**Abstract:**

As computer technologies improve year by year, advanced and intuitive user interfaces become increasingly necessary to allow proper manipulation of various types of programs, in different fields and markets. In the case of haptic glove devices, a fraction of the fields that come to mind are rehabilitation after injury, training simulations to learn particular movements or even allowing surgeons to operate on a patient on the other side of the globe. The aim of this project is to design a haptic glove that is lightweight, simple, easy to manipulate while still being precise enough to perform as a user interface for virtual reality systems or any other applications it may have. The glove equipped with sensors that sense the movements of the hand and it’s fingers and interfaces those movements with a computer. Haptic gloves are commonly used in virtual reality environments where the user sees an image of the glove and can manipulate the movements of the virtual environment using the glove.

Table of Contents

**Acknowledgements……………………………………………………………. 3**

Introduction ………………………………………………………….. 4

Chapter 1: System Design …………………………………………… 5

1.1 Master Section………………………………………………………………. 5  
 1.2 Processer Section …………………………………………………………… 7  
 1.3 Slave Section ……………………………………………………………….. 8

Chapter 2: Hardware……………………………………………… 9

2.1 Flex Sensor………………………………………………………………….. 9

2.2 Accelerometers ……………………………………………………………. .. 11   
2.3 Microcontroller …………………………………………………………….. 12

2.4 Voltage Regulator ……………………………………………………………14

Chapter 3: Software Development ……………………………………… 17

3.1 Microcontroller Programming ….……..……………………...…………17

3.2 Human Hand Model…..………...……………………………………..…17

3.3 Problems………….…..………………………………………………..…19

**Chapter 4:** **Hardware/Software Interface Design ……………… 21**

**Conclusion …………………………………………………..… 23**

**Appendices ……..……………………………………………… 24**

**References…………………………………………………...….... 26**

**Acknowledgements:**

The Satisfaction and Euphoria that accompanies the successful completion of any task would be unjust without a profound acknowledgement of the people whose constant guidance, encouragement made it possible for us to execute our project “Haptic Glove” successfully.

The completion of the project work is a milestone in student life and its execution is inevitable in the hands of guide.

We wish to convey our profound gratitude and a sincere thanks to our project supervisors **Dr.Raed Al-Qadi, Dr.Samer Arandi** for being a supreme source of guidance and we are extremely grateful to him having accommodated us into his busy schedule. It is due to his enduring efforts, patience and enthusiasm, which has given a sense of direction and purposefulness to this project and ultimately made it a success.

We would wish to thank the teaching staff, our friends and parents who have helped us all the time in one way or the other.

Really it is highly impossible to repay the debt of all the people who have directly or indirectly helped us in performing the project successfully.

**Introduction :**

### Project Vision: To make virtual object interaction resemble real life.

Our vision is a new interaction paradigm that gives the user an unprecedented ability to touch and manipulate high contrast, high resolution, three-dimensional (3D) virtual objects suspended in space, using a glove that gives realistic whole hand haptic feedback closely resembling interaction with objects in real life.

We created a glove that provides tactile feedback of virtual object. when activated ,it provides real feedback about tactile sensation of holding a virtual object .feedback include (dimensional , movement ) of the object . While this is just a demo, potential applications for a haptic glove includes use in virtual reality, gaming (like playing a 3D video game), online shopping, and physical therapy!

**Chapter 1**

**System Design**

Our system consist of two main components: The first is a whole hand haptic system comprising a glove wired by user, this glove holds flex sensors on its fingers that detect the bending of each finger, also an accelerometer to detect slight changes in hand position in the three dimensions. These sensors are directly connected to the microcontroller which is programmed so that it can process these sensors values and pass it to second part of the system. The second component is a three-dimensional display of a hand model with its animation to simulate the real hand movements. Our driving application is a simple 3D video game that contains the hand model and user can play it using our glove.

The whole project design is divided into three sections:

* Master Section
* Processor Section
* Slave Section

1.1 **Master Section**

The Master Section is nothing but the *Motion Sensing Glove* with its circuitthatgives signals. The glove we use is a normal leather glove with flex sensors attached to each finger. As we move the fingers of the hand with the glove on, the movement causes the flex sensors to bend ,and accelerometer control the direction of movement left ,right , backward ,forward thus producing a signal which causes the virtual hand to move.

