DIY REPORT

COMMUNICATIVE HAND GLOVE

ABSTRACT AND MOTIVATION

- THE EMPLOYMENT RATE AMONG THE SPECIALLY ABLED PEOPLE IS STAGGERINGLY LOW.
- THIS IS MAINLY DUE TO THE NON-AVAILABILITY OF A PERSON WHO CAN INTERPRET AMERICAN SIGN LANGUAGE AT THE JOB SITE.
- DEAF AND DUMB PEOPLE ALSO DO NOT FIND IT EASY IN THEIR REGULAR LIVES AS NOT EVERYONE AROUND THEM KNOWS THE AMERICAN SIGN LANGUAGE.
- ONE SUCH EXAMPLE IS A CHILD WHO LIVES IN AN ORPHANAGE NEAR RACHANA'S PLACE. THIS GAVE US A PERSONAL MOTIVATION TO HELP HIM AND WORK ON THIS PROJECT.

PROOF OF CONCEPT

- WE CONNECT THE FLEX SENSORS AND ACCELEROMETER TO THE ANALOG PINS OF ARDUINO NANO R3.
- FLEX SENSORS WORK THROUGH A THIN SUBSTRATE THAT PRODUCES VARIOUS VALUES OF

RESISTANCE BASED ON THE ANGLE AT WHICH THEY ARE BENT.

- THE HC-05 BLUETOOTH MODEL IS CONNECTED TO THE TX-RX PINS OF THE ACCELEROMETER, WHILE THE MODEL ITSELF IS CONNECTED TO OUR PHONES VIA BLUETOOTH.
- WE USE AN APPLICATION CALLED “ARDUINO BLUETOOTH TEXT TO SPEECH” WHICH WILL INTERPRET THE SIGNS MADE THROUGH THE GLOVE AS DIFFERENT LETTERS AS PER THE CODE WE HAVE UPLOADED IN THE ARDUINO.

METHODOLOGY

- WE WISH TO HELP THE DEAF AND DUMB IN OVERCOMING THE PROBLEMS THEY FACE IN THEIR LIVES.
- THEY CAN USE THIS GLOVE, MAKE SYMBOLS USING THEIR HANDS AS PER THE AMERICAN SIGN LANGUAGE, AND ANY PERSON CAN INTERPRET WHAT THEY ARE TRYING TO CONVEY BY CONNECTING THEIR PHONES TO THE GLOVE VIA BLUETOOTH.
- WE USE THE AMERICAN SIGN LANGUAGE AS IT IS THE MOST USED SIGN LANGUAGE ALL OVER THE WORLD.
- HOWEVER, WE CAN MAKE SLIGHT CHANGES TO THE CODE AND IT CAN BE USED TO SPELL OUT SENTENCES AS WELL, WHICH IS QUITE USEFUL IN JOBS WHERE EMPLOYEES ARE EXPECTED TO USE REPETITIVE PHRASES.

IMPORTANT OBSERVATIONS

- WE CAN THE SEE THE VARIOUS LETTERS SPELLED OUT ON THE SCREENS OF OUR PHONES AS WE MAKE SYMBOLS WITH OUR FINGERS.
- WE NOTICED THAT THE SIDE AT WHICH THE FLEX SENSORS ARE GLUED TO THE GLOVES ALSO PLAYS AN IMPORTANT ROLE, ONE SIDE IS MORE FLEXIBLE THAN THE OTHER.
- SOLDERING IRON MUST BE USED VERY CAREFULLY, AS OVERUSING IT MIGHT DAMAGE THE HARDWARE.

CONCLUSIONS AND FUTURE SCOPE

- THUS WE HAVE MADE A GLOVE WHICH CAN VOCALISE GESTURES.
- THIS CAN BE USED TO GENERATE EMPLOYMENT IN VARIOUS SECTORS FOR THE SPECIALLY ABLED PEOPLE.
- WE CAN ALSO ATTACH A FITNESS SENSOR TO THE GLOVE, WHICH CAN GENERATE IMPORTANT PARAMETERS OF OUR BODY, FOR EXAMPLE PULSE RATE.

