Department of Electrical and Computer Engineering
North South University

Senior Design Project

Animatronic Hand Controller

Faculty Advisor

Dr. Mohammad Monirujjman Khan (KMM)
Assistant Professor, Department of Electrical and Computer Engineering
North South University, Dhaka, Bangladesh

Submitted By

Section: 7

NAME	ID
ABDULLAH AL ASIF	152 0374 643
ANIKA TAHSIN	152 0307 643

DECLARATION

This is to certify that this Project is our original work. No part of this work has been submitted elsewhere partially or fully for the award of any other degree or diploma. Any material reproduced in this project has been properly acknowledged.

Students' name & Signature

1. **Abdullah Al Asif**

   ____

2. **Anika Tahsin**

   ____

APPROVAL

The Senior design project entitled **"Animatronic Hand Controller" – A Cost Effective and User-Friendly System**" by **Abdullah Al Asif (ID#1520374643), Anika Tahsin (ID#1520307643),** is approved in partial fulfillment of the requirement of the Degree of Bachelor of Science in Computer Science and Engineering on May and has been accepted as satisfactory.

Supervisor's Signature

Dr. Mohammad Monirujjman Khan (KMM)
Assistant Professor
Department of Electrical and Computer Engineering
North South University,
Dhaka, Bangladesh.

Department Chair's Signature

Dr. Kazi Mohammad Abdus Salam
Professor & Chair
Department of Electrical and Computer Engineering
North South University,
Dhaka, Bangladesh.

ACKNOWLEDGMENT

First of all, we wish to express our gratitude to the Almighty for giving us the strength to perform our responsibilities and complete the report. The Senior design project program is very helpful to bridge the gap between the theoretical knowledge and real-life experience as part of Bachelor of Science (BSc) program. This report has been designed to have a practical experience through the theoretical understanding. We also acknowledge our profound sense of gratitude to all the teachers who have been instrumental for providing us the technical knowledge and moral support to complete the project with full understanding. It is imperative to show our appreciation for our honorable faculty member Dr. Mohammad Monirujjman Khan **(**for his undivided attention and help to achieve this milestone. Also, our gratefulness is divine to the North South University, ECE department for providing us a course such as EEE 499 in which we could really work on this project and materialize it the way we have dreamt of. We thank our friends and family for their moral support to carve out this project and always offer their support. We would also thank “Tech Shop BD” and “Robiul electronics”, for providing the necessary support and advising us during the development of this project.

ABSTRACT

This project is developed to eliminate the using of the bare hands for handling any kind of risky item which are harmful for our own hand. For some time, we have been interested in making some sort of robot based on the Arduino platform. This project is the first phase of this longer-term desired effort. Anything is possible with the mighty power of Arduino. It's compact, it's straightforward, and makes embedding electronics into the world-at-large fun and easy. Animatronics is the use of mechatronics to create machines which seem animate rather than robotic. Animatronic figures are most often powered by pneumatics (compressed air), and, in special instances, hydraulics (pressurized oil), or by electrical means. The figures are precisely customized with the exact dimensions and proportions of living creatures. Motion actuators are often used to imitate "muscle" movements, such as limbs to create realistic motions. Also, the figure is covered with body shells and flexible skins made of hard and soft plastic materials. Then, the figure can be finished by adding details like colors, hair and feathers and other components to make the figure more realistic. The project idea came to us after watching the movie named "Real Steel". We wanted to make a shadow robot from our curiosity. As the whole body of the robot would have been of much cost, we decided to make a shadow hand instead. Approximating the kinematics of the human hand was our top priority when developing this animatronic hand. Each joint of this hand has a movement range again the same as or very close to that of a human hand, including the thumb and even the flex of the palm for the little finger.

Keywords- Arduino, Servo motors, Flex sensors, Power IC, Programming of the Arduino.

Table of Content

List of Figures

List of Tables

CHAPTER 1

OVERVIEW

1.1 Introduction

This report mainly analysis about different topologies and designs regarding the construction of this Arduino based animatronic hand. Although more complicated and precise (more expensive) versions of this concept have been developed, this is a fun project with many potential applications. Interactive robot control of this level, I think, has many uses in industrial manufacturing, medical research, and anything you want to be able to do with precision that is unsafe to touch. The basic components of the hand and glove are the hand itself, the servos, the Arduino, the glove, and the flex sensors. The glove is mounted with flex sensors: variable resistors that change their value when bent. They're attached to one side of a voltage divider with resistors of a constant value on the other side. The Arduino reads the voltage change when the sensors are bent, and triggers the servos to move a proportional amount. The servo pull strings that act as tendons, allowing the fingers to move.

1.2 Project Details

There are Three user modules in the Animatronic Hand Controller system:

1. **Arduino UNO**

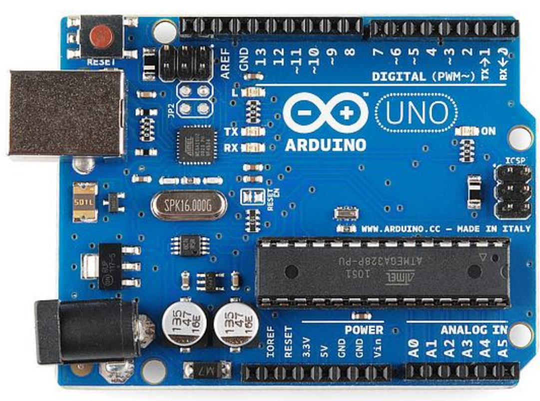


Figure 1: Arduino Uno

Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header & 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. We can tinker with the UNO without worrying too much about doing something wrong. Worst case scenario, we can easily replace the chip for a few dollars & start over again.

Specifications:

Microcontroller ATmega328

Operating Voltage 5V

Input Voltage (recommended) 7-12V

Input Voltage (limits) 6-20V

Digital I/O Pins 14 (of which 6 provide PWM output)

Analog Input Pins 6

DC Current per I/O Pin40 mA

DC Current for 3.3V Pin50 mA

Flash Memory 32 KB (ATmega328) of which 0.5 KB used by bootloader

SRAM 2 KB (ATmega328)

EEPROM 1 KB (ATmega328)

Clock Speed 16 MHz

2. Yourduino

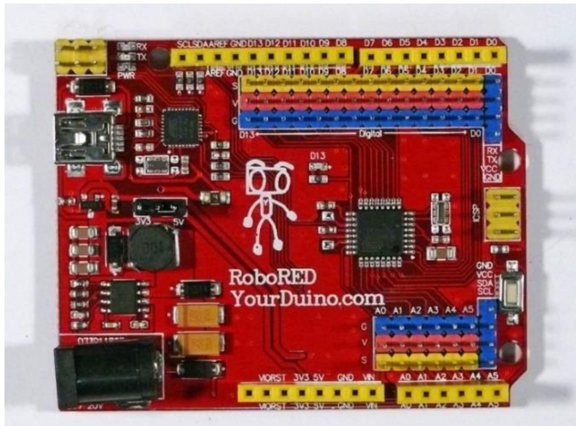


Figure 2: Yourduino

Logically and software equivalent to the Arduino UNO, with many important improvements ALL Digital and Analog I/O pins are provided on 3-pin connectors so you have Ground-Voltage-Signal pins together for Every signal.PWM: 3, 5, 6, 9, 10, and 11. Same as Arduino UNO. Provide 8-bit PWM output with the analog Write () function Dual operating Voltages: The processor chip can be run at 5.0 or 3.3 V by flipping a small switch. This can accommodate newer peripheral chips that can only operate at 3.3V signal levels. Advanced Power for your designs with new onboard 5.0V and 3.3V power supplies. Earlier Arduino versions had only 50 mA of current available at 3.3V but now Rebred provides up to 500 mA at 3.3V An advanced 5.0V switch-mode power supply onboard the RoboRED provides MUCH more 5V power than the original: up to 2 Amps (2000 mA) to power many servos or powerful LEDs, etc. And the Input Voltage rating is higher at 7-23V. (EXTERNAL POWER OF 9 to 20V REQUIRED FOR 2A CURRENT). Repositioned LEDs and Reset button closer to board edges. New 4-pin communications pin groups for Serial: (Receive/Transmit/Voltage/Ground) and I2C (Ground/Voltage/SDA/SCL) make it easy to connect to devices such as LCD displays, GPS, Realtime Clock, etc. Better Analog noise levels: Careful layout of Analog input circuits reduces interference from I2C signals on A4 and A5.

3. Flex Sensors

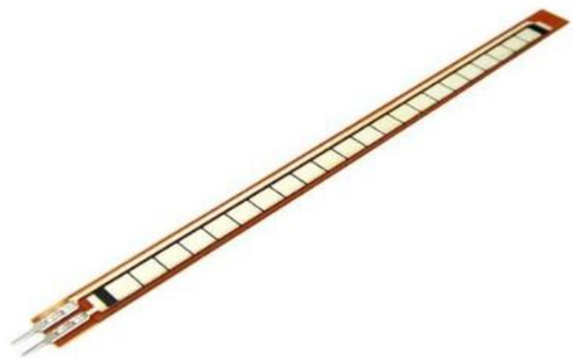


Figure 3: Flex Sensor

This sensor can detect bending in one direction. They were popularized by being used in the Nintendo Power Glove as a gaming interface. These sensors are easy to use, they are basically resistors that change value based on how much their flexed. If they're unflexed, the resistance is about ~10K. When flexed all the way the resistance rises to ~20K. You can use an analog input on a microcontroller (with a pull-up resistor) or a digital input with the use of a 0.1uF capacitor for RC timing. The bottom part of the sensor (where the pins are crimped on) is very delicate so make sure to have strain relief – such as clamping or gluing that part so as not to rip out the contacts.

4. NRF Module

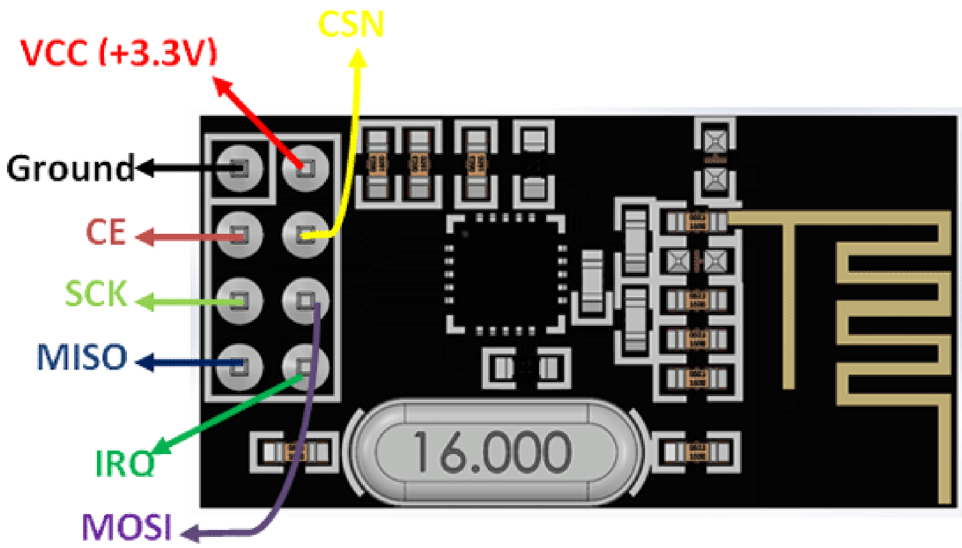


Figure 4: NRF Module

The n**RF24L01 is a wireless transceiver module**, meaning each module can both send as well as receive data. They operate in the frequency of 2.4GHz, which falls under the ISM band and hence it is legal to use in almost all countries for engineering applications. The modules when operated efficiently can cover a distance of 100 meters (200 feet) which makes it a great choice for all wireless remote-controlled projects. The module operates at 3.3V hence can be easily used with 3.2V systems or 5V systems. Each module has an address range of 125 and each module can communicate with 6 other modules hence it is possible to have multiple wireless units communicating with each other in a particular area. Hence mesh networks or other types of networks are possible using this module. So, if you are looking for a wireless module with the above properties then this module would be an ideal choice for you.

