Table of Contents

Chapter 1

Introduction

Science and Technology have made Human life addictive to comfort but still there exists an underprivileged group of people who are fighting for finding an innovative way that can make the process of communication easier for them. According to the World Health Organization, about 285 million people in the world are blind, 300 million are deaf and 1 million are dumb. According to census 2011, In India there are 1,640,868 citizens who can't speak as well as 1,261,722 citizens who can't listen. More than 70% of Deaf population of India is working in Government as well as Private sectors. For communication they are dependent on Sign Languages. In India most popularly Indian Sign Language is used. In rest of the population of India very few citizens are able to use Indian Sign Language, as they really don't need to learn sign language. This causes a communication barrier between Deaf Dumb and Normal person. This leads to disqualification of hearing impaired from main stream of the society. To overcome this problem a communication assistant is required, to convert Sign Language to auditory speech. Round the world people who are deaf, mute or blind find it hard to communicate normally with people. This project investigates a new way that can be productize so that a new gadget can be developed that can bridge the gap in communication among differently abled people who suffer from any of the possible combinations of the disability of Blindness, Deafness and Dumbness. For this we are looking for some sort of wearable technology that can satisfy our purpose. The device focuses on eliminating this problem by providing a tool to assist the communication. Our target population is mostly well versed with ASL (American Sign Language) which is the convention followed worldwide. The device has a dataset loaded with words and alphabets that can be invoked by hand symbols/signs. The dataset can be altered by the user providing him/her with the option of personalizing the tool and enhancing the experience and utility.

1.1 Aim of the Project

In this project, we are going to propose a new wearable communication device in an effort to bridge the gap in the process of communication between the Blind, Deaf and Dumb people. The prototype will make use of the flex sensor and Arduino microcontroller to provide a means of communication to differently-abled people having one or all of the above mention disabilities. It is assumed that a person who is deaf is also dumb but vice versa is not true. The device can convert the hand gestures to desired letter/word/phrase and then display it on a 16*2 LCD as well as can send this letter/word/phrase to a mobile device using wireless communication for conversion of text to speech.

1.2 Comparison of Existing and Proposed System

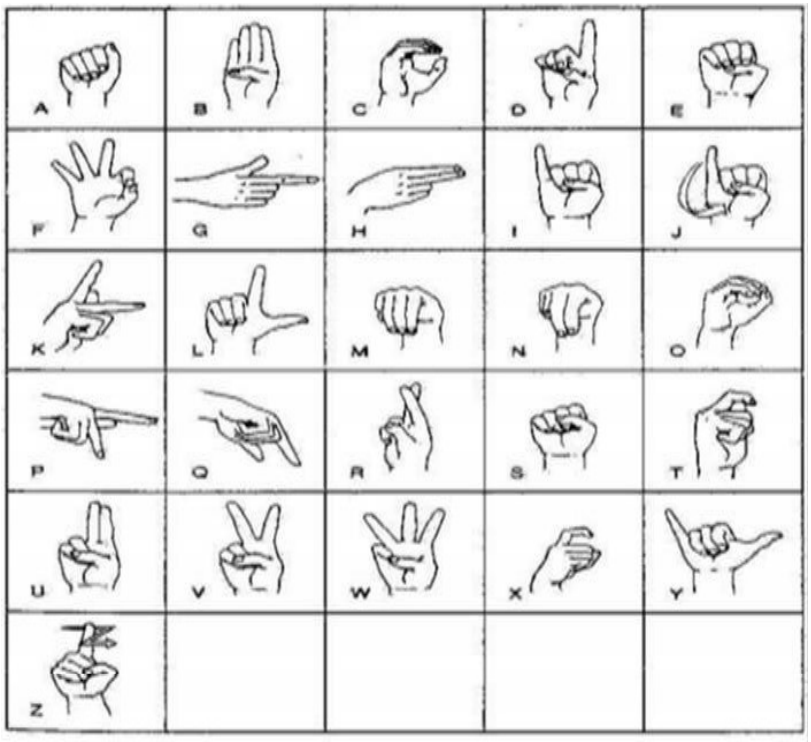
Existing System	Proposed System
In Existing system American Sign Language, Is the only way of communication amongst impaired people.	The main motive of our prototype is to introduce an idea that can help to ease the way the disabled people use to communicate with each other or with the world.
But general people can't understand the Sign language so it becomes difficult for An impaired to communicate with them.	In this project an electronic speaking system would be developed to ease the communication process of impaired people.
There is no system that enables communication between the impaired and the world.	A glove-based prototype would be developed which consists of five flex sensors enabling communication between the impaired and the world.

Chapter 2

Review of Literature

2.1 American Sign Language

American Sign Language (ASL) is a natural language that serves as the predominant sign language of deaf communities the United States and most of Anglophone Canada. Besides North America, dialects of ASL and ASL-based Creoles are used in many countries around the world, including much of West Africa and parts of Southeast Asia. ASL is also widely learned as a second language.



