3D LASER SURFACE MAPPING

By

Aqsa Hasan Siddiqui

Jehanzaib Yousaf

Ruba Zaheer

Submitted to the Faculty of Department of Electrical Engineering

Military College of Signals

National University of Sciences and Technology, Islamabad

in partial fulfillment for the requirements of a B.E Degree in

Electrical (Telecom) Engineering

JUNE 2015

ABSTRACT

3D LASER SURFACE MAPPING SYSTEM

The 3D laser surface mapping project aims to build a 3D model of an unknown environment that is to be scanned. In the project, laser distance meter and synchronized motion control is used to record accurate distance measurements between the device and a specific point on the surface to be scanned. A rotating mechanical assembly is constructed which is used to move the laser distance meter in two degrees of freedom. This is done with the help of two stepper motors, which provide incremental movement in azimuth and inclined plane. With the help of an interfacing board, distance measurements are sent from laser distance meter to a personal computer through USB port and processed. The incremental motion of the distance meter in three dimensional spaces is translated into an appropriate coordinate system. The processed data is then exported to MeshLab, a 3D modeling software, which is used to construct a 3D representation of the surface. A Graphical User Interface (GUI) is created to provide users control of the system and produce surface scans according to their requirement.

Lt. Col Dr. Adil Masood

Head of Computer Software Engineering Department

Military College of Signals

Humayun Road

Rawalpindi.

Dedicated to Allah Almighty, the Lord of the Worlds, our family and friends who believed in us and for their unwavering support.

ACKNOWLEDGEMENTS

Praise be to Allah, the Magnificent and the Merciful, for showering us with His countless blessings and for giving us the strength and patience to bear all hardships. We are grateful to our parents without their prayers, support and hard work this would not have been possible. We are thankful to our teachers and especially our supervisor, Dr. Adil Masood, for being our torch bearer and guiding us when else failed.

We would like to thank our family for bearing with us patiently and our friends who have been there to support us morally and technically.

CHAPTER 1 INTRODUCTION

1.1 MOTIVATION

We live in a three dimensional world. Over the time, scientists and engineers have strived to re-create reality in three dimensions through maps and drawings. However, the 3D view of objects changes depending on the viewing angles, and so this process is quite tedious. Traditional maps offer a very limited 3D impression.

This lead to the advent of 3D modeling which made it possible to transfer reality into 3D digital models. These 3D models were independent of the viewing angles and adequate cover of the aspects of reality under investigation requires it understanding from many different viewpoints. The disciplines of geology, hydrology, civil engineering, environmental engineering, landscape architecture, archeology, meteorology, mineral exploration, 3D urban mapping, all draw on modeling for the efficient completion of their tasks [1]. Among the various methods to obtain 3D data, laser scanning is widely used due to its accuracy and modest complexity of the measured laser data [2]. 3D laser scans are cost-effective, accurate, and provide reduced exposure to hazards on the plant site.

Today, there are many 3D laser scanning devices available in market e.g. Faro Focus, Trimble TX5 etc. These scanners are expensive, hard to calibrate, produce intensive point clouds which require long processing time and produce poor quality images [3]. The motivation of our project is to develop a system that models an unknown environment in 3D usinglaser scanning and is automated, efficient, economical and easy to use.

1.2 PROJECT OVERVIEW

The 3D laser surface mapping project’s goal is to build a 3D model of an unknown environment that is to be scanned. This project will scan a surface using a laser distance meter (LiDAR), obtain coordinate data and reconstruct the data to realize a 3-D model of the scanned surface. Laser distance meter will be used to record accurate distance measurement between the device and a specific point on the surface. Distance measurements are sent from laser distance meter to a personal computer through USB communication. To obtain data in three dimensions, this laser distance meter will measure distance in two degrees of freedom. For this a mechanical assembly will be created to move the laser distance incrementally in azimuth and inclined plane. The incremental motion is achieved with the help of stepper motors, which provide accurate and stable movement. Motor control will be carried out with the use of a programmed microcontroller. Also, spatial coordinates of the motors will be recorded in a personal computer in spherical coordinate system. After required processing of the data coordinates, they will be exported to MeshLab, an advanced 3D mesh rendering software with 3D image rendering filters, which will be used to construct a 3D representation of the surface. A Graphical User Interface (GUI) will be created to provide users control of the system and produce surface scans according to their requirement. A block diagram of the 3D Laser Surface Mapping project is given below.

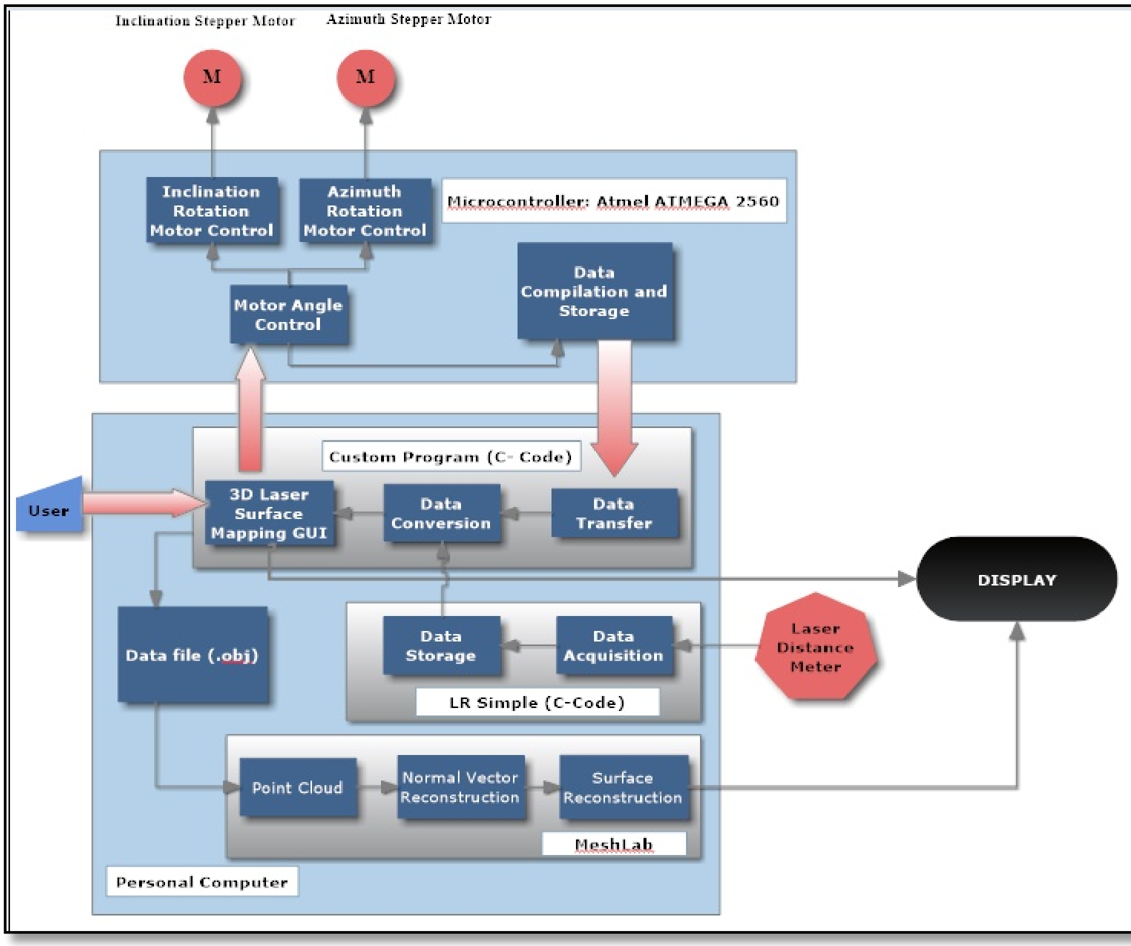


Figure 1-1Block diagram of 3D Laser Surface Mapping System

As it can be observed in the block diagram shown in Figure 1.1 above, there are various modules and sub-modules involvedin 3D laser surface mapping project which will be explained in Chapter 3 of this document.

The 3D laser surface mapping project will give a detailed and accurate representation of the scanned area in three dimensions thereby giving the user an opportunity to explore unknown and susceptible environments closely. The model will have an ability to be viewed from different angles and aspects which is useful for analysis. The laser distance meter will acquire distance measurements with significant accuracy and precision at large distances producingrigorous 3D models of scanned environments. Otherprospective applications of the project include high portability, user flexibility and efficiency.

The main application of this project is room surface mapping and construction site mapping. Further, it finds application in different and more advantageous manners which include fully automated 3D modeling and accurate, cost effective laser scanning solution.

1.3 SCOPE AND SPECIFICATIONS OF THE PROJECT:

The 3D laser surface mapping project aims to scan an unknown environment in order to create its 3D model. This is done with the help of a laser distance meter which will be used to record accurate distance measurements between the device and a specific point on the surface. Distances up to 40 meters with accuracy of ±3mm will be measured[5].The laser distance meter will be mounted on a rotating mechanical assembly which will be able to rotate the meter in two degrees of freedom. The mechanical assembly will rotate incrementally with the help of two stepper motors, one for each degree of motion, with a step size of 1.8^0. It will move 100 steps(360^0) in the horizontal plane and 70 steps (126^0) in the inclined plane. The rotating mechanical assembly will be constructed from aluminum and mild steel and its height would be varied to four different levels, up to a maximum of 1 meter, so it will be capable of measuring environments of various dimensions. Distance measurements and spatial information of the laser distance meter will be transferred to a personal computer. After required manipulation of this data, it will be exported to MeshLab[4] where a 3D model of the scanned surface will be created in the Cartesian plane using various filters and mesh algorithms. Users would be able to view the model from different angles and aspects for detailed analysis. An interactive

Graphical User Interface will be created on a personal computer to provide users sufficient control of the 3D Laser Surface Mapping system by choosing their desired scan from three different available scans of varying accuracy and time, selecting the position of mechanical assembly, starting and stopping the system and viewing 3D data coordinates. The overall system will be automated and compact to save users from any hassle.

CHAPTER 2 LITERATURE REVIEW

3D scanning captures as-built conditions rapidly and precisely. Once assembled, the scanned images reveal building relationships in ways difficult to imagine, let alone measure. It provides detailed spatial information to architects, design professionals, engineers, and builders.

For many years, three-dimensional (3D) scanning has been widely used for industrial applications such as reverse engineering and part inspection. Over the past few years, dramatic decreases in the cost of 3D scanning equipment has led to its increased use for many other applications, including rapid prototyping and modeling, development of realistic computer graphics in the video gaming market, and more recently, in the biomedical field. [7] 3D scanning is also used for architecture surveys and mapping [8], as built deliverables of plans, sections and elevation. The potential exists to expand the use of 3D models even further, by continuing to develop simpler, more cost effective systems for acquiring external shape features of arbitrary objects. [7]

There are many possibilities to acquire 3D information from the surrounding environment. The measurement methods can be divided into three major categories based on applied sensor and sensing technology: stereo vision with two or more cameras, active triangulation and time-of-flight (TOF) measurements. One of the most precise TOF measurement systems is based on laser scanners. [9]

Among the various methods used for 3D scanning, some of them are mentioned below:

1. **Microsoft Kinect [10] -** The current technology in 3D room mapping is the recently invented Microsoft Kinect system, a system that takes live depth data from a moving Kinect camera and in real-time creates high-quality, geometrically accurate, 3D models. As the camera is moved closer to objects in the scene more detail can be added to the acquired 3D model. However, this system requires an overhead video tracking system which limits its application to scanning small indoor spaces.
2. **3-D mapping using ultrasounds-** Acoustic waves are used to obtain a room model which could be used to predict the way sound propagates inside a room[11]**.**Sound waves are suitable alternative for finding range where exact precision is not required. Ultrasound transducers are relatively inexpensive, used for short distance detection and usually ultrasound beams reflect well off rocky surfaces.[3] The lack of precision is due to the difficulty in obtaining a narrow ultrasound beam and the problem of shaping a precisely defined sound pulse.
3. **RGB-D Mapping (using depth cameras for dense 3D Mapping) [12] -** The goal of this research is to create dense 3D point cloud maps of building interiors using newly available inexpensive depth cameras. These cameras provide per-pixel depth information aligned with image pixels from a standard camera. It is referred as RGB-D data (for Red, Green, Blue plus Depth).Again there are constraints of high bandwidth and large data space.

