Human Computer Interface Glove for Sign Language Translation

A
Project Report
Submitted
By

AkankshaManocha
Arpita Gupta
Parnika Gupta

For The Degree Of
B.Tech[EC] 6th Semester

Mentor

Mrs. Lajwanti Singh (Asst. Professor)
Mr. Akshay Sharma (Technical Asst.)

Students

AkankshaManocha(BTBTE16019)
ArpitaGupta(BTBTE16068)
ParnikaGupta(BTBTE16017)

Department of Electronics

BanasthaliVidyapith, Rajasthan 304022

www.banasthali.org

Candidate's Declaration

I hereby declare that the work, which is being presented in the Project, entitled**''Human Computer Interface Glove for Sign Languaue Translation"**in partial fulfillment for the award of Degree of "*Bachelor* of Technology" in Deptt. of Electronics with Specialization in Electronics and Communication**and submitted to the Department of Electronics, BanasthaliVidyapith, Rajasthan** is a record of my own investigations carried under the Guidance of Mrs. Lajwanti Singh and Mr. Akshay Sharma, Department of Electronics, Banasthali Vidyapith**.**
I have not submitted the matter presented in this report anywhere for the award of any other Degree.

Akanksha Manocha
Arpita Gupta
Parnika Gupta

Electronics and Communication,
Student ID:
Akanksha Manocha (BTBTE16019)
Arpita Gupta (BTBTE16068)
Parnika Gupta (BTBTE16017)

Banasthali Vidyapith, Rajasthan

CONTENT Pg. No.

Abstract

A human computer interface glove was developed with the aim of translating sign language to text & speech. The glove utilizes five flex sensors and an inertial measurement unit to accurately capture hand gestures. All components were placed on the backside of the glove providing the user with full range of motion, and not restricting the user from performing other tasks while wearing the glove.

Introduction

Integrating people from deaf & mute communities into modern society, especially the modern workplace, is currently a major challenge. There is a major language barrier between users of Sign-language (signers or signing people) and those who do not understand it (non-signing people). Currently, the solution to this is to hire a Sign-language interpreter who will be present by the signer's side at all times. However, the employer often has to pay out of their pocket to hire this interpreter which results in it being more expensive than hiring another employee who does not have any communication barriers. Individuals who are deaf or mute are not only segregated in the modern workplace but also in everyday life causing them to live in their own separate communities. Worldwide there are 120 million deaf people who use Sign-language as their primary and most important means of communication. There is no single form of Sign-language as it varies from region to region; American Sign Language (ASL), British Sign Language (BSL), Indian Sign Language(ISL) are just a few of the different forms. ASL is often referred to as the fourth most common language in the United States.

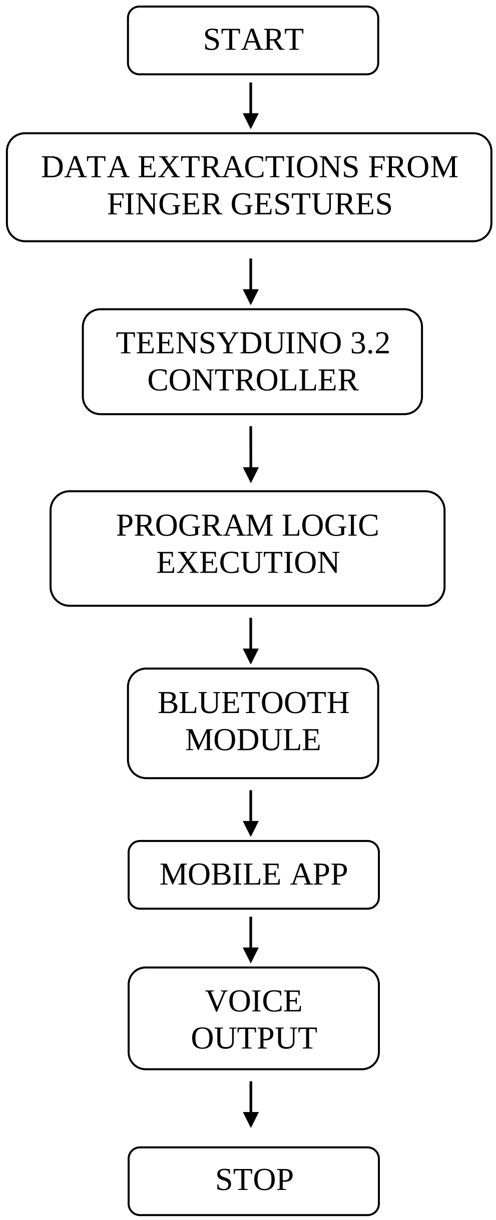
There has been a rapid progression in medicine recently, leading to solutions for some deaf & mute individuals. For instance, there have been improvements in hearing aids and cochlear implants for the deaf and artificial voice-boxes for the mute with vocal cord damage. However, these solutions do not come without disadvantages and costs. Cochlear implants have even caused a huge controversy in the deaf community and many refuse to even consider such solutions. Therefore, we believe society still requires an effective solution to remove the communication barrier between deaf & mute individuals and non-signing people.

