This is the Project Documentation of “The Hand”, a project presented by Maru-a-Pula School to the Botho University Linkz ICT Championship 2013 for the IT Innovation Category.

The Hand

The Project Documentation

Amrit Amar & Tawanda W.T. Mulalu

Table of Contents

About Us

Amrit Amar is an up-and-coming young programmer hailing from India and living in Botswana. Known in his school for his recent victory in the ‘Best Game Programmer In Africa’ contest, Amrit spends his days writing code… and assisting those who can’t (he teaches the afternoon Beginners and Advanced Programming activities at Maru-a-Pula). He and Tawanda formed the team that won last year’s Botho College ICT Challenge.

Tawanda W.T Mulalu is a wannabe writer and scientist who enjoys chronicling his and Amrit’s daily adventures. Aside from occasionally assisting and encouraging Amrit’s IT work, Tawanda is also an active debater and has received Best Speaker last year and this year at the Botswana Debating Nationals’ for Secondary Schools and served as Captain of the 2012/2013 National Debating Team.

1. Introduction

a. The Botho College ICT Linkz Challenge 2013 Specifications and Document Provided

Botho University
LINKZ
ICT Challenge 2013

BOTHO
UNIVERSITY

IT INNOVATION/WORKING MODEL

EVENT MANAGER: Ms SETHUNYA JANE

EMAIL: sethunya.jane@bothouniversity.ac.bw

CONTACT NUMBER: 72232251

EVENT CO-ORDINATORS:

- Ms. BUCHILO MOSWEU
- Ms. AGNES MOATSWI

EVENT DESCRIPTION:

Here participants will be required to come up with innovative ideas like coming up with a working model to a problem identified in the domain. Here we expect a high level of originality, the benefits derived from the model identified, how the model will be implemented and the life cycle to be adopted. We also expect the participants to bring their ideas with relevant documentation.

EVALUATION CRITERIA:

The judges will evaluate each model based on the following aspects:

Innovation (15%)
{Originality, impact, practicality, benefits, applicability}

Knowledge content (10%)
{Background knowledge, implementation technology, software engineering concepts}

Presentation skills (5%)
{Communication skills, slides, confidence, group coordination}

Documentation (15%)
{Requirements, problem statement, analysis, design, implementation, testing, deployment}

Demonstration of prototype (5%)
{Clarity, user interface, code}

Botho University

LINKZ

ICT Challenge 2013

BOTHO UNIVERSITY

LOCATION: SWANENG BUILDING, ROOM NO. S3
SET-UP TIME: 09:00 AM – 09:30 AM
EVALUATION TIME: 09:30 AM – 12:30 PM
PUBLIC VIEWING: 12:45 PM onwards
HARDWARE PROVIDED: POWER SOURCE
MAXIMUM POINTS: 50

EVENT RULES:

- Maximum of 2 participants per team.
- One team per institution only.
- There are no restrictions regarding the tool(s) to be used.
- The innovation must reflect a real life situation.
- The project must be a working model.
- The poster and document must provide all the relevant details
- The innovation must be the participants' original work.
- Any signs of plagiarism would lead to automatic disqualification.

POINTS FOR THE ROLLING TROPHY:

The winner will get 50 points.
The first runner up will get 35 points.
The second runner up will get 15 points.

At the end of the day points from all events will be added up together and the institution which has accumulated maximum points will be awarded a rolling trophy. All the event descriptions are approved by the event management committee. The event management committee has the right to make any changes on the given proposal at any time.

****

[Above] Pictures showing the Document for the Challenge

"In 2009 Botho University launched an ICT based competition called ***LINKZ "The ICT Challenge"****. The aim of which was to gather together the best ICT students in the country. To date, it is the only competition of its kind. LINKZ's main aim was to foster learning excellence through healthy competition by pitting the best of the Botswana student community in the field of IT from across the country.*

That event brought together over 300 bright young people from over 6 major tertiary institutions. Its success in the local scene saw it established to become a annual event and although it missed 2010, in 2011, LINKZ proved itself yet again with an increased number of participants totaling up to 400. In the third edition of this event, which was in 2012, LINKZ opened its doors to include secondary schools, and further saw participation skyrocket to an estimated 500."

~ The Official Website of the Competition

This is the original design specification given by Botho College as a public document. The paragraph is from the Competition Website.

b. The Problem Statement

i. The Current Situation

Botswana is a mining country which receives most of its national revenue from diamonds found all over the country. Mines are mostly safe, but every once in a while, one hears of a devastating accident, caused by equipment, negligence and many other factors. Unfortunately mines also have terrain that is too dangerous for human work. Many valuable minerals can be found in dangerous terrain but companies ignore them due to issues of worker safety.

Space exploration and travel represent the next frontier in the mankind's scientific endeavors. However missions to space are expensive and there is the serious risk of safety for people onboard rockets. Millions of Dollars are spent on creating machines to work in space however none of this machinery has the ability to match the precision and movement of the human hand. Currently engineers create expensive robots to automatically mine resources on the moon and send them back to us. Yet space presents immense difficulty to engineering due to its nature, if a machine makes a mistake in space it is very difficult to account for resources that find their way floating into space. This can lead to millions of dollars being wasted due to simple mistakes in calibration. Human cannot be always sent to mine for resources on Earth and in space because of dangerous environments but we can send unmanned probes to investigate and examine the surface of far away planets. But can humanity go further in this exploration?

Aside from mining and space exploration there are many other jobs that require machinery with precision similar to that of the human body, most specifically and notably the human hand. Not only has that, but machinery with this precision needed a great amount control in order to function at high efficiency. Such high levels of control can best be achieved via the human mind itself as AI has not yet reached a level of development at which software can respond to novel problems as well as human beings can, or perform tasks outside of defined programming.

So exactly what machinery could possibly hope to achieve the above demands? What could possibly be able to use tools as well as a human can? And not only that but be able to function in a variety of environments and used in a variety of disciplines and environments.

ii. The Solution

Control robotics already exists, but the solution presented here strays away from simple remote control and presents machines with a Human Control Interfaces. These machines represent the latest and greatest in IT innovation, connecting the two disciplines of Engineering and Programming in an exciting and useful way. Machines with human control interfaces differ by other machines in that they are designed specifically to mimic human movement by the movement input of the user. To illustrate this concept, a Robot Hand has been created as the working model, with a Control Glove as the input.

The Robot Hand has been designed in such a way that it will mimic the movements made by user’s actual hand in the Control Glove. The mimicking of the movement is done as far as the physical calibration of the Robot Hand itself and of the programming will allow.

To start with building this machine, a Hand Model was build first. The hand is one of the most used parts of the Human Body and one of the most precision controlled We decided to create a machine to replicate the Hand. In essence, the User wears a control device, and using his/her movements, controls the robot.

c. Analysis of Problem and Solution

i. The Chosen Solution

The working model and design was brought into inception after careful consideration of the world of development robotics. A general aim in the research and development of robotics has been to create increasingly complex machines that are efficient at what they do, and this has largely led to the mimicry of many biological mechanisms – i.e. robots that mimic human walking patterns or of other animals. The main problem however is the issue of how these mechanisms are controlled. One can either give these machines a mind of their own to do things or perhaps have a human control them through their brain patterns. However, machines having their own mind or AI technology are still very much in its infancy, and the ability for AI machines to respond to novel situations, rather than simply previously recognized patterns, is nowhere near to that of a human adult, let alone a child. Furthermore, owing to the complexity of the human brain we cannot hope to create a machine that read our brain patterns when we Neurologists still have very little understanding of how our brain actually functions. Thus, a solution was found on how to effective control increasing complex machinery with the aim of the refined subtlety and precision that humans have- a human control interface that works via mimicry.

The main reason the project was done however was to combat the issue of human safety in dangerous situations, and to extend the capabilities of a human across long distances. Considering Botswana is a country which sources most of its revenue from its mining, and considering that mining is quite a dangerous activity for those involved in the actual physical process (as numerous news stories have revealed in the past), and also considering the limits of human reach in physical spaces then it can be seen that the working model presented is an effective solution to the issue of safety and the ability to perform tasks from afar. A specific example:

> *A source reveals to a mining company that possible mineral reserves can be found in an unsafe area beneath the ground (say, one that would be prone to a cave-in). A human worker cannot be sent for issues of safety, and it is not possible to attempt to use heavy*

machinery to mitigate the circumstances due to the sensitivity of the earth and rock in the area. A probe with the finished version of the working model and a video camera could be sent it. This probe would have the ability to use the sensitivity a human worker has that would be needed to examine the area. As the probe also does not constitute heavy machinery the danger of a cave in is significantly reduced.

ii. The Objective of the Solution

The objective of the solution was to create a Robotic Hand that will be controlled by a glove the User wears. This Glove will have sensors that will detect finger movement and will then replicate that movement on the Robotic Hand's fingers.

iii. Why this solution?

It was felt that the given solution best solved the problem as it effectively deals with one issue - safety. An important goal considered for the project was the issue of the safety of a human. By creating a robot that is controllable by the human directly he or she does not physically have to be present in the working environment to ensure his/her safety, yet still be able provide the same precision and care a human would be able give to the job.

2. Design

a. The Full Project

The solution will be referred to in this document as simply The Hand. Now, for *Human-Controlled Machines*, we need a way to control the hand. Since we will be using the User's hand to control the machine, we decided to create a *Control Glove*. The Control Glove will read signals from the hand and send it to the Hand. The problem was linking this up.

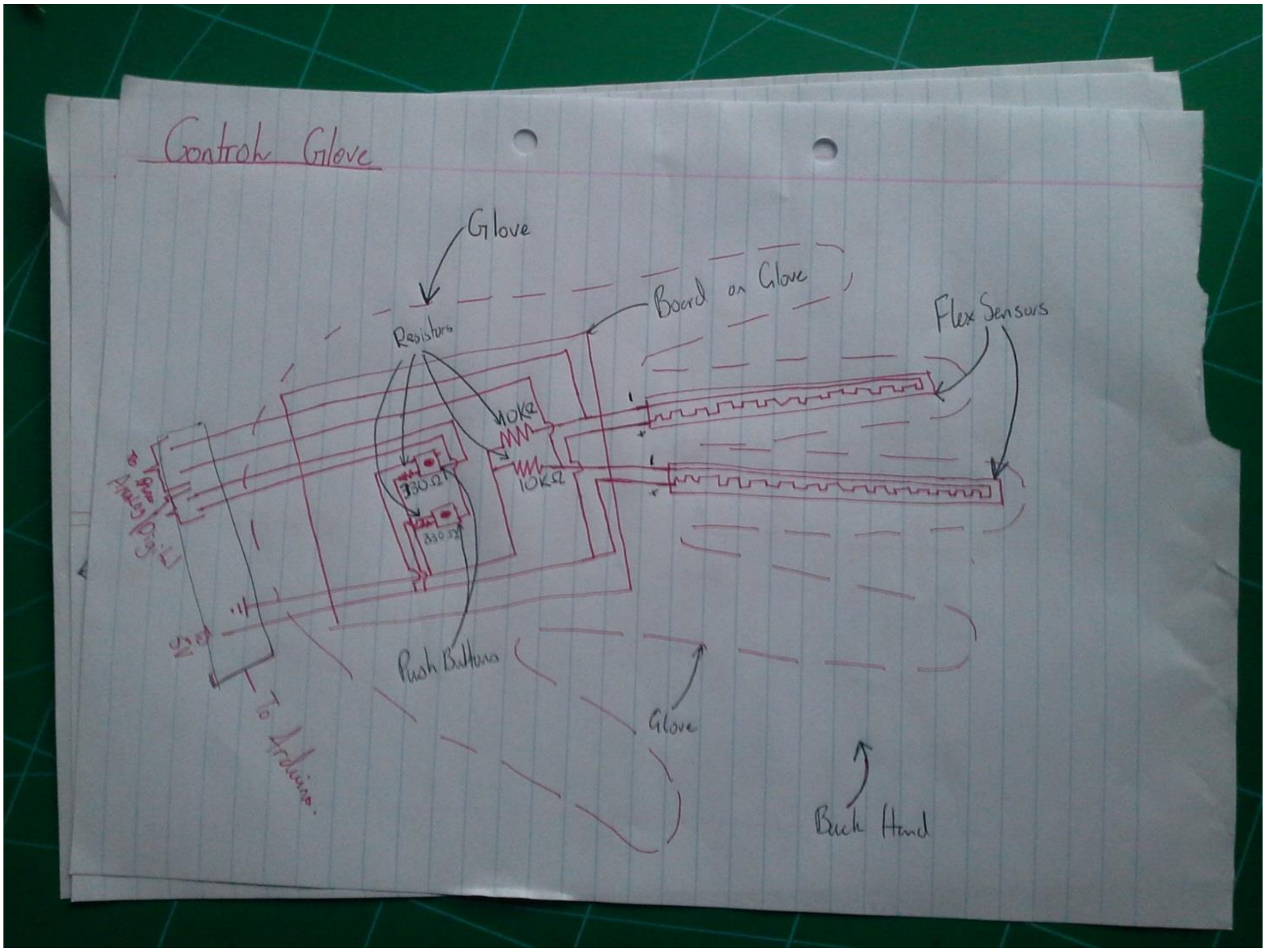
We then heard of *Arduino*. Arduino is prototyping board that can be programmed to do a lot of things. It has Analog Input and Digital I/O pins. It was perfect. To read the signals from the Control Glove, we decided to use *Flex Sensors*. This is a type of Variable resistor that changes values depending on the bend. We could read signals from this and send it to the Arduino. The Arduino would process this and send it to the Robotic hand.

