Accelerometer Technology Based Hand Gesture Recognition Glove

M.Tech Thesis Report

Submitted in partial fulfillment of the requirements for the

Award of the Degree

of

Master of Technology

in

Computer Science and Systems Engineering

by

Renitto Jose E

Reg.No:22414

under the guidance of

Mr.Philumon Joseph
Staff Advisors

Department of Computer Science and Engineering
GOVERNMENT ENGINEERING COLLEGE
IDUKKI, KERALA INDIA
August 2013

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

GOVERNMENT ENGINEERING COLLEGE IDUKKI
PAINAVU

CERTIFICATE

Certified that this Main Project report, ***"Accelerometer Technology Based Hand Gesture Recognition Glove "*** *is the bonafide work of* ***"Renitto Jose E"(Reg No:22414)*** *in partial fulfillment of the requirements for the award of the degree of Master of Technology in Computer Science & Systems Engineering under Mahatma Gandhi University during the year 2012-2014.*

Mr.Philumon Joseph	Mr.Sumesh Divakaran
STAFF ADVISOR	**HEAD OF THE DEPARTMENT**

Acknowledgement

First and foremost I thank God Almighty for His blessings for this Thesis. I thank this opportunity to express my gratitude to all those who have helped me in completing seminar successfully. I wish to express my sincere gratitude to our Management for providing infrastructural facilities for me and our Principal, **Dr. J.David** for providing good faculty for guidance.

I wish to express a great depth of gratitude towards our Head of the Department **Mr.Sumesh Divakaran** for helping me. It has been said that gratitude is the memory of the heart. I am thankful to my Main Project guide and Coordinator **Mr.Philumon Joseph** for the guidance throughout the Project.

I am indebted to my beloved teachers whose co-operation and suggestions throughout the project which helped me a lot. I also thank all my friends and classmates for their interest, dedication and encouragement shown towards the seminar. I convey my hearty thanks to my parents for the moral support, suggestions and encouragement to make this venture a success.

Abstract

Communicating with computer involves use of touch screens, wireless/wired mouse along with keyboards. In this Thesis, I propose a Human Computer Interfacing Device mapping Hand Gestures, most intuitive communication gesture, to communicate with computers, robots and other household electronic devices. Developments in field of communication have enabled computer commands being executed using hand gestures. This proposed novel solution achieves effective interaction with Internet giving quality feedback, intuitive control with complete mobility. Simple inertial navigation sensor like an accelerometer is utilized to get dynamic/static profile of movement to navigate the mouse or gyroscope to rotate 3-D virtual objects. Accelerometer profiles are converted into wireless interactivity. The device involves non-tactile interaction with computer screen where mouse pointer reacts to and manipulates screen content in accordance with hand gestures. The applications envisioned: interaction using gesture technology for effective communication empowering physically challenged to interact with computers including 3-D graphic interactions and simulations.

List of Figures

2.1 AcceleGlove . . . 9
2.2 Wired glove . . . 10
2.3 x-OSC Gloves . . . 10
2.4 key Glove . . . 11
2.5 Music Glove . . . 11
2.6 Clove 2 . . . 11
2.7 Glove Mouse . . . 12
2.8 Kitty . . . 12
2.9 Peregrine Gaming Glove . . . 13
2.10 P5 Gaming Glove . . . 13
2.11 Thumbcode . . . 13
2.12 Nintendo Power Glove . . . 14

3.1 I-Glove Top portion . . . 15
3.2 I-Glove Bottom Portion . . . 16

4.1 Circuit Design . . . 19
4.2 Data flow diagram of I-Glove . . . 20
4.3 MicroController side Micro-C code fragment . . . 20
4.4 Client side C-sharp code fragment for PC . . . 20
4.5 Client side android code fragment for Android Device . . . 21
4.6 Packet Format between Transmitter and Receiver . . . 22
4.7 Action and Packet Type Encoded by the Transmitter . . . 22
4.8 Action and Packet Type Decoded by the Base Station(pc) . . . 22

8.1 Screen shot of client software running on pc . . . 31
8.2 Screen shot of client software running on pc and device is connected . . . 31
8.3 Screen shot of client software running on Android device . . . 31

8.4 Screen shot of client software scanning for bluetooth devices in Android device i
8.5 Screen shot of client software paired with I-Glove and the port is listening to incoming data i
8.6 Screen shot of client software running on android device and music app is opened i

Contents

1 Introduction **1**
- 1.1 Aims and Objectives . . . 3
- 1.2 Feasibility Study . . . 3
- 1.3 Structute of the report . . . 3

2 Literature Survey **5**
- 2.1 What is Gestures and Gesture Recognition . . . 5
- 2.2 Motivation . . . 7
- 2.3 Means of Acquisition . . . 8
- 2.4 Overview of the Existing Systems . . . 9
  - 2.4.1 AcceleGlove . . . 9
  - 2.4.2 Wired glove . . . 10
  - 2.4.3 x-OSC Gloves . . . 10
  - 2.4.4 key Glove . . . 10
  - 2.4.5 Music Glove . . . 11
  - 2.4.6 Clove 2 . . . 11
  - 2.4.7 Glove Mouse . . . 12
  - 2.4.8 KITTY . . . 12
  - 2.4.9 Peregrine Gaming Glove . . . 12
  - 2.4.10 Essential Reality P5 Gaming Glove . . . 13
  - 2.4.11 Thumbcode . . . 13
  - 2.4.12 Nintendo Power Glove . . . 13

3 Introduction to the proposed method **15**
- 3.1 Base Design . . . 15

4 Detailed Study on the Implementation **17**
- 4.1 I-Glove . . . 17

4.2 Components 17
4.3 Working 19
4.3.1 Code Fragments 20
4.3.2 Hardware Requirements 20
4.3.3 Software Requirements 21
4.3.4 Wireless Communication 21

5 Performance Evaluation **23**
5.1 Speed of Execution 23
5.2 Accuracy 23
5.3 Usability 24
5.4 Interfacing Issues and Solutions 24
5.5 Algorithm for Data interpretation and Execution 25
5.5.1 Getting data from the Serial port 26
5.5.2 Calibrating 26
5.5.3 Error Detection 27
5.5.4 Mapping Dynamic Data onto Positional Parameters . . 27
5.5.5 Processing Gestures 28
5.6 Advantages of the system 28

6 Conclusion **29**

7 Future Work **30**

8 Appendix **31**
8.1 Screenshots 31

Bibliography **ii**

Chapter 1

Introduction

In the current era, computers are being integrated into every aspect of our lives (eg. Microsoft Surface and Automated Smart Homes) making it essential to move away from the conventional keyboard/mouse interface and delve into intuitive methods of interacting with the computers and other appliances around us.

