UNIVERSITY OF GAZİANTEP

GRADUATE SCHOOL OF

NATURAL & APPLIED SCIENCES

DEVELOPMENT OF 3D LASER SCANNER USING LASER RANGE FINDER

M. Sc. THESIS
IN
ELECTRICAL-ELECTRONIC ENGINEERING

BY
ALİ EMRE ÖZTÜRK
JANUARY 2014

Development of 3D Laser Scanner using Laser Range Finder

M.Sc. Thesis
in
Electrical-Electronic Engineering
University of Gaziantep

Supervisor
Prof. Dr. Ergun ERÇELEBİ

by

Ali Emre ÖZTÜRK

January 2014

UNIVERSITY OF GAZİANTEP
GRADUATE SCHOOL OF
NATURAL & APPLIED SCIENCES
ELECTRICAL-ELECTRONIC ENGINEERING DEPARTMENT

Name of the thesis: Development of 3D Laser Scanner using Laser Range Finder
Name of the student: Ali Emre ÖZTÜRK
Exam date: 17.01.2014

Approval of the Graduate School of Natural and Applied Sciences

Assoc. Prof. Dr. Metin BEDİR
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

E. Ercelel -
Prof. Dr. Ergun ERÇELEBİ
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

E. Ercelel -
Prof. Dr. Ergun ERÇELEBİ
Supervisor

Examining Committee Members — Signature

Prof. Dr. Ergun ERÇELEBİ — E. Ercelel -

Assoc. Prof. Dr. Metin BEDİR ____________

Assoc. Prof. Dr. Nuran DOĞRU ____________

Asst. Prof. Dr. Sema KAYHAN ____________

Asst. Prof. Dr. A. Mete VURAL ____________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Ali Emre ÖZTÜRK

ABSTRACT

DEVELOPMENT OF 3D LASER SCANNER USING LASER RANGE FINDER

ÖZTÜRK, Ali Emre
M.Sc. in Electrical-Electronics Engineering
Supervisor: Prof. Dr. Ergun ERÇELEBİ
January 2014, 60 pages

Generally, 3D laser scanners are used to create 3D model file of any object or any environment to help architecture or robot navigation. A 3D model can be generated using one camera and laser light source or stereo camera or a laser distance meter.

In this study, Laser scanner hardware has been designed and implemented to create a 3D model in AutoCAD software using a laser range finder. The hardware can generate 3D modeling of environment using the distance measures from different points on spherical coordinate plane. Two axis (φ, θ) movable mechanical parts have been designed and implemented to get distance measurement from different points. Step motor driver has been developed to control the step motors that provide the movement of mechanical parts. Another electronic card has been developed to provide duplex communication between computer and electronic device. A computer software and GUI have been coded that are able to generate modeling using collected data. Approximate modeling of environment can be generated by computer software owing to data that are collected by scanner. These datasets contain necessary distance information to detect borders, barriers or proper ways for a robotic system that roams environment. A laser scanner can obtain information about distance between objects in environment. Some information about environment like slope of ground or object distribution can be analyzed by these datasets

Keywords: 3D laser scanner, robotic systems, AutoCAD.

ÖZET

LAZER MESAFE BULUCU KULLANARAK 3 BOYUTLU LASER TARAYICI TASARIMI

ÖZTÜRK, Ali Emre
Yüksek Lisans Tezi, Elektrik-Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Ergun ERÇELEBİ
Ocak 2014, 60 sayfa

3 boyutlu lazer tarayıcılar genellikle herhangi bir nesnenin veya herhangi bir bölgenin modelini oluşturmak için kullanılır. 3 boyutlu bir model bir kamera ve lazer ışık kaynağı veya çift kamera veya bir lazer mesafe ölçer kullanılarak oluşturulabilir. Bu çalışmada, lazer mesafe ölçer kullanarak AutoCAD programında 3 boyutlu model oluşturan bir donanım tasarlandı ve uygulandı. Donanım küresel koordinatta farklı noktalardan ölçülen uzaklık bilgilerini kullanarak ortamın 3D modellemesini oluşturabilmektedir. Farklı noktalardan ölçüm alabilmek için 2 eksende (φ, θ) hareketli mekanik kısım tasarlanmıştır. Mekanik kısmın hareketini sağlayan step motorlar kontrol etmek için step motor sürücü devresi geliştirilmiştir. Bilgisayar – tarayıcı arasında çift taraflı iletişimi sağlamak için başka bir elektronik kart geliştirilmiştir. Toplanan bilgileri kullanarak modelleme oluşturabilecek bir ara yüz ve bilgisayar yazılımı kodlanmıştır. Alınan veriler sayesinde ortamın yaklaşık modellemesi yapılabilmektedir. Bu veri kümesi robotik bir sistemin bölgede dolaşırken sınırları, engelleri veya kendisi için uygun yolları tespit edebilmesi için gerekli uzaklık bilgilerini içerir. Bir lazer tarayıcı uzaklık, nesneler arası uzaklık gibi bilgileri sağlayabilir. Bu bilgileri yorumlayarak çevreyle ilgili eğim, nesne yoğunluğu gibi bilgilerin analizini yapabilir.

Anahtar Kelimeler: 3D lazer tarayıcı, , OOP, robotik sistemler, AutoCAD.

ACKNOWLEDGEMENT

I express sincere appreciation to my supervisor Prof. Dr. Ergun ERÇELEBİ for his continuous support, guidance and encouragement. Also my great thanks go to him for his advice and suggestions.

I would like to thank to my father, my mother and my sister for their patience and support. Their continuous encouragement and advice have helped me to complete this thesis and I would like to thank to my friends for their supports.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS/ABBREVIATIONS

LRF	Laser range finder
2D	two dimensions
3D	three dimensions
VRML	Virtual Reality Modeling Language
AEC	Architecture Engineering and Constructions
bps	Bit per second
CW	Counter wise
CCW	Counter clock wise
SATA	Serial Advanced Technology Attachment
LPT	Line Printer
USB	Universal Serial Bus
UI	User Interface
API	Application Programming Interface
ADC	Analog to Digital Convertor
CNC	Computer Numerical Control

CHAPTER 1

INTRODUCTION

The different visual techniques such as sensors, cameras and 3D laser scanner are used by robotic systems to sense the environment variables (Xiaofeng Lian, 2008). The techniques have advantage and disadvantages. The visual techniques should be selected due to environment and where to us. Although cameras have high level view angle, they produce 2D images including height and width data. Of course, for generation 3D image from 2D images there are algorithms in literature. View angle of laser scanners is high level and getting data are 3D image. The size of object can be detected by height, depth and width data. In addition to detection of size of object, the distance between objects in scanned environment can be measured. The image obtained by laser scanner can be used by robot to detect barrier and find its path. Also, 3D laser scanners can be used to define mountain surface, building structure, or city plan.

In this study, the scanner can model a 10 m radius area and this view can be modeled in 3D in AutoCAD. Scanner consists of a 2D laser range finder (LRF), motors that let move it over those axes, electronic cards and software. The name of the LRF model which is used is SICK DT 00P1113 is able to measure the only one point of the range image. LRF is fixed on the mechanic component which is controlled by 2 step motors. (Fig.1.1) while the step motors are rotating 0.9 degree in each segment, mechanic component is the mechanism that rotates in the angle of θ and φ and it is able to scan in the spherical area. The scanning area is shown in Figure 1.2. This scanning area can be changed with the help of computer software.

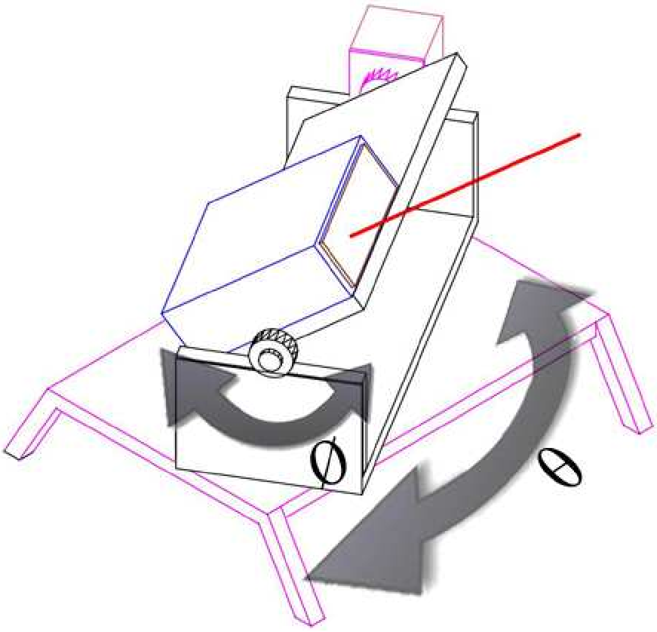


Figure1.1 2D movable mechanical part that named tilt

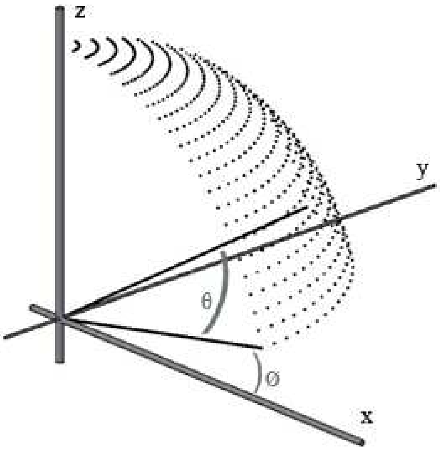


Figure1.2 an example of scanning area

Two electronic cards were developed to control the step motors. One of the cards is power card of step motors. The other one is control card which synchronization of step motors. It called laser scanner hardware. The duty of the step motor power card is to provide a motion for step motor with command that coming from laser scanner hardware. The hardware sends one pulse command for each turn to the driver and when the pulse comes to power card, the card provides 0.9 degree rotation to the step motor. The angle of step motor affects the image resolution. When step angle is smaller, 3D model is going to be more quality and will be able to catch sensitive details of environment. The hardware has a 48 MHZ processor which is programmed by using ‘C’ programming language. Another to duty of laser scanner hardware is providing to communication between laser scanner and computer. This communication is provided using by RS232 port on duplex communication. The communication port is used to transfer data packets which are consist of setting parameters and measured distance information. The information packets are sent between computer and laser scanner hardware. Distance information that comes from the device which is one of the functions of software transforms the spherical coordination level information to Cartesian coordination level. The used software is improved in .NET platform by using C # computer language. AUTOCAD is a programme which is CAD structured which is used in three dimensional objects, building, and machine drawing that belonged to the Autodesk firm. With the help of the API library which was established for the NET platform by Autodesk firm, the relation between .NET and AutoCAD is established and accessible for all commands.

CHAPTER 2

LITERATURE SURVEY

Laser scanners are being used in many areas such as defense industry, robotic systems, reverse engineering, manufacturing architecture, banking and finance, etc. Furthermore Laser scanners find application areas in modeling a mechanical object, road direction of the robots, restructure studies, security systems that have sense motion. Therefore, the important things for laser scanners are the speed and resolution. When they are needed to make more clear modeling, laser scanners systems can be used with cameras.

The aim of this study is to design and produce a low cost laser scanner. Kazunori Ohno, 2008 has used 2D Laser Range Finer (LRF) to model the environment (Kazunori Ohno, 2008). These types of laser scanners are expensive than 1D LRF and because of this the cost will be more expensive to design it with 2D LRF.On the other hand 3d models which has created using 2D LRF are more quality(Ohno, K., 2007). In addition the electronic cards used in these systems are for general purpose applications and this may also increase the costs. To reduce the costs control cards should be used for the purposes that are need. In this study, it has been aimed to reduce the costs with producing our own electronic cards and using 1D LRF instead of 2D LRF.