The Robotic Hand would be controlled by Servo Motors. These are essentially DC Motors, but with more precision and torque. We used these 3 components to drive the Robotic Hand based on the Control Glove.

Of course, we wanted to add more functionality and with the Arduino, this was possible. We implemented a Light Detector, a Light and Sound module. This would be triggered using buttons on the Control Glove. Throughout the design process, we made various diagrams to make the hand. The final design we chose is below.

Now, because of equipment constrains, we decided to program only 2 of the Fingers to move on the Robot Hand depending on the Control Glove. This was only a prototype and we felt that it was enough to show the functionality with 2 fingers only.

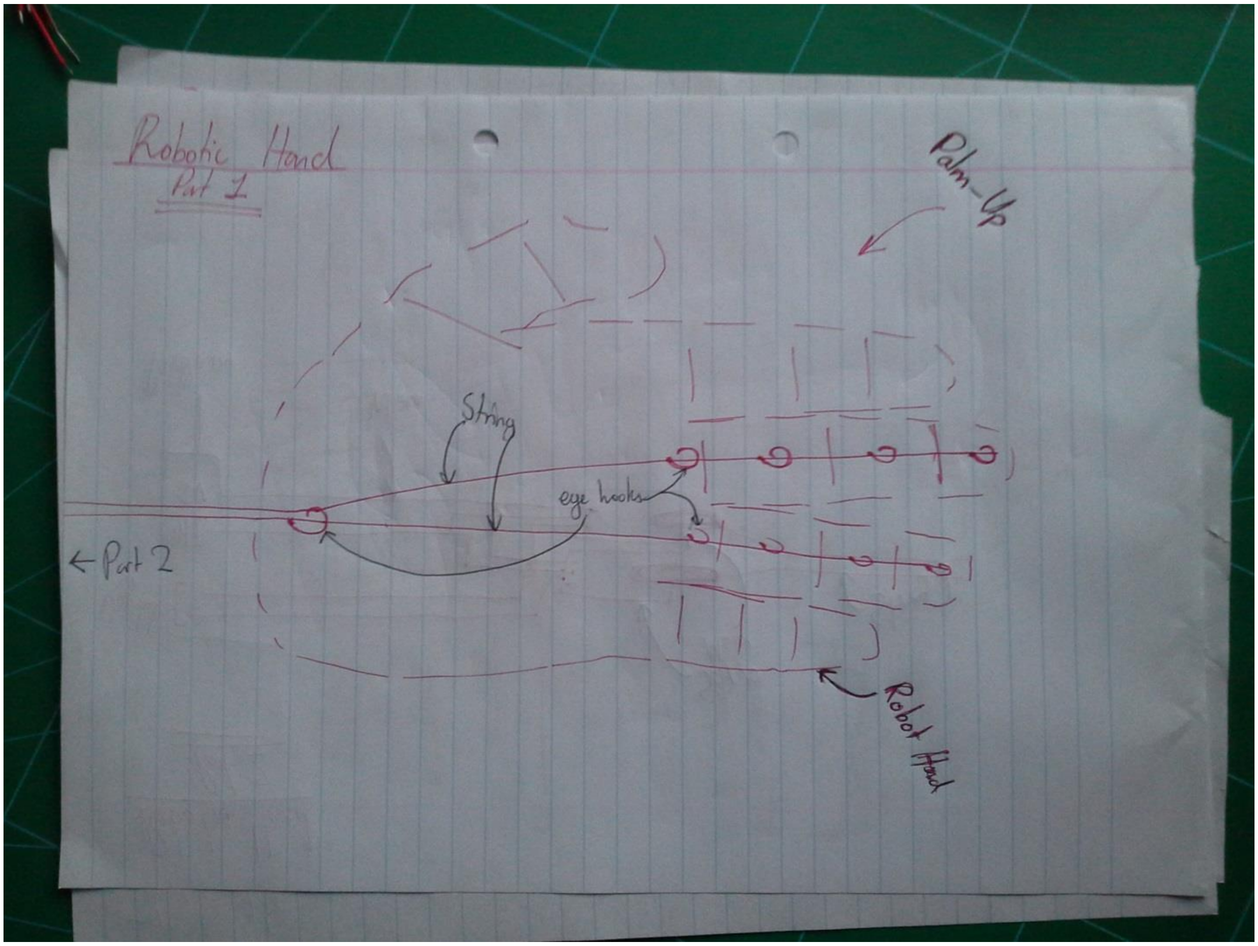
We explained how this whole project works in the **Explanation** section.



[Above] Picture showing the Control Glove Design.

The design for the control glove was mainly developed by the idea of having the full control unit on the hand itself. Here, we took a glove and fixed two Flex Sensors on the fingers we would be testing. On the back of the hand, we had a board that housed all the electronics of the glove. On the top of the board, we only showed the Resistors and the Buttons. The rest were hidden under the board. The board was fixed to the glove, and a total of 6 wires came out of the Glove. These were the 5V, Ground, Button 1, Button 2, Flex Sensor 1, and Flex Sensor 2.

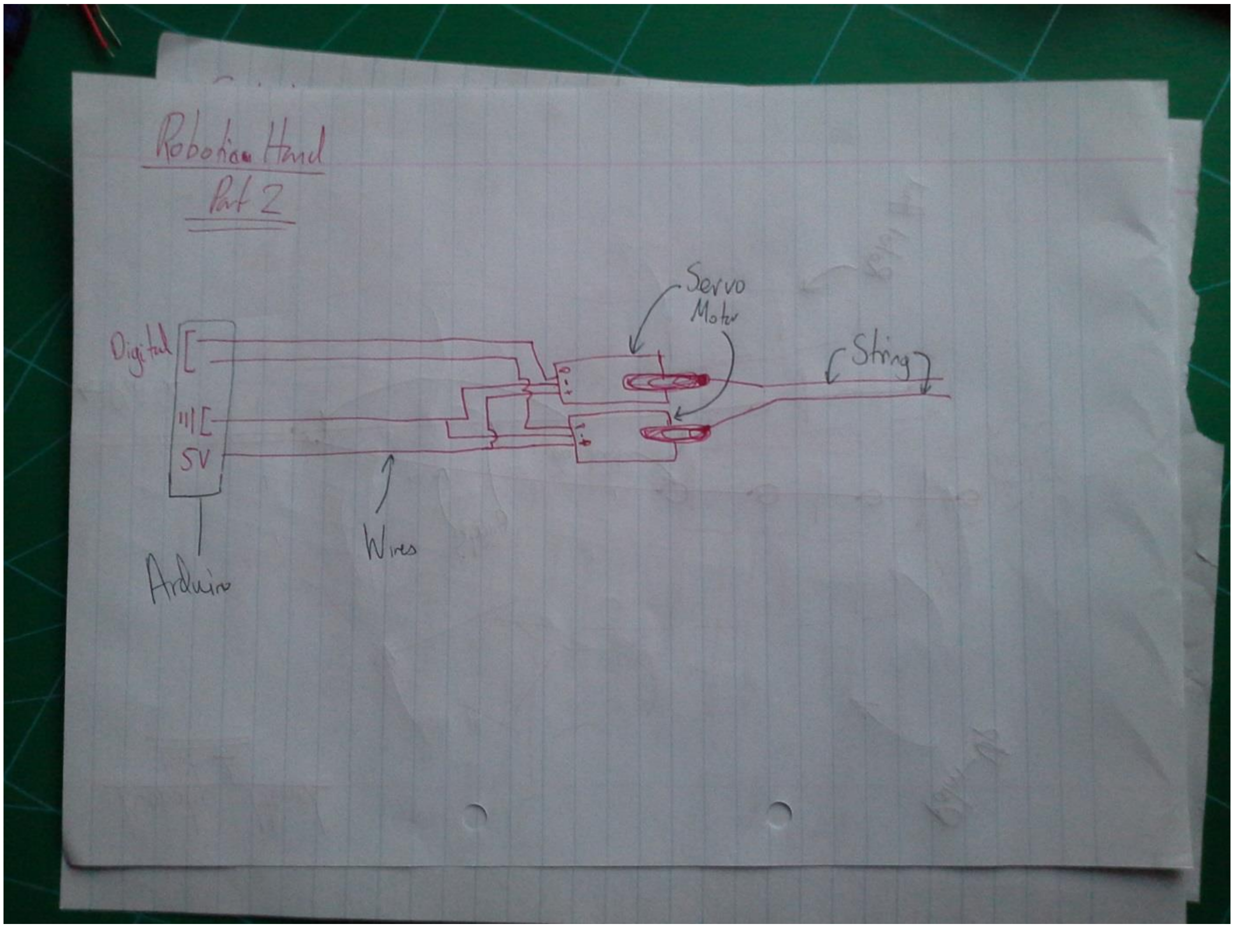
We were happy with the design of the control glove. Only 6 wires came out and these were easily managable. (Look at the **Applications and Limitations** section to see how we planned to get rid of all the wires).



[Above] Picture showing the Robot Arm Design Part 1 with the front face-up.

Originally, we planned to make a robot hand out of wood, but we realized that it would not work. Eventually, we got a Wooden Mannequin Hand. These were the hands that they used at shops to display accessories. Looking at the Biological Model of the Hand, we attached rope and small screw-eyes. This kept the fingers in place and the joints to be manipulated.

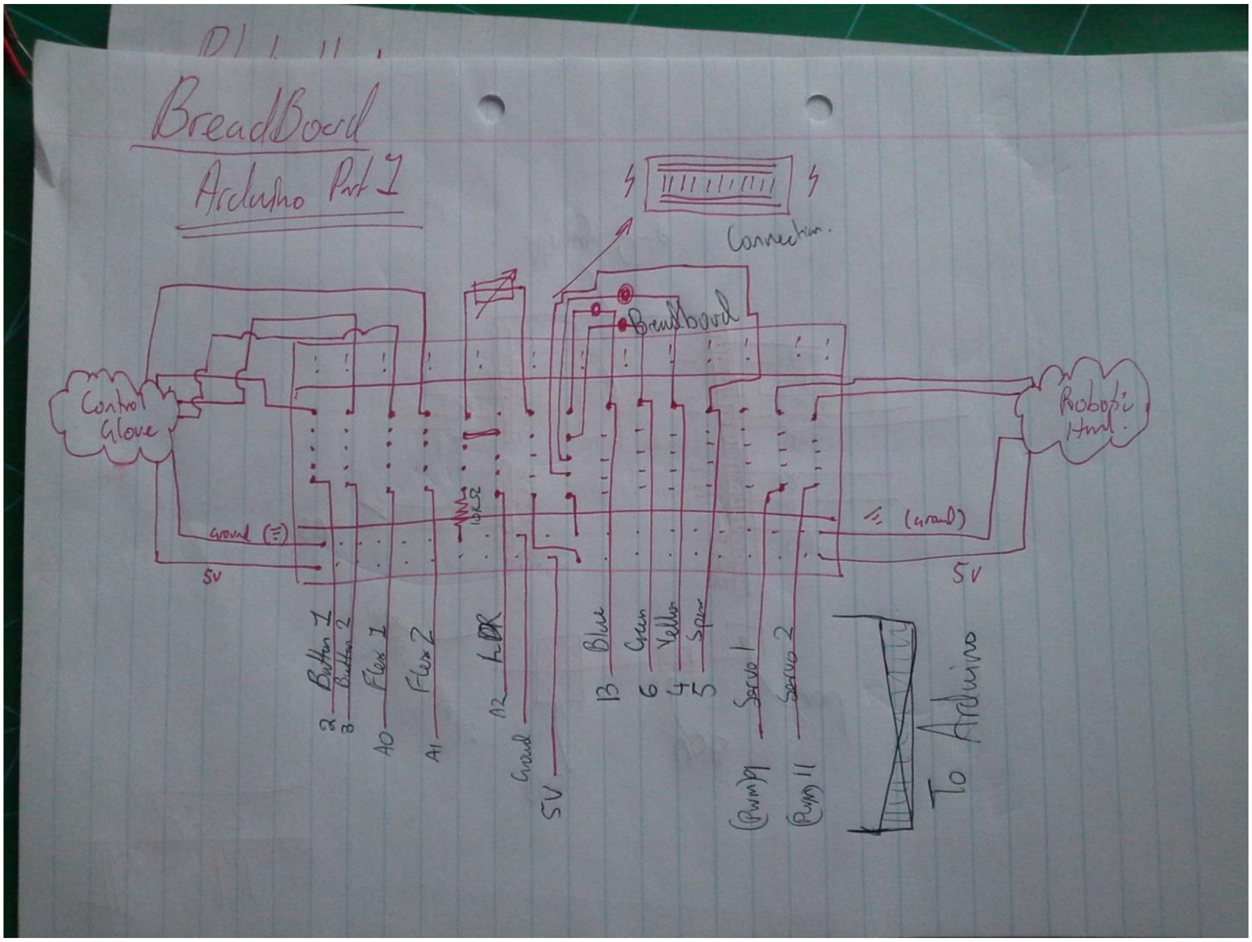
Although we used a pre-made hand, we made sure that all the joints were loose by filing them and lubricating them. The two strings went on to the Servo Motors. For a future design, we plan to make a custom hand, capable of moving all joints freely.



