GESTURE BASED VIRTUAL REALITY SYSTEM

A PROJECT REPORT

Submitted by

MOHAMMED SADIQUE G [Reg No: 1041210566]
SHYAM S [Reg No: 1041210549]
GEETHA R [Reg No: 1041210579]

Under the guidance of

Mr.M.MARIA DOMINIC SAVIO

(Assistant Professor(O.G), Department of Electronics & Communication Engineering)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

ELECTRONICS & COMMUNICATION ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

S.R.M. Nagar, Kattankulathur, Kancheepuram District

MAY 2016

SRM UNIVERSITY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this project report titled "**GESTURE BASED VIRTUAL REALITY SYSTEM**" is the bonafide work of " **MOHAMMED SADIQUE G [Reg No: 1041210566], SHYAM S [Reg No: 1041210549], GEETHA R [Reg No: 1041210579], ,** ", who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Mr.M.MARIA DOMINIC SAVIO
GUIDE
Assistant Professor(O.G)
Dept. of Electronics & Communication Engineering

SIGNATURE

Dr.T.RAMA RAO
HEAD OF THE DEPARTMENT
Dept. of Electronics & Communication Engineering

Signature of the Internal Examiner

Signature of the External Examiner

ABSTRACT

Modern world relies on several mediums for communication and one of the most immersive of them all is the electronics entertainment. Current generation of electronics relies heavily on audio, visual and rumble trigger feedbacks. Whereas the next generation will rely on immersive or be in the world experiences for not just in entertainment also for research and education. In fact companies like Tesla and oculus has already started working in these products to make this experiences a reality. Virtual reality HUD is just one of stepping needed to make this technology possible. HUD is a device which is worn by a user like a glass and the inbuilt displays combined with series of refined sensors make the user experience a whole new virtual, imaginary and perceptive worlds. So we believe VR is the next step to not just to entertainment but on bringing radical changes to education, exploration and researches as a whole.

ACKNOWLEDGEMENTS

Apart from the efforts put in by us, the success of the project depend largely on encouragement, guidance and support of many others. we take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of the project

we would like to thank the director of Engineering Technology**Dr.C.Muthamizhchelvan** for providing technical aid and support needed throughout the project.We are also very grateful to our Head of the Department**Dr.T.Rama Rao** (Professor & Head), Department of Electronics and Communication Department for his kind encouragement and co-operation which helped us in completing the project.

we would also like to express our gratitude towards our project coordinator

MR.A.V.M.Manikandaswamy(Assistant professor),Electronics and Communications Department for his kind encouragement and co-operation which helped us in completing this project.

we are highly indebted to class In charge **Mrs.T.Ramya** (Assisstant professor)for her guidance and constant supervision as well as providing necessary information regarding the project topic and also for her support in completing the project.

we would like to express our deepest gratitude to our guide, **Mr.M.Maria Dominic Savio**,(Assistant Professor) his valuable guidance, consistent encouragement, personal caring, timely help and providing us with an excellent atmosphere for doing research. All through the work, in spite of his busy schedule, he has extended cheerful and cordial support to us for completing this research work.

AUTHORS

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

ASCII–American Standard Code for Information Interchange

API–Application Programming Interface

AAC–Advanced Audio Coding

CAD–Computer Aided Deaign

DMP–Digital Motion Processor

DSI–Display Serial Interface

ESD–Electrostatic Discharge

GY521–Gyro sensor521

GPU–Graphics Processing Unit

GPS– Global Positioning System

HMD–Head Mounted Display

LCD– Liquid Crystal Display

LPDDR–Low Power Double Data Rate

LVDS–Low Voltage Data Signal

LDR–Light Dependent Resistor

PPI–Pixel Per Inch

WMC–Winter Music Conference

CHAPTER 1

BACKGROUND

Our Main motivation for this project came, When oculus, a crowd funded start-up focused on providing compact VR devices to masses, was bought by Facebook .This acquisition resulted in escalation of prices from a mere 250 dollars to 600 dollars.Due to this, VR system turned to be costly even for the customers in USA .While taking the economic status of India, this becomes out of reach for most people. As the World is marching towards VR, we believe it is the next step in not just entertainment but also on bringing radical changes in the fields of education, exploration and researches as a whole. As a rising power like India, it is one of the needs to create more skilled intellectuals to realize the dream of turning a leader in technological advancements.

More over there is no proper open source or DIY kit for Virtual Reality available in the market. It makes it difficult for aspiring youngsters in India to learn and develop something new in VR hardware as well as software.

The existing system uses High end processors, sensors and two separate GPUs for providing Virtual Reality feeling to the users, this shoots up the cost of the VR system and makes it unaffordable. To reduce the cost without compromising on quality, Our Project will be using PC for processing along with dragon board 410c, which is the first development board based on a Qualcomm Snapdragon 400 series processor. It features advanced processing power, Wi-Fi, Bluetooth connectivity, and GPS, all packed into a board the size of a credit card and is based on the 64-bit capable Snapdragon 410 processor. To do the head tracking, we will be using the standard 6 axis mpu6050 sensors which are connected with dragon board 410c using Linux kernel for tracking movements of head.

To reduce the complexity in the controlling mechanisms of VR systems, our project also proposes the hand gesture recognition for controlling the VR system. The gesture recognition is achieved by placing the flex sensors on each finger of the gloves to recognize the position of the fingers and we also use the same gyro sensors mpu6050 to

identify the arm movements. Arduino Leonardo board is used for processing the data from the sensors and use it to control the VR system by using ASCII value substitution process. We are transmitting these signals to the pc by using a Bluetooth extension module HC05 connected to the Arduino board.

Since we are transmitting the data from PC to the VR system, there is possibility of delay in the transmission which causes disorientation for the user, so we are transmitting the data via the HDMI cable from the PC which has a High Bandwidth up to 18Gbps for transmitting data which reduces the delay to a very small amount.

The main requirement of this project is to make VR compatible with all OS applications, even the ones which do not provide side-by-side format during display. To convert them to the side-by-side format, we use Tridef 3D, which is a software that ventures into the applications core program and alters it to the required display mode, thereby making the application equipped with the ability to render complete 3D immersive environment.

We are using 1080p TFT LCD display that refreshes at a rate of 60 Hz capable of refreshing the display at 60 fps and also has a wide angle display which uses an interface MIPI 4L 45-pin for pixel communication. We are connecting the display with the pc using an interface card which converts the HDMI input signal to the MIPI 4L signal.

For applying VR in various applications we have taken many non VR application and converted them to VR applications with the help of the 3d rendering software Tridef 3D which basically uses the DirectX Library for converting them. Lumion software is used to convert the cad model/plans for the house to a complete 3D model which can be used for the Civil engineering purposes. We have also converted Google Earth to be able to view in the 3D format for tourism purposes.

CHAPTER 2

INTRODUCTION

2.1 Introduction

Virtual reality is a burgeoning field that has the inherent potential of manipulating peoples mind with a 3D experience using human image perception and illusions. Virtual Reality is a computer simulated environment that gives the user the experience of being present in that environment. Modern world relies on several mediums for communication and one of the most immersive of them all is the electronics entertainment systems. Current generation of electronics relies heavily on audio, visual and rumble trigger feedbacks, whereas the next generation will rely on immersive or be in the world experiences for not just in entertainment, but also for research and education. For instance in the field of education, VR is an asset for students which empowers them to not just visualize, but to immerse into their subject of study. VR also finds an important place in the field of Medicine, Tourism, Gaming, Real Estate, Engineering, Automobile, Military and Armed forces training

2.2 History

Virtual reality may have popped into the headlines only in the past few months, but its roots reach back four decades. Different scientists had different approach to virtual reality starting from Canaletto painting 250 years ago which differs from normal painting and provides the viewers with an illusion of getting immersed into the environment portrayed by it 10.6.

Thomas Alva Edisons invention of Kinetoscope in 1890 provided the next step in evolution by creating the first ever movie which involved continuous motion of a series of 2D images. 10.6.

This painting of Canaletto was the first 3d view of the place which gives illusion of seeing that place for real

Figure 2.1: Painting of Canaletto

Figure 2.2: Edisons Camera

By running continuous images with less difference at high speed Thomas alva Edison created the illusion of moving objects.

His work was followed by numerous improvements in VR but the most important one being Morton Heiligs first head mounted 3-D display which gives artificial sensory experience which is called as Sensoram. 10.6.