Also, such devices being tools designed to aid humans, it is in best inter ests if i can make them adapt to our natural communication patterns rather than otherwise. Human hand gestures are a means of nonverbal interaction among people. They range from simple actions of pointing at objects and moving them around to the more complex ones that express our feelings or allow us to communicate with others. To exploit gestures in Human Com puter Interfacing, it is necessary to provide the means by which they can be interpreted by computers. Though various efforts are being made in this direction they face issues of variable reliability, cost and convenience and often need to provide samples to train the device. There was an ef- fort in this direction which eventually lead to

ling a patent A careful study reveals that this device is inadequate for 3-D interactions. This paper describes non-tactile interaction using hand gestures to control number of ap plications running on the computer. The inertial sensor, the accelerometer along with a microcontroller, maps the orientation and position of the hand in 3-D space. This data is then transmitted to the end application where it can be interpreted and acted on, thereby simulating wireless interactivity.

The inspiration to develop such a device is manifold. Firstly, it aims at venturing into the

field of physiological computing and permitting convenient interaction with the surrounding appliances through minimal hardware, as compared to complicated magnetic ux sensors or ultrasonics. Also, the use of hand gestures for interaction avoids the most prevalent injury due to continuous use of the keyboard and the mouse, the Carpal Tunnel Syndrome which occurs when the median nerve that runs from the forearm into the hand gets pressed at the wrist. For the users suffering from Rheumatoid Arthritis causing loss of function at the

nger joints it would be easier to use simple hand gestures which dont require the use of

fingers but only hand movements to perform tasks on a computer. This technology is also of prime importance in dealing with 3-dimensional objects, effective teaching aid and for providing an alternative to the traditional joysticks in the gaming industry.

The hand gesture technology device, the hand glove, can be used to suc cessfully perform the following activities on the computer screen with 100 percentage success rate :

1. Navigate through various Microsoft Windows menus.
2. Drag a window to a different location on the screen.
3. Move the cursor on screen.
4. Interact with on-screen elements.
5. Maximize and minimize windows.
6. Interact with 3-D objects/images on screen.
7. Gaming.
8. Control of external hardware such as robots.

1.1 Aims and Objectives

The project is aimed at improving Accelerometer based Gesture recognition methodology. The main aims and objectives of the project can be listed as

1. To develop an effective Human Interfacing System.
2. Modify the system with relevance feedback from the existing systems so that the accuracy of the results are improved.
3. To overcome the flaws of exisiting Image processing based systems(such as light , colour problems).
4. To create a Low power Consumption device.
5. developing a platform independent device.
6. To make the device as plug n play device.
7. to write driver level program for the developing device.

1.2 Feasibility Study

Feasibility studies aim to objectively and rationally uncover the strengths and weak nesses of an existing system or proposed venture. The opportunities and threats present in the environment, the resources required to carry through, and ultimately the prospects for success. In its simplest terms, the two criteria to judge feasibility are cost required and value to be attained. Here we have the well defined drawbacks for the existing gesture recognition systems.So the proposed idea opens an opportunity to bridge certain deficiencies. The cost required will be worth when we consider the value gained through the work.

1.3 Structute of the report

The Research work is segmented into two phases

1. Phase-1 : In the phase-1 of the project,the base paper is implemented. My base paper is actually proposed in Visual basic, since gesture recognition with Visual basic is not so promising i went for C-sharp. So in

the phase-1 , I developed a Smart Helmet which could control a wheel chair for the physically challenged people on the basis head movement based on accelerometer technology.

2. Phase-2 : In phase-2 I changed the circuit design little bit for higher performance , high accurate data and low power consumption such as plug n play .Designed a glove which works on the basis of accelerometer technology.This Glove can be used as a controlling device for both windows based systems and also for the android based devices .

So the report will be having the following structure

1. Introductory part - where the basic overview, feasibility and the impact of the project is discussed.

2. Literature Review - where the existing system is evaluated, its drawbacks are identified and discuss the solutions to overcome those problems.

3. Proposed Method - where the enhanced system is evaluated and studied in detail. The developing methodologies are discussed and the suitable languages are selected.

4. Design and Implementation - the real time system is developed and the details regarding the implementation is discussed.

5. Performance Evaluation - This is the measure for identifying whether the system proposed was useful or not. The newly developed system will be evaluated and the results will be compared with that of the old system.

6. Conclusion and Future work - The effective results are concluded, possible en- hancements and future works are specified in this section.

Chapter 2

Literature Survey

2.1 What is Gestures and Gesture Recognition

Expressive and meaningful body motions involving physical movements of the hands, arms, or face can be extremely useful for 1) conveying meaningful information, or 2) interacting with the environment. This involves: 1) a posture: a static configuration without the movement of the body part and 2) a gesture: a dynamic movement of the body part. Generally, there exist many-to-one mappings from concepts to gestures and vice versa. Hence gestures are ambiguous and incompletely specified. For example, the concept stop can be indicated as a raised hand with the palm facing forward, or an exaggerated waving of both hands over the head. Similar to speech and handwriting, gestures vary between individuals, and even for the same individual between different instances. Sometimes a gesture is also affected by the context of preceding as well as following gestures. Moreover, gestures are often language- and culture-specific.

They can broadly be of the following types

1. hand and arm gestures: recognition of hand poses, sign languages, and entertainment applications (allowing children to play and interact in virtual environments).

2. head and face gestures: Some examples are a) nodding or head shaking, b) direction of eye gaze, c) raising the eyebrows, d) opening and clos-

ing the mouth, e) winking, f) flaring the nostrils, e) looks of surprise, happiness, disgust, fear, sadness, and many others represent head and face gestures.

3. body gestures: involvement of full body motion, as in a) tracking movements of two people having a conversation, b) analyzing movements of a dancer against the music being played and the rhythm, c) recognizing human gaits for medical rehabilitation and athletic training.

Gesture recognition refers to the process of understanding and classifying meaningful movements of the hands, arms, face, or sometimes head, however hand gestures are the most expressive, natural, intuitive and thus, most frequently used. Gesture recognition has become one of the hottest fields of research for its great significance in designing artificially intelligent human-computer interfaces for various applications which range from sign language through medical rehabilitation to virtual reality. More specifically, gesture recognition can be extremely useful for

1. Sign language recognition in order to develop aids for the hearing impaired. For example, just as speech recognition can transcribe speech to text, some gestures representing symbols through sign language can be transcribed into text.

2. Socially assistive robotics. By using proper sensors and devices, like accelerometers and gyros, worn on the body of a patient and by reading the values from those sensors, robots can assist in patient rehabilitation.

