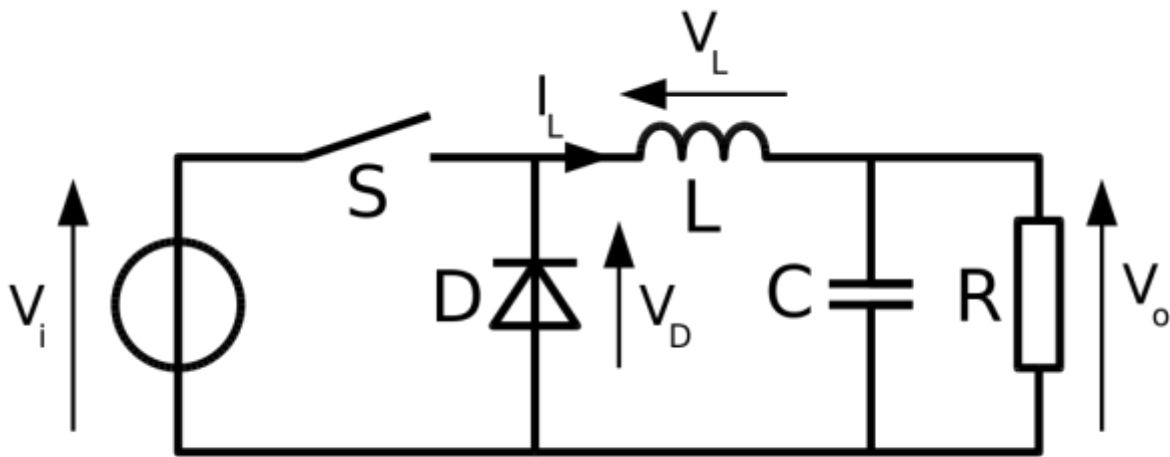


Figure 43: Buck Converter Simplified Circuit

The basic buck converter circuit can be seen in figure 43. The input to the circuit would be that of the 12V input at 2A. There are two switches within the circuit, namely that of S and D, with S being the active controllable MOSFET and D being the passive diode. An inductor-capacitor filter is used to produce a DC output. The load represented by the resistor R will be that of the servo motor.

Two modes of the system are possible, being that of CCM (Continuous Conduction Mode) or that of DCM (Discontinuous Conduction Mode). For the understanding of this circuit, the operation will be in CCM. If the current consumption of the circuit is assumed to be a constant (steady state), the following formula can be used to calculate the duty ratio (active switching time) (Mohan, Robbins & Undeland, 2007):

$$D_t = \frac{\langle V_0 \rangle}{V_i}$$

Where:

D_t = the duty ratio

$\langle V_0 \rangle$ = the required output

V_i = the input voltage

$$D_t = \frac{5.5}{12}$$

$$D_t = 45.83\%$$

Thus, the MOSFET switch will need to be active for 45.83 % of a total cycle. The duty ratio is the component of the Buck converter that can be varied to obtain a varied output.

If the efficiency of the buck converter is taken as a maximum value obtained from the supplier which is 92% the power dissipated by the circuit may be calculated via the formula:

$$P_{switch} = (1 - \eta)P_{out}$$

$$P_{switch} = (1 - 0.92)(5.5 \times 1)$$

$$P_{switch} = 0.44 \text{ W}$$

Table 8 : Voltage Regulator Vs Buck Converter

	Option 1 LM7805	Option 2: Buck boost
Power dissipation	14W	0.44W
Efficiency	20.83%	Max 92%
Peripherals needed	Heat sink, smoothing capacitor	none
Variability of output	4.8V to 5.2V	1.25V to 37V
Cost	R6 (Hobbytronics.com)	R45 (Hobbytronics.com)

From table 8 the student had decided to utilize the benefits of adding a buck converter to the system. Although the cost of the circuit would be more than the use of a linear voltage regulator the system would operate more efficiently with no need for an added heat sink.




4.1.2.4 Laser Ranging Sensor Selection



To select an appropriate laser ranging sensor for the system the student would need to consider the following aspects.

- Size and weight of the sensor, the size of the sensor would directly affect the need for actuation with a system being smaller the demands on both actuation and power supply would be decreased.
- Accuracy of the sensor, the minimum resolution of the sensor directly affects the output of the points found, Sensor resolution would directly affect the quality of the exported point cloud seen by the user
- Refresh rate, the quicker the sensor can record distance values the quicker it would be able to move. For the purposes of updating a point cloud in real time the sensor would need to operate at a high refresh rate.
- Power consumption, the smaller the power consumption of the sensor the less strain would be put on the power supply units to the system i.e. ideally the sensor could be powered via the MCU.
- Range of the sensor would be directly responsible for its applicability to the operation of navigation, the higher the range the more confidently the system could operate within a warehouse instance.
- Price, the laser ranging sensor would need to fit in with the R5000 budget allocated to the student.

Comparisons of alternatives can be seen to be compared within table 9.

Table 9: Device Properties

Component	Method of operation	Accuracy	Resolution	Range	Dimensions (l x b x h)	Refresh Rate	Price	Image of component
Garmin Lidar lite v3	LIDAR	< 5m +-2.5cm ≥ 5m +- 10cm	1cm	0-40m	20x48x40 22g	270Hz	R2500 Micro Robotics (2017)	 <p>Figure 44: Garmin Lidar lite v3 Robotshop (2017)</p>
SICK-NAV350-3232	LIDAR	≤10m +- 4mm ≤20m +- 10mm ≤30m +- 15mm ≤ 50m +- 25mm	1mm	0.5-70m	120x116x224 2.4Kg	8Hz (360° scan)	R126000* ELTRA (2017)	 <p>Figure 45: SICK NAV-350 Sensortrade (2017)</p>
Ultra-Sonic (HCSR04)	Ultrasound	≤ 4m +- 3mm	0.3cm	0.02-4m	45x20x15 15g	40Hz	R30 Hobbytronics (2017)	 <p>Figure 46: HCSR04 sensor MACFOS (2017)</p>

Component	Method of operation	Accuracy	Resolution	Range	Dimensions (l x b x h)	Refresh Rate	Price	Image of component
Microsoft Kinect	RGB-D Camera	Not released by Microsoft	Depth 1cm	0.8-3.5m	280x63.5x38 1.36Kg	30 Fps (Hz)	R2000 (second-hand)	 <p>Figure 47: Kinect sensor Engadget (2017)</p>
Laser and Webcam	Optical (Pixel calculation)	Subjective to studies	Dependant on camera (1080 pixels in HD webcam)	Dependent on laser pointer	Camera and laser pointer Dependant	Camera Dependant 30 Fps (Hz)	R700 (Price of an HD webcam)	 <p>Figure 48: Laser webcam Codeproject (2017)</p>

Information regarding the operation of the device may be found in appendix J.

The preferred laser ranging sensor selected by the student is that of the Garmin Lidar Lite v3. Although the sensor is a more expensive option in comparison to that of the Ultra-sonic (HCSR04) sensor or that of using a camera unit (Microsoft Kinect or Webcam) it has the benefits of size, accuracy, and refresh rate. The refresh rate of the sensor being that of 270Hz would allow for a point cloud to be generated as the sensor is moved through a full 360°.

In comparison to the industrial sensor the Lidar lite is disadvantaged as a full circular sweep is done with a rate of 8Hz By the SICK NAV 350 in comparison to that of a single point update of 270Hz. The price comparison of the SICK NAV 350 in comparison to the Lidar lite is highly exorbitant as the cost of one unit is more than 50 times the price of a Lidar lite unit.

The Garmin Lidar Lite V3 may be controlled in either a PWM or I2C configuration values for signal strength and noise may be returned when connected in the I2C configuration. The Garmin Lidar Lite v3 draws current of 135mA thus could be powered via the MCU. The student had proposed the use of these values to determine the location of reflective material in the surroundings of the sensor. Experiments of these parameters can be seen in Section 5.1.

4.1.2.5 I2C configuration

For the sensor to operate via the I2C configuration the student would need to connect the Garmin Lidar lite v3 to both the SCL and SDA lines of the MCU. The SCL line is known as the clock line whilst the SDA line is known as the data line seen in figure 49. The I2C bus seen in figure. allows for systems connected to act as either a master or slave device. The master device is the driver of the SCL clock line, in this case the MCU, and the slave is the device, Garmin Lidar lite, that responds to the master accordingly.

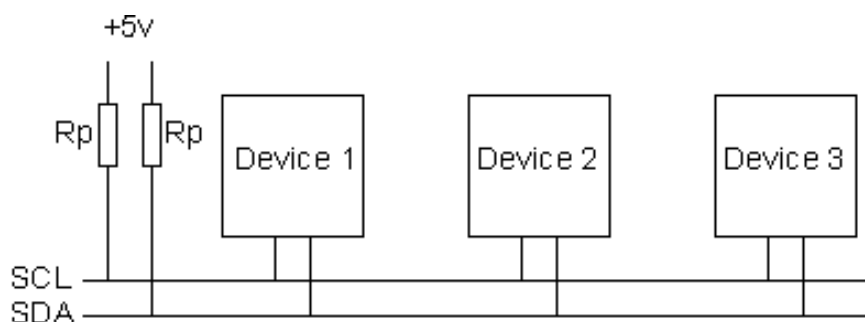


Figure 49: I2C operation

Data is transferred along the I2C bus in sequences of 8 bits which are then placed on the SDA line Starting with the most significant bit or the MSB seen in figure 49. The SCL line will then pulse high and low. After the 8 bits are sent the receiving device will then send an acknowledge bit thus for every 8 bits or 1 byte the SCL line must pulse 9 times. If the Acknowledge bit is not received by a device data transfer is halted as the device is not ready send/receive another byte and thus a stop sequence terminates the operation.

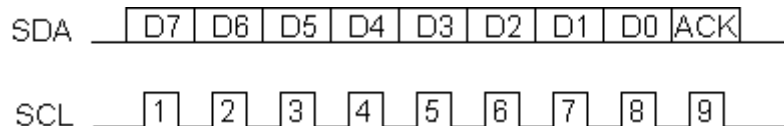


Figure 50: SDA and SCL Operations

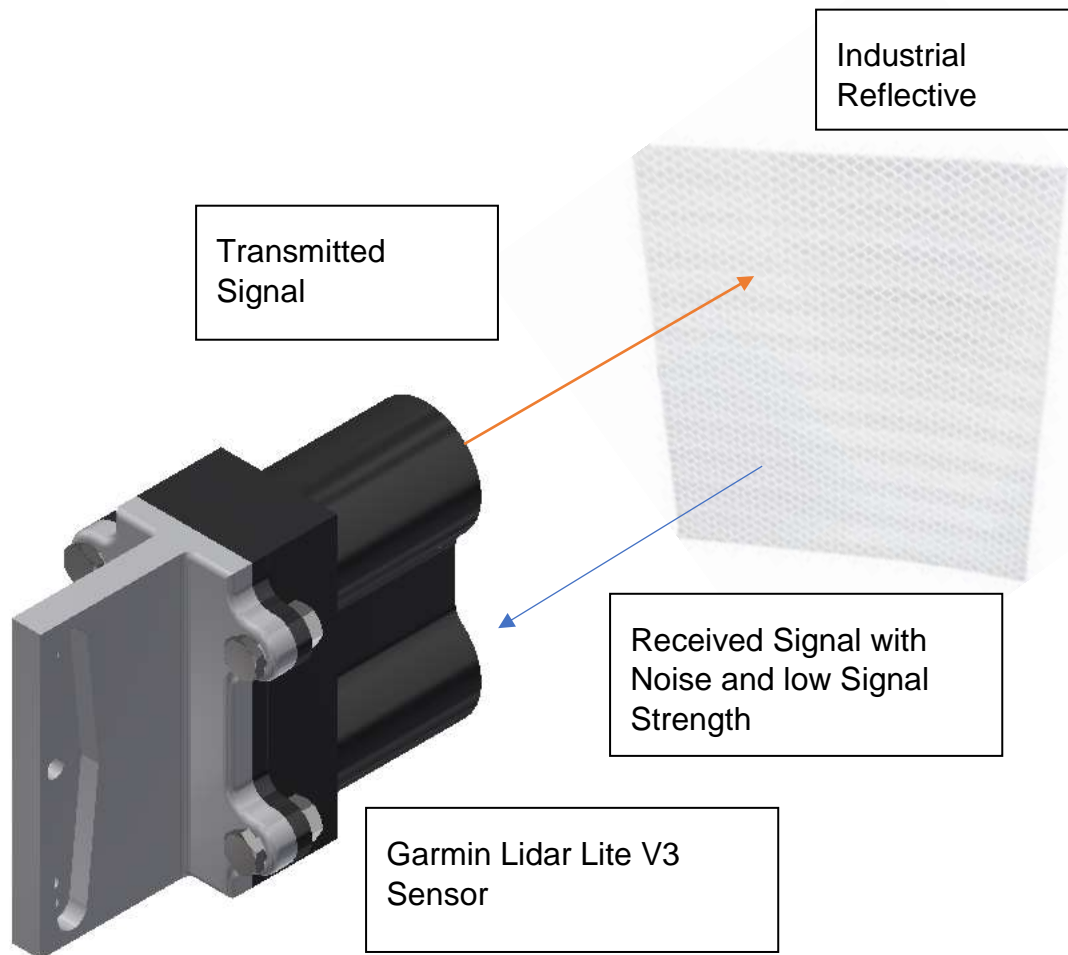
4.1.2.6 Proposed sensor operation

Sensing of Landmarks via reflective material

The system would need to achieve the following to determine exclusive points in the surroundings of the sensor. The student had determined that the best method of differentiating landmarks would be to emulate the operations of the industrial sensor namely that of the SICK NAV 350. Thus, the system would need to utilize industrial grade reflective material. The student had obtained a piece of this material from a currently ongoing post-graduate project.

The method of operation seen in figure 51 would be:

- Scan surroundings with Garmin Lidar Lite V3 in I2C configuration.
- Lidar Lite would be actuated via stepper motor and Servo motor.
- Analysis of the feedback from the sensor as distance would be found via TOF operation.
- If Signal strength diminishes bellow a determined threshold value or if the Noise value detected rises above a determined threshold a reflective landmark is present.
- This method was tested in section 5.2.1



4.1.2.6 Stepper Motor Control

The control of the stepper motor would be provided by the stepper motor controller namely that of the Spark-fun easy driver V4.4 this would control the switching frequency of controlling the bi-polar stepper motor. Bi-polar stepper motors are controlled via the energising of windings found within the motor itself seen in figure 52. The action of energising the combinations of windings will cause a magnetic which drives the stepper motor.

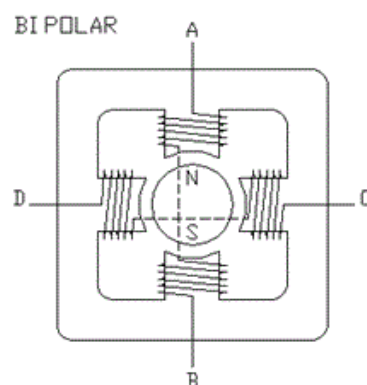


Figure 52: Stepper motor windings

The stepper motor would need to be controlled through micro stepping since the student had opted for a higher resolution than that provided by the stepper motor itself which is 200 steps per revolution thus the micro steps option on the driver would need to be activated. Micro steps would allow for a better resolution of the system i.e. the use of half steps which would increase the said resolution from 200 steps to 400 steps per revolution. The input to the easy driver stepper is that of 12V supplied by 230V AC to 12 DC converter module. Configuration can be seen in figure 53.

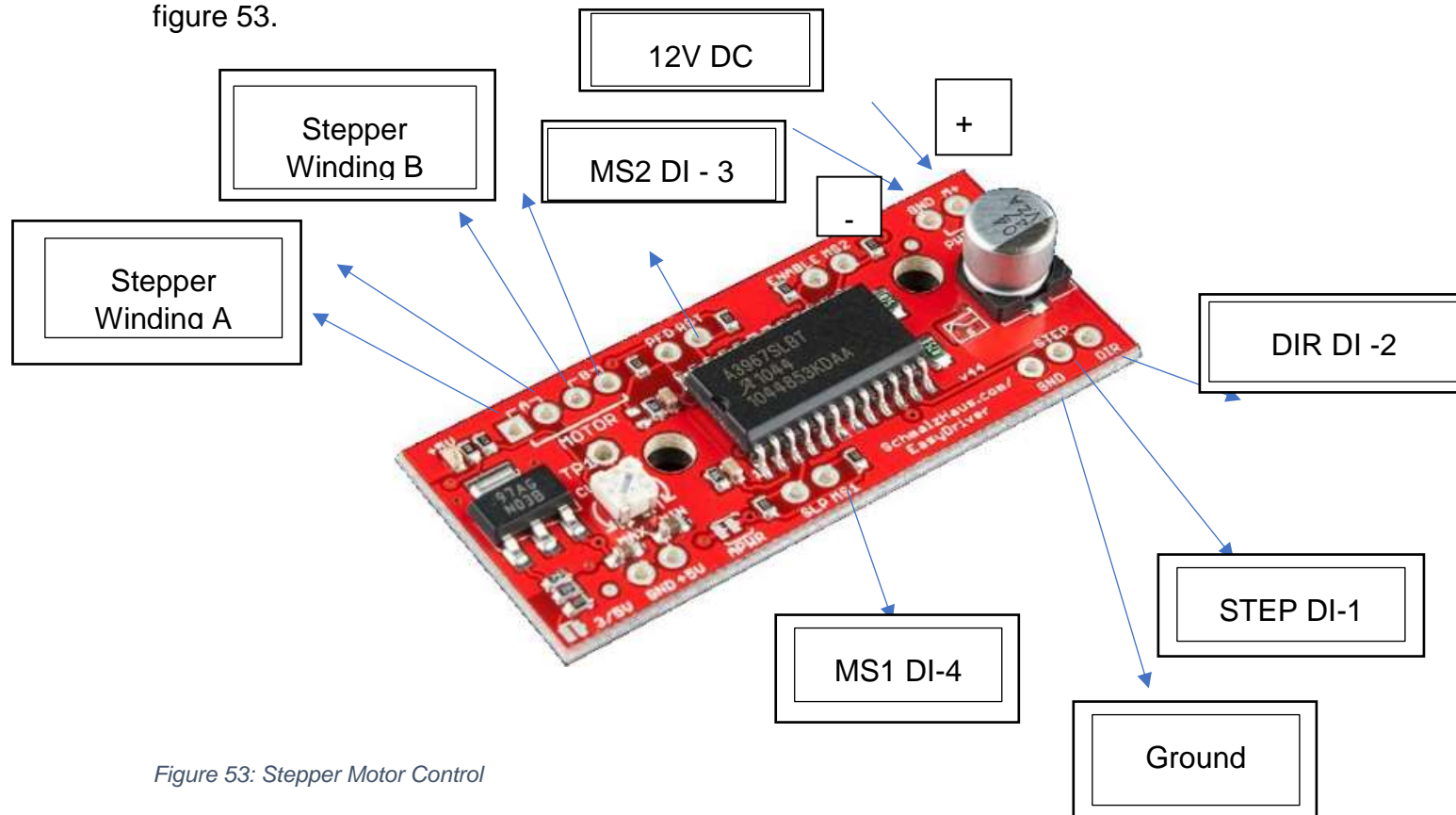


Figure 53: Stepper Motor Control

An issue with the use of a stepper motor is the feedback of position which would normally require the use of a rotary encoder. The student had not included the use of a rotary encoder into their design but rather the use of an LDR (Light Dependent Resistor) seen in figure 53 that would provide feedback of position on the output gear. The system would work under the principle of having a Black dot placed on the underside of the output gear seen in figure 54 for which the LDR could detect. If the dot is detected the output gear has travelled to its "home position". This method will be used to determine at which point the servo motor should elevate the sensor.