1.3 Project Goals

There are several major goals that we want to focus on with our project. They are as follows:

Goals:

Automatization is the feat of the modern civilization. For some years now, the key goal is to replace the need of human help with robots that have lines of codes as their brain. But even with the advent of new technology and sophisticated programming language, there are some tasks still too complicated for droids. These tasks need the fine touch of a human to execute an operation precisely and with intricate details. Animatronics is using electronic machines or robots which can animate or mimic human gestures. Here we are presenting an animatronic hand which accurately resembles a human hand and can mimic the gestures by a human.

Features:

From a bird's eye view, it may seem that our project is pretty straight forward. At first, we have implemented most used tools in a new way. This implementation has given our project versatility & cost effectiveness both. Almost all the animatronic robots are costly because of using some costly tools such as "Flex sensors". But this tool increases the cost of the project about two times. That's where we come in with our unique feature. We are using NRF module which are low in cost & versatile. The NRF module will measure the distance & the servo motors will rotate according to the command which results in moving the fingers of the animatronic hand.

1.4 Summary

This chapter gave us the insight of the modules that we have in the proposed system. This chapter provided a clear picture on how this system is effective to use in home with the help of the elaboration of the project goals.

CHAPTER 2

MOTIVATION

2.1 Introduction

In this chapter we discuss the motivation due to which we thought of implementing this system. We will also discuss in this chapter as to why we have chosen the home automation field apart from all other fields to work it.

2.2 Motivation towards our project

Our dream is to build an animatronic hand controller system that helps you to make existing system easy, more efficient and reliable. Our system will help you to understand the automation system and will help to control it more rigorously. To find the answer to the question at hand, we did a lot of research, especially using the IEEE papers which were based on Animatronic hand controller. In the past, a lot of people have worked with related topics and have come up with different solutions. In the process, they have also used multiple platforms. After research we obtained adequate information which helped us develop a system which would be far more efficient, economic, reliable and user-oriented than the rest.

The system that we have constructed is based on Arduino-Wireless technology. It is controlled by wearing a glove in our own hand and by putting the animatronic hand in a moving arm we can control that animatronics hand by moving our hand wearing the gloves accordingly with our own inclination. There is a NRF module for transmitting and receiving signal between the gloves. Further comparisons, advantages and disadvantages are discussed in the later sections.

2.3 Summary

This chapter provided the idea about the motivation towards our project which aims to eliminates the using of our own hand for any kind of tasks which are harmful for our own hand and keeps the safety in the first priority.

CHAPTER 3

RELATED WORK

3.1 Introduction

In this chapter we discuss the types of animatronic hand controller systems that currently exist in the market. We also focus on other system that the current system entails and a proper justification will be provided as to why our system is the ideal one in the current circumstances.

3.2 Systems related to our project

The list of systems has been discussed thoroughly including the drawbacks that they possess. The most common systems that are currently available in the market are as follows.

1. **UTAH/MIT Hand**

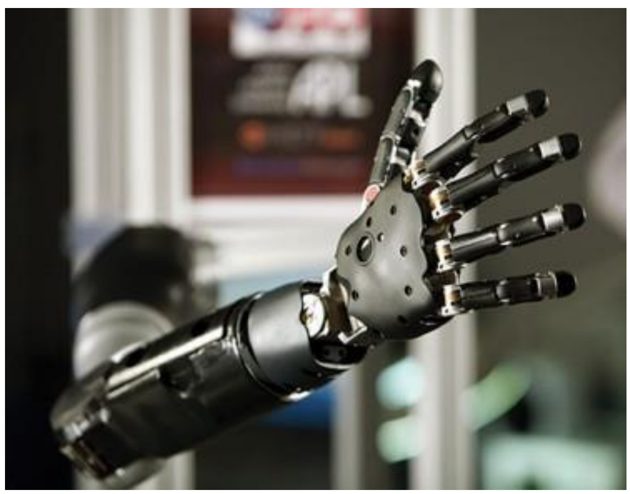


Figure 5: Utah/MIT hand

The Utah/MIT hand shown in figure was developed by the Center for Engineering Design at the University of Utah and the Artificial Intelligence Laboratory at the Massachusetts Institute of Technology (MIT) in 1985. It was intended to function as a general-purpose research tool for the study of machine dexterity (Jacobsen, etc., 1986). The Utah/MIT hand has the same size as the human's hand. It has four fingers (three fingers and a thumb) in a very an-thropomorphic configuration. Each finger has four degrees of freedom and can move at five times of human speed, including the grip of a firm handshake. There are totally 17 links (1 on the wrist, 4 on each of the 4 fingers) on the Utah/MIT hand. Among them, 16 joints on the fingers have degrees of freedom. Their bending and extension are controlled by cable driven by pneumatic pistons, which is integrated into the hand. By utilizing the finger tendon forces the grasp can react, to some degree, to the object being grasped. On the Utah/MIT hand, there are antagonistic tendons for each finger joint. The antagonistic tendons and the large amount of coupling between finger joints have complicated the work on the Utah/MIT hand (Jacobsen et al 1986). There are 4 kinds of sensors on the hand, motor position sensors, joint position sensors, tendon tension sensors and tactile array sensors. According to the signals detected, movement of the hand can be illustrated.

2. **NASA Hand (Robonaut Hand):**

Figure 6: NASA HAND

The NASA hand was designed in 1999 by C.S. Lovchik in Robot-ics Technology Branch of NASA Johnson Space Center and M.A. Diftler in Automation and Robotics Department of Lockheed Martin Corporation. It was developed for space extravehicular activity (EVA) use. It is close in size and capability to a suited astronaut’s hand (Lovchik 1999). This five-finger hand combined with its inte-grated wrist and forearm has fourteen independent degrees of freedom. It consists of a forearm which houses the motors and drive electronics, a two degree of freedom wrist, and a five finger, twelve degree of freedom hand. The forearm, measures four inch-es in diameter at its base and is approximately eight inches long. It houses all fourteen motors, 12 separate circuit boards, and all of the wiring for the hand. The dexterous finger set consists of two 3 degree of freedom fingers (pointer and index) and a 3 degree of freedom opposable thumb. The grasping set consists of two, 1 degree of freedom fingers (ring and pinkie) and a palm degree of freedom. All of the fingers are shock mounted into the palm. Over-all the hand is equipped with forty-three sensors not including tactile sensing. Each joint is equipped with embedded absolute position sensors and each motor is equipped with incremental encoders (Lovchik 1999). The Utah/MIT hand has antagonistic tendons for each finger joint on the hand, but on NASA hand there is just one tendon sensor for each finger. This reduces the amount of coupling between finger joints which complicated the Utah/MIT hand. Figure 2 shows the working NASA hand.

3. GIFU Hand

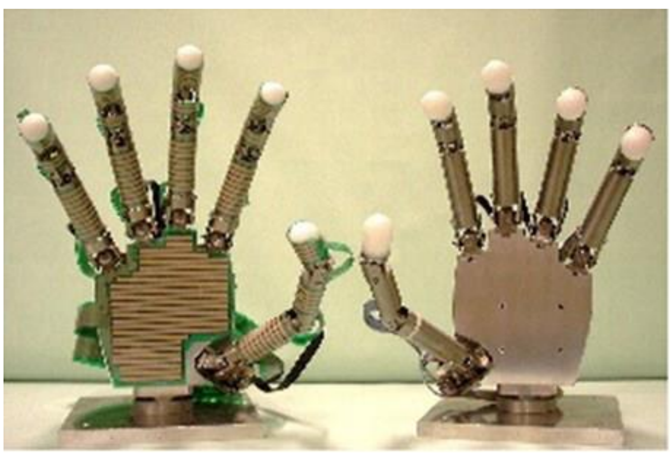


Figure 7: Gifu Hand

The Gifu hand was designed in 2001 by Jacobsen and Kawasaki at Gifu University and is highly anthropomorphic with the total size of thumb, four modular fingers, and palm being only slightly larger than the human hand. Each of the fingers has four joints with the thumb providing four

DOFs and each finger providing three DOFs. With a bandwidth greater than that of the human hand, the Gifu hand provides an excellent test bed for controls research.

4. **Animatronic Hand:**

Figure 8: Animatronic Hand

In the field of robotics and industrial equipment's, use of proper method of handling the chemicals. equipment's etc is very important from safety point of view. Traditional method of handling the chemicals as well as equipment's was done normally with the help of bare hands. With the bare arms it is not possible to hold the equipment's and chemicals for a large amount of time and due to this there is a matter of safety as well as there is no proper precision. With the new innovations in the field of science i.e. Animatronic hand, we can do all such tough tasks with higher amount of precision and higher amount of safety. So in this project, electronic hand is developed which is controlled via control glove. The control glove has flex sensors which detects the movements of fingers and accordingly electronic hand perform the required tasks. This animatronic hand is able to be controlled according to the controller's . It is capable of moving at the required degrees of freedom. It ca also pick up things upto to minimum desired weight. We can use it as shadow hand of ours which is of various use. With the help of sensors this hand can provide detailed telemetry, which can be exploited to generate innovative manipulation, control systems, or to provide detailed understanding of the external environment.

3.3Problems with the current systems

The existing system could never reach its pinnacle in terms of its efficiency as the existing system requires huge amounts of budget which is difficult to manage for every category people. Furthermore, whenever there is a need for particular information to be retrieved it becomes really difficult to find out at a given point of time. Moreover, the existing system are more advanced so it is no much user-friendly system.

3.4 Proposed Solution

Traditional method of handling the chemicals as well as equipment's was done normally with the help of bare hands. With the bear arms it is not possible to hold the equipment's and chemicals for a large amount of time and due to this there is a matter of safety as well as there is no proper precision. With the new innovations in the field of science i.e. Animatronic hand, we can do all such tough tasks with higher amount of precision and higher amount of safety. There is a control glove which has a flex sensor mounted on it. The flex sensors are the special sensors which calculate the change in resistance when they bend. When control glove is bending the flex, sensor changes its value. This value is an analogue value which is given to microcontroller. The microcontroller con-verts the input analogue signal into digital signal. This digital value is transmitted wirelessly through RF transmitter. On the receiver side this digital signal is received by the RF receiver which is then given to microcontroller connected to servos on animatronic hand and the animatronic hand functions just as the movement of the control glove.

3.5 Summary

Considering all aspects, our proposed solution is an optimal one. The Animatronic Hand Controller system is an open door to get a hand automation system within a reasonable cost. It is reasonable because we are using numerous custom designed and self-programmed frameworks that are going to help us to develop our project. The system interface is easy to operate and can be maintained by a non-IT professional as well. This system has been developed in such a way that it gives a feeling of user friendliness and there will not arise any issues with regards to complications. Therefore, this chapter gives the idea about the current systems that are available in the market and about the motivation towards developing this dynamic system.

CHAPTER 4

TECHNICAL DESIGN

4.1 Introduction

In this chapter we discuss the aspect of the technical design of our system. By going through the system level design, it would be easier to conceptualize the entire data flow of the system.