3. Developing alternative computer interfaces. Foregoing the traditional keyboard and mouse setup to interact with a computer, gesture recognition can allow users to accomplish frequent or common tasks using hand gestures to a camera.

4. Interactive game technology. Gestures can be used to control interactions within video games providing players with an incredible sense of immersion in the totally engrossing environment of the game.

5. Remote Controlling. Through the use of gesture recognition, various hand gestures can be used to control different devices, like secondary devices in a car, TV set, operating a garage door, and many others.

Gesture recognition consists of gesture spotting that implies determining the start and the end points of a meaningful gesture trace from a continuous stream of input signals, and, subsequently, segmenting the relevant gesture. This task is very difficult due to two main reasons. First of all, the segmentation ambiguity in the sense that as the hand motion switches from one gesture to another, there occur intermediate movements as well. These transition motions are also likely to be segmented and matched with template traces, and need to be eliminated by the model. The spatio-temporal variability is the second reason since the same gesture may vary dynamically in duration and, very possibly in shape even for the same gesture.

2.2 Motivation

A variety of spontaneous gestures, such as finger, hand, body, or head movements are used to convey information in interactions among people. Gestures can hence be considered a natural communication channel with numerous aspects to be utilized in humancomputer interaction. Up to date, most of our interactions with computers are performed with traditional keyboards, mouses, and remote controls designed mainly for stationary interaction. With the help of the great technological advancement, gesture-based interfaces can serve as an alternate modality for controlling computers, e.g. to navigate in office applications or to play some console games like Nintendo Wii. Gesture-based interfaces can enrich and diversify interaction options and provide easy means to interact with the surrounding environment especially for handicapped people who are unable to live their lives in a traditional way.

On the other hand, mobile devices, such as PDAs, mobile phones, and other portable personal electronic devices provide new possibilities for interacting with various applications, if equipped with the necessary devices especially with the proliferation of low-cost MEMS (Micro-Electro-Mechanical Systems) technology. The majority of the new generation of smart phones, PDAs, and personal electronic devices are embedded with an accelerometer for various applications. Small wireless devices containing accelerometers could be integrated into clothing, wristwatches, or other personal electronic devices to provide a means for interacting with different environments. By defining some simple gestures, these devices could be used to control home appliances for example, or the simple up and down hand movement could be used to operate a garage door, or adjust the light intensity in a room or an office.

2.3 Means of Acquisition

The first step in recognizing gestures is sensing the human body position, configuration (angles and rotations), and movement (velocities or accelerations). This can be done either by using sensing devices attached to the user which can take the form of magnetic field-trackers, instrumental (colored) gloves, and body suits or by using cameras and computer vision techniques. Each sensing technology varies along several dimensions, including accuracy, resolution, latency, range of motion, user comfort, and cost. Glovebased gestural interfaces typically require the user to wear a cumbersome device and carry a load of cables connecting the device to the computer. This hinders the ease and naturalness of the users interaction with the computer. Vision-based techniques, while overcoming this problem, need to contend with other problems related to occlusion of parts of the users body. Most of the work on gesture recognition available in the literature is based on computer vision techniques.Vision-based techniques vary among themselves in the number of cameras used,their speed and latency, the structure of environment such as lighting and speed of movement,any user requirements such as any restrictions on clothing,the low-level features used such as edges, regions, silhouettes, moments, histograms, and others,whether 2D or 3D is used. Therefore, these limitations restrict the applications of vision-based systems in smart environments. More specifically, suppose you are enjoying watching movies in your home theatre with all the lights off. If you decide to change the volume of the TV with a gesture, it turns out to be rather difficult to recognize your gesture under poor lighting conditions using a vision-based system. Furthermore, it would be extremely uncomfortable and unnatural if you have to be directly facing the camera to complete a gesture.

A very promising alternative is to resort to other sensing techniques such as accelerationbased techniques or electromyogram-based (EMG-based) techniques. Acceleration-based gesture control is well-suited to distinguish noticeable, larger scale gestures with different hand trajectories. However, it is not very effective when it comes to detecting more subtle finger movements which is completely overcome by electromyogram-based techniques since they are very sensitive to muscle activation and thus provide rich information about finger movements. Yet, due to some inherent problems with EMG measurements including separability and reproducibility of measurements, the size of discriminable hand gesture set is limited to 4-8 gestures. As a result, in this thesis and after examining all the sensing devices and techniques in literature, a 3-axis accelerometer is the sensing device utilized to acquire the data pertaining to gestures. Gesture recognition based on data from an accelerometer is an emerging technique for gesture-based interaction after the rapid development of the MEMS technology. Accelerometers are embedded in most of the new generation personal electronic devices such as Apple iPhone , Nintendo wii mote which provide new possibilities for interaction in a wide range of applications, such as home appliances, in offices, and in video games. In the sequel, we represent vectors by bold lower case letters.

2.4 Overview of the Existing Systems

This part contains a Survey on Gesture Recognition Systems.

2.4.1 AcceleGlove

Figure 2.1: AcceleGlove

The AcceleGlove instrumented gesture recognition glove (designated iGlove for DoD/NIH applications) has been developed under SBIR grants from the U.S. Army and Department of Education. The iGlove is a low cost gesture recognition system based on patented technology detecting the individual motions of the finger, hand, wrist, and arm. The iGlove is being further refined under a grant from the National Institutes of Health to assist with physical therapy, and the Office of Naval Research for military hand signal recognition. The iGlove, as a robot controller, uses the natural movements of the operators hand/arm as the input device to control both the movement of a robot itself, as well as the movement of ancillary devices such as grasping and lifting arms.

2.4.2 Wired glove

Figure 2.2: Wired glove

A wired glove (sometimes called a "dataglove" or "cyberglove") is an input device for humancomputer interaction worn like a glove. Various sensor technologies are used to capture physical data such as bending of fingers. Often a motion tracker, such as a magnetic tracking device or inertial tracking device, is attached to capture the global position/rotation data of the glove. These movements are then interpreted by the software that accompanies the glove, so any one movement can mean any number of things. Gestures can then be categorized into useful information, such as to recognize Sign Language or other symbolic functions. Expensive high-end wired gloves can also provide haptic feedback, which is a simulation of the sense of touch. This allows a wired glove to also be used as an output device. Traditionally, wired gloves have only been available at a huge cost, with the finger bend sensors and the tracking device having to be bought separately. Wired gloves are often used in virtual reality environments.