Figure 55: LDR



Figure 54: Output Gear Home position

4.1.2.7 Servo Motor Control

Servo motors operate via the functionality of a small geared DC motor, a potentiometer with an imbedded control circuit seen in figure 56. As the motor rotates the resistance supplied by the potentiometer changes, which allows the control circuit to regulate direction and movement.

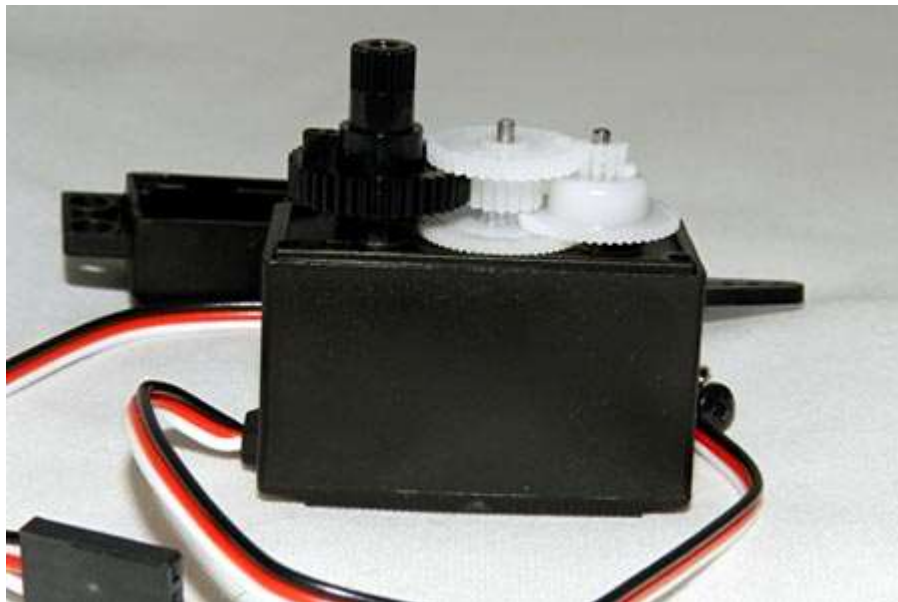


Figure 56: Servo Motor

Servo motors consist of 3 connections namely that of voltage in, ground and a signal input. The desired position required for the operation is sent via the signal input this can dictate movement to the servo in either direction with a maximum degree of

movement being 180° . Since the maximum movement required for the devices operation is 0° to 90° this is ideal.

The control of position sent via the signal input is done via Pulse Width Modulation or (PWM). The servo motor expects a PWM signal every 20 milliseconds, based on the duration or width of these pulses the servo will react by altering its shaft position seen in figure 57.

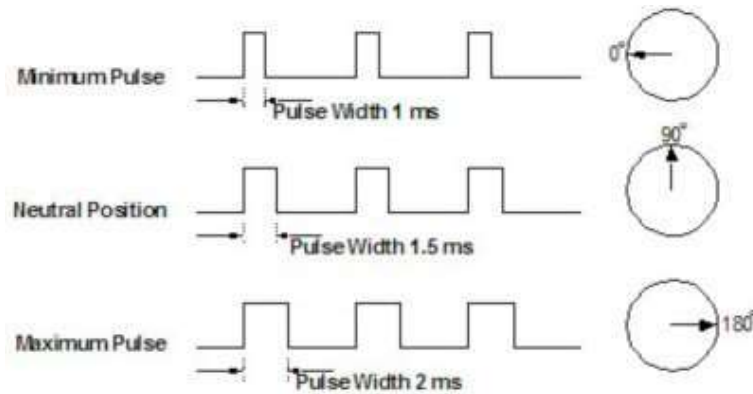


Figure 57: PWM for a Servo Motor

Thus, for the control of the servo motor the student would need to apply the correct input voltage (which had been designed for by the buck converter) and an appropriate PWM signal supplied by digital input pin 5.

The total power Supplied for the system can be seen in figure 58

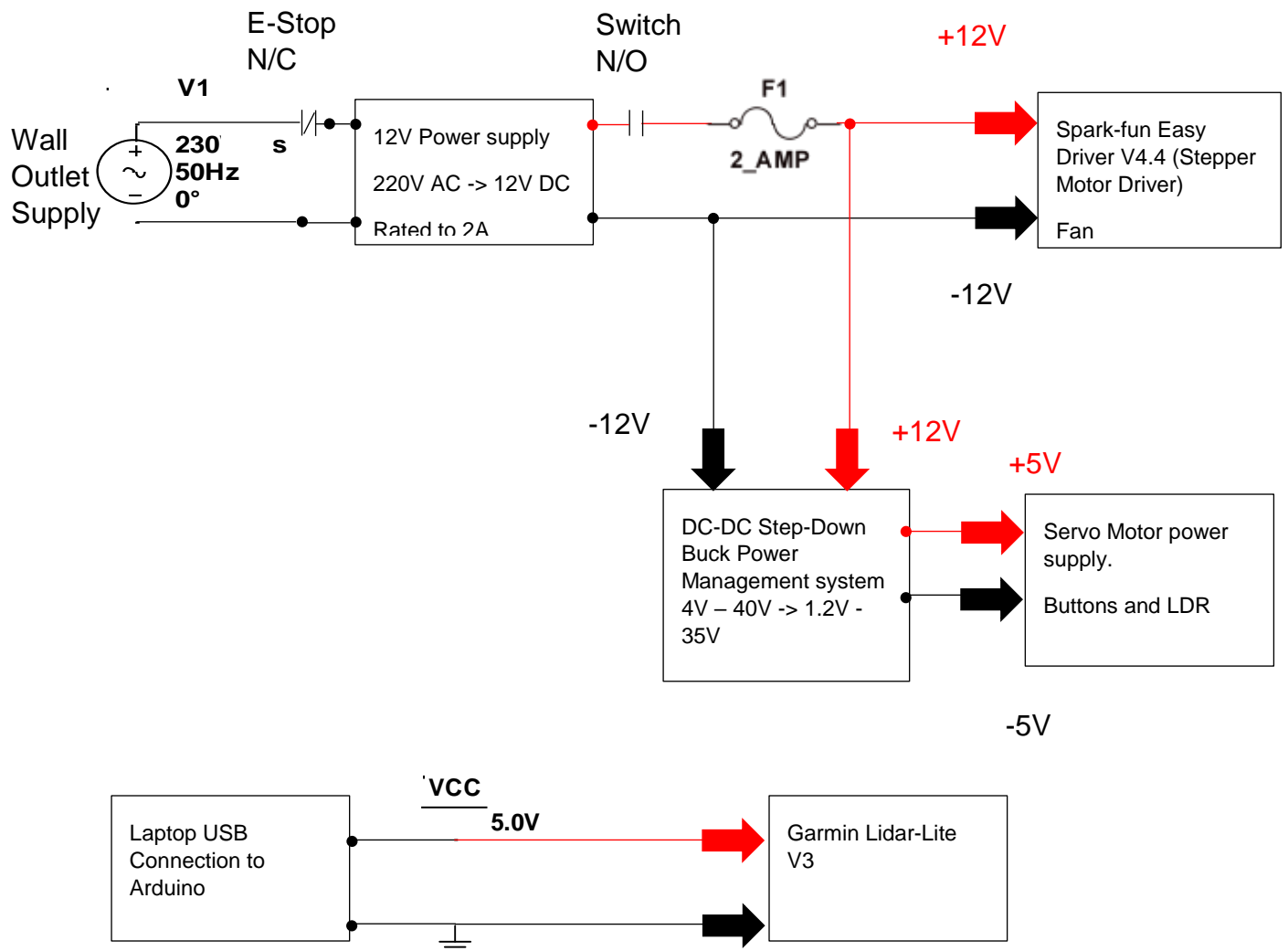


Figure 58: Power Supply Summary

4.1.3 Information Technology design

To correctly implement the design the student would need to utilize many different forms of simulation software, programming development environments and that Microsoft office software. These were inclusive of the following:

- Microsoft Word for the main report documentation.
- Microsoft Excel for statistical calculations.
- Microsoft Project for project management and planning.
- Autodesk Inventor 2017 For CAD designs and Finite Element Analysis.
- NI Multisim for circuit simulations.
- Arduino IDE for system programming and sensor interface (C++).
- Processing IDE for Point Cloud Visualisations (Java).
- MATLAB (2014) for data analysis.
- Fritzing for circuit and schematic diagrams.

After the construction of the system from the proposed Mechanical and Electrical design the system would need to be programmed to perform the TPM's set out by the student.

The inputs into the system were identified as the following:

- Start button input.
- Mode select input.
- Stop Button input.
- Garmin Lidar lite V3 distance, signal strength and noise.
- Servo motor position.
- Stepper motor number of steps.

Outputs of the system were identified as the following:

- Step command to Stepper motor.
- PWM signal sent to the servo motor.
- LDR value to indicate stepper in home position.
- Lidar sensor distance reading, Lidar sensor Signal Strength, Lidar sensor Noise value.

Using this information, the student would need to put together a flow diagram of the logic. The outputs required for construction of a point cloud would be the stepper motor position namely that of the output gear, the servo motor position and that of the distance at that instant