Our proposed solution and goal is to design a Human Computer Interface (HCI) device that can translate sign language, specifically ASL, to text and speech providing any deaf & mute individuals with the ability to effortlessly communicate with anyone. Sign language involves the use of gestures, mainly specific hand shapes and movements, instead of sound to convey words and sentences. The idea is to design a device placed on a hand with sensors capable of capturing hand gestures and then transmitting the information to a processing unit which performs the sign language translation. The final product will also be able to efficiently teach sign language to any user since they would be able to receive instant feedback. We hope to be able to improve the quality of life of deaf and mute individuals with this device. During the development, we will also have the opportunity to learn a new language. We believe there is great potential in a device with these capabilities and know that it has far more applications in creating seamless gesture-based human-computer interfaces.

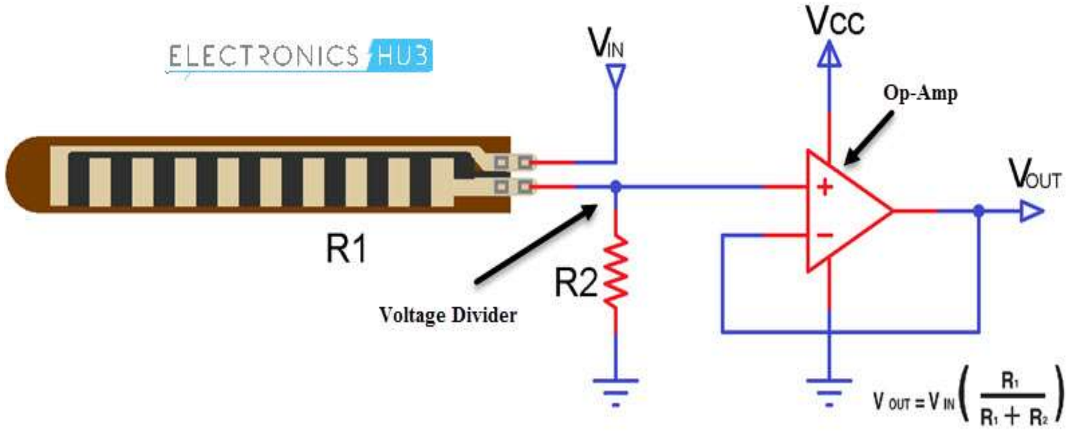
Current device solutions in the market are optic based such as LeapMotion. The disadvantage of using optical sensors is the light requirements necessary for function, and the limited space of use. Errors can arise from disturbances in the environment that obstruct the sensor's operating space. They are also not portable which is a necessity for sign language communication.