2.4.3 x-OSC Gloves

It uses conductive thread and Eeontex fabric for detecting finger flex, and communicates to a host via wifi. It is a curious application of wireless technology, but as the blog reports and this video clearly shows, it does what it

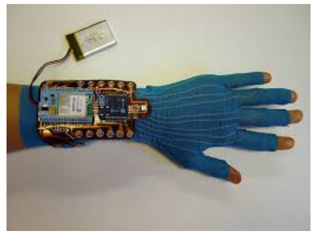

Figure 2.3: x-OSC Gloves

is designed to do quote well. Still a different target application than what Im going for with the Keyglove, but a great glove nonetheless.

2.4.4 key Glove

Figure 2.4: key Glove

The Keyglove is an innovative new way to interact with your technology. A wearable, wireless, open-source input device, the Keyglove provides unprecedented flexibility and convenience for gaming, design, art, music, data entry, device control, 3D object manipulation, and even inexpensive telepresence. The Keyglove uses customizable touch combinations and gestures to enter text data, control the mouse, switch between applications, perform multiple operations with a single action, and even have some fun with equipping weapons, attacking, and defending in MMORPGs and other games.

2.4.5 Music Glove

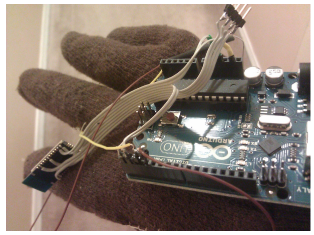

Figure 2.5: Music Glove

The Music Glove by Tyson Bailey is a glove that is meant to control the music player on your smartphone via Bluetooth. Although the intended application is a bit more specific, this warrants mentioning here since the basic control concept is the similar to what the Keyglove does.

2.4.6 Clove 2

Figure 2.6: Clove 2

The Clove 2 is remarkably similar in concept to the Keyglove. It uses the same basic touch combination procedure and allows for full customization through software. However, it only supports 1-to-1 touch combinations, and requires the use of toggled modifier keys to achieve most keys (all lowercase letters can be achieved without modifiers though). The most remarkable aspect of this glove to me is its use of a Bluetooth interface that is cannibalized from an existing wireless keyboard.

Figure 2.7: Glove Mouse

2.4.7 Glove Mouse

The Glove Mouse is an MIT project by project by Tony Hyun Kim and Nevada Sanchez which demonstrated intuitive control of a map application using only ones hands. The gloves combine pushbuttons for direct action with a visually distinct fingertip cap that can be tracked by a webcam to provide mouse cursor control.

2.4.8 KITTY

Figure 2.8: Kitty

The KITTY input device from kittytech.com is another design. KITTY stands for Keyboard Independent Touch Typing. This is a unique kind of glove device which was originally designed for both hands and made to be easy for people who are already good at touch typing. It is not quite like any of the other above mentioned devices, but it is similar in some ways to the Keyglove. The KITTY tries to emulate the QWERTY muscle movements as closely as possible to work with people who already know how to touch type with a real keyboard. As most of the other gloves here, this does not have any mouse control.

2.4.9 Peregrine Gaming Glove

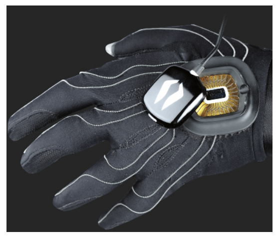

Figure 2.9: Peregrine Gaming Glove

The Peregrine is a well-established glove targeted towards the gaming community. It uses variable-position touch detection (likely via a programmable analog voltage range set) and wound stainless steel wires throughout. The glove has 18 touch points and 3 activator pads which can generate over 30 unique operations. It does not have any motion-based mouse control though.

2.4.10 Essential Reality P5 Gaming Glove

Figure 2.10: P5 Gaming Glove

The P5 Gaming Glove by Essential Reality is an innovative, glove-like peripheral device, based upon proprietary bend sensor and remote tracking technologies, that provides users total intuitive interaction with 3D and virtual environments, such as games, websites and educational software.

2.4.11 Thumbcode

Figure 2.11: Thumbcode

The Thumbcode glove from Stanford has only buttons on the three segments of each of four fingers, and it achieves multiple combinations by detecting which fingers are touching together.

2.4.12 Nintendo Power Glove

Figure 2.12: Nintendo Power Glove

The Power Glove is a controller accessory for the Nintendo Entertainment System.The glove has traditional NES controller buttons on the forearm as well as a program button and buttons labeled 0-9. A person presses the program button and a numbered button to input commands, such as changing the firing rate of the A and B buttons. Along with the controller, the player can perform various hand motions to control a character on-screen.

Chapter 3

Introduction to the proposed method

In the first phase i designed a device named SMART CAP to control movements and clicks of a normal mouse with the help of micro swiches.This Smart Cap was actually meant for the physically challenged persons to control their wheel chair using the movement of this cap. This part is clearly described on the phase-1 report.

In the second phase designed the glove named I-Glove. This glove based on the accelerometer technology and this glove could be used as the controller for windows and android based devices.

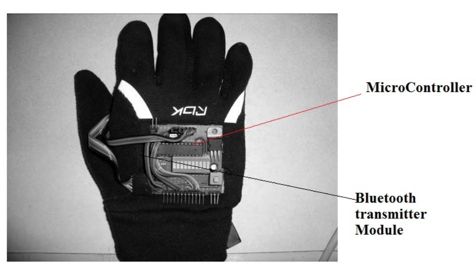

Figure 3.1: I-Glove Top portion

3.1 Base Design

A simple cotton glove is fitted with an accelerometer . This glove is then worn by the user and the desired actions are performed . The accelerometer senses the orientation of the hand in space through an inbuilt capacitive system and feeds out electrical signals as unique voltages for each unique orientation. These voltages are converted to digital values by the microcontroller and then sent to the wireless Bluetooth transmitter-receiver pair, which passes on this data to the computer where it is manipulated by our program running in the background, to understand these gestures and perform actions according to the gestures.

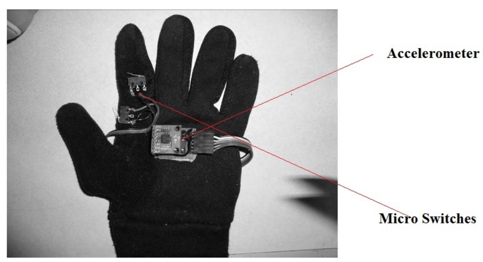

Figure 3.2: I-Glove Bottom Portion

Chapter 4

Detailed Study on the Implementation

4.1 I-Glove

The smart I-Glove is a device consist a normal glove in which a 3D accelerometer is placed and a voltage is generated according to the movement of the accelerometer/hand .This voltage is passed to a microcontroller and it is converted to some digital value .This value is transmited via a bluetooth module wirelessly to the PC.A client program is running on the PC which accepts the digital value send from the glove and according to that value mouse movements are controlled.Micro switches are used to serve the purpose of left click and right click of the mouse and an extra button to control the left movement.