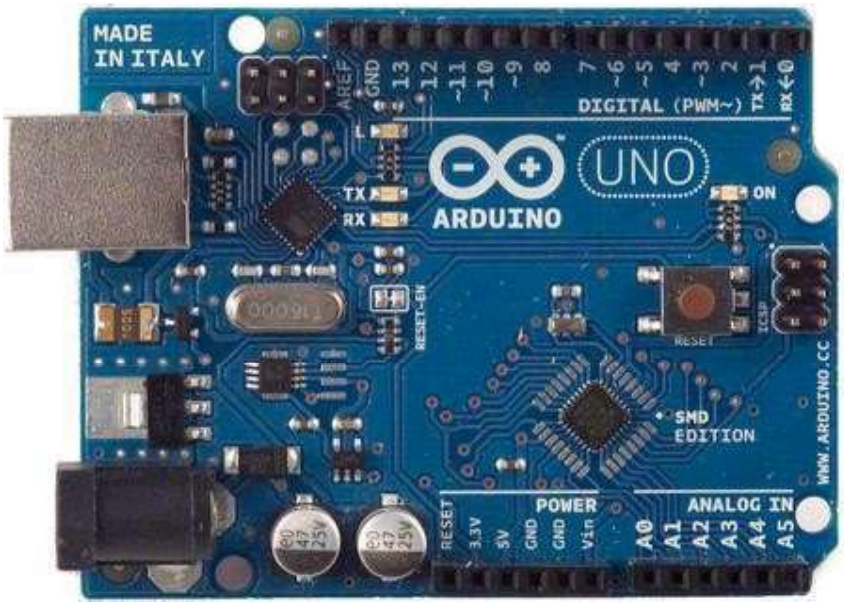
The ASL (American Sign language) consist of gestures. These are mainly of types:

Static Hand Gestures: Static hand gestures are fixed with respect to time, these gestures do not depend on motion.

Dynamic Hand Gestures: In dynamic hand gestures movement of some portion of the hand is involved with respect to time. These gestures require motion and are based on the trajectory that is formed during the motion in question.

2.2 Arduino UNO

The Arduino UNO is an open-source microcontroller board based on the microchip ATmega328p microcontroller and developed by Arduino. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 Digital pins, 6 Analog pins, and programmable with the Arduino IDE (Integrated Development Environment) via a type B USB cable. It can be powered by a USB cable or by an external 9-volt battery, though it accepts voltages between 7 and 20 volts.



2.3 Flex Sensors

Flex senor is a such type of sensor which measures the amount of bending or defection. Actually, it is made up with carbon surface on a plastic strip when this strip is bend or deflect then the resistance of this flex sensor is changed therefore it is also called bend sensor. Because its changing resistance is directly proportional to amount of bend therefore it can also be used as a goniometer. There are two types of flex sensors are currently used in industry according to their size first one is 2.2-inch flex sensor and second one 4.5-inch flex sensor. The resistance and size of both are different but working principle is same. So, the appropriate size is chosen according to requirement. Here we are going to discuss 2.2-inch

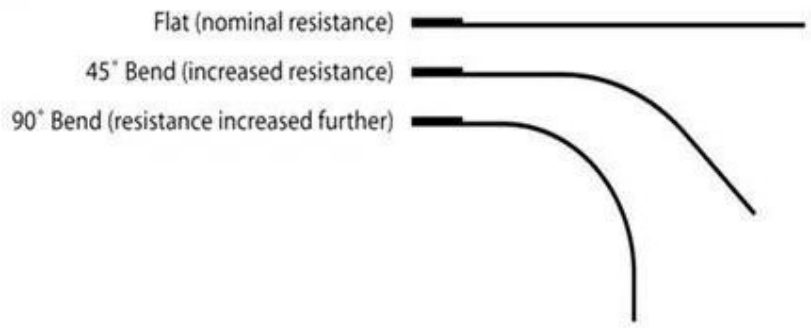
flex sensor. These types of flex sensor have been using in different applications such as in security system, rehabilitation, computer interface, music interface, servo motor control, intensity control or where the user wants to change the resistance during bending.

Pin Configuration of Flex Sensor

Flex sensor is a two pin or two terminal device such as p1 and p2. It does not have any polarized terminal like capacitor or diode, means there is no positive or negative terminal. For power on this flex sensor 3.3V to 5V dc voltage are applied on its terminals and these voltages are gained from interfacing device which could be any type of controller.

Working Principle of Flex Sensor

This sensor device works on the principle of bending strip means when its strip is bent its resistance changes which is measured with any controller. In simple words, this sensor works like a variable resistance whose resistance is changed when it is bent. This change in resistance depends upon surface linearity, means the resistance of this sensor would be different at different angle such as the resistance would be different. When it is flat, the resistance would be different from when it is bent by 45^0, similarly the resistance would be different when it is bent by 90^0. If we divide it into three cases. In case one when there is no bend in this flex sensor then resistance would be comparatively less. In case two when the bend in flex sensor is 45^0 then the resistance would be double as compared to case one. Similarly, in case three when the bend in the sensor is 90^0 then the resistance would be double as compared to case two and four time as compared to case one. So, if we talk about ratio then this resistance is directly proportional to bending angle means it is increased as the angle is increased.



2.4 Bluetooth Module (HC-05)

HC-05 module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup. The HC-05 Bluetooth Module can be used in a Master or Slave configuration, making it a great solution for wireless communication. This serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation with complete 2.4GHz radio transceiver and baseband. It uses CSR Bluecore 04-External single chip Bluetooth system with CMOS technology and with AFH (Adaptive Frequency Hopping Feature).

The Bluetooth module HC-05 is a MASTER/SLAVE module. By default the factory setting is SLAVE. The Role of the module (Master or Slave) can be configured only by AT COMMANDS. The slave modules cannot initiate a connection to another Bluetooth device, but can accept connections. Master module can initiate a connection to other devices. The user can use it simply for a serial port replacement to establish connection between MCU and GPS, PC to your embedded project, etc.