4.2 Technical Design: System Level

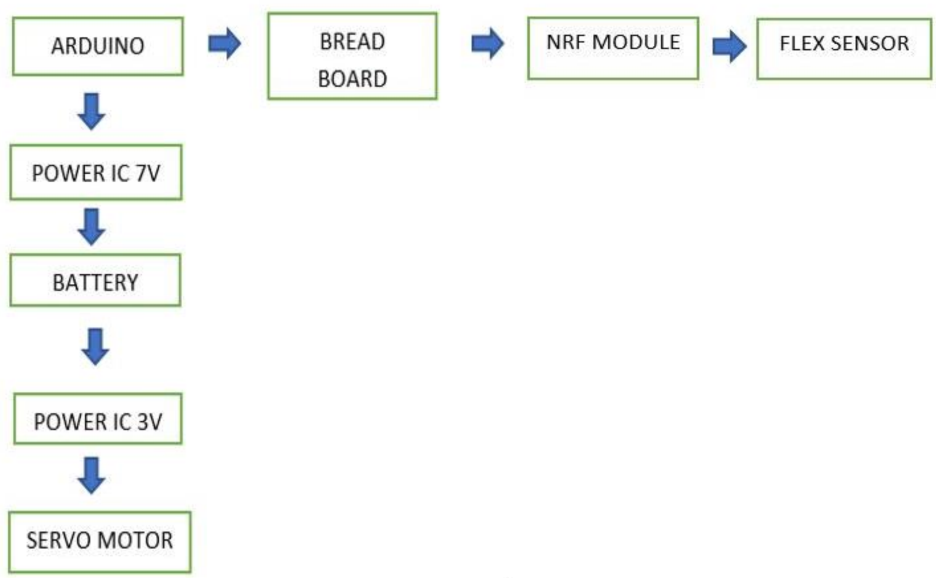


Figure 9: Block Diagram

In figure, Block diagram give data to the system and get data output from the system. NRF module is used here for transmitting and receiving the signal between the sensors. Arduino microcontroller processes the signal. Servo motors moves accordingly with the signal from the flex sensors.

Arduino:

Arduino is programmed using the Arduino IDE which can be downloaded from the internet. Arduino carries out instructions according to the code that is uploaded in it. When the ASCII code of ‘a’ for example transmitted from the Bluetooth module, matches with the code, Arduino executes the command. If ‘a’ is coded as HIGH, Arduino makes the output HIGH, if it’s LOW, the output is low. The multiple outputs of the Arduino are connected to the multiple components. The AC components are connected via Relays.

Flex Sensor:

This sensor can detect bending in one direction. They were popularized by being used in the Nintendo Power Glove as a gaming interface. These sensors are easy to use, they are basically resistors that change value based on how much their flexed. If they’re unflexed, the resistance is about ~10K. When flexed all the way the resistance rises to ~20K. You can use an analog input on a microcontroller (with a pull-up resistor) or a digital input with the use of a 0.1uF capacitor for RC timing. The bottom part of the sensor (where the pins are crimped on) is very delicate so make sure to have strain relief – such as clamping or gluing that part so as not to rip out the contacts.

NRF Module

The n**RF24L01 is a wireless transceiver module**, meaning each module can both send as well as receive data. They operate in the frequency of 2.4GHz, which falls under the ISM band and hence it is legal to use in almost all countries for engineering applications. The modules when operated efficiently can cover a distance of 100 meters (200 feet) which makes it a great choice for all wireless remote-controlled projects. The module operates at 3.3V hence can be easily used with 3.2V systems or 5V systems. Each module has an address range of 125 and each module can communicate with 6 other modules hence it is possible to have multiple wireless units communicating with each other in a particular area. Hence mesh networks or other types of networks are possible using this module. So, if you are looking for a wireless module with the above properties then this module would be an ideal choice for you.

Servo Motor

Unlike dc motors, with servo motors you can position the motor shaft at a specific position (angle) using control signal. The motor shaft will hold at this position as long as the control signal not changed. This is very useful for controlling robot arms, unmanned airplanes control surface or any object that you want it to move at certain angle and stay at its new position. Servo motors may be classified according to size or torque that it can withstand into mini, standard and giant servos. Usually mini and standard size servo motors can be powered by Arduino directly with no need to external power supply or driver.

4.3 Summary

As mentioned earlier, the technical design has enabled us to get a clear picture of how our system is operating. Therefore, considering the above block diagram we can comprehend the method in which our system is being operated.

CHAPTER 5

SOFTWARE DESIGN

5.1 Introduction

In this chapter we discuss the software design of our system. One can consider this part as the essential structure of the system as it will provide as the necessary features that our system entails.

5.2 Software Design: Module Level

Figure 10: Software Design

1.

Connected our Arduino to the computer with the USB cable. We did not need the battery for that time. The green PWR LED will light. If there was already a program burned into the Arduino, it will run.

2.

Started the Arduino development environment. In Arduino-speak, programs are called “sketches”, but here we will just call them programs. Our window would look something like above figure.

5.3 Summary

This chapter provided all the necessary details on the software design framework of our system.

CHAPTER 6

SYSTEM FRAMEWORK

6.1 Introduction

In this chapter we explore the system in a real time manner through pictorial illustrations which will further help us to visualize the system before getting hold on to it physically. This chapter entails the step by step process on how the system is operated. Through the help of the visual representation it will enable the reader to get a clearer picture of the **“Animatronic Hand Controller”** system.

6.2 Steps by Step System Design Framework

1.Working Principal of Arduino

The Arduino microcontroller is an easy to use yet powerful single board computer that has gained considerable traction in the hobby and professional market. The Duemilanove board features an Atmel ATmega328 microcontroller operating at 5 V with 2 Kb of RAM, 32 Kb of flash memory for storing programs and 1 Kb of EEPROM for storing Parameters. The clock speed is 16 MHz, which translates to about executing about 300,000 lines of C source code per second. The board has 14 digital I/O pins and 6 analog input pins. There is a USB connector for talking to the host computer and a DC power jack for connecting an external 6-20 V power source, for example a 11.1 V battery, when running a program while not connected to the host computer. Headers are provided for interfacing to the I/O pins using 22 g solid wire or header connectors. The Arduino programming language is a simplified version of C/C++. If you know C, programming the Arduino will be familiar. If you do not know C, no need to worry as only a few commands are needed to perform useful functions. An important feature of the Arduino is that we can create a control program on the host PC, download it to the Arduino and it will run automatically. Remove the USB cable connection to the PC, and the program will still run from the top each time you push the reset button. Remove the battery and put the Arduino board in a closet for six months. When you reconnect the battery, the last program you stored will run. This means that you connect the board to the host PC to develop and debug your program, but once that is done, you no longer need the PC to run the program.

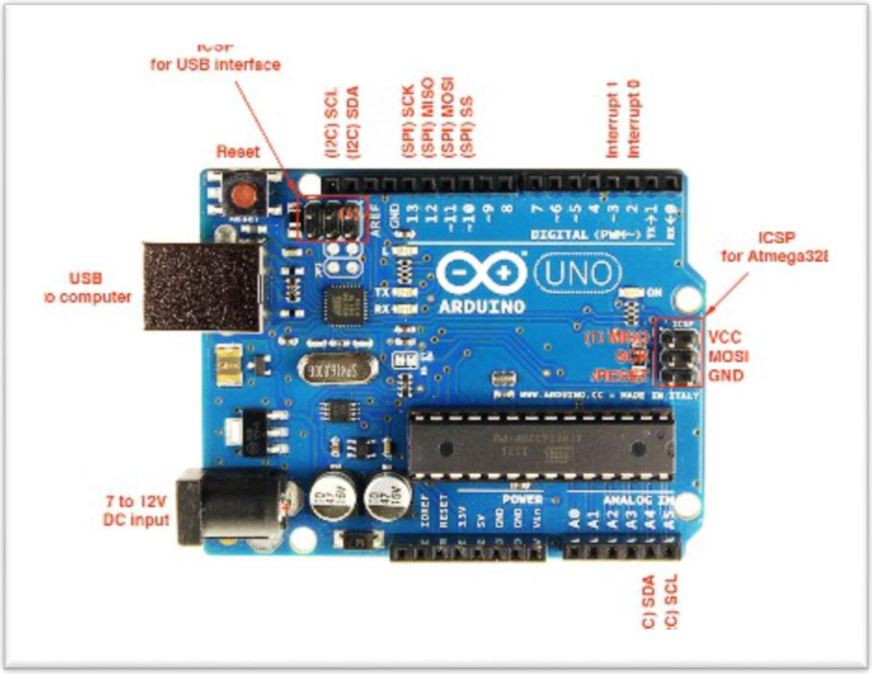


Figure 11: Arduino Hardware

The power of the Arduino is not its ability to crunch code, but rather its ability to interact with the outside world through its input-output (I/O) pins. The Arduino has 14 digital I/O pins labeled 0 to 13 that can be used to turn motors and lights on and off and read the state of switches. Each digital pin can sink or source about 40 mA of current. This is more than adequate for interfacing to most devices, but does mean that interface circuits are needed to control devices other than simple LED's. In other words, you cannot run a motor directly using the current available from an Arduino pin, but rather must have the pin drive an interface circuit that in turn drives the motor. A later section of this document shows how to interface to a small motor. To interact with the outside world, the program sets digital pins to a high or low value using C code instructions, which corresponds to +5 V or 0 V at the pin. The pin is connected to external interface electronics and then to the device being switched on and off. The sequence of events is shown in this figure.

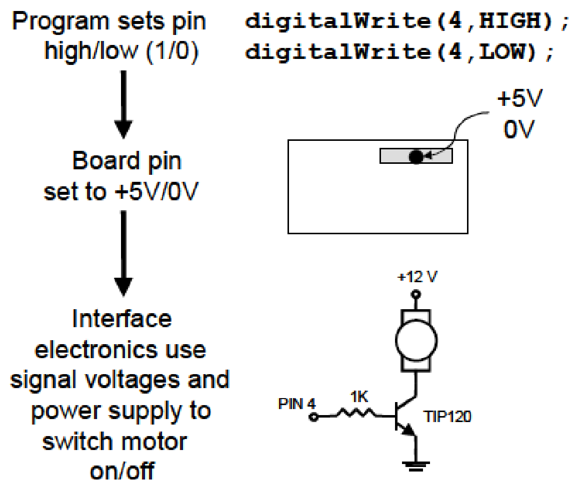


Figure 12: Configuration of Arduino Hardware

To determine the state of switches and other sensors, the Arduino is able to read the voltage value applied to its pins as a binary number. The interface circuitry translates the sensor signal into a 0 or +5 V signal applied to the digital I/O pin. Through a program command, the Ardiomp interrogates the state of the pin. If the pin is at 0 V, the program will read it as a 0 or LOW. If it is at +5 V, the program will read it as a 1 or HIGH. If more than +5 V is applied, you may blow out your board, so be careful. The sequence of events to read a pin is shown in this figure:

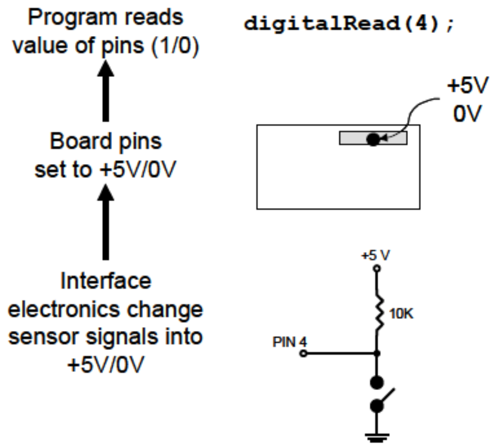


Figure 13: Configuration of Arduino Hardware

Interacting with the world has two sides. First, the designer must create electronic interface circuits that allow motors and other devices to be controlled by a low (1-10 mA) current signal that switches between 0 and 5 V, and other circuits that convert sensor readings into a switched 0 or 5 V signal. Second, the designer must write a program using the set of Arduino commands that set and read the I/O pins. When reading inputs, pins must have either 0 or 5V applied. If a pin is left open or "floating", it will read random voltages and cause erratic results. This is why switches always have a 10K pull up resistor connected when interfacing to an Arduino pin. The reason to avoid using pins 0 and 1 is because those pins are used for the serial communications between the Arduino and the host computer. The Arduino also has six analog input pins for reading continuous voltages in the range of 0 to 5 V from sensors such as potentiometers.

2. WORKING PRINCIPAL OF SERVO MOTOR

Unlike dc motors, with servo motors you can position the motor shaft at a specific position (angle) using control signal. The motor shaft will hold at this position as long as the control signal not changed. This is very useful for controlling robot arms, unmanned airplanes control surface or any object that you want it to move at certain angle and stay at its new position.