Working of project

- Block diagram



- Circuit Diagram



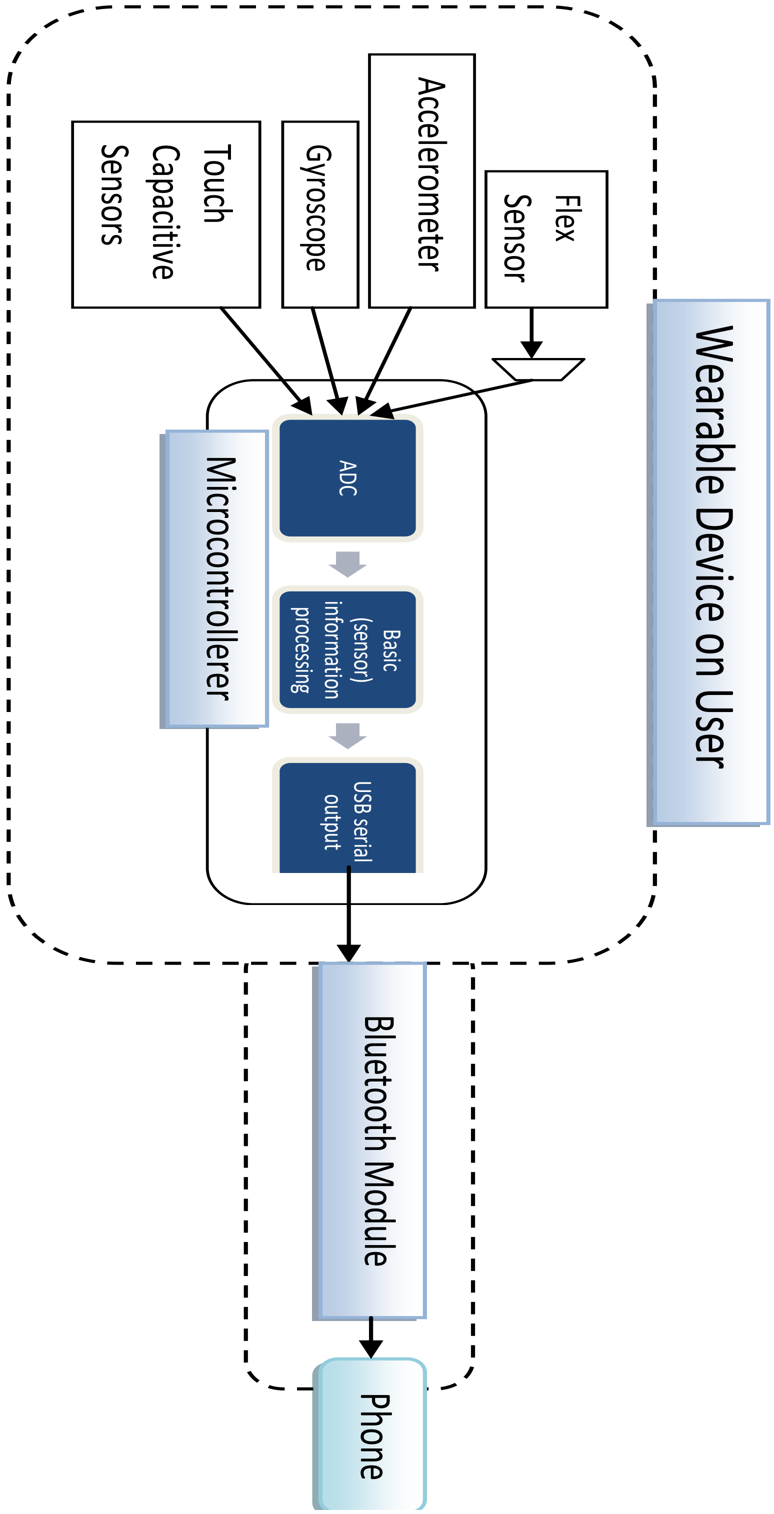
Flex sensors circuit diagram

- Explainations

We have placed 5 flex sensors, one on each finger to capture the movement of fingers and also an adafruit bno055 sensor with the controller to capture rotation of the hand. Therefore our glove device would be able to capture all the possible gestures that a hand can make. Then we have interfaced the sensors with Arduino software and took the output on serial monitor.

Instead of the sensor's readings we have taken the output of what the mute person wants to say with the specific gesture. Finally we have used an in-build serial monitor app which takes the input from the Arduino software via Bluetooth module and speaks it up.

- Methedology



Costing of Project

S. No.	Components	Quantity	Specifications and uses	Approximate Cost(in Rs)
1	HC -06 Bluetooth Module	1	For connecting teensyduino with the circuit	300
2	Flex Sensors	5	Strip sensors of recording the bending of fingers	2,160
4	Teensy 3.2 microcontroller	1	Microcontroller compatible with arduino code and has GPIO pins.	1,500
5	Adafruit BNO055 9-DOF IMU	1	Contains accerlerometer and gyroscope data.	275
6	Wires	20	For connections between pins.	100
7	Skin tight glove	1	Base for the project.	500
TOTAL COST				4,835

Component Description

1. Teensyduino

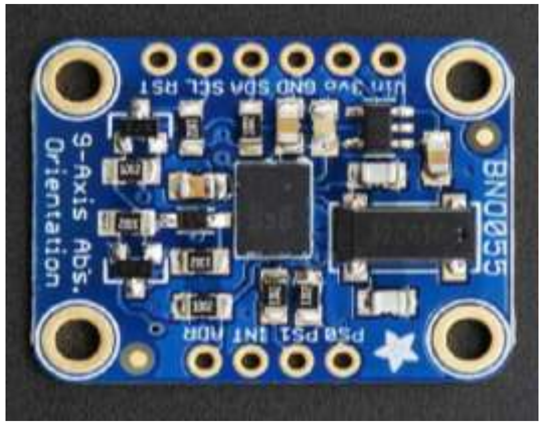
The Teensy 3.2 microcontroller has the ability to use Arduino code and libraries. As a result, “Teensyduino” was used to implement the code on theTeensyto perform sensordatacollection and processing. Since this project does not involve electrophysiological signals, a sampling rate of 50Hz was selected for all sensor data (quaternions, linear acceleration, flex sensors). The sampling rate could be increased but this is a reasonable starting point for the sensor data that is required.