Bluetooth Pin Description

The HC-05 Bluetooth Module has 6pins. They are as follows:

ENABLE:

When enable is pulled LOW, the module is disabled which means the module will not turn on and it fails to communicate. When enable is left open or connected to 3.3V, the module is enabled i.e. the module remains on and communication also takes place.

VCC:

Supply Voltage 3.3V to 5V

GND:

Ground pin

TXD & RXD:

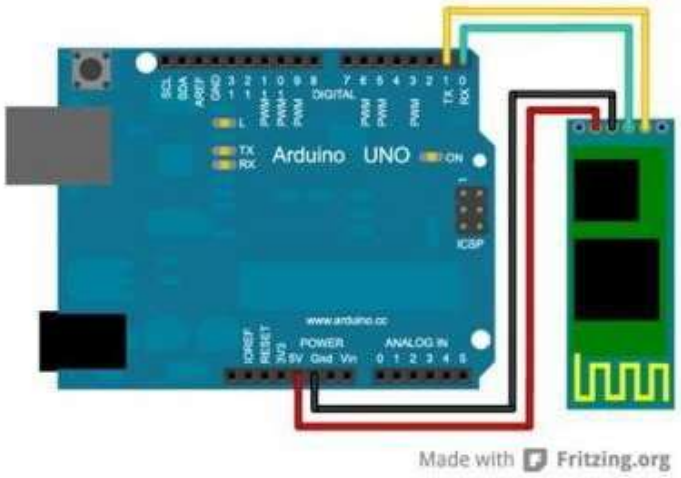
These two pins acts as an UART interface for communication

STATE:

It acts as a status indicator. When the module is not connected to / paired with any other Bluetooth device, signal goes Low. At this low state, the led flashes continuously which denotes that the module is not paired with other device. When this module is connected to/paired with any other Bluetooth device, the signal goes High. At this high state, the led blinks with a constant delay say for example 2s delay which indicates that the module is paired.

BUTTON SWITCH:

This is used to switch the module into AT command mode. To enable AT command mode, press the button switch for a second. With the help of AT commands, the user can change the parameters of this module but only when the module is not paired with any other BT device. If the module is connected to any other Bluetooth device, it starts to communicate with that device and fails to work in AT command mode.



2.5 MIT App Inventor

MIT App Inventor is a web application integrated development environment originally provided by Google, and now maintained by the Massachusetts Institute of Technology (MIT). It allows newcomers to computer programming to create application software(apps) for two operating systems (OS): Android (operating system), and iOS, which, as of 8 July 2019, is in final beta testing. It is free and open-source software released under Multi-licensing/dual licensing.

It uses a graphical user interface (GUI) very similar to the programming languages Scratch (programming language) and the StarLogo, which allows users to drag and drop visual objects to create an application that can run on mobile devices. In creating App Inventor, Google drew upon significant prior research in educational computing, and work done within Google on online development environments.

Open Blocks is distributed by MIT's *Scheller Teacher Education Program* (STEP) and is derived from master's thesis research by Ricarose Roque. Professor Eric Klopfer and Daniel Wendel of the Scheller Program supported the distribution of Open Blocks under an MIT License. Open Blocks visual programming is closely related to StarLogo TNG, a project of STEP, and Scratch, a project of the MIT Media Lab's Lifelong Kindergarten Group led by Mitchel Resnick. App Inventor 2 replaced Open Blocks with Blockly, a blocks editor that runs within a web browser.

App Inventor and the projects on which it is based are informed by constructionist learning theories, which emphasize that programming can be a vehicle for engaging powerful ideas through active learning. As such, it is part of an ongoing movement in computers and education that began with the work of Seymour Papert and the MIT Logo Group in the 1960s, and has also manifested itself with Mitchel Resnick's work on Lego Mindstorms and StarLogo.

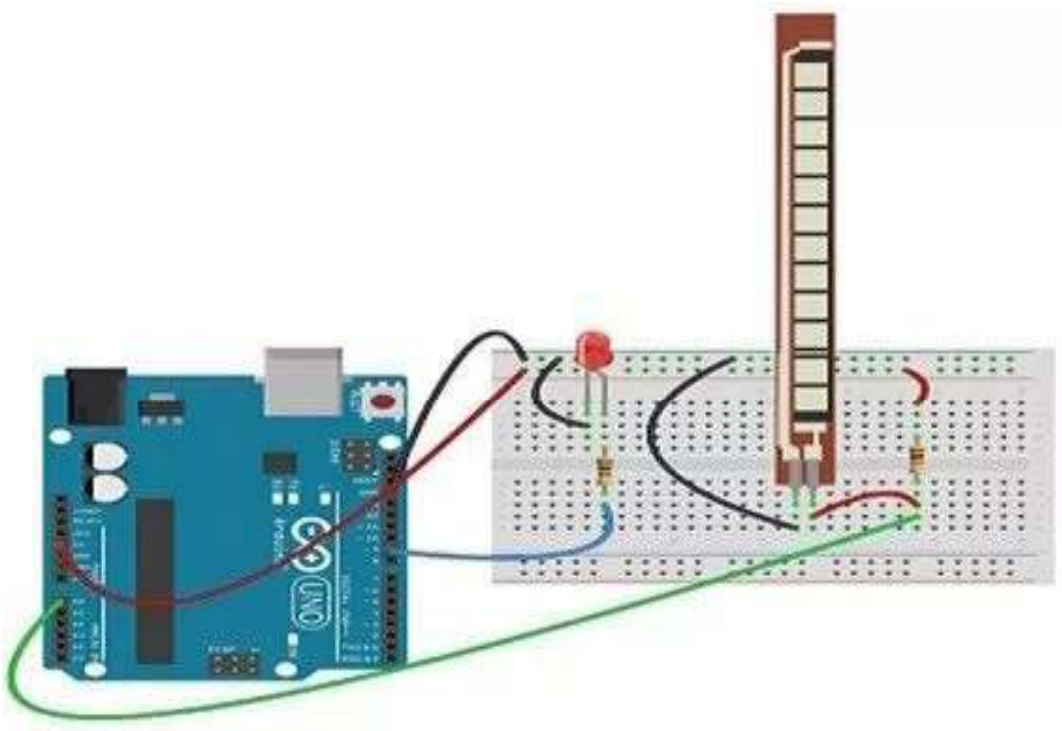
App Inventor also supports the use of cloud data via an experimental Firebase Realtime component.