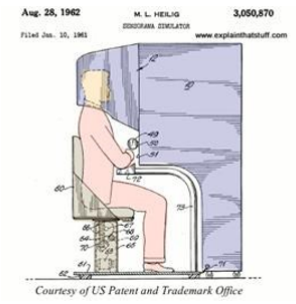


Figure 2.3: Sensoram

Sensoram was the first complete virtual reality device which created a artificial 3d environment. But none of these was able to provide the actual experience of a complete VR system until Oculus Rift revolutionized the field with the invention of a compact VR headset for computer generated graphics in 2012. Now many Electronics Giants

like Samsung, Sony, Google and many more companies are working in virtual reality. Recently Google has released Google cardboard box for smart phones to experience virtual reality. Virtual reality caught everyone eyes when Google has bought magic leap for 541 million dollars and Facebook has bought oculus rift for 2 billion dollars. It is estimated that VR will be 150 billion dollar industry by 2020. 10.6.

Palmer luckey is the first person to invent compact 3d electronic VR headset and started the oculus Rift

Figure 2.4: Oculus VR Headset

Palmer luckey is the first person to invent compact 3d electronic VR heaset and started the oculus Rift.

2.3 Existing System

The existing VR system uses a head mount displays (HMD) .The goal of the hardware is to create what appears to be a life size, 3D virtual environment without the boundaries we usually associate with TV or computer screens. So whichever way you look, the screen mounted to your face follows you. This is unlike AR which overlays graphics onto your view of the real world. To achieve this, the unit employs two small convex lenses placed in front of users eyes to view the complete image/video portrayed by the display screen. 10.6.

Video is sent from the console or computer to the headset in the case of headsets such as Oculus Rift. For Google Cardboard and the Samsungs Gear VR, its already on the smartphone slotted into the headset. Samsungs VR is very much advanced compared to his counterpart Google cardboard as it has additional features such as in-built 9axis Gyro sensor, accelerometer, 2 GPUs for generating separate graphics for each eye and a

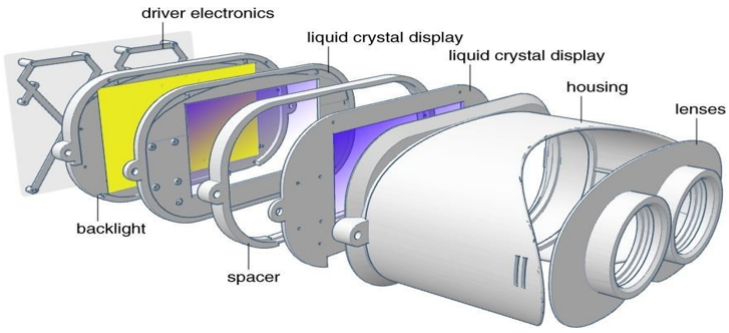


Figure 2.5: VR Headset Architecture

touchpad within its headset whereas Google cardboard simply employs the smartphones in-built sensors which causes a reduction in the quality.

2.3.1 Working Principle

The HMD works based on the principle where the lens is placed between our eyes and the pixels as the light from the display falls on the lenses and converges. The lens focal length and position is chosen in such a way that it will be able to render the complete view to the users. Many sensors such Gyro sensor and Accelerometer are used to track the movements of the head and velocity of the movement respectively. Based on the values of Gyro sensor which depicts the position of the head with respect to x, y and z axes, the image is varied in the display screen, so that the user is able to venture into the virtual 3D environment. 10.6.

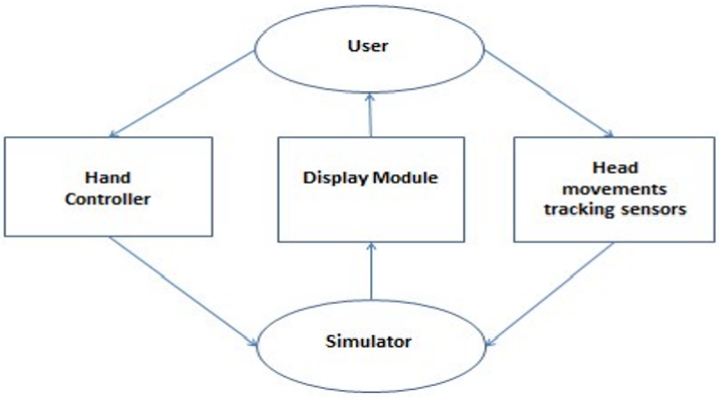


Figure 2.6: Working Principle Flow Chart

2.3.2 Distortion

Geometric distortion is a common and important type of optical distortion that occurs in VR goggles as well as in other optical systems. Geometric distortion results in straight lines not being seen as straight lines when viewed through the goggle optics. The two common types of distortion are barrel distortion and pincushion distortion. A barrel distortion is one where the perceived location of a point in space (e.g. the intersection of two of the grid lines) is farther away from the center relative to where it really is. A pincushion distortion is one where the perceived location of a point in space is closer from the center relative to where it really is. Both these distortions are often radial, meaning that the amount of distortion is a function of how far a point is relative to the optical axis (e.g. the center of the lens system). The reasons distortions are radial is that many optical systems have radial symmetry. 10.6.

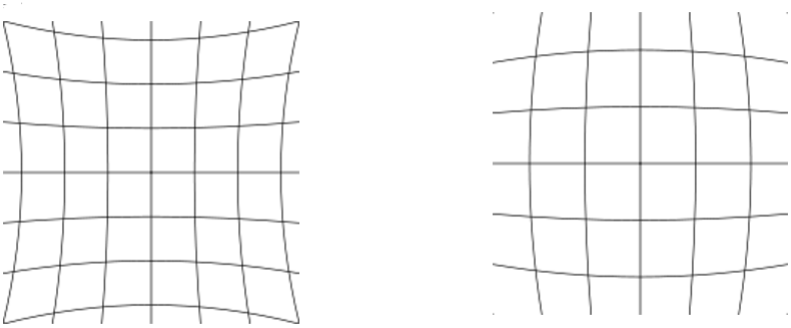


Figure 2.7: pincushion distortion and barrel distortion

2.3.3 Optical illusion

We dont live the world of reality but live in the world of perceptions. Virtual reality uses optical illusions to produce 3d effects to the user. Here an example of a famous optical illusion is explained below. 10.6.

This checkerboard looks curved/slanting until your eyes follow the lines from right to left and you'll realize the only difference are the black and white dots in each corner. This optical illusion makes the image to look curved even though the original image

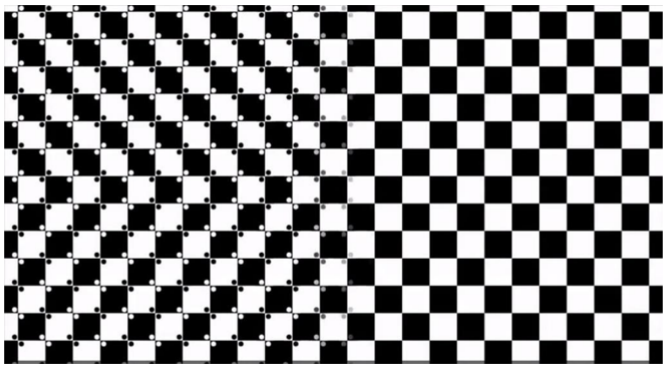


Figure 2.8: Check board Illusion

is straight. Optical illusion that is used in virtual reality is side by side illusion which shows the same image slightly different to both left eye and right eye separately. The brain processes the image seen by both the eyes combines them to single image which gives the feeling of depth to the object of the image in the brains perceptive, this gives the 3d effect to the user. The side by side optical illusion is shown below. 10.6.

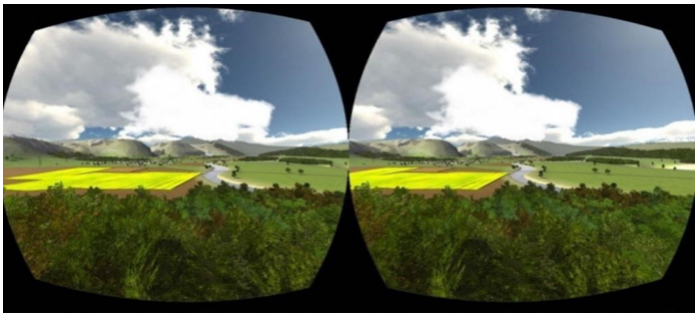


Figure 2.9: side by side optical illusion

This optical illusion along with the head tracking which is provided by the gyro sensor and accelerometer gives the user illusion of being present in that environment. 10.6.