2.1 Robotic systems

A 3D model can convey much more useful information than the typical 2D maps (e.g. occupancy grids) used in most current mobile robotic applications. (Peter Biber, 2004). Unified version of the scanners separately shown in Figure 2.1

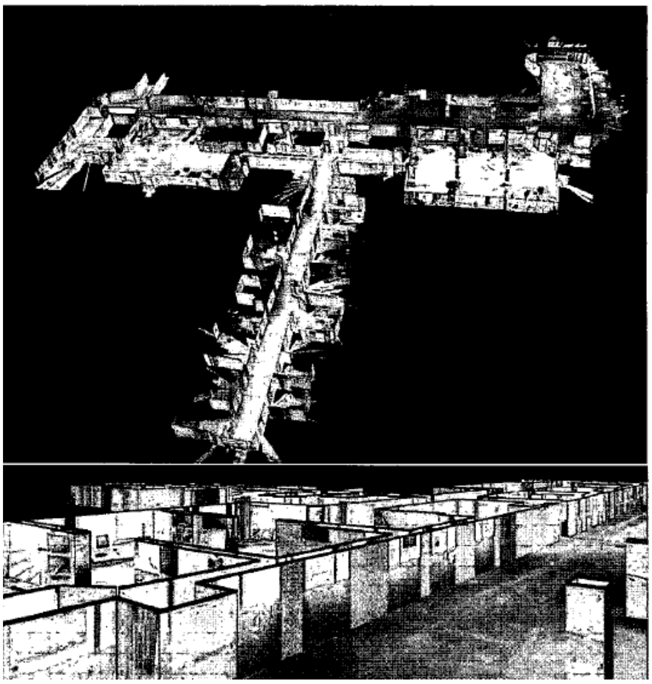


Figure2.1 Two views of the resulting VRML model (Peter Biber, 2004)

Mobile robot self-localization and navigation have become research hot of intelligent robot. 3D reconstruction is the base of mobile robot self-localization technology and provides the environment modeling for mobile robot navigation and obstacle avoidance research. As to the control system of mobile robot, which is mainly the PC control system in the past, more and more embedded system, including embedded hardware and embedded software, is applied to mobile robot control (Jianhua Wang, 2008).

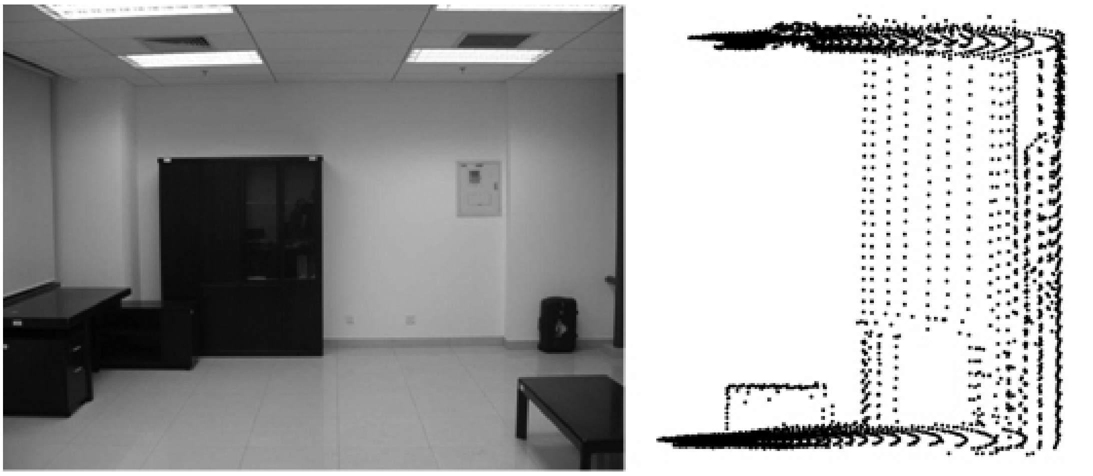


Figure2.2 Real Environment and 3D point cloud

2.2 Architecture

Laser scanners and ASER scanners are rapidly gaining acceptance as a tool for three dimensional (3D) modeling and analysis in the architecture, engineering, and construction (AEC) domain (Daniel Huber, 2010).

2.3 Security

A three-dimensional laser scanning system has been developed as a test bed for producing 2D and 3D profiles of human, animals, and vehicles. This type of device could be used in border security applications as part of a system for detecting and classifying border crossings. The system will be used to produce 2D and 3D profiles of people, people with backpacks, and animals.(Thomas Layton, 2009).

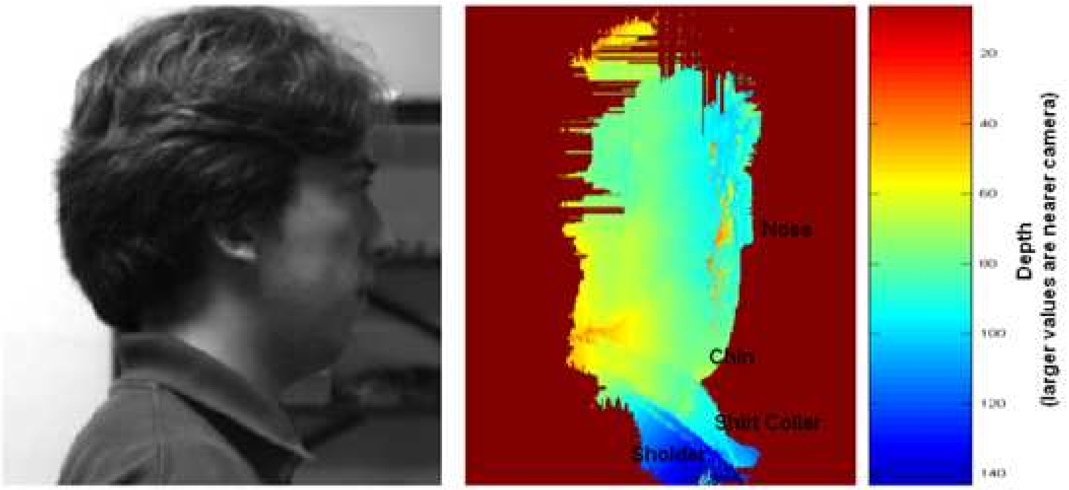


Figure2.3 3D close range image of a human(Thomas Layton 2009).

CHAPTER 3

METHODS

3.1 Components of 3D Laser Scanner

3D Laser scanner consists of LRF, mechanical parts, power supply, and two electronic cards. This hardware is connected to software which is in the computer via RS232 port. These components and laser scanner hardware run in real-time duplex communication. The schema of hardware is shown in Figure 3.1.1.

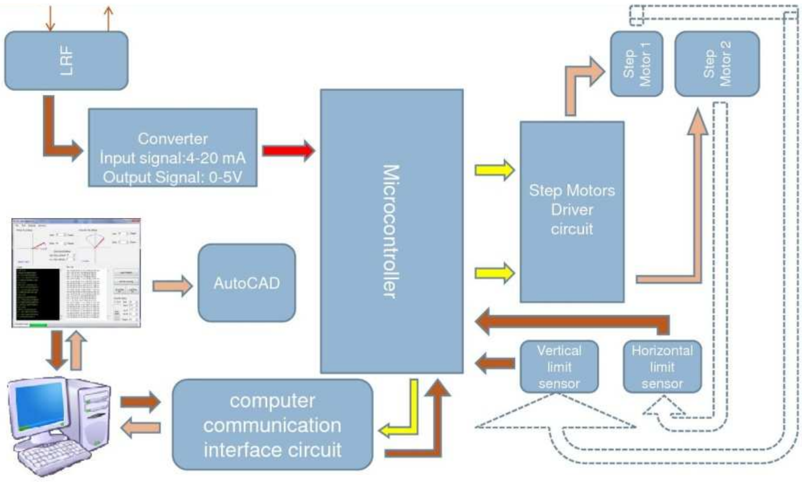


Figure3.1.1 Laser scanner device modeling

3.2 LRF (Laser Range Finder)

In this Project 1D laser range finder is used to measure distance of point. The product code of LRF is DT50-P1113, and the measurement range is 0.02 and 10 meter. LRF measures the distance using laser light. Technical feature of the LRF is given in the Table 3.1. LRF that we used has an analog output and additionally it has one digital output but analog output is enough for our study. LRF used in this study gives output between the range of 4 and 20 mA. 20 mA is equal to 10 meter, 4 mA can be equal to 0.02 meter. Since the micro controller is able to read analog voltage rather than analog current we have changed the 4-20 mA output signals to 0-5 voltage output signal. We have used a basic circuit so as to do this transformation. This circuit show in fig. 3.2.2 Owing to circuit the current turns into a voltage and it can be easily read by the controller. How to transform the 4-20 mA output signal into 0-5 V analog system signal is given in equation 3.2.1.As seen in the equation, the voltage range for analog signal is between 0.88- 4.4V. In this case, 0.02m is equal to 0.88 V and 10 meter is equal 4.4 V.

$$Min:(4mAx220ohm = 0.88V), Max:(20mAx220ohm = 4.4V) \qquad (3.2.1)$$

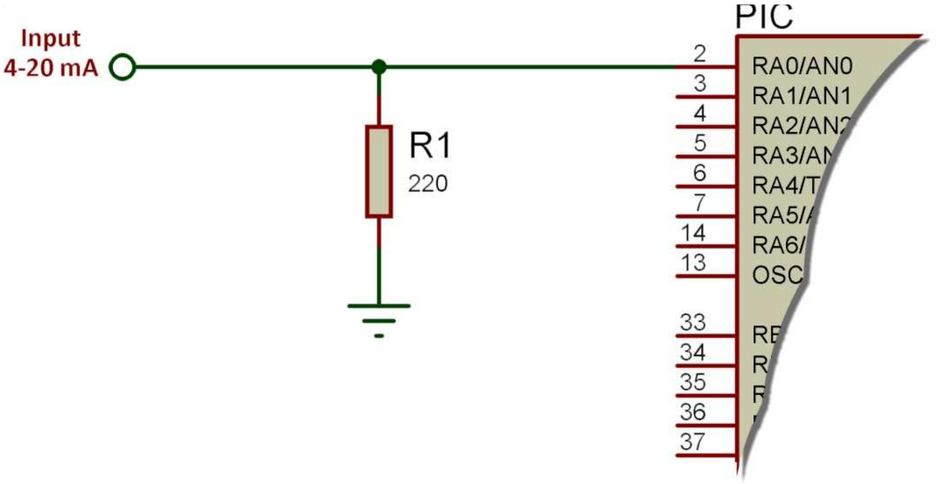


Figure3.2.2 4-20mA convert to analog signal

Table3.1Features of Laser range finder (SICK)

Measuring range	200 … 10,000 mm
Resolution	1mm
Accuracy	±10mm
Response time	20/30 ms
Output rate	4ms
Resolution analog outputs	16 bit
Laser light wavelength	650 nm

3.3 2D Movable Mechanical Part

A mechanical device has been designed in a way that can move in both axes in order to provide the LRF spherical coordinator motion. This designed device has two turn able parts and they turn along with vertical (θ) and horizontal (φ) axis. With the help of this LRF θ can move with the range of 0, +170 and φ angle can move with the range of -180+180. This mechanic mechanism is shown in the Figure 3.3.1. These angle ranges provide us wide range of scanning. We can think of the area that we are going to scan as cube which has 10 meter radius. When the scanning area increases, number of measurement increases as well. The formulation of this relation is given equation 3.3.1.

$$\#\text{ of measured point} = \frac{Vertical\_angle(\text{Ø\_axis})}{0.9} x \frac{Horizontal\_angle(\theta\_axis)}{0.9} \quad (3.3.1)$$

Example of calculation: to 0+170 Angle Ø axis, and -180+180 θ axis

$$\frac{170}{0.9} x \frac{360}{0.9} = 75600 \text{ Points.}$$

Aluminum material has been used to produce 2D tilt. The weight of the sensor that has been moved onto the flexible parts is 200 gr. That is why there has been some problem with the balance of the mechanic mechanism after the sensor has been replaced. Therefore, motors that move the axes become to drive more current. It even makes the motion more difficult. In order to solve this problem, 200 gr weights have been fixed to the point where the sensor coordination of the rotate axes symmetric is taken.

3.4 Power supply

The duty of the power supply is to provide necessary energy. The hardware needs energy that provides by power supply. Input voltage of the power supply is 220 V AC. The power supply provides 24 V-DC and 12 V-DC. 24 V-DC and 12 V-DC have been required to operate LRF and laser scanner hardware. To operate step motor driver 5V-DC and 12V DC. For all operations, required total power is 39.1 W. You can see the power requirement schema of hardware in figure 3.3.1.