Chapter 3

Implementation

3.1 Interfacing Flex Sensors with Arduino

For measuring bend, bending angle of any instrument, flex sensor is used with type of controller such as microcontroller or Arduino etc. For interfacing with Arduino board this sensor is powered up with Arduino board. When it is connected with Arduino board then its pin one is connected to +5V Arduino board pin through 68K ohm resistor and pin two is directly connected to Arduino ground pin. Similarly, a wire is connected between the centre point of 68K ohm resistor and flex sensor to Arduino board A0 pin. The remaining 4 sensors are connected in a similar manner using different values of resistors (50k, 50k, 34k and 34k respectively) for different flex sensors. The Arduino board pins A1, A2, A3 and A4 are used for these 4 sensors.

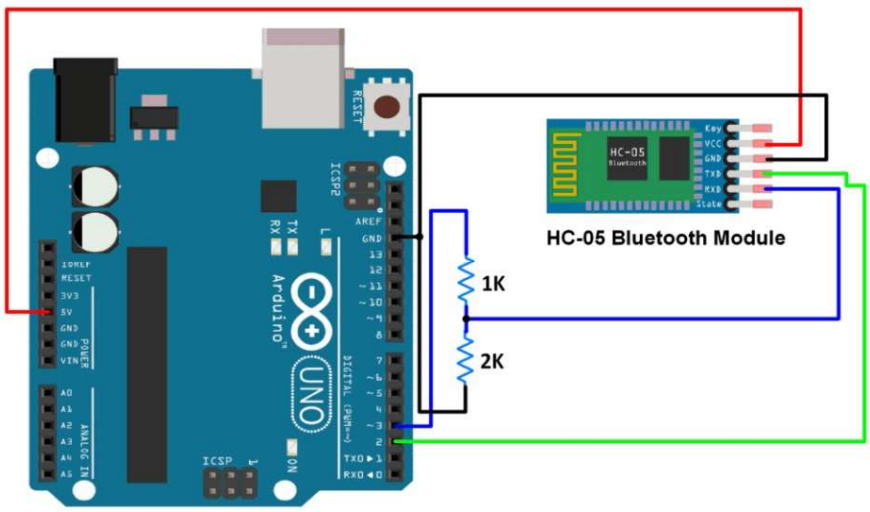


3.2 Thresholding of Flex Sensors

The resistance of flex sensors increases on bending but each sensor has its own range of values of resistance. The typical range of flex sensor is 30k-80k ohms but some sensors may have a smaller range of 30k-50k ohms or may have a larger range of 30k-110k ohms. Therefore, each flex sensor needs to be given a threshold value individually. The analog pin readings varies from 0 to 1000 depending upon the voltage at the center point. The range of resistance can be measured using a multi-meter and the resistor to be used in series can be decided accordingly. The resistor is chosen such that its resistance lies close to the mean value of the bent and unbent resistance of the flex sensor to keep the threshold value near 500 for all sensors.

3.3 Interfacing Bluetooth Module with Arduino

The particular module that I have can be powered from 3.6 to 6 volts, because it comes on breakout board which contains a voltage regulator. However, the logic voltage level of the data pins is 3.3V. So, the line between the Arduino TX (Transmit Pin, which has 5V output) and the Bluetooth module RX (Receive Pin, which supports only 3.3V) needs to be connected through a voltage divider in order not to burn the module. On the other hand, the line between the Bluetooth module TX pin and the Arduino RX pin can be connected directly because the 3.3V signal from the Bluetooth module is enough to be accepted as a high logic at the Arduino microcontroller.