[Above] Picture showing the Robot Arm Design Part 2 with the Servo Motors.

Here, the strings from the Robot arm connect to the Servo Motor's Propellers. From there, the Motors are connected to the Arduino Breadboard. There are 4 wires in total coming out (The 5V, the ground, Servo 1, and Servo 2).

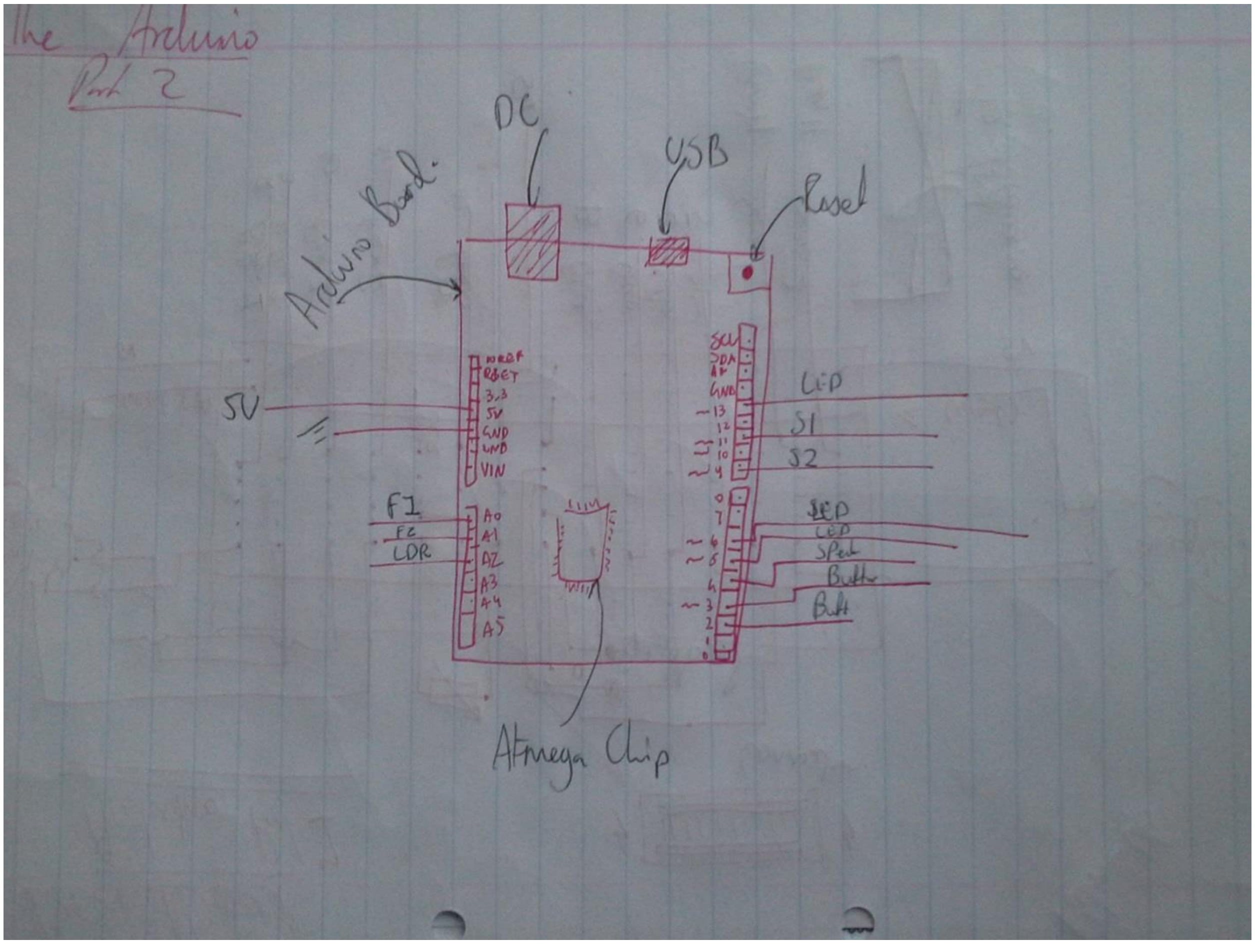
As continuing from the last part, we were pleased by the Position of the Servo Motors when we built the hand. Mannequin Hands always have space at the bottom and so we used this to our advantage and fixed the motors there. See **Method** for an exact explanation of what we did.



[Above] Picture showing the Arduino Breadboard Connections.

It looks complicated but it isn't. The wires from the Control Glove and Servo Motors merge here. All the 5V and Ground wires are connected together and taken to the Arduino. Other individual wires were carried to the Arduino on its own. The diagram shows exactly to which Pins. In addition to going to the Arduino, we also showed where the LDR, and the LED's go.

We were very happy with this design as it was clear where each wire was going and its purpose. It helped us manage our wires more efficiently.



[Above] Pictures the Arduino Board with Connections.

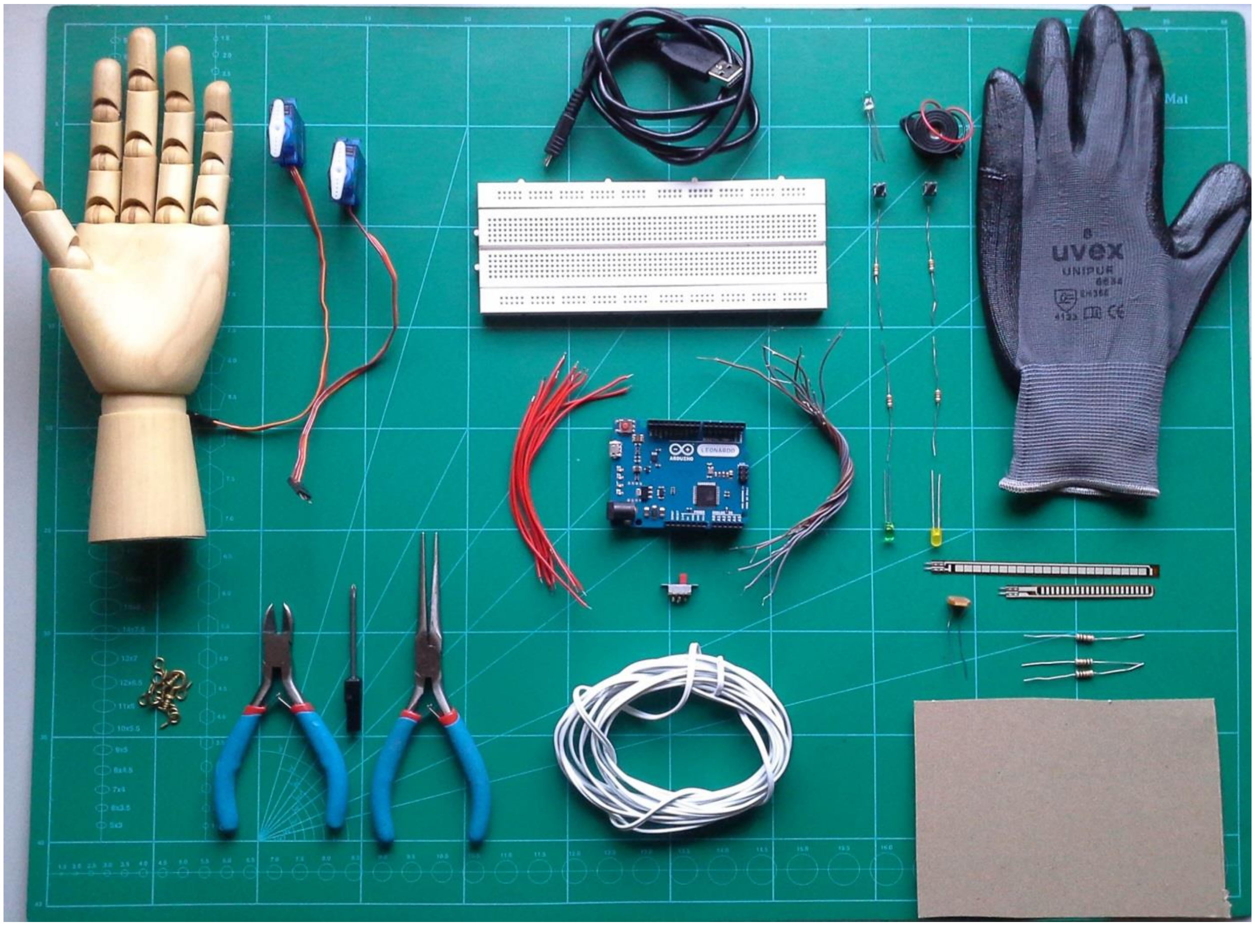
This is the CPU of the whole project, the Arduino. It is powered by the Computer through USB, but can be powered using a DC plug. All the input and output wires are shown here. The code is in the **Method** section.

As all the wires met here, we had a clear understanding of how the whole project would work now.

3. Materials

a. An Overview of All Materials

There were more than 25 components used to build the *Hand*. The most important was the *Arduino Leonardo Board*. Other major components included *Flex Sensors* and *Servo Motors*.



[Above] Picture showing the Components used in building the Project

The full list is:

- 1 Wooden Mannequin Hand (Left Handed)
- 1 Glove (Left Handed)
- String
- Rubber Band

- 1 Arduino Board with USB Cable (Model: Leonardo)
- 1 Prototyping Breadboard (Standard Size)
- 20+ Jumper Wires (Varying Sizes)
- 2 Flex Sensors (2.2 and 4.5 Inch Sizes)
- 2 Servo Motors (1.2 Kilograms per Centimeter Torque, 9 Grams)
- 3 Light Emitting Diodes (Blue, Green and Yellow Colors)
- 1 Speaker
- 12+ Hook Eyes (Small and Medium Size)
- A Switch
- 4 330Ω Resistors (For LED's and Push Buttons)
- 3 10KΩ Resistors (For Variable Resistors)
- 1 Light Dependent Resistor
- 2 Push Buttons
- Small Cardboard Piece
- Tools (Screwdriver, Pliers and Wire Cutters etc.)
- A Computer with Arduino Development IDE

The components are explained in detail in the next 3 Subsections.

b. The Control Glove Materials Explanation

The Control Glove is the glove the user will wear. The user will use this to control the Hand. The materials here are:

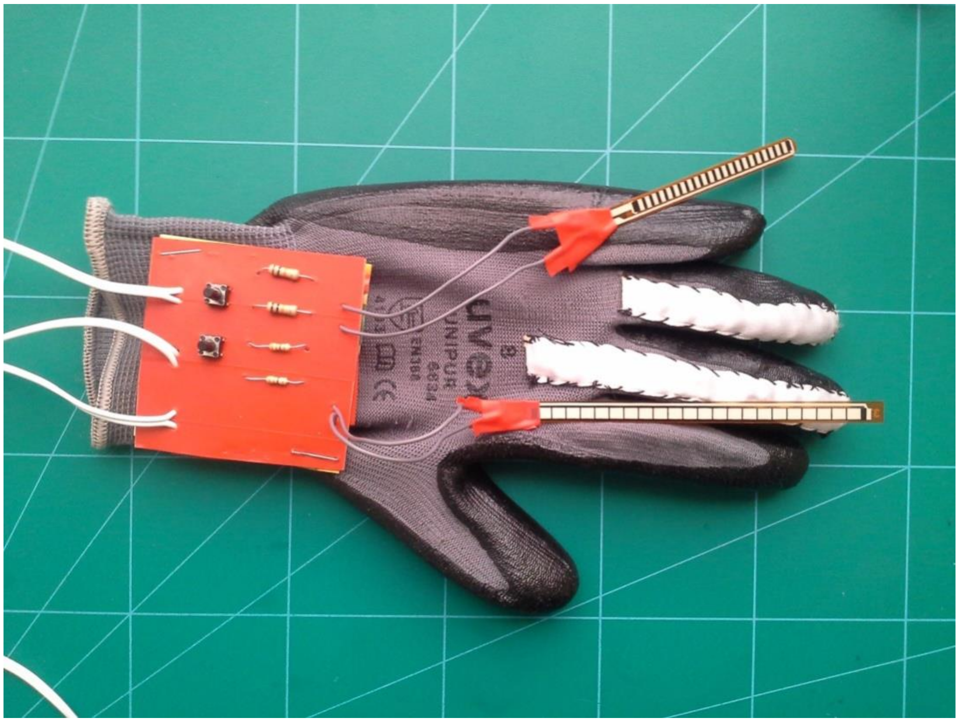
- 2 Flex Sensors
- 2 10KΩ Resistors
- 2 330Ω Resistors
- Connecting Wires
- Glove
- Cloth
- Stiff Board

We used the Middle and Ring fingers for the prototype. The Flex Sensors are basically a type of Variable Resistors, that changes values as it is bent. This creates readings, which can be read by the Arduino.