4.2 Components

Cotton Glove

This is a normal cotton glove used by workers in which the the whole module is placed.

Accelerometer/MMA7260Q

This is a 3D accelerometer which generate voltage according to the movement of the hand.An accelerometer is a device that measures proper acceleration.the accelerometer sees the acceleration associated with the phenomenon of weight experienced by any test mass at rest in the frame of reference of the accelerometer device.

MicroController/PIC16F876

A microcontroller (sometimes abbreviated C, uC or MCU) is a small computer on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals. Program memory in the form of NOR flash or OTP ROM is also often included on chip, as well as a typically small amount of RAM. Microcontrollers are designed for embedded applications, in contrast to the microprocessors used in personal computers or other general purpose applications.

Universal asynchronous receiver/transmitter

A universal asynchronous receiver/transmitter, abbreviated UART is a piece of computer hardware that translates data between parallel and serial forms. UARTs are commonly used in conjunction with communication standards such as EIA, RS-232, RS-422 or RS-485. The universal designation indicates that the data format and transmission speeds are configurable. The electric signaling levels and methods are handled by a driver circuit external to the UART.

Bluetooth Module

Bluetooth is a wireless technology standard for exchanging data over short distances (using short-wavelength UHF radio waves in the ISM band from 2.4 to 2.485 GHz from fixed and mobile devices, and building personal area networks (PANs). Invented by telecom vendor Ericsson in 1994,it was originally conceived as a wireless alternative to RS-232 data cables. It can connect several devices, overcoming problems of synchronization.

Micro Switches

A miniature snap-action switch, also trademarked and frequently known as a micro switch, is an electric switch that is actuated by very little physical force, through the use of a tipping-point mechanism, sometimes called an ”over-center” mechanism. Switching happens reliably at specific and repeatable positions of the actuator, which is not necessarily true of other mechanisms.

A windows based Personal Computer

A personal computer (PC) is a general-purpose computer, whose size, capabilities and original sale price makes it useful for individuals, and which is intended to be operated directly by an end-user with no intervening computer operator. This contrasted with the batch processing or time-sharing models which allowed larger, more expensive minicomputer and mainframe systems to be used by many people, usually at the same time. Large data processing systems require a full-time staff to operate efficiently.Personal Computer(pc) in which the client program is running. The Client program will be running on this PC which is written in Csharp in Visual Studio IDE.

Android Device

An APK is created as part of the project such that all the android devices which is installed with this APK can be controlled using I-Glove . This APK is written in android in Eclipse IDE