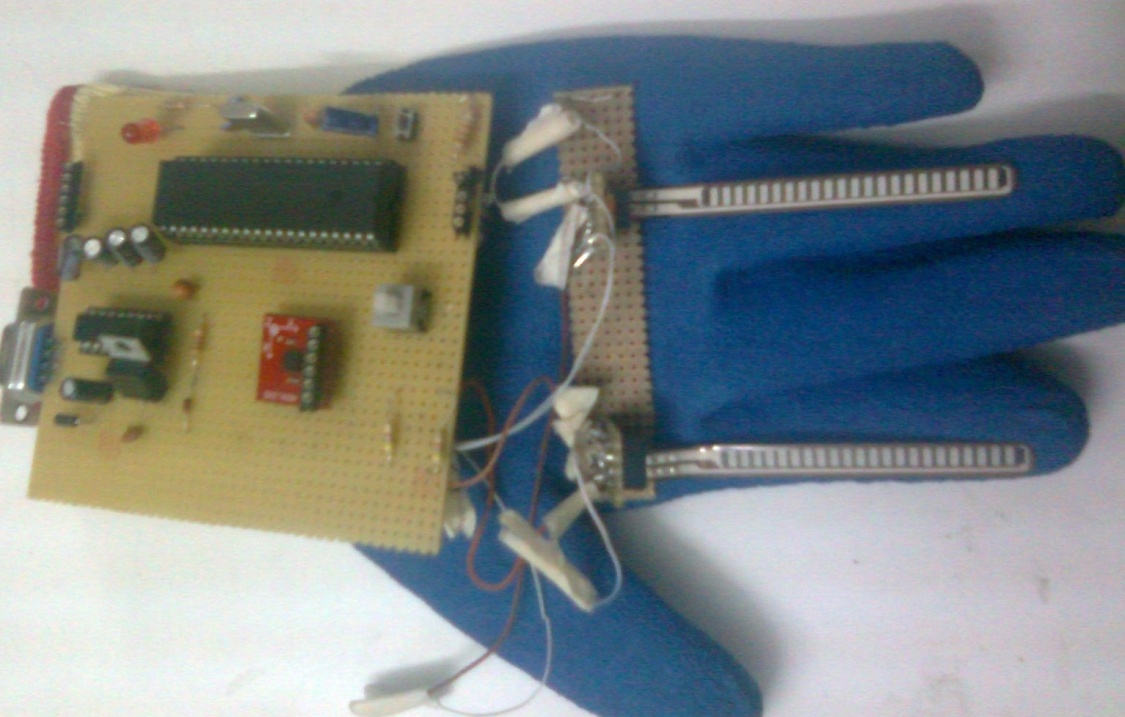


Figure1.1

The flex sensors are connected to a voltage divider circuit .



Figure1.2: *flex Sensor connected to a Voltage divider circuit*

**1.2 Processor Section**

The processor section accepts the signals from the circuit box through the ADC ports. The circuit for the processor section is given below.

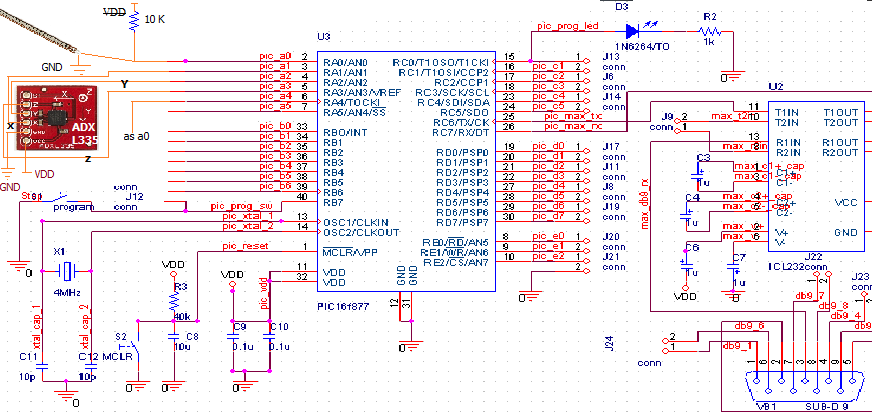


Figure 1.3:Microcontreller connection with sensors

**1.3 Slave Section**

The Slave section is nothing but the *software hand* that mimics the master controller movements. The software hand receive the signal from the master section via the controller.