The resistors were mainly for limiting the current so that it would not damage any equipment. The 2 10KΩ Resistors were for the Flex Sensors. The 2 330Ω Resistors were for the buttons. These were all connected to the Ground.

We used cloth to create pockets, into which the delicate Flex Sensors slid into. This ensured that they were in place and that they would not get damaged. We stitched the pockets on.

The Stiff Board was for building the whole control mechanism for the Glove. All the wiring was on this board and we connected everything under it, wiring all the connections so that a maximum of 6 wires came out of the Control Glove. We made sure that this wouldn't shock the user by creating another board under it. This created a box-like structure, housing all the connections of the Control Glove.



[Above] Picture of the final Control Glove

c. The Hand Materials Explanation

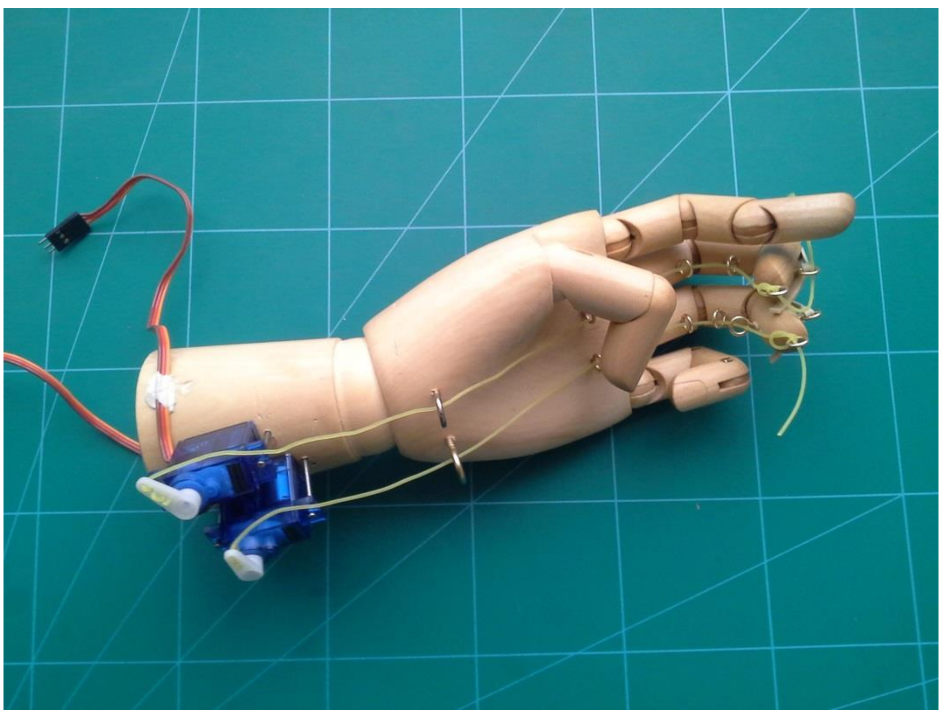
The Hand is controlled by the Control Glove. We got a simple mannequin hand for this project. This hand is ideal because:

a. It has all the Fingers
b. It has all the Finger Joints

Using a biological model, we worked out how the hand works and thus we replicated the many muscles using simple string and rubber bands. They are held in place using small eye hooks. (See **Explanation** Section).

At the base of the hand, we had the big eye hook which threaded the strings to the Servo Motors.

Servo Motors are basically precise-rotation motors that are controlled by Digital Signals. The digital signals came from the Arduino. The Servo Motors have a rotation of up to 179 Degrees, and can be programmed to any angle, hence precise. The strings are tied to the Servos and the Servos rotate, thus pulling the Fingers individually per Servo.



[Above] Picture showing final Hand and Servo Motors

d. The Arduino Materials

The brain of the whole project is the Arduino.

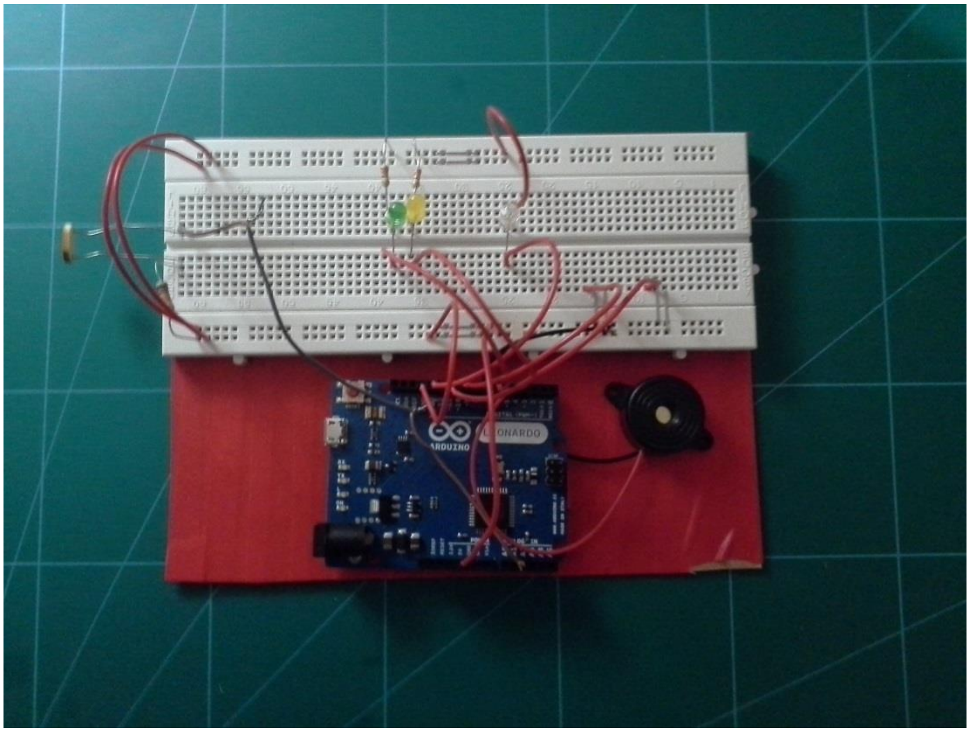
*"Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It's intended for artists, designers, hobbyists and any*one interested in creating *interactive objects or environments."*

~ *www.arduino.cc*

There are many models but the one we used is the Arduino Leonardo. This board has 14 Digital I/O pins (7 PWM pins), with 6 Analog Inputs. The board is programmable using the Arduino IDE. The programs are written in the C/C++ Programming Language. The Arduino got its power source from the Computer through a USB cable. We could have also used a DC Power Source, but we wanted to monitor the Arduino continuously. The whole project is powered from the Arduino. It provides an output voltage of 5V. This was enough for all of the components used.

In the project, we also used a Breadboard. This is a special board that allows people to test electronics simply by connecting them in to the holes. The holes are connected in certain patterns. It removes the job of soldering electronic equipment. To connect the holes, we used Jumper Wires, which are meant for the board.

On the board, we had some LEDs (Light Emitting Diodes) and a LDR (Light Dependent Resistors), along with connections from the Hand and Control Glove. We also used resistors for the LEDs (330Ω) and the LDR (10KΩ).



[Above] Picture showing Final Arduino and Breadboard

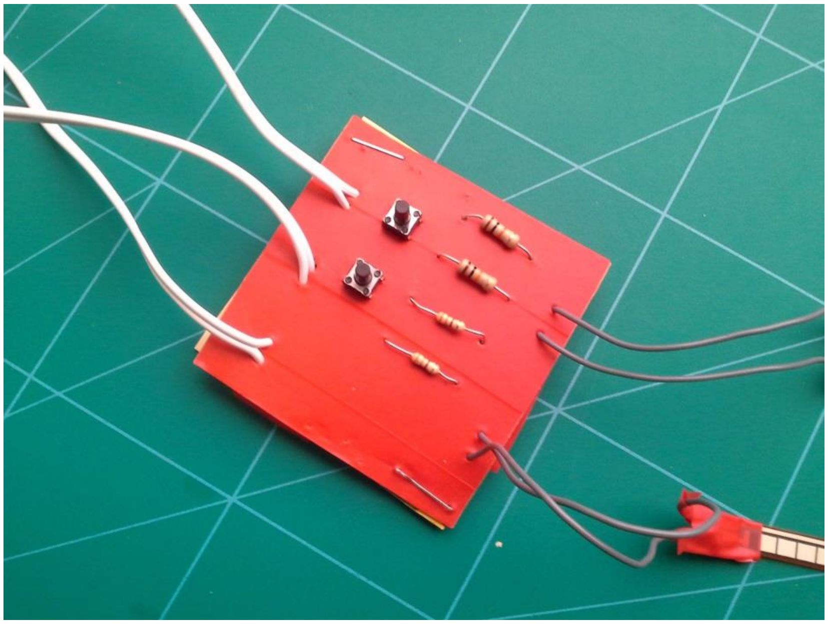
4. Method

a. Building the Control Glove

The control glove was the trickiest part of the build, as they were many connections to be made without the use of a solder-less breadboard. Firstly, we started off with stitching the pockets on the glove. We had an adult supervise this.

[Above] Picture showing the hand with the Stitched parts

After this, we started creating the "Control Center" of the glove. We firstly covered the cut cardboard piece with electrical taping. This ensured that no current would pass through the board. Next, we attached the resistors and buttons. We then made the simple connections. The trickiest part here was wiring the Flex Sensors to the Control Center. This was because the pins were too close together, and they were a separate part of the whole build. We had to separate pins using electrical wiring and then attach the wires. After this, we connected the six long wires that would go to the Arduino Breadboard.



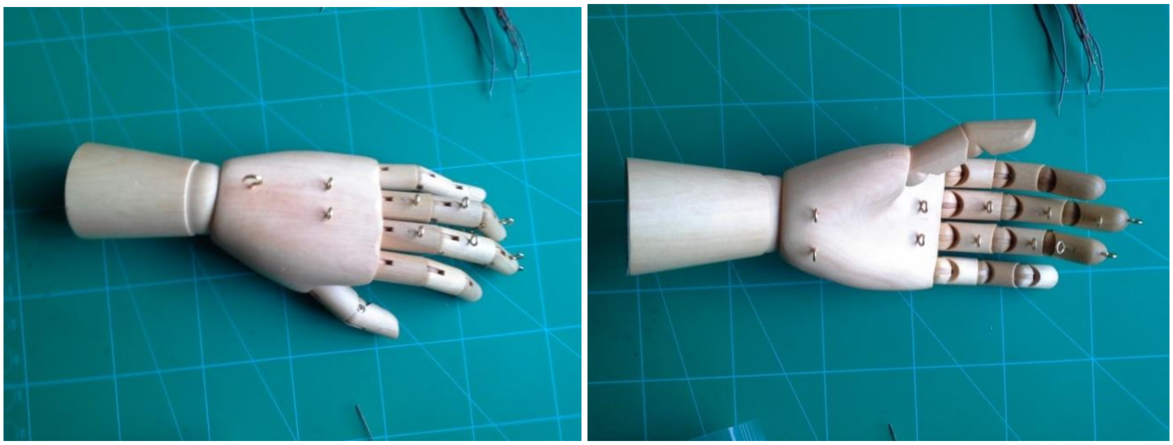
[Above] Picture showing the Control Center

The Control glove was complete with the control center. We tested this by uploading simple code to the Arduino and made sure that each piece read the correct values within the parameters. The Control glove passed all the tests.