However, there are numerous advantages of 3D laser surface mapping over other methods. Cameras take 2D images instead of 3D and take blurry pictures on large distances. They generate large amount of data, require high data rate and bandwidth. Cameras cannot create 3D surface maps, which can be manipulated for example in military applications, medical and in engineering. Sophisticated maps can be created by using long range laser distance meters where camera will not be suitable. It reconstructs maps from few data points or completes holes in models. Instead of transmitting images and models, only data points are transmitted to remote servers where 3D maps can be created.

After a detailed background study, time of flight (TOF) method was selected for 3D scanning. Cheng says three-dimensional laser scanner (TLS) emits controlled laser beam on an object according to certain laws, and obtains the 3D surface information of object by receiving the reflection.[13] In Wang's method, a pulse of laser beam is emitted, and the pick of the returned pulse is detected. By measuring the flight time of the laser beam, the distance is calculated by $D = c\Delta t/2$; where D is the distance between the laser diode and a target, c is light velocity, Δt is the flight time. [4] Weingarten observes commercial laser range finders like Sick, Leica, Riegl or Velodyne make use of a rotary mirror system through which the laser beam is swept along a surface in order to gain 2D or 3D information. [14] Hence, we implement the above concept for obtaining distance measurements using a laser distance meter, Fluke 414D, which uses a Leica microprocessor.

Surman and Jensfelt say that in order to get information from the third dimension, the standard 2D laser scanners are often used with an auxiliary rotary mechanical system. The 2D laser is then mounted on that system, obtaining the third degree of freedom for the laser beam. Such an approach based on servo actuator system was used.[15] [16] Wang, says that the laser rangefinder performs horizontal scan and vertical scan by two scanning mechanisms. The scanning angles and velocities can he controlled by two servomechanisms.[4] Also, Cheng uses Cartesian coordinates to represent target coordinates obtained using 3D scanning.[13] Keeping in mind the above ideas, our 3D laser surface mapping project consists of a rotating assembly which rotates the laser distance meter horizontally and vertically. It obtains data in spherical coordinates which are later on converted into Cartesian coordinates.

Silvestre used the MeshLab software for the 3D-mesh generation of the cave chamber.[3] MeshLab is free and open source software for mesh processing and editing.Furthermore, it works with a huge number of common 3D file formats and has several algorithms to do the surface reconstruction from point clouds.[6] After a detailed review we decided to work with MeshLab for 3D surface map construction in our project.

2.1 PREVIOUS WORK DONE IN MCS:

Until now, the work done in MCS in the field of imaging is done by means of interrogating microwaves.[23] Microwave imaging is a technique which senses a given scene through microwaves. It is generally perceived as the reconstruction of permittivity distribution function of the medium. This technique is used in diagnostics. Through the wall imaging is one of the fields of microwave imaging which consists of a setup in which target is localized and reconstructed by using an imaging system position behind a wall. However in our project, laser scanning and then mapping of the 3D environment will be done by emitting a laser beam and rotating it in three dimensions to obtain 3D data coordinates.

CHAPTER 3 DESIGN AND DEVELOPMENT

3D Laser Surface Mapping projectis divided into four basic modules as shown below.

3.1 DATA ACQUISITION UNIT

The Data Acquisition Unit uses a laser distance meter, Fluke 414D, to obtain distance measurements. Laser distance meter is used for easy targeting even at large distances. It has a typical range of 35 meters. Typical measuring tolerance of Fluke 414D is ± 2 mm and the maximum measuring tolerance is ± 3 mm. Its measurement rate is 2.5Hz, which is 0.4seconds for a measurement.[5]

The laser distance meter works on the principle of TOF. It uses a laser diode to produce a pulsed beam of light that is bounced off a surface from which distance is to be measured. A sensitive optical sensor detects the reflected light pulses and a microprocessor determines the time difference between the pulses being produced and detected. With the knowledge of the speed of light, the distance to the object can be calculated.

$$distance = \frac{1}{2} * c * \Delta t \quad (1)$$

Equation (1) gives distance travelled by light, where c is the speed of light and Δt is the time taken for light to travel the distance from the laser diode to the object and back. The figure below represents the basic working principle of the laser distance meter.

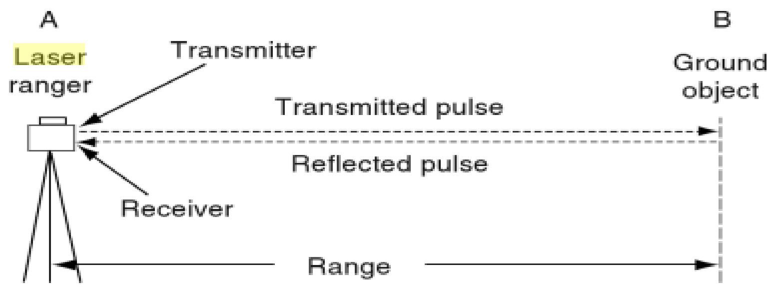


Figure3-1 Basic operation of a laser range finder which is working on the TOF method

As can be seen from the figure above, the laser distance meter, also known as laser range finder, transmits and receives a pulsed beam of light to find the distance from the object.

As it was not possible to acquire data directly from laser distance meter, so it was the requirement of the project to use an interfacing board.In order to acquire data from Fluke 414D, Porcupine LR4[17] board is used. The laser distance meter is interfaced with this board through flex cables. Flex cables are used to retrieve data from the meter via its screen connection and control the meter via its key pad connection. The battery wires of Fluke 414D are plugged into LR4 board without soldering to supply the laser distance meter with power.

Initially, attempts were made to interface LR4 [17] with microcontroller [18] to acquire distance measurements. Distance measurements from Porcupine LR4 were serially transmitted to Arduino ATMega 2560 [18] and control commands were serially transmitted from the microcontroller to Fluke 414D. A code was developed in Arduino IDE software for serial communication of Porcupine LR4 and ATMega 2560. The code was able to successfully start and stop the distance meter however distance measurements from the LR4 board could not be obtained correctly. A bottleneck in this method was compatibility of logic levels of serial ports of LR4and microcontroller. The data obtained from serial port of microcontrollerwas mostly garbage data and not the required measurements. Furthermore, acquiring distance measurements through the Arduino board increased its computational complexity and was a burden on it meager memory.

To resolve the problem, a USB mini “B” connector on the LR4 is used. The distance measurements are transmitted from LR4 to the PC through this USB connector. The LR4 demo software includes a Windows application, LRSimple, that shows how to create software that interfaces to the LR4. The LRSimple application is a bare bones Win32 Console app that implements the simplest method to start the LR4 and read distance measurements. This source code was modified to create text files (.txt) which stores distance measurements serially. The figure below shows the Data Acquisition unit measuring distances using a laser pulse that is incident on the object from which distance is to be measured.

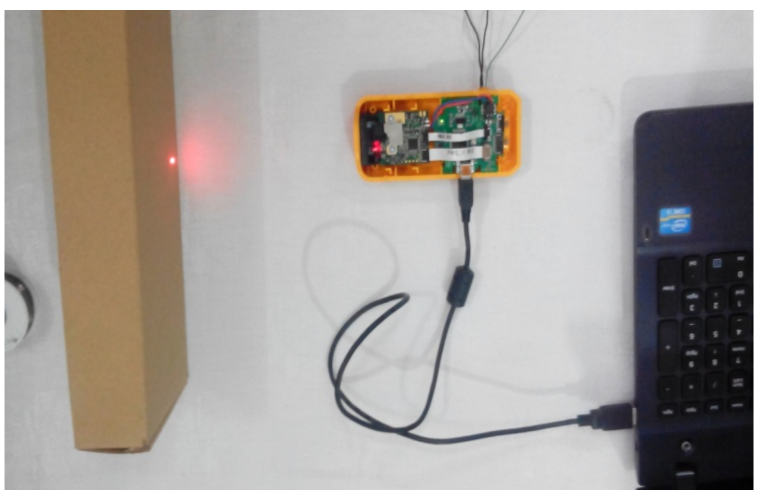


Fig2-2 Data Acquisition Unit connection with PC

In the figureabove,the Data Acquisition Unit is connected to a PC via its USB port and is transmitting distance measurements to it.

3.2 MECHANICS AND MOTION CONTROL

The Data Acquisition Unit will be mounted on a rotating assembly which will be designed to rotate the sensor array in two degrees of freedom i.e. in azimuthplane and inclined plane. In the azimuth plane, the Data Acquisition Unit covers 360^{o} while in the inclined plane it covers 126^{o} starting from the zenith direction. For this, two stepper motors[19] will be used, one for each degree of freedom. An approximate representation of assembly that is made in the project is shown in Figure 3-3.

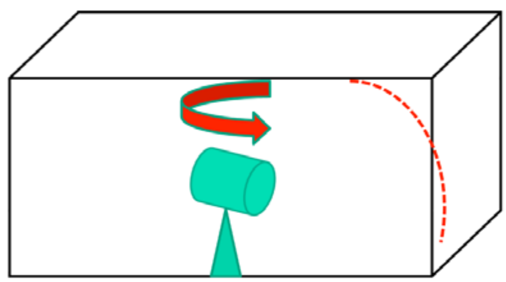


Figure 3-3 Mechanical assembly rotating in two degrees of freedom

The figure shows a diagrammatic representation of a laser distance meter mounted on a rotating mechanical assembly.

A rotating mechanical assembly is designed and manufactured for obtaining distance measurements in three dimensions. A square plate of side 18 cm and height 1.2 cm is made from aluminum. This square plate has a circular groove 1 cm wide and radius 12 cm for bearings which will be helpful for smooth motion of the rotating plate. In the center of the square plate is a circular hole to fix the stepper motor. On top of this fixed square plate will be a rotating plate of radius 16 cm. Similar to the fixed square plate, a small hole will be present at the center of the rotating plate for coupling the motor's shaft to the plate. Also a circular groove 1 cm wide and radius 12 cm is present below the rotating plate to accommodate bearings for smooth rotation. The fixed square plate along with the rotating plate will be mounted on four mild steel rods. The rods can be fixed in four different positions so overall the height of the assembly can be varied to four different heights depending on the environment being scanned. They are 16 cm long and have four holes in the rod at equal intervals of 12 cm. This rod is fixed inside another rod of 50 cm length with a single hole. The height of the assembly can be varied using a screw through one of the holes and tightening it with the only hole on the other rod. The laser distance meter is coupled to the shaft of a stepper motor attached on its side. The designs of various components described above were made in Auto Cad and are attached in Appendix A.

A bottleneck in manufacturing the rotating assembly was the height of the stand on which laser distance meter would be mounted. A simple stand of fixed length would not be able to measure environments of different dimensions. Hence, this bottleneck was overcome by manufacturing rotating assembly which could be varied in length.

Bipolar stepper motors[19] of 1.8^{o} step size areutilized in the rotating mechanical assembly. The stepper motors will take 200 steps in azimuth plane and 70 steps in inclined plane.The Data Acquisition Unit moves in the inclined plane from 0^{o} to 126^{o} relative to the zenith direction, takes 1.8^{o}steps in the azimuth plane and moves backward 126^{o} to 0^{o} in the inclined plane. This process is repeated until the Data Acquisition Unit covers 360^{o} in azimuth plane, in 1.8^{o} increments, covering 126^{o} in the inclined at every step. The motion of stepper motors[19] will be commanded by the Arduino board[18] which provides it pulses of power through the Adafruit Motor Shield[20]. The Adafruit Motor Shield[20] allows you to easily control motor direction and speed using an Arduino[18]. It also allows you to power a motor with a separate power supply of up to 12V.

The stepper motors are controlled through PWM signals from the Arduino. The Adafruit Motor Shield[20] pins are inserted into the Arduino board and it is mounted on top of it. The four wires of stepper motor are connected to the motor shield. A code has been developed on Arduino IDE software[18] to control motor speed and no of steps per revolution. The motor shield connection with Arduino[18] and stepper motor[19] is shown in Figure 5.