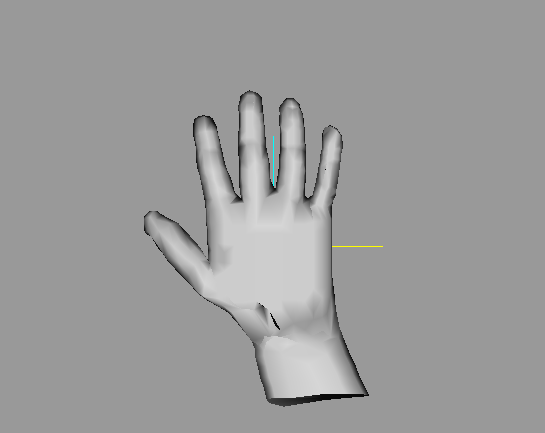


Figure 1.4: 3D Hand Model

**Chapter 2**

**Hardware**

The system is basically a master-slave system where in the master motion sensing glove sits on hosts arm sensing motions of the finger and then using this data to control the virtual hand.

* 1. **Flex Sensors:**

**Flex sensors** are sensors that change in resistance depending on the amount of bend on the sensor. They convert the change in bend to electrical resistance - the more the bend, the more the resistance value. They are usually in the form of a thin strip from “1-5” long that vary in resistance from approximately 10 to 30 kilo-ohms. They are often used in gloves to sense finger movement.



Figure2.1: Flex Sensor \*

The Flex point Bend Sensor was developed through testing a thin potentiometer which would show analog feedback from finger movement. The bend-sensitive carbon-based ink was developed at that point.

**Applications**

Flex sensors are used in gaming gloves, auto controls, fitness products, measuring devices, assistive technology, musical instruments, joysticks, and more.

**How They Work**

Flex sensors are analog resistors. They work as variable analog voltage dividers. Inside the flex sensor are carbon resistive elements within a thin flexible substrate. More carbon means less resistance. When the substrate is bent the sensor produces a resistance output relative to the bend radius. With a typical flex sensor, a flex of 0 degrees will give 10K resistance will a flex of 90 will give 30-40 K ohms. The Bend Sensor lists resistance of 30-250 K ohms.

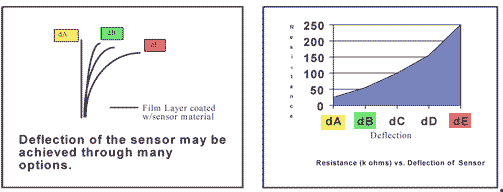


Figure2.2\*

**2.2** **Accelerometers:**

The accelerometer can detect the three-dimensional orientation of the hand.

We use an accelerometer of type ADXL335,the ADXL335 is a triple axis accelerometer with extremely low noise and power consumption - only 320uA! The sensor has a full sensing range of +/-3g. There is no on-board regulation, provided power should be between 1.8 and 3.6VDC. Board comes fully assembled and tested with external components installed. The included 0.1uF capacitors set the bandwidth of each axis to 50Hz.

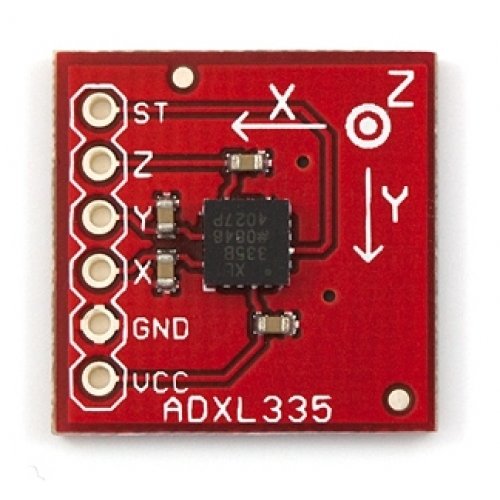
****

Figure2.3\*\*

An accelerometer measures proper acceleration, which is the acceleration it experiences relative to free fall and is the acceleration felt by people and objects. Put another way, at any point in space-time the equivalence principle guarantees the existence of a local inertial frame, and an accelerometer measures the acceleration relative to that frame.

**2.3 Microcontroller**

The processor that is used in our project is microcontroller **Pic18F46.**

*.*

Figure2.4\*\*\*

**Features**

* Operating Frequency.
* Program memory (bytes).
* Program memory (instruction).
* Data Memory.
* Data EEPROM Memory.
* Interrupt Sources.
* I/O Ports.
* Timers.
* Capture/compare/PWM Module.
* Enhanced capture /compare/PWM Module.
* Serial communication.
* Parallel communication.
* 10 bit Analog to Digital Module.
* Resets and delays.
* Programmable low voltages detect.
* Programmable Brown-low voltage reset.
* Instruction set.
* Packages.