3.4 Arduino Code

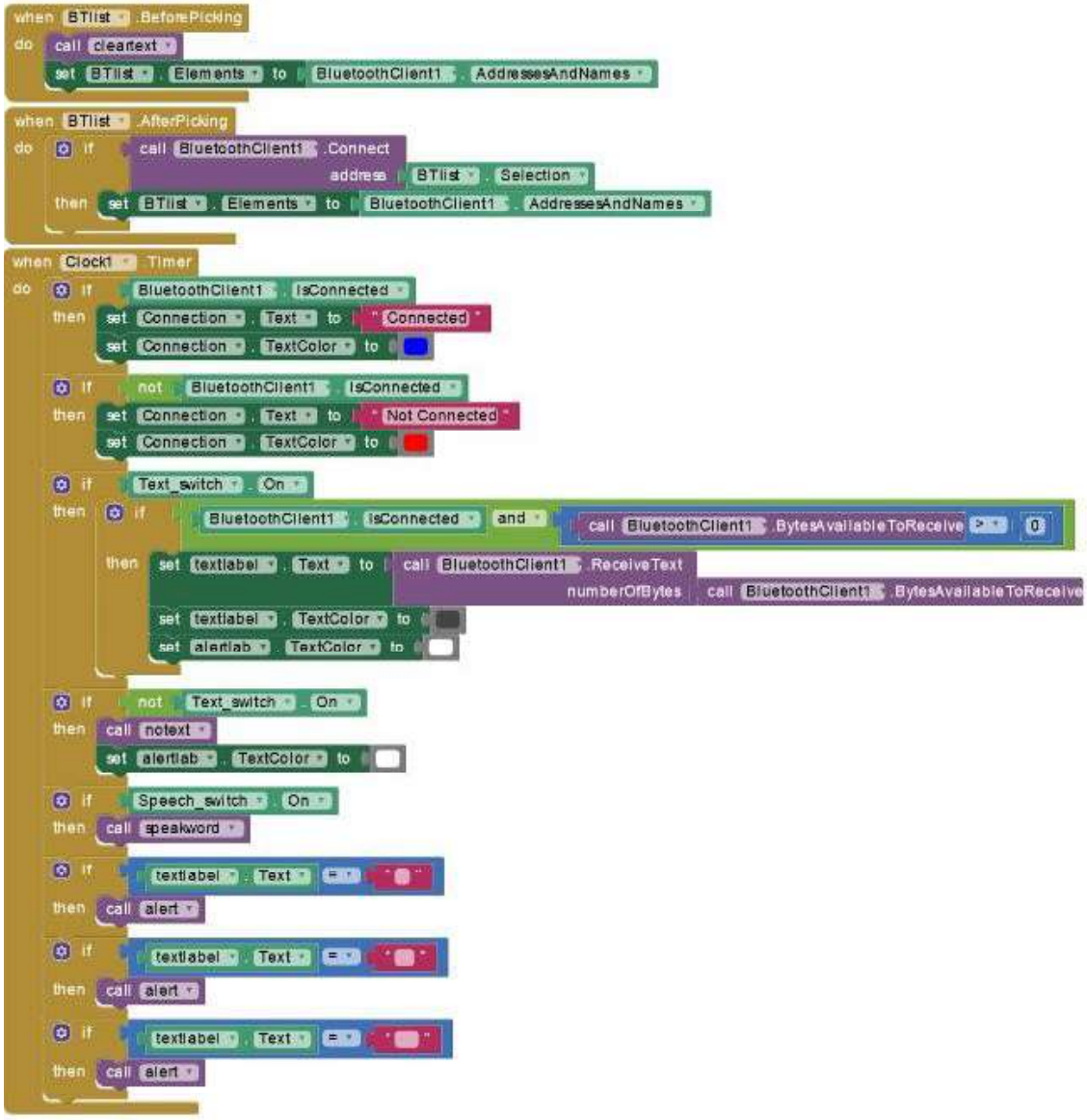
```
#include<SoftwareSerial.h>
const int analogInPin1 = A0;
const int analogInPin2 = A1;
const int analogInPin3 = A2;
const int analogInPin4 = A3;
const int analogInPin5 = A4;
SoftwareSerial bt(2,3);
int a=0,b=0,c=0,d=0,e=0,f=0;
int option1=0,option2=0;
int
sensorValue1=0,sensorValue2=0,sensorValue3=0,sensorValue4=0,sensorValue5=0;
int t1=520,t2=520,t3=520,t4=480,t5=520;
void setup()
{
    Serial.begin(9600);
    bt.begin(9600);
    bt.write("Choose an Option \n 1 Letter \n 2 Word");
    while(f==0)
    {
        sensorValue1  =  analogRead(analogInPin1);
        sensorValue2  =  analogRead(analogInPin2);
        sensorValue3  =  analogRead(analogInPin3);
        sensorValue4  =  analogRead(analogInPin4);
        sensorValue5  =  analogRead(analogInPin5);
        Serial.print(sensorValue1);
        Serial.print("    ");
        Serial.print(sensorValue2);
        Serial.print("    ");
        Serial.print(sensorValue3);
        Serial.print("    ");
        Serial.print(sensorValue4);
        Serial.print("    ");
        Serial.println(sensorValue5);
        if(sensorValue1>t1)
            a=1;
            else
            a=0;
        if(sensorValue2>t2)
            b=1;
            else
            b=0;
        if(sensorValue3>t3)
            c=1;
            else
            c=0;
        if(sensorValue4>t4)
            d=1;
            else
            d=0;
        if(sensorValue5>t5)
            e=1;
            else
            e=0;
        if(a==1&&b==0&&c==1&&d==1&&e==1)
        {
            option1=1;
            f=1;
            bt.write("Letter Selected \n ");
```