4.3 Working

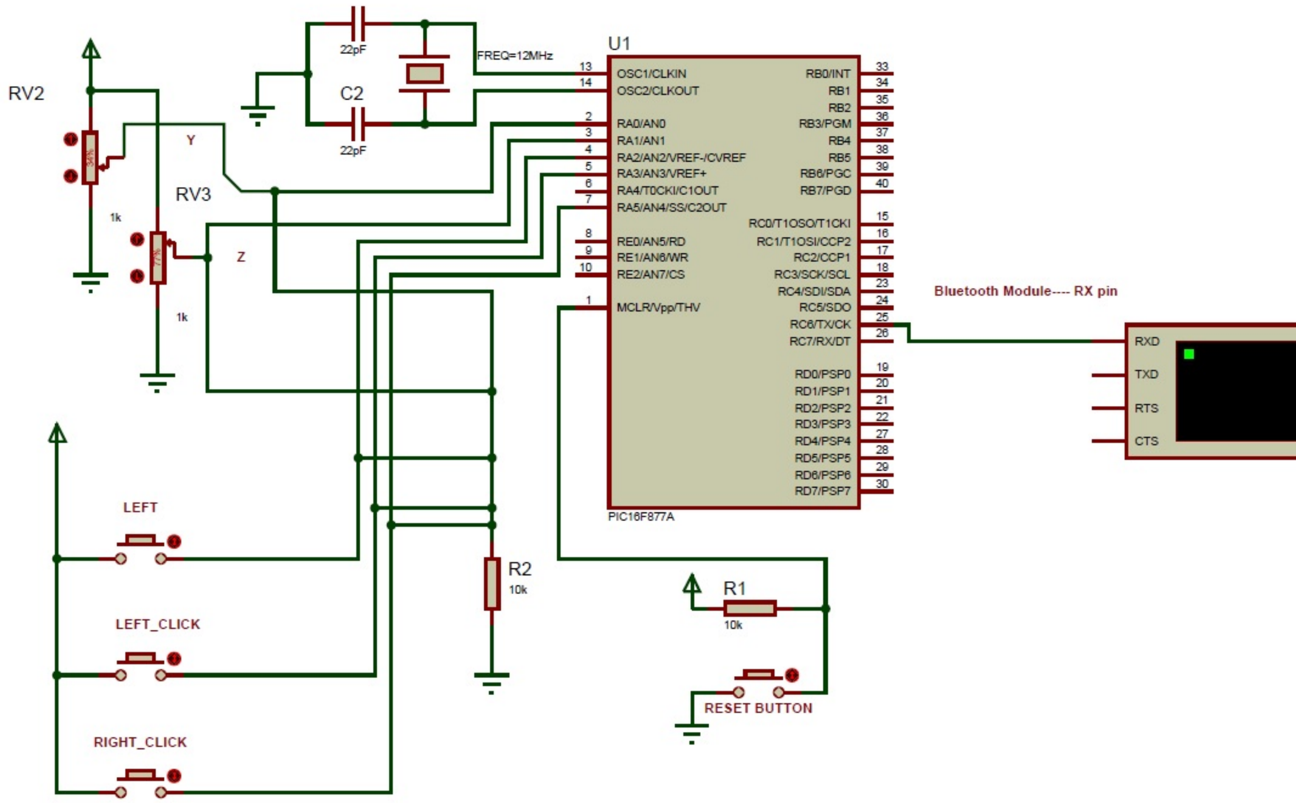

Figure 4.1: Circuit Design

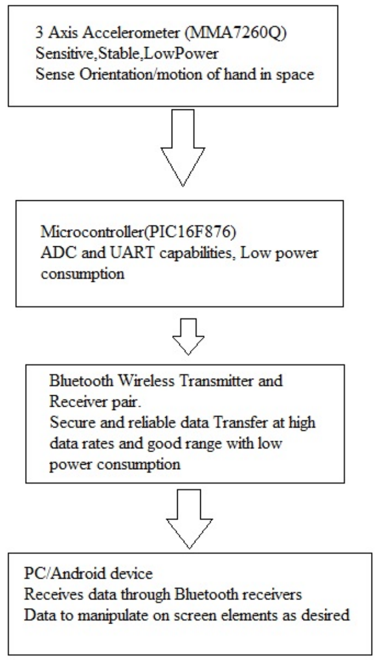

Figure 4.2: Data flow diagram of I-Glove

4.3.1 Code Fragments

MicroController side Micro-C code

Client side code for PC

Client side code for Android Device

4.3.2 Hardware Requirements

1. PROCESSOR : INTEL 2.6 GHz
2. RAM : 512 MB DDR RAM
3. HARD DISK : 40 GB
4. Android Device

```
#include"built_in.h"
unsigned int adc_x,adc_y,adc_z,ch,adc_r,adc_l;
char *text;
long tlong,tlong1,tlong2,tlong3,tlong4;
float a,x,y,rc,lc;



void main()
{

|

       TRISC = 0;          // PORTC is output
       PORTC = 0x00;       //DECLARE INITIAL VALUE OF PORTC IS
ZERR
       TRISB = 0;               // PORTB is output
       PORTB = 0;



       ADCON1     = 0x82;             // configure VDD as Vref,
and analog channels
       TRISA      = 0xFF;                     // designate
PORTA as input

       UART1_Init(9600);              // Initialize UART module
at


  while(1)
       {

           out:
             UART1_Write_Text("out");
              UART1_Write(13);
               UART1_Write(10);

         adc_y  = ADC_read(0);                // get ADC value
from 3rd channel
```

Figure 4.3: MicroController side Micro-C code fragment

4.3.3 Software Requirements

1. FRONT END : .NET frame work , Visual Studio 2012
2. OPERATING SYSTEM : WINDOWS 7/8
3. FRONT END : Eclipse for Android
4. OPERATING SYSTEM : Android Gingerbread or above

4.3.4 Wireless Communication

For wireless communication between the glove and the base station (Teensy 2.0++), two radio transceivers (WI.232FHSS-25-R) were used. Each radio

```
        public static void LeftClick()
        {
            mouse_event(MOUSEEVENTF_LEFTDOWN,
System.Windows.Forms.Control.MousePosition.X,
System.Windows.Forms.Control.MousePosition.Y, 0, 0);
            mouse_event(MOUSEEVENTF_LEFTUP,
System.Windows.Forms.Control.MousePosition.X,
System.Windows.Forms.Control.MousePosition.Y, 0, 0);
        }

        public static void RightClick()
        {
            mouse_event(MOUSEEVENTF_RIGHTDOWN,
System.Windows.Forms.Control.MousePosition.X,
System.Windows.Forms.Control.MousePosition.Y, 0, 0);
            mouse_event(MOUSEEVENTF_RIGHTUP,
System.Windows.Forms.Control.MousePosition.X,
System.Windows.Forms.Control.MousePosition.Y, 0, 0);
        }



        private void Form1_Load(object sender, EventArgs e)
        {
            timer1.Interval = 10;
            timer1.Enabled = true;
            Info_box.Text = "Port disconnected";
            update_avaliable_COM_ports();

        }

        private void textBox1_TextChanged(object sender,
EventArgs e)
        {

        }

        private void ConnectButton_Click(object sender, EventArgs
e)
        {
            try
            {
```

Figure 4.4: Client side C-sharp code fragment for PC

transceiver and microcontroller communicates through wired connection between USART ports. The glove side MCU converts the users glove control into a packet. The packet is then loaded onto the transceiver and is transmitted to the base station. The transceiver on the base station receives this packet and passes it to the base station MCU (Teensy 2.0++). The base station MCU finally decodes the packet information and converts it into a real mouse cursor movement or click.

All four units in our project are programmed with a baud rate of 57600 bits/second. The glove MCU and transmitting transceiver pass data to the base station transceiver and MCU respectively. Thus, glove MCU is initialized as transmitter enabled while the base station MCU is initialized as

```
                        });
                }
                else if (data.contains("UP")) {
                        value = value + data;
                        if (data.length() > 10) {
                                Log.d("Value", "UP");
                                Intent myIntent = new
Intent(

        "android.intent.action.MUSIC_PLAYER");
                                startActivity(myIntent);
                                value = "";
                        }

                } else if (data.contains("RIGHT")) {
                        value = value + data;
                        if (data.length() > 10) {
                                Log.d("Value", "RIGHT");
                                value = "";
                        }

                }

                else if (data.contains("L")) {
                        value = value + data;
                        if (data.length() > 10) {
                                Log.d("Value", "LEFT");
                                value = "";
                        }

                }

                else if (data.contains("LC")) {

                        Intent i = new
Intent("com.android.music.musicservicecommand");
                        i.putExtra("command",
"previous");
                        sendBroadcast(i);

                }
                else if (data.contains("RC")) {

                        AudioManager mAudioManager =
(AudioManager) getSystemService(Context.AUDIO_SERVICE);
```

Figure 4.5: Client side android code fragment for Android Device

receiver enabled for USART setting. The USART is operated in SPI mode with odd parity check enabled, 2-bit stop bits, and 9-bit data size in a frame.

The packet transmitted from the glove side to the base station contains 6 different bytes.

Data Type	Description
SYNC (0xFF)	This byte allows the receiver to synchronize with the transmitter and thus receive a packet in order.
ADDR (0xAA)	This byte prevents the receiver from reading data packets of transceivers other than the glove transceiver.
X axis	The change of the cursor moved on x axis
Y axis	The change of the cursor moved on y axis
Scroll	The change of the scroll moved
Clicks	The click status for left click, middle click, and right click

Figure 4.6: Packet Format between Transmitter and Receiver

Action	Packet Type
Tilt	Only if move is enabled by the move enable button and scroll is not enabled, transmit a packet with x and y axis movement every 8 ms.
Scroll	If scroll is enabled, transmit a packet with 1 or -1 as the scroll data every 720ms, depending on the sign of the y tilt position.
Button Click	Transmit a packet with click packet set to corresponding click value. This contains left click, middle click, and right click.
Left Click with Rapid Fire	Only if rapid fire mode is enabled and the left click button is pressed down, transmit a packet with rapid fire mode value every 30 ms.

Figure 4.7: Action and Packet Type Encoded by the Transmitter

Packet Type	Action
Tilt or Scroll	Move the cursor or scroll by calling *usb_mouse_move*
Button Click	Press or release corresponding button by calling *usb_mouse_buttons*
Rapid Fire	Click the left click button every 60ms by calling *usb_mouse_buttons* repeatedly whenever the rapid fire packet arrives

Figure 4.8: Action and Packet Type Decoded by the Base Station(pc)

Chapter 5

Performance Evaluation

5.1 Speed of Execution

The glove responds to user input fairly quickly. There is a slight delay on the order a several milliseconds when clicking and changing cursor direction. This very small and barely noticeable delay is due to how tightly bound the several computations that take place in our interrupt service routine. During the development phase of this project, sucessfully tested the gloves's primary functions of cursor motion and clicks with viritually no noticeable delay. The addition of several extra user preference features added several calculations on our system, and caused us to slightly slow down the cursor update rate.