**Pin Configuration**

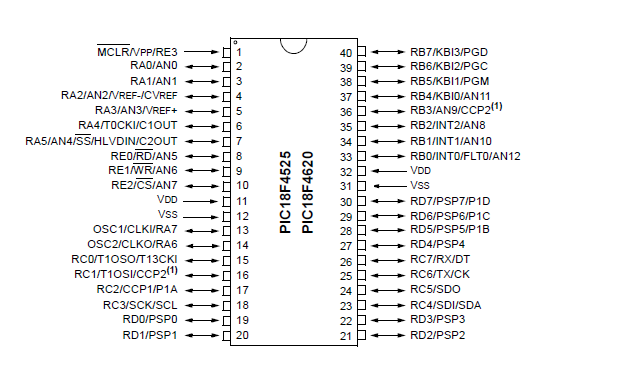


Figure2.5 \*\*\*

**Pin Description**

**VDD:**Digital supply voltage. (+5V)

**VSS:** Ground. (0 V) Note there are 2 ground Pins.

**Port A (PA7 - PA0)**

Port A serves as the analog inputs to the A/D Converter. Port A also serves as an 8-bit bi-directional I/O port, if the A/D Converter is not used. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running.

**Port B (PB7 - PB0)**

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit).

**Port C (PC7 - PC0)**

Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit).

**Port D (PD7 - PD0)**

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit).

**RESET:**Reset Input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running.

**2.4 Voltage Regulator**

A voltage regulator is an electrical regulator designed to automatically maintain a constant voltage level. A voltage regulator may be a simple "feed-forward" design or may include negative feedback control loops. It may use an electromechanical mechanism, or electronic components. Depending on the design, it may be used to regulate one or more AC or DC voltages.

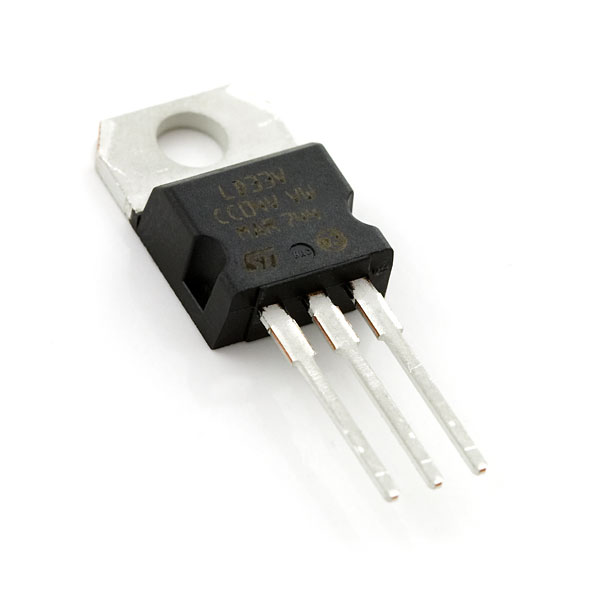


Figure2.6

We use the regulator in our project to convert a 8 volt to 5 volt DC .

**\* Zener Diode as Voltage Regulator**

The function of a regulator is to provide a constant output voltage to a load connected in parallel with it in spite of the ripples in the supply voltage or the variation in the load current and the zener diode will continue to regulate the voltage until the diodes current falls below the minimum IZ(min) value in the reverse breakdown region. It permits current to flow in the forward direction as normal, but will also allow it to flow in the reverse direction when the voltage is above a certain value - the breakdown voltage known as the Zener voltage. The Zener diode specially made to have a reverse voltage breakdown at a specific voltage. Its characteristics are otherwise very similar to common diodes. In breakdown the voltage across the Zener diode is close to constant over a wide range of currents thus making it useful as a shunt voltage regulator.

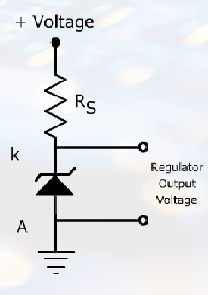


Figure2.7\*\*\*\*

In our project zener diode as voltage regulator convert 5 volt to provided power between 1.8 and 3.6 V DC.