4.1.3.2 Flow Diagram

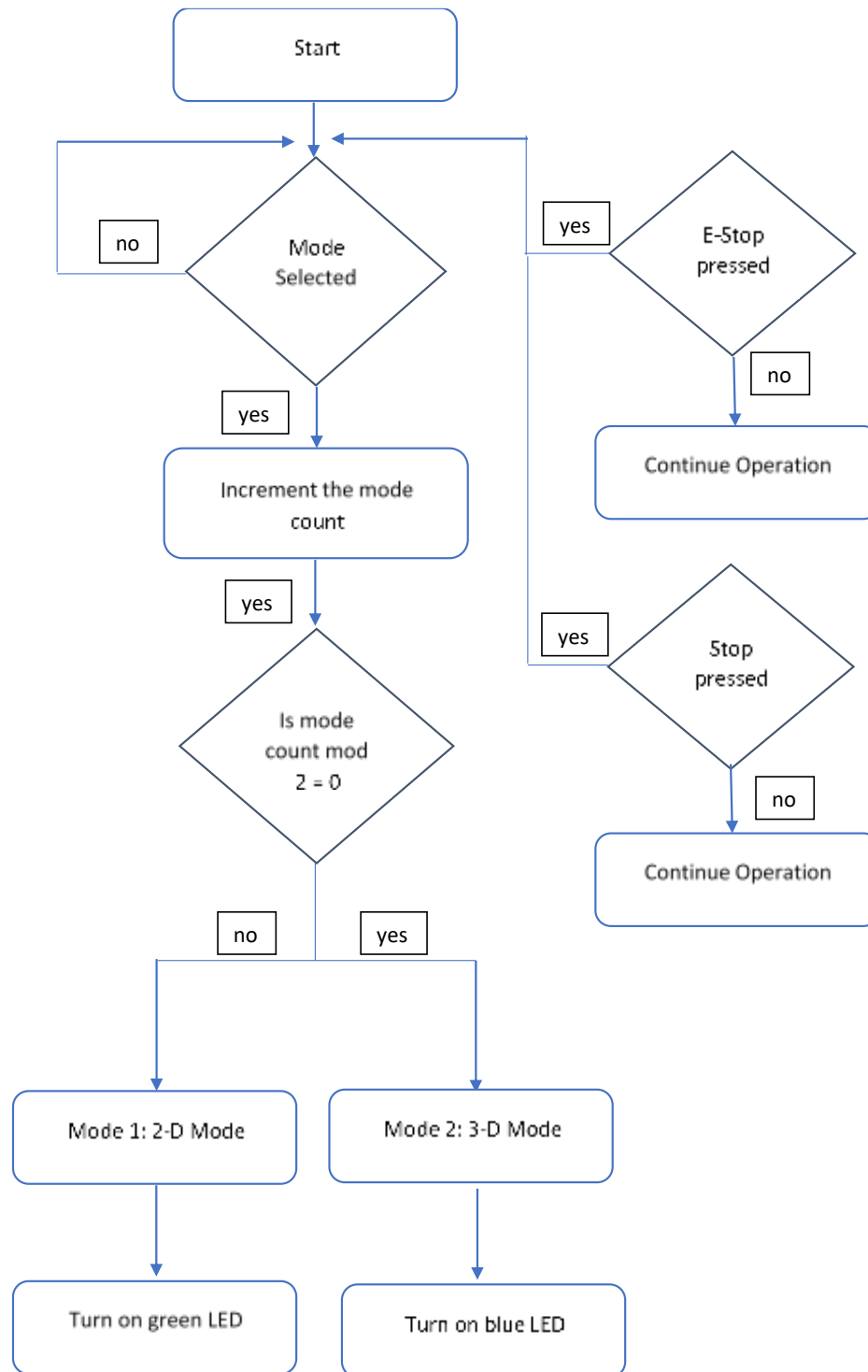


Figure 59: Main Flow Diagram

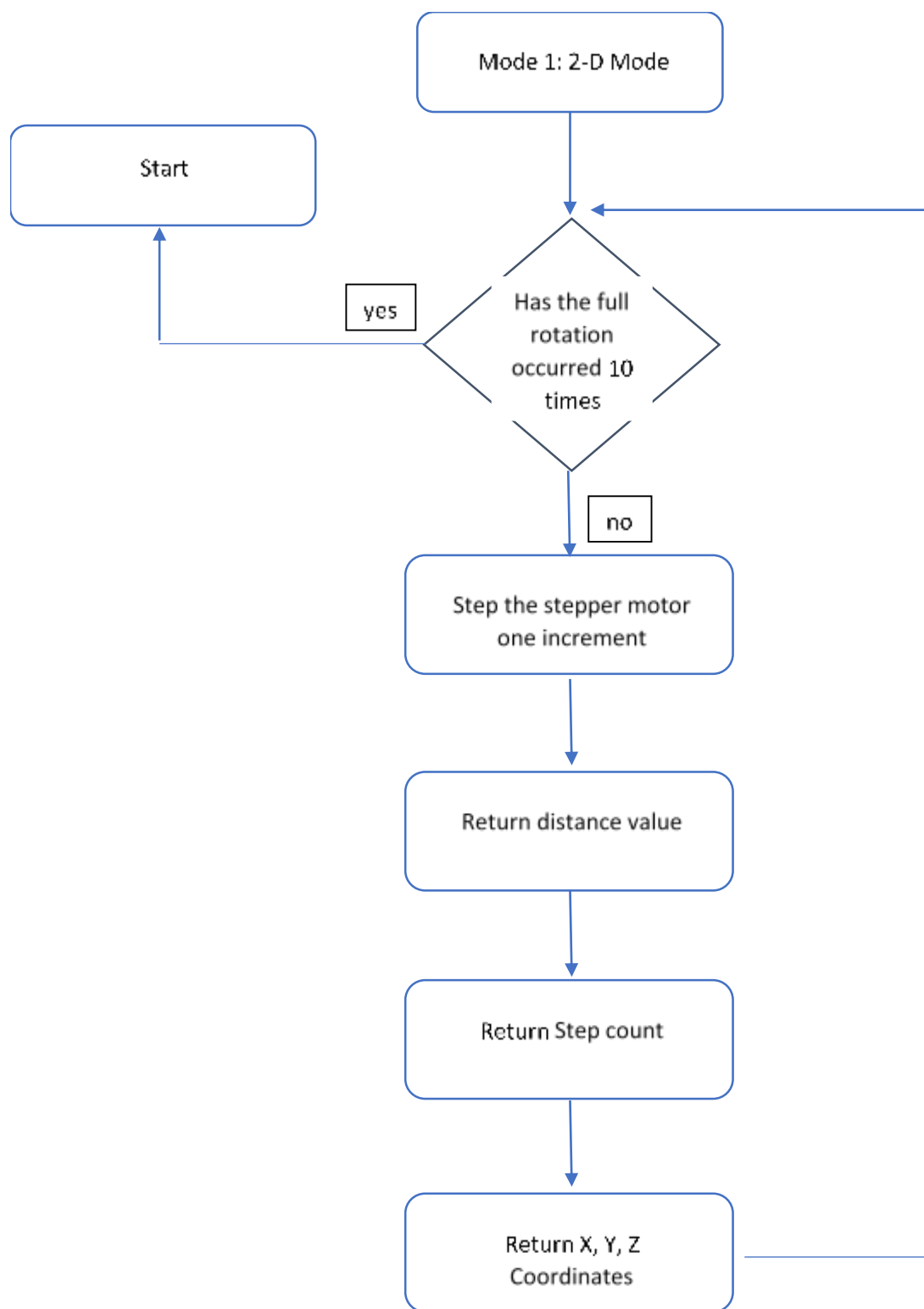


Figure 60: 2-D Mode

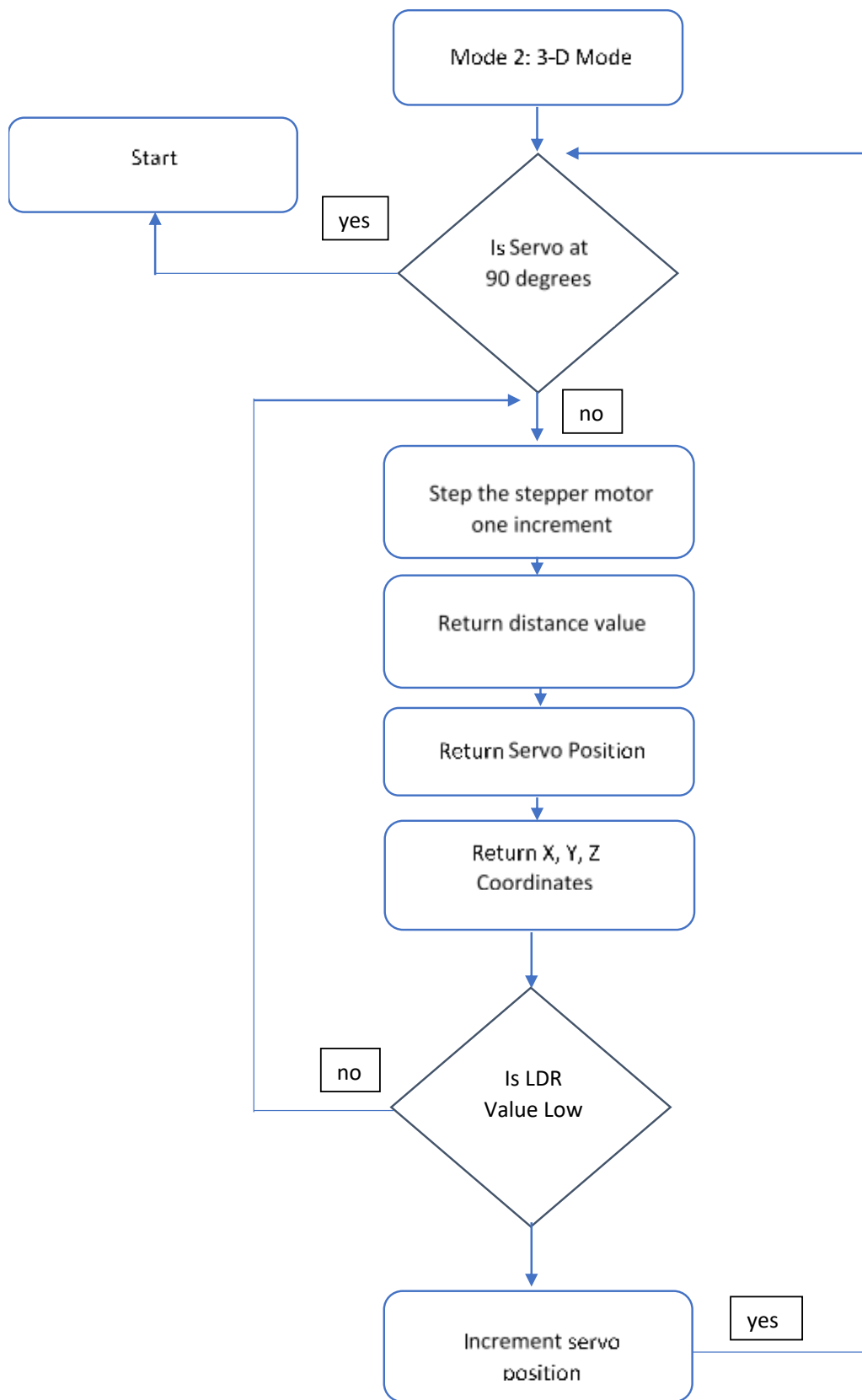


Figure 61: 3-D Mode

4.1.3.1 Point Cloud Output

If the user were to only require performing a 2-D scan of the room the Lidar sensor would need to rotate an appropriate number of times without the elevation of the servo motor to calculate X, Y, and Z (which would subsequently be 0) co-ordinates. After the scanning procedure is completed or a stop command is given the operation should stop and wait for the user to begin the process again.

If the user were to require a 3-D scan of the room the sensor would need to be rotated until the “home” position of the output gear is reached. After the sensor is returned to the home position the servo would need to elevate accordingly. The X, Y, and Z co-ordinates would need to iteratively be calculated whilst the motion of both the stepper and servo motor is occurring. This procedure would need to stop when the servo has reached 90 degrees or either the stop or E-stop is pressed.

4.1.3.1.1 Spherical Co-ordinates

The information returned from the system would be 2 angle values and a distance value. These values could be used by the system to determine the actual position of a point in a 3 axis cartesian plane.

The value returned by the relative position of the stepper motor with respect to the “home” position of the output gear would relate the Azimuth angle of the point in space or the angle θ seen in figure 62

The value returned from the servo motor position would relate to the elevation of the point in space but this value would need to be inverted such as to give the angle from the z axis to the x-y plane thus, from figure 62 the angle ϕ would be $(90 - \text{servo position})$.

Finally, the last point required to determine the relation of the scan to a cartesian plot would be that of the distance value. The distance value is a vector made up from components of both the azimuth angle and that of the elevation angle or the value ρ seen in figure ...

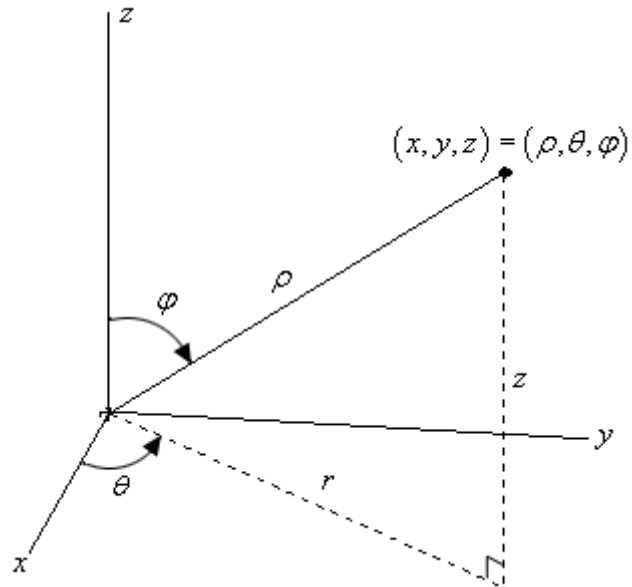


Figure 62: Spherical Co-Ordinates

From an understanding of Calculus, the x , y and z co-ordinates could be calculated via the following formulas (with angles in radians):

$$x = \rho \cdot \sin(\phi) \cdot \cos(\theta)$$

$$y = \rho \cdot \sin(\phi) \cdot \sin(\theta)$$

$$z = \rho \cdot \cos(\phi)$$

4.1.3.2 Trilateration

To localize the device a method of trilateration would need to be applied which allows for a relative distance to be calculated between known landmarks. Initially the student had intended on performing the trilateration process in 3 dimensions such that a position (X, Y) and an altitude (Z) of the device could be determined. When visualizing the trilateration process in 3 dimensions it can be used to determine the position of 3 intersecting spheres seen in figure 63

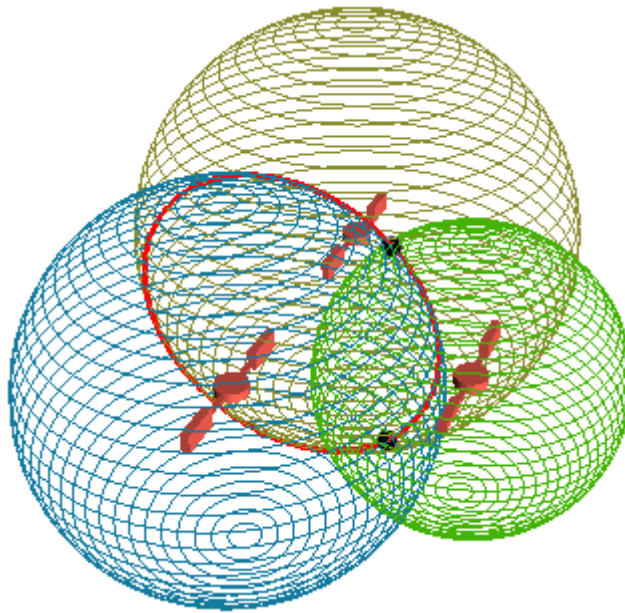


Figure 63: Spherical Trilateration

The process of calculating the point of intersection between 3 spheres is complex and requires many calculations. Thus, the student had rather opted to localize the device in 2 dimensions namely that of X and Y. Since the altitude of an AGV on a factory floor would not necessarily be critical information localization within 2 dimensions would be focused upon.

An Example of usage of the trilateration process can be seen in the figure 63 landmark points are denoted by A1, A2 and A3. Cartesian co-ordinates of landmark points are known to the system and become the centre of the circles with radius of each being the distance the unknown red point. The intersection point of these 3 circles is the location of the point itself.

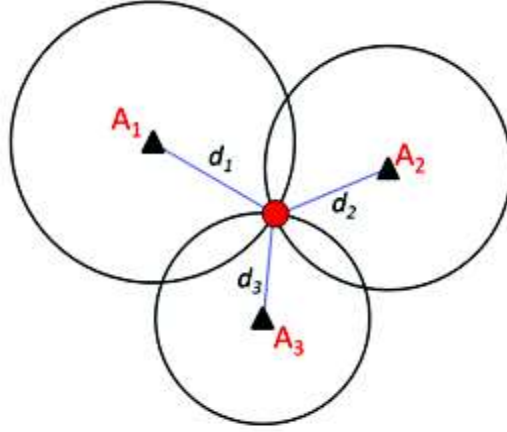


Figure 64: 2-D Trilateration

The algebraic formulas required to compute the distance needed can be seen to be calculated below in equations below (Jarvis et al., 2011). Distances d_1 , d_2 and d_3 would need to be obtained from the Garmin Lidar Lite.

$$(x - x_1)^2 + (y - y_1)^2 = d_1^2$$

$$(x - x_2)^2 + (y - y_2)^2 = d_2^2$$

$$(x - x_3)^2 + (y - y_3)^2 = d_3^2$$

$$2(x_2 - x_1)x + 2(y_2 - y_1)y = (d_1^2 - d_2^2) - (x_1^2 - x_2^2) - (y_1^2 - y_2^2)$$

$$2(x_3 - x_1)x + 2(y_3 - y_1)y = (d_1^2 - d_3^2) - (x_1^2 - x_3^2) - (y_1^2 - y_3^2)$$

$$X = \frac{\begin{vmatrix} (d_1^2 - d_2^2) - (x_1^2 - x_2^2) - (y_1^2 - y_2^2) & 2(y_2 - y_1) \\ (d_1^2 - d_3^2) - (x_1^2 - x_3^2) - (y_1^2 - y_3^2) & 2(y_3 - y_1) \end{vmatrix}}{\begin{vmatrix} 2(x_2 - x_1) & 2(y_2 - y_1) \\ 2(x_3 - x_1) & 2(y_3 - y_1) \end{vmatrix}}$$

$$Y = \frac{\begin{vmatrix} 2(x_2 - x_1) & (d_1^2 - d_2^2) - (x_1^2 - x_2^2) - (y_1^2 - y_2^2) \\ 2(x_3 - x_1) & (d_1^2 - d_3^2) - (x_1^2 - x_3^2) - (y_1^2 - y_3^2) \end{vmatrix}}{\begin{vmatrix} 2(x_2 - x_1) & 2(y_2 - y_1) \\ 2(x_3 - x_1) & 2(y_3 - y_1) \end{vmatrix}}$$

The correct application of Trilateration is highly dependent upon the system being able to differentiate between distinct landmarks. Before the student could initiate the trilateration process the method of determining distinct separable landmarks should be investigated

4.1.3.3 Processing Sketch

To produce a point cloud display from the coordinates calculated from within the Arduino program the student would need to produce the relevant processing sketch. Since Processing can communicate with the Arduino via Serial communication ports the sketch would first need to open the appropriate com port. The student had found that "COM Port: 4" was active via the Universal Serial Bus connection from the Arduino to their computer. This port would not only be used to transfer executable instruction code to the Arduino but in addition monitor the appropriate signal output. The Baud rate of serial communication was found to be highly important as communications would only work properly if the same baud rate was not employed. Baud rate is defined by ("Serial Communication - learn.sparkfun.com", 2017) as the speed at which data is transmitted over a serial line and is usually expressed in bits per second or (bps). A recommended baud rate for use with microcontrollers such as the Arduino is 115200 bps as a higher rate would introduce errors since the MCU's chipset would not be able to keep up with commands. Since Processing utilises JAVA based code the student searched online to find code examples and tutorials on its use.

To better understand the use of Processing for point cloud production the student had found an example of code online from ("LIDAR - qcontinuum", 2017) which utilized two servo motors used in a tilt and pan setup to create a basic point cloud. The point cloud created from this example utilized the maximum 180° movement available from each servo motor to determine a normalized scale of both azimuth and elevation. The data fed back from these two motors were used to convert the spherical co-ordinates into cartesian points within 3 axes within a Processing sketch. The student then used this example as a basis for their own program as the following issues were clearly explained:

- Delimited input from Serial communications.
- Formation of a vector array.
- The use of RGB input to form colour changes in the Processing sketch.
- The generation of a point in space using the Stroke command.
- The use of key inputs in real time to tilt, pan and zoom in/out of the point cloud.

The student had found this example to contain most of the functionality required to produce a point-cloud thus it would be used as the basis of the student's program. The student had deemed in addition to this functionality points found should be written to a text file which could be accessible through MATLAB code which would be used for post scan analysis. Data flow may be seen in figure...

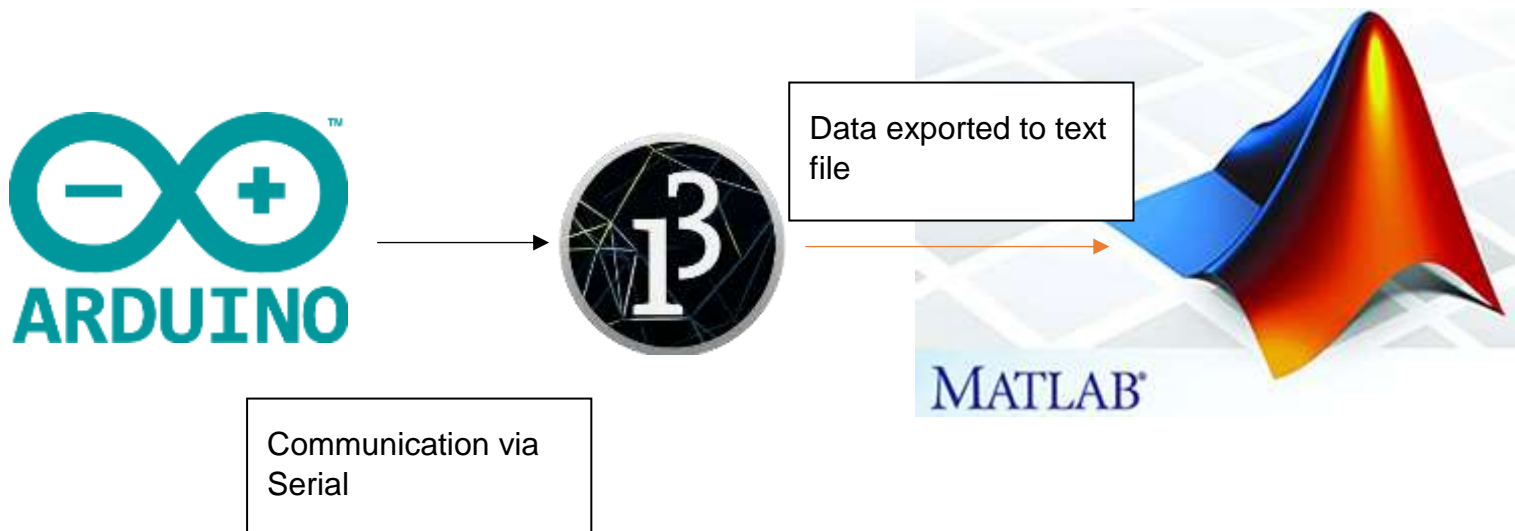


Figure 65: Program Flow

4.1.4 System Integration

After all the separate components within the design had been explained and expanded upon the processes should not be regarded as isolated processes. Each of the components within the Mechanical, Electrical and Information Technology aspects must work together towards a viable Mechatronics solution. The Project solution may be summed up as the following integrated operations which may be graphically shown in figure 65

- A Control command being input by a user for the scanning process.
- The MCU then receives this command for measurement and issues a protocol for measurement.
- Measurement is done via the Garmin Lidar Lite v3 sensor.
- Feedback of this sensor allows for the next mechanical movement to be made by either the stepper or servo motor.
- Stepper motor is driven by the Spark-fun Easy driver which is an electrical operation.
- Servo is driven by a PWM signal which is an electrical operation.
- Mechanical Movement relates back to an analogue value from the LDR and a digital value servo position from PWM signal which are electrical operations.
- The co-ordinate is then calculated via the MCU using spherical to cartesian coordinate translations which is an Information Technology operation.
- A Real Time Point cloud is generated via serial Communication with Processing a subsequent Information Technology operation
- A final process of data analysis done in MATLAB which is an additional Information Technology operation.