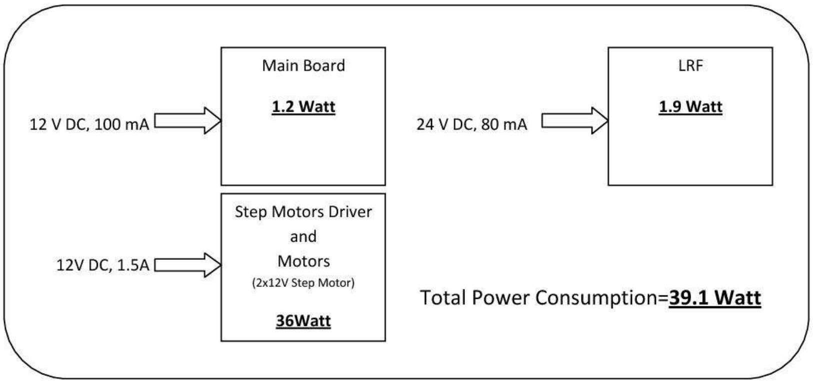


Figure 3.3.1 Power requirement of hardware

3.5 Design of laser scanner hardware and embedded software

Laser scanner hardware controls the moveable mechanical tilt (2D). In additional, it provides the communication between computer and components of laser scanner. It works as a transceiver that can receive and send data. The schema of the electronic card that has been designed for control of system is shown in the Figure 3.4.1. Card has 48 MHZ clock frequency micro controller and connection ports to supply communication with other cards. At the same time, card has RS232 port which enables the communication between computer and the card. 3D drawing of the hardware illustrated in the Figure 3.4.2. In order to make the PIC 184550 model microcontroller which is on the hardware there is coded software with the help of ‘C’ programming language. The duty of the software is to send the pulse which is necessary to move the step motor and transform the voltage that has been read from LFR to get distance information. This transformation formula is given in the Equation 3.4.1. Later on software sends the converted information to computer as a pack.

$$dis\tan ce(mm) = \left[\left(\left(ADC\_val - ADC\_ofs\right) * Max\_dist / Max\_ADCv\right)\right] \qquad (3.4.1)$$

ADC_val: Reading voltage value from sensor

ADC_ofs: Minumum output value of sensor -> 172

Max_dist: Maximum distance value of sensor(mm) -> 9800

Max_ADCv: Maximum output value of sensor -> 569

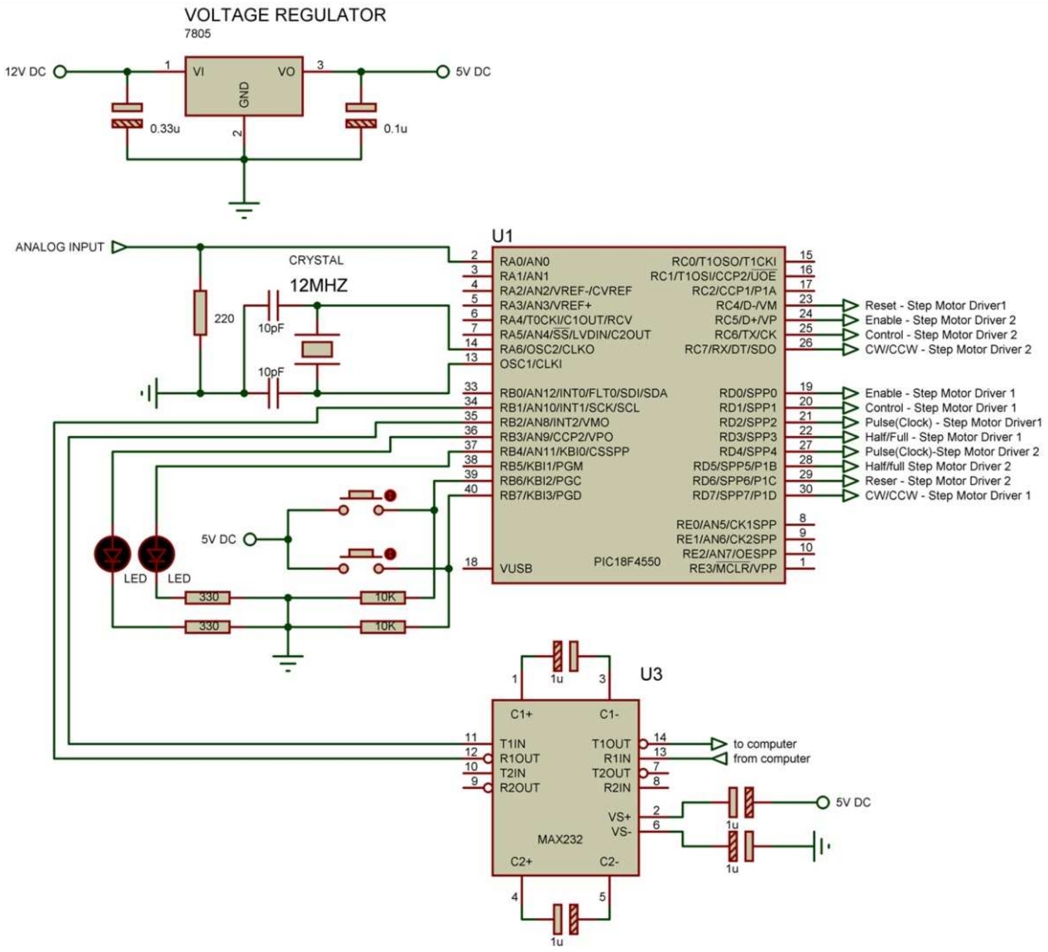


Figure3.4.1 The schema of the controller part of laser scanner hardware

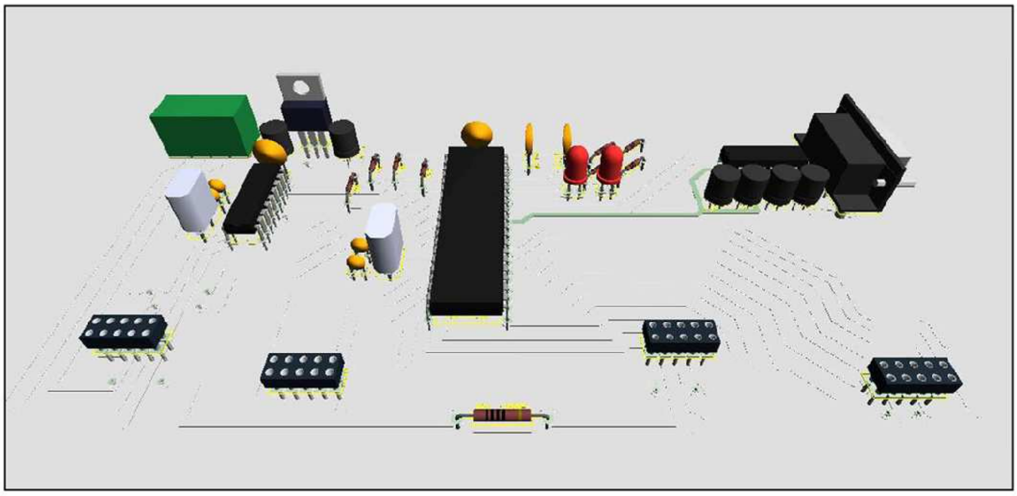


Figure 3.4.2 3D view of the laser scanner hardware

These scanning processes have been completed step by step in order. As soon as the system energized the axis has turned to starting position which is called as reference point. The other name of process is called homing. After the homing, the hardware waits for scanning area definition. Scanning area definition has defined by the user using by computer software, after the definition process, area setting data packet is sent to laser scanner hardware via communication port. End of the definition of scanning area, start command is waited by laser scanner hardware. When start command comes to laser scanner hardware, scanning process is started by the hardware. At the same time, each measured distance information and positions of turn able axis positions are sent to computer. When θ axis completes its one round, φ axis turns one step angle further with the 0.9 degree, when φ axis complete one turn, Ø axis start turning again step by step. This process carries on till the scanning area is scanned. Software flow chart is shown in Figure 3.4.3.

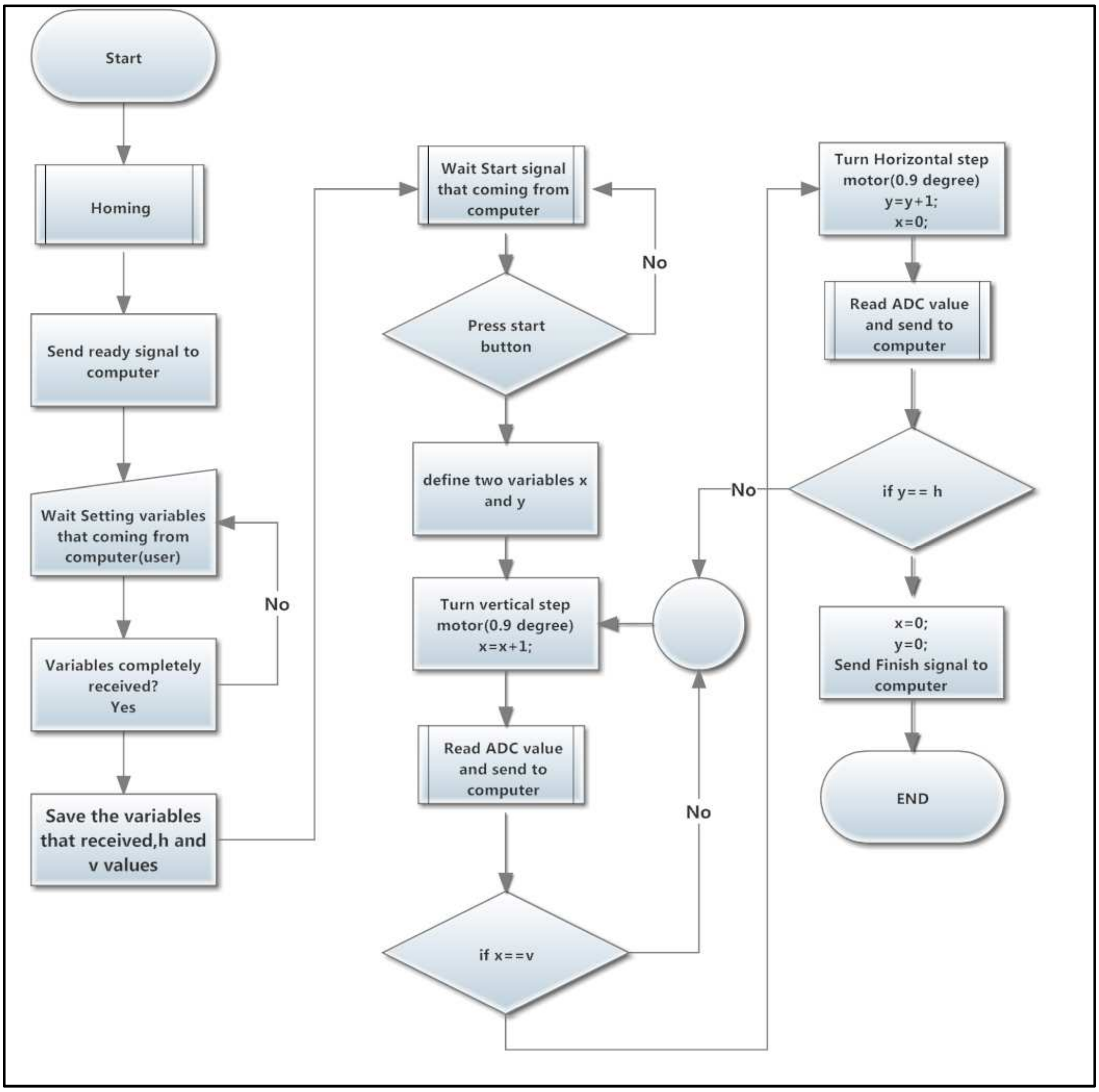

Start
Homing
Send ready signal to computer
Wait Setting variables that coming from computer(user)
Variables completely received?
Yes
No
Save the variables that received,h and v values
Wait Start signal that coming from computer
Press start button
No
define two variables x and y
Turn vertical step motor(0.9 degree)
x=x+1;
Read ADC value and send to computer
if x==v
No
Turn Horizontal step motor(0.9 degree)
y=y+1;
x=0;
Read ADC value and send to computer
if y== h
No
x=0;
y=0;
Send Finish signal to computer
END

Figure 3.4.3Flow Chart of executing system and operating scanning.