Servo motors may be classified according to size or torque that it can withstand into mini, standard and giant servos. Usually mini and standard size servo motors can be powered by Arduino directly with no need to external power supply or driver.

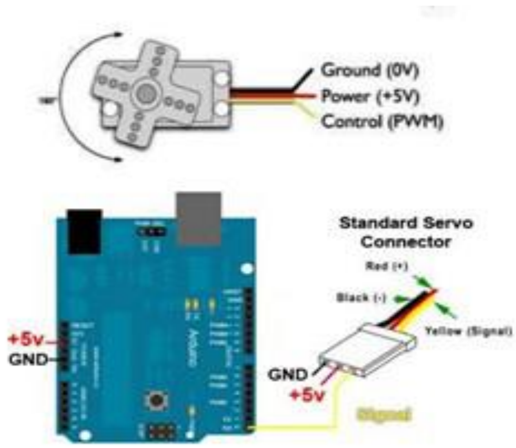


Figure 14: Servo Motor Connection

The third pin accept the control signal which is a pulse-width modulation (PWM) signal. It can be easily produced by all micro- controllers and Arduino board. This accepts the signal from your controller that tells it what angle to turn to. The control signal is fairly simple compared to that of a stepper motor. It is just a pulse of varying lengths. The length of the pulse corresponds to the angle the motor turns to. The pulse width sent to servo ranges as follows: **Minimum:** 1 millisecond ---> Corresponds to 0 rotation angle. **Maximum:** 2 millisecond ---> Corresponds to 180 rotation angle. Any length of pulse in between will rotate the servo shaft to its corresponding angle. For example : 1.5 ms pulse corresponds to rotation angle of 90 degree. This is will explained in figure below:

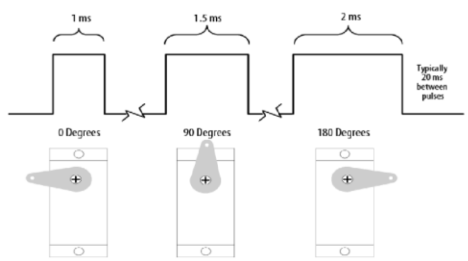


Figure 15: Rotating Configuration of Servo

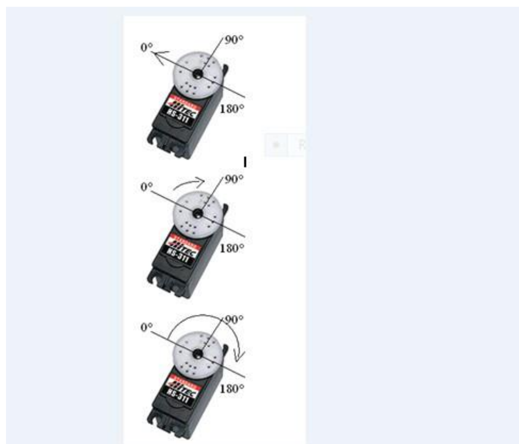


Figure 16: Rotational Configuration of Servo

3.WORKING PRINCIPAL OF FLEX SENSOR

Flex sensors are sensors that change in resistance depending how much the sensor is bend. Sensors convert the change in bend to electrical resistance - the more the sensor bend, the higher the resistance value. Using the Flex Sensor is very easy. There are couple of different manufacturers in the market. Datasheet instructs you to use operational amplifier (opamps). That may be useful if you plan to use flex sensor as stand-alone device (without any microcontroller).Because We are using arduino, We skipped all OpAmps and made a very simple circuit with only one additional resistor.

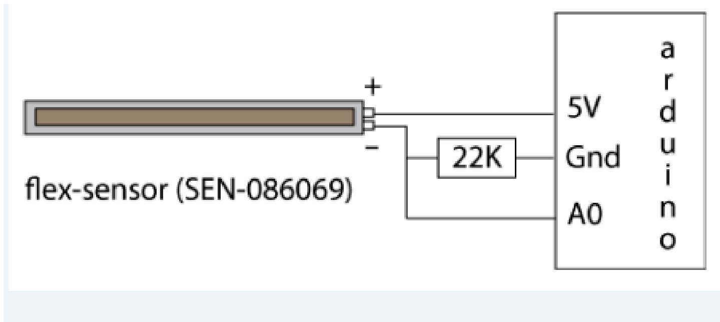


Figure 17: Sample Flex Sensor Circuit

Varying the value of the resistor will results different readings. With 22k Ohm resistor I will get values between 300-700. This works fine for us. In our code we assumed that all values under 400 mean that the sensor is bend. All values above 600 mean that sensor is nor bend. Note that Flex sensor give reliable readings ONLY if you bend it on the specific direction (usually towards on the text side of the sensor).

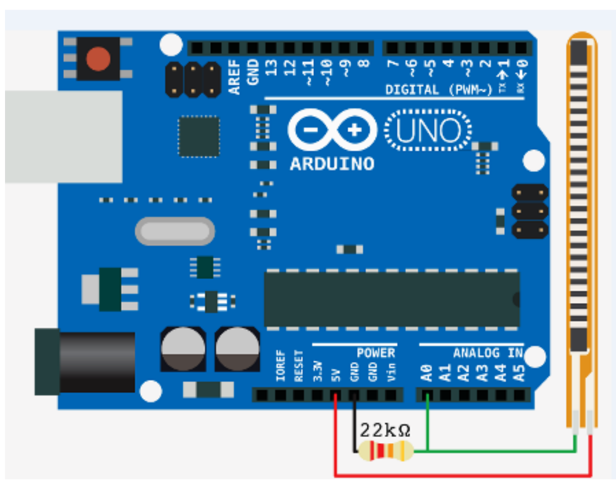


Figure 18: Flex Sensor Connection with Arduino

5. Motor Control Using Arduino

A. **OPERATING ONE SERVO WITH ARDUINO: Standard servo motor control using Arduino is extremely easy. This is because t**he Arduino software comes with a sample servo sketch and servo library that will get you up and running quickly: 1. Connect the black wire from the servo to the Gnd pin on the Arduino 2. Connect the red wire from the servo to the +5V pin on the Arduino 3. Connect the third wire (usually orange or yellow) from the servo to a digital pin on the Arduino

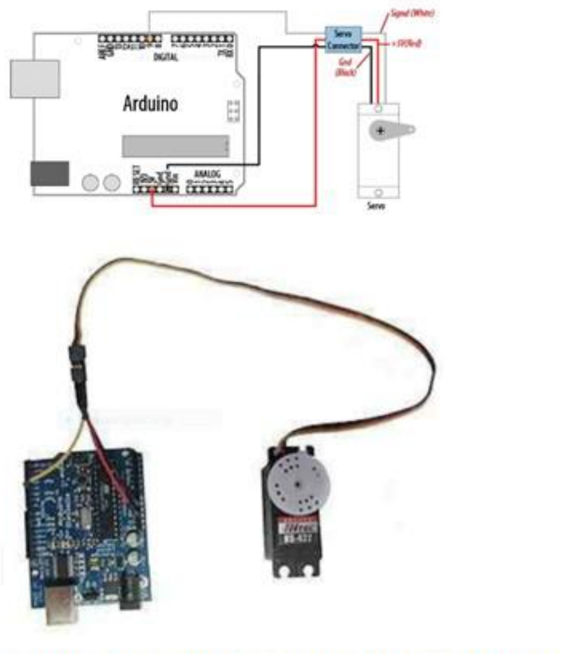


Figure 19: Motor Control Using Arduino

 # Important Notes: 1- It is not a good idea to connect a motor of any kind directly to the Arduino because it usually requires more power than the board can provide. 2- In our example, the servo is being used to demonstrate code and is not encountering any resistance. Note that you should use a standard or small size. if you are uncertain, check the servo's no load current rating (it should usually be under 150mA). 3-You may need an external source of 5 or 6 volts when connecting multiple servos. Four AA cells work well if you want to use battery power. **Remember that you must connect the ground of the external power source to Arduino ground.**

B. OPERATING SERVO WITH ARDUINO:

The Arduino can control two servos with the same ease as one. All it takes is creating a second instance (copy) of the Servo object, giving it a unique name. In this project we had to use six servos, five of which has to connect directly with arduino at the same process of connecting one servo with arduino. **When wiring the solderless breadboard, be especially careful not to mix positive and negative leads to the servo**. **Reversing the power will permanently damage it**. In order for everything to function properly, the ground connections for the Arduino and the servo battery supply must be connected together.This is shown in both the schematic and pictorial circuit views. Make sure to also properly orient the connectors for the servos when you plug them into the board. Servo power leads are color-coded, but the colors aren't universal. Here two servos are being connected with the arduino in the figure given below:-

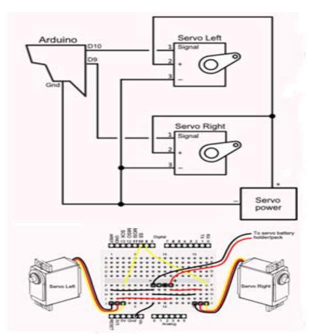


Figure 20: Operating Two Motors Using Arduino

6.3 Summary

This chapter provided all the necessary details on the system framework.

CHAPTER 7

SKILLS

7.1 Introduction

In this chapter we discuss the skills that we have obtained in order to develop this massive sophisticated system.

7.2 Skills obtained

Through this project the following skills have been developed:

Arduino Coding: The most difficult part of the project was the coding of the Arduino. The Arduino had to be coded using simple C language. The required code is attached in the Index section. The coding was done by Arduino IDE.

Design and Implementations of electrical circuit.
Operations of Arduino, NRF module and Flex sensors.

7.3 Summary

In this chapter we discussed the list of skills that have been obtained throughout the process of developing and materializing this system.

CHAPTER 8

IMPLEMENTATION

&

CONSTRUCTION

8.1 Introduction

In this chapter we discuss the construction of the whole system developed.

8.2 Implementation & Construction

1.Set up the Sensor Circuit

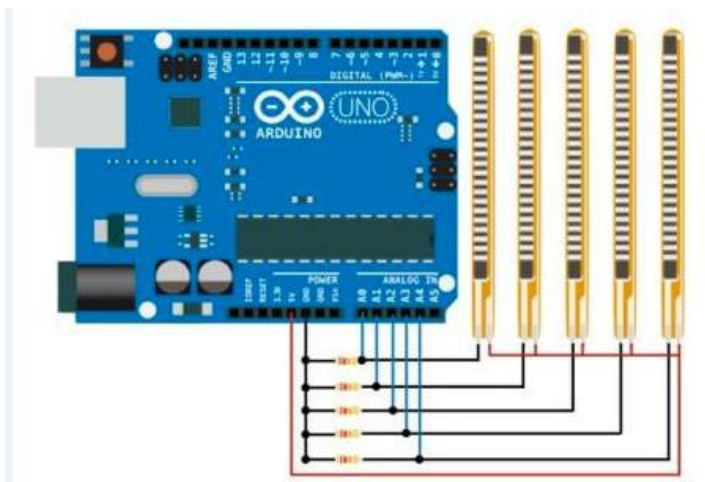


Figure 21: Construction of sensor circuit

The flex sensors require a circuit in order for them to be compatible with Arduino. It's a voltage divider: the flex sensors are variable resistors, and when paired with resistors of a static value, change in resistance (in this case bending the sensor) can be sensed through the change in voltage between the resistors. This can be measured by the Arduino through its analog inputs. The schematic is attached (red is positive voltage, black is negative, and blue goes to the Arduino). The resistors in the photo are 22K. I color-coded the wires we used in the same way as the schematic, so we can see more easily. The main GND wire, which is connected to all the individual GND wires from the sensors, gets plugged into the Arduino's GND. The +5V from the Arduino goes to the main positive voltage wire, and each blue wire gets plugged into a separate analog input pin.