[Above] Picture showing final Control Glove

b. Building the Hand

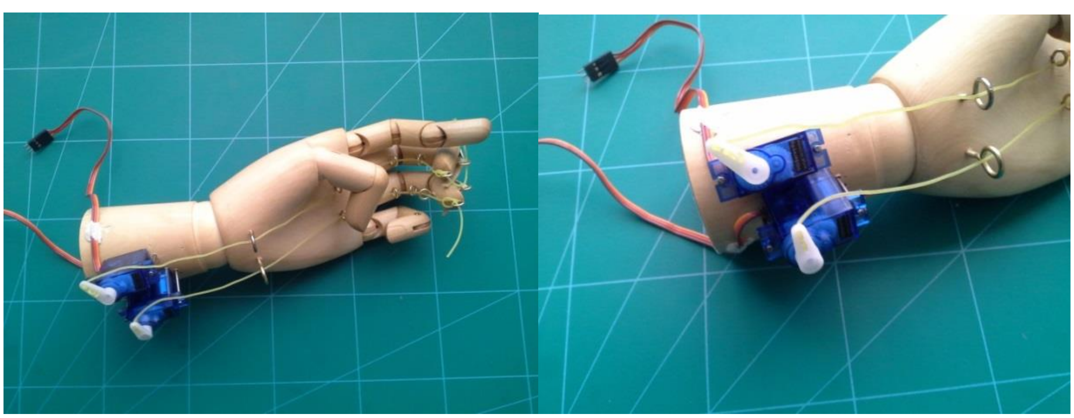
The hand we used was made out of tough wood. This required us to drill holes and then attach the screw-eyes. We made a few mistakes by breaking the screw-eyes half way in the wood but we eventually go everything done.



[Above] Pictures showing the Wooden Hand with screw-eyes

Next we attached the Servo Motors. This was by far the hardest part of this stage. We had to drill holes and carefully hammer the nails, through the Servo holes and make sure we don't hit the Servo motor as it was made out of plastic. The back nail was not supposed to tear the wires. Eventually, using care and precision, we hammered the nails and fixed the servo motors.

After this, we started threading the strings and rubber bands. We had to experiment to make sure that the finger was easy to bend but had enough force to come back up. Because this was a wooden mannequin hand, the joints were not free to move and hence it got stuck often. We soon reached a calibration we were happy with.

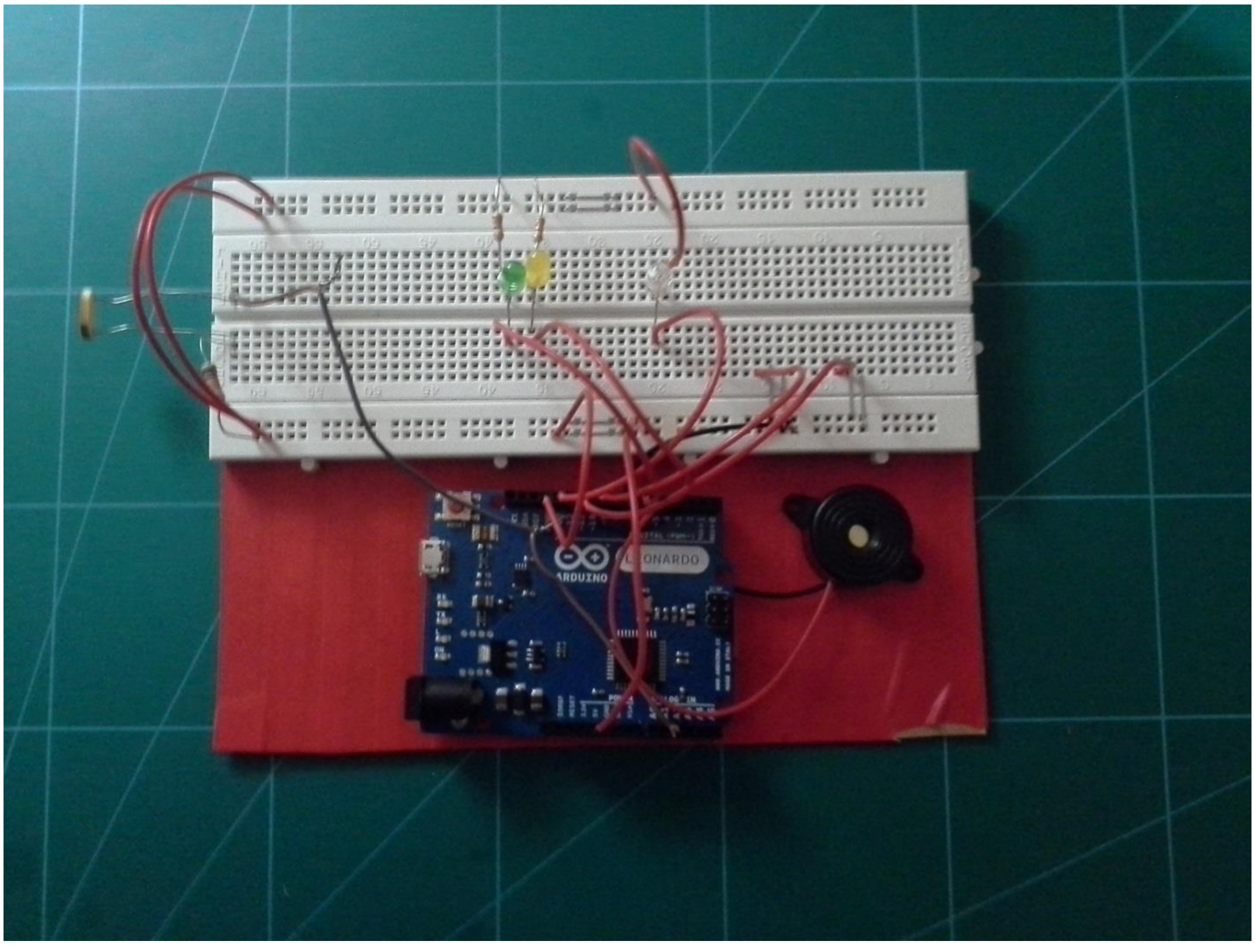


[Last Page, Last Pictures] Pictures of final Robot Arm

We tested the Servo Motors by uploading a simple program to it, and made sure both Servo Motors moved and worked. It passed all tests and now we moved to the third and final stage; wiring everything up and writing the source code.

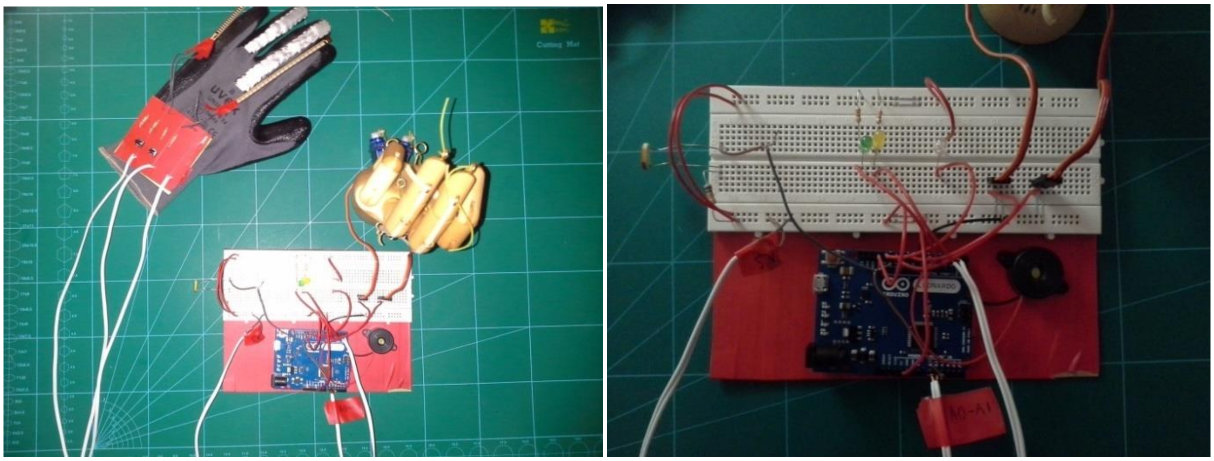
c. Connecting the Arduino and Programming the Code

Now we needed all the wires to connect to a central breadboard. We did this by first connecting all the Ground Wires and the 5 Volts together. After this, we then did the components section and installed all the components one by one following the design documents. We did not put the Control Glove and Hand connections yet.



[Above] Picture showing the wire-up of the Arduino

After this, we then connected the Hand connection and the Control Glove. We connected the Control Glove directly to avoid any unnecessary wiring. After that, we tested both connections with simple code and it worked!



[Above] Pictures showing the Final wiring

Now that we had tested the code, we needed to write the source code for the whole project to work. The code is below with some minor explanation in the pictures. The full explanation for the code is in the **Explanation** section.

Now, this is the Arduino IDE. It is where all code is written in and then uploaded to the Arduino. The programming language is C/C++. Arduino code files are called *Sketches*.

The Code: *TheHandSourceCode.ino*

```
/* ----------------------------------------------------------------------
                                                     THE HAND SOURCE CODE

        All code in this file belongs to Amrit Amar and no one has the right
        to reproduce this code without permission from Amrit Amar.

        This code is for the BOTHO COLLEGE ICT LINKZ CHALLENGE 2013. The
        project is called "The Hand". This code calculates the Input from
        connected Flex Sensors and produces the Output to Servos, which
        control Finger Movement on a Artificial Hand. The code also Implements
        2 Buttons which control various functions.

        -------------------------------------------------------------------- */

// ----- CODE DECLARATIONS ----- \\

//Imports Servo Library for using it's Functions
#include <Servo.h>

//Creates 2 Integers to store Mapped Servo Positions
int servoPosition1;
int servoPosition2;

//Creates 2 Servo Objects
Servo servo1;
Servo servo2;
```

```
//Creates 2 Integers to denote the Pins to which the Flex Sensors are attached to
int flexSensor1 = 0;
int flexSensor2 = 1;

//Create 2 Integers to denote the Values taken from Flex Sensor Readings
int flexSensorValue1;
int flexSensorValue2;

//Creates 2 Integers to denote the Pins to which the Buttons are attached to
int pushButton1 = 2;
int pushButton2 = 3;

//This creates 2 Integers to denote the Values taken from Button Readings
int pushButtonReadings1;
int pushButtonReadings2;

//Creates 2 Integers to denote the Pins to which the Servos are attached to
int servoPin1 = 9; //This Pin is a PWM
int servoPin2 = 11; //This Pin is a PWM

//Create 2 Integers to denote the Pins to which a LED and a Speaker is attached to
int lightPin = 13;
int soundPin = 5;

//Creates 2 Integers to denote the Pins to which the Flex Sensors are attached to
int ldrSensor = 2;

//Creates 1 Integer to denote the Values taken from the LDR Readings
int ldrValue;

//Create 2 Integers to denote the Pins for the LDR Light LED connections
int ldrHigh = 6;
int ldrLow = 4;


// ----- CODE METHODS ----- \\

//This is the Setup Method.
//This is run first by Any Arduino.
void setup() {

        //This starts the Serial Monitor to start, at 9600 Bauds
        Serial.begin(9600);

        //This attaches each Servo to the Given Pin number
        servo1.attach(servoPin1); //Attaches a Servo to Pin 9 which is PWM
        servo2.attach(servoPin2); //Attached a Servo to Pin 11 which is PWM

        //Taking the Pin Number, it declares it either an INPUT or an OUTPUT
        pinMode(pushButton1, INPUT); //Pin 2 is now an INPUT.
        pinMode(pushButton2, INPUT); //Pin 3 is now an INPUT.
        pinMode(lightPin, OUTPUT); //Pin 13 is now an OUTPUT.
        pinMode(soundPin, OUTPUT);//Pin 5 is now an OUTPUT.
        pinMode(ldrHigh, OUTPUT);//Pin 6 is now an OUTPUT.
        pinMode(ldrLow, OUTPUT);//Pin 4 is now an OUTPUT.

} //End of Setup Method
```

```
//This is the Loop Method.
//This is run by the Arduino over and over again after the Setup Method
void loop() {

        //I call a method to get the Readings of the Push Buttons
        //This is a Digital Read Function because the Buttons are connected to the Digital Pins
        pushButtonReadings1 = readDigitalValues(pushButton1); //Store Value in Variable
        pushButtonReadings2 = readDigitalValues(pushButton2); //Store Value in Variable

        //I call a method to get the Readings of the Flex Sensors and the LDR
        //This is a Analog Read Function because the Flex Sensors and the LDR are connected to the Analog Pins
        flexSensorValue2 = readAnalogValues(flexSensor2); //Store Value in Variable
        flexSensorValue1 = readAnalogValues(flexSensor1); //Store Value in Variable
        ldrValue = readAnalogValues(ldrSensor); //Store Value in Variable

        //I call a method to Map the Values of the Sensors to the Servos
        //This uses the Readings from the previous Methods
        //This uses the map() function
        mapServos();

        //I call a Method to get the mapped values and assign them to the Servos
        //This uses the Readings from the previous Methods
        //This uses the .write function which is part of the Servo Library
        writeServos(servo1, servoPosition1);
        writeServos(servo2, servoPosition2);

        //I call a Method to check the Button State and Switch on the Light/Sound
        //This uses Readings from the previous Methods
        //This uses an If function
        checkStateButton(pushButtonReadings1, lightPin);
        checkStateButton(pushButtonReadings2, soundPin);

        //I call a Method to check the LDR and then light up the correct lights
        //This uses Readings from the previous Methods
        //This uses an If function
        checkLDRState(ldrValue);

        //I call a Method to Print to the Monitor
        //This uses all the Readings
        //Prints Lines of Readings
        printMonitor();


        //This Delays the Code for 15 Milliseconds.
        //This allows the Serial Monitor to print the lines and for the Servos to Get to their Angles.
        delay(15);


} //End of Loop Method

//Method for Calling Push Button Values
//Public Method that returns an Integer
int readDigitalValues(int Pin) {

        int Value; //Create new Integer called "Value"
        Value = digitalRead(Pin); //Get the Digital Value from given Pin Number

        return Value; //Return the Value

} //End of Method
```

```
//Method for Calling Analog Values
//Public Method that returns an Integer
int readAnalogValues(int Pin) {

        int Value; //Create new Integer called "Value"
        Value = analogRead(Pin); //Get the Analog Value from given Pin Number

        return Value; //Return the Value

} //End of Method


//This Method will Map the Flex Sensor Values to the Servo Position
//Public Method that returns nothing (Setter Method)
void mapServos() {

        //Gets the Variable and assigns the Angle mapped
        servoPosition1 = map(flexSensorValue1, 240, 150, 0, 179); //Maps and stores Variable
        servoPosition2 = map(flexSensorValue2, 467, 300, 0, 179); //Maps and stores Variable

} //End of Method

//This Method will set the Servo position
//Public Method that returns nothing (Setter Method)
void writeServos(Servo servoObject, int Value) {

        servoObject.write(Value); //Writes Angle to Servo

} //End of Method

//This Method will check the State of a button and Write Accordingly
//Public Method that returns nothing (Setter Method)
void checkStateButton(int Value, int Pin) {

        if (Value == 1) //Check if the Button is being Clicked
        { //If True then

                digitalWrite(Pin, HIGH); //Send out a current on the Pin

        } else { //Otherwise

                digitalWrite(Pin, LOW); //Do Not Send out a current on the Pin

        } //End of Checking and if Statement

} //End of Method


//This Method will check the LDR Value and Write Accordingly
//Public Method that returns nothing (Setter Method)
void checkLDRState(int Value) {

        if (Value > 300) //Check if LDR is more than Relevant Light Level
        { //If True then

          digitalWrite(ldrLow, LOW); //Set Orange LED off
          digitalWrite(ldrHigh, HIGH); //Set Green LED on


        } else { //Otherwise

          digitalWrite(ldrHigh, LOW); //Set Green LED off
          digitalWrite(ldrLow, HIGH); //Set Orange LED on

        } //End of Checking and if Statement

} //End of Method
```

```
//This Method will Print to the Monitor
//Public Method that Returns nothing (Print Method)
void printMonitor() {

        //This code writes Values to the Serial Monitor.
        //This is useful for Debugging and Tweaking Values
        Serial.print("Finger 1 Sensor Input = " );
        Serial.print(flexSensorValue1);
        Serial.print("\t Finger 2 Sensor Input = ");
        Serial.print(flexSensorValue2);
        Serial.print("\t Servo 1 Output = ");
        Serial.print(servoPosition1);
        Serial.print("\t Servo 2 Output = ");
        Serial.print(servoPosition2);
        Serial.print("\t Button 1 Input = ");
        Serial.print(pushButtonReadings1);
        Serial.print("\t Button 2 Input = ");
        Serial.print(pushButtonReadings2);
        Serial.print("\t LDR Input = ");
        Serial.println(ldrValue);

} //End of Method
```