3.6 Homing Function of 2D Mechanical Part

Homing function is a process to definite starting position information of axis. Starting point position information is very important to generate 3D model. The model is generated using measured distance data. Computer program starts to create 3D model from first measures points. Since The software know to what degree do the motors rotate in every step, it might know in which angular coordinator is the every measurement, but this information is not enough. The software has to know the starting coordination of the system as well. Therefore in the system starting process, the axes locate the starting point. Two pieces of limit sensor and magnets are used to this process. These magnets have placed to the starting point of the axes. During homing process φ axis motor and θ axis motor start turning. When limit sensor detects the magnet, turning of motors is stopped by hardware. So φ axis and θ axis find the starting point. The starting points of the axes are shown in the Figure 3.5.1.

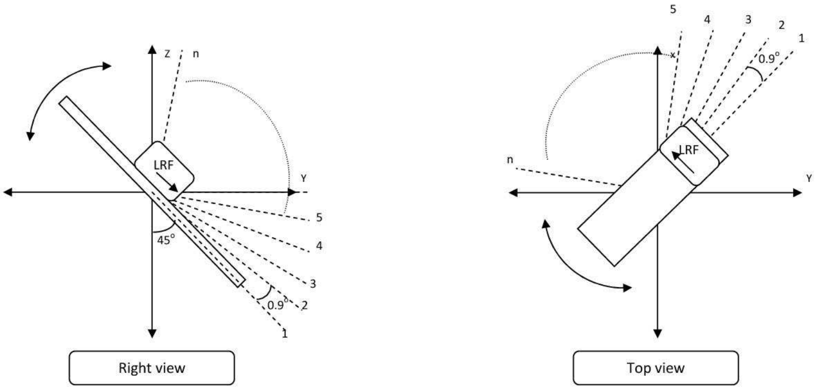


Figure 3.5.1 Starting points positions of the axes

3.7 Step Motor and Driver

θ and φ axes motors have to turn step by step in order to collect the distance information. Step motors are the motors that turn in a degree. The degrees of the motors when turned in each step are defined in the stage of motor production and the number of motor coil. Each coil group position is named as phase. Every phase means one step. The more the number of the coil is, the less is the step angle of motor. Step motors are appropriate for the position control. That is why 4 coiled 1.8 degree step motors are used in this study. Interior structure of the step motors are shown in Figure 3.7.1.

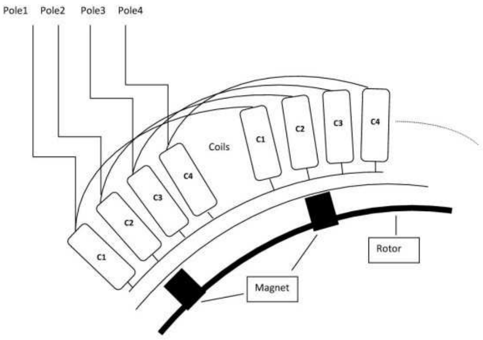


Figure 3.7.1 Stepping Motor

As soon as we energize the coils one by one motor starts to turn and the speed has a direct proportion with the frequency of energizing coils.

3.8 Step Motor Driver Design

A driver has been designed and implemented for controlling step motor. The duty of the driver is to help the motor turn step by step by giving energy to the related coils. An electronic card has designed and implemented for this study. The pin connections of the step motor driver are shown in Figure 3.7.2 and the view of 3D modeling of the step motor driver is shown in Figure 3.7.3. Step motor driver has same pins which held to control of direction. CW and CCW case of motor is controlled by step motor driver pins. We can control the motor using microcontroller with the help of driver. Driver circuit is shown in Figure 3.7.5. Step angle of the motor that we have used is 1.8 degree but we can decrease the degree to 0.9 by using the feature of half wave/ full wave on the driver. The less degree of step angle is the more resolution of the 3D modeling and more clear detail will be. That is why the used step angle is 0.9 degree.

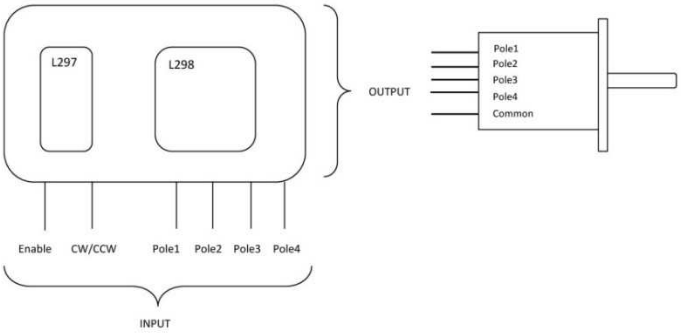


Figure 3.7.2 Pin connections of Step Motor Driver

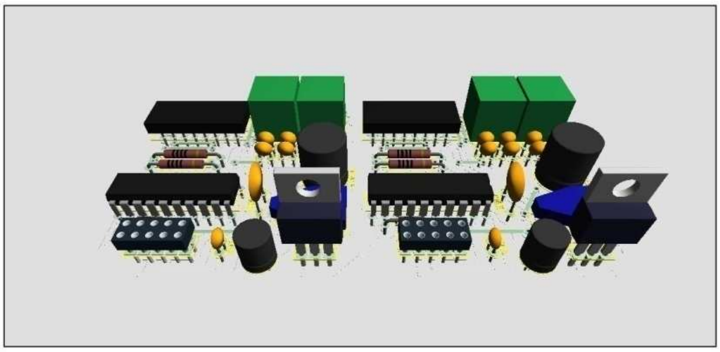


Figure 3.7.3 3D View of Step Motor Driver

Microcontroller has turned the motor by sending pulse to pins step by step. The number of the pulse determines the turning degree of the motors. The frequency of the pulse determines the turning speed of the motor. Pulse sequence to turning the motors is shown in Figure 3.7.4. The relation between motor angle and the number of pulse are given in Equation 3.7.1.

$$Turning\_angle\_of\_motor = (\#\,pulsex0.9\deg ree) \qquad (3.7.1)$$

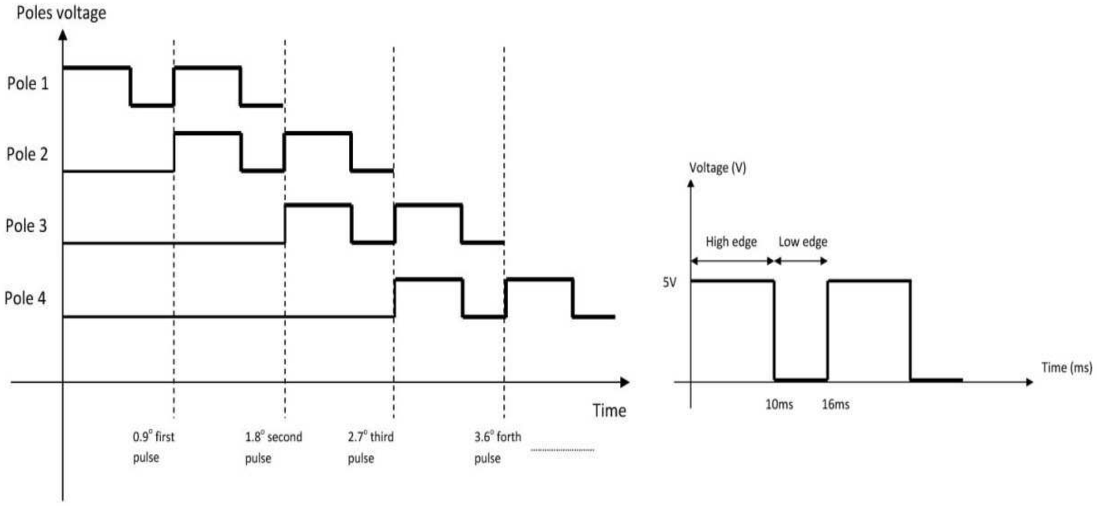


Figure 3.7.4 Pulse sequence to turning the motors

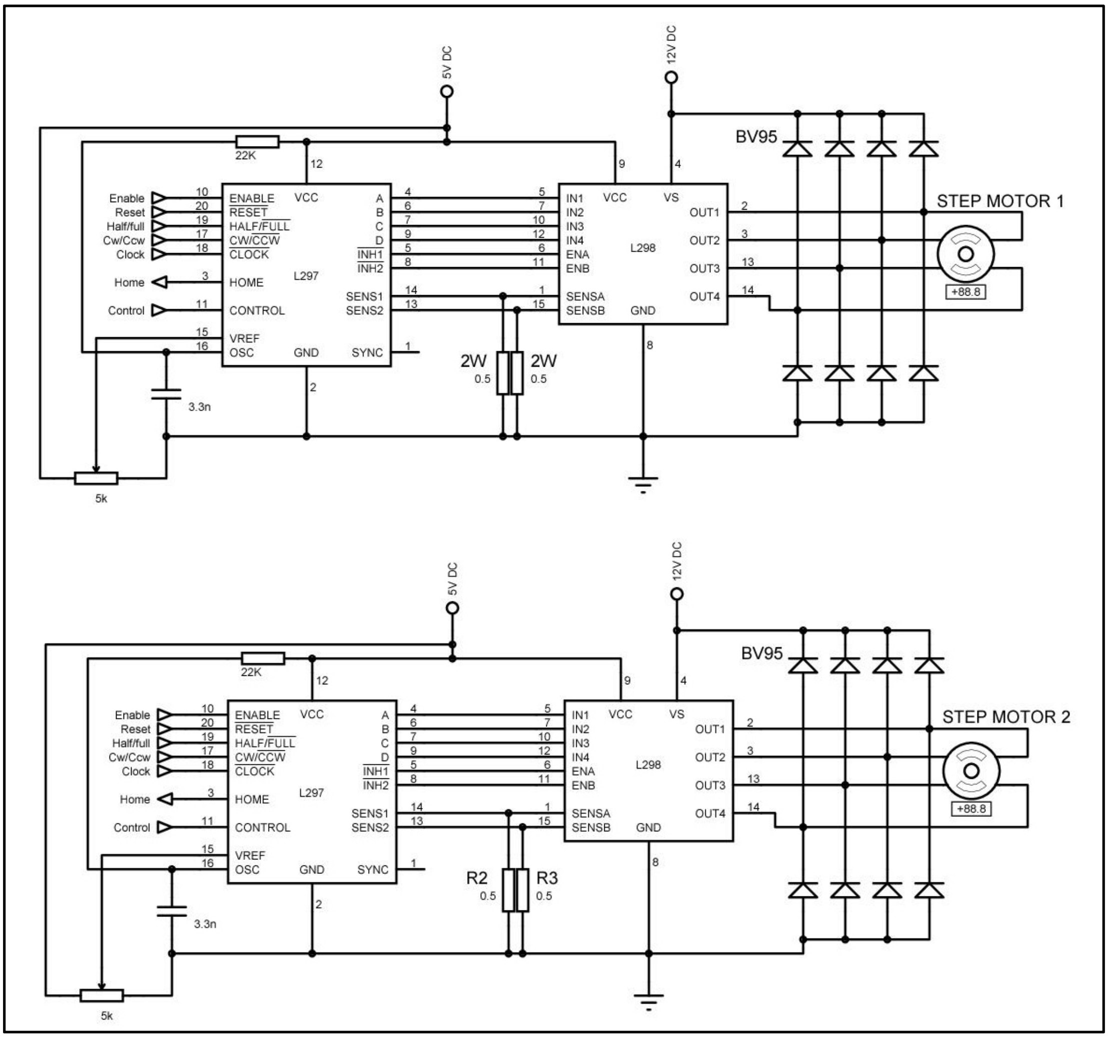


Figure 3.7.5 Step Motor Driver circuit

3.9 Communications between Computer and Laser Scanner Hardware

Communication between laser scanner hardware and computer is provided with hardware and computer program which is written in this study. Some ports like LPT, COM, USB, and SATA on the computer can be used so as to enable the communication. The port that is used in this project is COM-RS232. The communication protocol that has been used has duplex and serial communication. The communication speed of the port is 9600 bps. Since 1 byte is 8 bit, we can have the 1.2 kb communication speed with this port. The measurement speed is smaller than communication speed so 1.2kb communication speed is enough for our study.