CREDITS AND RESPONSIBILITIES OF GROUP MEMBERS

- N RACHANA GUPTA(21CE10038) AND AKULA TEJASWINI(21EC30009) – PHYSICAL MODEL AND FINAL VIDEO
- M VAISHNAVI(21IE10022) – CODING, PRESENTATION AND VIDEO EDITING
- SEJAL SAXENA(21BT10028) – CODING, REPORT AND VIDEO EDITING

REFERENCES

- https://youtu.be/60ch5FFG5nI
- https://forum.arduino.cc/
- https://www.elprocus.com/
- https://www.tme.eu/en/news/library-articles/

APPENDIX 1 (code)

```
/*

NAME OF THE PROJECT : SIGN LANGUAGE TRANSLATOR


THIS PROJECT READS SENSOR VALUE AND TRY TO UNDERSTAND SYMBOLS GENERATED FROM AMERICAN SIGN LANGUAGE (ASL) CHART

AND DISPLAY ON A SMARTPHONE.

-------PIN CONFIGURATION----------

A0-A4 : FLEX SENSOR

D4&D5 : FOR BLUETOOTH RX AND TX

A5&A6 : XPIN AND YPIN FOR ACCELROMETER

*/


#include <SoftwareSerial.h>


SoftwareSerial mySerial(5,4);


char temp = '0';


//variable initializtion

int xpin = A5;

int xadc = 0;

int xmax = 0;

int xmin = 1023;


int ypin = A6;

int yadc = 0;

int ymax = 0;

int ymin = 1023;
```

```
int FLEX_PIN1 = A0;
int flexADC1 = 0;
int sensorMin1 = 1023;
int sensorMax1 = 0;

int FLEX_PIN2 = A1;
int flexADC2 = 0;
int sensorMin2 = 1023;
int sensorMax2 = 0;

int FLEX_PIN3 = A2;
int flexADC3 = 0;
int sensorMin3 = 1023;
int sensorMax3 = 0;

int FLEX_PIN4 = A3;
int flexADC4 = 0;
int sensorMin4 = 1023;
int sensorMax4 = 0;

int FLEX_PIN5 = A4;
int flexADC5 = 0;
int sensorMin5 = 1023;
int sensorMax5 = 0;


void setup()
{
mySerial.begin(9600);
while (!Serial)
{
; // wait for serial port to connect. Needed for native USB port only
}
// callibrating the sensors for adaptivity with different bends
while(millis()<15000)
{
```

```
if(digitalRead(7)==HIGH)
{
float flexADC1 = analogRead(FLEX_PIN1);
float flexADC2 = analogRead(FLEX_PIN2);
float flexADC3 = analogRead(FLEX_PIN3);
float flexADC4 = analogRead(FLEX_PIN4);
float flexADC5 = analogRead(FLEX_PIN5);

if(flexADC1<sensorMin1)
{
sensorMin1=flexADC1;
}
if(flexADC1>sensorMax1)
{
sensorMax1=flexADC1;
}

if(flexADC2<sensorMin2)
{
sensorMin2=flexADC2;
}
if(flexADC2>sensorMax2)
{
sensorMax2=flexADC2;
}

if(flexADC3<sensorMin3)
{
sensorMin3=flexADC3;
}
if(flexADC3>sensorMax3)
{
sensorMax4=flexADC4;
}

if(flexADC5<sensorMin5)
```