As mentioned in the hardware implementation, the flex sensors involve a simple voltage divider circuit. Each flex sensor was tested individually for linearity between finger bending and voltage output. The linearity was good enough to create a linear mapping from 0°-90° of finger flexion. Each flex sensor was individually mapped for best performance. Theoutputvoltagefromthe flex sensor circuit (represented as a decimal corresponding to the 12bit ADC output) was measured for 0°-90° bending on each joint. Teensyduino allows the use of “analogRead()” to read the voltage output from the flex sensor circuit which is performed every 20ms.

Another reason the Teensy 3.2 microcontroller was chosen was due to the built-in “Touch Pins”.

The Adafruit BNO055 class library was used to interact with the IMU as it involved a lot of tedious register reading and writing. TheIMUhasitsown processor to perform sensor fusion of the accelerometer, gyroscope, and magnetometer for better accuracy andlessdrift..The appendix shows more details about the various outputs of the IMU.

2. Adafruit BNO055



“ The BNO055 is a System in Package (SiP), integrating a triaxial 14-bit accelerometer, a triaxial 16-bit gyroscope with a range of ±2000 degrees per second, a triaxial geomagnetic sensor and a 32-bit cortex M0+ microcontroller running Bosch Sensortec sensor fusion software, in a single package. The corresponding chip-sets are integrated into one single 28-pin LGA 3.8mm x 5.2mm x 1.1 mm housing. For optimum system integration the BNO055 is equipped with digital bidirectional I2C and UART interfaces. The I2C interface can be programmed to run with the HID-I2C protocol turning the BNO055 into a plug-and-play sensor hub solution for devices running the Windows 8.0 or 8.1 operating system.
“

The IMU can output the following data:

1.Orientation (Euler Vector, 100Hz)
Three axis orientation data based on a 360° sphere
2.Orientation (Quaternion, 100Hz)
Four point quaternion output for more accurate data manipulation
3.Angular Velocity Vector (100Hz)
Three axis of 'rotation speed' in rad/s
4.Acceleration Vector (100Hz)
Three axis of acceleration (gravity + linear motion) in m/s^2
5.Magnetic Field Strength Vector (20Hz)
Three axis of magnetic field sensing in micro Tesla (uT)
6.Linear Acceleration Vector (100Hz)
Three axis of linear acceleration data (acceleration minus gravity) in m/s^2
7.Gravity Vector (100Hz)
Three axis of gravitational acceleration (minus any movement) in m/s^2
8.Temperature (1Hz)

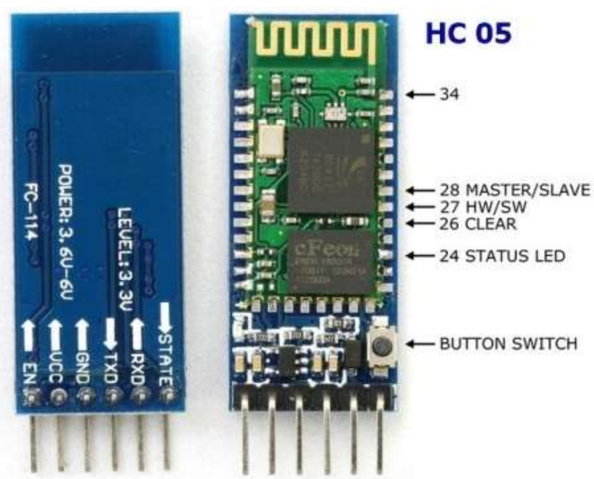
Ambient temperature in degrees Celsius

3.6.3 Fusion Output data rates

Table 3-14: Fusion output data rates

BNO055 Operating Mode	Data input rate			Algo calling rate	Data output rate			
	Accel	Mag	Gyro		Accel	Mag	Gyro	Fusion data
IMU	100Hz	NA	100Hz	100Hz	100Hz	NA	100Hz	100Hz
COMPASS	20Hz	20Hz	NA	20Hz	20Hz	20Hz	NA	20Hz
M4G	50Hz	50Hz	NA	50Hz	50Hz	50Hz	NA	50Hz
NDOF_FMC_OFF	100Hz	20Hz	100Hz	100Hz	100Hz	20Hz	100Hz	100Hz
NDOF	100Hz	20Hz	100Hz	100Hz	100Hz	20Hz	100Hz	100Hz

3. Bluetooth module



HC-05 module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup.

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). It has the footprint as small as 12.7mmx27mm. Hope it will simplify your overall design/development cycle.

Hardware Features

- Typical -80dBm sensitivity
- Up to +4dBm RF transmit power
- Low Power 1.8V Operation ,1.8 to 3.6V I/O

- PIO control
- UART interface with programmable baud rate
- With integrated antenna
- With edge connector

Software Features

- Default Baud rate: 38400, Data bits:8, Stop bit:1,Parity:No parity, Data control: has.

Supported baud rate: 9600,19200,38400,57600,115200,230400,460800.