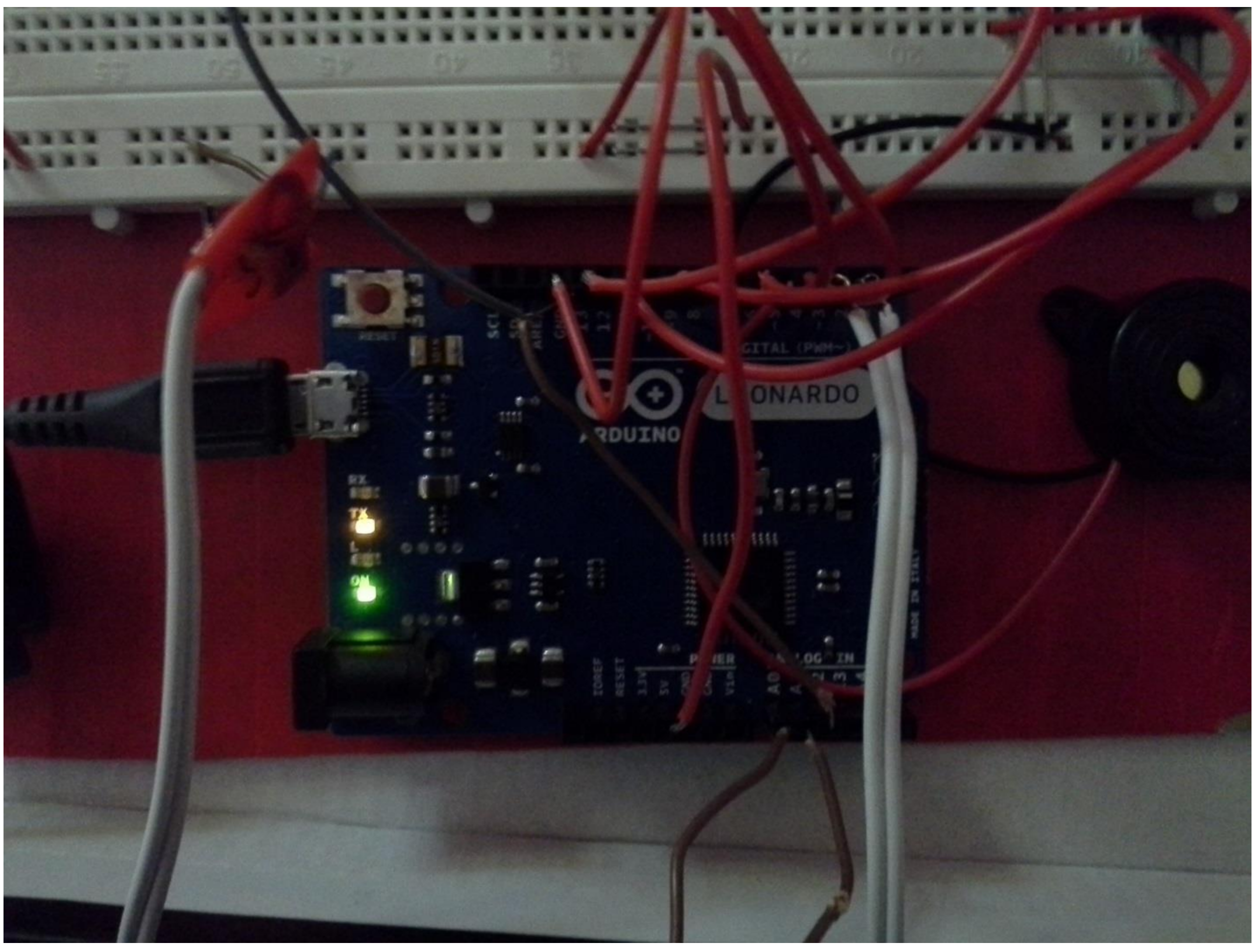
[Above] Picture showing all the Code used in the Project

5. Testing

a. Powering Up and Uploading the Code

After we finished making all the components, we decided to test everything. Firstly, we connected the Arduino to the Computer and made sure that the main switch was off. The Arduino powered up, but didn't do anything because there was no Sketch on the board.

We then uploaded the code to the board. There were no errors and it uploaded successfully. Now we proceeded to test everything.

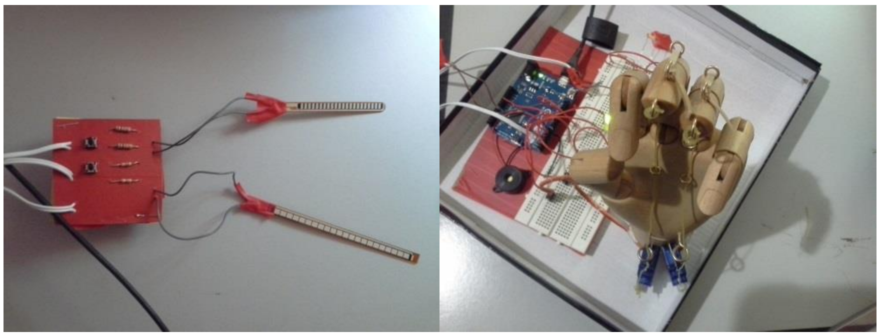


[Above] Picture showing Arduino Rebooting

b. Finger Movement Testing

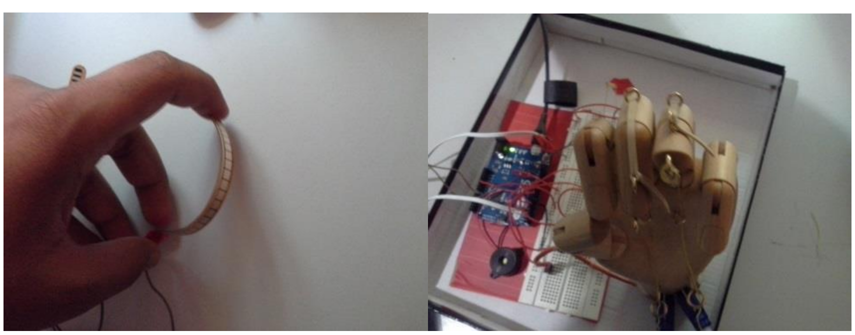
To test this, I first wore the glove then switched on the connections. We then proceeded to test each finger individually. I did not put on a glove for this test as I needed free hands to make sure that this was working.

I first started with the Ring Finger and made sure it moved. Then I did the Middle Finger and made sure it moved. Then I did them both at the same time and made sure they moved in synchronization. It worked! The Serial Monitor was up for me to inspect the readings but there were no abnormal readings detected. The test was successful.