2.3.4 Comparison of various displays

The number of pixels in the display module is very important for virtual reality. This is because the image displayed is zoomed by the convex lens present in the headset, so pixels look separated from others .This problem can be avoided by increasing the number of pixels and pixels per inch (ppi) in the display module. As we approach to 4k

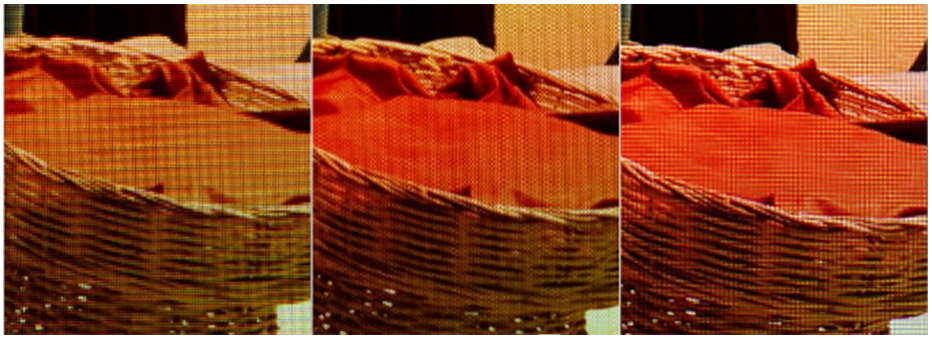


Figure 2.10: Comparison of various displays

display this problem vanishes but perfect virtual reality can be achieved in 8k display but with present technology 8k display cannot created.

There is also a problem called Latency where the user tends to get disturbed by the delay in motion of the image/video displayed in the screen compared to the fast head movement of the user. This causes a severe problem as the users get disoriented and leads to dizziness.

2.3.5 Problems in present system

Although the present VR system seems promising and technologically advanced, there seems to be a lot of problems in it which makes it less user-friendly. VR is out of reach for most of the people due to the high cost. Cost of a high quality VR system is around Rs30, 000/- to 40,000. The present virtual reality systems require very high processing power for providing advanced graphics, so high end processors are used which increases the cost to a mammoth value. Also the present VR systems have very complex controlling mechanisms. The present VR either uses joysticks or touch pads, controlling the VR systems using the joysticks/touch pads are very tough as users have to use them blindfolded. Controls using motion sensor is easier but motion sensors are very costly. In case of Google cardboard, even though card board box are cheap, they fail miserably in high graphics and gives a very low quality experience. And the necessity to use a high end smartphone causes more trouble as their price ranges in high values. The biggest problem lies in the area of OS compatibility as all these systems depend only upon Android OS and IOS which restricts its use with Windows PC.

CHAPTER 3

LITERATURE SURVEY

Sarmad M. Hadi and Ali A. Hussein published in i-Managers Journal on Computer Science, Architectural design validation based on human behaviour analysis in a virtual reality environment. March-May 2015.

At present most architectural designs are not being validated for the clarity of navigation from the customer perspective, and in cases where verification is obtained; testing is conducted after finishing the construction of the facility, where design modification is relatively more expensive and time consuming. Design validation is not a standard process in the design construction chain, and even if validation is executed, it happens after the construction phase, where change is slow and more expensive. The author prefers the usage of VR through which the feedback from customers can be easily speeded up, thus allowing quick design modification at lower cost.

Vanessa N. Palter and Teodor P. Grantcharov published in Annals of surgery review magazine, Individualized Deliberate Practice on a Virtual Reality Simulator to improve Technical Performance of Surgical Novices in the Operating Room, March 2014.

The purpose of the study was to investigate whether individualized deliberate practice on a virtual reality simulator results in improved technical performance in the operating room. Training on VR simulators has been shown to improve technical performance in the operating room. A curriculam of deliberate individualized practice on a VR simulators improves technical performance in the room. This has implications to greatly improve the feasibility of implementing simulation-based curricula in residency training programs, rather than having then being limited to research protocols.

Jamini Prasad Burman, Novel design of low cost flex sensor for automatic controlling of robotic car, International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE), March 2015.

To propose a very low cost flex Sensor less complex in structure, designed with daily used materials and graphite powder. The proposed flex sensor works due to movement of the body that changes the resistance of the sensor according to the bending position of the body. The flex sensor produces different resistance values correlated to the bending radius and can be controlled by using graphite powder with great degree of accuracy. Plot of resistance Vs bend position of flex sensor, Plot of bend position Vs voltage of the flex sensor and plotting of resistance Vs voltage gives clear picture about its working. Velostat, a carbon based conductive sheet share similar properties to graphite powder. So the graphite powder is replaced by the velostat in the flex sensor. The proposed sensor can also be used in industrial purpose, gaming device and measuring device apart from controlling robot car. This sensor is greatly advantageous due to its bidirectional motion control capability. so, using the proposed technology , we can enhance the use of conventional robots by adding human intelligence as decision are taken by operator and working capability of robots.

Head Tracking for the Oculus Rift By Steven M. LaValle Anna Yershova Max Katsev Michael Antonov 2014 IEEE International Conference on Robotics Automation (ICRA) May 31 - June 7, 2014.

The previous methods for head tracking were not efficient and feasible. For efficiently maintaining human head orientation using low-cost MEMS sensors. So by particularly addressing gyroscope integration and compensation of dead reckoning errors using gravity and magnetic fields. Although these problems have been well-studied, their performance criteria are particularly tuned to optimize user experience while tracking head movement in the Oculus Rift Development Kit, which is the most widely used virtual reality headset to date. They were the first present novel predictive tracking methods that dramatically reduce effective latency (time lag), which further improves the user experience. From the above surveys, the fact that should be noted is all these systems suggested by these authors seem to be costly and unaffordable when considering Indian economy. So, In order to reduce the cost and to provide the complete virtual reality to people, mainly students, we concentrate on replacing the present hardware system with cost effective ones which also do not compensate on quality. It also gives students the freedom to develop the entire kit on their own.

CHAPTER 4

HARDWARE REQUIREMENTS

List of hardware requirements

3.1. Low cost flex sensor

3.2. Dragon Board 410C

3.3. 1080p TFT LCD Display

3.4. HDMI to MIPI 4L Interface card

3.5. Arduino Leonardo

3.6. Gyro sensor (GY521 mpu6050)

3.7. VR Headset

3.8. Personal Computer (PC)

4.1 Low cost Flex sensor

Flex sensors are passive resistive devices that can be used to detect bending or flexing. It varies in resistance in accordance to the variation in bending. These are placed on top of each finger to detect their bending. The sensors available in the market seem to be costly as each one costs around Rs.1500-2000. As the aim of this project is to be costeffective, we opted to prepare our own low cost flex sensor which involves the following components.

4.1.1 Black conductive film (Velostat)

Black conductive film (Velostat) is made of linear low density polyethylene to achieve high toughness and strength. The film is carbon loaded for consistent, nonhumidity dependent conductivity. With a surface resistance of less than 105 ohms per square, this material shields against electrical fields. It has a shelf life measured in years and is resistant to common solvents. Typical film thickness is 72 micron (300 gauges). The material is printed in yellow and meets the requirements of EN 61340-5-1: 2001 and includes the ESD warning triangle. It is sold as lay flat tubing on rolls or as bags. 10.6

Figure 4.1: Velostat

Availability and dimensions:

Tubing is available in six standard widths of 3", 4", 6", 8", 10" and 12". Other widths up to 32" (813 mm) are made to order. Standard roll length is 500 m but we will also make to other lengths. Bags are available in standard sizes 3" x 5", 4" x 6", 6" x 8", 6" x 10", 8" x 10", 8" x 12", 10" x 12", 12" x 16" but here too we are pleased to make other requirements. The first dimension is the width of the opening of the bag; the second is the length of the inside of the bag. The bottom skirt of the bag, i.e. from weld to bottom of the bag is approximately 4 mm.

4.1.2 Aluminium Foil

Aluminium is an alloy of 98.5 percent aluminium and the rest being small amounts of tin and chromium. The thickness of foil ranges from the thinnest currently produced

commercially at about 0.0065 mm (or 6.5 m) to the defined upper limit of 0.2 mm (or 200 m).

4.1.3 Masking Tape

Masking tape, also known as sticky tape, is a type of pressure-sensitive tape made of a thin and easy-to-tear paper, and an easily released pressure-sensitive adhesive. It is available in a variety of widths. The adhesive is the key element to its usefulness, as it allows the tape to be easily removed without leaving residue or damaging the surface to which it is applied

4.1.4 Resistor:

Resistors are used here to create a variable voltage drop across the flex sensor which can be read in the microcontroller and processed to find the unknown resistor value. In this project, the resistors used are of the value of 1 kilo ohms (1 Kohm).