3.10 Data Packets to Communication

The data packet transfer between computer and laser scanner hardware have been done by data packs. The hardware has sent a packet for each measurement. Every packet has included θ and φ axis location and distance data. The packet structure is shown in the Figure 3.9.1

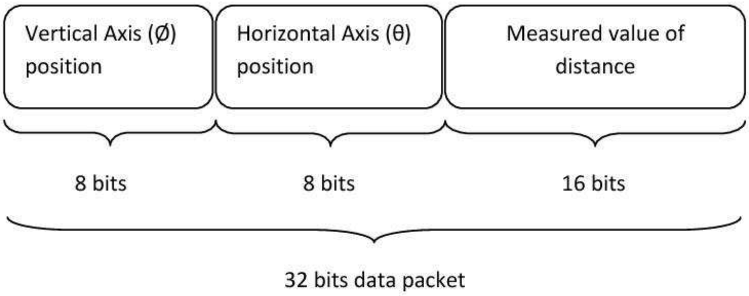
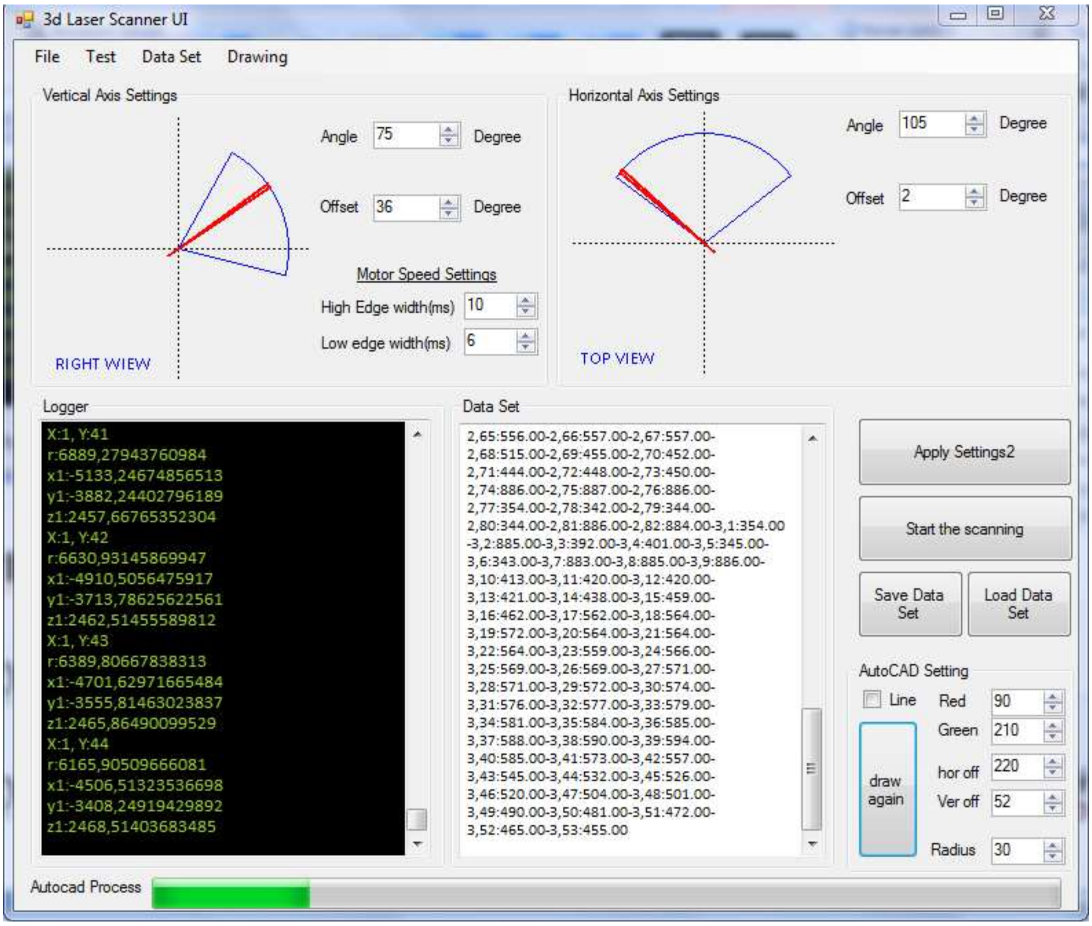

Vertical Axis (Ø) position
Horizontal Axis (θ) position
Measured value of distance
8 bits
8 bits
16 bits
32 bits data packet

Figure3.9.1 Data Packet of each measured point

3.11 Computer Software and Monitoring

The computer software has coded by using C# programming language on .NET platform. The duty of software is to collect and monitor the information that comes from the laser scanner hardware. Furthermore, the computer software has sent scanning parameters such as to scanner with the help of user interface. In addition to computer software has drawn the shape of the scanned environment with AutoCAD programme by using the data. Computer software is used to set scanning area parameter; speed of motors and color of AutoCAD platform .Software adjusts the color changes automatically according to distance change while drawing on the AutoCAD. The angle setting in the software determines the starting point and the scanning area of the scanner. The setting of high edge and low edge determines the structure of the pulse that is being sent to the motor. Software UI is shown in Figure 3.10.1



Vertical Axis Settings: Angle 75 Degree; Offset 36 Degree

Motor Speed Settings: High Edge width(ms) 10; Low edge width(ms) 6

Horizontal Axis Settings: Angle 105 Degree; Offset 2 Degree

AutoCAD Setting: □ Line; Red 90; Green 210; hor off 220; Ver off 52; Radius 30

Logger:

```
X:1, Y:41
r:6889,27943760984
x1:-5133,24674856513
y1:-3882,24402796189
z1:2457,66765352304
X:1, Y:42
r:6630,93145869947
x1:-4910,5056475917
y1:-3713,78625622561
z1:2462,51455589812
X:1, Y:43
r:6389,80667838313
x1:-4701,62971665484
y1:-3555,81463023837
z1:2465,86490099529
X:1, Y:44
r:6165,90509666081
x1:-4506,51323536698
y1:-3408,24919429892
z1:2468,51403683485
```

Data Set:

```
2,65:556.00-2,66:557.00-2,67:557.00-
2,68:515.00-2,69:455.00-2,70:452.00-
2,71:444.00-2,72:448.00-2,73:450.00-
2,74:886.00-2,75:887.00-2,76:886.00-
2,77:354.00-2,78:342.00-2,79:344.00-
2,80:344.00-2,81:886.00-2,82:884.00-3,1:354.00
-3,2:885.00-3,3:392.00-3,4:401.00-3,5:345.00-
3,6:343.00-3,7:883.00-3,8:885.00-3,9:886.00-
3,10:413.00-3,11:420.00-3,12:420.00-
3,13:421.00-3,14:438.00-3,15:459.00-
3,16:462.00-3,17:562.00-3,18:564.00-
3,19:572.00-3,20:564.00-3,21:564.00-
3,22:564.00-3,23:559.00-3,24:566.00-
3,25:569.00-3,26:569.00-3,27:571.00-
3,28:571.00-3,29:572.00-3,30:574.00-
3,31:576.00-3,32:577.00-3,33:579.00-
3,34:581.00-3,35:584.00-3,36:585.00-
3,37:588.00-3,38:590.00-3,39:594.00-
3,40:585.00-3,41:573.00-3,42:557.00-
3,43:545.00-3,44:532.00-3,45:526.00-
3,46:520.00-3,47:504.00-3,48:501.00-
3,49:490.00-3,50:481.00-3,51:472.00-
3,52:465.00-3,53:455.00
```

Figure 3.10.1 Computer program UI

The software has generated a matrix which of size related to number of measured points. After scanner has completed the scanning process, the computer software has generated 3D model by using matrix on AutoCAD. The matrix consists of distance and angle information. Measured values have converted from polar coordinating system to cartesian coordinating system. This conversion and drawing have completed by computer software. Sample matrix structure is shown in Figures 3.10.1 and 3.10.2

$$A = \begin{bmatrix} 373.00 & 374.00 & 372.00 & 737.00 & 717.00 & . & . & v \\ 372.00 & 374.00 & 381.00 & 386.00 & 395.00 & . & . & . \\ 387.00 & 385.00 & 390.00 & 730.00 & 738.00 & . & . & . \\ 377.00 & 380.00 & 386.00 & 387.00 & 395.00 & . & . & . \\ 398.00 & 399.00 & 401.00 & 732.00 & 758.00 & . & . & . \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ h & . & . & . & . & . & . & vxh \end{bmatrix} \tag{3.10.1}$$

$$v = \frac{\phi}{Step\_angle}, h = \frac{\theta}{Step\_angle} \tag{3.10.2}$$

$Step\_angle = 0.9^o$

ϕ: Vertical angle of scanning area that defined by user.

θ: Horizontal angle of scanning area that defined by user.

3.12 Parallel running between C# and AutoCAD

Autodesk firm is the manufacturer firm of the AutoCAD program. Autodesk firm designed API which enables applications to be connected without having a user connection with program called AutoCAD. Firm published great tutorial and sample codes related to usage of these API on their websites. We have issed the AutoCAD program by having a relation between AutoCAD and C# with the help of the API that Autodesk firm enables. The software can generate the object modeling automatically with the help of the data that we received from scanner.

CHAPTER 4

EXPERIMENTAL STUDY

4.1 Resolution

The resolution of the image generated by designed system depends on the step angle of the step motor. When the distance of measured point has increased the resolution of model will decrease. To create a model with high resolution, the step angle of the motors should be decreased. For instance, while the system scans an object with 10 meter distance, it takes a point measurement in every 15 cm. It will take more frequent measurement when the distance decreases. The calculation related to resolution is shown in the equation 4.1.1. Points distribution graph is shown in figure 4.1.1.

$$Range\_between\_pnts = \frac{2x\Pi xLx\phi}{360} \tag{4.1.1}$$

L: Distance of measured point

ϕ: Step motor angle value that changing each step

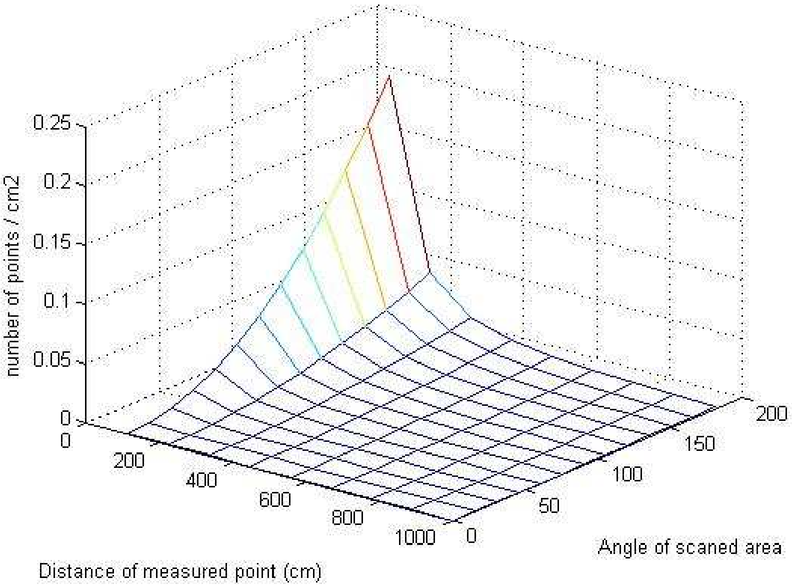


Figure 4.1.1 Measured points distribution graph

4.2 Scanning Time

Scanning time is the time that starts from the beginning of the scanning and the end of scanning. The time depends of the width of the scanning area and the speed of the motors. When the speed of the motors is increased in order to decrease the scanning time, it is clearly observed that the measured value correctness is not stabled. Scanning time relation equation is given in Equation 4.2.1. Scanning time, vertical and horizontal angle graph shown in Figure 4.2.1.

$$ST = \eta x \frac{\phi}{\varphi} x \frac{\theta}{\varphi} x \rho \qquad (4.2.1)$$

ST: Scanning Time.

η: Step time value.

ϕ: Vertical angle of scanning area.

θ: Horizontal angle of scanning area.

φ: Step time value.

ρ: Duration time of conversion signal from analog to digital.