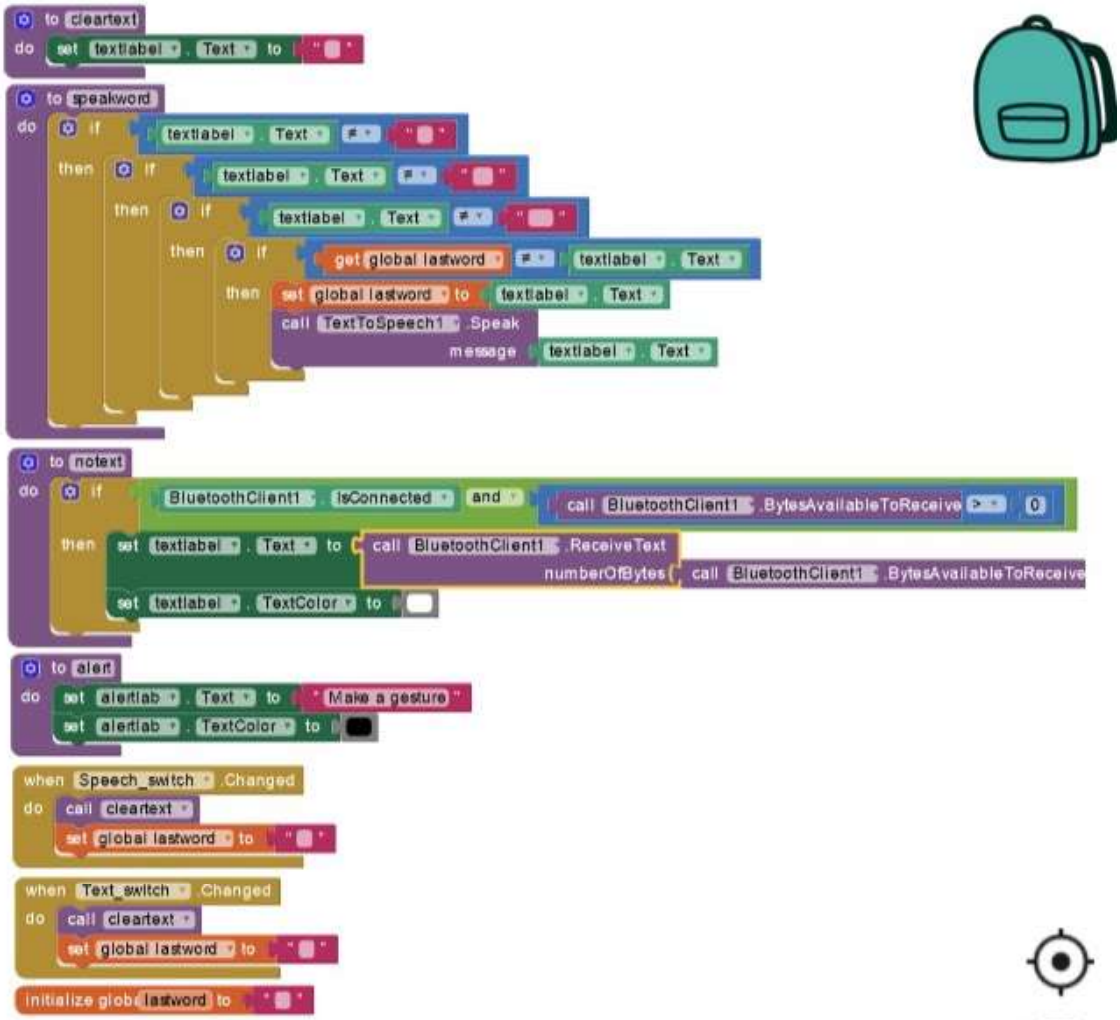
```
        }
        else if(a==1&&b==0&&c==0&&d==1&&e==1)
        {
            option2=1;
            f=1;
            bt.write("Word Selected \n");
        }
    }
    delay(2000);
    bt.write("Make a Gesture ");
    delay(2000);
}
void loop()
{
        sensorValue1  =  analogRead(analogInPin1);
        sensorValue2  =  analogRead(analogInPin2);
        sensorValue3  =  analogRead(analogInPin3);
        sensorValue4  =  analogRead(analogInPin4);
        sensorValue5  =  analogRead(analogInPin5);
        Serial.print(sensorValue1);
        Serial.print("    ");
        Serial.print(sensorValue2);
        Serial.print("    ");
        Serial.print(sensorValue3);
        Serial.print("    ");
        Serial.print(sensorValue4);
        Serial.print("    ");
        Serial.println(sensorValue5);
        if(sensorValue1>t1)
            a=1;
            else
            a=0;
        if(sensorValue2>t2)
            b=1;
            else
            b=0;
        if(sensorValue3>t3)
            c=1;
            else
            c=0;
        if(sensorValue4>t4)
            d=1;
            else
            d=0;
        if(sensorValue5>t5)
            e=1;
            else
            e=0;
    if(option1==1)
    {
        if(a==0&&b==0&&c==0&&d==0&&e==0)
            bt.write(" ");
        else if(a==0&&b==1&&c==1&&d==1&&e==1)
            bt.write("A");
        else if(a==1&&b==0&&c==1&&d==1&&e==1)
            bt.write("B");
        else if(a==1&&b==1&&c==0&&d==1&&e==1)
            bt.write("C");
        else if(a==1&&b==1&&c==1&&d==0&&e==1)
            bt.write("D");
```

```
        else if(a==1&&b==1&&c==1&&d==1&&e==0)
            bt.write("E");
        else if(a==0&&b==0&&c==1&&d==1&&e==1)
            bt.write("F");
        else if(a==0&&b==1&&c==0&&d==1&&e==1)
            bt.write("G");
        else if(a==0&&b==1&&c==1&&d==0&&e==1)
            bt.write("H");
        else if(a==0&&b==1&&c==1&&d==1&&e==0)
            bt.write("I");
        else if(a==1&&b==0&&c==0&&d==1&&e==1)
            bt.write("J");
        else if(a==1&&b==0&&c==1&&d==0&&e==1)
            bt.write("K");
        else if(a==1&&b==0&&c==1&&d==1&&e==0)
            bt.write("L");
        else if(a==1&&b==1&&c==0&&d==0&&e==1)
            bt.write("M");
        else if(a==1&&b==1&&c==0&&d==1&&e==0)
            bt.write("N");
        else if(a==1&&b==1&&c==1&&d==0&&e==0)
            bt.write("O");
        else if(a==0&&b==0&&c==0&&d==1&&e==1)
            bt.write("P");
        else if(a==0&&b==0&&c==1&&d==0&&e==1)
            bt.write("Q");
        else if(a==0&&b==0&&c==1&&d==1&&e==0)
            bt.write("R");
        else if(a==0&&b==1&&c==0&&d==0&&e==1)
            bt.write("S");
        else if(a==0&&b==1&&c==0&&d==1&&e==0)
            bt.write("T");
        else if(a==0&&b==1&&c==1&&d==0&&e==0)
            bt.write("U");
        else if(a==1&&b==0&&c==0&&d==0&&e==1)
            bt.write("V");
        else if(a==1&&b==0&&c==0&&d==1&&e==0)
            bt.write("W");
        else if(a==1&&b==0&&c==1&&d==0&&e==0)
            bt.write("X");
        else if(a==1&&b==1&&c==0&&d==0&&e==0)
            bt.write("Y");
        else if(a==0&&b==0&&c==0&&d==0&&e==1)
            bt.write("Z");
        else
            bt.write("Invalid Sign");
    }
    else
    {
        if(a==0&&b==0&&c==0&&d==0&&e==0)
            bt.write(" ");
        else if(a==0&&b==1&&c==1&&d==1&&e==1)
            bt.write("Water");
        else if(a==1&&b==0&&c==1&&d==1&&e==1)
            bt.write("Food");
        else if(a==1&&b==1&&c==0&&d==1&&e==1)
            bt.write("Milk");
        else if(a==1&&b==1&&c==1&&d==0&&e==1)
            bt.write("Pen");
        else if(a==1&&b==1&&c==1&&d==1&&e==0)
```