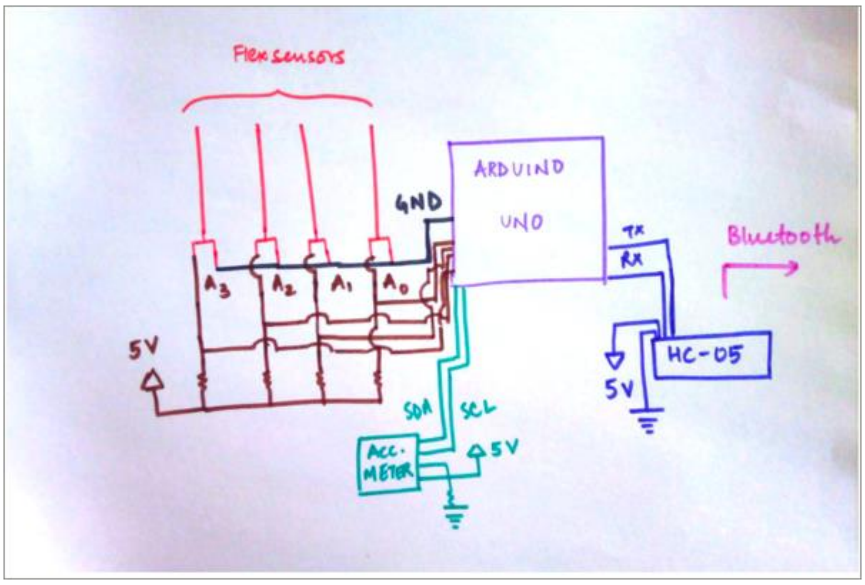
```
{
sensorMin5=flexADC5;
}
if(flexADC5>sensorMax5)
{
sensorMax5=flexADC5;
}

if(flexADC4<sensorMin4)
{
sensorMin4=flexADC4;
}
if(flexADC4>sensorMax4)
{
sensorMax4=flexADC4;
}
}
}
}


void printfun(char cp) //to avoid printing repeating symbols
{
if(cp!=temp)
{
mySerial.print(cp);
temp=cp;
}
}

void loop()
{
// reading sensor value
float flexADC1 = analogRead(FLEX_PIN1);
float flexADC2 = analogRead(FLEX_PIN2);
float flexADC3 = analogRead(FLEX_PIN3);
```

```
float flexADC4 = analogRead(FLEX_PIN4);

float flexADC5 = analogRead(FLEX_PIN5);


flexADC1 = constrain(flexADC1,sensorMin1, sensorMax1);

flexADC2 = constrain(flexADC2,sensorMin2, sensorMax2);

flexADC3 = constrain(flexADC3,sensorMin3, sensorMax3);

flexADC4 = constrain(flexADC4,sensorMin4, sensorMax4);

flexADC5 = constrain(flexADC5,sensorMin5, sensorMax5);



float angle1= map(flexADC1, sensorMin1, sensorMax1, 0, 90);

float angle2= map(flexADC2, sensorMin2, sensorMax2, 0, 90);

float angle3= map(flexADC3, sensorMin3, sensorMax3, 0, 90);

float angle4= map(flexADC4, sensorMin4, sensorMax4, 0, 90);

float angle5= map(flexADC5, sensorMin5, sensorMax5, 0, 90);


xadc = analogRead(xpin);

yadc = analogRead(ypin);


if(((angle1>=70)&&(angle1<=82))&&((angle2>=77)&&(angle2<=95))&&((angle3>=70)&&(angle3<=86))&&((angle4>=73)&&(angle4<=85))&&((angle5>=0)&&(angle5<=45)))

printfun('A');

if(((angle1>=0)&&(angle1<=10))&&((angle2>=0)&&(angle2<=10))&&((angle3>=0)&&(angle3<=12))&&((angle4>=0)&&(angle4<=10))&&((angle5>=65)&&(angle5<=80)))

printfun('B');

if(((angle1>=40)&&(angle1<=72))&&((angle2>=50)&&(angle2<=90))&&((angle3>=51)&&(angle3<=75))&&((angle4>=42)&&(angle4<=66))&&((angle5>=34)&&(angle5<=50)))

printfun('C');

if(((angle1>=50)&&(angle1<=72))&&((angle2>=45)&&(angle2<=90))&&((angle3>=35)&&(angle3<=75))&&((angle4>=0)&&(angle4<=10))&&((angle5>=45)&&(angle5<=80))&&!(((xadc>=412)&&(xadc<=418))&&((yadc>=340)&&(yadc<=360))))

printfun('D');

if(((angle1>=0)&&(angle1<=10))&&((angle2>=0)&&(angle2<=10))&&((angle3>=0)&&(angle3<=10))&&((angle4>=15)&&(angle4<=45))&&((angle5>=34)&&(angle5<=65)))

printfun('F');

if(((angle1>=75)&&(angle1<=90))&&((angle2>=75)&&(angle2<=90))&&((angle3>=65)&&(angle3<=90))&&((angle4>=0)&&(angle4<=15))&&((angle5>=0)&&(angle5<=30))&&(((xadc>=400)&&(xadc<=420))&&((yadc>=340)&&(yadc<=360))))

printfun('G');

if(((angle1>=70)&&(angle1<=85))&&((angle2>=75)&&(angle2<=90))&&((angle3>=0)&&(angle3<=10))&&((angle4>=0)&&(angle4<=10))&&((angle5>=50)&&(angle5<=65))&&!(((xadc>=410)&&(xadc<=420))&&((yadc>=368)&&(yadc<=380))))
```