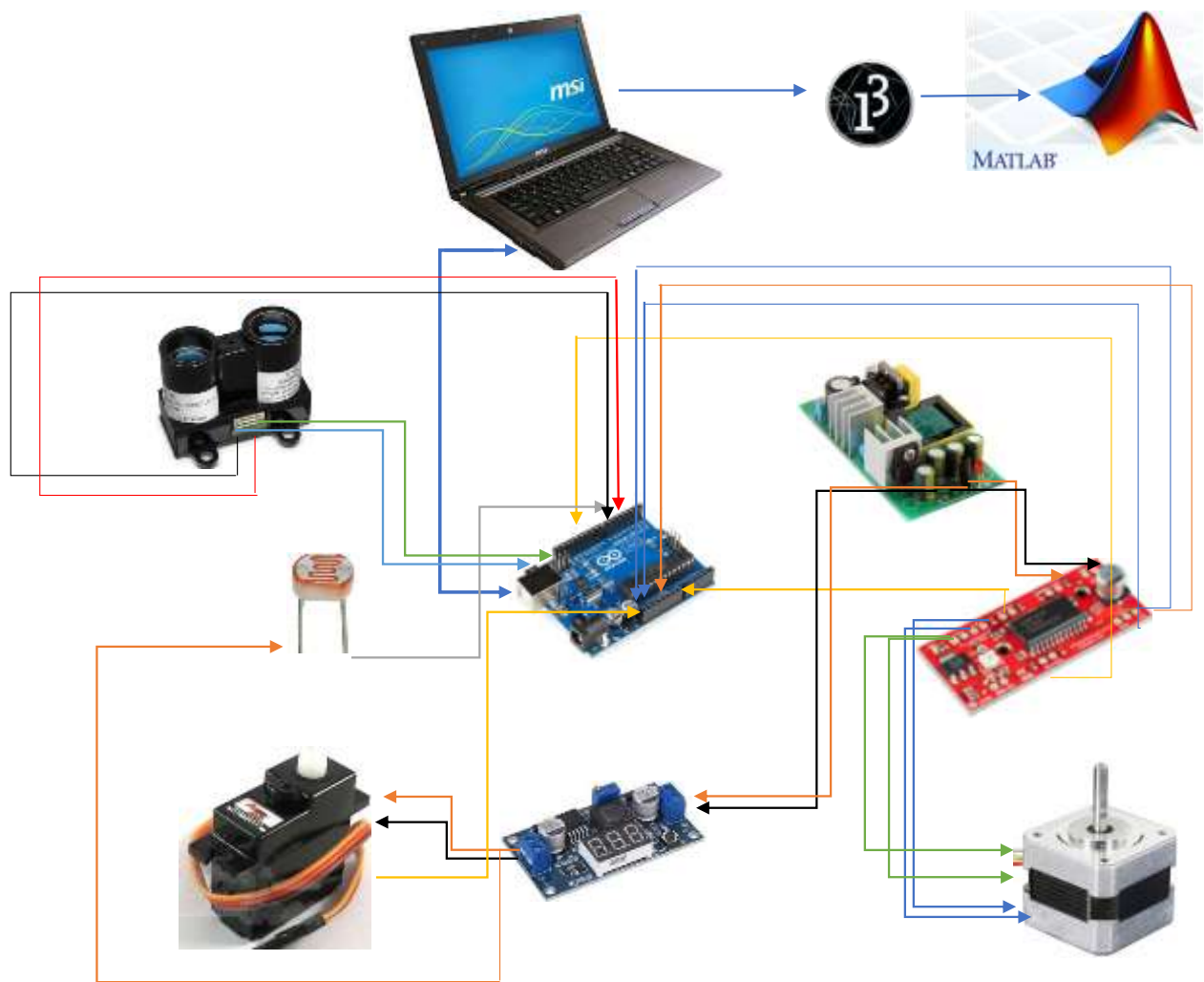
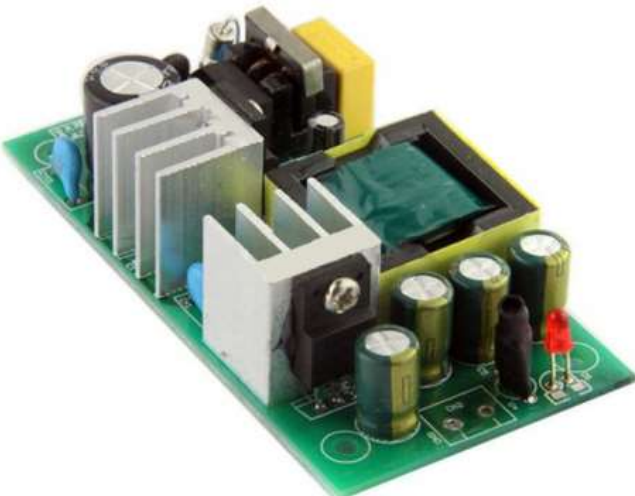



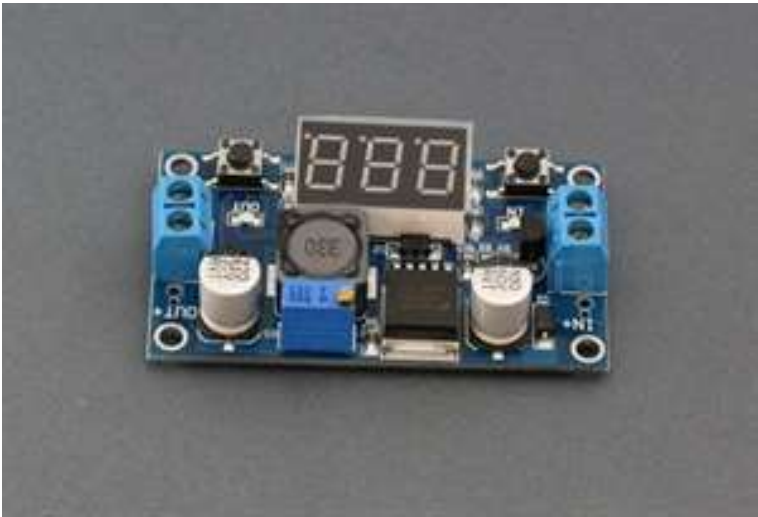
Figure 66: System Integration

4.2 Design Specification


4.2.1 Component Specification


Component 220V DC to AC module	Specifications
	<ul style="list-style-type: none">• Isolated industrial power module• Isolated industrial power modules• Input voltage: AC 85V - 265V 50/60HZ• Output voltage: DC12V ($\pm 0.2V$)• Output current: 2A• Output power 24W• Operating temperature: -20 ° C to +60 ° C• Relative humidity :40-90% RH without cooling No cooling• Input surge current: 20A• Output voltage range: 12.0~12.4V• Input Current:0.368 (AC 110V) 0.123 (AC 220V)• Output ripple: 120mV• Minimum current: 2A• Output efficiency: 85% Rated load• Switch Overshoot: 150%• Output rise time: 100mS• Dimensions:8.7cm x 4.6cm x 3cm

Component Power HD- 1160 A Servo	Specifications
	<ul style="list-style-type: none"> • Torque 2.7 kg/cm at 4.8V, 3 kg/cm at 6.0V • Speed 0.14 sec/60° at 4.8V, 0.12 sec/60° at 6.0V • Weight 16 g • Gear material Plastic • No Bearing • Cable length 150 mm • Operating Voltage DC 4.8V - 6V • Dimensions (H x W x L) 28 x 13.2 x 30.

Component DC-DC CV Step down Buck Power Module	Specifications
	<ul style="list-style-type: none"> • Input voltage range: 4-40VDC • Output voltage range: 1.25-37DC adjustable • Output current: 2A • Voltmeter range: 0 to 40V, error +-0.1V • Input reverse polarity protection • Built in output short protection function • Built in thermal shut down function • L x W x H=6.1 x 3.4 x1.2cm

Component Wantai 42BYGHW208 Bipolar Stepper Motor	Specifications
	<ul style="list-style-type: none"> • Step angle 1.8° • Motor Length 34 mm • Rated Voltage 12V • Rated Current 0.4A • Phase Resistance 30 ohm • Phase inductance 37 mH • Holding Torque 2800 g.cm • 4 Lead Wire

Component Spark fun Easy driver 4.4	Specifications
	<ul style="list-style-type: none"> • +-750mA ,30V output rating • Slatington Sink Drivers • Automatic Current – Decay Mode Detection/ Selection • 3.0v to 5.5 V logic supply Voltage Range • Cross over current protection • Mixed, Fast , and Slow current decay Modes

Component	Garmin Lidar Lite V3	Specifications
		<ul style="list-style-type: none"> • Range 40m • Resolution $\pm 1\text{cm}$ • Accuracy $< 5\text{m} \pm 2.5\text{cm}$ • Accuracy $> 5\text{m} \pm 10\text{cm}$ • Update rate 270Hz • Repetition Rate 50HZ • User interface PWM, I2C • Size LxWxH (20x48x40) • Weight 22g • Power 5V dc • Current Consumption 105mA idle

4.2.2 Device build methodology



Figure 67: ABS base

The 3-D printed ABS base would be the beginning of the devices construction as provision for each component has been made as seen in figure

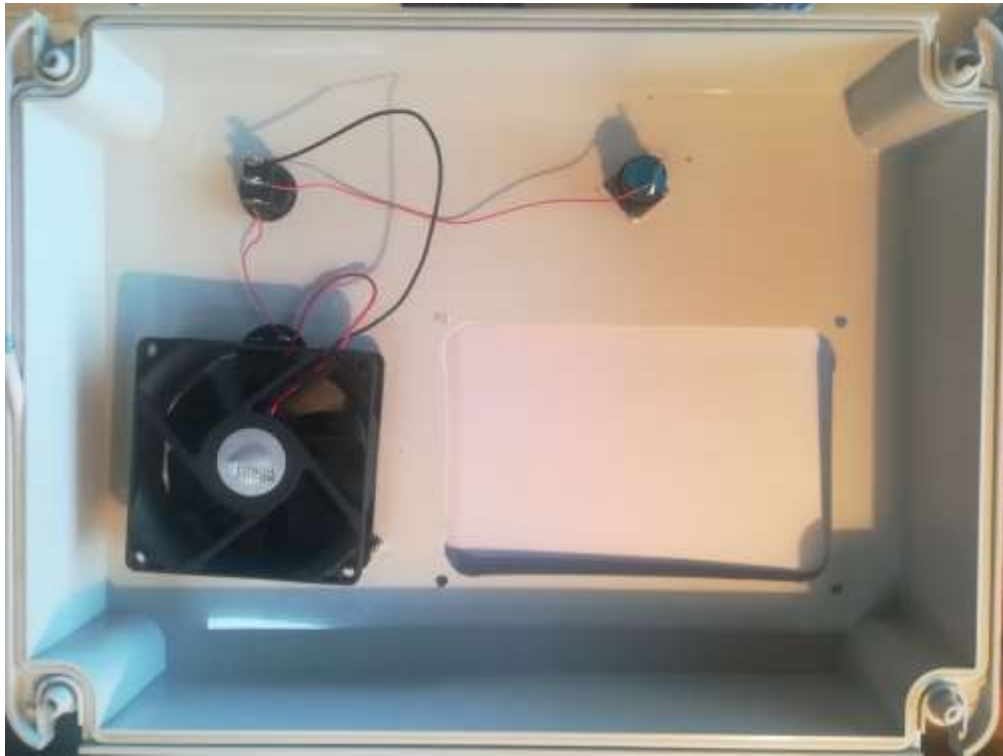


Buttons were soldered onto Veroboard and attached into the cut-outs provided by means of M2 screws seen in figure

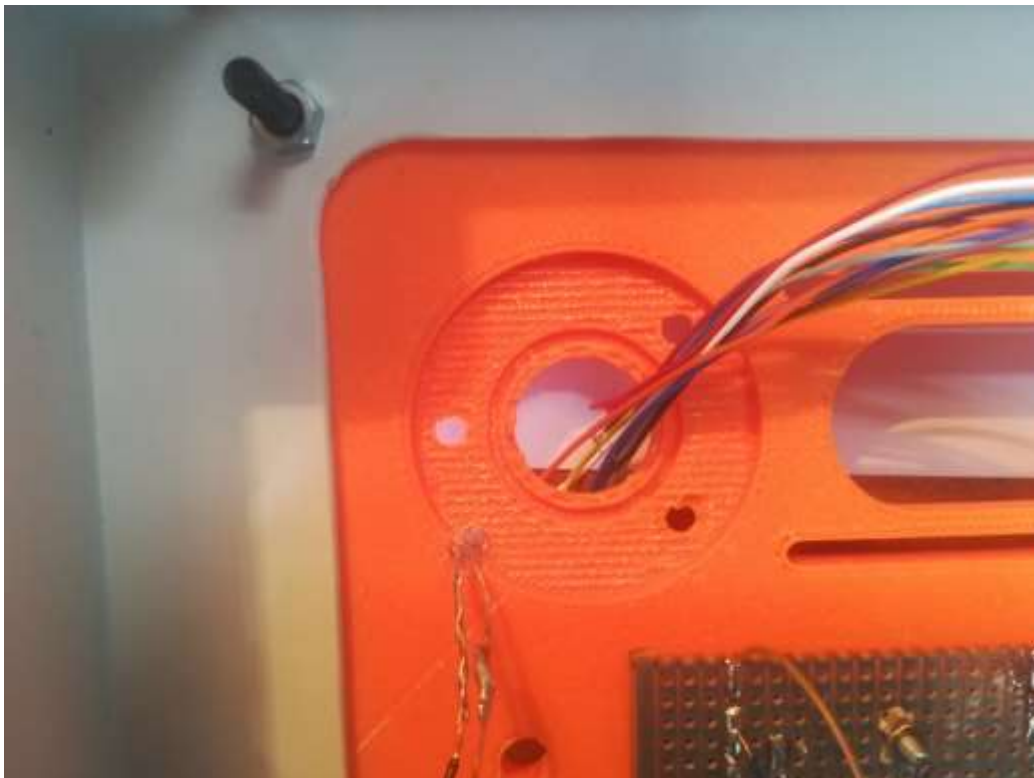


A

GEWISS Enclosure was cut out to accommodate the buttons and fan



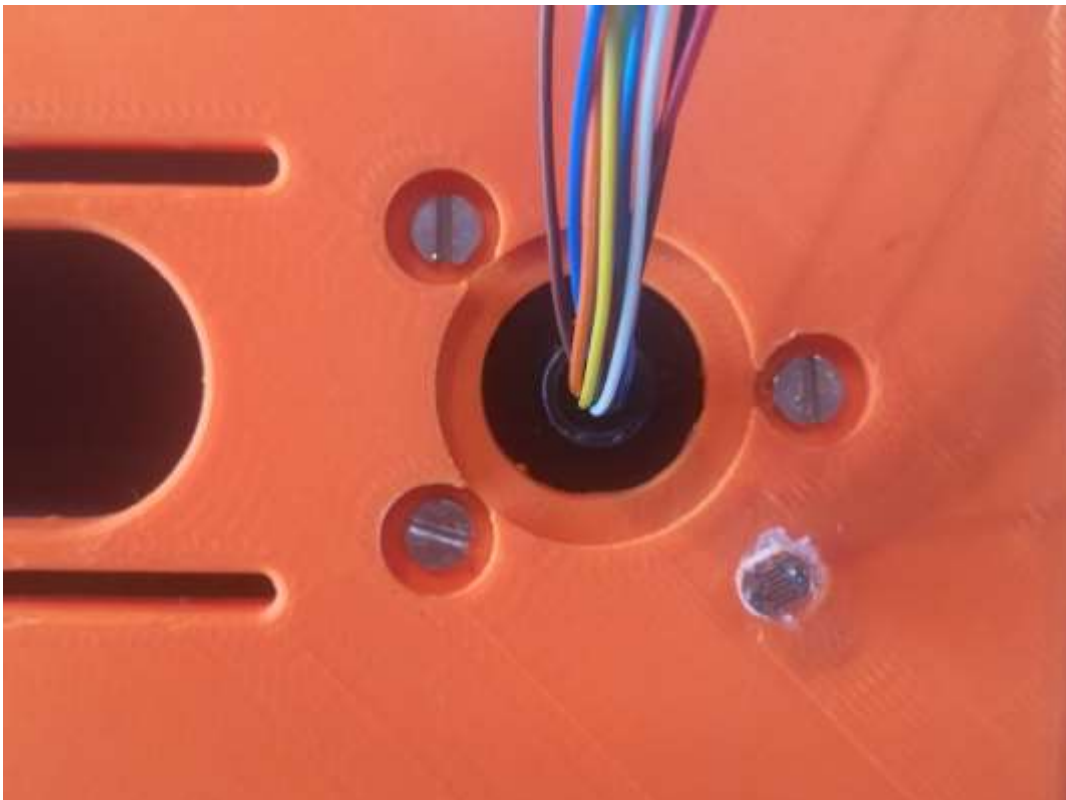
Fan Module is attached to the inner side of the GEWISS enclosure



LDR module Is attached to the inner side of the container



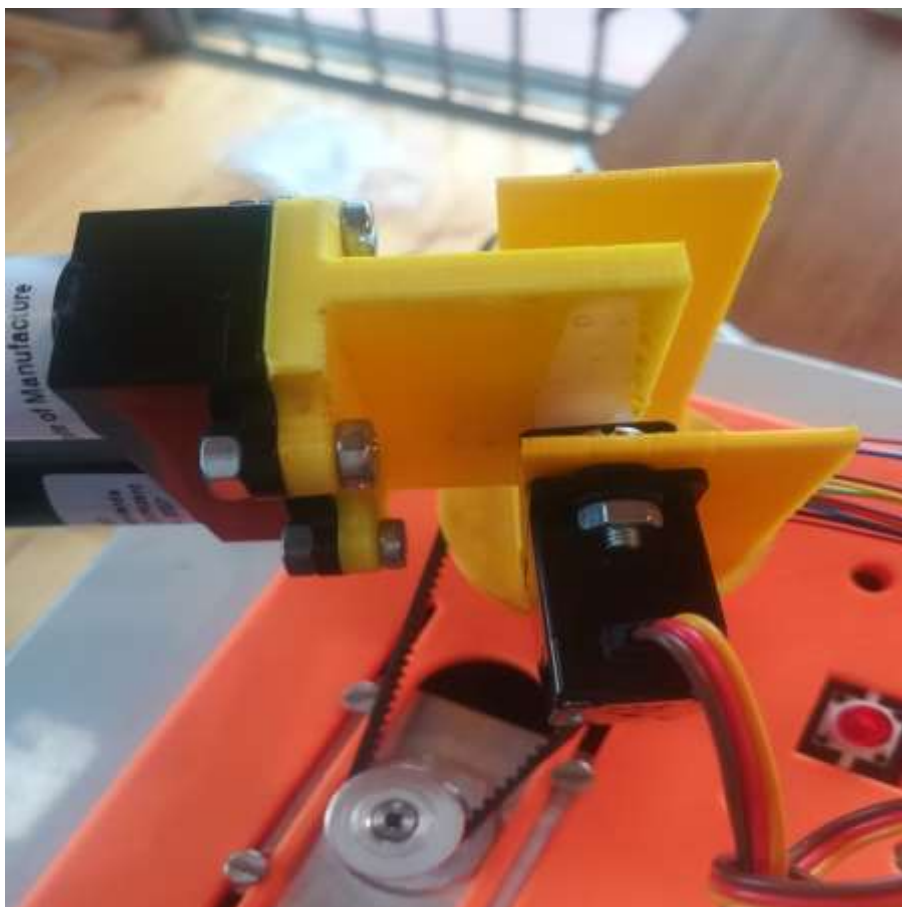
Base is then mounted to the GEWISS Enclosure using M4 screws