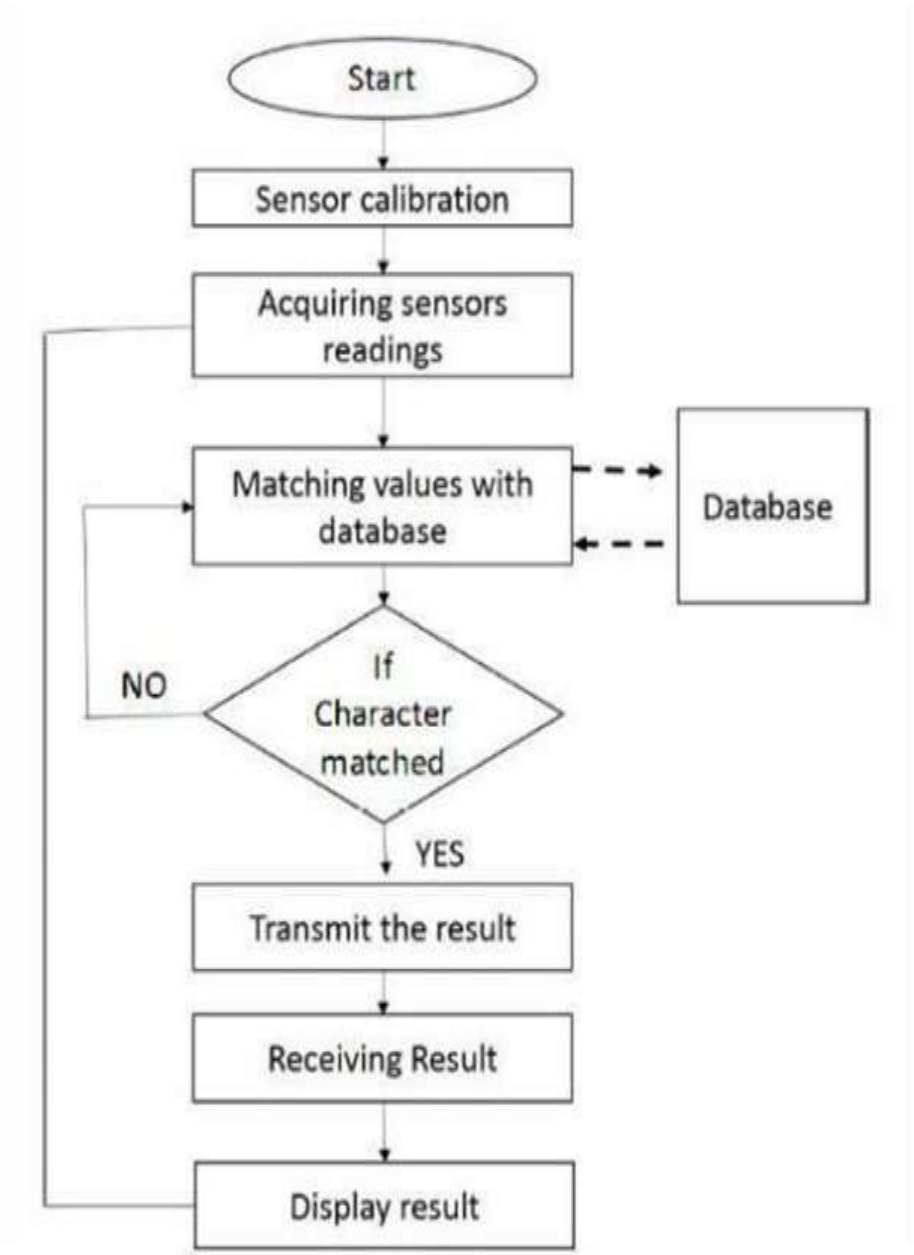
```
            bt.write("Book");
        else if(a==0&&b==0&&c==1&&d==1&&e==1)
            bt.write("Eat");
        else if(a==0&&b==1&&c==0&&d==1&&e==1)
            bt.write("Drink");
        else if(a==0&&b==1&&c==1&&d==0&&e==1)
            bt.write("Yes");
        else if(a==0&&b==1&&c==1&&d==1&&e==0)
            bt.write("No");
        else if(a==1&&b==0&&c==0&&d==1&&e==1)
            bt.write("OK");
        else if(a==1&&b==0&&c==1&&d==0&&e==1)
            bt.write("Come");
        else if(a==1&&b==0&&c==1&&d==1&&e==0)
            bt.write("Get");
        else if(a==1&&b==1&&c==0&&d==0&&e==1)
            bt.write("Give");
        else if(a==1&&b==1&&c==0&&d==1&&e==0)
            bt.write("Take");
        else if(a==1&&b==1&&c==1&&d==0&&e==0)
            bt.write("Help");
        else if(a==0&&b==0&&c==0&&d==1&&e==1)
            bt.write("Hello");
        else if(a==0&&b==0&&c==1&&d==0&&e==1)
            bt.write("Toilet");
        else if(a==0&&b==0&&c==1&&d==1&&e==0)
            bt.write("Work");
        else if(a==0&&b==1&&c==0&&d==0&&e==1)
            bt.write("Go");
        else if(a==0&&b==1&&c==0&&d==1&&e==0)
            bt.write("Thank You");
        else if(a==0&&b==1&&c==1&&d==0&&e==0)
            bt.write("Sorry");
        else if(a==1&&b==0&&c==0&&d==0&&e==1)
            bt.write("Fine");
        else if(a==1&&b==0&&c==0&&d==1&&e==0)
            bt.write("Sick");
        else if(a==1&&b==0&&c==1&&d==0&&e==0)
            bt.write("Rest");
        else if(a==1&&b==1&&c==0&&d==0&&e==0)
            bt.write("Sleep");
        else if(a==0&&b==0&&c==0&&d==0&&e==1)
            bt.write("Time");
        else if(a==0&&b==0&&c==0&&d==1&&e==0)
            bt.write("Busy");
        else if(a==0&&b==0&&c==1&&d==0&&e==0)
            bt.write("Stop");
        else if(a==0&&b==1&&c==0&&d==0&&e==0)
            bt.write("Home");
        else if(a==1&&b==0&&c==0&&d==0&&e==0)
            bt.write("Play");
        else if(a==1&&b==1&&c==1&&d==1&&e==1)
            bt.write("Clean");
    }
    delay(3500);
}
```