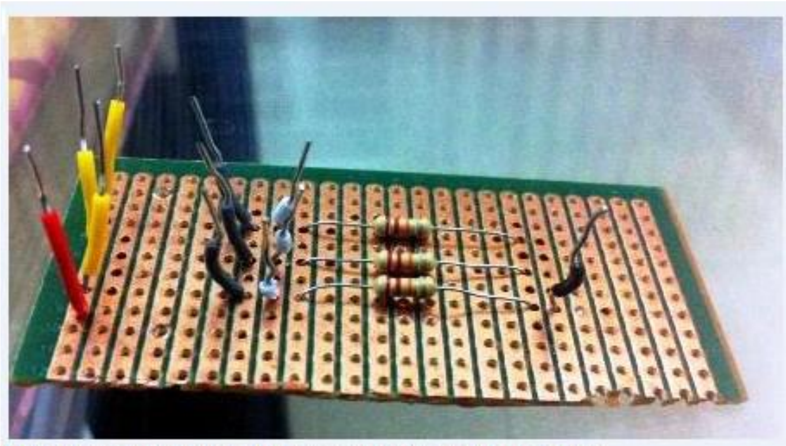


Figure 22: Soldered sensor circuit on PCB

Then we soldered the circuit onto a small PCB.One that could be easily mounted onto the glove.

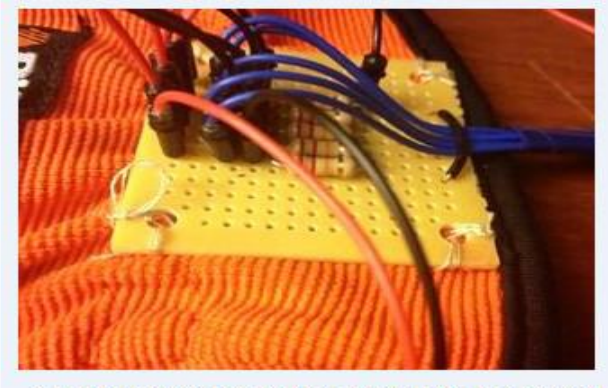


Figure 23: PCB mounted on gloves

We were able to solder the wires to the sensors relatively easily also, and used heat shrink to make sure there were no shorts.

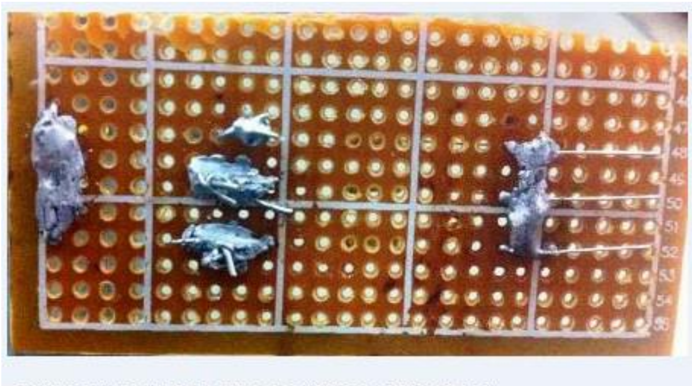


Figure 24: Rear view of PCB

We then wrapped the area where the wires are connected to the sensors with electrical tape to stabilize the sensors.

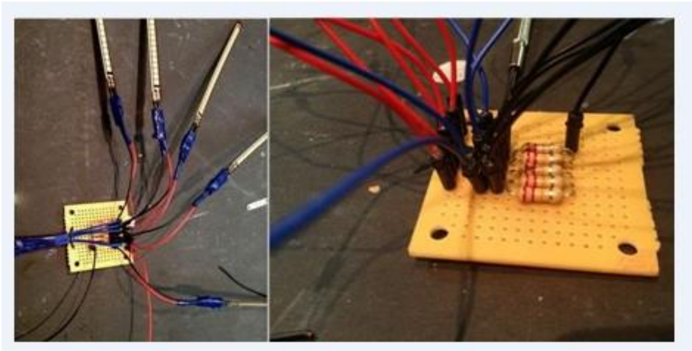


Figure 25: Construction of Sensor circuit

Near the bottom, where the leads are attached, the sensors are a bit weaker and the tape ensures that they won't bend too far and won't get damaged.

2. FLEX SENSOR MOUNTING

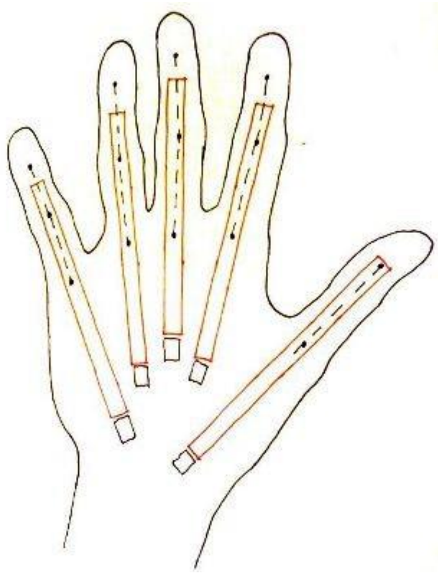


Figure 26: Flex Mounting

Now it's time to mount the sensors and their circuit onto the glove itself. First, We drilled a tiny hole in the plastic of the sensors (at the top, once the resistive material has ended). Be sure not to hit the resistive material! Then, put on the glove and pull it tightly to your hand. On each finger, with a pencil or pen, make small lines over the tops of each joint/knuckle. This will tell you where to sew the sensors. Sew each sensor tip to the area of each finger just above where each of your fingernails would be (use the hole you just drilled). Then, for each sensor, make loose loops around them with thread at both joints in each finger. Once each sensor is in place and slides under the loops of thread nicely.

Figure 27: Flex Sensor mounting on the glove

Then We sewed the PCB onto the wrist part of the glove tightly.

Figure 28: Mounted PCB on the glove

REMEMBER: for each step in this process, be sure We're not sewing the glove itself closed. That's quite a hassle.

3. HAND AND SERVO BED CONSTRUCTION

1. FINGERS ASSEMBLY

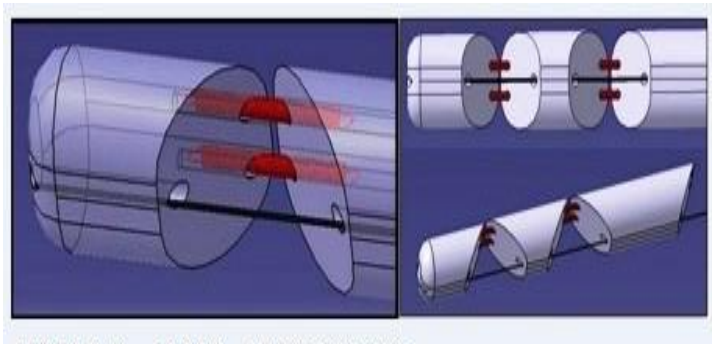


Figure 29: Finger construction

We used plastic hollow pipes to construst the finger parts.When assembling the fingers, we made sure that the parts were oriented correctly before gluing. Also, we made make sure to re-drill the holes on the finger parts so the 3mm screws will act as hinge pins without causing friction.Then we connected the part using a string and screws.

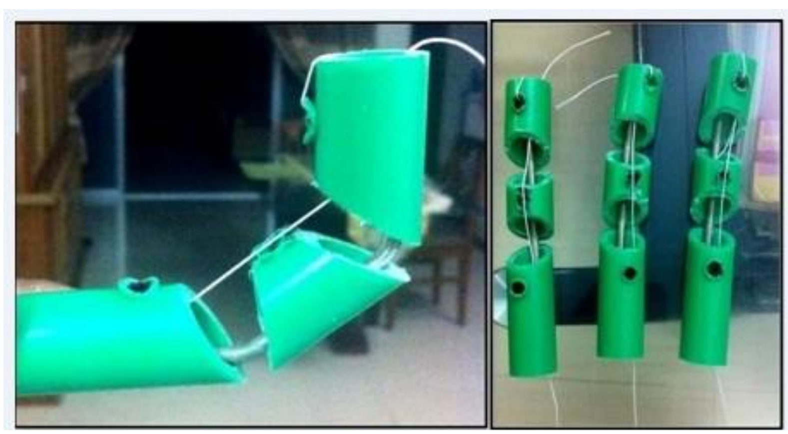


Figure 30: Fingers Assembly

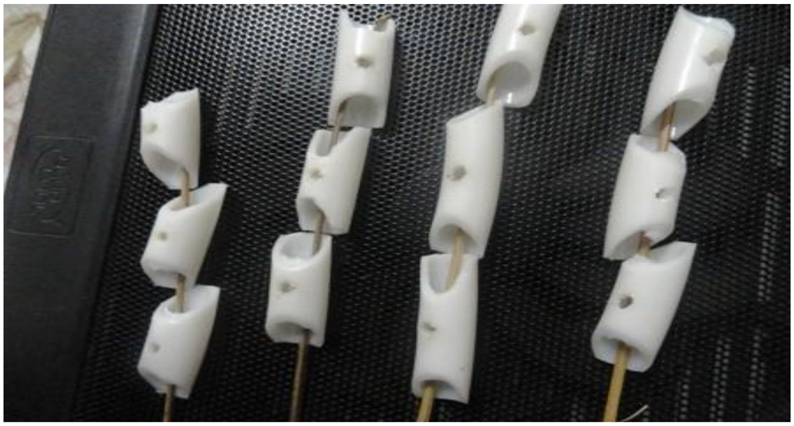


Figure 31: Fingers Assembly

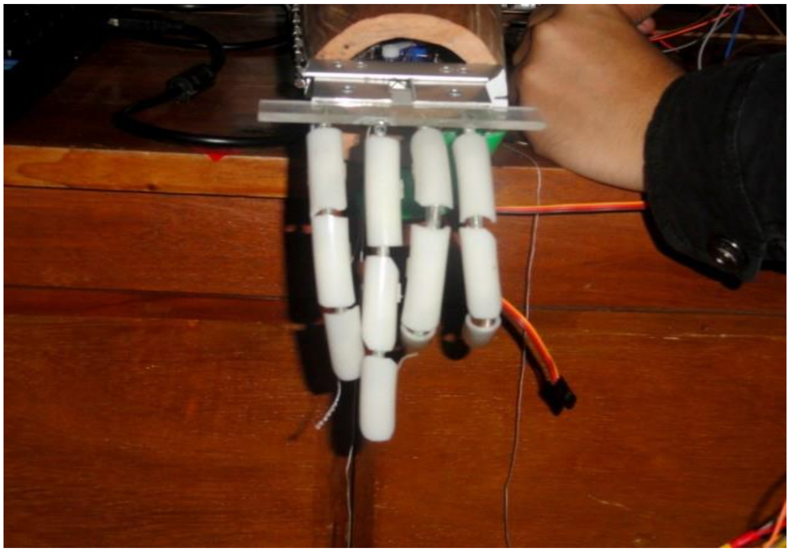


Figure 32: Fingers Assembly

Then we kept the screws in with a dab of hot glue on the outside of the fingers.

2. SERVO BED CONSTRUCTION AND SERVO MOUNTING: -

We have used a wooden piece to make the servo bed. Here in the bed we made 5 servo motors to be held perfectly within the bed. It's just kind of a bed for the servos. Later we have to put the strings to connect the servos with each other. Here is the construction figure shown below: -

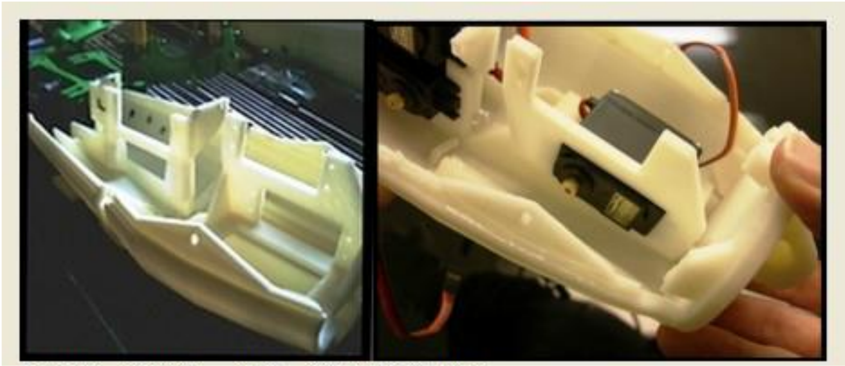


Figure 33: Servo bed construction

Here is the picture of our wooden aervo bed shown below: -

Figure 34: PVC constructed servo bed

We waited before installing the strings into the hand as we wanted to make sure the servos are working first.