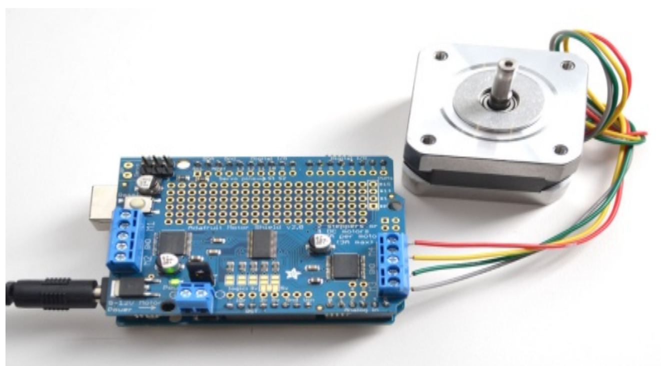


Figure 3-4 Adafruit motor shield v1 with motor and
Arduino ATMega 2560

As you can see in the figure above, a stepper motor[19] is connected to Adafruit motor shield[20]which is stacked on top of Arduino Mega 2560.

One limitation faced while rotating the Data Acquisition Unit in the inclined plane was the angle it can cover. On trying to move the laser distance meter beyond 126° the laser beam fell on the base plate resulting in erroneous readings. Therefore, the angle covered by the laser distance meter was restricted from 0° to 126° from the zenith direction.

3.3 DATA PROCESSING UNIT

The 3D data coordinated obtained are in spherical coordinate system (R, Theta, Phi). They are obtained by synchronized horizontal and vertical motion of stepper motors. Radius will be the distance measurements obtained directly from the laser distance meter. Theta will be the angle that depicts the inclination of the laser distance meter with respect to positive zenith direction. While Phi is the angle that laser distance meter makes in the azimuth plane with respect to starting point. The figure below represents the spherical coordinate system and its respective coordinates.

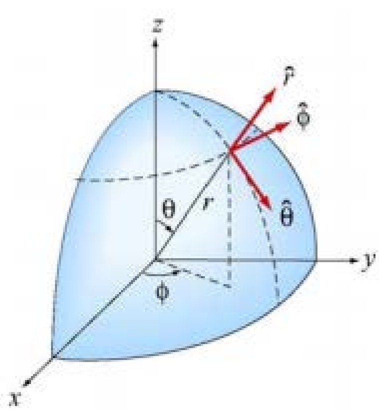


Figure 3-5 Spherical coordinates (r, θ, φ): radial distance r, polar angle θ (theta), and azimuthallyangle φ (phi)[21]

As shown in Figure above, a spherical coordinate system is a coordinate system for three-dimensional space where the position of a point is specified by three numbers: the radial distance of that point from a fixed origin, its polar angle measured from a fixed zenithdirection, and the azimuth angle of its orthogonal projection on a reference plane that passes through the origin and is orthogonal to the zenith, measured from a fixed reference direction on that plane.

Theta and phi coordinates are obtained by controlled motion of motors with reference to initial position. These coordinates, i.e. their angular measure in the spatial domain,are measured via a C code in a personal computer. The distance measurements along with spatial coordinates of the laser distance meter are stored in a text file. With the help anotherC-code written in Visual Studio the spherical coordinates stored in the text file are read andconverted to Cartesian coordinates. The Cartesian coordinates are stored in another text file which is then exported to MeshLab. To convert spherical coordinates to Cartesian coordinates, the following equations have been used:

$$x = rsin\theta cos\varphi \quad (2)$$

$$y = rsin\theta sin\varphi \quad (3)$$

$$z = rcos\theta (4)$$

The above equations were implemented in C code to convert the spherical coordinates stored in text file to Cartesian coordinates.

A GUI (Graphical User Interface) has been developed in Visual Studio using C# to provide users control over the 3D Laser Surface Mapping System and produce scans according to their requirement. This GUI sends commands to laser distance meter and Arduino[18] and outputs data coordinates.

Its main features are:

- Start and stop the 3D Laser Surface Mapping system.
- To choose one scan from three different scans available. The different scans take different times to complete. Based on user requirement any one of the scans can

be selected. A tradeoff here is accuracy. A faster scan will be less accurate, while a slower scan will be more accurate. Each scan result in different motor speed. Slower scans reduce the speed of motors resulting in more accurate scans. A fast scan compromises the accuracy of 3D model.

- Select position of 3D laser surface mapping system depending on the environment to be scanned.
- Displays distance measurements in spherical coordinates stored in text file.
- Displays distance measurements in Cartesian coordinates stored in text file.

The figure below shows the GUI created in Visual Studio using C#.

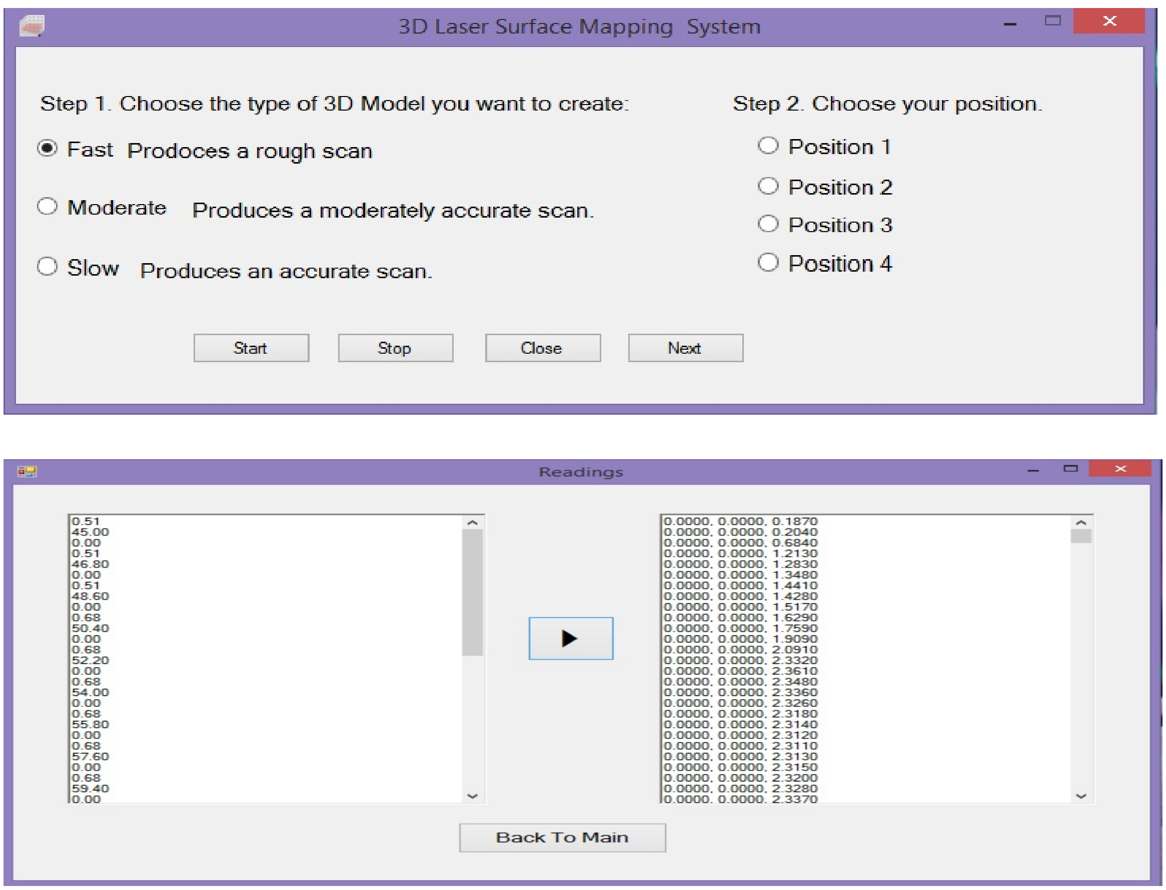


Figure 3-6 3D Laser Surface Mapping System GUI

The above figures show the two windows of GUI created in order to provide users the opportunity to produce scans according to their requirement and view 3D data coordinates.

3.4 3-D IMAGE GENERATION

In this module a 3D image will be constructed in MeshLab[6] using data processed in previous module. It is free software used for creating 3-D surfaces maps. The various steps involved in 3D image construction are:

Importing mesh- Once MeshLab is open the “Import Mesh” icon on the main toolbar will allow you to navigate to the files you have stored. The file containing Cartesian coordinates is opened and the point cloud is imported in MeshLab.

Deleting redundant point clouds- The point cloud captured from the 3D Laser Surface Mapping system has points that are not part of the actual object we want to create a 3D model of. So to avoid have spikes or deformities in our data we should apply a few methods in eliminating them when possible. To remove the redundant data points select those using *Select Vertex Clusters*in toolbox. After selecting them, go to *Filters/Selection/Delete Selected Faces and Vertices* to remove them.

Normal Reconstruction- Normals will have to be created on the sub-sample we just created so MeshLab knows which side of the point is facing “out” and which is “in”. Normals are constructed in MeshLab using *Filters/ Normals, Curvature and Orientation/ Compute normals for point sets*.

Surface Reconstruction-Once rough normals are created you will need to choose one of the surface reconstruction algorithms that MeshLab offers. Among the various surface reconstruction algorithms available Poisson surface reconstruction is a good choice. This Poisson formulation considers all the points at once, without resorting to heuristic spatial partitioning or blending, and is therefore highly resilient to data noise. It is a spatially adaptive multiscale algorithm whose time and space complexities are proportional to the size of the reconstructed model.[22]. In MeshLab, we apply Poisson Surface Construction algorithm by *Filters/Remeshing, Simplification and Reconstruction/Surface Reconstruction: Poisson.* Using the original point cloud with the computed normals we build a surface at the highest resolution.

Improving mesh- Various cleaning filters are to be applied to edit the mesh and remove duplicated vertexes, remove redundant faces and add texture to the 3D model created. Also, additional mesh layers can be added to improve the model.

Surface modeling will be done in MeshLab[6] as it is designed for 3-D rendering of large unstructured data. It offers features such as filtering, data editing, hole-filling and noise removal using smoothing filters. MeshLab[6] offers a far greater variety of 3-D rendering options than Matlab while being open-source. It can import and export many different file formats like PLY, STL, OFF, OBJ, 3DS, COLLADA, PTX, V3D, PTS, APTS, XYZ etc.

In the project data points are imported to MeshLab through a text file. This file will contain Cartesian coordinates of the scanned environment. The data points are plotted in two and three dimensions as per user requirement. 3D models are created using various algorithms and functions available in the toolbar. The resulting model can be viewed from different angles for analysis.

Chapter 4: Analysis and Evaluation

3D Laser Surface Mapping system was used to produce scans of different environments in order to obtain data points in three dimensions and produce a 3D model of it in MeshLab [4]. The following tests were performed:

1. **Scanning a classroom in Cadet's Wing**

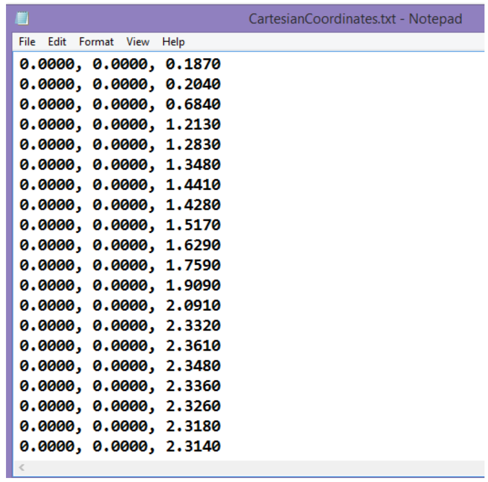
Using the 3D Laser Surface Mapping system, an accurate scan was performed of a classroom (CR-25), by selecting a 'Slow' scan at Position 0, from the application '3D Laser Surface Mapping System'. Using the data coordinates obtained a 3D model of the classroom was created in MeshLab. The following figure shows various views of this classroom.

Figure 4-1Classroom that is scanned by 3D Laser Surface Mapping System.

The figure above shows the classroom that was scanned in order to create a 3D model in MeshLab.