3.5 MIT App Inventor Blocks




to cleartext
do set textlabel . Text to
to speakword
do if textlabel . Text ≠
then if textlabel . Text ≠
then if textlabel . Text ≠
then if get global lastword ≠ textlabel . Text
then set global lastword to textlabel . Text
call TextToSpeech1 .Speak
message textlabel . Text
to notext
do if BluetoothClient1 . isConnected and call BluetoothClient1 .BytesAvailableToReceive > 0
then set textlabel . Text to call BluetoothClient1 .ReceiveText
numberOfBytes call BluetoothClient1 .BytesAvailableToReceive
set textlabel . TextColor to
to alert
do set alertlab . Text to " Make a gesture "
set alertlab . TextColor to
when Speech_switch .Changed
do call cleartext
set global lastword to
when Text_switch .Changed
do call cleartext
set global lastword to
initialize glob lastword to

3.6 Flow Chart



Chapter 4

Result and Discussion

4.1 Result

We were able to identify the hand gestures for the following letters and words successfully:

LETTERS	WORDS
A	WATER
B	FOOD
C	MILK
D	PEN
E	BOOK
F	EAT
G	DRINK
H	YES
I	NO
J	OK
K	COME
L	GET
M	GIVE
N	TAKE
O	HELP
P	HELLO
Q	TOILET
R	WORK
S	GO
T	THANK YOU
U	SORRY

V	FINE
W	SICK
X	REST
Y	SLEEP
Z	TIME
	BUSY
	STOP
	HOME
	PLAY
	CLEAN

4.2 Discussion

Our final implementation consists of two modes 1. Letters and 2. Words which can be selected with the help of hand gestures. The device can identify hand gestures for all 26 English alphabets and 31 frequently used English words and sends them to dedicated Android application through Bluetooth module.

Chapter 6

Summary and Conclusion

- Firstly, thresholding of the flex sensors was done.
- The five flex sensors were mounted on a glove.
- The hand gestures were detected and the dataset was completed on the basis of the results obtained.
- The circuit design and hardware implementation were completed by interfacing Bluetooth module and flex sensors with Arduino.
- The code was written to implement dual operation and uploaded to the Arduino.
- The software implementation was done using MIT App Inventor.
- The Bluetooth module serves the purpose of transferring data between Hardware part and the Software counterpart.

References

- www.wikipedia.org (ASL, Arduino, Flex Sensors)
- www.arduino.cc (Interfacing of Bluetooth Module)
- www.electronicshub.org (Interfacing of Flex Sensors)
- www.google.com (Images)
- http://ai2.appinventor.mit.edu

Acknowledgments

i. Project report for "Gesture detection-based communication device for disabled people" submitted in partial fulfilment of the requirement for the degree of Bachelor of Technology.

ii. Thesis Guide was referred to for preparing this thesis. Specifications regarding thesis format have been closely followed.

iii. The contents of the thesis have been organized based on the guidelines.

iv. The thesis has been prepared without resorting to plagiarism.

v. All sources used have been cited appropriately.

vi. The thesis has not been submitted elsewhere for a degree.