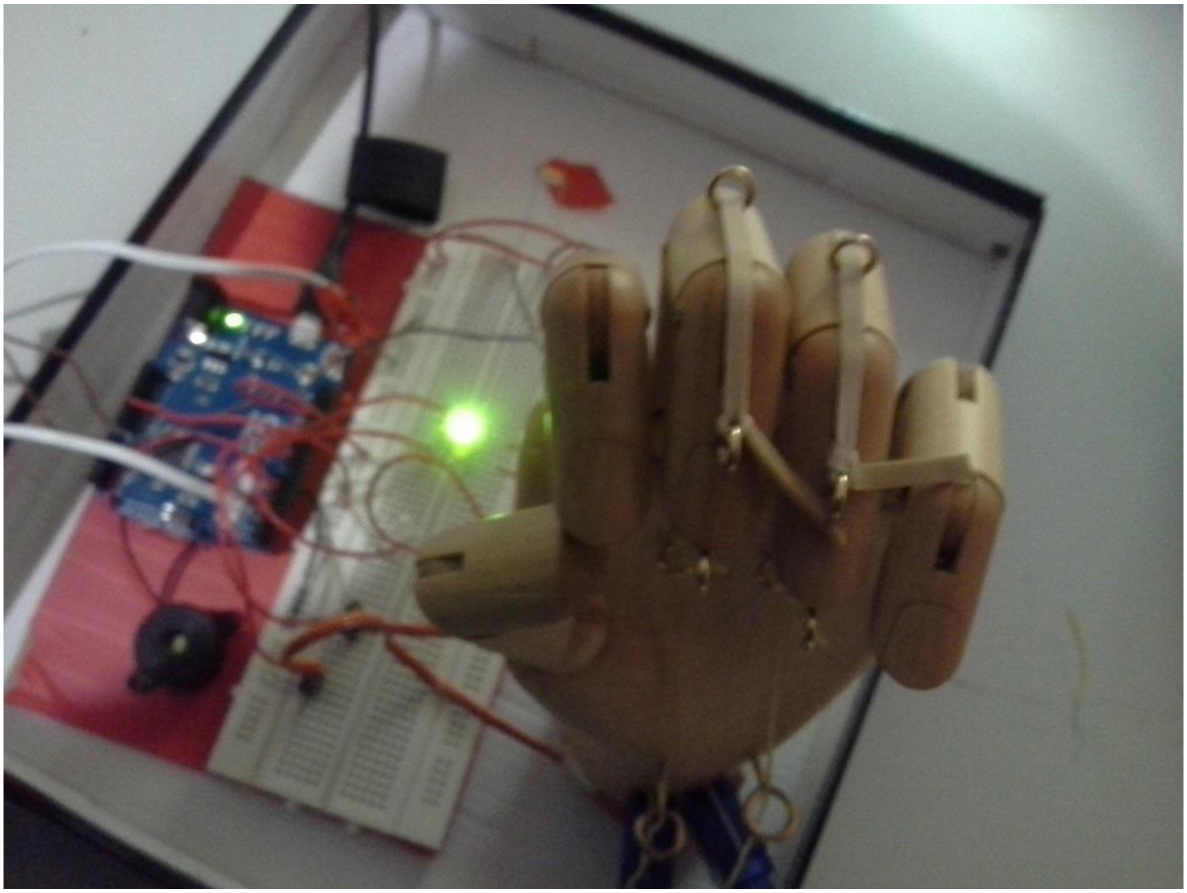


[Above] No Flex

[Above] Ring Finger Flex



[Last Page, Last Picture] Middle Finger Flex

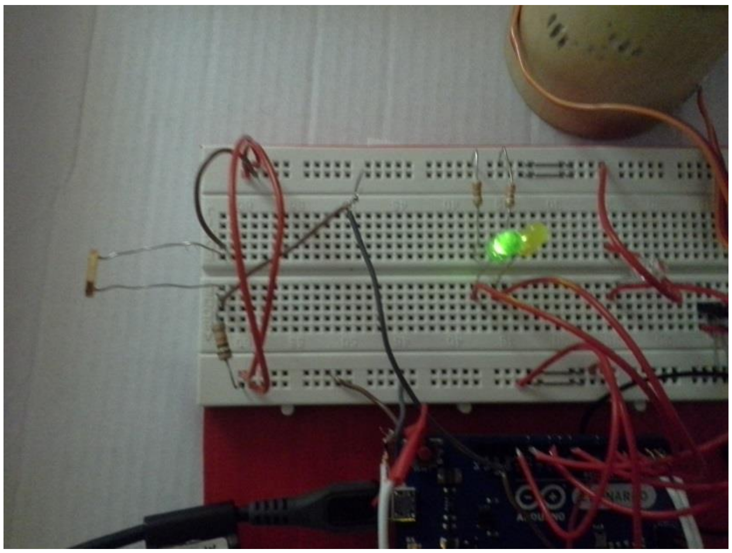


[Above] Full Finger Flex

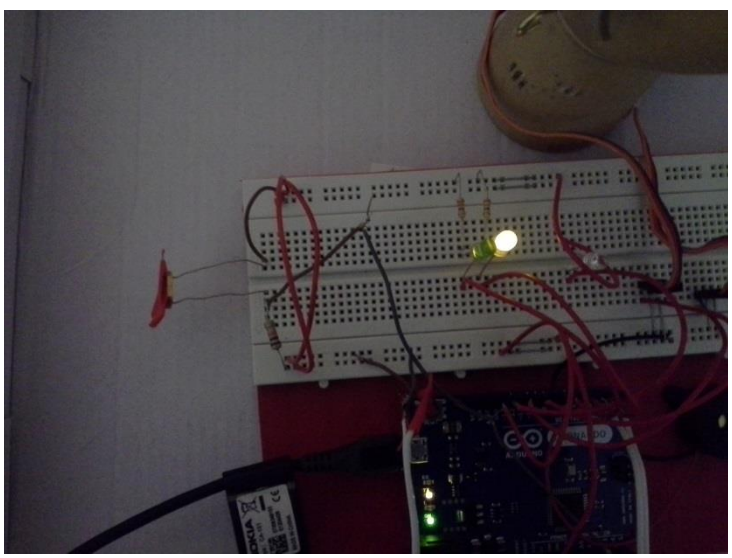
c. Light Detector Testing

To test this, I went to a dark place, where the yellow LED was on due to the darkness. I then used a torch on the LDR and checked if it turned Green when the light level was reasonable.

This test showed that whenever the hand is in a dark place where visibility is limited; the LED would light up, showing the wearer the light level of the area.



[Above] Picture showing Torch on LDR lighting up the Green LED



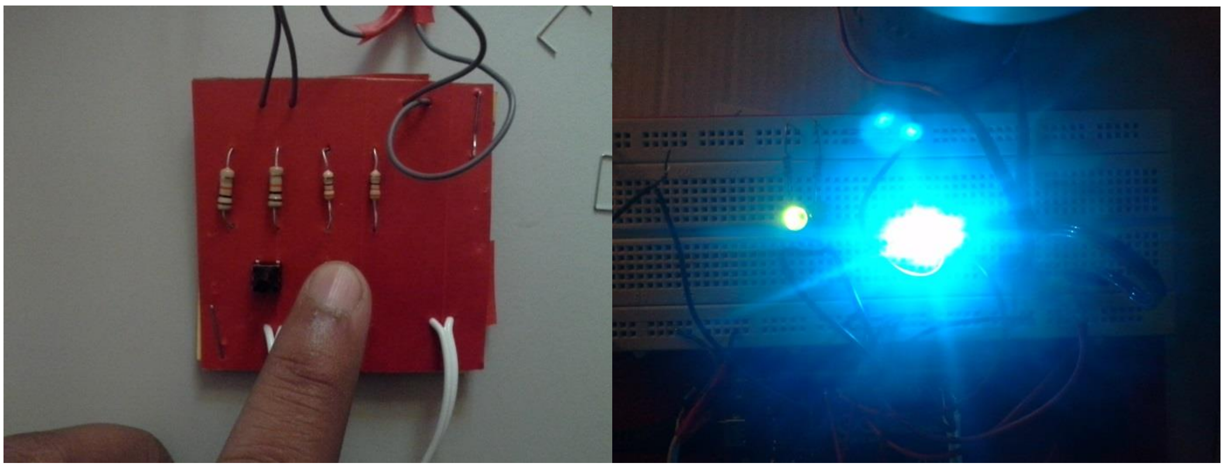
[Above] Picture showing the Torch on the LDR lighting up the Yellow LED

d. Button Testing

i. Lighting

Testing this was very straight forward. I just clicked the button on my control glove and checked if the LED lit up on the Breadboard.

This showed that a light can be used in order to check the position of the hand in a dark area. It can also be used to communicate between 2 people in Morse code.



[Above] Picture showing LED lit up

ii. Sound

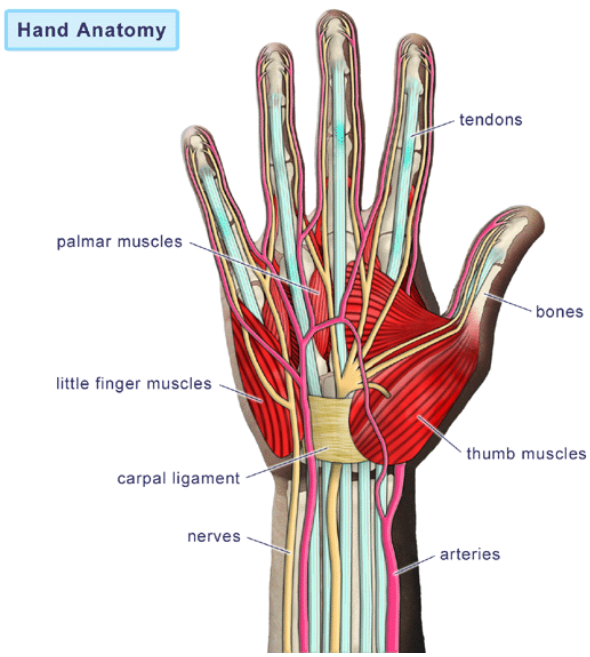
Testing this was also very straight forward. I just clicked the button on my control glove and checked if the speaker produced sound.

This showed that a sound can be used in order to check the position of the hand in a dark area or an area where the hand is not easily found. It can also be used to communicate between 2 people in Morse code, where the listening person has limited sight.

6. Explanation

a. The Biological Hand and our Hand

To move the hand using Servo Motors, we first studied the actual hand and came up with many ways to replicate the movement. The biological hand we have is special. It cannot be replicated exactly but closely.



[Above] Picture of the anatomy of a hand from *healthfavo.com*

The biological hand has tendons attached to each bone and joint. These tendons pull when we lift anything heavy, thus flexing the fingers causing them to wrap around the object. This is called *power grip*.

Precision grip is another way of gripping, where each joint is controlled individually and can grasp the smallest of objects perfectly. Sadly, this was not possible by our model. However, in the future, we hope to implement this in a newer model.

So we went with *Power Grip*, where the fingers wrap around the object. Now, how do we replicate this? Well, each tendon is connected to the joint. This limits it from breaking out of the hand. For the tendon, we used string, and for connecting it to the individual joints, we used Eye hooks. This allowed the hand to bend forward whenever force was applied on the string.

We weren’t done yet though. The hand moved forward, but not backwards, to its natural position. We needed something elastic to hold the fingers back. We used rubber bands for this, connecting them in the same way, except at the back of the fingers and hand. This allowed us to pull the tendon (string) and the finger would bend forward, but when we released it, the finger would return back to its original position because of the rubber bands. It was simple yet effective.

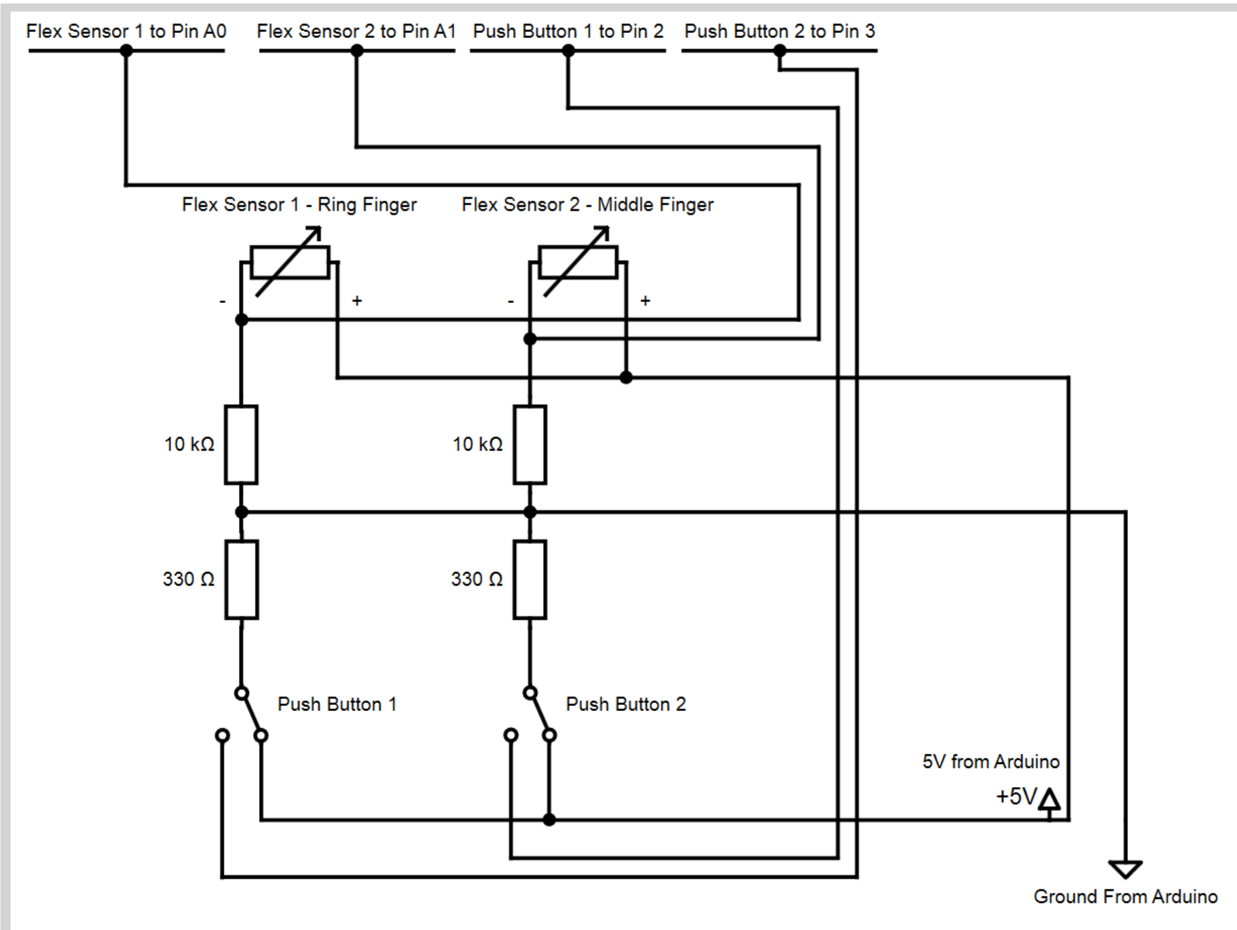
[Above] The Hand Design

b. Circuit Schematics of the Project

The circuit schematic of the whole project is here. It connects everything to the Arduino. The Arduino handles all the components through code.