3. ADDING THE STRINGS:

Adding the strings is by far the hardest and most tedious part of this project. It's simple in concept, but difficult to actually execute. Threading the fingers takes patience: remember that. The one difference between my installation of the strings and I used hot glue. To me, hot glue is more adjustable when calibrating each finger because it can be easily melted and re-hardened. We connected the servo motor in a way we could move our fingers with the exact comfort and flexibility. For that we calibrated the servo motors to connect the strings with the exact process.

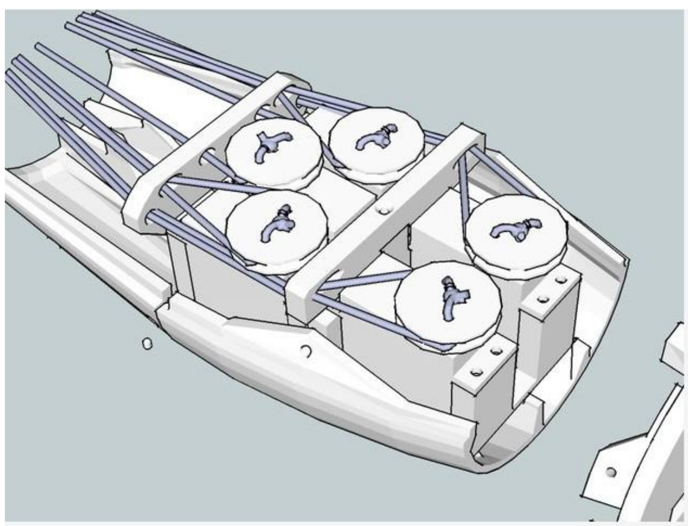


Figure 35: Adding the string with the servo

To calibrate each servo ring so it flexes and relaxes its finger when we wanted it to based on the input, first we plugged in our Arduino and servo battery and run the program. Then we put the glove on and flex the finger that corresponds to the servo we're working on. We adjusted the servo ring so one hole is closest possible to the fingers and pulled the "relax" string of that finger as tightly as we can without bending the finger. We put it through the closest hole of the ring and glue it in place. Then, straightened my finger and pulled and secured the other string into the other hole. Then we repeated this process with each finger. It's important to make each string taut.

4. SETTING UP THE SERVO MOTOR CIRCUIT:

1. SERVO MOTOR WIRING WITH THE ARDUINO:

We connected the 5 servo motors (which had been mounted on the servo-bed) and the another servo (mounted on the hinge portion) with the ARDUINO PWM pins. The PWM input wire (orange) for each servo would be attached to the Arduino pins as follows:

1. SERVO MOTOR WIRING WITH THE ARDUINO:

- Servo One - PWM -> Digital 9,
- Servo Two - PWM -> Digital 8,
- Servo Three - PWM -> Digital 7,
- Servo Four - PWM -> Digital 6,
- Servo Five - PWM -> Digital 5,
- Servo Six - PWM -> Digital 4,

Following is the diagram showing the wiring of 6 servo motors with the Digital (PWM) pins of ARDUINO: -

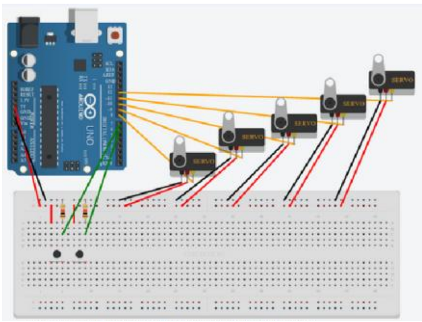


Figure 36: Servo wiring with Arduino

1. **WIRING OF THE SIXTH (WRIST) SERVO MOTOR:**

The servo motor which we have mounted on the wrist for it's movement. Then it's control(orange) wire was first connected with the PWM pin 4.

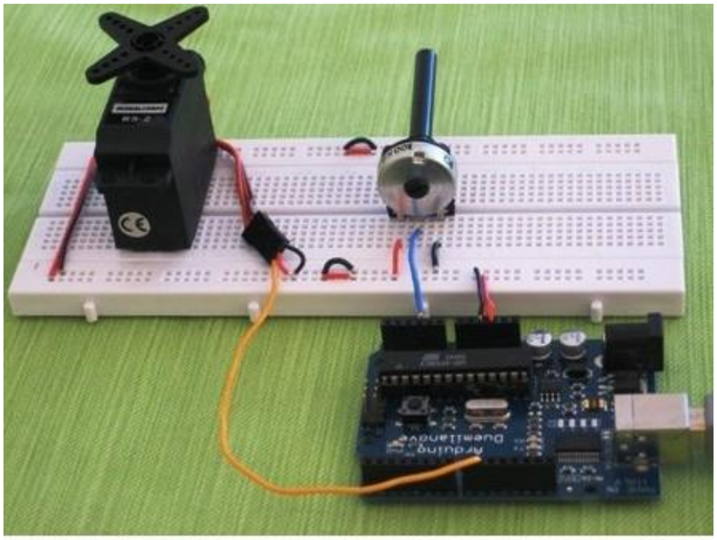


Figure 37: Wiring of the sixth servo

After that we connected its red and black (power) wire of the servo with a small solder-less bread board. We also connected a Potentiometer in series with the power wires. We connected the positive and the ground wire of the POT in series with the power wires of the servo and connected the variable resistive wire of the POT with an analog pin of the ARDUINO.

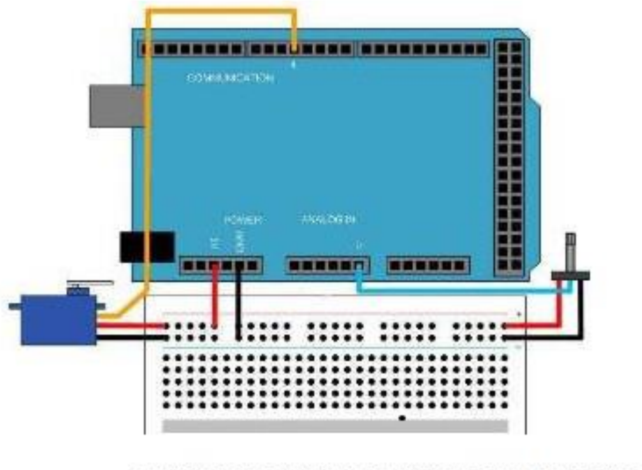


Figure 38: Wiring of the Wrist servo motor

5. THUMP CONSTRUCTION:

We used aluminum plate and a piece of plastic glass for the thump construction. The larger portion of the wrist (Top surface) has been constructed with aluminum plate and has been threaded using threading machine. The rest of the wrist has been constructed using a plastic glass plate with which we assembled the 5 fingers using super glue and screws keeping the comfortable position of the fingers in mind.

5. JOINING THE FINGERS WITH THE THUMP:

We assembled the 5 fingers with the plastic glass made portion of the wrist using super glue and screws. According to the following pictures we tried to join all the fingers keeping comfort area and the top and bottom view of the fingers in mind.

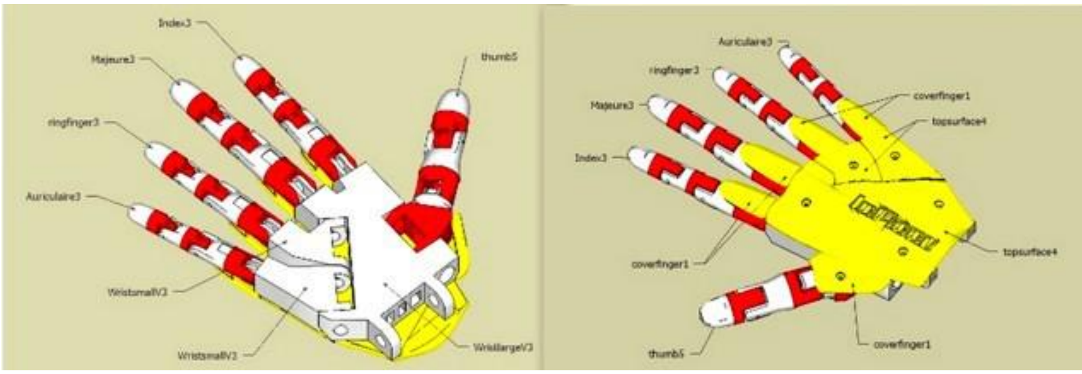


Figure 39: Figure construction

5.BREAD BOARD DETAIL:

We used a bread board here to construct a particular circuit for the operation of 16 k torque servo motor for the control over the total hinge of the hand.

8.3 Summary

In this chapter we discussed the list of skills that have been obtained throughout the process of implementing and constructing this system.

CHAPTER 9

ESSENTIAL PARTS

AND DEVICES

9.1 Introduction

In this chapter, we shed light on the tools that we used to develop this sophisticated system and we also discuss what tools will be required if one wants to test this system in there one spheres.

9.2 Design Requirements

The system needs to have several modules, wires, breadboard, Arduino coding software ,app building software. We used the following tools for our system:

1. **Arduino UNO Module:**

   Arduino in this project acting as a microcontroller. It is the controlling unit of the whole system it is synchronous with other two module also.

2. **Yourduino Module:**

   Bluetooth module in the project is used for transmitting and receiving the signal from the app to the Arduino where the signal will be processed.

3. **NRF Module:**

   Four channel relays are used for controlling the four appliances of the home. Relay plays the role of switching automatedly. Arduino process the signal and send the data to the relay and relay switches accordingly.

4. Software:

Arduino Software

Figure 40: Arduino Control Panel

5. Operating System:

- Android device with version 7.0.8 or higher.
- Windows Pc with windows 7 or higher

9.3 Test Requirements

In order to test the home automation for system elderly and disabled the following tools are necessary:

1. Electrical Connectivity

2. Bluetooth connectivity

3. Hardware:

- Arduino UNO
- Yourduino
- NRF module
- Wires
- Breadboard

4. Operating System:

- Win-XP, Win-7, Win-8 or higher version
- Arduino coding app
- Bluetooth controller app in the android device

9.4 Summary

In this chapter we described what tools we have taken benefit of in order to build this sophisticated system. Furthermore, we have also learnt what tools we need in order to conduct a thorough testing of this system from both the user and the admin's perspective.

CHAPTER 10

WORKING SHEETS

10.1 Introduction

In this chapter, we observe the entire work structure, meaning how the scheduling was maintained throughout the developmental phase. We shall also see the financial foundation of this project and furthermore the feasibility study should be also discussed.

10.2 Work Breakdown Structure

In order to develop this system, we gave ample importance to scheduling because we believed if we want to provide the best of quality then we must give due importance to scheduling which helped us to garner better results. The figure below focuses the weekly work we had accomplished.