During the scan, two sets of data are obtained, distance measurements and spatial coordinates of the laser distance meter in three dimensions. The distance measurements are obtained from the distance measurement device and spatial coordinates are obtained from the orientation of the device with respect to its initial position due to incremental motion of stepper motors. These sets of data are inherently in Spherical coordinate system and stored serially in a text file. The first row contains the value of R (the distance from the laser distance meter to the surface), and the next two rows contains Theta and Phi coordinates respectively. The format is repeated in a similar fashion throughout the text file. The laser distance meter makes an inclination angle of 0^{o} to 126^{o}, in increments of 1.8^{o} degrees, with the help of stepper motor. This behavior is repeated at every step in the azimuth plane, as the laser distance meter takes 1.8^{o} degree incremental steps, and completes a 360^{o} degree rotation.

The entire set of data in Spherical coordinates is converted to Cartesian coordinates with the help of a C Code where spherical coordinates are read from the text file and converted to Cartesian coordinates using simple conversion formula, mentioned in Chapter 3. The Cartesian coordinates are stored in a text file. The figure below shows a text file showing these data points in Cartesian coordinates.


CartesianCoordinates.txt - Notepad
File Edit Format View Help

```
0.0000, 0.0000, 0.1870
0.0000, 0.0000, 0.2040
0.0000, 0.0000, 0.6840
0.0000, 0.0000, 1.2130
0.0000, 0.0000, 1.2830
0.0000, 0.0000, 1.3480
0.0000, 0.0000, 1.4410
0.0000, 0.0000, 1.4280
0.0000, 0.0000, 1.5170
0.0000, 0.0000, 1.6290
0.0000, 0.0000, 1.7590
0.0000, 0.0000, 1.9090
0.0000, 0.0000, 2.0910
0.0000, 0.0000, 2.3320
0.0000, 0.0000, 2.3610
0.0000, 0.0000, 2.3480
0.0000, 0.0000, 2.3360
0.0000, 0.0000, 2.3260
0.0000, 0.0000, 2.3180
0.0000, 0.0000, 2.3140
```

Figure 4-2 Data points obtained from 3D scan

The figure above shows data points in Cartesian coordinates. Each row contains X, Y and Z coordinates separated by commas and are accurate to four decimal places.

Data points generated from scanning the given environment are imported to MeshLab to construct a 3D representation of the scanned surface. The text file containing data points in Cartesian coordinates is imported to MeshLab [6]as XYZ coordinates. These are then plotted in the two dimensions as shown in the figure below.

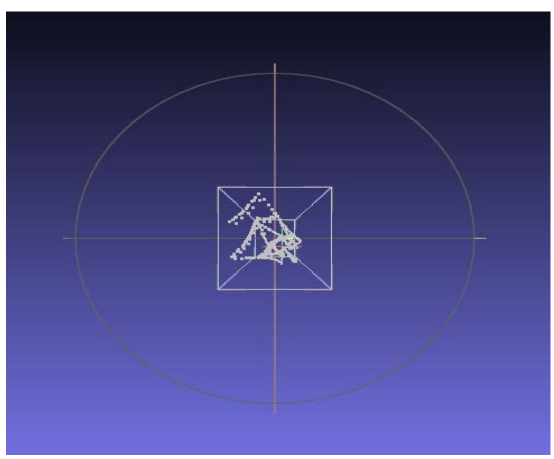


Figure 4-3 Data points plotted in two dimensional plane.

The figure above shows data points plotted in two dimensions. Using the Manipulators tool the data points are translated to three dimensions. The figure below represents data points plotted in three dimensions.

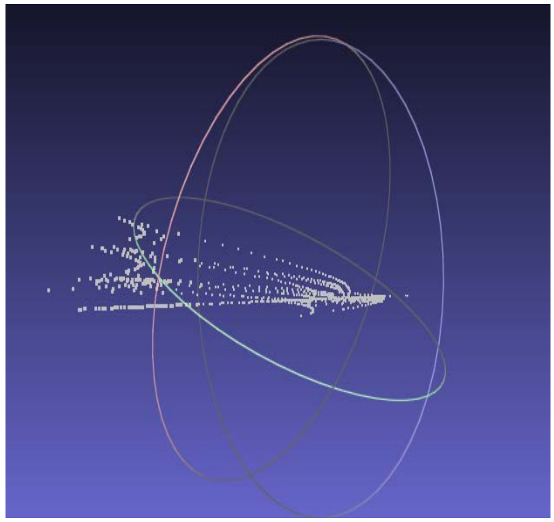


Figure 4-4 Cloud points plotted in MeshLab

Figure 4-4 above shows cloud points of the data imported in MeshLab [6] plotted in three dimensions. These data points are obtained by distance measurements taken at angular increments of 1.8 degrees.

To create a 3D model, remove the redundant data points by selecting those using Select Vertex Clustersin toolbox. After selecting them, go to Filters/Selection/Delete Selected Faces and Vertices to remove them. This results in a cleaner mesh to create an accurate 3D model. The next step is to compute normal for the point cloud using Filters/ Normals, Curvature and Orientation/ Compute normalsfor point sets. After creating normal, construct surface using Filters/Remeshing, Simplification and Reconstruction/SurfaceReconstruction: Poisson. Various cleaning filters were applied to edit the mesh and remove duplicated vertexes, remove redundant faces and add texture to the 3D model created. Figure below shows the final model of the classroom created in three dimensions.

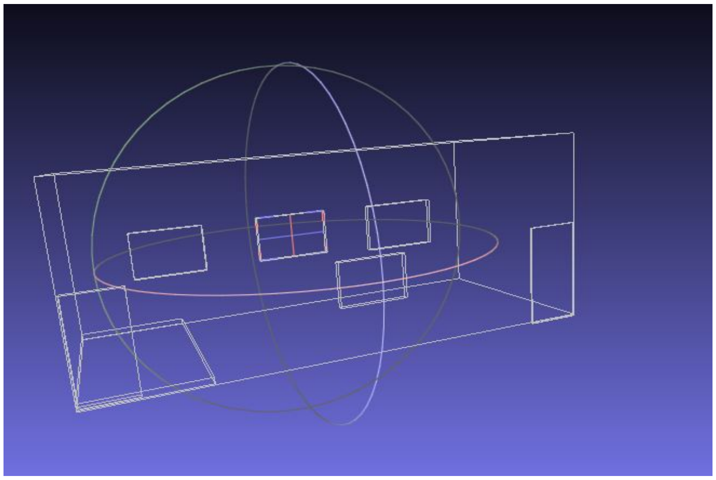


Figure 4-5 3-D model of scanned room.

As can be seen from the above figure, a 3D model is created from the data points imported to MeshLab. This model can also be viewed at different angles as shown in figures below.

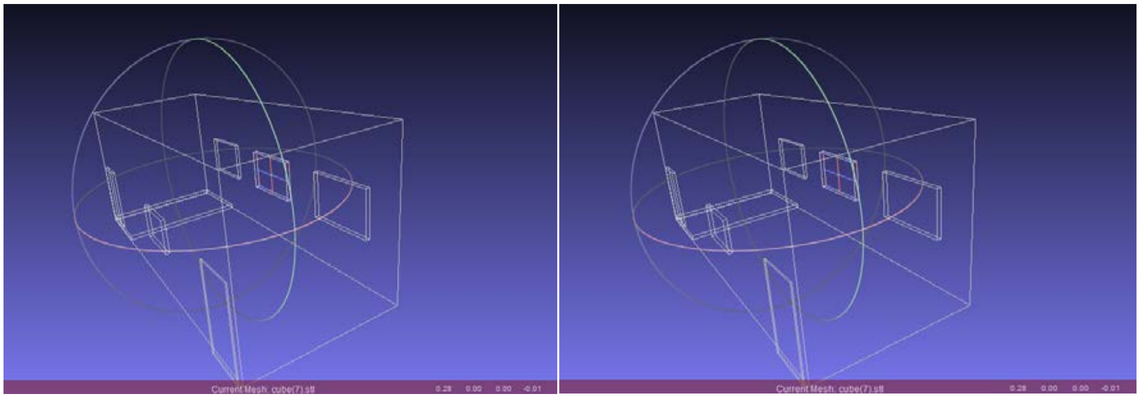


Figure 4-6 View of the 3D model from different angles

The figure above shows both point clouds and 3D model of scanned area from different angles. This model can also zoomed in and out and manipulated for detailed analysis.

Also, using the measuring tool the dimensions of the room can be found out and the distance between any two points in the model.

2. Scanning a small model of a room

To test the 3D laser surface mapping system further, more tests were carried out. A model of a room was created as shown in the figure below. The laser surface mapping system was placed in the room and a 3D model of it was created by scanning it in three dimensions. Figure shows the room model on which various scans was performed.

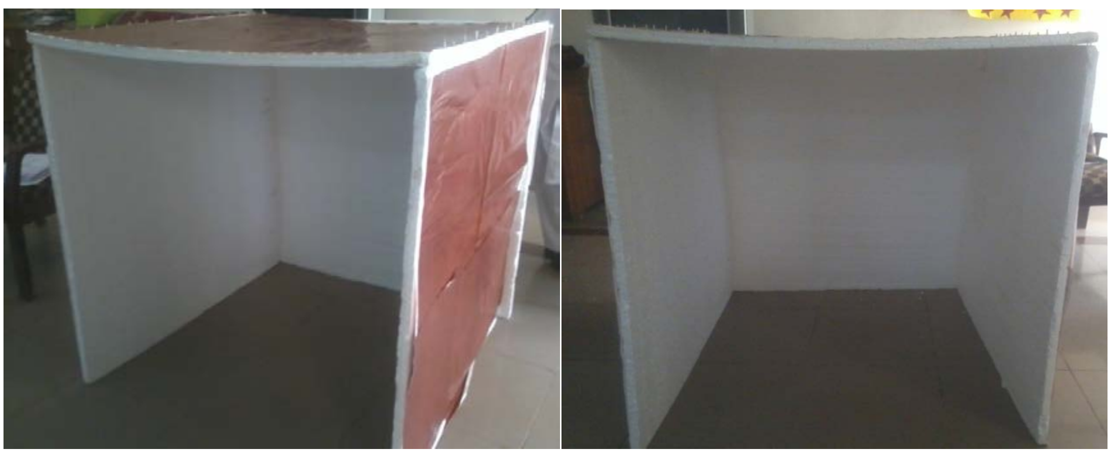


Figure 4-7 Model of a room

As can be seen in the figure, the 3D Laser Surface Mapping system was placed inside the room and accurate scan was performed. The procedure explained above was repeated, and a three dimensional model of the room was created in MeshLab [6]. The figure below shows a model of the room created in MeshLab [6].

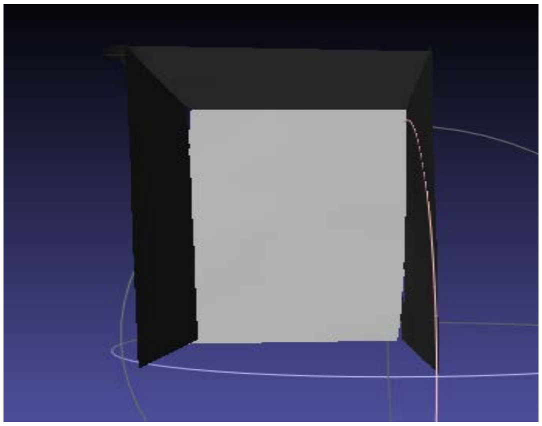


Figure 4-8 3D model of the room

As can be seen in the figure above a three dimensional model of the room was created and can be viewed from different angles.

CHAPTER 5 RECOMMENDATIONS FOR FUTURE WORK

The 3D Laser Surface Mapping system is designed to scan environments of different dimensions by varying its height to four different heights. The mechanical assembly is made of aluminum and mild steel to make it robust and stable. However, to further expand the project the entire system can be mounted on a rover which can be guided by a remote control. By mounting it on a rover, the system will become more mobilized and can be sent into deep caves and humanly inaccessible and hazardous environments.

Present assembly is made from mild steel for robustness and stability but it is heavy. The same system can be made from a lighter material which reduces the weight of the assembly and at the same time provides stability. As an alternative, other mechanisms can be introduced to ensure stability.

The present step size of stepper motors is 1.8 degree. In future,the accuracy of the model can be improved by further reducing the step size of the stepper motors. In this way, the no of point clouds will be large and closely spaced resulting in more accurate models. Large number of point clouds provides us with more defined and detailed 3D models. Moreover, advanced processing techniques for creating more accurate and interactive 3D models can be applied in the future.