4.2 Dragon board 410c

10.6

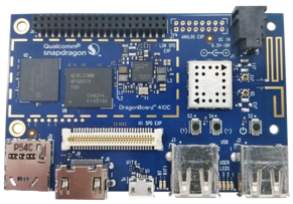


Figure 4.2: Dragon board front view

4.2.1 Processor

- Qualcomm Snapdragon 410
- Quad-core ARM Cortex A53 at up to 1.2 GHz per core 64-Bit capable

- Qualcomm Adreno 306 400MHz GPU for PC-class graphics with support for advanced APIs, including OpenGL ES 3.0, OpenCL, DirectX, and content security

4.2.2 Memory/Storage

- 1GB LPDDR3 533MHz
- 8 GB eMMC 4.51
- SD 3.0 (UHS-I)

4.2.3 Video

- 1080p@30fps HD video playback and capture with H.264 (AVC), and 720p playback with H.265 (HEVC)
- Camera Support
- Integrated ISP with support for image sensors up to 13MP

4.2.4 Audio

- PCM/AAC+/MP3/WMA, ECNS, Audio+ post-processing (optional)

4.2.5 Connectivity

- WLAN 802.11 b/g/n 2.4GHz
- Bluetooth 4.1
- GPS
- On-board GPS antenna
- On board BT and WLAN antenna

4.3 1080p TFT LCD Display

4.3.1 General description

The TF60008A-FUNN model is colour TFT LCD and without touch panel. This module has a 5.98 inch diagonally measured active area with FHD (1080 Horizontal by 1920 Vertical pixel array). Each pixel is divided into red, green and blue dots which are arranged in vertical stripes.

4.3.2 General Feature

- Display mode: Normally black. Trans missive mode
- Viewing mode: Full o clock
- Driving method: asi TFT active matrix
- Input signals: MIPI 4LINES Interface to MPU MIPI
- Outside dimensions: 77.88mm,141.05mm,1.83mm
- Active area : LCD :74.52mm,132.48mm
- Number of Pixels: 1080(RGB),1920 pixels
- Pixel Pitch: 1080(RGB),1920 pixels

4.4 HDMI to MIPI 4L Interface card

MIPI DSI 4 Lane Display S/ HDMI Interface is used here. The Display Serial Interface (DSI) is a high speed packet-based interface for delivering video data to recent LCD/OLED displays. It uses several differential data lanes which frequencies may reach 1 GHz depending on the resolution and frame rate required. With increasing resolution and image quality of multimedia content, huge amounts of data are being transferred between cameras, LCDs and other peripheral devices. These transfers require high speed data rates to keep up with todays baseband and application processors. Toshiba offers interface bridges called Mobile Peripheral Devices (MPDs) that support high-speed data transfer protocols such as MIPI, LVDS, DisplayPort and HDMI.

Toshiba MPDs can not only transfer data at high speeds, but also bridge between main processors and peripherals with different interfaces. Toshiba also offers an extensive portfolio of peripheral devices including IO expanders and SC card controllers. 10.6.

Figure 4.3: DSI Interface

4.5 Arduino Leonardo

The Arduino Leonardo is a microcontroller board based on the ATmega32u4. It has 20 digital input/output pins (of which 7 can be used as PWM outputs and 12 as analog inputs), a 16 MHz crystal oscillator, a micro USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. 10.6. The Leonardo differs from all preceding boards in that

Figure 4.4: Arduino Leonardo Front and rear

the ATmega32u4 has built-in USB communication, eliminating the need for a secondary processor. This allows the Leonardo to appear to a connected computer as a mouse and keyboard, in addition to a virtual (CDC) serial / COM port

4.5.1 Physical Characteristics

The maximum length and width of the Leonardo PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Four screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

4.5.2 Power

The Arduino Leonardo can be powered via the micro USB connection or with an external power supply. The power source is selected automatically. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts. 10.6.

Microcontroller	ATmega32u4
Operating Voltage	5V
Input Voltage recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	20
PWM Channels	7
Analog Input Channels	12
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega32u4) of which 4 KB used by boot loader
SRAM	2.5 KB (ATmega32u4)
EEPROM	1 KB (ATmega32u4)
Clock Speed	16 MHz
Length	68.6 mm
Width	53.3 mm
Weight	20g

Figure 4.5: Summary

4.6 Gyro sensor (mpu 6050)

Gyro sensors, also known as angular rate sensors or angular velocity sensors are devices that sense angular velocity. In simple terms, angular velocity is the change in rotational angle per unit of time. The MPU-6050 is a serious little piece of motion processing tech By combining a MEMS 3-axis gyroscope and a 3-axis accelerometer on the same silicon die together with an on-board Digital Motion Processor (DMP) capable of processing complex 9-axis Motion Fusion algorithms. The device can access external magnetometers or other sensors through an auxiliary master IÂšC bus, allowing the devices to gather a full set of sensor data without intervention from the system processor. The devices are offered in a 4 mm x 4 mm x 0.9 mm QFN package. 10.6.

Figure 4.6: Gyro Sensor

4.6.1 Gyro sensor Features

- Digital-output X-, Y-, and Z-Axis angular rate sensors (gyroscopes) with a user programmable full-scale range of 250, 500, 1000, and 2000/sec
- External sync signal connected to the FSYNC pin supports image, video and GPS synchronization
- Integrated 16-bit ADCs enable simultaneous sampling of gyros Enhanced bias and sensitivity temperature stability reduces the need for user calibration
- Improved low-frequency noise performance
- Gyroscope operating current: 3.6 mA
- Standby current: 5ÂţA
- User self-test

4.7 VR Headset

A virtual reality (or VR) headset is a device that you wear over your eyes like a pair of goggles. It blocks out all external light and shows you an image on high definition screens in front of your eyes. The goal of the VR headset is to immerse you in a game. In VR games, your point of view is your characters point of view; most VR headsets track your head movement, so that wherever you look, your character looks, too. If done well, you will feel like you are inside the game. These headsets are not consoles on their own. They must be connected by a cable to a gaming system computer for the Oculus Rift, or a PlayStation 4 for Sonys Morpheus. They are also called as a head mount displays (HMD). 10.6.

Figure 4.7: VR Headset

4.8 Personal Computer

To process high level graphics, our project employs a PC which is connected to the Dragon board 410C processor board. The PCs with minimum specifications also can be used but the performance varies according to the applications used by the end user. It is recommended for the system to have a good GPU for performing high graphics applications. The project is designed in such a way that the dragon board processor automatically switches those applications with high graphics to PC. It also acts as the source of power supply to the entire model. Also some of the softwares used in the project can be run only with the help of a PC which are rendered to the display screen after converting them into side-by-side viewing format suitable for VR.

CHAPTER 5

SOFTWARE REQUIREMENTS

List of softwares used

- 5.1.Arduino
- 5.2.Lumion
- 5.3.Matlab
- 5.4.Linux Kernel
- 5.5.Tridef 3D
- 5.6.Sweet FX
- 5.7.Google Earth

5.1 Arduino

Arduino is common term for a software company, project, and user community that designs and manufactures computer open-source hardware, open-source software, and microcontroller-based kits for building digital devices and interactive objects that can sense and control physical devices. These systems provide sets of digital and analog I/O pins that can interface to various expansion boards (termed shields) and other circuits. The boards feature serial communication interfaces, including Universal Serial Bus (USB) on some models, for loading programs from personal computers. For programming the microcontrollers, the Arduino project provides an integrated development environment (IDE) based on a programming language named Processing, which also supports the languages C and C++.In our project, Arduino is mainly used in Hand control for calibrating the values of Gyro sensor and flex sensor and using those for implementing gesture recognition and hand movement sensing which can be found helpful for creating user-friendly customizable controls for different applications and mouse control.

5.2 Lumion

Lumion is software that makes it possible for every architect to create videos, images and 360 panoramas. This means the user does not have to outsource. The user does not have to wait for two weeks for someone to build a model of his/her design. He can do it himself and immediately get a sense of how it feels. He can edit his work in real-time and rendering is extremely fast. Effortlessly, the user can edit large areas and add tens of thousands of trees, people or buildings. Export plug-ins is available for Archicad and Revit. Direct export of Sketchup (.skp) files is possible. Many file types including .DAE, .FBX, .SKP and .DWG files can also be imported into Lumion from many CAD software packages. By using this software in VR, the users can get into the virtual 3D world of their property before it is actually built in real life. It makes it helpful for builders and customers to perfectly plan and design, modify, decorate and perform interior designing in advance due to its life-like experience.

5.3 MATLAB

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non- interactive language such as C or FORTRAN. The name MATLAB stands for matrix laboratory. It includes high-level commands for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level commands that allow you to fully customize the appearance of graphics as well as to build complete Graphical User Interfaces on your MATLAB applications. In our project, MATLAB finds use for creating a GUI (Graphical User Interface) at the start of the whole process which is so essential for making it easier for users to select the mode they wish to use. For instance, the GUI

displays a screen which has options for users to choose whether or not to employ hand gesture or gyro control at that time for the specific application they like to use.