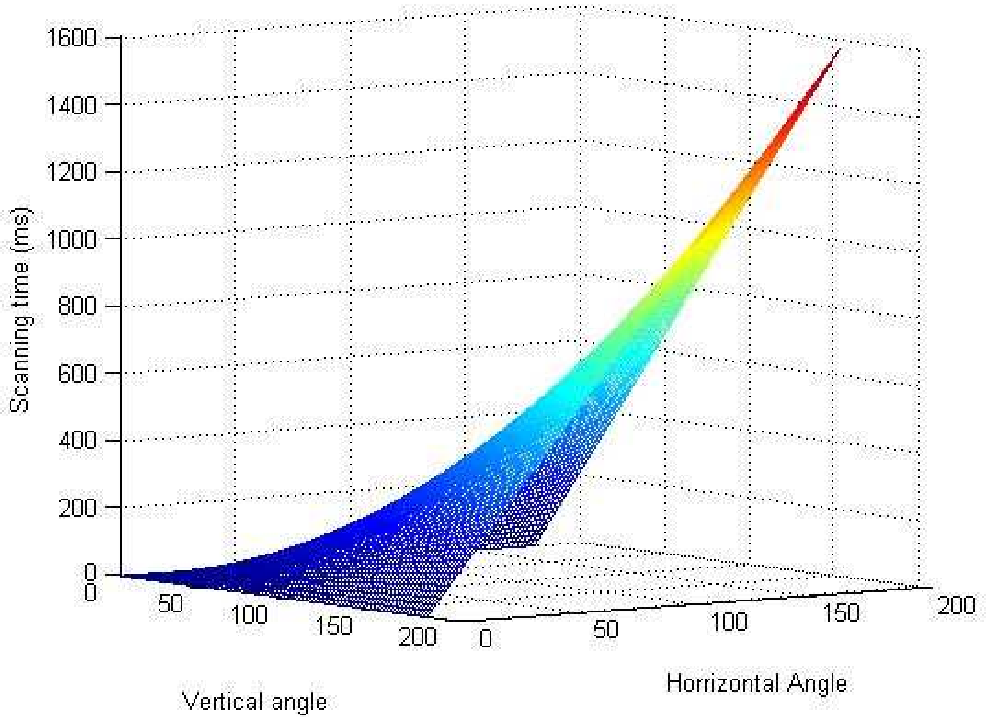


Figure 4.2.1Scanning time graph

4.3 Results

Designed and implemented hardware and software have been evaluated at different environments. Scanning results have shown some pictures and model views that scanned from different places. Each result consists of photography of place, views of created model and information about scanning time, number of measured point and scanning area.

Figure 4.3.1 shows scan results have been obtained in a outdoor of a tree. Some scan specifications are given in Table 4.3.1.

Table 4.3.1 Scanning parameters of tree

Vertical angle	65 ^{o}
Horizontal angle	65 ^{o}
Scanning time	182 seconds
Measured point number	4520 points
Used point number to create model	1551 points

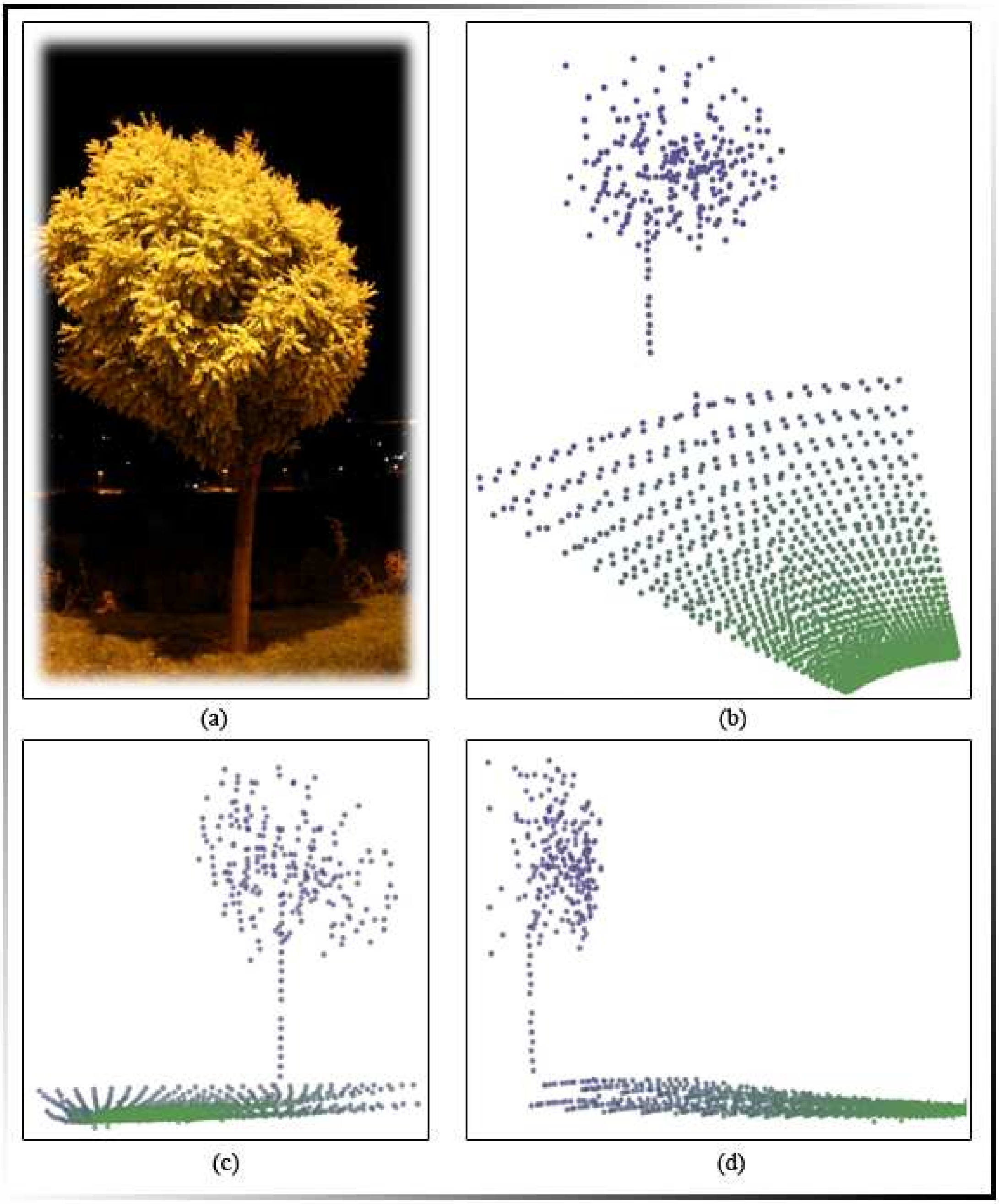


Figure 4.3.1 Scanning result of tree from different views point, (a) Photography of tree, (b) Perspective view of created model, (c) opposite view of created model, (d) Left view of created model.

Figure 4.3.2 shows scan results have been obtained in a closed area of a hall and plant. Some scan specifications are been given in Table 4.3.2.

Table 4.3.2 Scanning parameter of hall

Vertical angle	100 ^{o}
Horizontal angle	110 ^{o}
Scanning time	541 seconds
Measured point number	13542 points
Used point number to create model	8874 points

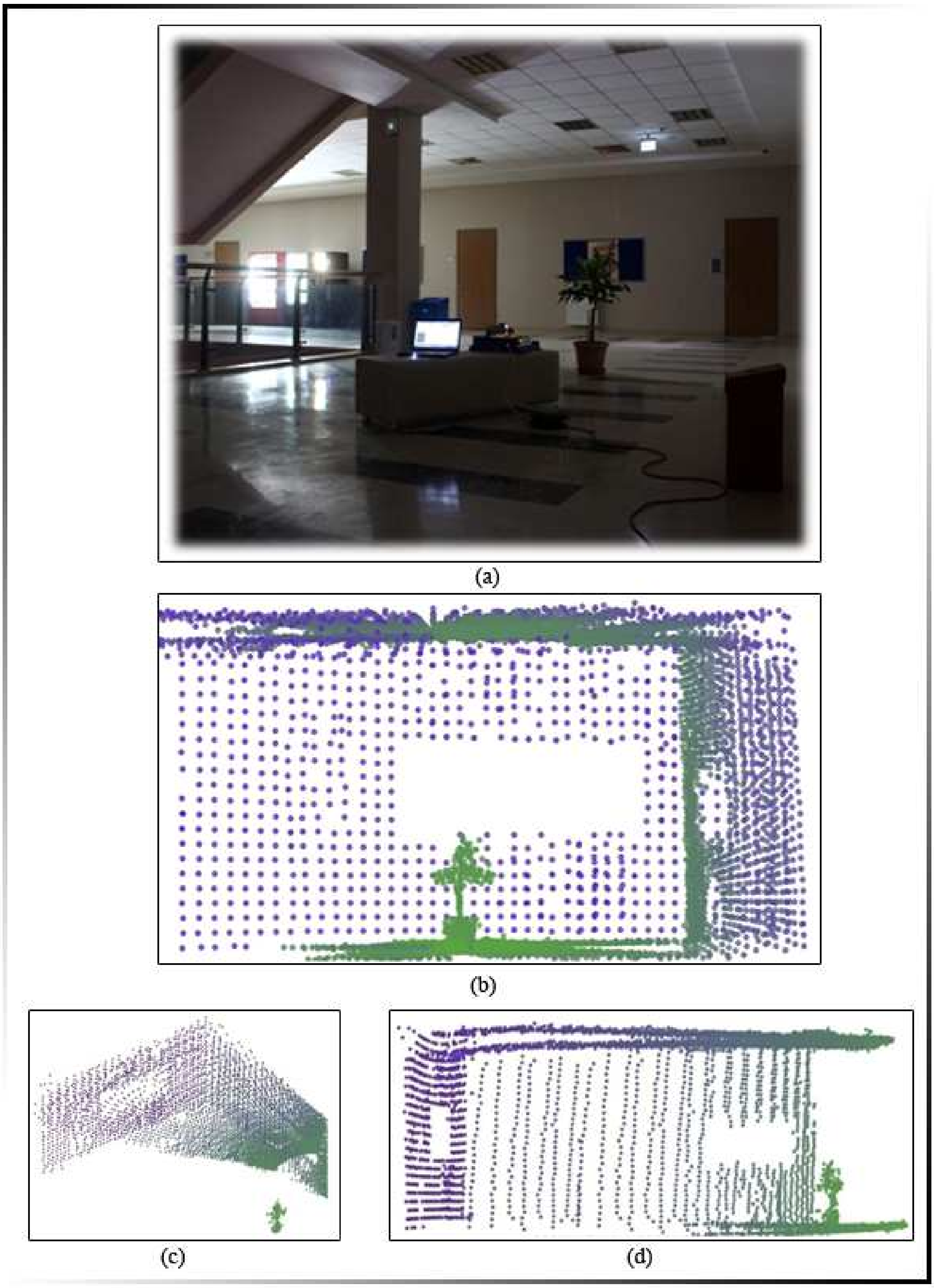


Figure 4.3.2 Scanning result of hall from different views point, (a) Photography of hall, (b) opposite view of created model, (c) perspective view of created model, (d) Left view of created model.

Figure 4.3.3 shows scan results have been obtained in a closed area of a wall and a window. Some scan specifications are given in Table 4.3.3.