Slipring is mounted to the base by use of M3 Screws



The Garmin Lidar lite V3 is secured the mount by means of 4 M4 screws



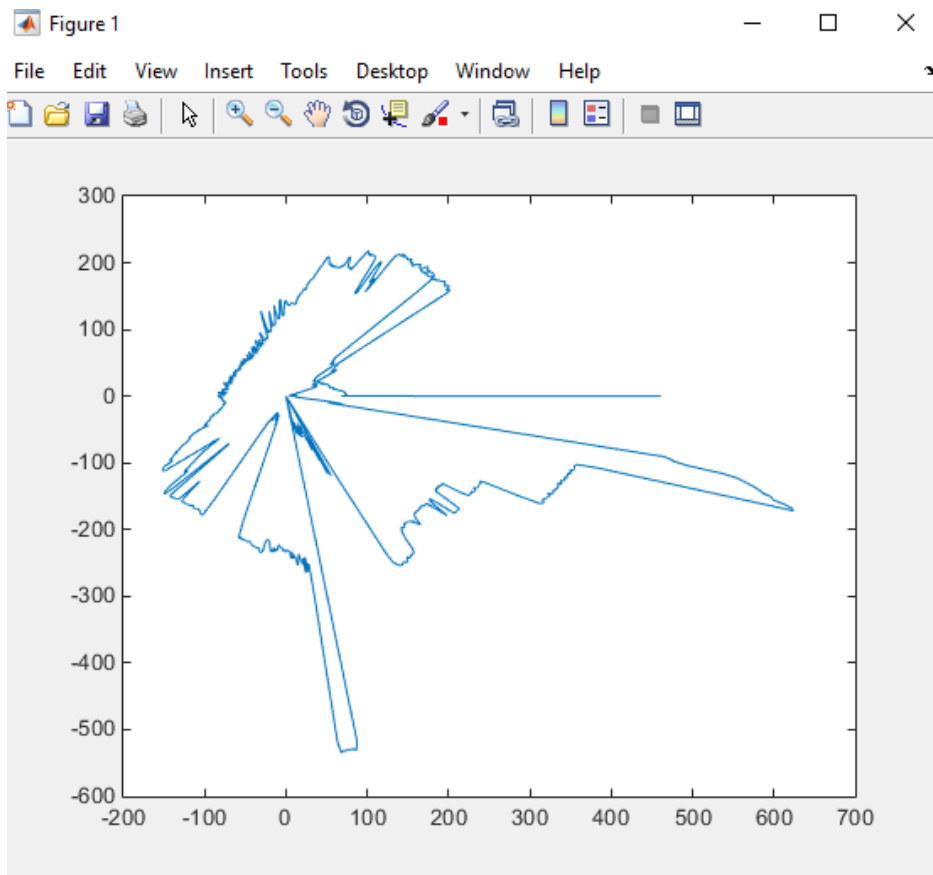


Figure 68: Example of a X, Y Mab produced by MATLAB

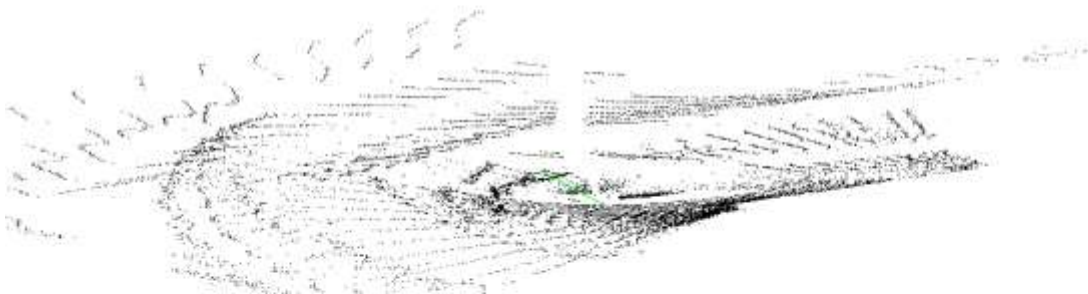


Figure 69: Example of point cloud output by the system



Figure 70: Final Designed Device

Figure 71: Device Final Design



Cost Analysis

Component	Number	Price	Total
Lidar Lite V3	2	R2938.92	R5877.84
220V Ac to DC Module	1	R118	R118
GEWISS enclosure	1	R493.95	R493.95
Gt2 pulley	1	R69	R69
Gt2 belt	2	R50	R50
E-stop	1	R40	R40
Solder iron	1	R40	R40
Buttons	1	R35	R35
PC Fan	1	R80	R80
100K Resistors	4	R2	R2
Easy driver stepper	1	R109	R109
DC-DC Step Down	1	R57	R57
Power HD servo	1	R168	R168
Total			R7138.81

The student had exceeded their budget for the project this has occurred as damage was sustained to the Lidar Lite V3 during construction. Th student had ran the easy-driver stepper 5V output to the HD servo and Garmin Lidar lite. A Back EMF had been caused by the servo drawing too much current subsequently the Garmin Lidar Lite sustained serious damage.

5. Performance Evaluation Experiments and Data analysis

5.1.1 Design stage uncertainty of Instruments

For an accurate experiment conditions must be kept as constant as possible.

Instruments that could influence readings determined from the system. In addition to these there exists external factors which could influence measurements such as:

- Instrumental errors
- Human errors
- Hysteresis errors
- The uneven placement the sensor with respect to the target
- The resolution of the measuring tape
- Errors made in the process procedure
- Uncertainty of the MCU

During the design stage, all elements which contribute towards the final measurement contribute an error. Thus, every element must be analysed correctly to find their individual uncertainty. Design stage uncertainty refers to an analysis performed prior to testing and calibration. This will provide a minimum uncertainty based on instruments used and methods chosen.

The zero -order uncertainty is the uncertainty that will exist even if all other errors are 0. This uncertainty is dependent on the resolution of the instrument. This can be calculated to a 95% probability by (Equation 5.1) (Beasley, F.a., 2011)

$$u_0 = \pm \frac{1}{2} \times \text{resolution} (P\%)$$

The resolution of an instrument is defined as the smallest change in quantity that the instrument can detect.

Instruments, when working towards a single final reading, induce uncertainty in the final reading. Each Instrument uncertainty, u_c must be combined by use of the Root Sum Squared (RSS) method: The overall error is given by (Equation 5.2) (Beasley, F.a., 2011):

$$u_x = \sqrt{\sum_{k=1}^K u_k^2} (P\%)$$

- Where K is the total number of instruments contributing and u_k is the individual error. u_k represents the errors of hysteresis, linearity, sensitivity, repeatability etc.

A combination of the zero-order uncertainty and the instrument errors give the overall design-stage uncertainty. The design-stage uncertainty, u_d , is used as a guide to select equipment, is given by (Equation 5.3) (Beasley, F.a., 2011)

$$u_d = \sqrt{u_o^2 + u_c^2} (P\%)$$

5.1.2.1 Uncertainty of Garmin Lidar Lite v3

From the specifications of the Garmin Lidar lite the following data was found

- Range: 70m
- Resolution: +/- 1cm
- Accuracy < 5 m: +/- 2.5cm
- Accuracy > 5m: +/- 10cm, Mean + - 1% of Distance Maximum, Ripple +- 1% of distance maximum

$$\begin{aligned} u_0 &= \pm \frac{1}{2} \times (\text{resolution}) \\ &= \pm \frac{1}{2} \times (1) \\ &= \mathbf{0.5cm(95\%)} \end{aligned}$$

For a measurement taken at maximum distance the uncertainty of accuracy and ripple voltage is

$$\begin{aligned} u_c &= \pm \sqrt{0.4^2 + 0.4^2} \\ &= \mathbf{0.5656cm} \end{aligned}$$

Hence the design stage uncertainty from Equation 5.3 equals

$$(\mu_d)_E = \pm \mathbf{0.7549cm}$$

5.1.2.1 Uncertainty of Digital data acquisition

Uncertainty of Arduino Uno

The uncertainty associated with digitising can be taken as the resolution of the LSB which is equal to the full-scale voltage divided by 2^N where N is equal to the number of bits used in the A/D process (Experimentation and Uncertainty Analysis for Engineers)

thus, for the Arduino the uncertainty can be calculated from:

$$u_0 = \frac{\text{Voltage Range}}{2^N}$$

$$u_0 = \frac{5 - (-5)}{2^{10}}$$

$$u_0 = 0.00976$$

Thus, the total Design stage uncertainty would be:

$$u_d = \pm\sqrt{0.5^2 + 0.7549^2 + 0.00976^2} = \mathbf{0.90552\text{ cm}}$$

5.2.1 Experiment Method of identifying reflective material

The experimental method determined by the student would be that of measuring actual distances up to 10m, although the LIDAR Lite V3 is rated up to 40m performance relationships could be determined at a shorter distance. The 10m range was measured by the student using a measuring tape at 0.5m increments seen in figure 69 these were then marked on a floor space ensuring that measurements were straight using laser directed from the viewpoint of the sensor.

A Baseline value would be determined for the sensor using a sheet of blank A4 printing paper as a non-reflective material seen in figure 70.

The material was attached to a hard-cover book as to ensure that the material could be correctly positioned on the exact point measured.



Figure 73: Sensor Baseline experiment

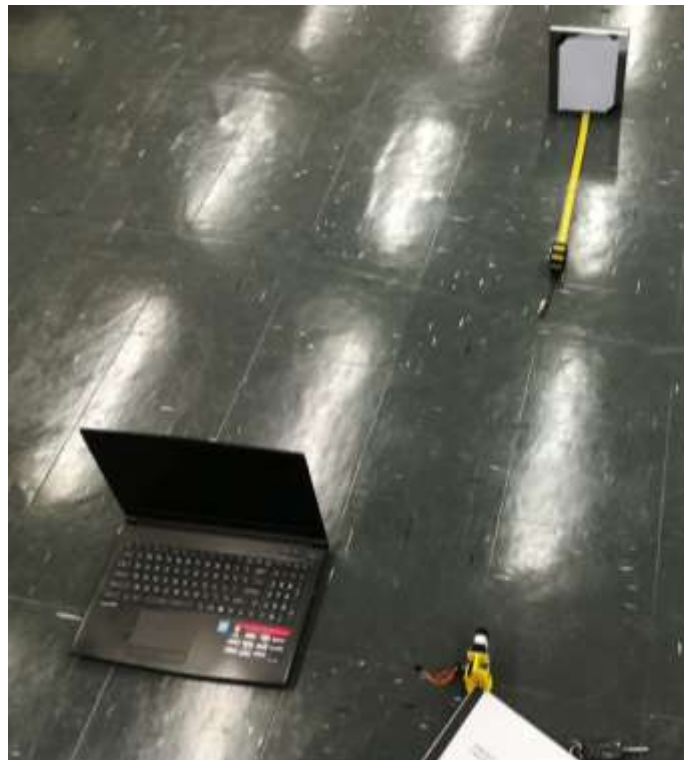


Figure 72: Sensor Test at different lengths

The Lidar Lite V3 was then setup at the 0-point with the servo control set to 0 degrees from where measurements were taken and connected to view readings of the characteristics for the sensor. The student first measured the characteristics of the material when positioned right in front of the sensor receiver at approximately 1 cm away as this is the minimal distance the sensor can read. Thereafter the student would position the material at 0.5m increments up to the 10m mark. There after values would be read from 10m back to 1cm to determine any possible hysteresis error present in the sensor. For each instance of measurement 5 readings were taken.



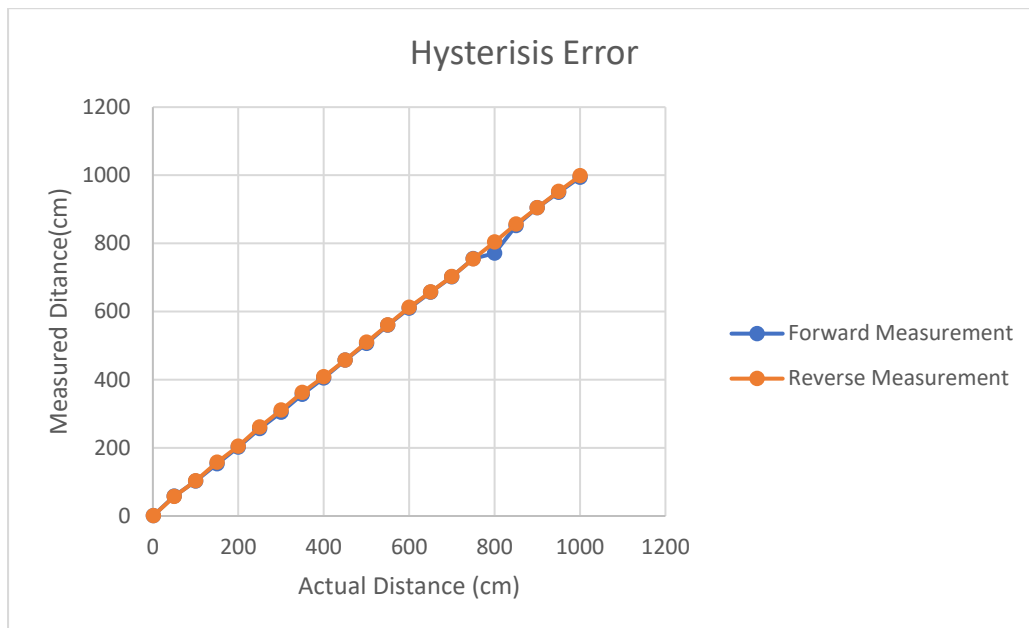
Figure 74: Reflective Material Calibration

The Measuring tape error is calculated by:

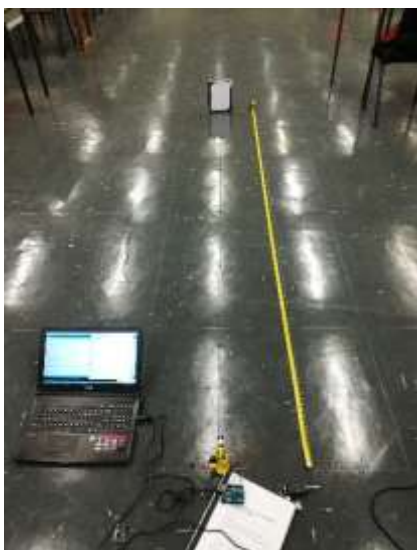
$$u_0 = \frac{\text{resolution}}{2}$$
$$= \mathbf{0.5mm}$$