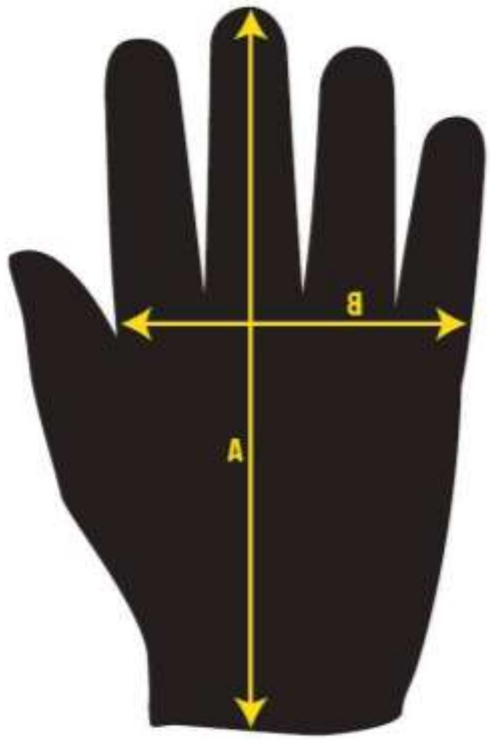
- Given a rising pulse in PIO0, device will be disconnected.
- Status instruction port PIO1: low-disconnected, high-connected;
- PIO10 and PIO11 can be connected to red and blue led separately. When master and slave

are paired, red and blue led blinks 1time/2s in interval, while disconnected only blue led blinks 2times/s.

- Auto-connect to the last device on power as default.
- Permit pairing device to connect as default.
- Auto-pairing PINCODE:"0000" as default
- Auto-reconnect in 30 min when disconnected as a result of beyond the range of connection.