Presently, the 3D laser surface mapping system has a range of 40m. Long range laser distance meters for large surfaces can be used. This will increase the overall cost of the project. Additionally, various sensors can be introduced to obtain a variety of information. This includes color, temperature and motion sensors in the system in order to obtain the more information of the environment being scanned. This will ensure better, more accurate 3D models and will translate reality into three dimensions more closely.

The 3D Laser Surface Mapping system can also be modified to eliminate the need of a personal computer. The distance measurements can be acquired by the Arduino serially through Porcupine LR4 and they can be sent, along with the spatial coordinates, wirelessly to a nearby server. These data coordinates can be transmitted either through a Wi-Fi module or a Bluetooth module interfaced to the Arduino.

CHAPTER 6 CONCLUSION

6.1 OVERVIEW

The 3D laser surface mapping project builds a 3D model of an unknown environment that is to be scanned. In the project, laser distance measuring device and synchronized motion control are used to record accurate distance measurements between the device and a specific point on the surface. Using stepper motors, which provide accurate movement, a stable mechanism to control the orientation of the laser measuring device has been developed in order to obtain data points in three dimensions. With the help of a programmed microcontroller, data points has been sent to a personal computer and processed. MeshLab, a free program that is used for 3D image construction with 3D image rendering filters, is used to construct a 3D representation of the surface. A Graphical User Interface (GUI) is created to provide users control of the system and produce surface scans according to their requirement.

6.2 OBJECTIVES ACHIEVED

3D laser surface mapping system gives a three dimensional representation of the scanned area in MeshLab thereby giving the user an opportunity to explore unknown and susceptible environments. The following objectives were achieved:

- Accurate distance measurements were recorded and obtained using the principle of laser scanning to obtain radial distance of a surface from the 3D laser surface mapping system.

- Obtained spatial coordinates in three dimensions by controlling the motion of assembly in two degrees of freedom with the help of stepper motors.
- Hands-on experience on Arduino board, programming and interfacing with LiDAR, motors and PC.
- Designed, modeled and created a rotating, variable mechanical assembly to scan environments of different dimensions.
- Successfully obtained three dimensional coordinates in spherical coordinate system using laser distance measurements, spatial coordinates and C Code in Visual Studio.
- 3D data coordinates were converted to Cartesian coordinate system using C Code in Visual Studio.
- A GUI with the name of '3D Laser Surface Mapping System' was created, using C# in Visual Studio, to provide users an opportunity to interact with the system and produce scans of their choice, visualize data coordinates in spherical and Cartesian coordinates system and control the 3D laser surface mapping system.
- Surface re-construction from point clouds in MeshLab.
- A 3D model was created in MeshLab of the environment being scanned thereby giving the user an opportunity to explore unknown environments.
- Simplified 3D mapping to a great extent by building a low cost, automated system.
- Presented this project as our Final Year Project of B.E. Electrical(Telecom).

6.3 LIMITATIONS

The 3D laser surface mapping system can scan plane surfaces with no obstacles accurately. However, it does not have the capability to model and map curved surfaces, locations with obstacles and details due to insufficient number of point clouds obtained. The laser distance meter used in the project can measure up to 40m only. The accuracy of the point cloud is limited to 1.8^{o} and the laser distance meter can only cover 126^{o} in the inclined plane because greater angles cause the laser beam to fall on the base plate.

6.4 APPLICATIONS

The main application of this project is room surface mappingand construction site mapping. Further it finds application in different and more advantageous manners which are as follows:

- Accurate and cost effective laser scanning solution
- Fully automated 3D modeling
- Cave mapping [4]
- Site planning
- Can be used for mapping of mines, tunnels and archeological sites. It can also provide archeologists with the ability to reconstruct models from ruins.
- Map difficult and hazardous terrains effectively without human intervention.

CHAPTER 5 BIBLIOGRAPHY

[1]A. Abdul-Rahman, M. Pilouk, “Spatial Data Modelling for 3D GIS”, Springer Science & Business Media, pp: 3, United States of America, 2008.

[2]J. Shan, C. K. Toth, “Topographic Laser Ranging and Scanning, Principles and Processing”, CRC Group, Taylor and Francis Group, pp:2, United States of America, 2009.

[3] L.Wang, J.Shu, T.Emira, N. Iiirnagai, “A 3D Scanning Laser Rangefinder and its Application to an Autonomous Guided Vehicle”, Department of Mechatronics Engineering, Tohoku University, pp:331, Japan, 2000.

[4] I. Silvestre, J. I. Rodrigues, M. Figueiredo,C.Veiga-Pires, “Cave chamber data modeling and 3D Web Visualization”, 17th International Conference on Information Visualization, 2013.

[5] “Fluke 414D, 419D and 414D Distance Meters- Technical Data”, Fluke Corporation.

[6] “MeshLab release notes", official MeshLab Documentation wiki.

[7] B. D. Bradley, A.D.C. Chan, M.J. D. Hayes, “A Simple, Low Cost, 3D Scanning System Using the Laser Light-Sectioning Method”, IEEE International Instrumentation and Measurement Technology Conference. FMTC 2008, 12-15 May, 2008.

[8] G. Zuo, G. Shi, “Research on 3D Laser Scanning Field working method of Ancient Architecture Scene”, Anhui Institute of Architecture and Industry Hefei, China, 2010.

[9] L. C. Goron, L. Tamas, I. Reti, G. Lazea, “3D Laser Scanning System and 3D Segmentation of Urban Scenes”, Robotic Research Group, Technical University Cluj-Napoca.

[10] S. Izadi, R. Newcombe, D. Kim, O. Hilliges, D. Molyneaux, S. Hodges, P. Kohli, A. Davison, A. Fitzgibbon, “KinectFusion: Real-Time Dynamic 3D Surface Reconstruction and Interaction”, SIGGRAPH Talks, August, 2011.

[11] D. Ba, F. Ribeiro, C. Zhang, D. Florêncio, “L1 Regularized Room Modeling With Compact Microphone Arrays”, 2010.

[12] J. Du, Y. Ou, W. Sheng, “Improving 3D indoor mapping with motion data”, Robotics and Biomimetics (ROBIO), 2012, IEEE International Conference on, pp: 489 – 494, 11-14 Dec, 2012.

[13] X.J. Cheng, H. Zhang, R. Xie, “Study on 3D Laser Scanning Modeling Method for Large-Scale History Building”, International Conference on Computer Application and System Modeling (ICCASM 2010), 2010.

[14] J.G. Weingarten, G. Gruener, R. Siegwart. “A state-of-the-art 3D Sensor for Robot Navigation”, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots (IROS), vol. 3, pp. 2155- 2160, 2004.

[15] K. Surmann, K. Lingemann, A. Nüchter, J. Hertzberg, “A 3D Laser Range Finder for Autonomous Mobile Robots”, in Proceedings of the International Symposium on Robotics (ISR), 2001.

[16] P. Jensfelt, S. Kristensen, “Active Global Localisation for a Mobile Robot Using Multiple”,IJCAI, pp. 13-22, 1999.

[17] “LR4- Laser Rangefinder Interface”, Product Data Sheet, Porcupine Electronics, February 5, 2013.

[18] Schmidt, "Arduino: A Quick Start Guide", Pragmatic Bookshelf, 2011, Pg. 20.

[19] Noble G.Larson, “Stepping Motor Control System”. Massachusetts Institute of Technology, Feb,1979.

[20] Lady Ada, “Adafruit Motor Shield”, Adafruit Learning system.

[21] “Review B: Coordinate Systems”, Department of Physics, Massachusetts Institute Of Technology.

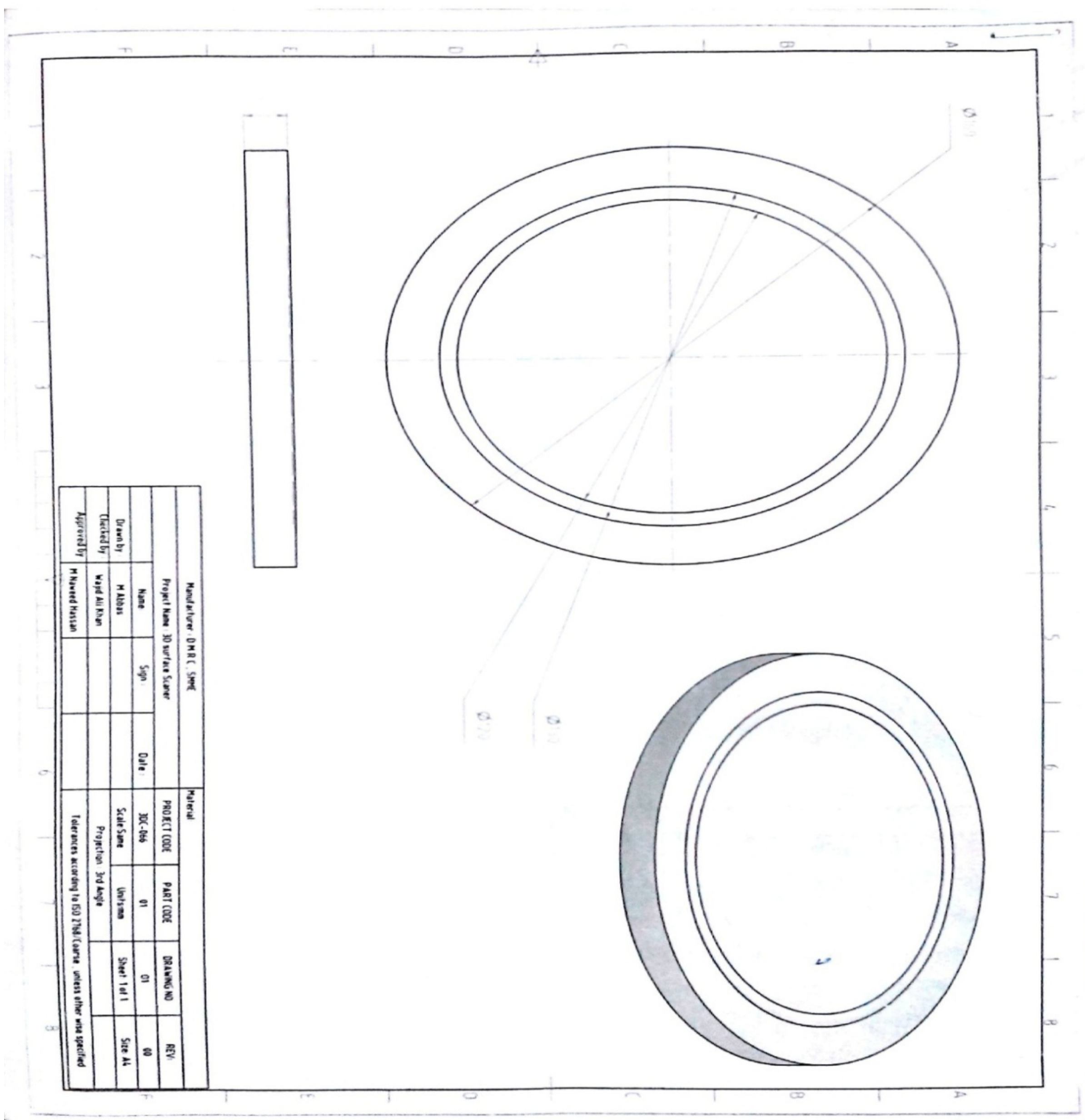
[22] M. Kazhdan, M. Bolitho, H. Hoppe, “Poisson Surface Reconstruction”, Eurographics Symposium on Geometry Processing, 2006.