Task	4/6/18	11/6/18	18/6/18	25/6/18	2/7/18	9/7/18	16/7/18	23/7/18	30/7/18
Project Topic Selection		-	-	-	-	-	-	-	-
Project Proposal Presentation	-				-	-	-	-	-
Component's Collection & Briefing	-	-	-	-		-	-	-	-
Wiring Implementation on Circuit	-	-	-	-			-	-	-
Code Build & Test	-	-	-	-	-	-		-	-
Bug Fixing & Final Test	-	-	-	-	-	-	-		-
Final Demonstrations	-	-	-	-	-	-	-	-	

Table 1: Work plan

10.3 Financial Plan

Name of Component's	Price
ARDUINO MEGA2560	2000
SERVO SG90	2500
SERVO 14K TORQUE	1200
SERVO 10K TORQUE	2000
BATTERY	1800
CHARGER	1500
FLEX SENSOR	10500
BREAD BOARD	180
JUMPER WIRE	450
BOSTER	500
POT	20
RESISTORS	10
SPRING	50
CHARGER CONNECTING CABLE	160
PCB	120
WOOD & PLASTIC PIPES	100
TOTAL	22,990/- (Twenty two thousand & nine hundred ninety only)

Table 2: Equipment's Cost

A brief detail of Financial cost is given in the above table. As it's a junior design Project so its cost is within budget. The cost for the whole project is divided by four parts each part is contributed by the members of our group.

The expense for the project is very cheap, As the available Home automation system in the market are quite bit expensive and those are not user friendly. The project is basically developed to reach each category people. Our Project price is below 2000/-taka (two thousand taka only) which is very cheap to nowadays people.

10.4 Feasibility Study

Depending on the results of the initial investigation the survey is now expanded to a more detailed feasibility study. "**FEASIBILITY STUDY**" is a test of system proposal according to its workability, impact of the User, ability to meet needs and effective use of the resources.

Our Project cost is around twenty-three thousand taka which is very cheap nowadays, as the Animatronic hand controller devices available in the market are quite expensive so we think that our project is much more cost effective.

The project is based on basically three component's, Arduino Uno, NRF module, Flex Sensor, Servo motor. The resources for this project are available in any electronics shop around the city. The benefits of animatronic hand typically fall into a few categories, including savings, safety, convenience, and control. Additionally, consumers purchase animatronic hand for comfort and elimination of using bare hand. This project has no legal issues & environmental issues.

10.5 Summary

To conclude, we discussed the scheduling processes of developing this system. Additionally, we have also identified how feasible the system is through the lens of evaluating using various feasibility studies.

CHAPTER 11

FUTURE WORK

10.1 Introduction

This chapter discusses the future scope or the implementation of this system. As our system is very basic, various forms of new features can be incorporated to this system as per the requirements.

10.2 Future Scope of Work

The main objective of developing this system is to provide a basic user-friendly animatronic hand controller to eliminate the use of our own hand. The system can be more improved than the current form. This project is designed for every category people, in future we will develop this project with many more advance technologies will be enhanced in our project.

Our challenge is to keep the price of the project within budget & will try to reduce the device price if it can be produced in a larger scale.

10.3 Summary

This chapter has described the possible future applications of the design. But there are a lot of possibilities with the designed system. The system may need some research for different applications, though the principle of the designed system will remain as it is.

CHAPTER 12

DESIGN IMPACT

12.1 Introduction

In this chapter, we discuss about the various impacts that our system has been able to generate.

12.2 Environmental Impact

By introducing this system for the consumers, we can say that it has no environmental impact.

12.3 Economic Impact

The economic impact that this system entails is that by introducing this system for the people who are working with their own hand and which has a several risk factors so to eliminate those risk of using own hand.

12.4 Social Impact

The Animatronic Hand Controller system will be socially acceptable as this kind of system is the need for the benefit of us. In this era of ours, everything has been automated to provide comfort for the users. Therefore, our system is no exception.

12.5 Sustainability

Our system has been able to reach every category people as this is the very basic hand controller system so consideration of the sustainability for this project is justified.

12.6 Summary

This chapter has covered the different types of impacts that our system offers and those has been described and discussed. From the above given impacts we can conclude that our designed system is good enough to use under any circumstance.

CHAPTER 13

COMPLIANCE WITH IEEE STANDARDS

13.1 Introduction

In this section we discuss about the consistence of our task with diverse standards. There are a few distinct standards, amongst which the IEEE standards, US standards and European standards are talked about in this part.

13.2 Compliance with IEEE standard

There are a few distinct guidelines put forward by IEEE Standards affiliation. The majority of them however are not material for our framework. We have included idea of operation as for the IEEE standard. A conference paper has additionally been submitted and affirmed by IEEE standards entitled "Animatronic Hand Controller – A Cost Efficient and User-Friendly System" that points of interest out our work on this task.

13.3 Summary

In this section we have examined the different compliant standards and made sure that we are in accordance with. These standards have been put without hesitation so as to control things, guarantee well-being and ensure there are no well-being dangers to the use of distinctive segments. It is imperatively essential to maintain these measures and we have done as such over the span of our task work.

CHAPTER 14

RESULTS

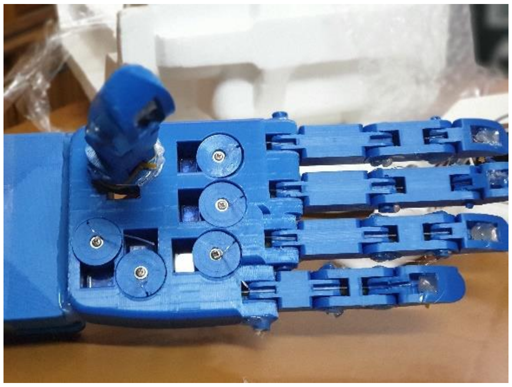
14.1 Introduction

This chapter of the report contains the results that we achieved throughout the course of using this system.

14.2 Results Achieved

From initiation through conclusion of developing this system the following results has been achieved. They are as follows:

- The system can be administered by a non-IT technician.
- The system is market ready for commercial use.
- The system can play its role properly form 0.5(half kilometer) apart from the desired position.



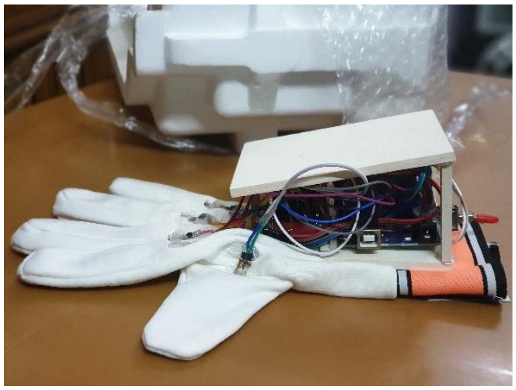


Figure 41: System working result

14.3 Summary

This chapter has covered the different types of results that we have managed to obtain throughout the course of using this system.

CHAPTER 15

CONCLUSION

In conclusion, this product will be completely sound, and has great market value. The advantage of this product is that there are no known competitors out there. However, the product developed is a very user-friendly. First, All the devices ran successfully using the android devices; this is an obvious advantage, as users are more likely to carry it around with them, Secondly, this project allows for greater development of products. The code is open source and it is providing for development by others for application in several ways. In particular, this will provide installation instructions on setting up for the task. Hence, that there is no better product available in the market.

From the beginning of developing this system, the main goal was to develop a fully functional state of the art system which could cater to the masses and it would become a tremendous commercial entity. By introducing the Animatronic Hand Controller, it would be a stepping stone to take the initiative of introducing such automated hand system. The focus was not only to finish the project in due time but to design this system in such a way that it would genuinely be useful in the real world. To conclude, a long way in achieving such a significant milestone has been believed by us and no qualms in mentioning by us. Our project really surpassed our expectation in order to achieve what has been accomplished so far. Thus, this project is just the beginning of innovation in the automation field.

BIBLIOGRAPHY

[1] Ingall T, —Stroke--incidence, mortality, morbidity and risk,‖ Journal of insurance medicine, 36(2),pp. 143-152, 2004.

[2] Prange GB, Jannink MJ, Groothuis-Oudshoorn CG, Hermens HJ, Ijzerman MJ, —Systematic review of the effect of robot-aided therapy on recovery of the hemiparetic arm after stroke,‖ Journal of Rehabilitation Research & Development, 43 (2), pp. 171-184, 2006.

[3] Barreca S, Wolf SL, Fasoli S, Bohannon R, —Treatment interventions for the paretic upper limb of stroke survivors: a critical review,‖Neurorehabilitation and Neural Repair, 17(4); pp. 220-226, 2003.

[4] Feys HM, De Weerdt WJ, Selz BE, Cox Steck GA, Spichiger R, Vereeck CE, Putman KD, Van Hoydonck GA, —Effect of a therapeutic intervention for the hemiplegic upper limb in the acute phase after stroke: a single-blind, randomized, controlled multicenter trial,‖ Stroke,29(4); pp. 785-792, 1998.

[5] Kwakkel G, Wagenaar RC, Twisk JW, Lankhorst GJ, Koetsier JC. —Intensity of leg and arm training after primary middle-cerebral-artery stroke: a randomised trial,‖ Lancet, 354(9174), pp. 191-196, 1999.

[6] Krebs HI, Hogan N, Volpe BT, Aisen ML, Edelstein L, Diels C, —Overview of clinical trials with MANUS: a robot- aided neuro-rehabilitation facility,‖ Technol Health Care, 7(6), pp. 419-23, 1999.

[7] Reinkensmeyer DJ, Kahn LE, Averbuch M, McKenna- Cole A, Schmit BD, Rymer WZ, —Understanding and treating arm movement impairment after chronic brain injury: progress with the ARM guide,‖ Journal of Rehabilitation Research & Development, 37(6), pp. 653-662, 2000.

[8] Burgar CG, Lum PS, Shor PC, Machiel Van der Loos HF,—Development of robots for rehabilitation therapy: the Palo Alto VA/Stanford experience,‖Journal of Rehabilitation Research & Development, 37(6), pp. 663-73, 2000.

[9] Hesse S, Schulte-Tigges G, Konrad M, Bardeleben A, Werner C,—Robot-assisted arm trainer for the passive and active practice of bilateral forearm and wrist movements in hemiparetic subjects,‖ Arch Phys Med Rehabil. 84(6), pp. 915- 20, 2003.

[10] Susan C, Emma KS, —Effect of robot-mediated therapy on upper extremity dysfunction post-stroke—a single case study,‖Physiotherapy, 91 (4), pp. 250–256, 2005.

[11] Panagiotis Polygerinos, Zheng Wang, Kevin C. Galloway, Robert J. Wood, Conor J. Walsh, —Soft robotic glove for combined assistance and at-home rehabilitation,‖ Robotics and Autonomous Systems, 73, pp. 135–143, 2015

[12] Ueki S, Nishimoto Y, Abe M, Kawasaki H, Ito S, Ishigure Y, Mizumoto J, Ojika T,—Development of Virtual Reality Exercise of Hand Motion Assist Robot for Rehabilitation Therapy by Patient Self-Motion Control,‖ in: 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2008, EMBS 2008, 2008, pp. 4282–4285.

[13] N.G. Kutner, R. Zhang, A.J. Butler, S.L. Wolf, J.L. Alberts, —Quality-of-life change associated with robotic- assisted therapy to improve hand motor function in patients with subacute stroke: a randomized clinical trial,‖ Phys. Ther. 90, pp. 493–504, 2010.

[14] A. Polotto, F. Modulo, F. Flumian, Z.G. Xiao, P. Boscariol, C. Menon, —Index finger rehabilitation/assistive device,‖ in: 2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics, BioRob, pp. 1518– 1523,2012.

[15] M. Takagi, K. Iwata, Y. Takahashi, S. Yamamoto, H. Koyama, T. Komeda, —Development of a grip aid system using air cylinders,‖ in: IEEE International Conference on Robotics and Automation, ICRA‘09, pp. 2312-2317, 2009.