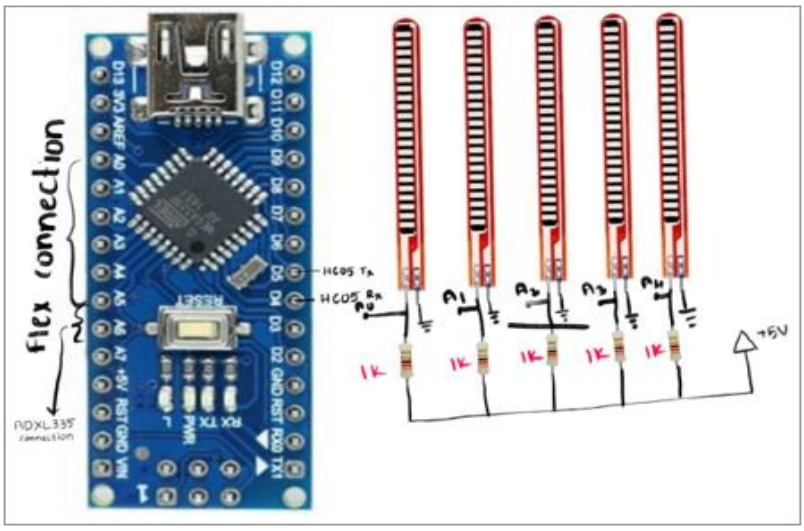
```
printfun('H');

if(((angle1>=0)&&(angle1<=10))&&((angle2>=50)&&(angle2<=70))&&((angle3>=50)&&(angle3<=70))&&((angle4>=50)&&(angle4<=70))&&((angle5>=50)&&(angle5<=85)&&((xadc>=410)&&(xadc<=420))&&((yadc>=330)&&(yadc<=370))))

printfun('I');

if(((angle1>=0)&&(angle1<=10))&&((angle2>=50)&&(angle2<=70))&&((angle3>=50)&&(angle3<=70))&&((angle4>=50)&&(angle4<=70))&&((angle5>=50)&&(angle5<=85))&&(!((xadc>=410)&&(xadc<=420))&&((yadc>=355)&&(yadc<=370))))

printfun('J');

if(((angle1>=60)&&(angle1<=75))&&((angle2>=60)&&(angle2<=85))&&((angle3>=0)&&(angle3<=10))&&((angle4>=0)&&(angle4<=15))&&((angle5>=30)&&(angle5<=55))&&(((xadc>=404)&&(xadc<=415))&&((yadc>=368)&&(yadc<=380))))

printfun('K');

if(((angle1>=75)&&(angle1<=90))&&((angle2>=75)&&(angle2<=90))&&((angle3>=70)&&(angle3<=90))&&((angle4>=0)&&(angle4<=15))&&((angle5>=0)&&(angle5<=30))&&(((xadc>=390)&&(xadc<=405))&&((yadc>=360)&&(yadc<=380)))&&!((xadc>=270)&&(xadc<=300))&&((yadc>=360)&&(yadc<=390)))

printfun('L');

if(((angle1>=40)&&(angle1<=61))&&((angle2>=72)&&(angle2<=84))&&((angle3>=45)&&(angle3<=65))&&((angle4>=62)&&(angle4<=75))&&((angle5>=65)&&(angle5<=86)))

printfun('O');

if(((angle1>=60)&&(angle1<=75))&&((angle2>=60)&&(angle2<=85))&&((angle3>=0)&&(angle3<=10))&&((angle4>=0)&&(angle4<=15))&&((angle5>=30)&&(angle5<=55))&&(((xadc>=270)&&(xadc<=290))&&((yadc>=360)&&(yadc<=380))))

printfun('P');

if(((angle1>=75)&&(angle1<=90))&&((angle2>=75)&&(angle2<=90))&&((angle3>=65)&&(angle3<=90))&&((angle4>=0)&&(angle4<=15))&&((angle5>=0)&&(angle5<=30))&&(((xadc>=270)&&(xadc<=300))&&((yadc>=360)&&(yadc<=390))))

printfun('Q');

if(((angle1>=40)&&(angle1<=72))&&((angle2>=45)&&(angle2<=90))&&((angle3>=20)&&(angle3<=45))&&((angle4>=0)&&(angle4<=10))&&((angle5>=45)&&(angle5<=80))&&(((xadc>=412)&&(xadc<=418))&&((yadc>=340)&&(yadc<=360))))

printfun('R');

if(((angle1>=40)&&(angle1<=61))&&((angle2>=72)&&(angle2<=84))&&((angle3>=45)&&(angle3<=65))&&((angle4>=44)&&(angle4<=63))&&((angle5>=65)&&(angle5<=86))&&(digitalRead(6)==HIGH))

printfun('T');

if(((angle1>=70)&&(angle1<=90))&&((angle2>=80)&&(angle2<=90))&&((angle3>=0)&&(angle3<=10))&&((angle4>=0)&&(angle4<=10))&&((angle5>=60)&&(angle5<=80)))

printfun('U');

if(((angle1>=70)&&(angle1<=90))&&((angle2>=0)&&(angle2<=10))&&((angle3>=0)&&(angle3<=10))&&((angle4>=0)&&(angle4<=10))&&((angle5>=60)&&(angle5<=80)))

printfun('W');

if(((angle1>=50)&&(angle1<=72))&&((angle2>=45)&&(angle2<=90))&&((angle3>=35)&&(angle3<=75))&&((angle4>=80)&&(angle4<=89))&&((angle5>=45)&&(angle5<=80)))//&&!(((xadc>=412)&&(xadc<=418))&&((yadc>=340)&&(yadc<=360))))

printfun('X');

if(((angle1>=0)&&(angle1<=10))&&((angle2>=70)&&(angle2<=90))&&((angle3>=60)&&(angle3<=80))&&((angle4>=80)&&(angle4<=90))&&((angle5>=15)&&(angle5<=35)))

printfun('Y');
```