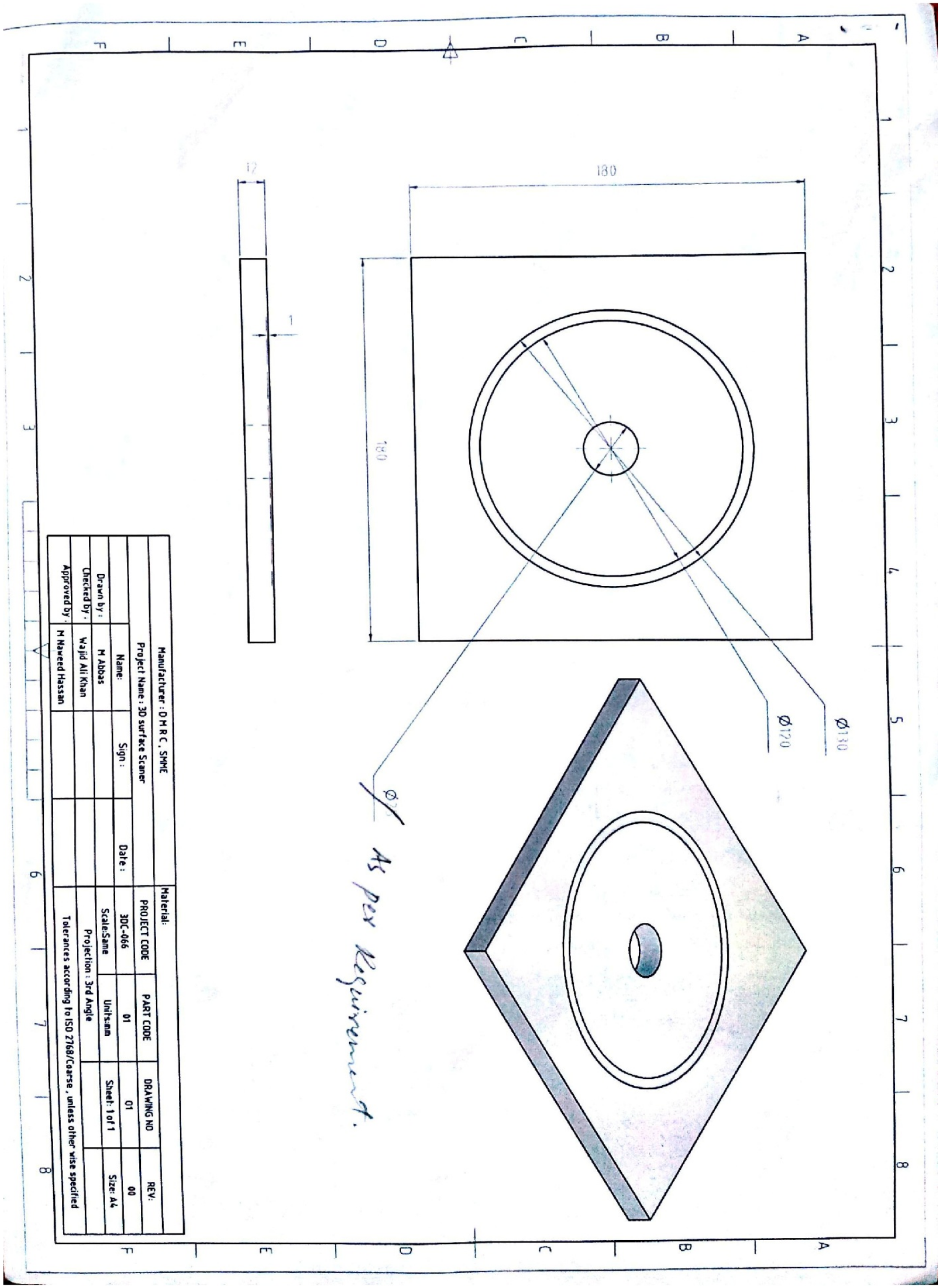
[23] N. Shafi, B. Noor, U. Mirza, M.A. Chatta, “3D Through The Wall Microwave Imaging”, National University Science and Technology, Pakistan, August, 2011.

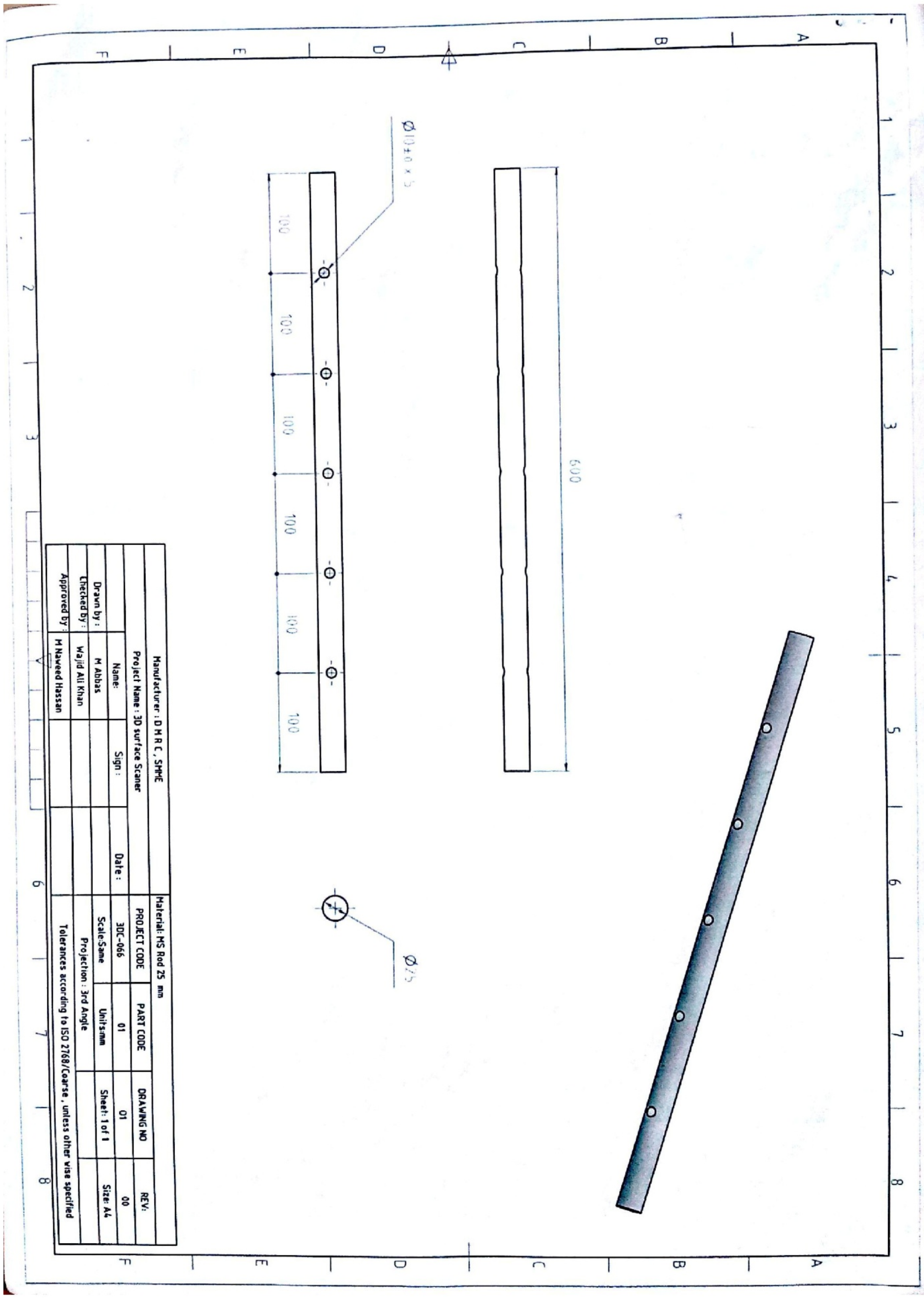
APPENDICES

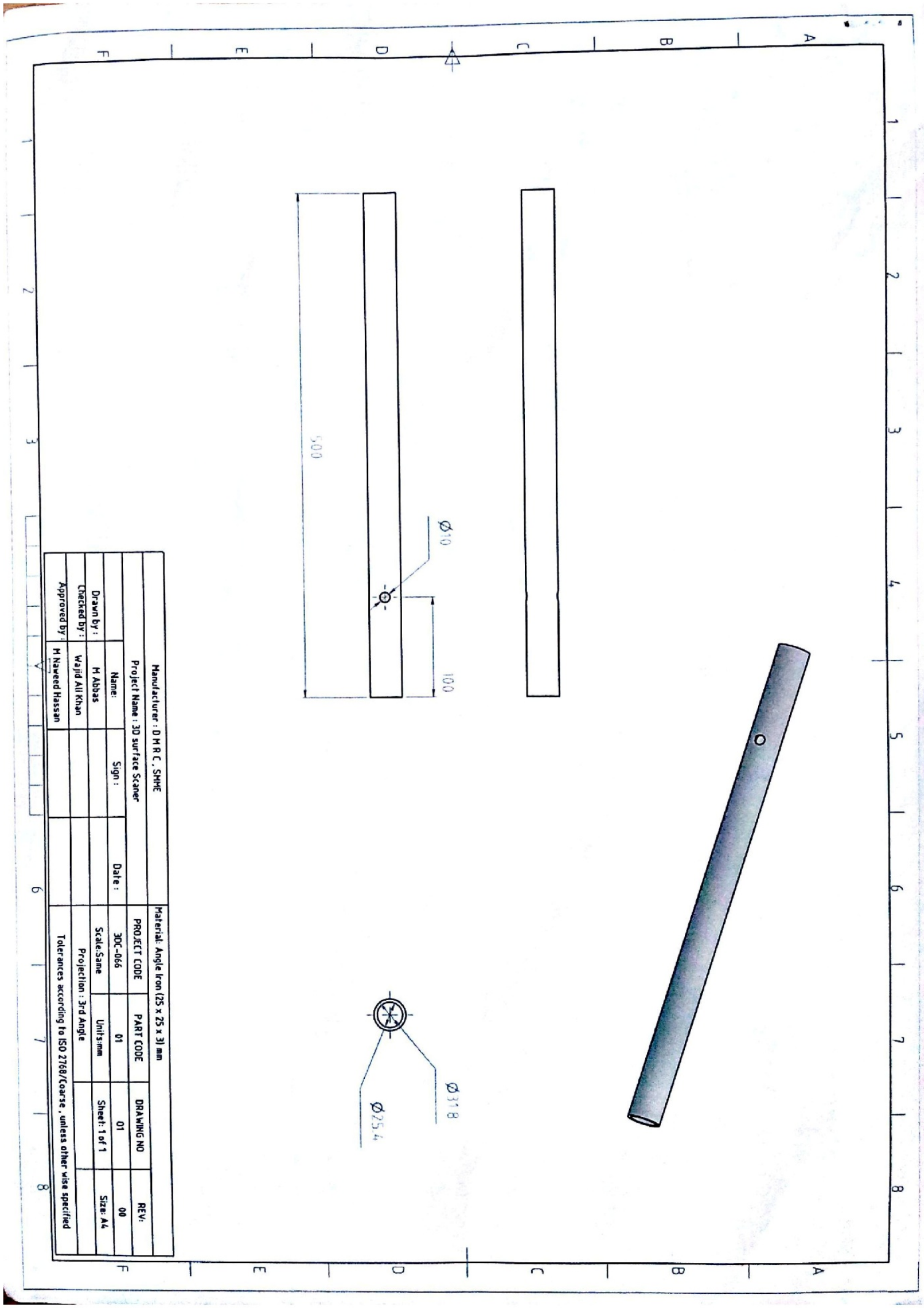
APPENDIX A

DRAWINGS OF ROTATING MECHANICAL ASSEMBLY




180
180
12
1
Ø130
Ø120
As per Requirement.
Manufacturer : D M R C , SMME
Project Name : 3D surface Scaner
Name:
Sign :
Date :
Drawn by :
M Abbas
Checked by :
Wajid Ali Khan
Approved by :
M Naveed Hassan
Material:
PROJECT CODE
PART CODE
DRAWING NO
REV:
3DC-066
01
01
00
Scale:Same
Units:mm
Sheet: 1 of 1
Size: A4
Projection : 3rd Angle
Tolerances according to ISO 2768/Coarse , unless other wise specified


Ø10±0 × 5
100
100
100
100
100
100
600
Ø25
Manufacturer : D M R C , SMME
Project Name : 3D surface Scaner
Name: Sign : Date :
Drawn by : M Abbas
Checked by : Wajid Ali Khan
Approved by : M Naweed Hassan
Material: MS Rod 25 mm
PROJECT CODE 3DC-066
PART CODE 01
DRAWING NO 01
REV: 00
Scale:Same
Units:mm
Sheet: 1 of 1
Size: A4
Projection : 3rd Angle
Tolerances according to ISO 2768/Coarse , unless other wise specified


500
Ø10
100
Ø31.8
Ø25.4
Manufacturer : D M R C , SMME
Project Name : 3D surface Scaner
Material: Angle Iron (25 x 25 x 3) mm
Name:
Sign :
Date :
PROJECT CODE
PART CODE
DRAWING NO
REV:
Drawn by :
M Abbas
3DC-066
01
01
00
Checked by :
Wajid Ali Khan
Scale:Same
Units:mm
Sheet: 1 of 1
Size: A4
Approved by :
M Naweed Hassan
Projection : 3rd Angle
Tolerances according to ISO 2768/Coarse , unless other wise specified

APPENDIX B

3D LASER SURFACE MAPPING- SYNOPSIS

Extended Title: 3D Surface Mapping using LiDAR
Brief Description of The Project / Thesis with Salient Specs: The purpose of the project is to design a system that scans a surface using a laser distance meter (LiDAR), and reconstruct the data to obtain a 3D image of the surface. Synchronized motion control based on stepper motors will be used to obtain distance measurements between LiDAR device and a specific point on the surface. These distance measurements will be used to construct a 3D representation of the surface.
Scope of Work : The system will consist of a LiDAR scanning system which is controlled by two stepper motors. Stepper motors will be controlled by the microcontroller which will move the device to a specific azimuth and elevation. The measurement of LiDAR device in the form of point clouds will be transferred from to the PC for further processing. The point cloud data will be utilized to construct a 3D image of the surface. A GUI will be constructed to control the overall system and its overall functionalities.