4. Wire

5. Skin tight glove



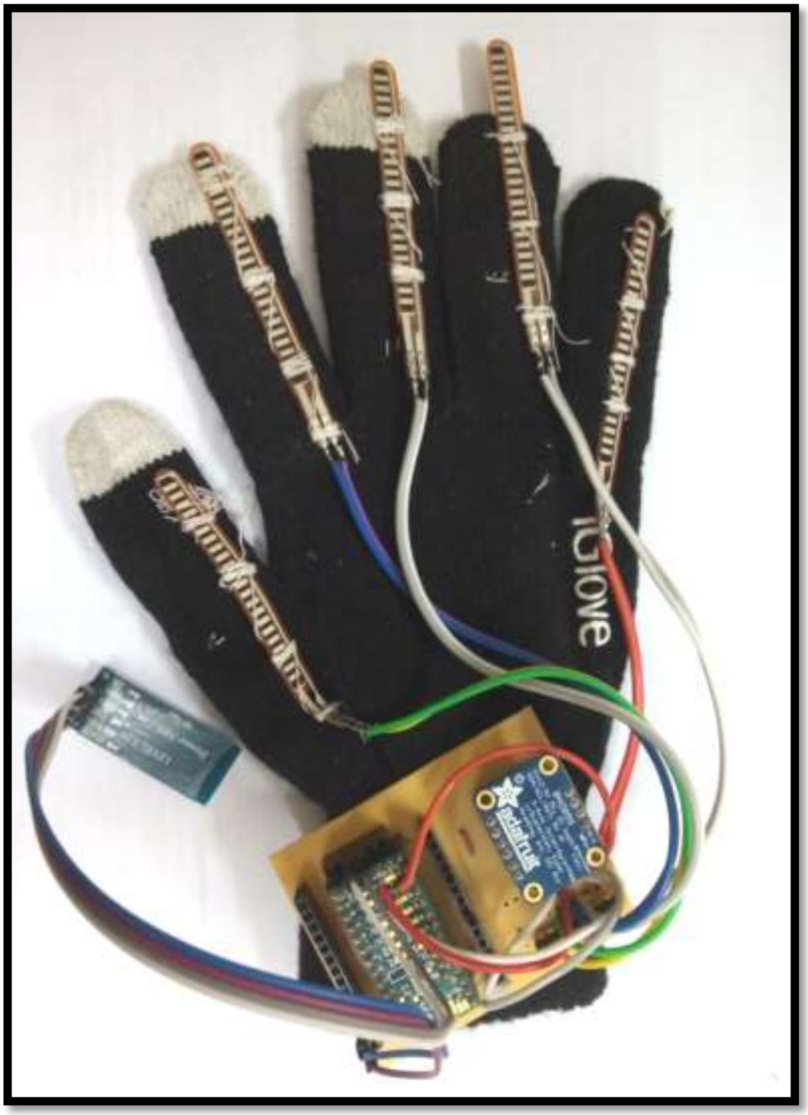
6. USB Connector

7. Flex Sensors

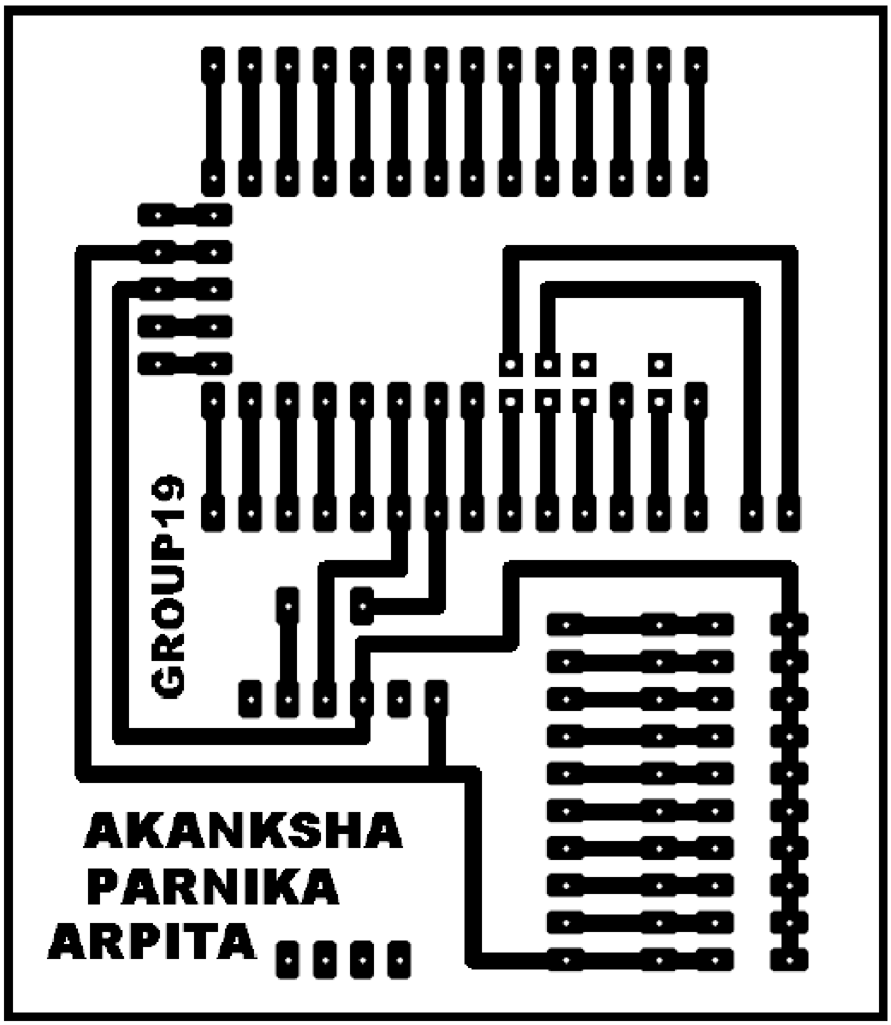
A **flex sensor** or **bend sensor** is a sensor that measures the amount of deflection or bending. Usually, the sensor is stuck to the surface, and resistance of sensor element is varied by bending the surface. Since the resistance is directly proportional to the amount of bend it is used as goniometer, and often called flexible potentiometer.

a) Operating voltage of FLEX SENSOR: 0-5V
b) Can operate on LOW voltages
c) Power rating : 0.5Watt (continuous), 1 Watt (peak)
d) Life: 1 million
e) Operating temperature: -45ºC to +80ºC
f) Flat Resistance: 25K Ω
g) Resistance Tolerance: ±30%
h) Bend Resistance Range: 45K to 125K Ohms(depending on bend)

- Glove



PCB Designing



Advantages

- The most prominent movement that can be performed by the four fingers (pinkie, ring, middle, and index) is bending towards the palm and then returning to the initial position. The thumb has unique advantages over the other fingers, thus enabling it to move freely in six degrees of freedom (DOF).
- The major advantage of a sensory-based approach is that gloves can acquire data directly (degree of bend, wrist orientation, hand motion, etc.) in terms of voltage values of the computing device, thus eliminating the need to process raw data into meaningful values. Furthermore, this approach is not subject to environmental influences.
- Doesn't cause vocal cord damage like in cochlear implants for the deaf and artificial voice-boxes for the mute.
- The disadvantage of using optical sensors is the light requirements necessary for function, and the limited space of use. Our glove uses no light.
- Our device is portable.

Disadvantages

- Cost should be reduced as much as possible so that it could be utilized for commonpeople.
- The main disadvantage of contact based devices is the health hazards, which are caused by its devices like mechanical sensor material which raises symptoms of allergy, magnetic devices which raises risk of cancer etc.
- The contact based can be uncomfortable for user since they require physical contact with the user, still having a verge over the accuracy of recognition and less complexity of implementation goes in favor of these devices.