5.4 Linux Kernel

The Linux kernel is a Unix-like computer operating system kernel. It is used worldwide: the Linux operating system is based on it and deployed on both traditional computer systems such as personal computers and servers, usually in the form of Linux distributions. The Android operating system for tablet computers, smartphones and smartwatches is also based atop the Linux kernel. The Linux kernel, developed by contributors worldwide, is a prominent example of software. Here, Linux finds its application useful for implementing Head-tracking operations. While using Windows OS, Linux kernel programming helps to correlate movement of head with the mouse movement on the screen. While using Android OS, it is helpful in rendering the view of the user according to the movement of their head.

5.5 Tridef 3D

It is an open source software which converts normal images, videos, games and applications into 3D format. The conversion is so quick and easy. The software also renders side-by-side viewing which is so essential for the user to experience VR. Apart from this, it also gives top-bottom view. These conversions are done by splitting the same display in two screens with each side slightly oriented towards their respective views to bring about a complete VR experience to the user.

5.6 SweetFx

SweetFx is a set of libraries that were designed by open source Modding communities to improve textures and filters of a program. We are using this to create the vignette effect to account for lens distortion. Originally used by modders to create scope like

effect for applications, we are using it for different purpose.

5.7 Google Earth

Google Earth is a virtual globe, map and geographical information program It maps the Earth by the superimposition of images obtained from satellite imagery, aerial photography and geographic information system onto a 3D globe. Google Earth displays satellite images of varying resolution of the Earth's surface. In this project, Google Earth is used mainly for tourism. Here, users can view maps, cities, lakes and hills of different countries depending on their interest which while viewing in VR gives an incredible experience of being present at that place and surrounded by that environment. This application finds a great place in tourism as the 3D experience is sure to attract users and also help them to decide on which location to tour. It also makes it easier for touring agencies to give a real feel of their places of tour.

CHAPTER 6

METHODOLOGY

First Requirement for a VR is to provide a suitable image formats that tricks our brain into thinking as a 3D object, Many formats have been used to provide such illusions. One of the main format that is good for close viewing and wide field of view is SIDE BY SIDE. This is nothing but two images viewed in slightly different angle and by using lenses the brain can be tricked into thinking those images are from a single source effectively bringing in an illusion of 3D. Programs like tridef 3D are used in creation of these images and visual feed. A android development board known as DragonBoard 410c is used to provide platform for integration of sensors for head tracking as well as opening support for Google VR project. A 1080P 5.8 INCH Display is used. It has a pixel transition period in few microsecs and has a PPI of 491. This is connected to a hdmi switch with options for PC and Android mode. On PC mode the sensor values will be taken by PC for tracking head through USB and output will be given to the display via hdmi. In android mode the display will be connected to the board and the sensor already integrated into it will feed the sensors for android VR Apps. 10.6.

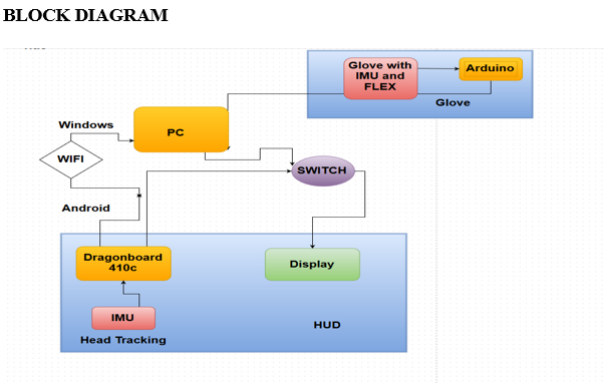


Figure 6.1: BLOCK DIAGRAM

6.1 Explanation

6.1.1 Headset

In the above block diagram, the upper portion within the dotted rectangular box represents the VR headset. The 1080p TFT (Thin Film Transistor) LCD (Liquid Crystal Display) screen is placed inside the headset to which the image, video or application is supplied using an HDMI to MIPI 4L Interface card (shown as interfacing circuit in the diagram) which is connected via the mixer. The input to the Interface card is given using Dragon Board 410C processor board which contains a snapdragon processor. For High graphics applications, the computer automatically switches itself to act as the processor for which the PC uses the same Interface card.

6.1.2 Gesture based hand controller

Pair of hand gloves, each with 5 flex sensors placed on each finger and a gyro sensor placed on the back of hand are employed. Each hand uses a separate Arduino board for itself which is programmed to perform the required hand control functions using ASCII value substitution. The output from these boards are transmitted to the Dragon Board processor/ PC using Bluetooth transmitter HC-05 connected to the Arduino and received on the other side using a Bluetooth receiver.

6.1.3 Personal computer

Here we have used the PC to use all the software and applications available in the windows OS for virtual Reality. For the head tracking on PC side we have concentrated on the mouse control using the gyro sensor.

CHAPTER 7

PROCESS OF IMPLEMENTATION

7.1 Hand Controller

To reduce the complexity in controlling the VR systems, the hand gesture recognition is undergone using flex sensors. The flex sensors available in the market are costly, so we first opted for an alternative for flex sensor which is designed using an LED at one end and Light dependent resistor (LDR) at the other end of well insulated black tube radius of 2-3 mm. In this method, as the amount of light from the LED which falls on LDR varies with respect to the variation in bending of the tube, the resistance in LDR changes whose value can be used for calibrating the sensor. But this method also contains its own disadvantages of not being user-friendly by making it difficult for users to bend the fingers easily with those tubes placed on fingers. As a result, we opted for a better cost- effective flex sensor by using Velostat

7.1.1 Flex sensor using Velostat

Materials and Tools

- Velostat (30 x 25 cm) - 1 sheet.
- Aluminium foil (30 x 300 cm) - 1 sheet.
- Thin wire - few.
- Resistor (1 kohm) - 10 nos.
- Masking tape - 1.

Preparation

Cut the velostat with breadth of 0.8cm and height comfortable to each fingers as per requirement. Also cut two aluminum foils per finger with same dimensions of the

velostat. Solder a thin wire at the end of the each aluminum foils. Cut masking tapes with breadth greater than the velostat and aluminum foil.

Paste the aluminum foil on the sticky side of the masking tape and make two of the same. Place the velostat in-between the two so that the aluminum foils and the thin wires are not in contact with the one at the other side. Now wrap up the sides of the two masking tapes. Place a 1kohm resistor at the middle of one of the wires. The resistor is placed to the gnd; the end of that wire gives the output value. The thin wire at the other end of the flex sensor acts as the Vcc.

Working

The working principle of proposed flex sensor depends upon bending phenomenon. When it is in normal condition i.e. 180, resistance offered by it is maximum. When the sensor is bent to 90, its resistance decreases. The resistance value decreases with further decrease in bend position. This phenomenon is validated with some numerical values. Figure 3 indicates the graphical plot of bent position versus resistance value. It is clearly seen from Figure 3 that as the bending position decreases the corresponding resistance also decreases hence conductivity increases. 10.6.

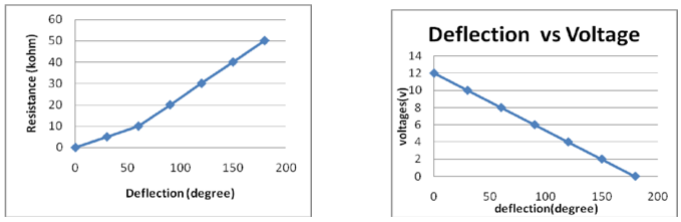


Figure 7.1: Deflection vs Voltage and Resistance vs Deflection

The Arduino cannot read the resistance value directly, it can only read the voltage drop across the unknown resistance, so there should be a reference resistor connected in series with it to form a voltage divider circuit. Here the 1K ohm resistor acts as the reference which makes the voltage drop across the flex sensor variable in accordance with the value of resistance offered by the flex sensor. Now the Arduino can be programmed to find the resistance value from the voltage drop. Then, as the bending of users finger

causes some pressure on the flex sensor placed on the respective finger, the variation in resistance value is used for calibrating it for each position of fingers.

In the circuit shown below, R1 is the 1 K ohm resistor, which is used as the reference resistor. R2 is a variable resistor. 5V is supplied to R2 using Arduino board and R1 is connected to the Gnd pin Arduino. The output is taken at a point between R1 and R2 which is connected to the analog pin A0 of Arduino. 10.6.