**Chapter 3**

**Software Development**

**3.1- Microcontroller Programming.**

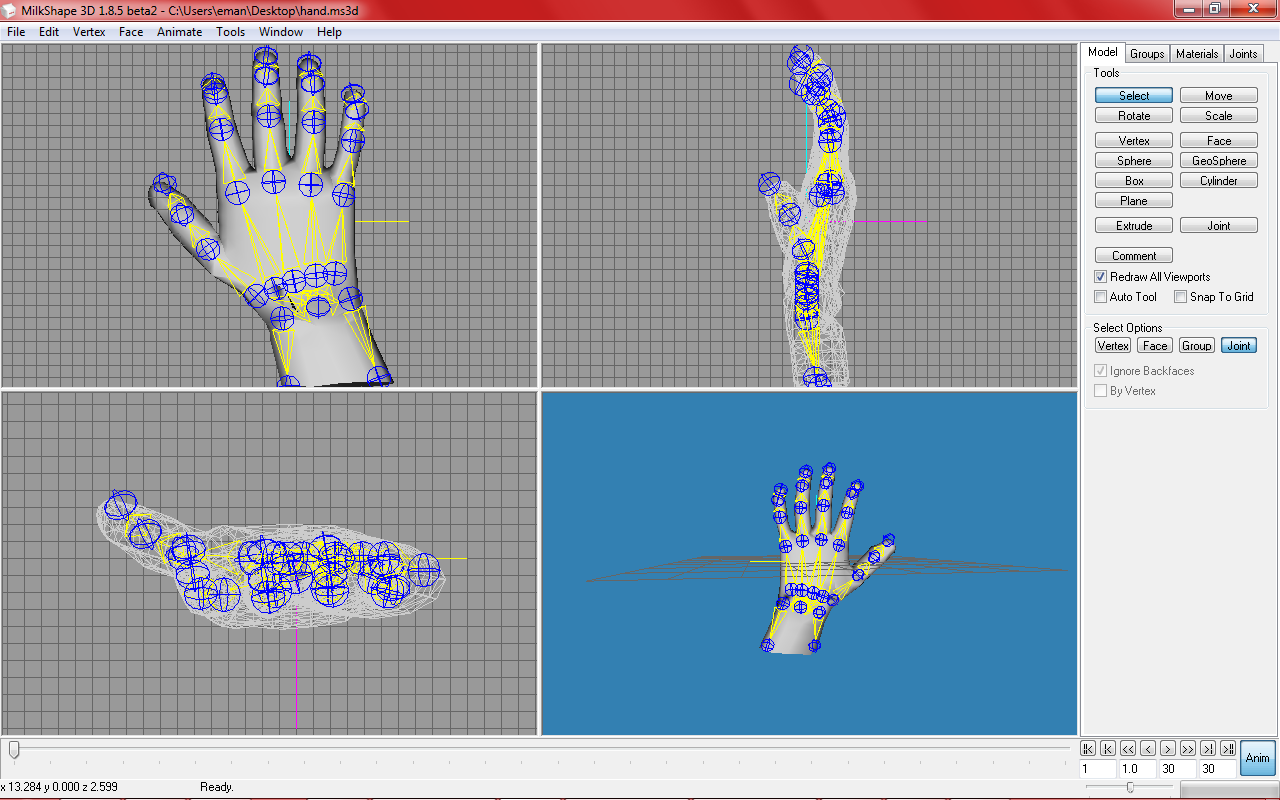
As we mention in H.W part , our basic hardware components are:  
- Flex sensors .

- Accelerometer

We programed the Pic 18F4620 using PIC-C , because we connect these components to the ADC port on pic , in the code we read the ADC from it and convert it to voltage and send it to serial then to software part .

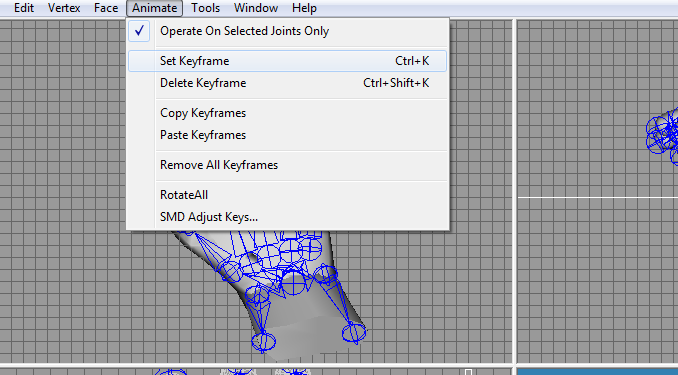
**3.2**- **Human Hand Model**

We created a human hand model using “Milk Shape” software program   
by drawing “Joints” that simulates bones and joints in our hands ,and adding some textures to appear like in this picture

****

*Figure 3.1*

After that we added animation on this model like : rotate fingers, rotate hand left ,right, up and down using “Animate” and then adding Key Frames

****

*Figure 3.2*

“Milk Shape” divides the model into many frames the user control their number, and at each frame we added some animation (key frame) and save it .