Future Scope

In future, as an extension to our past work, there are many there are many future improvements that have already been considered. There are already plans to continue working on this project to see how far it can go. Due to less amount of time we only had time to focus on the functionality of the glove. One of the first steps would be to create a PCB to replace the perfboard and minimize the number of wires. Also, conductive thread could be used to wire the flex sensors.

Another major improvement would be to reduce the cost. The bulk of the cost was due to the flex sensors. There are a few guides for custom-made flex sensors, some of which are fabric-based. These would allow easy integration into the glove and would be aesthetically pleasing. The disadvantage is that it likely has reduced accuracy/resolution and linearity.And also we will replace the woolen glove with rubber made glove, so that we can have better flexibility, orientation and proper movement of hand with the PCB connection over it.

Conclusion

Our group had very high aspirations for this HCI glove. As a result, the goals for this project were clearly met. The device accurately classified ASL letters and numbers and performed text-to-speech on the spelled word. There are many plans to continue improving the device with the goal of developing most efficient and accurate device for the communication of mute and deaf people so that they do not have any communication barriers. Later we plan to test it with the actual users with a view to receive their feedbacks and iterate our produc

Appendix

CODE:

```
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BNO055.h>
#include <utility/imumaths.h>
#include <SoftwareSerial.h>
SoftwareSerialBTserial(0, 1); // RX | TX
constintflexPin = 34; //pin A0 to read analog input
int value; //save analog value
String sp = " ";
String fingerDegrees = "FingerDegrees: ";
String knuckleDegrees = "KnuckleDegrees: ";
#define BNO055_SAMPLERATE_DELAY_MS (100)
Adafruit_BNO055 bno = Adafruit_BNO055(55);
voiddisplaySensorDetails(void)
{
sensor_t sensor;
bno.getSensor(&sensor);
Serial.println("------------------------------------");
Serial.print  ("Sensor:       "); Serial.println(sensor.name);
Serial.print  ("Driver Ver:   "); Serial.println(sensor.version);
Serial.print  ("Unique ID:    "); Serial.println(sensor.sensor_id);
Serial.print  ("Max Value:    "); Serial.print(sensor.max_value); Serial.println(" xxx");
Serial.print  ("Min Value:    "); Serial.print(sensor.min_value); Serial.println(" xxx");
Serial.print  ("Resolution:   "); Serial.print(sensor.resolution); Serial.println(" xxx");
Serial.println("------------------------------------");
Serial.println("");
delay(500);
}
/*
   Display some basic info about the sensor status */
voiddisplaySensorStatus(void)
{
  /* Get the system status values (mostly for debugging purposes) */
  uint8_t system_status, self_test_results, system_error;
system_status = self_test_results = system_error = 0;
bno.getSystemStatus(&system_status, &self_test_results, &system_error);
  /* Display the results in the Serial Monitor */
Serial.println("");
```

```
Serial.print("System Status: 0x");
Serial.println(system_status, HEX);
Serial.print("Self Test:     0x");
Serial.println(self_test_results, HEX);
Serial.print("System Error:  0x");
Serial.println(system_error, HEX);
Serial.println("");
delay(500);
}
/*
    Display sensor calibration status
*/
/*********************************************************************
***/
voiddisplayCalStatus(void)
{
  /* Get the four calibration values (0..3) */
  /* Any sensor data reporting 0 should be ignored, */
  /* 3 means 'fully calibrated" */
uint8_t system, gyro, accel, mag;
system = gyro = accel = mag = 0;
bno.getCalibration(&system, &gyro, &accel, &mag);
  /* The data should be ignored until the system calibration is > 0 */
Serial.print("\t");
if (!system)
  {
Serial.print("! ");
  }
  /* Display the individual values */
Serial.print("Sys:");
Serial.print(system, DEC);
Serial.print(" G:");
Serial.print(gyro, DEC);
Serial.print(" A:");
Serial.print(accel, DEC);
Serial.print(" M:");
Serial.print(mag, DEC);
}


void setup()