5.2 Accuracy

The accuracy of the entire device as a whole is fairly good. In terms of basic functionality 2-D cursor movement and mouse clicking is successfully achieved. Accelerometers placed on the glove measure tilt very well, and thus allow the glove Mouse to adjust the cursor speed based on the magnitude of tilt. The I-Glove can also detect the starting orientation and calibrate the initial axes well. The contact pads and button debouncing state machine are also reponsive and function properly. However, it is noticed the springiness of the contact pads degrades over time. the attribute this to depressing of the foam after hours of testing and hundreds of button presses put them through.

believe that some accuracy is lost via dropped packets between our the tranmission of data from the glove unit to the reception at the base station. first noticed while testing the rapid fire clicking feature that added towards the end of the design phase. Occasionally, the mouse does not unclick after constant rapid fire clicking. Although explicitly send a release left click message to the base station, the base station does not always finish rapid-fire by unclicking the cursor and leaves it "held down. only notice this very occasionally while testing rapid fire clicking, and believe this is because mode of operation subjects the system to many more clicks, thus increasing the chance to see a dropped packet in a noticeable situation. Other than that, the glove performs correctly with expected accuracy.

5.3 Usability

The device is usable to anyone who is able to use a standard computer mouse is able to pick up the glove and learn to use it. There is a small learning curve to controlling a computer mouse with hand tilt rather than existing paradigm everyone knows (2-D device positioning). More notably, users with differently sized hands/fingers, unable to fully control their fingers may, or unable to comfortablly turn their wrists in all directions may have some difficulting operating the glove. Depending on the specific situation, difficulty in using the glove may be alleviated through adjusting the sensitivity of the cursor movement and location of the contact pads.

For prototyping purposes, i have only built a right-handed glove, but note a left-handed glove can be built using the same exact design with the exception of using a complementary glove.

5.4 Interfacing Issues and Solutions

Various problems arose during the development phase while interfacing. Some of them, along with their solutions are listed below.

1. Selection of components had to be made keeping in mind the balance between the application requirements and the need for minimizing power consumption as well as weight, as the device was to be equipped on the users hand.

2. The major issues faced by us while interfacing included data communication between the PIC16F876 and the computer and calibration of the on screen actions to the input received via the bluetooth.

3. Selecting the right data type for carrying information was of atmost importance as it influences the entire data transfer process as well as transfer speeds. This was resolved by designating one data packet as four 16-bit frames. The frames contain button/click information, X Axis Voltage, Y Axis Voltage and Z Axis Voltage in that order.

4. Another issue of great importance was security and reliability of communication and interference by other devices operating in the 2.4 GHz ISM band. The selection of the Bluetooth with capability for device identification resolves this issue.

5. Scaling the data received by the computer to obtain the resulting motion on screen required a lot of tuning of the received data to obtain optimum sensitivity. The issue could be resolved by either hardware filters or software. The latter option was selected by us as it provided greater tuning and customizing ability without addition of any external hardware to the device.

6. Selecting the correct sampling rate of data to attain optimum sensitivity and accuracy as well as to avoid picking up stray natural motions/vibrations of the hand is important. This issue was resolved by selecting an appropriate data transfer rate as well as putting in software checks to resolve the problem of stray natural motion.

7. Another issue encountered was the question whether to totally replace the mouse input if present or to add to the existing mouse input stream, which was solved by selecting the latter, in order to provide a backup input.

5.5 Algorithm for Data interpretation and Execution

The serial data sent by the microcontroller via the UART is read by the software written in Visual sharp using the MSCOMM32.OCX module which interacts with the System Kernel.However one may use any system programming language that can provide access to the core kernel libraries to interact with the mouse parameters.

The following settings are required for initialization

1. Defining the Port no.
2. Setting up the baud rate (9600 bps is the default), setting up parity bits and other error correction parameters.

The following are the essential functions that would be needed to be reused as modules

1. Constructor for the Mouse Pointer Initialization.
2. State functions for defining the mouse clicks and the associated events.
3. Function for sending real time processed of the X, Y positional parameters to the kernel libraries.

The overall code function can be summarized in the following

5.5.1 Getting data from the Serial port

The data is sent in packets, each packet representing one single positional state. Each packet is 32 bits in size consisting of 8 bits each of X, Y, Z and Button State information. The values of X,Y and Z range from 0 to 255 which are mapped from the voltage from the accelerometer.

5.5.2 Calibrating

Since the values of acceleration may vary with altitude as well as with various people using it. The displacement of the positional parameters due to unintentional motions is to be neutralized. For this, we take the first 50 set of packets as sampling packets to these packets we separate out the individual values and take an average, the person wearing the glove is expected to rest his hands in the rest position. The normal value hence obtained is used for subsequent mapping of co-ordinates into proportionate displacement.

5.5.3 Error Detection

The reasons for errors can be

1. Signal noise.
2. Intermittent connection at the hardware.
3. Packet loss.

In order to detect errors in the data, we compare each individual packet data with the previous data and measure if the difference is within limits of the normal deviation expected (The estimate of the standard deviation is done at the Calibration stage) In case we have an abnormal deviation, we neglect the packet and compare the next packet with the last error-free packet. Since we have 300 packets arriving every second we can afford a packet loss up to 10

5.5.4 Mapping Dynamic Data onto Positional Parameters

In order to map the error free packet data onto the screen, we use the following formulae to come up with the proportionate placement of co-ordinates

Xonscreen=(-1)*X * 0.4 * ((X * X) / Xpix)

Yonscreen=Y * 0.4 * ((Y * Y) / Ypix)

where X and Y are the co-ordinates from the error free packets, the Xpix and the Ypix are the respective resolutions along the X and Y axis and Xonscreen and Yonscreen are the resulting positions on the screen. The 0.4 constant is estimated and adjusted according to the aspect ratio of the screen.

The co-ordinates (Xonscreen and Yonscreen) hence obtained are passed onto the MouseMove() function which position the mouse co-ordinates to the respective position.

5.5.5 Processing Gestures

The raw coordinates obtained after the error correction are used in identifying gestures which are pre-defined. A button press on the glove puts it into the Gesture Mode; when in this mode, the co-ordinates are passed onto the Gesture Functions which constantly monitor and store previous co-ordinate data to recognize a pattern. These patterns can be customized and are to be pre-defined as required.