Table 4.3.3 Scanning parameters of Wall and window

Vertical angle	100 ^{o}
Horizontal angle	55 ^{o}
Scanning time	541 seconds
Measured point number	6771 points
Used point number to create model	5574 points

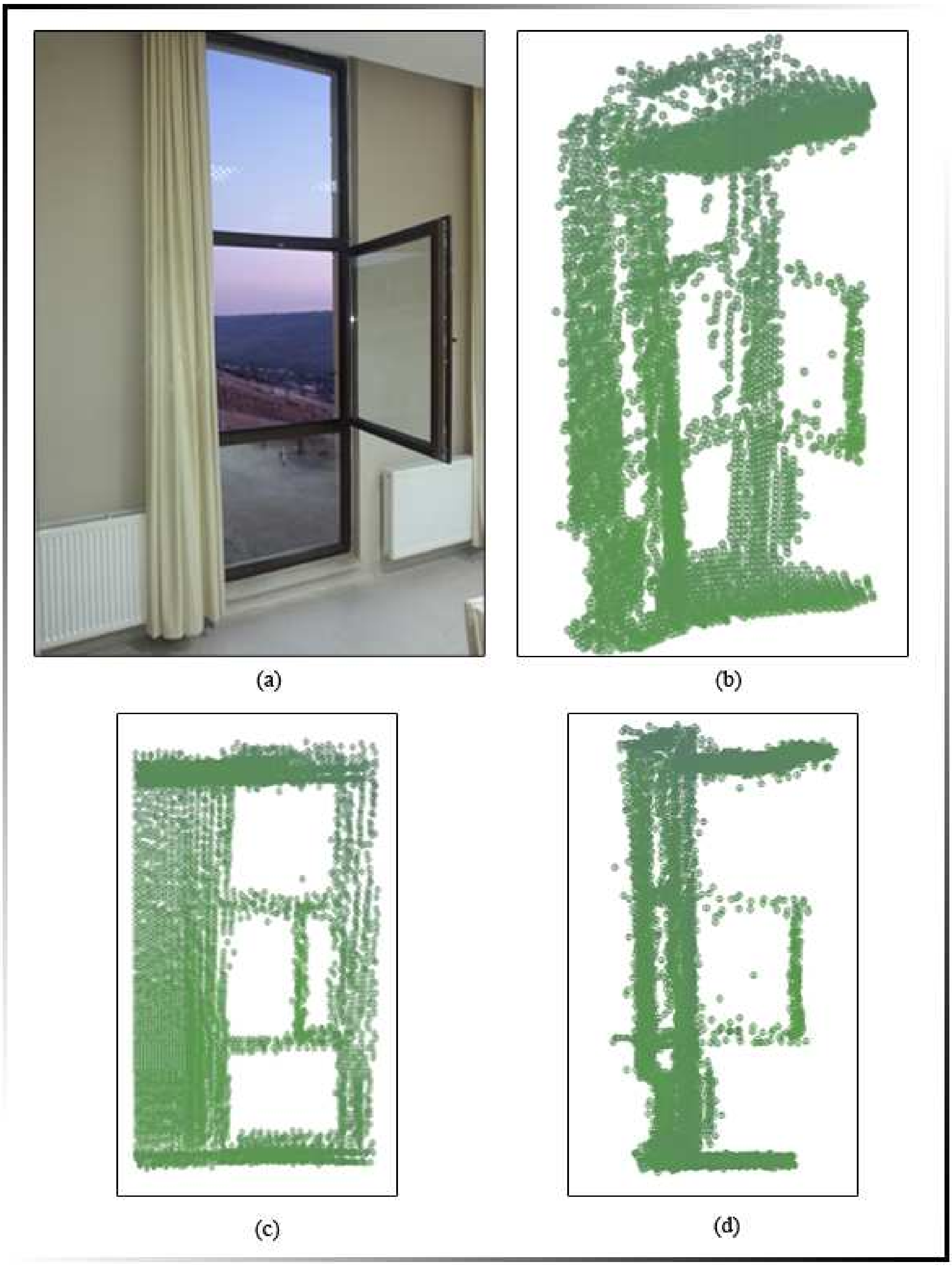


Figure 4.3.3 Scanning result of wall and window from different views point, (a) Photography of wall and window, (b) Perspective view of created model, (c) opposite view of created model, (d) Left view of created model.

CHAPTER 5

CONCLUSION

In this study, the scanner has created a 3D model of the environment. The scanner consists of 2 electronic cards, 1D LRF sensor and computer software. All units are operating in real-time. The distance measured by the scanner and the provided model give us information about the density of objects. The scanning time depends on the engine speed and size of scanning area. According to the equivalent laser scanners our scanner is slower but cheaper. To reduce the costs, 1D LRF has been used. Produced laser scanner cost is lower than equivalent laser scanner. But, Scanning speed of produced laser scanner is lower than equivalent laser scanner. Laser scanner can used at architecture restoration projects and detecting path of robotic application.

During the design of laser scanner and the 3D modeling implementation, I have gained experience about Stepper Motor Driver and real-time data streaming, AutoCAD and C# interaction.

5.1 Future Work

This device’s motion can be provided by moving it on the robotic system. Moreover, system can be improved by using servo motors, different LRF and fast microprocessors. When scanning speed of laser scanner increase, it is used many area as short range radar. For instance It can define the shape of the bird or a plane. Laser scanners are used in our houses with the smart televisions(Riza, Nabeel A., 2011). Laser scanners can be used to generate any geographical place (Zho, H, 2001). The increased addiction of technological development and robotic systems show that usage of laser scanners will increase.

REFERENCES

Biber, P., Andreasson, H., Duckett, T., Schilling, A, (2004). 3D modeling of indoor environments by a mobile robot with a laser scanner and panoramic camera. *Intelligent Robots and System.* **4**, 3430 – 3435.

Huber, D., Akinci, B., Pingbo Tang Adan, A., Okorn, B., Xuehan Xiong, (2010). Using Laser Scanners for Modeling and Analysis in Architecture, Engineering, and Construction. *Information Sciences and Systems (CISS), 2010 44th Annual Conference on.* 1-6.

Jianhua W., Bing Li, Weihai Chen, Lixia Rong, (2008). 3D reconstruction embedded system based on laser scanner for mobile robot. *Industrial Electronics and Applications.* **20**, 697-701.

Layton, T., Jacobs, E.L., Griffin, S.T., (2009). 3D laser scanner and profiler for application to border security. *Sensors Applications Symposium.* 124-127

Luo, R.C., Tzou, J.H., Yi Cheng Chang, (2000). The integration of 3D digitizing and LCD panel display based rapid prototyping system for manufacturing automation. *Industrial Electronics Societ.* **2**, 1255 – 1260.

Nemoto, Z., Takemura, H., Mizoguchi, H., (2007). Development of Small-sized Omni-directional Laser Range Scanner and Its Application to 3D Background Difference. *33rd Annual Conference of the IEEE*, 2284 – 2289.

Ohno, K., Tadokoro, S., Kawahara, Toyokazu, (2008). Development of 3D laser scanner for measuring uniform and dense 3D shapes of static objects in dynamic environment. *Robotics and Biomimetics* **2**, 2161–2167.

Riza, Nabeel A., Marraccini, P.J., (2011). Smart Two-Dimensional Laser-Based Display. *Display Technology, Journal of (Volume:7 , Issue: 2)* , **7,** 90 – 95.

Xiaofeng Lian, Zaiwen Liu, Xiaoyi Wang, Lihua Dou (2008). Reconstructing indoor environmental 3D model using laser range scanners and omnidirectional camera, (2008), 1640 – 1644.

Zho, H., Shibasaki, R. (2001). Reconstructing urban 3D model using vehicle-borne laser range scanners. *3-D Digital Imaging and Modeling,* 349 – 356.

APPENDIX

C CODE FOR MICROCONTROLLER

```
#include <18F4550.h>
#device adc=10
#fuses
HSPLL,WDT,PROTECT,LVP,NODEBUG,PLL3,CPUDIV1,NOVREGEN,USBDIV

#use delay(clock=48000000)
#use rs232(baud=9600,parity=N,xmit=PIN_B2,rcv=PIN_B1,bits=8)

int start=0;
int ver_degP=10; // pulse
int hor_degP=10;
int h=1;
int v=1;
int fW=15; // ms full width
int hE=10;// ms high edge
int upLoc=0;
int hor_offset=2;// pulse,
int ver_offset=12;// pulse
float adc=0;

char rcv;
int wait=0;



void start_up()
{
```

```
output_high(PIN_B3);
delay_ms(350);
output_high(PIN_B4);
delay_ms(350);
output_low(PIN_B3);
delay_ms(350);
output_low(PIN_B4);
delay_ms(350);
output_high(PIN_B3);
delay_ms(350);
output_high(PIN_B4);
delay_ms(350);
output_low(PIN_B3);
delay_ms(350);
output_low(PIN_B4);
delay_ms(350);

output_low(PIN_E0); // Not used
output_low(PIN_A1); // Not used
output_low(PIN_A2); // Not used
output_low(PIN_A3); // Not used
output_low(PIN_A4); // Not used
output_low(PIN_A5); // Not used

    output_low(PIN_D0); // step 1 is enable
    output_low(PIN_C4);  // reset is not active
    output_low(PIN_D3);  // half wave active
    output_low(PIN_D1); // step 1 is control pin
    output_low(PIN_C5); // step 2 is enable
    output_low(PIN_D6);  // reset is not active
    output_low(PIN_D5);  // half wave active
    output_low(PIN_C6); // step 2 is control pin
    output_low(PIN_D2); // step 1 clock
```

```
      output_low(PIN_C7); // step 2 Cw
printf("ready");
putc('R');
}
void stop_stepper(){
      output_low(PIN_D0); // step 1 is enable
      output_low(PIN_D1); // step 1 is control pin
      output_low(PIN_C5); // step 2 is enable
      output_low(PIN_C6); // step 2 is control pin
}
void start_stepper(int choose)// Enable, reset, half/full, control || choose == 1 => step
1 is active,==2 => step 2 is active, ==3 => twice step motor are active
{
 if(choose==1) // horizantoal step motor is preparing
 {
      output_high(PIN_D0); // step 1 is enable
      output_high(PIN_C4);  // reset is not active
      output_high(PIN_D3);  //  if HIGH = half wave active , else if LOW = Full wave
active
      output_high(PIN_D1); // step 1 is control pin
}
 else if(choose==2) // vertical step motor is preparing
 {
      output_high(PIN_C5); // step 2 is enable
      output_high(PIN_D6);  // reset is not active
      output_high(PIN_D5);  //  if HIGH = half wave active , else if LOW = Full wave
      output_high(PIN_C6); // step 2 is control pin
 }
 else if(choose==3)// all step motors are preparing
   {
      output_high(PIN_D0); // step 1 is enable
      output_high(PIN_C4);  // reset is not active
      output_high(PIN_D3);  // if HIGH = half wave active , else if LOW = Full wave
```

```
     output_high(PIN_D1); // step 1 is control pin
     output_high(PIN_C5); // step 2 is enable
     output_high(PIN_D6);  // reset is not active
     output_high(PIN_D5);  //if HIGH = half wave active , else if LOW = Full wave
     output_high(PIN_C6); // step 2 is control pin
  }
  else if(choose==4)// close all
  {
     output_low(PIN_D0); // step 1 is enable
     output_low(PIN_C4);  // reset is not active
     output_low(PIN_D3);// if HIGH = half wave active , else if LOW = Full wave
     output_low(PIN_D1); // step 1 is control pin
     output_low(PIN_C5); // step 2 is enable
     output_low(PIN_D6);  // reset is not active
     output_low(PIN_D5);  // if HIGH = half wave active , else if LOW = Full wave
     output_low(PIN_C6); // step 2 is control pin
  }
}
void homming()
{
int x=1;
int catched=2; // when the sensor catched it's value change to 3 and it will start
return.
printf("Homming started");
putc('R');
  output_high(PIN_D7); // step 1 Cw
  output_high(PIN_B3); // led red
    while(1){
        output_high(PIN_D2); //step 1 clock
        if(!input(PIN_E1)) // sensor catched
        {
           catched=4;
           break;
```

```
      }
      delay_ms(fW);
      output_low(PIN_D2);
      delay_ms(fW-hE);
    }
    if(catched==4)
    {
            delay_ms(20);
            output_low(PIN_D7);
            for(x=1;x<=ver_offset;++x)
            {
              output_high(PIN_D2); //step 1 clock
              delay_ms(fW);
              output_low(PIN_D2);
              delay_ms(fW-hE);
            }
      upLoc=0;
      catched=1;
    }
output_low(PIN_C7); // step 2 CCw
while(1){
    output_high(PIN_D4); //step 2 clock
    if(!input(PIN_E2)) // sensor catched
      {
             catched=3;
        break;
      }
    delay_ms(fW);
    output_low(PIN_D4);
    delay_ms(fW-hE);
}
if(catched==3)
{
  catched=1;
output_high(PIN_C7); // step 2 Cw
  for(x=1;x<=hor_offset;++x)
  {
    output_high(PIN_D4); //step 2 clock
    delay_ms(fW);
    output_low(PIN_D4);
    delay_ms(fW-hE);
  }
}
output_low(PIN_B3);
printf("at home");
putc('R');
}
void main()
{
  setup_adc_ports(AN0|VSS_VDD);
  setup_adc(ADC_CLOCK_Div_64);
  setup_psp(PSP_DISABLED);
  setup_spi(FALSE);
  setup_wdt(WDT_OFF);
  setup_timer_0(RTCC_INTERNAL);
  setup_timer_1(T1_DISABLED);
  setup_timer_2(T2_DISABLED,0,1);
  setup_timer_3(T3_DISABLED|T3_DIV_BY_1);
  setup_comparator(NC_NC_NC_NC);
  setup_vref(FALSE);
  enable_interrupts(INT_TIMER0);
  enable_interrupts(GLOBAL);
  setup_low_volt_detect(FALSE);
  setup_oscillator(False);
  start_up();
delay_ms(500);
```

```
start_stepper(3);
delay_ms(100);
homming();
delay_ms(200);
  while(1){
        // get character
          output_high(PIN_B3); // led red
          output_high(PIN_B4); // yellow red
    rcv=getc();
    if(rcv==0x56)// 0x56= V vertical angle
    {
        output_low(PIN_B4);
        ver_degP=getc();
        delay_ms(200);
        output_high(PIN_B4);
        printf("vertical Angle OK");
        putc('R');
        continue;
    }
    if(rcv==0x57) // horizantal angle
    {
        output_low(PIN_B4);
        hor_degP=getc();
        delay_ms(200);
        output_high(PIN_B4);
        printf("Horizantal Angle OK");
        putc('R');
        continue;
    }
    if(rcv==0x58) //vertical offset
    {
            output_low(PIN_B4);
        ver_offset=getc();
```

```
     delay_ms(200);
     output_high(PIN_B4);
     printf("Veretical offset OK");
     putc('R');
     continue;
}
if(rcv==0x59)// horizantal offset
{
     output_low(PIN_B4);
     hor_offset=getc();
     delay_ms(200);
     output_high(PIN_B4);
     printf("Horizantal offset OK");
     putc('R');
     output_low(PIN_B3); // led red
     output_low(PIN_B4); // yellow red
     continue;
}
if(rcv==0x60) // vertical Highedge width
{
     output_low(PIN_B4);
     fW=getc();  // full width
     hE=getc();  // high Edge
     delay_ms(200);
     output_high(PIN_B4);
     printf("vertical and horizantal motor speeds Ok");
     putc('R');
     printf("Parameters Setted ...");
     putc('R');
     printf("Vertical angle: %d",ver_degP);
     putc('R');
     printf("Horizonral angle: %d",hor_degP);
     putc('R');
```

```
            printf("Vertical angle offset: %d",ver_offset);
            putc('R');
            printf("Horizontal angle offset: %d",hor_offset);
            putc('R');
            printf("Full width: %d",fW);
            putc('R');
            printf("High Edge width: %d",hE);
            putc('R');
            output_low(PIN_B3); // led red
            output_low(PIN_B4); // yellow red
            break;
        }
      }
putc('Z'); // start button enable
  while(1)
  {
a:
      if(input(PIN_b6))
        start=1;
      if(input(PIN_b6) || getc()==0x35)
      {
            set_adc_channel(0);// A0 adc starting
            adc=read_adc();
            printf("First dist=%f,"adc); // to filter "-"
            output_low(PIN_B3); // led red
            output_high(PIN_B4); // yellow red
            start_stepper(3);
            homming();
            delay_ms(150);
                for(h=1;h<hor_degP;++h)
                {
                  output_high(PIN_D4); //step 2 clock
                  delay_ms(fW);
```

```
    output_low(PIN_D4);
    delay_ms(fW-hE);
    delay_ms(15);
      if(upLoc==0){
        output_low(PIN_D7);}
      else if(upLoc==1){
        output_high(PIN_D7);}
      for(v=1;v<ver_degP;++v)
      {
        output_high(PIN_D2); //step 1 clock
          adc=read_adc();
          if(v==1 && h==1)
            printf("%d,%d:%f,"h,v,adc); // to filter "-"
          else
            printf("-%d,%d:%f"h,v,adc);
           if(input(PIN_b7))
             goto a;
        delay_ms(fW);
        output_low(PIN_D2);
        delay_ms(fW-hE);



      }
      if(upLoc==1){
        upLoc=0;}
      else if(upLoc==0){
        upLoc=1;}
  }
  printf("Scan Process Completed");
  putc('W');
  start=2;
output_low(PIN_B4); // yellow red
```

```
            output_high(PIN_B3); // led red
            stop_stepper();
   }
   }
}
```