Figure 7.2: Flex Sensor Circuit

Calibrating Program for flex sensor:

```
int thumb = 0;
int If = 0; int
mf = 0; int rf =
0; int sf = 0;
int Vin= 5; float
Vout= 0; float
R1= 1000; float
R2= 0; float
R3= 0; float
R4= 0; float
R5= 0; float
R6= 0; float t=
0; void setup()
{
Serial.begin(9600);
}
void loop()
```

```
{ thumb=
analogRead(A0);If=
analogRead(A1);mf=
analogRead(A2); rf=
analogRead(A3); sf=
analogRead(A4); t=
thumb* Vin
Vout= (t)/1024.0;
t= (Vin/Vout) −1; R2=
R1*t;
Serial.print("thumb finger \t");
Serial.println(R2); t= If * Vin;
Vout= (t)/1024.0; t=
(Vin/Vout) −1; R3= R1 * t;
Serial.print("index finger \t");
Serial.println(R3); t = mf *
Vin; Vout= (t)/1024.0; t=
(Vin/Vout) −1; R4= R1 * t;
Serial.print("middle finger \t");
Serial.println(R4); t= rf * Vin;
Vout= (t)/1024.0; t=
(Vin/Vout) −1; R5= R1 * t;
Serial.print("ring finger \t ");
Serial.println(R5); t= sf *
Vin; Vout= (t)/1024.0; t=
(Vin/Vout) −1; R6= R1 * t;
Serial.print("small finger \t ");
Serial.println(R6);
delay(1500);    }
```

This program for the calculation of resistance of theflex sensor is done by using the formula of the

```
resistorsin series combination.
R1 + R2 =R
Current (I) will the same in a series circuit.
1/V = 1/V1+1/V2
V = I*R according to ohmï£¡s law
We can get V1 from the Arduino using it we
can calculate
the V2 value
1/V2 = 1/V-1/V1
```

7.2 GYRO SENSOR

To calculate the position of the arm, in our project, gy521 gyro sensor is used in the hand controller. Then by using the gyro sensor output yaw, pitch and roll, we can detect the position of the hand from these values easily. In order to get correct outputs, we must calibrate the gyro sensors and must compare the raw data values with position of the hand.

7.2.1 Calibration of gyro sensor

To calibrate the gyro sensor we must first calculate the offsets of the gyro sensors otherwise the gyro sensors sensitivity will be very high. The program to calibrate the gyro sensors is given below:

```
"I2Cd#include ev.h"
#include "MPU6050_6Axis_MotionApps20.h"
#if I2CDEV_IMPLEMENTATION ==

I2CDEV_ARDUINO_WIRE
#include "Wire.h"
#endif
MPU6050 mpu;
#define OUTPUT_READABLE_YAWPITCHROLL
```

```
#define LED_PIN 13  bool
blinkState = false; bool
dmpReady = false;
uint8_t mpuIntStatus;
uint8_t devStatus;
uint16_t packetSize;
uint16_t fifoCount;
uint8_t fifoBuffer[64];
Quaternion q;
VectorInt16 aa;
VectorInt16 aaReal;
VectorInt16 aaWorld;
VectorFloat gravity;     float
euler[3];
float ypr[3];
uint8_t teapotPacket[14] =
{ '$', 0x02, 0,0, 0,0, 0,0,
0,0, 0x00, 0x00, '\r', '\n' };
 volatile bool mpuInterrupt
 = false;
 void  dmpDataReady()
 {     mpuInterrupt = true;
} void setup()
{
#if I2CDEV_IMPLEMENTATION ==
 I2CDEV_ARDUINO_WIRE
Wire.begin();
TWBR = 24;
#elif I2CDEV_IMPLEMENTATION ==
 I2CDEV_BUILTIN_FASTWIRE
Fastwire::setup(400, true);
#endif
```

```
Serial.begin(115200);
while (!Serial);
Serial.println(F
("Initializing I2C devices..."));
mpu.initialize()
Serial.println(F
("\nSend any character to begin DMP
 programming and demo: "));
while (Serial.available() &&
 Serial.read());
 while (!Serial.available());
while (Serial.available()
 && Serial.read());
devStatus = mpu.dmpInitialize();
 mpu.setXGyroOffset(220);
mpu.setYGyroOffset(76);
mpu.setZGyroOffset(-85);
mpu.setZAccelOffset(1788);
if (devStatus == 0) {
Serial.println(F("Enabling DMP..."));
mpu.setDMPEnabled(true);
Serial.println(F
("Enabling interrupt detection
 (Arduino external interrupt 0)..."));
   attachInterrupt(0, dmpDataReady, RISING);
   mpuIntStatus =
mpu.getIntStatus();
Serial.println(F
("DMP ready! Waiting for
first interrupt..."));
dmpReady = true;       packetSize =
mpu.dmpGetFIFOPacketSize(); }
```

```
  else { Serial.print(F("DMP
Initialization failed (code "));
Serial.print(devStatus);
Serial.println(F(")")); }
Pinmode(LED_PIN, OUTPUT); }
void loop() {  if (!dmpReady)
 return; while
(!mpuInterrupt &&
fifoCount < packetSize) { }
mpuInterrupt = false;
mpuIntStatus =
mpu.getIntStatus();
 fifoCount =
mpu.getFIFOCount();
if ((mpuIntStatus & 0x10) ||
fifoCount == 1024) {
// reset so we can continue cleanly
mpu.resetFIFO();
Serial.println(F("FIFO overflow!")); }
 else if (mpuIntStatus &   0x02)
{
while (fifoCount < packetSize)
 fifoCount =
mpu.getFIFOCount();
 mpu.getFIFOBytes(fifoBuffer,
 packetSize);
 fifoCount -= packetSize;
#ifdef OUTPUT_READABLE_QUATERNION
mpu.dmpGetQuaternion(&q, fifoBuffer);
endif
#ifdefOUTPUT_READABLE_EULER
mpu.dmpGetQuaternion(&q, fifoBuffer);
```

```
mpu.dmpGetEuler(euler, &q);
#endif
#ifdef OUTPUT_READABLE_YAWPITCHROLL
mpu.dmpGetQuaternion(&q, fifoBuffer);
mpu.dmpGetGravity(&gravity, &q);                    mpu.dmpGetYawPitchRol
&gravity);
//A the program for flex
 and ASCII substitution
 must be inserted here
Serial.print("ypr\t");
Serial.print(ypr[0] * 180/M_PI);
Serial.print("\t");
Serial.print(ypr[1] * 180/M_PI);
Serial.print("\t");
Serial.println(ypr[2] * 180/M_PI);
#endif
#ifdef OUTPUT_READABLE_REALACCEL
mpu.dmpGetQuaternion(&q,
 fifoBuffer);
mpu.dmpGetAccel(&aa, fifoBuffer);
mpu.dmpGetGravity(&gravity, &q);                  mpu.dmpGetLinearAccel(&
 &aa, &gravity);
#endif
#ifdefOUTPUT_READABLE_WORLDACCEL
mpu.dmpGetQuaternion(&q, fifoBuffer);
mpu.dmpGetAccel(&aa, fifoBuffer);
 mpu.dmpGetGravity(&gravity, &q);                   mpu.dmpGetLinearAccel(
mpu.dmpGetLinearAccelInWorld
(&aaWorld, &aaReal, &q);
#endif    #ifdefOUTPUT_TEAPOT
teapotPacket[2]
 = fifoBuffer[0];
```

```
teapotPacket[3] = fifoBuffer[1];
 teapotPacket[4] = fifoBuffer[4];
teapotPacket[5] = fifoBuffer[5];
 teapotPacket[6] = fifoBuffer[8];
teapotPacket[7] = fifoBuffer[9];
teapotPacket[8] = fifoBuffer[12];
teapotPacket[9] = fifoBuffer[13];
Serial.write(teapotPacket, 14);
teapotPacket[11]++;
#endif
 blinkState = !blinkState;
digitalWrite(LED_PIN, blinkState);   }   }
```

The program of the calculation of the flex sensor has to be inserted in between this calibration program at label A marked inside the program. These decreases the program delay as we are avoiding the use of function for taking the values of the yaw, pitch, roll from the program. From these values of yaw, pitch and roll we can use these values in the ASCII substitution for mouse control and keyboard control for the gesture based hand controller. Using these Ypr array values in the conditional statement, we can easily control the VR system using this below given program.