{
```

```
Serial.begin(9600);
Serial.begin(115200);
Serial.println("Arduino is ready");
// HC-05 default serial speed for commincation mode is 9600
Serial.begin(9600);
Serial.println("Orientation Sensor Test"); Serial.println("");
  /* Initialise the sensor */
if(!bno.begin())
  {
    /* There was a problem detecting the BNO055 ... check your connections */
Serial.print("Ooops, no BNO055 detected ... Check your wiring or I2C ADDR!");
while(1);
  }
delay(1000);
  /* Display some basic information on this sensor */
displaySensorDetails();
  /* Optional: Display current status */
displaySensorStatus();
bno.setExtCrystalUse(true);
}
void loop()
{
intflexRaw[9];
intflexDegrees[9];
intflexDegCal[9][2];
intflexPinMap[9];
  /* Mapping to Analog pins on teensy
fromflexPinMap[0] to flexPinMap[8]
    Index, IndexKnuckle, Middle, MiddleKnuckle
    Ring, Ring Knuckle, Pinky, Thumb, Thumb Knuckle*/
flexPinMap[0] = 28; flexPinMap[1] = 27; flexPinMap[2] = 26; flexPinMap[3] = 31;
flexPinMap[4] = 30; flexPinMap[5] = 29; flexPinMap[6] = 35; flexPinMap[7] = 34;
flexPinMap[8] = 36;
// Calibrated mapping from analog values to degrees for each joint
flexDegCal[0][0] = 490; flexDegCal[0][1] = 662; // Middle Knuckle
flexDegCal[1][0] = 578; flexDegCal[1][1] = 720; // Pinky
flexDegCal[2][0] = 500; flexDegCal[2][1] = 697; // Ring Knuckle
flexDegCal[3][0] = 488; flexDegCal[3][1] = 600; // Thumb Knuckle
flexDegCal[4][0] = 590; flexDegCal[4][1] = 662; // Thumb
flexDegCal[5][0] = 542; flexDegCal[5][1] = 705; // Index Knuckle
flexDegCal[6][0] = 539; flexDegCal[6][1] = 680; // Middle
flexDegCal[7][0] = 503; flexDegCal[7][1] = 708; // Index
```

```
flexDegCal[8][0] = 503; flexDegCal[8][1] = 752; // Ring
for (inti = 0; i< 9; i++){
flexRaw[i] = analogRead(flexPinMap[i]);
flexDegrees[i] = map(flexRaw[i], flexDegCal[i][0], flexDegCal[i][1], 0, 90);
  }
 // print the finger and knuckle bending degrees to serial output
Serial.println(fingerDegrees + flexDegrees[7] + sp + flexDegrees[6] + sp +
flexDegrees[8] + sp + flexDegrees[1] + sp + flexDegrees[4]);
Serial.println(knuckleDegrees + flexDegrees[5] + sp + flexDegrees[0] + sp +
flexDegrees[2] + sp + flexDegrees[3]);
Serial.println("FlexSensorBending(raw)");
Serial.print("Index: ");
Serial.println(flexRaw[7],DEC);
//Serial.print(F(" "));
Serial.print("Middle: ");
Serial.println(flexRaw[6],DEC);
//Serial.print(F(" "));
Serial.print("Ring: ");
Serial.println(flexRaw[8],DEC);
//Serial.print(F(" "));
Serial.print("Pinky: ");
Serial.println(flexRaw[1],DEC);
//Serial.print(F(""));
Serial.print("Thumb: ");
Serial.println(flexRaw[4],DEC);
value = analogRead(flexPin);        //Read and save analog value from potentiometer
if(value>900)
    {
intval = value;
Serial.println("Drink water");
BTserial.write("Need water");
    //if(val<(val+30) || val>(val+30))
delay(2000);
  }
 // get incoming byte:
  // inByte = Serial.read();
///Serial.println(value);              //Print value
value = map(value, 700, 900, 0, 255);//Map value 0-1023 to 0-255 (PWM)
/* Get a new sensor event */
sensors_event_t event;
bno.getEvent(&event);
  /* Display the floating point data */
```

```
Serial.print("X: ");
Serial.print(event.orientation.x, 4);
Serial.print("\tY: ");
Serial.print(event.orientation.y, 4);
Serial.print("\tZ: ");
Serial.print(event.orientation.z, 4);
/* Optional: Display calibration status */
displayCalStatus();
  /* Optional: Display sensor status (debug only) */
  //displaySensorStatus();
/* New line for the next sample */
Serial.println("");
  /* Wait the specified delay before requesting nex data */
delay(BNO055_SAMPLERATE_DELAY_MS);
delay(1000);
}
```

References

[1]
Sheet, Data. "BNO055 Intelligent 9-axis Absolute Orientation Sensor." (n.d.): n. pag. Bos ch, Nov. 2014. Web. Dec. 2015.

[2]
ABDULLAH AL MAMUN, MD SARWAR JAHAN KHAN POLASH and FAKIR MASHUQUE ALAMGIR , International Journal of Computer Networks and Communications Security VOL. 5, NO. 2, FEBRUARY 2017, 38–48

[3]
Joshua Project.(2012).Deaf in Bangladesh[Online].Available URL:https://joshuaproject.net/people_groups/1 9007/BG

[4]
Cxnull.(2011,November 15).Flexion [Online]. Available URL:http://www.sensorwiki.org/doku.php/sens ors/flexion