C SHARP CODE FOR COMPUTER

Coordinate system Transformation spherical to Cartesian

```
        private void autocad_tx()
        {

            int x, y;
            double r;
            double p3;
            int cnt=0;
            double ox, oy, oz;
            ox = 0;
            oy = 0;
            oz = 0;
            for (x = 1; x <= h1; x++)// hor
            {

                for (y = 1; y <= v1; y++)// vertical
                {
                    cnt++;
                    r =  data_packet[x, y]; //   220 ohm 4-20 mA
                    p3 = ((r-172) * 9800 / 569); // 9800 mm so 9.8 meter full 
range

                    if (x % 2 == 0)
                    {
                        z1 = p3 * Math.Sin((((v1 - y) * step_angle) - 
(int)vo.Value + (double)verOffset.Value) * (Math.PI / 180.0));
                        x1 = p3 * Math.Cos((((v1 - y) * step_angle) - 
(int)vo.Value + (double)verOffset.Value) * (Math.PI / 180.0)) * 
Math.Cos(((int)ho.Value - (double)horOffset.Value - (x * step_angle)) * 
(Math.PI / 180.0));
                        y1 = p3 * Math.Cos((((v1 - y) * step_angle) - 
(int)vo.Value + (double)verOffset.Value) * (Math.PI / 180.0)) * 
Math.Sin(((int)ho.Value - (double)horOffset.Value - (x * step_angle)) * 
(Math.PI / 180.0));

                        textBox1.AppendText("\r\nX:" + x.ToString() + ", Y:" + 
y.ToString() + "\r\nr:"+p3.ToString() + "\r\nx1:" + x1.ToString() + "\r\ny1:" 
+ y1.ToString() + "\r\nz1:" + z1.ToString() + "\r\nox:" + ox.ToString() + 
"\r\noy:" + oy.ToString() + "\r\noz:" + oz.ToString());
                        addSpr(x1, y1, z1, (int)rsphr.Value, ox, -oy, oz,r);
                        ox = x1;
                        oy = y1;
                        oz = z1;
                    }
                    else
                    {
                        z1 = p3 * Math.Sin(((-(int)vo.Value + 
(double)verOffset.Value) + (y * step_angle)) * (Math.PI / 180.0));
                        x1 = p3 * Math.Cos(((-(int)vo.Value + 
(double)verOffset.Value) + (y * step_angle)) * (Math.PI / 180.0)) * 
Math.Cos(((int)ho.Value - (double)horOffset.Value - (x * step_angle)) * 
(Math.PI / 180.0))
```

```
                        y1 = p3 * Math.Cos(((-(int)vo.Value + 
(double)verOffset.Value) + (y * step_angle)) * (Math.PI / 180.0)) * 
Math.Sin(((int)ho.Value - (double)horOffset.Value - (x * step_angle)) * 
(Math.PI / 180.0));
                        textBox1.AppendText("\r\nX:" + x.ToString() + ", Y:" + 
y.ToString() + "\r\nr:"+p3.ToString()+"\r\nx1:" + x1.ToString() + "\r\ny1:" + 
y1.ToString() + "\r\nz1:" + z1.ToString());
                        addSpr(x1, y1, z1, (int)rsphr.Value, ox, -oy, oz,r);
                        ox = x1;
                        oy = y1;
                        oz = z1;

                    }

                }



            }


        }
```

Monitoring the position of Axis

```
private void draw_shapes(int i
        {

            Point orj = new Point(95, 95);// x  hor y ver

            System.Drawing.Pen myPen;
            System.Drawing.Pen myPen2;
            System.Drawing.Pen mypen_hash;
            System.Drawing.Pen mypen_blue;
            float[] dashValues = { 2, 2 };
            mypen_hash = new System.Drawing.Pen(System.Drawing.Color.Black);
            mypen_blue = new System.Drawing.Pen(System.Drawing.Color.Blue);
            mypen_hash.DashPattern = dashValues;

            myPen2 = new System.Drawing.Pen(System.Drawing.Color.Yellow);
            myPen = new System.Drawing.Pen(System.Drawing.Color.Red, 2);
            Font myFont = new Font("Calibri", 10);

            if (i == 1)
            {
                verpan.Refresh();
                System.Drawing.Graphics formGraphics = 
this.verpan.CreateGraphics();
                formGraphics.DrawLine(mypen_hash, 0, 95, 190, 95);  // horiz 
axis  // hor,ver,hor,ver
                formGraphics.DrawLine(mypen_hash, 95, 0, 95, 190); // vertical 
axis


                formGraphics.DrawPie(mypen_blue, new Rectangle(15, 15, 160, 
160), (50 - (float)verOffset.Value) , -(float)verAngle.Value);
                formGraphics.DrawPie(myPen, new Rectangle(15, 15, 160, 160), 
50 - (float)verOffset.Value - turning_value_ver, -2); // lazer show
```

```
                formGraphics.DrawString("RIGHT WIEW", myFont, Brushes.Blue,
new PointF(4, 170));


                formGraphics.Dispose();
            }
            else if (i == 2)
            {
                horpan.Refresh();
                System.Drawing.Graphics formGraphics2 =
this.horpan.CreateGraphics();
                formGraphics2.DrawLine(mypen_hash, 0, 95, 190, 95);  // horiz
axis  // hor,ver,hor,ver
                formGraphics2.DrawLine(mypen_hash, 95, 0, 95, 190); //
vertical axis

                formGraphics2.DrawPie(mypen_blue, new Rectangle(15, 15, 160,
160), 215 + (float)horOffset.Value, (float)horAngle.Value);
                formGraphics2.DrawPie(myPen, new Rectangle(15, 15, 160, 160),
215 + (float)horOffset.Value + turning_value_hor, 2); // lazer show
                formGraphics2.DrawString("TOP VIEW",myFont,Brushes.Blue,new
PointF(4,170));


                formGraphics2.Dispose();
            }

            myPen.Dispose();
            mypen_blue.Dispose();
            mypen_hash.Dispose();
            myPen2.Dispose();

        }
```

This Function Sends Setting Values to Electronic Board

```
     private void set()
        {
            // Send the one character buffer.
            byte[] c = new byte[1];
            if (stp == 0)
            {

                c[0] = 0x56; //  0x56= V vertical angle
                sp.Write(c, 0, 1);
                c[0] = (byte)Math.Round(verAngle.Value * 10 / 9);
                sp.Write(c, 0, 1);
                stp++;
                button1.Text = "Apply Settings \n ->2.Step";
                textBox1.AppendText("\r\n");
                return;

            }
            if (stp == 1)
            {

                c[0] = 0x57; //  horizontal angle
                sp.Write(c, 0, 1);
                c[0] = (byte)Math.Round(horAngle.Value * 10 / 9);
                sp.Write(c, 0, 1);
                stp++;
                button1.Text = "Apply Settings \n ->3.Step";
                textBox1.AppendText("\r\n");
                return;

            }

            if (stp == 2)
            {
                c[0] = 0x58; //vertical offset
                sp.Write(c, 0, 1);
                c[0] = (byte)Math.Round(verOffset.Value * 10 / 9);
                sp.Write(c, 0, 1);
                stp++;
                button1.Text = "Apply Settings \n ->4.Step";
                textBox1.AppendText("\r\n");
                return;
            }


            if (stp == 3)
            {
                c[0] = 0x59; //  horizontal offset
                sp.Write(c, 0, 1);
                c[0] = (byte)Math.Round(horOffset.Value * 10 / 9);
                sp.Write(c, 0, 1);
                stp++;
                button1.Text = "Apply Settings \n ->5.Step";
                textBox1.AppendText("\r\n");
                return;

            }
            if (stp == 4)
            {
                c[0] = 0x60; //  V vertical highedge width
                sp.Write(c, 0, 1);
                c[0] = (byte)(hE.Value + lE.Value); // full width
                sp.Write(c, 0, 1);
                c[0] = (byte)(hE.Value);
                sp.Write(c, 0, 1);
                c[0] = (byte)(hE.Value);
                sp.Write(c, 0, 1);
                stp = 0;
                button1.Text = "Apply Settings2";

                textBox1.AppendText("\r\n");
                return;

            }

        }
```