7.3 Program for mouse control and keyboard control

7.3.1 Keyboard control

```
 int a = ypr[1] * 180/M_PI;
int b = ypr[2] * 180/M_PI;
int gyrocontrol=1;          if
(gyrocontrol==1)
{          if
(a<-15)
{
```

```
Keyboard.press('a');
delay(200);
Keyboard.releaseAll();
Table 5.1 Gyrosensor calibration values
}       if
(a>15)
{
Keyboard.press('d');
delay(200);
Keyboard.releaseAll();
}       if
(b<-20)
{
Keyboard.press('s');
delay(200);
Keyboard.releaseAll();
            }
if (b>20)
      {
Keyboard.press('w');
delay(200);
Keyboard.releaseAll();
            }
```

Here the pitch and Roll from the gyro sensor is used to control the characters motion in the virtual reality. The keyboard.press() function is used to emulate the ASCII value of the specified key . The Keyboard.realease() is used to simulate release of that key from the pressed state. These values are taken from calibration program, Based on the angle of inclination of the arm of the user, these values change. So we have programed in such a way where the sensitivity can be decided by the user. In order to vary the time for which the key is pressed we must replace the if conditional statement with loop

statement.

7.3.2 Mouse Control

```
if (R3<= 500)
{x=1;} else
 {x=0;}
 if(R2<=900&&R3<=500&&R4<=380&&R5<=490&&R6<=450)
 {   y=1; }  else
 { y==0;}  if (x == 1) {      if
(!Mouse.isPressed(MOUSE_LEFT)){
Mouse.press(MOUSE_LEFT);
}}
else {
if (Mouse.isPressed(MOUSE_LEFT)) {
Mouse.release(MOUSE_LEFT);
}
}delay(10);
if (y == 1) {
if(!Mouse.isPressed(MOUSE_RIGHT)) {
Mouse.press(MOUSE_RIGHT) ; }  }   else {   if (Mouse.isPressed(MOUSE
{ Mouse.release(MOUSE_RIGHT);
}}      delay(10);
Here the control for the
mouse button program is written.
It is very much similar the
keyboard functions.
Based on the resistance value of
the flex sensor
calculated the mouse buttons are
controlled here.
6.3.3. Mouse pointer motion control:
// initialization part will be in
```

```
the Setup#onetime
int x=0;
// int y=0;
int x1=0;
int y1=0;
int t=10;
int t1=11;
unsigned long int time = 0;
// calibration for gyro sensor for
calculation of ypr valuesï£¡ï£¡
 //will be in the loop
 part along with calibration
prg in ypr output part
int gyrocontrol=1;
int a1 = time/100
time = millis();
if (a1==t) { x=c; x1 =a; }
if (a1==t1)
{ y=c;  y1=a; t1=t+1;}
int z = x-y;
int z1 = x1-y1;
t=a;
if(-2<z<2){Serial.println(z);}
else {Serial.println(z);}
if(-2<z1<2){
Serial.println(z1);}
else {  Serial.println(z1);}
if (gyrocontrol==1)
{if (z!=0||z1!=0) {Mouse.move(z,z1, 0);
delay(50);}
```

Here this program is used to move the mouse pointer as the hand moves. But the

gyro sensor values often change even at same position which causes instability in movement of mouse pointer. In order to maintain the stability, in this code the process of differentiation is implemented with respective time. There is no direct differential functions in Arduino so it implemented by taking values from the same variable in difference of one second and calculation of the difference between these variable will give its variation with time. Time from the program execution is calculated using the millis() function which gives time in milliseconds

7.4 Bluetooth Module Interfacing

10.6. The Bluetooth module HC-05 does not need any interfacing circuit or specific

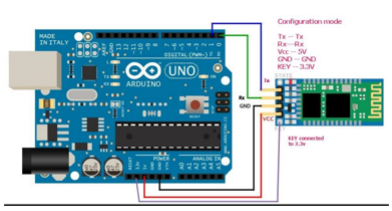


Figure 7.3: Bluetooth Module Interfacing

program to connect it with the Arduino board for performing wireless transmission. The interfacing can be performed by connecting few pins as shown in the above picture.but since the Rx pin operates at 3.3V we should have to join a votage divider circuit along with it so that board doesnt get damage we have to emulate the Bluetooth port as com port for communication.

7.5 LINUX KERNEL

10.6. 10.6. For executing the head tracking for both windows PC and android OS, we use the dragon board in order to avoid the delay in head tracking and disorientation. So we added the Linux kernel program of the dragon board 410c board in order to achieve the head tracking

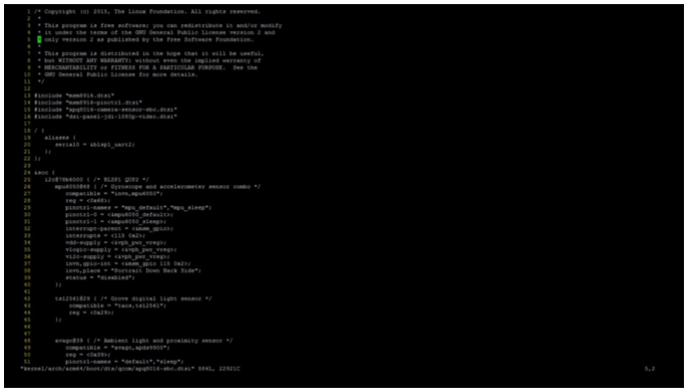
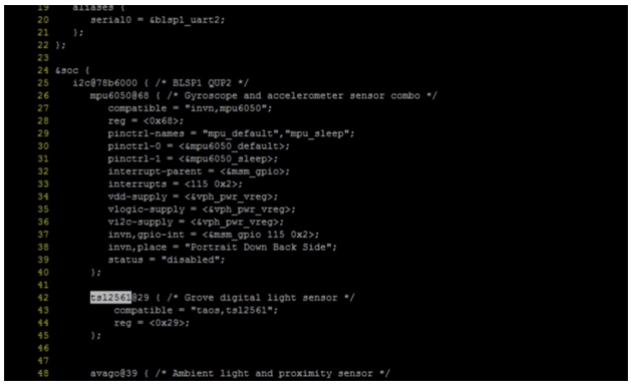


Figure 7.4: Linux Kernel



```
    aliases {
        serial0 = &blsp1_uart2;
    };
};

&soc {
    i2c@78b6000 { /* BLSP1 QUP2 */
        mpu6050@68 { /* Gyroscope and accelerometer sensor combo */
            compatible = "invn,mpu6050";
            reg = <0x68>;
            pinctrl-names = "mpu_default","mpu_sleep";
            pinctrl-0 = <&mpu6050_default>;
            pinctrl-1 = <&mpu6050_sleep>;
            interrupt-parent = <&msm_gpio>;
            interrupts = <115 0x2>;
            vdd-supply = <&vph_pwr_vreg>;
            vlogic-supply = <&vph_pwr_vreg>;
            vi2c-supply = <&vph_pwr_vreg>;
            invn,gpio-int = <&msm_gpio 115 0x2>;
            invn,place = "Portrait Down Back Side";
            status = "disabled";
        };

        tsl2561@29 { /* Grove digital light sensor */
            compatible = "taos,tsl2561";
            reg = <0x29>;
        };


        avago@39 { /* Ambient light and proximity sensor */
```

Figure 7.5: Linux Kernel

7.6 MATLAB

```
function varargout = testgui1(varargin)

gui_Singleton = 1;

gui_State = struct('gui_Name',    mfilename, ...
'gui_Singleton',  gui_Singleton, ...
'gui_OpeningFcn',        @testgui1_OpeningFcn, ...
'gui_OutputFcn',  @testgui1_OutputFcn, ...
'gui_LayoutFcn',  [] , ...
'gui_Callback',   []);

if nargin && ischar(varargin{1})
gui_State.gui_Callback = str2func(varargin{1}); end


if nargout
[varargout{1:nargout}] = gui_mainfcn(gui_State,
varargin{:}); else
gui_mainfcn(gui_State, varargin{:}); end


function testgui1_OpeningFcn(hObject, eventdata, handles,
 varargin)


function varargout = testgui1_OutputFcn(hObject, eventdata,
 handles)
 ha = axes('units','normalized','position',[0 0 1 1]); uistack(ha,'b
I=imread('C:\Users\magesh\Desktop\vr images\
gear-vr-virtual-reality-samsung.jpg'); hi = imagesc(I);
```

```
colormap gray
set(ha,'handlevisibility','off','visible','off');


function pushbutton1_Callback(hObject, eventdata, handles)
A = 1

function pushbutton2_Callback(hObject, eventdata, handles)
A = 0
```

MATLAB finds use for creating a GUI (Graphical User Interface) at the start of the whole process which is so essential for making it easier for users to select the mode they wish to use. For instance, the GUI displays a screen which has options for users to choose whether or not to employ hand gesture or gyro control at that time for the specific application they like to use.

CHAPTER 8

RESULT

The Virtual reality system is designed and number of applications has been successfully executed with it. Here the results are tabulated and explained. By using the above system we were able to create the virtual environment and give a feeling of being visually present in that environment to the users. We have used tridef3D in Non VR software to convert them to side by side format so that it can be viewed in the virtual reality 10.6.