After finishing adding animation, we imported the model as” MD2 format”

We choose MD2 because it save the animation with the model after exporting it.

**3.3- Problems and time wasters on this section:**

To create the hand model we started working on “3D Max “ and created a complete hand model with all animation we need with 3ds format(good results with Large time .

Then when we export the model from 3DMAX we realize that the animation not exported with this format ,so we started creating the animation from scratch in OpenGl (coding) , but we find it difficult, need much time , and results was not as efficient as we want.

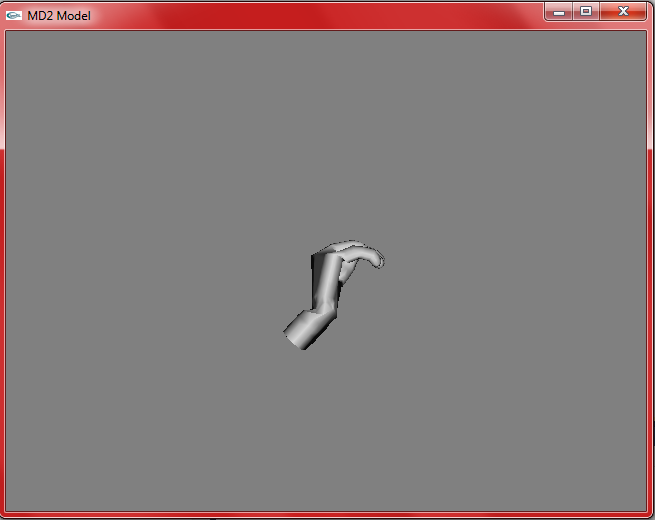
Then search for alternatives , and finally know about MD2 format (that save animation) but MD2 was not found in 3DMAX so leave MAX after this long time of working on it , and moved to “Milk Shape” that can create models with bones and animation to it then export models with MD2 formats.

All these steps take so much time.

**3.4- MD2 Loader** :

Load MD2 model to Open Gl in C++ (read the model and redraw it also play it’s animation).

We search and find ready loaders for md2 models (C++ code). we download one and improve the code to meet our needs and our case .



*Figure 3.3*

So we have a C++ code that view the hand and play it’s animation , then we need to play this model animation dynamically according to values of sensors on Hardware part (Values of flex and accelerometer).So we worked on transferring data from H.W to S.W to control the open GL code (playing specific animations ).

**Chapter 4**

**Hardware/Software Interface Design**

Controlling any virtual object using an external controller requires a communication channel between the controller and the controllable object. This channel is the interface that links the two separate systems. There are several sophisticated and efficient methods available through serial communication that allow a programmer to both establish a real-time virtual interactive system and link in to an external controller. The end result, if accomplished, is utterly impressive as depicted below.