Academic Objectives :

Learning to construct 3D surface mapping using the following skills:

- Using the principle of LiDAR to obtain spatial information of a surface.
- Developing skills to scan in three dimensions using motors and a rotating assembly.
- Hands-on experience on Arduino board and Porcupine LR4, programming and interfacing with LiDAR, motors and PC.
- Creating Arduino to PC software interface for data processing.
- Surface re-constructing from point clouds.

Application / End Goal Objectives :

Exploration of unfamiliar and unknown environments through data acquisition in order to create a 3D surface map.

In addition to spatial information, other types of information are often desired such as temperature, color and light level may be included in the project.

There are numerous applications in which 3D surface mapping can be used. These are listed are as follows:

- Military applications include infiltration of sensitive installations, spying, providing access to unfriendly environments.
- Cave exploration, construction site surveying, forest mapping.
- Accessing bio-hazardous environments including toxic, radioactive

surroundings.

- Hazardous and hostile environments where human access is not possible for examples in disaster effected areas, in fire-fighting to detect hotspots.

Previous Work Done on The Subject :

- 3D reconstruction from 2D data, MCS (BESE-11)
- 3D through the wall microwave imaging, MCS (TCC-19)
- J. Lan, Z.Shen. "New approach of imaginary generation and target recognition based on 3D LIDAR data," The Ninth International Conference on Electronic Measurements. ICEMI '09, Vol.1, pp: 612~ 616, 16-19 August, 2009.
- J. Li-Chee-Ming, D. Gumerov, T. Ciobanu, C. Armenakis , "Generation of Three Dimensional Photo-Realistic Models from Lidar and Image Data" ,Science and Technology for Humanity (TIC-STH),2009 IEEE Toronto International Conference ,pp:445~450,26-27 September, 2009.

Material Resources Required :

- LiDAR (laser distance measurement device)
- Arduino Board
- Porcupine LR4
- Stepper Motors

• Personal Computer
No of Students Required : 3
Special Skills Required : • Arduino Board Programming • Porcupine LR4 Programming • Data processing in PC • Meshlab 3D image generation • Mechanical assembly building

Approval Status:

Supervisor Name: Lt Col. DR. Adil Masood

Supervisor Signature: ______________________________

Assigned to: ______________________________

____

____

HoD Signature: ______________________________

Coordinator Signature: ______________________________

APPENDIX C

COST BREAKDOWN

Sr No.	Product	Cost
1	Laser Distance Meter (LiDAR) – Fluke 414D	11000/-
2	LR4 Interfacing Board	15000/-
3	Stepper Motors	1200/-
4	Arduino Mega 2560	2000/-
5	Adafruit Motor Shield	2000/-
6	Assembly	15000/-
7	Miscellaneous	2000/-
	Total	**48200/-**

Table C-1 Cost Breakdown

APPENDIX D

DEMONSTRATION OUTLINE

Obtaining and Interfacing Fluke 414D with Porcupine LR4:

Fluke Laser Distance Meter 414D is used to obtain distance measurements. These measurements cannot be accessed directly from Fluke 414D therefore an interfacing board was used. To perform this task Porcupine LR4 is being used. Both 414D and LR4 were purchased online. After acquiring both the devices, Fluke 414D was unscrewed and its LCD display and keypad were unplugged. The battery wires were clipped and plugged into the LR4 without soldering to supply the laser distance meter with power. Flex cables were used to retrieve data from the meter via its screen connection and control the meter via its key pad connection. Fig D.1 showsFluke 414D interfaced with Porcupine LR4 board. As can be seen in the figure, Fluke 414D is connected to Porcupine LR4 via battery wires, data and control flex cables.

Figure D-1Fluke 414D interfaced with Porcupine LR4 [17]

The figure above shows laser distance meter, Fluke 414D, disassembled and interfaced to Porcupine LR4 through flex cables.

Obtaining And Displaying Distance Measurements

To obtain distance measurements from Porcupine LR4, the serial port of LR4 was initially used. The serial board on LR4 was connected to the serial ports on Arduino. A code had been developed, on Arduino IDE software, to serially transmit start and stop commands to LR4 and for serially receiving distance measurements from laser distance meter. However, the distance measurements could not be acquired successfully.

Therefore, another method was devised to obtain distance measurements through the USB port of Porcupine LR4. Porcupine LR4 was connected to a personal computer through its USB port. The LR4 demo software available online includes a Windows application, called LRSimple. It is a bare bones Win32 Console app that implements the simplest method to start the LR4 and read distance measurements. This source code was modified to create text files (.txt) which would store distance measurements serially. Figure D.2 below shows a Win 32 Console displaying distance measurements obtained by the data acquisition unit.

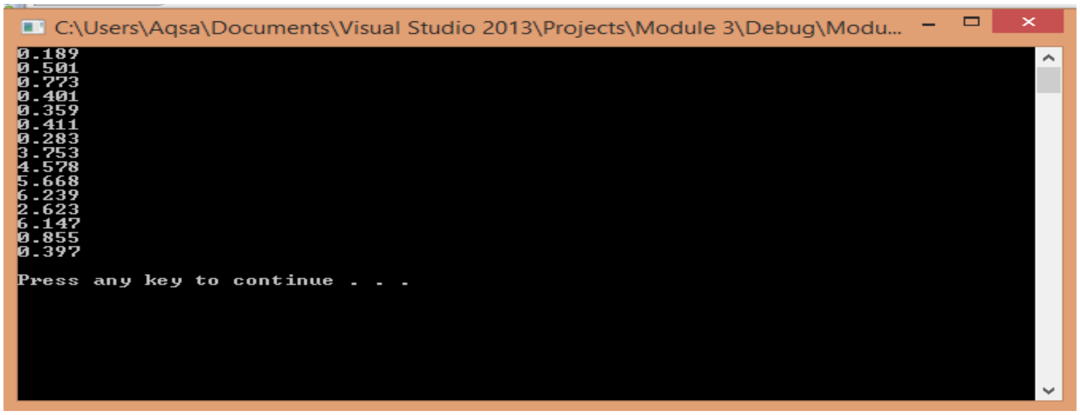


Figure D-2 Distance measurements displayed in Win32 console

Figure D.2 represents distance measurements displayed on Win32 console app. The values are floating point numbers, up to three decimal places. They also vary greatly in value according to distances measured.

Acquiring And Storing Spatial Coordinates

A C++ code is developed to store distance measurements and spatial coordinates of laser distance meter in a text file. The distance measurements, which are equivalent to the Radius in Spherical Coordinate system, are stored in a text file followed by Theta and Phi coordinates. These measurements are obtained from the laser distance meter and angular distance covered by the laser distance meter from its initial position respectively. Figure E.3 shows data points in spherical coordinates stored in a text file.

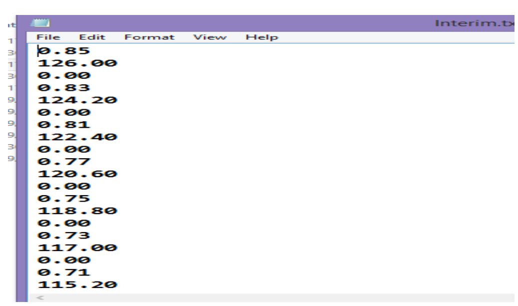


Figure D-3 Data points stored in text file

The figure above shows data points, radius, theta and phi, stored in a text file. The first value represents radius followed by theta and then phi. The pattern then repeats for the rest of the file.

Building A Rotating Mechanical Assembly

A rotating mechanical assembly is built to scan in three dimensions. The assembly consists of four mild steel rods upon which a square aluminum plate is fixed. A rotating aluminum plate with grooves for bearings is placed on top of the fixed plate to which the stepper motor will be coupled. Hence, the incremental motion of motor shaft causes the rotating plate to rotate in steps in the azimuth plane. The laser distance meter is screwed on a base plate which is attached to the motor to move it incrementally in the elevation plane. The length of the assembly can be varied by varying the length of the fixed rods up to one meter. The total height of the assembly can be varied up to four different levels. The design described above was made in Auto-Cad of the various parts involved. The designs of the parts of the rotating mechanical assembly are given in Appendix A.

Motors And Motion Control

A mechanical assembly is being created which will be able to rotate in two degrees of freedom using stepper motors. For rotation in the azimuth plane, Minebea K202 stepper motor was acquired. It has 1.8 degree step size and will complete 100 steps in a complete revolution. The step size will be varied according to user requirement. For rotation in the elevation plane, Minebea K502 stepper motor was acquired. It has 1.8 degree step size and will complete 150 steps to span 270 degrees.

Both these motors are connected to Adafruit Motor Shieldand Arduino Mega2560. These motors are controlled through PWM signals from the Arduino. The Adafruit Motor Shield pins are inserted into Arduino and it is placed on top of it. The four wires of stepper motor are connected to the motor shield. A code has been developed on Arduino IDE softwareto control motor speed and no of steps per revolution. Fig D.4 shows the stepper motor being controlled by Adafruit motor shieldwhich is stacked on Arduino board.

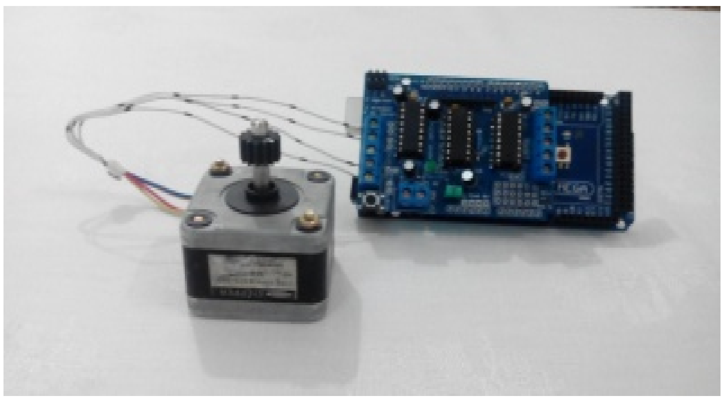


Figure D-4 Stepper motor interfaced with Arduino
Mega 2560 and Adafruit Motor Shield

Figure D.4 above shows a bipolar stepper motor connected to Adafruit motor shield via its four wires.

Creating a Graphical User Interface

An interactive Graphical User Interface is being created on Visual Studio 2012. A Windows application is being created which will provide users control of the mechanical assembly and laser distance meter. The various options being included in the 3D Laser Surface Mapping application are mentioned below:

- Start and stop the 3D Laser Surface Mapping system.
- To choose one scan from three different scans available. The different scans have varying degree of accuracy and time requirements to complete the scan. Based on user requirement any one of the scans can be selected.
- Select position of 3D laser surface mapping system.
- Displays distance measurements in three dimensions in spherical coordinate systems and Cartesian coordinate system.