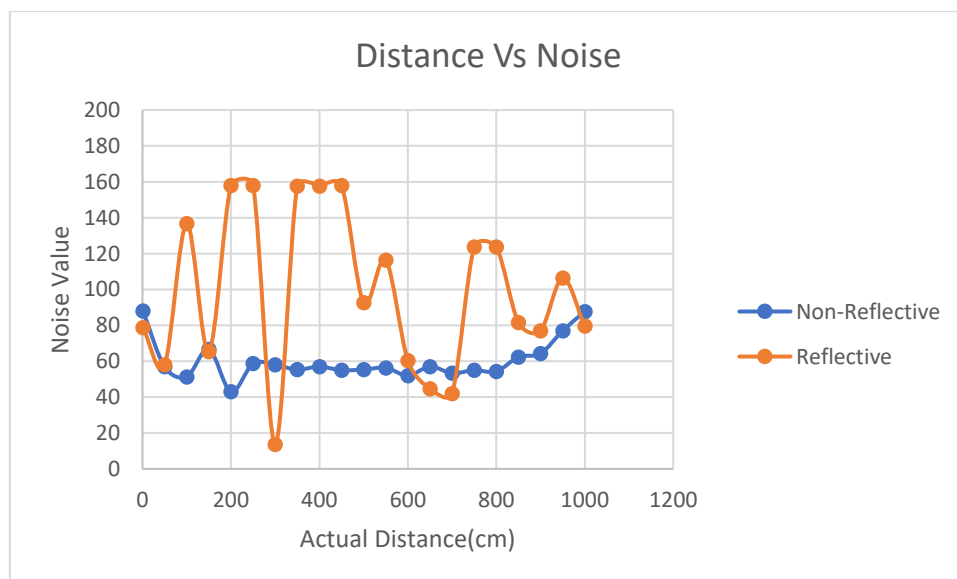
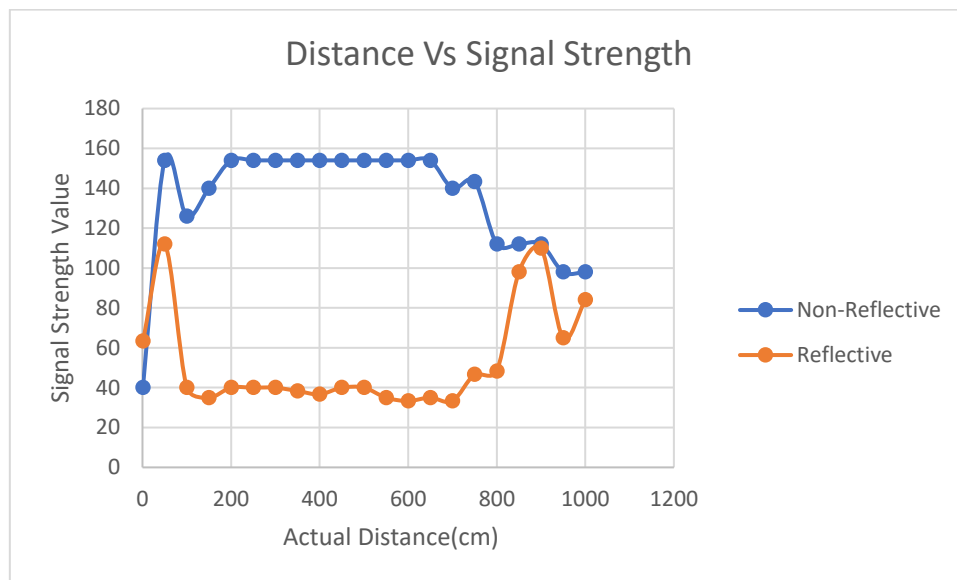
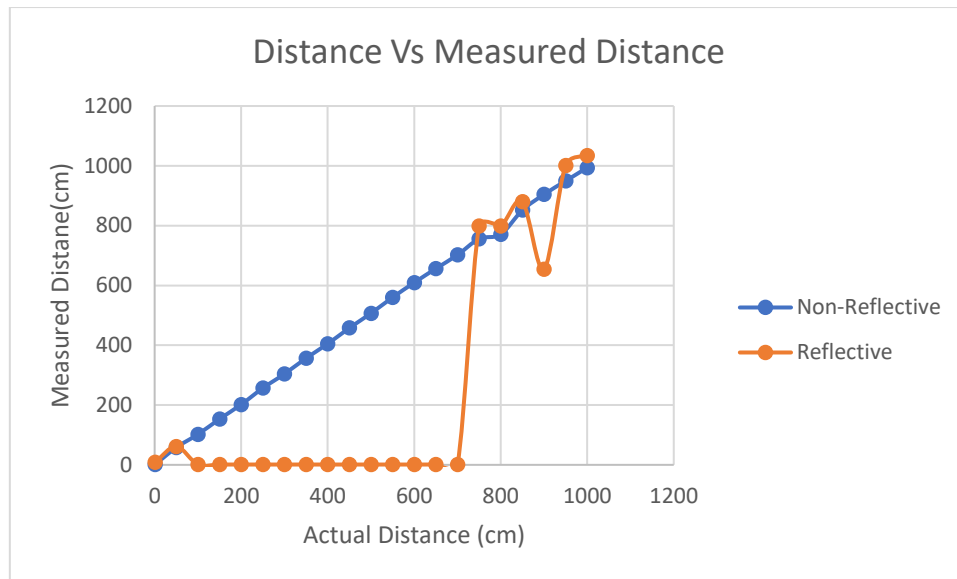
Total measurement range for this experiment would be consider signal strength of normal white paper

Hysteresis Error



Once the measurements were taken with the Non-Reflective material the student then repeated the measurements with the reflective tape present. This would allow for the student to determine the characteristic upon which to distinctly identify the reflective material.

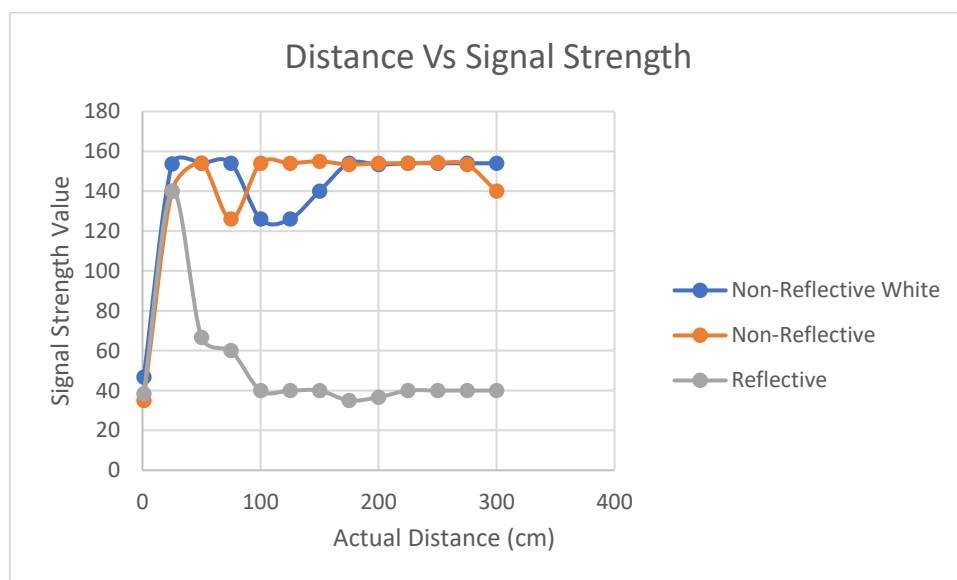
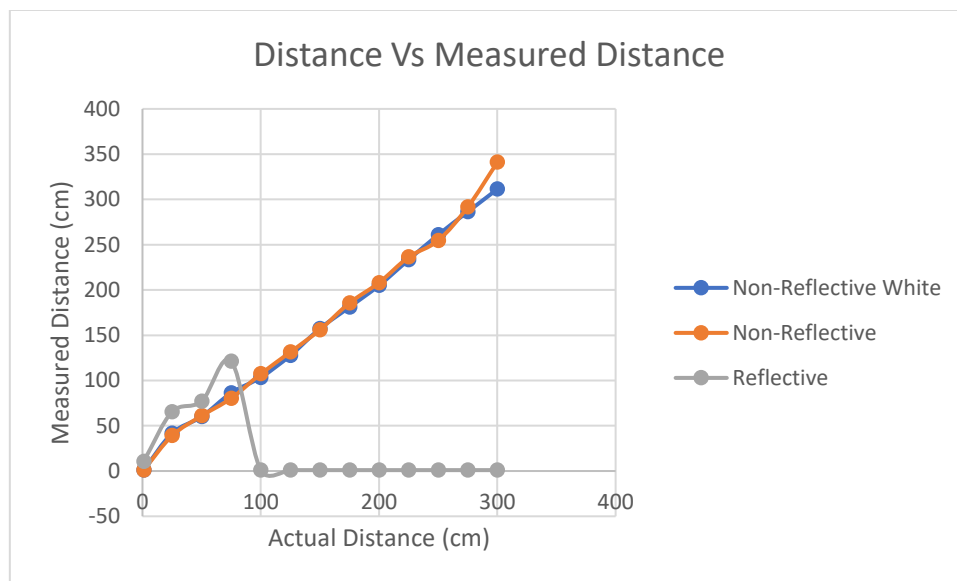


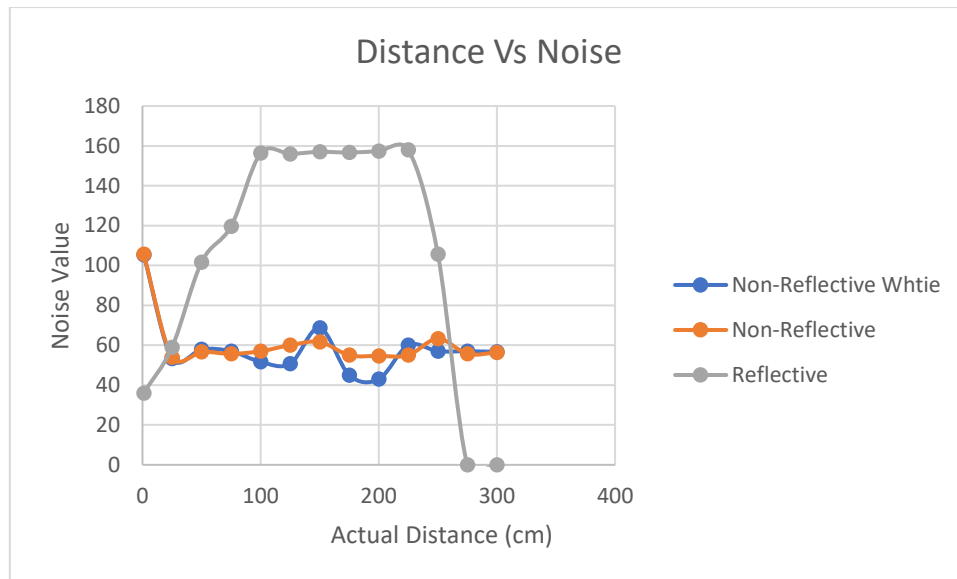


Experiment 2

For experiment 2 the same procedure was followed as that of experiment 1 but by reading values over a shorter distance and the inclusion of a non-reflective material other than that of white paper.

Distance measured was that up to 3m in 25cm increments which would allow for a better understanding of the sensor at lower distance measurements. Readings of the Sensors distance value, noise and signal strength were read. Since values fluctuated an average from 3 readings were taken.





Performance analysis

The Student was unable to determine a relationship between material reflectivity and the quality of the signal returned, thus the evaluation of reflective dependent land marks could not be accomplished. The accuracy of the Garmin Lidar lite how ever could be evaluated and was found to be highly accurate at distances greater than that of 1 to 10 cm. Reflectivity and size of the target highly influenced the quality of the measurement received from the Garmin Lidar Lite V3 as measurements were dependent on the return signal .

The Mechanical system of the Laser Navigation system could be evaluated by the student in its ability to reach the needed positions to form the point cloud thus this system achieved these requirements well.

Electrical performance of the system could be successful after damage had occurred to the first Garmin Lidar Lite sensor the student had to Isolate the functionalities of each of the output peripherals. Once the circuits were isolated no further damage was sustained to the workings of the device.

6. Conclusion

Many engineering aspects were dealt with within the design and analysis of the Laser Navigation System these were inclusive of design, manufacturing, and programming. The engineering knowledge gained by the student over the duration of the project could be seen as invaluable, as many problem-solving techniques and first principle methods of design were needed to be used.

According to the TPMs set out by the student the system is not fully successful as the localisation aspect of the project was not achieved, but the ability of the system to construct a point cloud was achieved. the system is lightweight and durable for operation within industry but the student would not see it as a viable solution in comparison the industrial specification devices such as that of the SICK systems.

The student had attempted to perform an alternative method of Landmark identification using a Neural Network with 3D shape analysis. The System would scan the surroundings gathering information of its current position whilst looking for pre-learned shapes. This solution was also found unusable as the Neural Network would need to individually learn these shapes in its surroundings and seek similarity within its future scans. This supervised learning method for navigation was found to be inefficient with deep learning algorithms would offer better solutions to navigational problems.

The student had determined that a more viable solution for the challenges of indoor navigation would be a system with high processing capacities and the ability to interact across the board with industrial machinery such as the use of a PLC system with access to the open source developments of ROS.

The system built by the student could be expanded upon for many different uses but the use for learning ROS programming being the most viable for its applications.

References

3D Universe. 2017. An Experiment: Using Dual Extrusion to 3D Print a Plastic Object with a Bronze Shell | 3D Universe. [ONLINE] Available at: <https://3duniverse.org/2014/06/12/an-experiment-using-dual-extrusion-to-3d-print-a-plastic-object-with-a-bronze-shell/>. [Accessed 01 June 2017].

A-vt.be. (2017). [online] Available at: <http://www.a-vt.be/images/site/automatisch-geleid-voertuig-transport-2.jpg> [Accessed 16 May 2017].

AGVS. 2017. AGVS. [ONLINE] Available at: <http://www.mhi.org/agvs>. [Accessed 11 May 2017]

Automated guided vehicle: laser navigation with reflectors | SICK. 2017. Automated guided vehicle: laser navigation with reflectors | SICK. [ONLINE] Available at: <https://www.sick.com/us/en/product-applications/automated-guided-vehicle-laser-navigation-with-reflectors/c/p170640>. [Accessed 12 April 2017].

Beasley, F. a., 2011. Theory and Design for Mechanical Measurements. 5 ed. s.l.:John Wiley & Sons.

Carullo, A. and Parvis, M. (2001). An ultrasonic sensor for distance measurement in automotive applications. IEEE Sensors Journal, 1(2), p.143.

Christopher M. K., T. W. (2011). Designing a System for Mobile Manipulation from an Unmanned Aerial Vehicle. 2011 IEEE Conference, 109 - 114.

Di, K., Zhao, Q., Wan, W., Wang, Y. and Gao, Y. (2016). RGB-D SLAM Based on Extended Bundle Adjustment with 2D and 3D Information. Sensors, 16(8), p.1285.

FOS Mechanical Engineering. 2017. *selection of factor of safety Archives | Mechanical Engineering*. [ONLINE] Available at: <http://www.mechanicalengineeringblog.com/tag/selection-of-factor-of-safety/>. [Accessed 29 May 2017].

GrabCAD - CAD library. 2017. GrabCAD - CAD library. [ONLINE] Available at: <https://grabcad.com/library/stepper-motor-35>. [Accessed 18 June 2017].

Gewiss 300x220x120mm IP56 PVC Enclosure. 2017. Available: https://www.dungannonelectrical.co.uk/Search-Results?prodref=GW44209&proddesc=Gewiss-300x220x120mm-IP56-PVC-Enclosure&category=pg_PVC+ENC166508&catdesc=PVC-Enclosures [2017 , November 02].

Icons8. 2017. Icons8 - 52,900 icons + Icon Tools. [ONLINE] Available at: <https://icons8.com/icon/set/Servo%20motor/all>. [Accessed 18 June 2017].

Jarvis, R., Mason, A., Thornhill, K. & Zhang, B. 2011. Indoor Positioning System. MEng. LouisianaState University.

International Lidar Mapping Forum. (2008). The Photogrammetric Record, 23(123), pp.342-342.

Kamarudin, K., Mamduh, S., Shakaff, A. and Zakaria, A. (2014). Performance Analysis of the Microsoft Kinect Sensor for 2D Simultaneous Localization and Mapping (SLAM) Techniques. *Sensors*, 14(12), pp.23365-23387.

Kumar, K Design of Automated Guided Vehicles, *International Journal of Mechanical Engineering* 1(3), April 2012.

Khan, R., Khan, S., Khan, S. and Khan, M. (2014). Localization Performance Evaluation of Extended Kalman Filter in Wireless Sensors Network. *Procedia Computer Science*, 32, pp.117-124

Kumar Das, S. (2016). Design and Methodology of Automated Guided Vehicle-A Review. *IOSR Journal of Mechanical and Civil Engineering*, 03(03), pp.29-35.

LIDAR-lite V3 - Micro Robotics . 2017. LIDAR-lite V3 - Micro Robotics . [ONLINE] Available at: <https://www.robotics.org.za/lidar-lite-v3-sen-14032?search=lidar>. [Accessed 06 May 2017].

LIDAR-Lite 3 Laser Rangefinder - RobotShop. 2017. LIDAR-Lite 3 Laser Rangefinder - RobotShop. [ONLINE] Available at: <http://www.robotshop.com/en/lidar-lite-3-laser-rangefinder.html>. [Accessed 02 June 2017].

<http://www.mio.com/technology-trilateration.htm> [Accessed 3 Nov. 2017].

MACFOS. 2017. HC-SR04-Ultrasonic Range Finder - Robu.in | Indian Online Store | RC Hobby | Robotics. [ONLINE] Available at: <https://robu.in/product/hc-sr04-ultrasonic-range-finder/>. [Accessed 18 June 2017].

Mohan, N, 2002. *Power Electronics: Converters, Applications, and Design*. 3. Wiley

NAV350-3232 SICK • Sensors by INT TECHNICS. 2017. NAV350-3232 SICK • Sensors by INT TECHNICS. [ONLINE] Available at: <http://sensorstrade.com/mpn/nav350-3232/>. [Accessed 03 June 2017].

NAV350-3232 Sick Laser scanners | ELTRA TRADE. 2017. NAV350-3232 Sick Laser scanners | ELTRA TRADE. [ONLINE] Available at: <http://eltra-trade.com/products/Sick-NAV350-3232>. [Accessed 02 June 2017].

Online Diagram Software to draw Flowcharts, UML & more | Creately. 2017. Online Diagram Software to draw Flowcharts, UML & more | Creately. [ONLINE] Available at: <https://creately.com/>. [Accessed 09 May 2017].

Proxaut. 2017. Navigation Systems Archivi - Proxaut. [ONLINE] Available at: <http://www.proxaut.eu/en/technology-categories/navigation-systems/>. [Accessed 31 May 2017].

Processing.org. (2017). *Environment (IDE) \ Processing.org*. [online] Available at: <https://processing.org/reference/environment/> [Accessed 7 Oct. 2017].

Quigley, Morgan, Brian Gerkey, Ken Conley, Josh Faust, Tully Foote, Jeremy Leibs, Eric Berger, Rob Wheeler and Andrew Ng. "ROS: an open-source Robot Operating System." (2009).

Riisgaard, S. & Blas, M. R. (2005), 'SLAM for Dummies: A Tutorial Approach to Simultaneous Localization and Mapping '.

Richard Zhi-Ling Hu, A. H. (2011). 3D Pose Tracking of Walker Users' Lower Limb with a Structured-Light Camera on a Moving Platform. Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on , 29 - 36 .

Schmidt, A., Fularz, M., Kraft, M., Kasiński, A. and Nowicki, M. (2013). An Indoor RGB-D Dataset for the Evaluation of Robot Navigation Algorithms. Advanced Concepts for Intelligent Vision Systems, pp.321-329.

Thingiverse.com. 2017. DGServo S05NF STD - Servo motor for prototyping by Cientifico_Loco - Thingiverse. [ONLINE] Available at: <https://www.thingiverse.com/thing:1326923>. [Accessed 18 June 2017].