i. The Control Glove

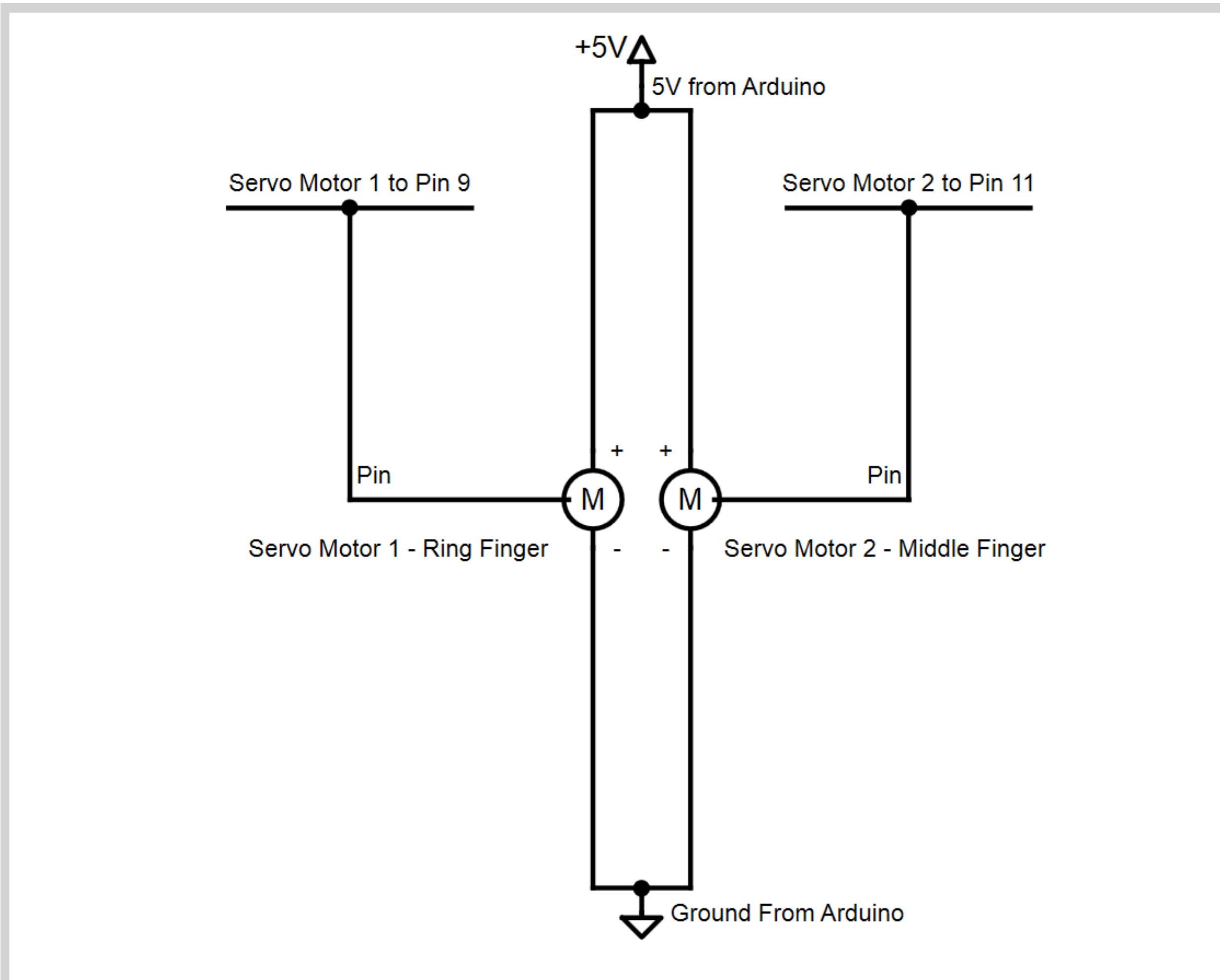
The Flex Sensors are types of Variable Resistors. They change their value depending on how much they bend. Both Flex Sensors are grounded to the Arduino and are also being supplied 5V by the Arduino. The Push Buttons are switches that change current from Ground to 5V, to Ground to Pin Number. The resistors are there in place to control the current given to each component so that the components don't get destroyed.



[Above] The Circuit Schematic for the Control Glove

ii. The Hand

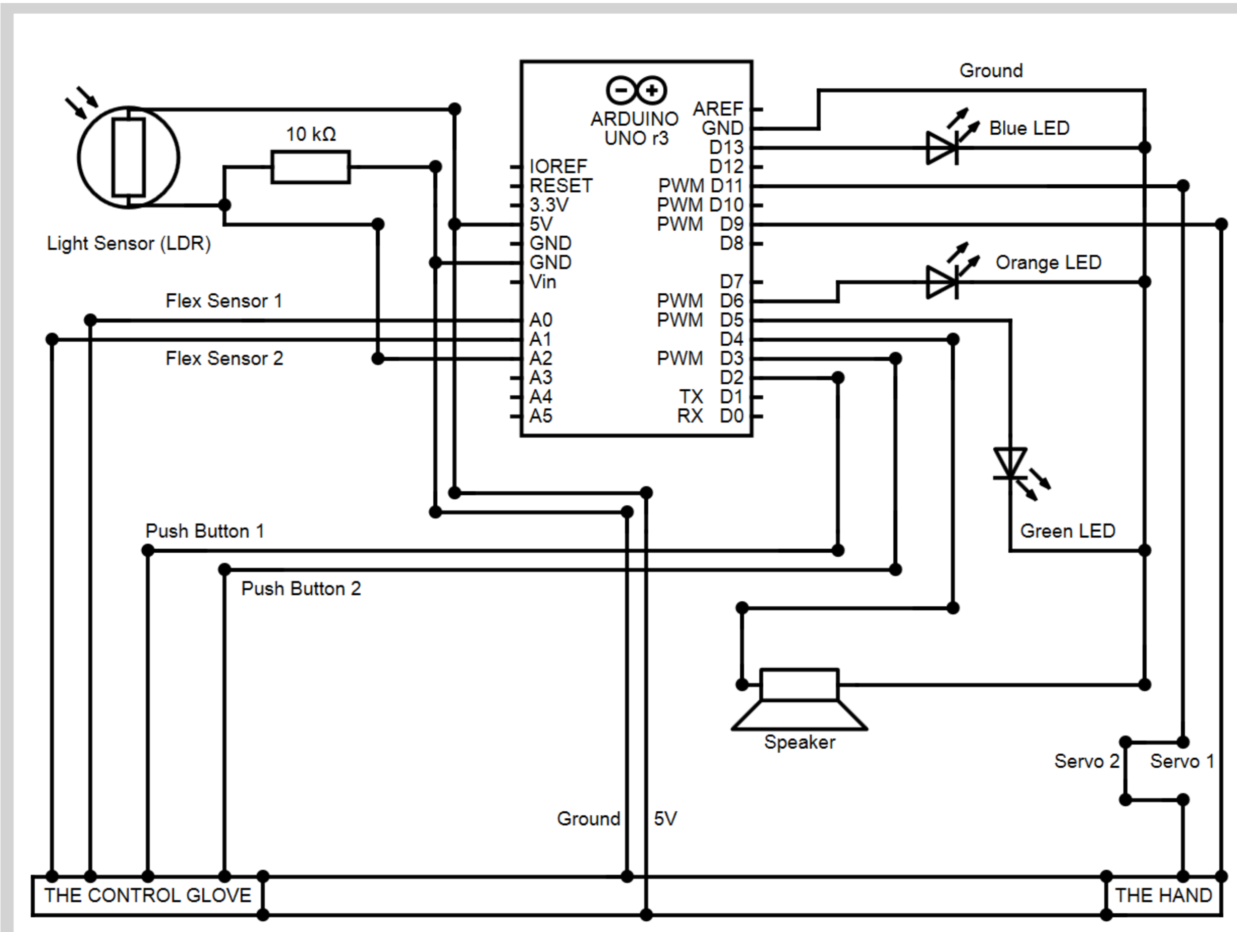
The Servo Motors have 3 wires coming out of them, the Black (Ground), the Red (Power) and Brown (Connection Pin). These Pins were directly connected to the Arduino as these Servo Motors operated at 5V, there was no reason for Resistors.



[Above] The Circuit Schematic for the Hand

iii. The Arduino

All the wires connected to the Arduino. The Arduino also controlled the LDR and the LED's including the Speaker. This was all controlled by Code.



[Above] The Circuit Schematic for the Arduino

c. Code Overview of the Project

I first started with the Global Declarations. I had to declare everything at a global level because of the fact that I used custom-made methods that require all the variables to be declared for the whole project. This basically makes sure that I have no code errors or invalid declarations.

After that, I started with the Setup Code. This was basically to create all the objects for use in the code, and make sure all the Pins were either INPUT or OUTPUT. This made sure that the Code started off without any errors carrying from any previous code. At every reset, this code will run, thus saving me the time to declare the Pin mode at the time the Pin is going to be used.

Next was the Loop Code. This is the main body of the whole code. This is where all the code ran. The loop function basically runs continuously, running code that checks and sets values on Variables. At first, this method was a heap of unreadable and unorganized code, which is a bad programming practice. I then grouped every single code block into one general method for that block. This reduced the lines of Code considerably, while still keeping the purpose of the code.

The first method I made was for reading the Push Button State, to see whether it was clicked or not. This gave a 1 (clicked) or 0 (not-clicked) readings.

The second method I created was for reading the Flex Sensors and Light Sensor through the analog input pins. This read values from the given components.

The third method I made was for mapping the values from the Flex Sensors to the Angles of the Servos. The result was a number from 0 to 179 mapped from the Flex Sensors.

The forth method I created was for setting the Servo to the mapped angle. This set the Servo to the given Angle (0 to 179).

The fifth method I made was for checking the Button State and given it, produces the required output. This basically turned the light and sound on if the button was on.

The sixth method I made was for checking the LDR values and therefore setting the appropriate light to turn on. I used the value of 300 just for an example; it can be tweaked to adjust to any environment.

The last method I made was for printing all the information to the Serial Monitor built-in with the Arduino IDE. This allowed to me check the values given by all the methods and components and to troubleshoot when a certain function was not working.

The code worked perfectly with all the hardware. The code did what it was meant to do.

7. Applications and Limitations

a. Applications of our Solution

i. Did it work?

The working model replicated the movement of the ring and middle finger on via the flex sensors on the Control Glove as predicted. However, the movement was limited slightly by the wooden mannequin used as a basis for the Robot Hand. The joints had to be filed repeatedly but still presented a problem in terms of friction. Furthermore, calibration between the rubber bands and the fishing wire is something that needs to be further worked on to allow the fingers to completely stand up straight and bend completely.

ii. Applications

Aside from mining and space exploration which have been discussed extensively thus far, the Hand has many other applications that can be considered in context of human safety issues.

- Work in electronics: the Hand can be used in repairing and dealing with dangerous and even live wire depending on the material used to construct, in this case wood. This is especially useful for highly dangerous situations or when is impossible or even undesirable to disable the current in an electric circuit.
- Can be used to substitute humans in radioactive environments such as nuclear power plants, this would be especially useful in times of emergency or even basic repair.
- Medicine: the Hand can be used for operations where it is not possible to have an actual doctor present or as a long-distance learning teaching tool or to allow experts from other countries to perform surgeries from far away. The Hand can also be used in circumstances where it is not advisable for a doctor to perform a surgery directly, e.g. an environment with serious risk of contraction of infectious disease.

b. Limitations and Improvements

i. Limitations and how we plan to counter them

The most significant limitation in the current working model is that of the lack of a wireless control module. Another limitation is that the current working model is limited in the movement of fingers only, and does not support the movement of the wrist. Also a problem is the issue of the opposable thumb and how it is to be designed and coded. Finally for the success of the project in future implementation a video camera needs to be somehow used in conjunction with the model to support the concept of human safety and the ability to work from afar.

To counter the wireless control limitation, the Arduino can be upgraded with shields. Shields are basic hardware that enhances the functionalities of the Arduino. In this case, the wireless shield would allow a computer and an Arduino to communicate wirelessly. Adding another Arduino and the shield to the project can allow the Arduinos to communicate directly without the use of a computer. This allows the control glove to have its own Arduino and the Hand to have its own thus allowing adding more features to the |Arduino.

The finger movement is limited because of the Servo Motor’s arm length. Adding a bigger Servo with a bigger turn radius would allow us to completely move the fingers. The wrist rotation can be controlled with another Servo at the base and the bending of the Wrist can be done with a motor acting as a pivot between the wrist and the hand. The opposable thumb can be redesigned and implemented easily with proper construction and design. A video camera can be attached to the back of the hand. Small cameras can be easily attached to anywhere in the hand.

8. Conclusion

The project started off as a simple idea, "*How can we make machinery easier and safer to control?"* Throughout the whole project, we came across many different situations and problems, where we had to think critically about the problem and create a solution. We built a Hand to show a part of our solution.

The project was successful and we will continue to enhance the functionalities of it day by day. The project was a prototype that was a significant step to solving the problem.

In conclusion, what should be noted is how the working model can be seen to contribute to the future of more intuitively controlled machinery via human controls.