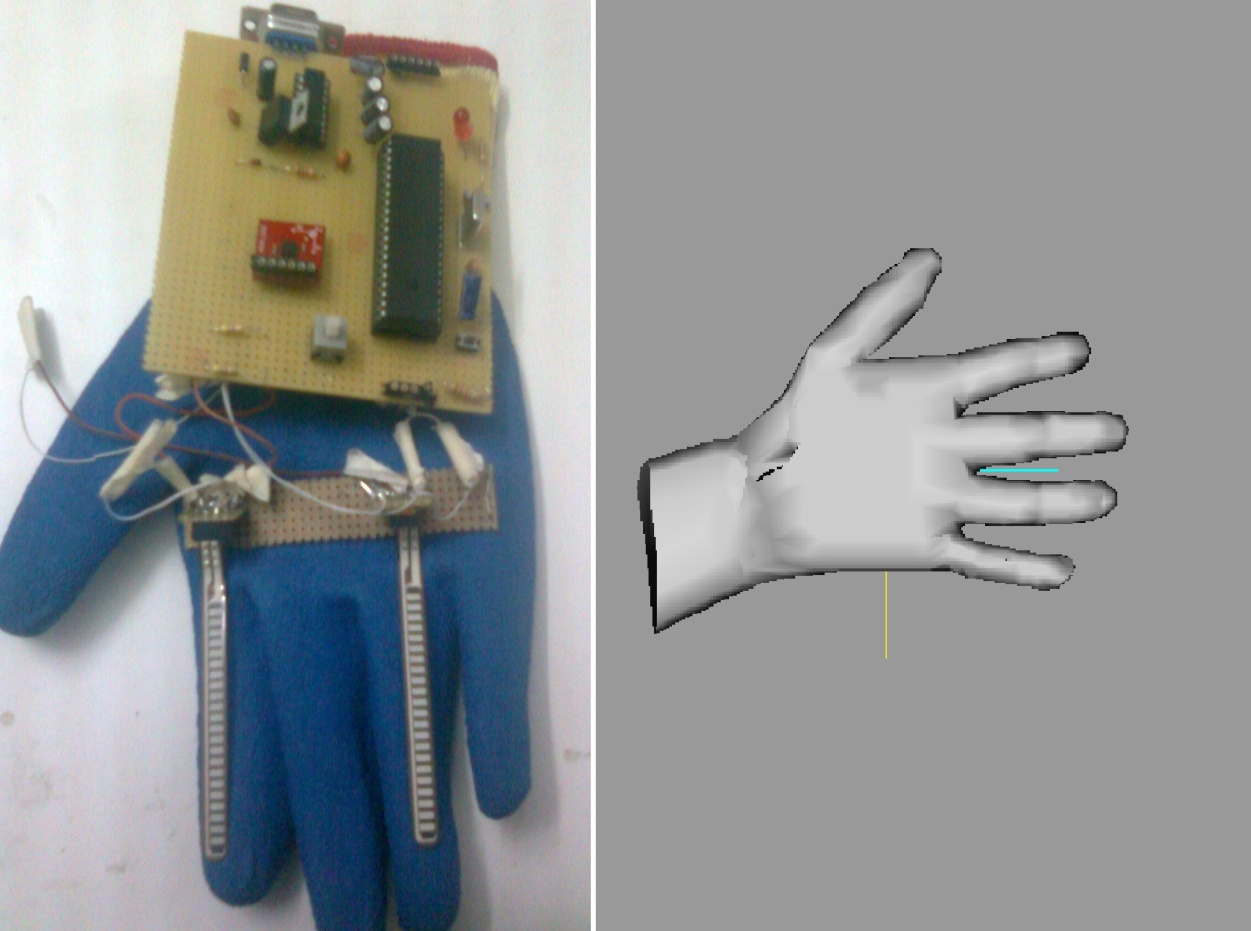


Figure 4.1

Microcontroller read analog value from flex sensor or accelerometer depend on bend degree in flex sensor or direction of accelerometer ,the microcontroller send data to pc using serial communication

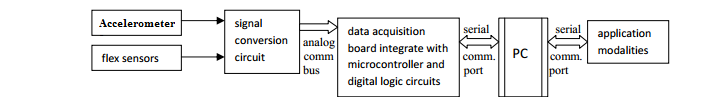


Figure 4.2: block diagram of interface between H.W/S.W

**Serial Port communication :**

We are working now on this part, trying to read serial port in C++ .we read serial successfully, according to these readings the C++ (Loader part) control which frames(specific animations) we want play at this period of time

**Problems in this part:**

We face many difficulties in reading serial port in C++

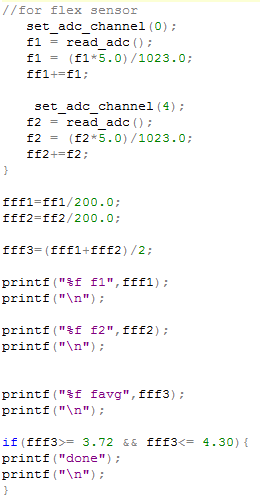
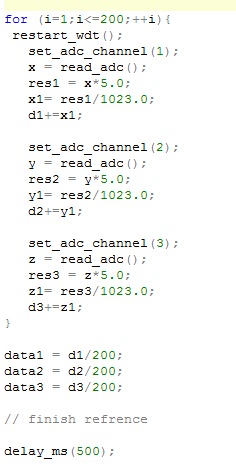
**Conclusion:**