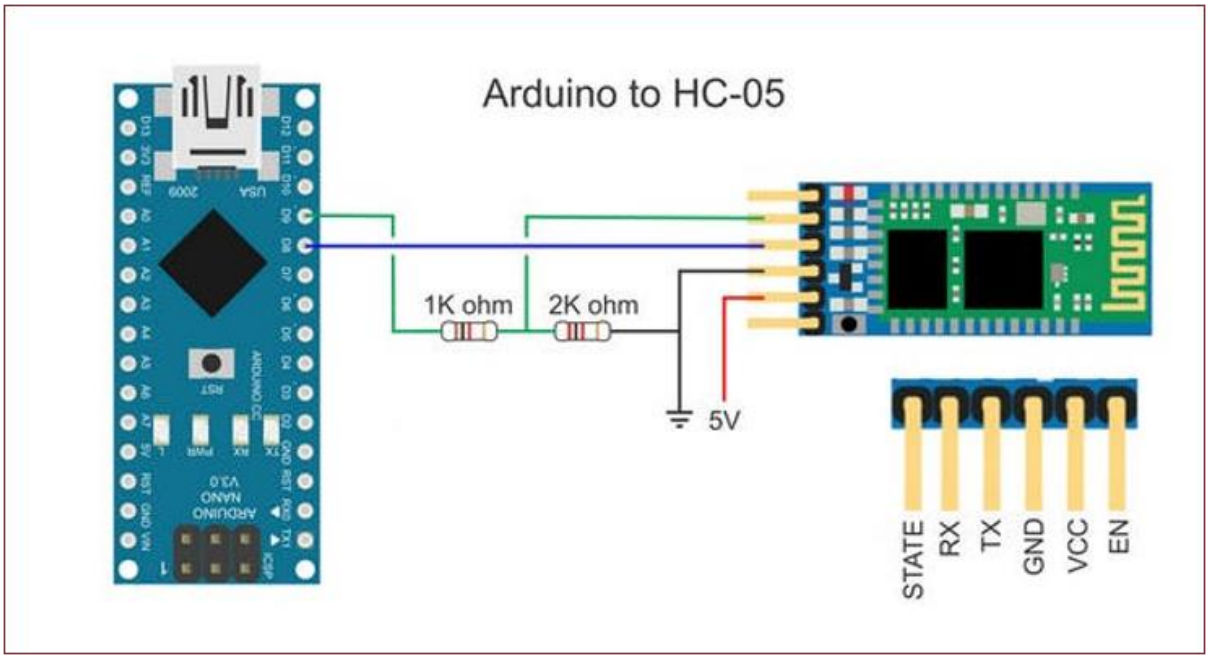
```
if(((angle1>=50)&&(angle1<=72))&&((angle2>=45)&&(angle2<=90))&&((angle3>=35)&&(angle3<=75))&&((angle4>=0)&&(angle
4<=10))&&((angle5>=45)&&(angle5<=80))&&(((xadc>=412)&&(xadc<=418))&&((yadc>=340)&&(yadc<=360))))

printfun('Z');



delay(200);


}

//----------------------END-----------------------------
```