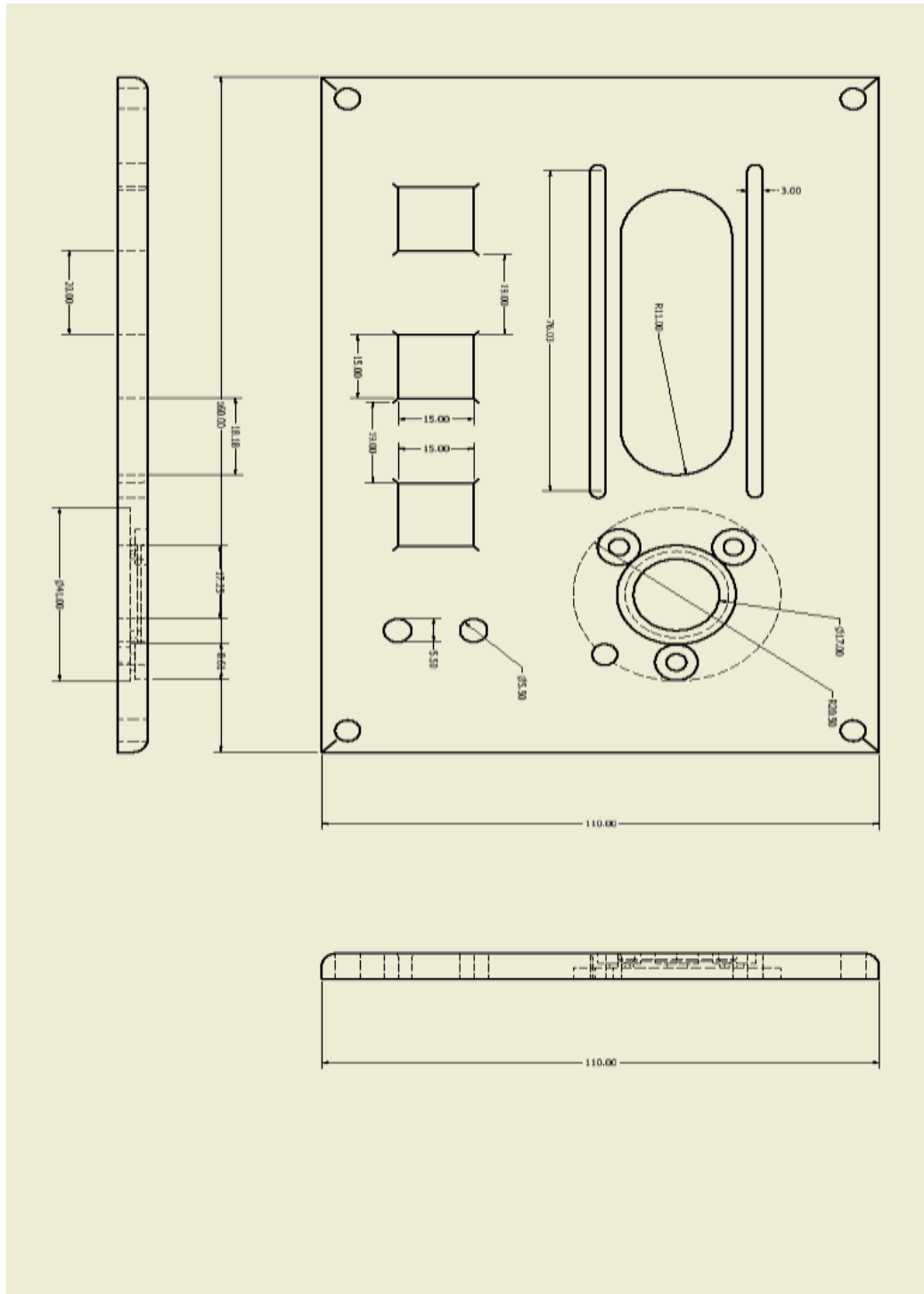
Webcam Based DIY Laser Rangefinder - Todd Danko. 2017. *Webcam Based DIY Laser Rangefinder - Todd Danko*. [ONLINE] Available at: https://sites.google.com/site/todddanko/home/webcam_laser_ranger. [Accessed 18 June 2017].

Webcam Based DIY Laser Rangefinder - Todd Danko. 2017. Webcam Based DIY Laser Rangefinder - Todd Danko. [ONLINE] Available at: https://sites.google.com/site/todddanko/home/webcam_laser_ranger. [Accessed 06 June 2017].

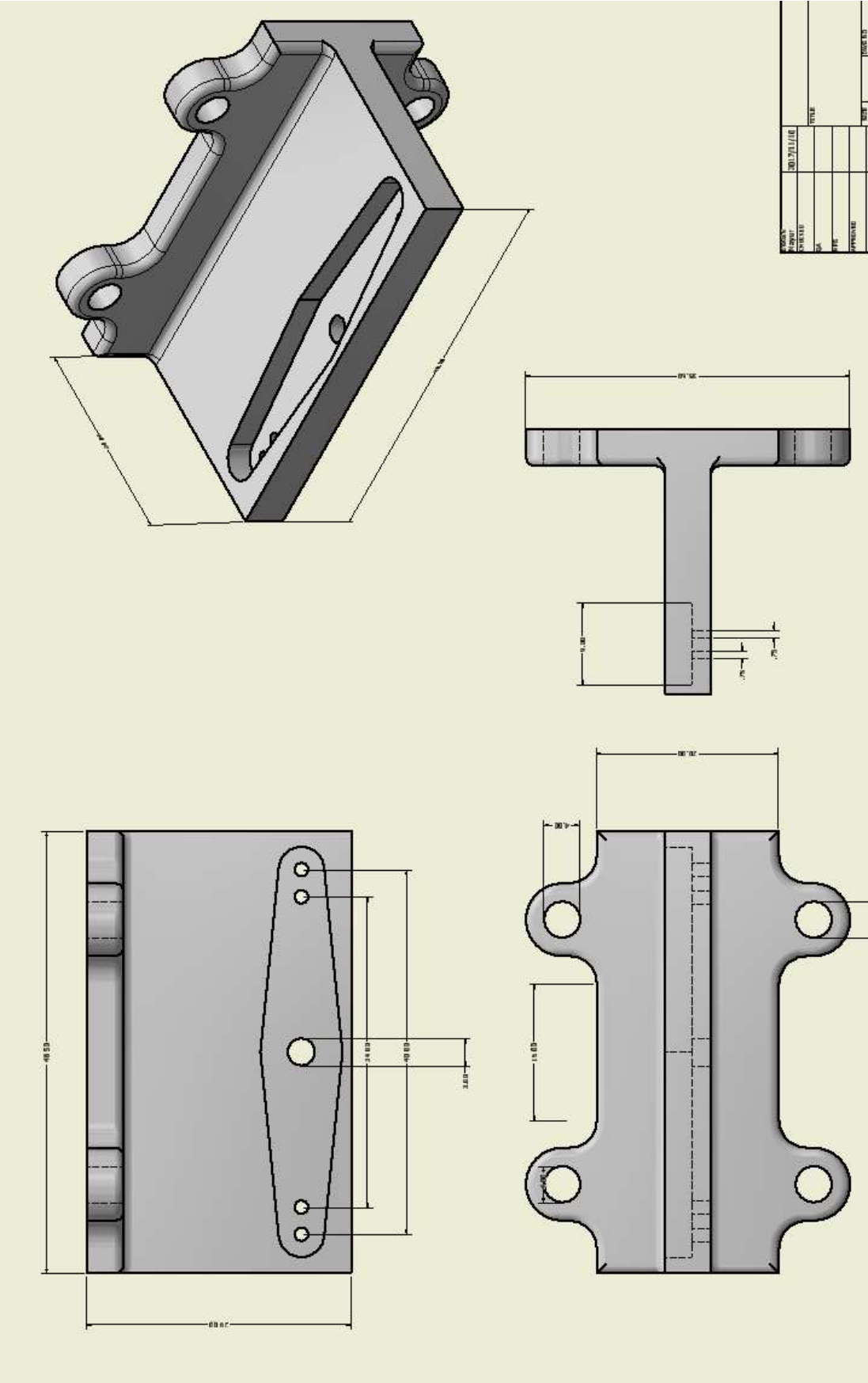
WU, X. (2011). Integrated Motion Control of Path Tracking and Servo Control for an Automated Guided Vehicle. Journal of Mechanical Engineering, 47(03), p.43.

Appendix A Working Drawings

Base Drawing



Mount



Appendix B Force Calculations

Force applied to the Mount:

$$F = mg$$

$$F = 0.022 \times 9.81$$

$$F = 0.21582 \text{ N}$$

Force applied to Bracket:

$$F = mg$$

$$F = (0.022 \times 9.81) + (0.01 \times 9.81) + (0.016 \times 9.81) + (0.016 \times 9.81)$$

$$F = 0.62784 \text{ N}$$

Moment about the y-axis:

$$M_y = Fl$$

$$M_y = 0.62784 \times 0.016$$

$$M_y = 0.01004 \text{ N.m} = 10.045 \text{ N.mm}$$

Moment about the x-axis:

$$M_x = Fl$$

$$M_x = 0.62784 \times 0.007$$

$$M_y = 0.004394 \text{ N.m} = 4.3948 \text{ N.mm}$$

Force applied to the Base:

$$F = mg$$

$$F = (0.022 \times 9.81) + (0.01 \times 9.81) + (0.012 \times 9.81) + (0.02 \times 9.81) + (0.014 \times 9.81) + (0.024 \times 9.81) + (0.2 \times 9.81)$$

$$F = 2.96262 \text{ N}$$

Output torque

$$M = nFl$$

$$M = 3 \times 0.2745862$$

$$M = 0.8237 \text{ N.m} = 823.7586 \text{ N.mm}$$

Appendix C Torque Calculation for actuation system

The effective tension in the belt is assumed to be equivalent to 500g of force.

$$T_e = T_1 - T_2 = 4.905 \text{ N} = \text{equivalent of 500g of force}$$

$$M = T_e \cdot \frac{d1}{2}$$

M = driving torque of the driven pulley (N.m)

$d1$ = driver pulley pitch diameter (12.73 mm taken from GT2 specifications for 20 tooth 5mm bore)

$$M = 4.905 \cdot \frac{0.01273}{2}$$

$$M = 0.03122 \text{ N.m}$$

Since:

$$P = T\omega$$

Power developed from motor:

$$P = VI$$

From 42BYGHW208 Stepper motor datasheet $V = 12 \text{ V}$ and $I = 0.4 \text{ A}$

$$P = 12 \times 0.4$$

$$P = 4.8 \text{ W}$$

Thus

$$\omega = \frac{P}{T}$$

$$\omega = \frac{4.8}{0.03122}$$

$$\omega = 153.747 \text{ rad.s}^{-1}$$

Thus, output torque is

$$\omega_2 = \omega_1 \cdot \frac{d1}{d2}$$

ω_2 = angular velocity of the driven pulley in (rad.s^{-1})

ω_1 = angular velocity of the drive pulley in (rad.s^{-1})

$d1$ = pitch diameter of the drive pulley = 12.73 mm

$d2$ = pitch diameter of the driven pulley 38.2 mm

$$\omega_2 = 153.747 \cdot \frac{12.73}{38.2}$$

$$\omega_2 = 51.2357 \text{ rad.s}^{-1}$$

$$\omega_2 \ll 1696.46 \text{ rad.s}^{-1}$$

Thus, the sensor would be able to update during maximum speed of the synchronous pulley operation.

From datasheet of slip ring max operating speed is 300rpm

$$\omega = \frac{\pi \cdot n}{30}$$

$$\omega = \frac{\pi \cdot 300}{30}$$

$$\omega = 31.4159 \text{ rad.s}^{-1}$$

Thus, limiting factor for speed is the slip ring, stepper motor should not travel faster than $\omega = 31.4159 \text{ rad.s}^{-1}$

$$M_1 = T_e \cdot \frac{d1}{2} = \frac{M_2}{\eta} \cdot \frac{d1}{d2}$$

$M_1 = \text{driving torque of the driven pulley (N.m)}$

$M_2 = \text{torque requirement}$

$\eta = \text{efficiency of belt drive (usually } 0.94 - 0.96)$

$$M_2 = \eta M_1 \frac{d2}{d1}$$

$$M_2 = 0.95 \times 0.03122 \times \frac{38.2}{12.73}$$

$$M_2 = 0.089 \text{ N.m}$$

Appendix D Thyristor Calculations

$$\begin{aligned} V_d &= \frac{1}{\pi} \int_{\alpha}^{\pi+\alpha} \sqrt{2} V_s \sin \omega t d(\omega t) \\ &= \frac{2\sqrt{2}}{\pi} V_s \cos \alpha \\ &= 0.9 V_s \cos \alpha \end{aligned}$$

Where:

V_s = the supply voltage

V_d = rectified output from the system

α = firing angle of the thyristor

Firing angle can thus be calculated as:

$$\begin{aligned} \alpha &= \cos^{-1} \left(\frac{V_d}{0.9 V_s} \right) \\ \alpha &= \cos^{-1} \left(\frac{12}{0.9 \times 220} \right) \\ \alpha &= 86.6^\circ \end{aligned}$$

A delayed firing angle will be:

$$\alpha = 86.6^\circ + 180^\circ$$

To Achieve a 1 Amp supply from the rectified signal a 12 Ω resistor may be added. To pulse the thyristors, voltage-pulse generators were necessary in the design. These voltage pulse generators would act at a time delay derived from the firing angle and delayed firing angle. The alternating voltage would be 0V to 1V within the time calculated below.

$$\alpha = \omega t$$

Where:

ω = angular velocity in rad.s^{-1}

t = time in seconds

Thus

$$t = \frac{\alpha}{\omega}$$

And we know that $\omega = 2 \pi f$ and frequency is 50 Hz

Thus

$$Td = \frac{86.5 \times \frac{\pi}{180}}{2\pi \times 50} = 4.805 \text{ ms}$$

This is the delay time that must be applied to T1 and T2 via the voltage pulse.

And

$$Td = \frac{266.5 \times \frac{\pi}{180}}{2\pi \times 50} = 14.805 \text{ ms}$$

This is the time delay that must be applied to T3 and T4 via the voltage pulse.

After implementing this circuit, a ripple voltage was observed thus to smooth out the output voltage a 15-mF capacitor was inserted to smooth the output from the rectifier.

Appendix E Spherical Coordinates Calculation

If the point returned from the device had the following characteristics:

- A relative output gear position of 200°.
- A normalised servo position of 40°.
- A distance value of 2.5m (returned in cm as 250 cm since lidar lite resolution is cm).

$$\theta = 200 \cdot \frac{\pi}{180} = 3.4906 \text{ rad}$$

$$\phi = (90 - 40) \cdot \frac{\pi}{180} = 0.87266 \text{ rad}$$

$$\rho = 250 \text{ cm}$$

$$x = 250 \cdot \sin(0.87266) \cdot \cos(3.4906)$$

$$x = -179.964 \text{ cm}$$

$$y = 250 \cdot \sin(0.87266) \cdot \sin(3.4906)$$

$$y = -65.489 \text{ cm}$$

$$z = 250 \cdot \cos(0.87266)$$

$$z = 160.6977$$

Appendix F Arduino Code

```
#include <LIDARLite.h>
```

```
#include <Servo.h>
```

```
#define Dir_pin 7
```

```
#define Step_pin 6
```

```
#define MS1 1
```

```
#define MS2 9
```

```
#define Enb 10
```

```
#define Mon 12
```

```
#define Trg 11
```

```
#define GLED 13
```

```
#define BLED 8
```

```
#define Gbtn 2
```

```
#define Rbtn 3
```

```
#define Wbtn 4
```

```
#define LL_Ad 0x62
```

```
Servo S1;
```

```
LIDARLite L1;
```

```
const int Ldr = A1;
```

```
int Ldv = 0;
```

```
int xi = 0;
```

```

int wbpc = 0; //mode select counter
int wbs = 0; //mode button state
int wbls = 0;

const float pi = 3.14159265;
const float d2r = pi/180;
const float s2g = 0.2903225806; //0.15 for qs
int TSC = 0;
int SC = 0;

float R = 0;

float Maxldr = 0;
float curLd = 0;
unsigned long pulseWidth;
int Gtn = 0;

volatile int Rbs = 0;

void setup() {
    // initialize the LED pin as an output:

    Serial.begin(115200);
    S1.attach(5);

```

```
pinMode(Trg,OUTPUT);  
digitalWrite(Mon,LOW);  
pinMode(Mon,INPUT);
```

```
pinMode(Gbtn,INPUT);  
pinMode(Wbtn,INPUT);  
pinMode(Rbtn,INPUT);
```

```
pinMode(GLED,OUTPUT);  
pinMode(BLED,OUTPUT);  
digitalWrite(GLED,LOW);  
digitalWrite(BLED,LOW);
```

```
pinMode(MS1 , OUTPUT); //halfstep  
pinMode(MS2 , OUTPUT); //halfstep  
pinMode(Step_pin , OUTPUT); //step pin  
pinMode(Dir_pin , OUTPUT); //clockwise / antclockwise  
pinMode(Enb, OUTPUT);  
//  
digitalWrite(MS1,HIGH);  
//HS //LOW Qs  
digitalWrite(MS2, LOW);  
//HS //HIGH Qs
```

```
xi = S1.read();
```

```

Serial.println(xi);
if (xi != 19)
{
    S1.write(19); //Zero pos servo
    delay(400);
}
Maxldr = analogRead(Ldr);

//attachInterrupt(1,halt, CHANGE);
}

void loop()
{

//Calibrate();

wbs = digitalRead(Wbtn);

if( wbs != wbls)
{
    if(wbs == HIGH)
    {
        wbpc++;
    }

    delay(50);
}
wbls = wbs;

```

```
while (digitalRead(Gbtn) != HIGH)
```

```
{
```

```
  if (wbpc % 2 == 0)
```

```
{
```

```
  digitalWrite(BLED,HIGH);
```

```
  digitalWrite(GLED,LOW);
```

```
}
```

```
else
```

```
{
```

```
  digitalWrite(GLED,HIGH);
```

```
  digitalWrite(BLED,LOW);
```

```
}
```

```
  if (digitalRead(Wbtn) == HIGH)
```

```
  {
```

```
    wbpc++;
```

```
  }
```

```
  delay(100);
```

```
}
```

```
if(wbpc % 2 == 0)
```

```
{
```

```
  Lscan();
```

```
}
```

```
else
```

```
{  
  Tscan();  
}  
  
}
```

```
void Step()  
{  
  
  digitalWrite(Dir_pin,HIGH);  
  digitalWrite(Step_pin,HIGH);  
  delay(1);  
  digitalWrite(Step_pin,LOW);  
  delay(2);  
}
```

```
void Calibrate()  
{  
  Ldv = analogRead(Ldr);  
  curLd = Ldv/Maxldr;  
  while(curLd > 0.8 || curLd < 1.2)  
  {  
    Step();  
    delay(10);  
  }
```

```

    Ldv = analogRead(Ldr);
    curLd = Ldv/Maxldr;
}
// while(Ldv <= 120)
// {
//   Step();
//   delay(10);
//   Ldv = analogRead(Ldr);
// }
//
// while(Ldv > 100)
// {
//   Step();
//   SC++;
//   Ldv = analogRead(Ldr);
// }
// }
}