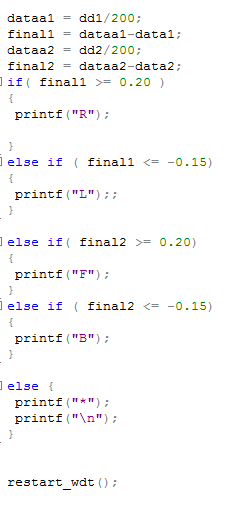
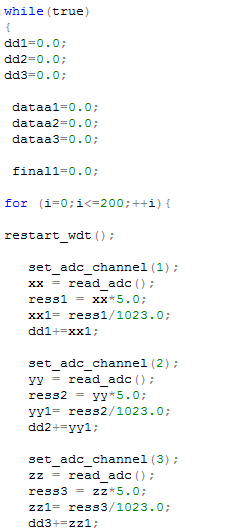
**Our Expectations**

We wish this project to work mostly according to our expectations. We were rather upset that time didn’t help us, so we could not complete what we was intended to do which is a 3D game as an application on simulation of hand motion we wish our problems to be solved soon, so we can start work on this application .However, getting the visual feedback on OpenGl to work far exceeded our expectations, and we are very happy about that. If we were to do this project again, we would have to put more thought and pre-search in design the hand model and it’s animations. This would have avoided the problem of time, and we may have been able to continue designing our own game . if we were to do this again, we would also find ways to make the glove wireless to allow full freedom of motion and incorporate that into our visual feedback as well.

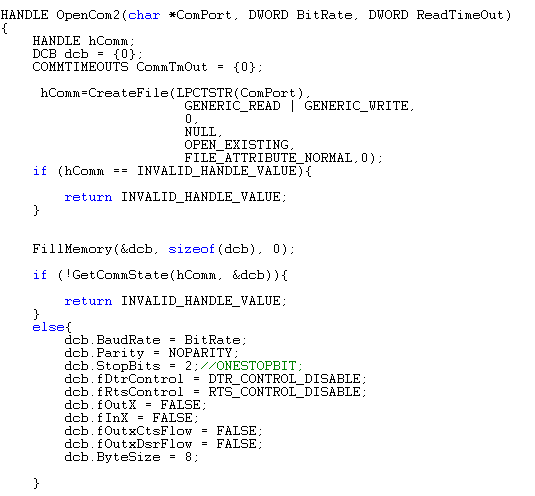
**Appendices:**

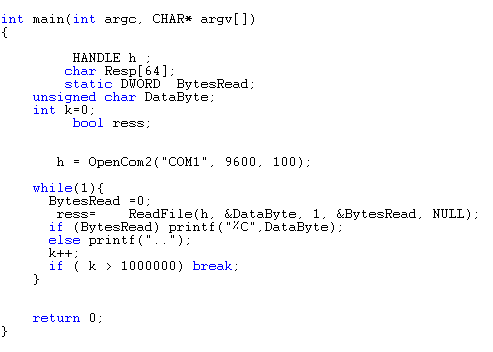
**Program Listings  
- Code for microcontroller**

****



* **Code for read from serial .**





**Cost Details**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Component Description** | **Qty.** | **Price** |
| 1 | Flex Sensor + Shipping Cost | 2 | 40$ |
| 2 | Microcontroller pic18F4620 | 1 | 10$ |
| 3 | 7805 Voltage Regulator | 2 | 4$ |
| 4 | Pin Connectors |  |  |
| 5 | Resistor | 6 |  |
| 6 | Capacitor | 5 |  |
| 7 | LED | 5 |  |
| 8 | 40 pin IC Base | 1 |  |
| 9 | Switch | 1 |  |
| 10 | Accelerometers | 2 | 65$ |
|  | **Total** |  | **120$** |

**Tasks**

Both partners collaborated on all aspects of the project, but Eman Masarweh was responsible for most of the 3d model designing, Yaman Salman was responsible for most of the microcontroller programing , and both partners are equally responsible for all aspects of the glove designing and hardware.

**References:**

(\*)[http://itp.nyu.edu/physcomp/sensors/Reports/Flex](http://itp.nyu.edu/physcomp/sensors/Reports/Flex" \t "_blank)

(\*\*)[http://en.wikipedia.org/wiki/Accelerometer](http://en.wikipedia.org/wiki/Accelerometer" \t "_blank)

(\*\*\*)[http://www.microchip.com/wwwproducts/devices.aspx?ddocname=en010304](http://www.microchip.com/wwwproducts/devices.aspx?ddocname=en010304" \t "_blank)

(\*\*\*\*)<http://www.wisconline.com/objects/ViewObject.aspx?ID=SSE7005> <http://homepage.eircom.net/~abyrne/Models/model9/tutorials_tut09.html>

<http://www.3dcodingtutorial.com/Your-first-OpenGL-program/Getting-GLUT.html>