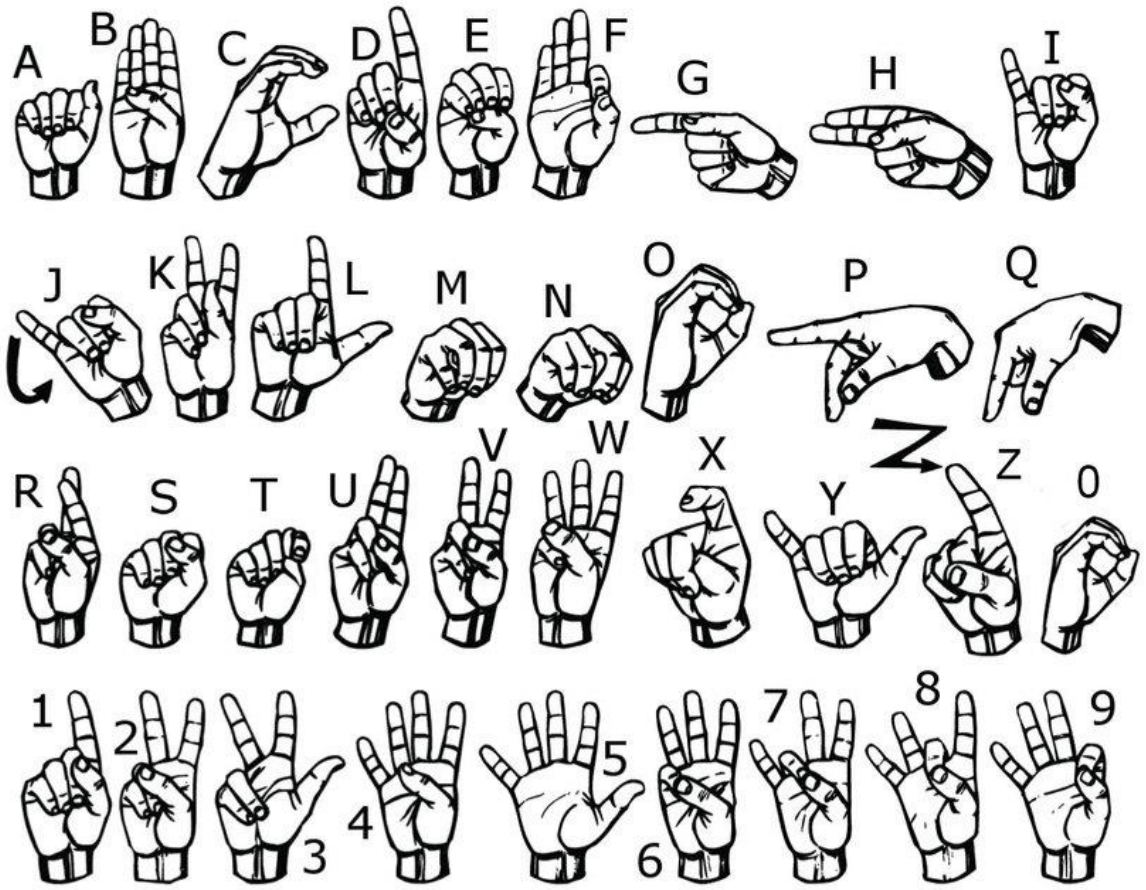
APPENDIX 2

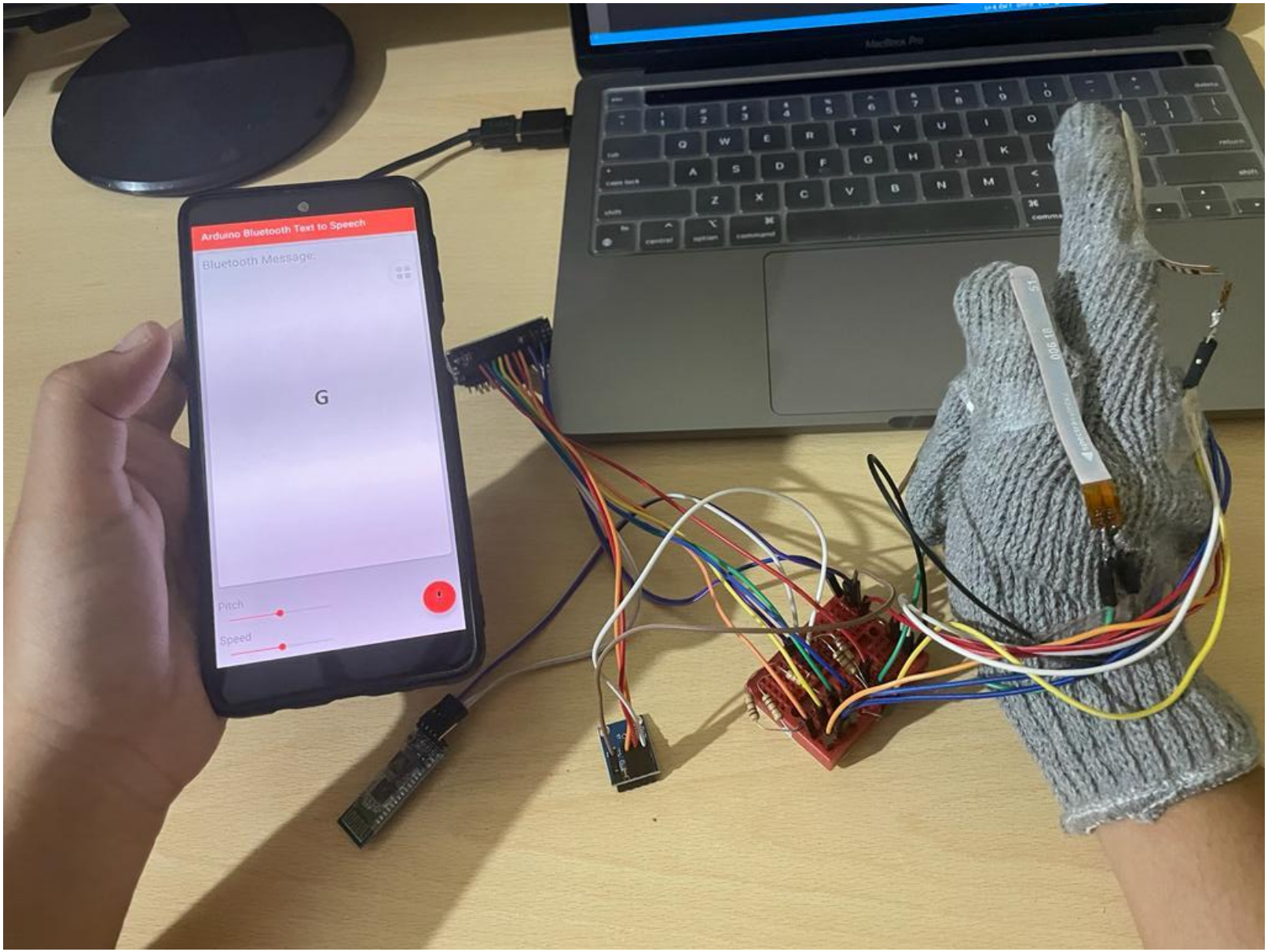


CIRCUIT DIAGRAM




Arduino to HC-05
1K ohm
2K ohm
5V
STATE
RX
TX
GND
VCC
EN


A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z
0
1
2
3
4
5
6
7
8
9


Arduino Bluetooth Text to Speech
Bluetooth Message:
G
Pitch
Speed

APPENDIX 3

S.NO	PART NAME	QUANTITY REQUIRED	APPROX. COST
1	ARDUINO NANO R3	1	550
2	FLEX SENSOR	5	2500
3	RESISTOR	5	50
4	HC-05 Bluetooth model	1	399
5	ADAFRUIT ANALOG ACCELEROMETER	1	230

6	GLOVE	1	200