5.6 Advantages of the system

1. Handicapped persons can use this device to control their wheel chair in future
2. Low cost only 1500 INR.
3. Easy to Wear.
4. Fully Wireless.
5. Platform Independent.

Chapter 6

Conclusion

This proposed multi-functional portable device for better human computer interaction using hand gestures can be applied in the following applications:

1. Replace the mouse as a more convenient and natural interaction peripheral.
2. Interacting with 3D objects on computer screen.
3. Easy control of Robots, Robotic Arms and Human Controlled Automation.
4. Easy Home Automation.
5. Effective Teaching / Animation / Design Aid.
6. Control of Machines for High-Precision Jobs.
7. Easy accessibility tool for people with disabilities.
8. Dedicated algorithms, when coupled with this technology, can be used to replace the keyboard as well.
9. When used with other inertial sensors (eg. gyros) the glove can be used to manipulate objects in 3 dimensions.
10. Control over distant actuators connected via a PC interface.
11. It can be used extensively in the gaming industry for remote location manipulation.

Chapter 7

Future Work

I look forward to facilitate rich interactive features which would enable the users to interact and take portability to the next level. Use of smaller packages of the integrated circuits will scale down the size of the device to that of a watch, Thereby improving the portability. To achieve further interactivity with appliances (example, Television), i'm plan to build interfaces (hardware and software) to act as intermediaries between our device and the appliance; these interfaces will be user customizable in terms of hand gestures and resulting actions, enabling control of appliances.

I am working on a Software Development Kit to ease the integration of our device into the software development tool-chain. It can be used by software developers to add hand gesture recognition to a variety of applications from games to educational software. Also thinking to make the system platform independent which works in linux , mac OS and even in android devices such as a Plug n play device with driver written in it.

Chapter 8

Appendix

8.1 Screenshots

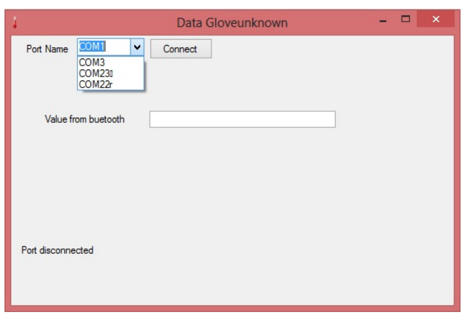

Figure 8.1: Screen shot of client software running on pc

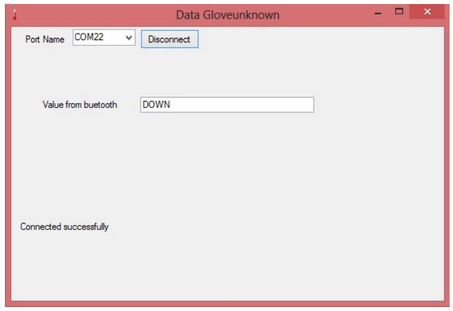

Figure 8.2: Screen shot of client software running on pc and device is connected

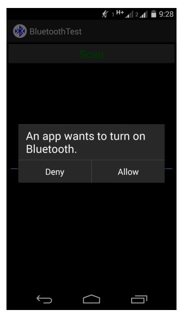

Figure 8.3: Screen shot of client software running on Android device

Figure 8.4: Screen shot of client software scanning for bluetooth devices in Android device

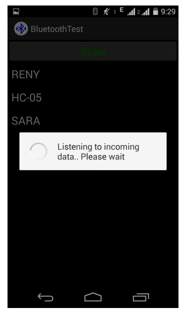

Figure 8.5: Screen shot of client software paired with I-Glove and the port is listening to incoming data

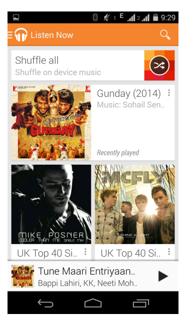

Figure 8.6: Screen shot of client software running on android device and music app is opened

Bibliography

[1] Mark E Lacy; Artificial Laboratories AI Magazine, Summer 1989.

[2] D. Weimer, S. K. Ganapathy; A syntactic Environment with hand gesturing voice input, CHI09 Proceeding pp 235-240.

[3] Microsoft Surface http://www.microsoft.com/SURFACE/Default.aspx.

[4] Vladimir I. Pavlovic, Rajeev Sharma, Thomas S. Huang, "Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 7, pp. 677-695, July 1997, doi:10.1109/34.598226.

[5] Thomas B. Moeslund and Lau Nrgaard,2003, A Brief Overview of Hand Gestures used in Wearable Human Computer Interfaces, Technical report: CVMT 03-02, ISSN: 1601-3646, Laboratory of Computer Vision and Media Technology, Aalborg University, Denmark.

[6] Thomas G. Zimmerman , Jaron Lanier , Chuck Blanchard , Steve Bryson , Young Harvill, A hand gesture interface device, Proceedings of the SIGCHI/GI conference on Human factors in computing systems and graphics interface, p.189-192, April 05- 09, 1987, Toronto, Ontario, Canada.

[7] Logitech Harmony One Advanced Universal Remote http://www.logitech.com/index.cfm/remotes/universalremotes/ devices/3898 and cl=us,en.

[8] Jiayang L., Wang Z., Zhong L., Wickramasuriya J., Vasudevan V., 2009, uWave: Accelerometer-based personalized gesture recognition and its applications, Pervasive Computing and Communications, IEEE International Conference on, Vol. 0 (2009), pp. 1-9.

[9] Acceleglove by Anthrotronix http://www.acceleglove.com/ Patent

[10] Carpal Tunnel Syndrome http://www.ninds.nih.gov/disorders/carpaltunnel/detailcarpal tunnel.htm.

[11] Rheumatoid Arthritis http://www.nlm.nih.gov/medlineplus/rheumatoidarthritis.htm.

[12] Tomoichi Takahashi , Fumio Kishino, Hand gesture coding based on experiments using a hand gesture interface device, ACM SIGCHI Bulletin, v.23 n.2, p.67-74, April 1991.

[13] Atmega8L Data Sheet www.atmel.com/dyn/resources/proddocuments/doc 2486.pdf.

[14] XBee Data Sheet www.mikrocontroller.net/attachment/29126/ZigBit OEMModuleProductDatasheet.pdf

[15] Microsft Kernel Modules and Visual Basic Libraray www.msdn.microsoft.com/

[16] Freeman, Weissman , 1995, Television control by hand gestures, Dept. of Computer Science, University of Zurich.

[17] http://www.keyglove.net/resources/similar/