APPENDIX

SOFTWARE LISTIN

[Receiving code]

```
#include <Servo.h>    //the library which helps us to control the servo motor
#include <SPI.h>      //the communication interface with the modem
#include "RF24.h"     //the library which helps us to control the radio modem
#include <nRF24L01.h>
String rcv;
//define the servo name
Servo myservo1;  // create servo object to control a servo
Servo myservo2;  // create servo object to control a servo
Servo myservo3;  // create servo object to control a servo
Servo myservo4;  // create servo object to control a servo
Servo myservo5;  // create servo object to control a servo


RF24 radio(7,8);     /*This object represents a modem connected to the Arduino.
                  Arguments 5 and 10 are a digital pin numbers to which signals
                  CE and CSN are connected.*/


const uint64_t pipe = 0xE8E8F0F0E1LL; //the address of the modem,that will receive data
from the Arduino.


int msg[1];
```

```
int data; //data variable
int pos; //position variable

void setup(){
  Serial.begin(9600);

  //define the servo input pins
  myservo1.attach(2);  // attaches the servo on pin 9 to the servo object
    myservo2.attach(3);  // attaches the servo on pin 9 to the servo object
      myservo3.attach(4);  // attaches the servo on pin 9 to the servo object
        myservo4.attach(5);  // attaches the servo on pin 9 to the servo object
          myservo5.attach(14);  // attaches the servo on pin 9 to the servo object


  myservo1.write(170);
  myservo2.write(170);
  myservo3.write(170);
  myservo4.write(170);
  myservo5.write(170);
```

```
  radio.begin();                  //it activates the modem.

  radio.openReadingPipe(1, pipe);   //determines the address of our modem which receive
data.

  radio.startListening();          //enable receiving data via modem

  }



//You don't need to make changes in this section

void loop(){

  if(radio.available()){









    radio.read(msg, 1);

  Serial.println(msg[0]);



    if(msg[0] <11 && msg[0] >-1){

      data = msg[0], pos=map(data, 0, 10, 175, 10);

      myservo1.write(pos);
```

```
 }
 if(msg[0] <21 && msg[0]>10){
   data = msg[0], pos=map(data, 11, 20, 175, 10);
   myservo2.write(pos);
 }
 if(msg[0] <31 && msg[0]>20){
   data = msg[0], pos=map(data, 21, 30, 175, 10);
   myservo3.write(pos);
 }

 if(msg[0] <41 && msg[0]>30){
   data = msg[0], pos=map(data, 31, 40, 175, 10);
   myservo4.write(pos);
 }

     if(msg[0] <51 && msg[0]>40){
   data = msg[0], pos=map(data, 41, 50, 175, 10);
   myservo5.write(pos);
 }
 delay(200);


}
else{
```

```
  myservo1.write(170);
  myservo2.write(170);
  myservo3.write(170);
  myservo4.write(170);
  myservo5.write(170);

delay(200);

  }
}
[/code]
```

[Transmitting code]

```
#include <SPI.h>                    //the communication interface with the modem
#include "RF24.h"                    //the library which helps us to control the radio modem
#include <nRF24L01.h>
int msg[1];
String rcv;
//define the flex sensor input pins
int flex_5 = A4;
int flex_4 = A3;
int flex_3 = A2;
int flex_2 = A1;
int flex_1 = A0;


//define variables for flex sensor values
int flex_5_val;
int flex_4_val;
int flex_3_val;
int flex_2_val;
int flex_1_val;
```

int flex_5_valgap=0;

int flex_5_valmax=0;

int flex_5_valmin=512;

int flex_4_valgap=0;

int flex_4_valmax=0;

int flex_4_valmin=512;

int flex_3_valgap=0;

int flex_3_valmax=0;

int flex_3_valmin=512;

```
int flex_2_valgap=0;

int flex_2_valmax=0;


int flex_2_valmin=512;


int flex_1_valgap=0;

int flex_1_valmax=0;


int flex_1_valmin=512;
```

```
RF24 radio(7,8);                    //5 and 10 are a digital pin numbers to which signals CE and
CSN are connected.


const uint64_t pipe = 0xE8E8F0F0E1LL; //the address of the modem, that will receive data
from Arduino.



void setup(void){

  Serial.begin(9600);

  radio.begin();                    //it activates the modem.

  radio.openWritingPipe(pipe);        //sets the address of the receiver to which the program
will send data.

}


void loop(){


  if (radio.available()) {

 //   char text[32] = "";

   // radio.read(&text, sizeof(text));

  //  Serial.println(text);

  //  rcv +=String(text);



  }
```

```
  flex_5_val = analogRead(flex_5);
Serial.print("Flex5:");Serial.print(flex_5_val);

  //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
if(flex_5_val>flex_5_valmax) flex_5_valmax=flex_5_val;
if(flex_5_val<flex_5_valmin) flex_5_valmin=flex_5_val;
flex_5_valgap=(flex_5_valmax-flex_5_valmin)/3;

  Serial.print(", max:");
 Serial.print(flex_5_valmax);
   Serial.print(", min:");
 Serial.print(flex_5_valmin);
   Serial.print(", gap:");
 Serial.print(flex_5_valgap);

                              //175 - 0
  //flex_5_val = map(flex_5_val, 300, 50, 0, 10);
```

```
 if(flex_5_val<flex_5_valmax && flex_5_val>flex_5_valmax-flex_5_valgap){
   msg[0] = 1;
 radio.write(msg, 1);
 delay(100);
 }
 else   if(flex_5_val<flex_5_valmax-flex_5_valgap && flex_5_val>flex_5_valmax-
2*flex_5_valgap){
      msg[0] = 5;
 radio.write(msg, 1);
 delay(100);
 }

   else   if(flex_5_val<flex_5_valmax-2*flex_5_valgap && flex_5_val>flex_5_valmin){
      msg[0] = 9;
 radio.write(msg, 1);
 delay(100);
 }

     Serial.print(", msg:");
 Serial.print(msg[0]);

 ////////////////////////////////////////////////////////////////////////////////////////////////////////
 // msg[0] = flex_5_val;
 // radio.write(msg, 1);
```

```
flex_4_val = analogRead(flex_4);
Serial.print(",  Flex4:");Serial.print(flex_4_val);

//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
if(flex_4_val>flex_4_valmax) flex_4_valmax=flex_4_val;
if(flex_4_val<flex_4_valmin) flex_4_valmin=flex_4_val;
flex_4_valgap=(flex_4_valmax-flex_4_valmin)/3;

Serial.print(", max:");
Serial.print(flex_4_valmax);
Serial.print(", min:");
Serial.print(flex_4_valmin);
Serial.print(", gap:");
Serial.print(flex_4_valgap);

                    //175 - 0
//flex_4_val = map(flex_4_val, 300, 50, 0, 10);

if(flex_4_val<flex_4_valmax && flex_4_val>flex_4_valmax-flex_4_valgap){
  msg[0] = 11;
```

```
 radio.write(msg, 1);

 delay(100);

 }

 else   if(flex_4_val<flex_4_valmax-flex_4_valgap && flex_4_val>flex_4_valmax-
2*flex_4_valgap){

      msg[0] = 15;

 radio.write(msg, 1);

 delay(100);

 }


   else   if(flex_4_val<flex_4_valmax-2*flex_4_valgap && flex_4_val>flex_4_valmin){

      msg[0] = 19;

 radio.write(msg, 1);

 delay(100);

 }


    Serial.print(", msg:");

Serial.print(msg[0]);


 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
                    //175 - 0
 // flex_4_val = map(flex_4_val, 300, 50, 11, 20);
 // msg[0] = flex_4_val;
 // radio.write(msg, 1);

  flex_3_val = analogRead(flex_3);
  Serial.print(",  Flex3:");Serial.print(flex_3_val);

 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////
if(flex_3_val>flex_3_valmax) flex_3_valmax=flex_3_val;
if(flex_3_val<flex_3_valmin) flex_3_valmin=flex_3_val;
flex_3_valgap=(flex_3_valmax-flex_3_valmin)/3;

  Serial.print(", max:");
 Serial.print(flex_3_valmax);
   Serial.print(", min:");
 Serial.print(flex_3_valmin);
   Serial.print(", gap:");
 Serial.print(flex_3_valgap);
```

```
                        //175 - 0
  //flex_3_val = map(flex_3_val, 300, 50, 0, 10);






  if(flex_3_val<flex_3_valmax && flex_3_val>flex_3_valmax-flex_3_valgap){
    msg[0] = 21;
  radio.write(msg, 1);
  delay(100);
  }
  else   if(flex_3_val<flex_3_valmax-flex_3_valgap && flex_3_val>flex_3_valmax-
2*flex_3_valgap){
        msg[0] = 25;
  radio.write(msg, 1);
  delay(100);
  }

    else   if(flex_3_val<flex_3_valmax-2*flex_3_valgap && flex_3_val>flex_3_valmin){
        msg[0] = 29;
  radio.write(msg, 1);
  delay(100);
  }
```

```
Serial.print(", msg:");
Serial.print(msg[0]);
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////





                    //175 - 0
// flex_3_val = map(flex_3_val, 300, 50, 21, 30);
// msg[0] = flex_3_val;
// radio.write(msg, 1);



flex_2_val = analogRead(flex_2);
Serial.print(",  Flex2:");Serial.print(flex_2_val);


//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
if(flex_2_val>flex_2_valmax) flex_2_valmax=flex_2_val;
if(flex_2_val<flex_2_valmin) flex_2_valmin=flex_2_val;
flex_2_valgap=(flex_2_valmax-flex_2_valmin)/3;
```

```
Serial.print(", max:");
Serial.print(flex_2_valmax);
Serial.print(", min:");
Serial.print(flex_2_valmin);
Serial.print(", gap:");
Serial.print(flex_2_valgap);

                    //175 - 0
//flex_2_val = map(flex_2_val, 300, 50, 0, 10);




if(flex_2_val<flex_2_valmax && flex_2_val>flex_2_valmax-flex_2_valgap){
  msg[0] = 31;
radio.write(msg, 1);
delay(100);
}
else  if(flex_2_val<flex_2_valmax-flex_2_valgap && flex_2_val>flex_2_valmax-
2*flex_2_valgap){
    msg[0] = 35;
radio.write(msg, 1);
delay(100);
}
```

```
  else   if(flex_2_val<flex_2_valmax-2*flex_2_valgap && flex_2_val>flex_2_valmin){
     msg[0] = 39;
 radio.write(msg, 1);
 delay(100);
 }



   Serial.print(", msg:");
Serial.print(msg[0]);
 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

                         //175 - 0
 //flex_2_val = map(flex_2_val, 300, 50, 31, 40);
// msg[0] = flex_2_val;
// radio.write(msg, 1);

 flex_1_val = analogRead(flex_1);
 Serial.print(",  Flex1:");Serial.print(flex_1_val);
```

```
 ///////////////////////////////////////////////////////////////////////////////////////////////////////////
if(flex_1_val>flex_1_valmax) flex_1_valmax=flex_1_val;
if(flex_1_val<flex_1_valmin) flex_1_valmin=flex_1_val;
flex_1_valgap=(flex_1_valmax-flex_1_valmin)/3;

  Serial.print(", max:");
 Serial.print(flex_1_valmax);
   Serial.print(", min:");
 Serial.print(flex_1_valmin);
   Serial.print(", gap:");
 Serial.print(flex_1_valgap);

                            //175 - 0
  //flex_1_val = map(flex_1_val, 300, 50, 0, 10);

  if(flex_1_val<flex_1_valmax && flex_1_val>flex_1_valmax-flex_1_valgap){
    msg[0] = 41;
  radio.write(msg, 1);
  delay(100);
```

```
 }
 else   if(flex_1_val<flex_1_valmax-flex_1_valgap && flex_1_val>flex_1_valmax-
2*flex_1_valgap){
      msg[0] = 45;
 radio.write(msg, 1);
 delay(100);
 }


   else   if(flex_1_val<flex_1_valmax-2*flex_1_valgap && flex_1_val>flex_1_valmin){
      msg[0] = 49;
 radio.write(msg, 1);
 delay(100);
 }


     Serial.print(", msg:");
 Serial.println(msg[0]);


 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
//175 - 0
 // flex_1_val = map(flex_1_val, 300, 50, 41, 50);
//  msg[0] = flex_1_val;
//  radio.write(msg, 1);
delay(200);


}
```

[/code]