Figure 8.1: Google earth in VR

8.1 Google earth

We Converted the Google earth which is non VR software. This can be used by for Tourisms and even for students when studying about a place. Our project gives the feeling to user for being present in that place itself. 10.6.

Figure 8.2: Google Earth is being displayed in the 1080p TFT display

Figure 8.3: Applications in VR view

10.6.

10.6. The mirror offers the user very high 3d graphics and gives complete immersive

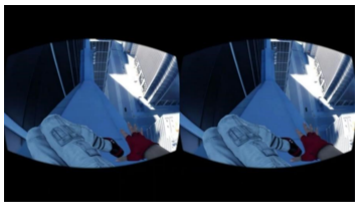


Figure 8.4: Games in VR view

feeling of being in that environment. This application is not just for entertainment but it can be used by doctors to remove the fear of height from the patients

8.2 Lumion

The lumion is a software which is by both civil engineers and real estate agents to convert CAD model plan for the house can be converted in to complete 3d model so now in people can see their house virtual reality just with a plan for that house. This will be very usefull for civil engineers and interior designers to view their design 10.6.

Figure 8.5: House Indoor VR view

10.6. Users not only can just view the plan in the 3d environment but also can make

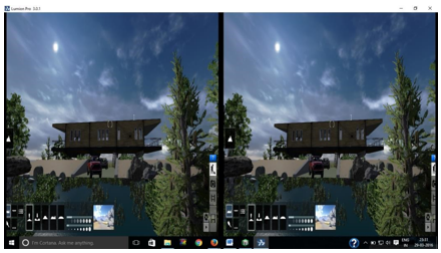


Figure 8.6: Outdoor VR view

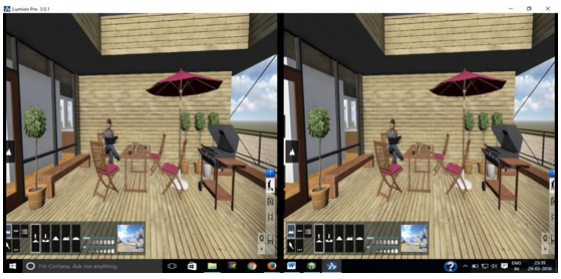


Figure 8.7: View using Lumion software

modifications to the paint and furniture and other details of the house. 10.6. We have also converted the traditional Non-VR shooting games to complete 3-d virtual reality which gives the user entertainment and it also can be used in gun shooting training given to individuals .

CHAPTER 9

CONCLUSION

The Virtual reality system is successfully constructed by using components of lower cost. The Virtual reality system works with features that are similar to the Virtual reality devices available in the market. The components used in this system are cost efficient and also has relatively better performance. Also various customizable hand gestures are provided as controls for the Virtual reality system by using a low cost flex sensor made from Velostat placed over the hand gloves making it user-friendly.

CHAPTER 10

APPENDIX A

10.1 LCD Display Structural Diagram:

10.6.

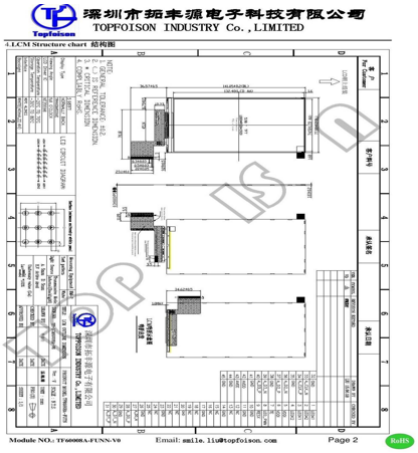


Figure 10.1: LCD Display Structural Diagram

10.2 RESISTOR

10.6.

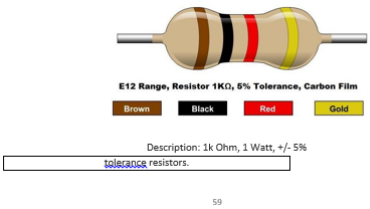


Figure 10.2: Resistor

10.3 Dragonboard 410c

10.6.

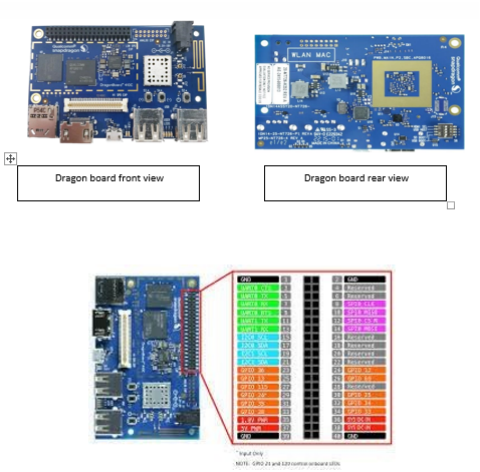


Figure 10.3: Dragonboard 410c

10.4 Dragonboard 410c Architecture

10.6.

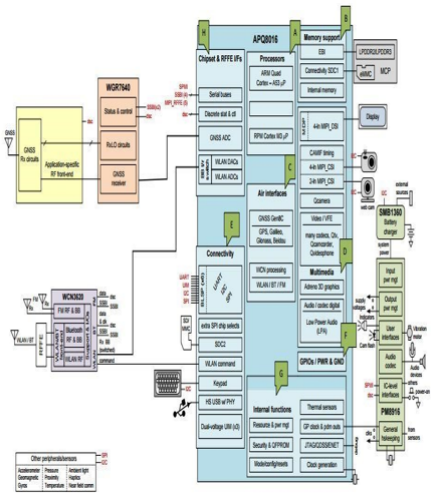


Figure 10.4: Dragonboard 410c Architecture

10.5 Aurdino Leonardo

10.6.

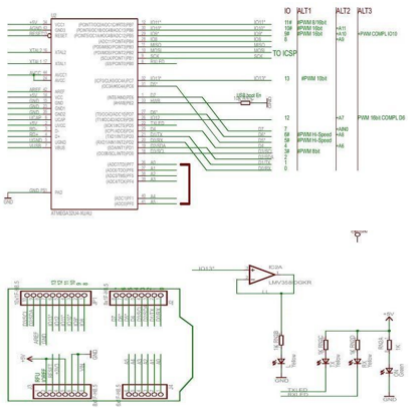


Figure 10.5: Schematic Design

10.6 Bluetooth HC-05

10.6.

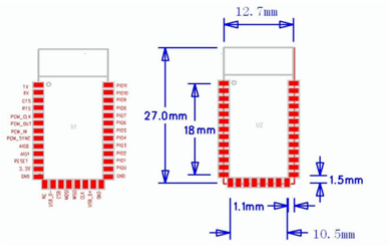


Figure 10.6: Bluetooth HC-05

CHAPTER 11

APPENDIX B

11.1 PROJECT PROPOSAL

REFERENCES

- RiftSteven and M. LaValle and Anna Yershova and Max Katsev and Michael Antonov, Head Tracking for the Oculus,2014 IEEE International Conference on Robotics Automation (ICRA),Hong Kong Convention and Exhibition Center, May 31 - June 7, 2014
- S. A. Zekavat and R. M. Buehrer, Handbook of Position Location. IEEE Press and Wiley, 2012.
- G. Welch and E. Foxlin, âĂIJMotion tracking: no silver bullet, but a respectable arsenal,âĂİ IEEE Computer Graphics and Applications, vol. 22, no. 6, pp. 24âĂŞ38, 2002.
- D. Titterton and J. Weston, Strapdown Inertial Navigation Tech-nology, 2nd ed. Institution of Engineering and Technology, Oct. 2004.
- D. G.-E. and G. H. Elkaim, J. D. Powell, and B. W. Parkin-son, âĂIJCalibration of strapdown magnetometers in magnetic field domain,âĂİ Journal of Aerospace Engineering, vol. 19, no. 2, pp. 87âĂŞ102, 2006.
- M. Abrash, âĂIJLatency: the sine qua non of AR and VR,âĂİ Dec. 2012. [Online]. Available: http://blogs.valvesoftware.com/abrash/latency-the-sine-qua-non-of-arand-vr/
- J. Carmack, âĂIJLatency mitigation strategies,âĂİ Feb. 2013. [Online]. Available: http://www.altdevblogaday.com/2013/02/22/latency-mitigation-strategies/
- S. M. LaValle, Sensing and Filtering: A Fresh Perspective Based on Preimages and Information Spaces, ser. Foundations and Trends in Robotics Series. Delft, The Netherlands: Now Publishers, 2012, vol. 1: 4.