```

```

void Lscan()
{
    digitalWrite(BLED,HIGH);
    digitalWrite(GLED,LOW);
    for(int i = 0 ; i < 10 ; i++)
    {
        R = 19.000;
        for(int j = 0 ; j < 1220 ; j++)
        {
            Step();

```



```

    Serial.println(Output (R,SC));
    delay(3);
    SC++;
}

SC = 0;

}
}

void Tscan()
{

for (int i = 0 ; i < 90 ; i++)
{
    R = S1.read();
    for(int j = 0 ; j < 1220 ; j++)
    {
        Step();
        Serial.println(Output(R,SC));
        delay(3);
        SC++;
    }
    SC = 0;

    S1.write(19+i);
    delay(200);

}

```

```
}
```

```
void halt()
```

```
{
```

```
  Rbs = digitalRead(Rbtn);
```

```
  /////check on this
```

```
  while(Rbs != HIGH)
```

```
  {
```

```
  }
```

```
}
```

```
int dis()
```

```
{
```

```
  int thisd = 0;
```

```
  pulseWidth = pulseIn(Mon, HIGH); // Count how long the pulse is high in  
  microseconds
```

```
  // If we get a reading that isn't zero, let's print it
```

```
  if(pulseWidth != 0)
```

```
  {
```

```
    pulseWidth = pulseWidth / 10; // 10usec = 1 cm of distance
```

```
    thisd = (pulseWidth); // Print the distance
```

```
  }
```

```
  return thisd;
```

```
}
```

```

String Output( float r , int sc)
{
// int radius = 0;
// if ((sc/100 == 1) || (sc/100 == 2) || (sc/100 == 3) || (sc/100 == 4) || (sc/100 == 5) ||
(sc/100 == 6))
// {
//   radius = dis(true,LL_Ad);
// }
//
// else
// radius = dis(false,LL_Ad);
//
float azi = ((sc * s2g) * d2r);
float eli = (90 - (r-19)) * d2r;
int radius = 0;
radius = dis();
double x = radius * sin(eli) * cos(azi);
double y = radius * sin(eli) * sin(azi);
double z = radius * cos(eli);
return (String(x, 5) + " " + String(y, 5) + " " + String(z, 5));
// return(String(sc)+ " " + String(r,4));
// return (String(radius) + " " + String(sc) + " " +String(r) + " " + " | " + String(x,4) + " "
+ String (y,4) + " " + String(z,4) );
}

```

Appendix G Processing Code

```
import processing.serial.*;
```

```
Serial scanserial;
```

```
int serialPortNumber = 0;
```

```
float angle = 6.5f;
```

```
float angleIncrement = 0;
```

```
float xOffset = 3.0;
```

```
float xOffsetIncrement = 0;
```

```
float yOffset = 152.0f;
```

```
float yOffsetIncrement = 0;
```

```
float scale = 2.6f;
```

```
float scaleIncrement = 0;
```

```
ArrayList<PVector> vectors;
```

```
int LPI = 0;
```

```
int LPC = 0;
```

```
PrintWriter output;
```

```
void setup() {
```

```
    size(1000, 1000, P3D); //size of 3d plot created
```

```
    colorMode(RGB, 255, 255, 255); //
```

```
    noSmooth();
```

```
    vectors = new ArrayList<PVector>();
```

```
    String[] serialPorts = Serial.list(); //check for available serial ports
```

```
    String serialPort = serialPorts[serialPortNumber];
```

```
    println("Using serial port \"" + serialPort + "\"");
```

```
    println("To use a different serial port, change serialPortNumber:");
```

```
    printArray(serialPorts);
```

```
    scanserial = new Serial(this, serialPort, 115200); //baudrate used by the arduino
    program
```

```
    output = createWriter("pos.txt");
}
```

```
void draw() {
```

```
    String input = scanserial.readStringUntil(10);
```

```
    if (input != null) {
```

```
        String[] components = split(input, ' ');
```

```
        if (components.length == 3) {
```

```
            vectors.add(new PVector(float(components[0]), float(components[1]),
float(components[2])));
```

```
            output.println((float(components[0]) + " " + float(components[1]) + " " +
float(components[2])));
```

```
        }
```

```
    }
```

```
    background(255);
```

```
    translate(width/2, height/2, -50);
```

```
    rotateY(angle);
```

```
    int size = vectors.size();
```

```
    for (int index = 0; index < size; index++) {
```

```
        PVector v = vectors.get(index);
```

```
        if (index == size - 1) {
```

```
            // draw red line to show recently added LIDAR scan point
```

```
            if (index == LPI) {
```

```
                LPC++;
```

```
            } else {
```

```
                LPI = index;
```

```
                LPC = 0;
```

```
            }
```

```
            if (LPC < 10) {
```

```

        stroke(0, 255, 0);
        line(xOffset, yOffset, 0, v.x * scale + xOffset, -v.z * scale + yOffset, -v.y * scale);
    }
}
stroke(0, 0, 0);
point(v.x * scale + xOffset, -v.z * scale + yOffset, -v.y * scale);
}
angle += angleIncrement;
xOffset += xOffsetIncrement;
yOffset += yOffsetIncrement;
scale += scaleIncrement;
}

void keyPressed() {
    if (key == 'q') {
        // zoom in
        scaleIncrement = 0.02f;
    } else if (key == 'z') {
        // zoom out
        scaleIncrement = -0.02f;
    } else if (key == 'a') {
        // move left
        xOffsetIncrement = -1f;
    } else if (key == 'd') {
        // move right
        xOffsetIncrement = 1f;
    } else if (key == 'w') {
        // move up
        yOffsetIncrement = -1f;
    } else if (key == 's') {

```

```

    // move down
    yOffsetIncrement = 1f;
} else if (key == 'c') {
    // erase all points
    vectors.clear();
}
else if (key == 'm'){
    output.flush();
    output.close();
} else if (key == CODED) {
    if (keyCode == LEFT) {
        // rotate left
        angleIncrement = -0.015f;
    } else if (keyCode == RIGHT) {
        // rotate right
        angleIncrement = 0.015f;
    }
}
}
}

```

```

void keyReleased() {
    if (key == 'q') {
        scaleIncrement = 0f;
    } else if (key == 'z') {
        scaleIncrement = 0f;
    } else if (key == 'a') {
        xOffsetIncrement = 0f;
    } else if (key == 'd') {
        xOffsetIncrement = 0f;
    } else if (key == 'w') {

```

```
    yOffsetIncrement = 0f;
} else if (key == 's') {
    yOffsetIncrement = 0f;
} else if (key == CODED) {
    if (keyCode == LEFT) {
        angleIncrement = 0f;
    } else if (keyCode == RIGHT) {
        angleIncrement = 0f;
    }
}
}
```


Appendix H Matlab Code

```
clear;
clc;
close all;

disp('-----Final Year Project Mayur Ramjee-----')
A = dlmread('pos.txt',' ');
Iloop = true;
Tloop = true;
while (Iloop);
    dime = input('2D or 3D graph to be displayed? : Please enter either 2
= 2D or 3 = 3D): ');
    if (dime == 2 || dime == 3)
        Iloop = false;
    end

end

if (dime == 2)
while (Tloop)
    pL = input('Select plane: 1 = XY , 2 = XZ , 3 = YZ: ');
    if (pL == 1 || pL == 2 || pL == 3)
        Tloop = false;
    end

    if (pL == 1)
        plot(A(:,1),A(:,2));
    end

    if (pL == 2)
        plot(A(:,1),A(:,3));
    end

    if (pL == 3)
        plot(A(:,2),A(:,3));
    end
end
end

if(dime == 3)
plot3(A(:,1),A(:,2),A(:,3));
end
```

Appendix I Trilateration Code

```
clc;
clear;
close all;

%-----Trilateration Calculator-----
x1 = input('input x1 co-ordinate: ');
y1 = input('input y1 co-ordinate: ');
x2 = input('input x2 co-ordinate: ');
y2 = input('input y2 co-ordinate: ');
x3 = input('input x3 co-ordinate: ');
y3 = input('input y3 co-ordinate: ');

r1 = input('input distance to point 1: ');
r2 = input('input distance to point 2: ');
r3 = input('input distance to point 3 : ');

matxi = [((r1^2 - r2^2)-(x1^2 -x2^2)-(y1^2 - y2^2)) 2*(y2-y1) ; ((r1^2 -
r3^2)-(x1^2 -x3^2)-(y1^2 - y3^2)) 2*(y3-y1)];
matxb = [2*(x2- x1) 2*(y2-y1) ; 2*(x3 - x1) 2*(y3-y1)];

matyi = [2*(x2-x1) ((r1^2 - r2^2)-(x1^2 - x2^2)-(y1^2 - y2^2)) ; 2*(x3 -
x1) ((r1^2 - r3^2)-(x1^2 - x3^2)-(y1^2 - y3^2))];
matyb = [2*(x2 - x1) 2*(y2 - y1) ; 2*(x3 - x1) 2*(y3 - y1)];

PtX = (det(matxi)/det(matxb))
PtY = (det(matyi)/det(matyb))

displayx = strcat('X coordinate := ', num2str(PtX));
displayy = strcat('Y coordinate := ', num2str(PtY));

disp(displayx);
disp(displayy);
```

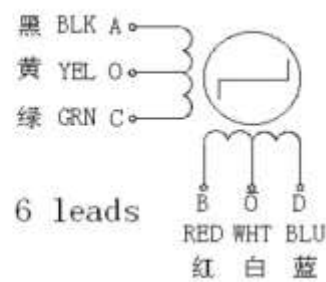
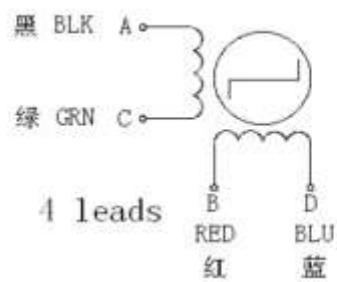
Appendix J Specification sheets of equipment used



<http://www.openimpulse.com>

42BYGHW208 Stepper Motor Datasheet

Wiring diagram (接线图):



Model	Step Angle (°)	Motor Length L(mm)	Rate Voltage (V)	Rate Current (A)	Phase Resistance (Ω)	Phase Inductance (mH)	Holding Torque (g.cm)	Lead Wire (NO.)	Rotor Inertia (g.cm ²)	Detent Torque (g.cm)	Motor Weight (kg)
42BYGHW208	1.8	34	12	0.4	30	37	2800	4	34	200	0.2

Specifications

Physical

Specification	Measurement
Size (LxWxH)	20 × 48 × 40 mm (0.8 × 1.9 × 1.6 in.)
Weight	22 g (0.78 oz.)
Operating temperature	-20 to 60°C (-4 to 140°F)

Electrical

Specification	Measurement
Power	5 Vdc nominal 4.5 Vdc min., 5.5 Vdc max.
Current consumption	105 mA idle 135 mA continuous operation

Performance

Specification	Measurement
Range (70% reflective target)	40 m (131 ft)
Resolution	±1 cm (0.4 in.)
Accuracy < 5 m	±2.5 cm (1 in.) typical*
Accuracy ≥ 5 m	±10 cm (3.9 in.) typical Mean ±1% of distance maximum Ripple ±1% of distance maximum
Update rate (70% Reflective Target)	270 Hz typical 650 Hz fast mode** >1000 Hz short range only
Repetition rate	~50 Hz default 500 Hz max

*Nonlinearity present below 1 m (39.4 in.)

**Reduced sensitivity

Interface

Specification	Measurement
User interface	I2C PWM External trigger
I2C interface	Fast-mode (400 kbit/s) Default 7-bit address 0x62 Internal register access & control
PWM interface	External trigger input PWM output proportional to distance at 10 µs/cm

Laser

Specification	Measurement
Wavelength	905 nm (nominal)
Total laser power (peak)	1.3 W
Mode of operation	Pulsed (256 pulse max. pulse train)
Pulse width	0.5 µs (50% duty Cycle)
Pulse train repetition frequency	10-20 KHz nominal
Energy per pulse	<280 nJ
Beam diameter at laser aperture	12 × 2 mm (0.47 × 0.08 in.)
Divergence	8 mRadian

Connections

Wiring Harness



Wire Color	Function
Red	5 Vdc (+)
Orange	Power enable (internal pull-up)
Yellow	Mode control
Green	I2C SCL
Blue	I2C SDA
Black	Ground (-)

There are two basic configurations for this device:

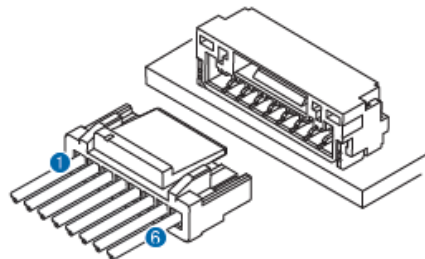
- I2C (Inter-Integrated Circuit)**—a serial computer bus used to communicate between this device and a microcontroller, such as an Arduino board ("I2C Interface", page 4).
- PWM (Pulse Width Modulation)**—a bi-directional signal transfer method that triggers acquisitions and returns distance measurements using the mode-control pin ("Mode Control Pin", page 4).

Connector

You can create your own wiring harness if needed for your project or application. The needed components are readily available from many suppliers.

Part	Description	Manufacturer	Part Number
Connector housing	6-position, rectangular housing, latch-lock connector receptacle with a 1.25 mm (0.049 in.) pitch.	JST	GHR-06V-S
Connector terminal	26-30 AWG crimp socket connector terminal (up to 6)	JST	SSH-002T-P0.2
Wire	UL 1061 26 AWG stranded copper	N/A	N/A

Connector Port Identification



Item	Pin	Function
1	1	5 Vdc (+)
2	2	Power enable (internal pull-up)
3	3	Mode control
4	4	I2C SCL
5	5	I2C SDA
6	6	Ground (-)

Apply Environmental Condition :

No.	Item	Specification
1-1	Storage Temperature Range	-20℃ ~ 60℃
1-2	Operating Temperature Range	-10℃ ~ 50℃
1-3	Operating Voltage Range	4.5V ~ 6V

Standard Test Environment :

2-1	Standard Test Environment	Every characteristic of the inspect must be normal temperature and humidity carry out the test , temperature 25±5℃ and relative humidity 65±10% of judgment made in accordance with this specification standard testing conditions.
-----	---------------------------	---

Electrical Specification (Function of the Performance) :

No.	Item	4.8V	6.0V
4-1	Operating speed (at no load)	0.12 sec/60°	0.11 sec/60°
4-2	Running current (at no load)	160 mA	180 mA
4-3	Stall torque (at locked)	2.0 kg-cm	2.7 kg-cm
4-4	Stall current (at locked)	680 mA	800 mA
4-5	Idle current (at stopped)	4 mA	5 mA

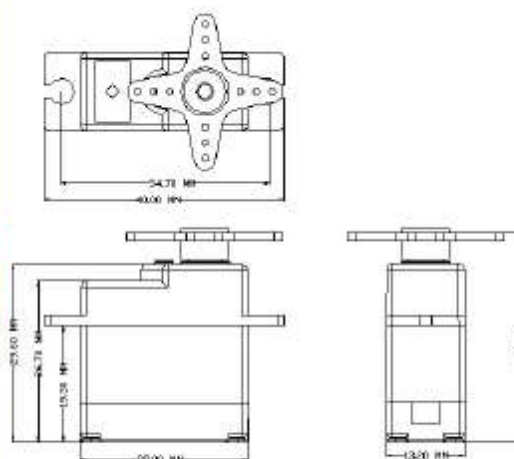
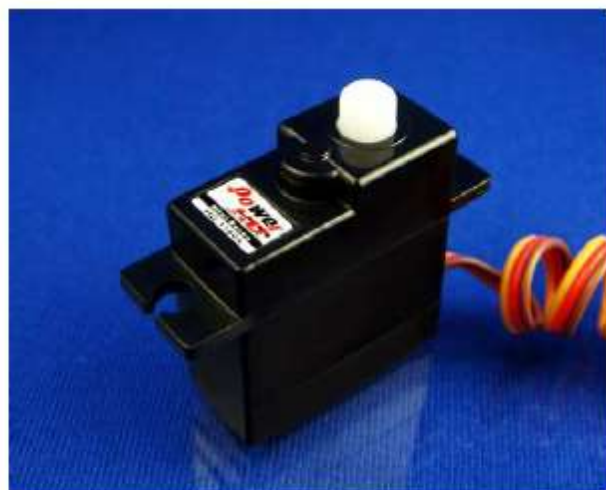
Note: Item 4-2 definition is average value when the servo running with no load

Mechanical Specification :

No.	Item	Specification
5-1	Overall Dimensions	See the drawing
5-2	Limit angle	180° ± 10°
5-3	Weight	17 ± 1 grams (without servo horn)
5-4	Connector wire gauge	#26 PVC
5-5	Connector wire length	150 ± 5 mm
5-6	Horn gear spline	25T/φ 5.77
5-7	Horn type	Cross , Disk , Star , Bar
5-8	Reduction ratio	1/313

Control Specification :

No.	Item	Specification
6-1	Control system	Pulse width modification
6-2	Amplifier type	Analog controller
6-3	Operating travel	90° (when 1000→2000 μsec)
6-4	Neutral position	1500 μsec
6-5	Dead band width	7 μsec
6-6	Rotating direction	Clockwise (when 1500→2000 μsec)
6-7	Pulse width range	800→2200 μsec
6-8	Maximum travel	Approx 165° (when 800→2200 μsec)



Power HD

Product Name
SERVO

Model No.
HD-1160A

Version
V1

Page
1/1