Data Conversion

Data points in spherical coordinate system stored in text file will be read by a C Code developed in Visual Studio. These data points are then converted to Cartesian coordinates through basic formulas of coordinate conversion. The calculated data points in Cartesian coordinates are stored in a separate text file. Figure D.5 below shows data points in Cartesian coordinates stored in a text file.

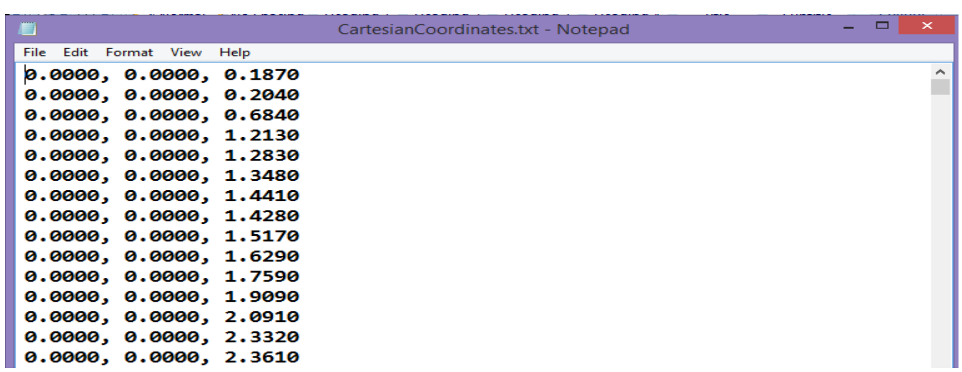


Figure D-5 Data points in Cartesian coordinates stored in a text file

The figure above represents data points in Cartesian coordinates stored in a text file. Each line represents x, y and z coordinates respectively separated by commas. This file is then exported to MeshLab to create a 3D model.

Creating 3D Model

The text file containing data points in Cartesian coordinates is then imported to MeshLab, a 3D modeling software. This file is opened by selecting XYZ as reference plane, comma as separator between coordinates and zero header rows to be skipped. The data points are plotted in a two dimensional plane. Through the Manipulator's tool in the main toolbar, the data points are translated in a three dimensional plane. The point clouds can be viewed from various angles. By removing redundant points, creating normals and using surface reconstruction Poisson filter a 3D model is created from the given cloud points. The mesh is further processed by applying cleaning filters, removing redundant faces and adding new layers to improve the mesh. This model can be viewed from different angles and zoomed in and out for detailed analysis. The figure below shows a 3D model of scanned environment created in MeshLab.

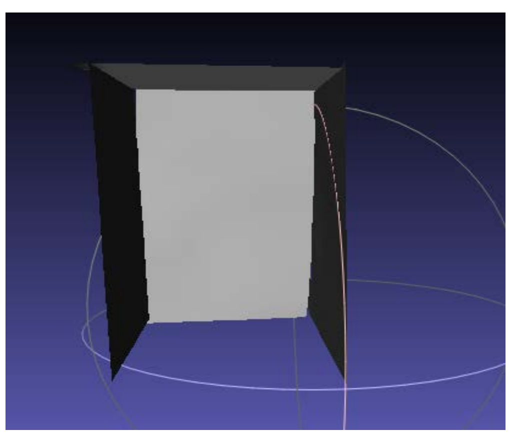


Figure D-63D model created from cloud points

As can be seen in the figure above, a 3D model is created from the point clouds imported to MeshLab.

APPENDIX E

PROJECT CODE

```
// Adafruit Motor shield library
// copyright Adafruit Industries LLC, 2009
// this code is public domain, enjoy!

#include <AFMotor.h>
#include <CustomStepper.h>

AF_Stepper motor1(200, 1);

CustomStepper stepper(4, 7, 8, 12);
void setup()
{
Serial.begin(9600);  // set up Serial library at 9600 bps
        motor1.setSpeed(1); // 1 rpm
}
void loop()
{
        delay(100);
        //for (int k=0; k<=25; k++)
        if(Serial.available() > 0)
        {
        char Rx = Serial.read();
        switch (Rx)
```

```
{
case 1:       //FOR 45 DEGREE ANGLE
{ AF_Stepper motor2(200, 2);
motor2.setSpeed(1); // 1 rpm
motor2.step(25, BACKWARD, SINGLE);

for (int j = 0; j < 100; j++)
{
        delay(1000);
        motor1.step(1, FORWARD, SINGLE);

        for (int i = 0; i <= 150; i++)
        {
                delay(100);
                if ((j % 2) == 0)
                        motor2.step(1, FORWARD, SINGLE);
                if ((j % 2) == 1)
                        motor2.step(1, BACKWARD, SINGLE);
                if (j == 100)
                        stepper.setDirection(STOP);
        }
}
}
        break;

case 2:
{   AF_Stepper motor3(200, 2);
```

```
  motor3.setSpeed(1); // 1 rpm
  motor3.step(25, BACKWARD, SINGLE);

  for (int j = 0; j < 100; j++)
  {
        delay(1000);
        motor1.step(1, FORWARD, SINGLE);

        for (int i = 0; i <= 75; i++)
        {
              delay(100);
              if ((j % 2) == 0)
                    motor3.step(1, FORWARD, MICROSTEP);
              if ((j % 2) == 1)
                    motor3.step(1, BACKWARD, MICROSTEP);
              if (j == 100)
                    stepper.setDirection(STOP);
        }
  }
  }
break;

  case 3:
  {   AF_Stepper motor4(400, 2);
  motor4.setSpeed(1); // 1 rpm
  motor4.step(25, BACKWARD, SINGLE);
```

```
for (int j = 0; j < 100; j++)
{
        delay(1000);
        motor1.step(1, FORWARD, SINGLE);

        for (int i = 0; i <= 67; i++)
        {
                delay(100);
                if ((j % 2) == 0)
                        motor4.step(1, FORWARD, MICROSTEP);
                if ((j % 2) == 1)
                        motor4.step(1, BACKWARD, MICROSTEP);
                if (j == 100)
                        stepper.setDirection(STOP);
        }
}
}
        break;

case 4: //FOR 35 DEGREE ANGLE
{   AF_Stepper motor2(200, 2);
motor2.setSpeed(1); // 1 rpm
motor2.step(20, BACKWARD, SINGLE);

for (int j = 0; j < 100; j++)
{
        delay(1000);
```

```
        motor1.step(1, FORWARD, SINGLE);

        for (int i = 0; i <= 140; i++)
        {
            delay(100);
            if ((j % 2) == 0)
                motor2.step(1, FORWARD, SINGLE);
            if ((j % 2) == 1)
                motor2.step(1, BACKWARD, SINGLE);
            if (j == 100)
                stepper.setDirection(STOP);
        }
    }
    }
        break;

    case 5:
    {
              AF_Stepper motor2(200, 2);
              motor2.setSpeed(1); // 1 rpm
              motor2.step(25, BACKWARD, SINGLE);

              for (int j = 0; j < 100; j++)
              {
                  delay(1000);
                  motor1.step(1, FORWARD, SINGLE);
```

```
                    for (int i = 0; i <= 75; i++)
                    {
                        delay(100);
                        if ((j % 2) == 0)
                            motor2.step(1, FORWARD,
MICROSTEP);
                        if ((j % 2) == 1)
                            motor2.step(1, BACKWARD,
MICROSTEP);
                        if (j == 100)
                            stepper.setDirection(STOP);
                    }
                }
    }
        break;

    case 6:
    {   AF_Stepper motor2(400, 2);
    motor2.setSpeed(1); // 1 rpm
    motor2.step(25, BACKWARD, SINGLE);

    for (int j = 0; j < 100; j++)
    {
        delay(1000);
        motor1.step(1, FORWARD, SINGLE);

        for (int i = 0; i <= 67; i++)
        {
```

```
delay(100);
if ((j % 2) == 0)
        motor2.step(1, FORWARD, MICROSTEP);
if ((j % 2) == 1)
        motor2.step(1, BACKWARD, MICROSTEP);
if (j == 100)
        stepper.setDirection(STOP);
    }
}
}
    break;

case 7:  //FOR 25 DEGREE ANGLE

{
            AF_Stepper motor2(200, 2);
            motor2.setSpeed(1); // 1 rpm
            motor2.step(14, BACKWARD, SINGLE);
            for (int j = 0; j < 100; j++)
            {
                delay(1000);
                motor1.step(1, FORWARD, SINGLE);

                for (int i = 0; i <= 128; i++)
                {
                    delay(100);
                    if ((j % 2) == 0)
```

```
                        motor2.step(1, FORWARD,
DOUBLE);
                    if ((j % 2) == 1)
                        motor2.step(1, BACKWARD,
DOUBLE);
                    if (j == 100)
                        stepper.setDirection(STOP);
                }
            }
    }
        break;

    case 8:
    {   AF_Stepper motor2(200, 2);
    motor2.setSpeed(1); // 1 rpm
    motor2.step(25, BACKWARD, SINGLE);

    for (int j = 0; j < 100; j++)
    {
        delay(1000);
        motor1.step(1, FORWARD, SINGLE);

        for (int i = 0; i <= 75; i++)
        {
            delay(100);
            if ((j % 2) == 0)
                motor2.step(1, FORWARD, MICROSTEP);
            if ((j % 2) == 1)
```

```
                motor2.step(1, BACKWARD, MICROSTEP);
            if (j == 100)
                stepper.setDirection(STOP);
        }
}

}
        break;

case 9:
{       AF_Stepper motor2(400, 2);
motor2.setSpeed(1); // 1 rpm
motor2.step(25, BACKWARD, SINGLE);

for (int j = 0; j < 100; j++)
{
        delay(1000);
        motor1.step(1, FORWARD, SINGLE);

        for (int i = 0; i <= 67; i++)
        {
            delay(100);
            if ((j % 2) == 0)
                motor2.step(1, FORWARD, MICROSTEP);
            if ((j % 2) == 1)
                motor2.step(1, BACKWARD, MICROSTEP);
            if (j == 100)
```

```
                                        stepper.setDirection(STOP);
                }
        }
        }
                break;

        case 'a'://FOR 15 DEGREE ANGLE
        {       AF_Stepper motor2(200, 2);
        motor2.setSpeed(1); // 1 rpm
        motor2.step(8, BACKWARD, DOUBLE);

        for (int j = 0; j < 100; j++)
        {
                delay(1000);
                motor1.step(1, FORWARD, SINGLE);

                for (int i = 0; i <= 112; i++)
                {
                        delay(100);
                        if ((j % 2) == 0)
                                motor2.step(1, FORWARD, DOUBLE);
                        if ((j % 2) == 1)
                                motor2.step(1, BACKWARD, DOUBLE);
                        if (j == 100)
                                stepper.setDirection(STOP);
                }
        }
```

```
}
        break;

case 'b':
{       AF_Stepper motor2(200, 2);
motor2.setSpeed(1); // 1 rpm
motor2.step(25, BACKWARD, SINGLE);

for (int j = 0; j < 100; j++)
{
        delay(1000);
        motor1.step(1, FORWARD, SINGLE);

        for (int i = 0; i <= 75; i++)
        {
                delay(100);
                if ((j % 2) == 0)
                        motor2.step(1, FORWARD, MICROSTEP);
                if ((j % 2) == 1)
                        motor2.step(1, BACKWARD, MICROSTEP);
                if (j == 100)
                        stepper.setDirection(STOP);
        }
}
}
        break;
```

```
        case 'c':
        {       AF_Stepper motor2(400, 2);
        motor2.setSpeed(1); // 1 rpm
        motor2.step(25, BACKWARD, SINGLE);

        for (int j = 0; j < 100; j++)
        {       delay(1000);
                motor1.step(1, FORWARD, SINGLE);

                for (int i = 0; i <= 67; i++)
                {
                        delay(100);
                        if ((j % 2) == 0)
                                motor2.step(1, FORWARD, MICROSTEP);
                        if ((j % 2) == 1)
                                motor2.step(1, BACKWARD, MICROSTEP);
                        if (j == 100)
                                stepper.setDirection(STOP);
                }
        }
        }
                break;
        default:
                break;
                }
                }
}
```