Sudan University of Science and Technology

Collage of Graduate Studies

Design of a Smart Glove for Gestures Conversion into Text and Speech for Disabled Individuals

تصميم قفاز ذكي لتحويل الإيماءات إلى نص وحديث للأفراد ذوي الإعاقة السمعية

A Research Submitted in Partial Fulfillment for the Requirement of the Degree of M.Sc. in Electronic Engineering (Computer and Network Engineering)

Prepared by:

Fatima Babiker Ahmed Mohammed

Supervisor:

Dr. Ala Eldeen Abd-Allah Awouda

April,2019

الإستهلال

قال تعالى :

(فَتَعَالَى اللَّهُ الْمَلِكُ الْحَقُّ ۗ وَلَا تَعْجَلْ بِالْقُرْآنِ مِنْ قَبْلِ أَنْ يُقْضَىٰ إِلَيْكَ وَحْيُهُ ۖ وَقُلْ رَبِّ زِدْنِي عِلْمًا)

سورة طه(١١٤)

DEDICATION

The sake of Allah, my Creator and my Master, my great teacher and messenger, Mohammed (May Allah bless and grant him), who taught us the purpose of life, My great parents, who never stop giving of themselves in countless ways, my supervisor Dr.Ala Eldeen Awouda who worked hard with me from the beginning till the completion of the present research and was always generous during all phases of the research, To all my family, the symbol of love and giving, My friends who encourage and support me, All the people in my life, Everybody does his best to help others, always give of his knowledge and experiences …

ACKNOWLEDGMENT

In the Name of Allah, the Most Merciful, the Most Compassionate all praise be to Allah, the Lord of the worlds; and prayers and peace be upon Mohamed His servant and messenger.

First and foremost, I must acknowledge my limitless thanks to Allah, the Ever-Magnificent; the Ever-Thankful, for His help and bless.

I highly appreciate the efforts expended by my supervisor Dr. Ala Eldeen Awouda.

I also would like to express my wholehearted thanks to my family for their generous support they provided me throughout my entire life.

I would like to take this opportunity to say warm thanks to all my beloved friends, who have been so supportive along the way of doing my thesis.

ABSTRACT

People with disability (deaf and dumb) using sign language as a language of expression, by using sign language which depend on hand gesture and movement, using smart glove to enable communication between people and people with disability, the smart glove is an electronic device that converts sign language gesture to text and voice, the glove is design and implemented using flex sensor, MPU 6050 sensor, Micro SD Adapter, Speaker, LCD 16X2 and Arduino Nano, by using 5-flex sensor and MPU6050 in order to pick up the gesture of the hand (bending degree of 5-fingers and direction of hand) then the output of flex sensor and MPU6050 is feed to analog input Arduino microcontroller, it processes the signals and perform analog to digital signal conversion to product the gesture ,the Arduino microcontroller compare the gesture with preserved gesture if it mismatched ignore it otherwise there is matched with gesture then from preserved audio (in Micro SD) and text runs equivalent audio and simultaneously display equivalent text to gesture Thus with the help of this Smart glove the barrier faced by these people in communicating with the society can be reduced to a great extent

المستخلص

الأفراد ذوي الإحتياجات الخاصة (الصم والبكم) يستخدمون لغة الإشارة كلغة للتعبير ،ولغة الإشارة هي مجموعة من الإشارت التي تعتمد على حركة وإتجاه اليد والأصابع ،وبإستخدام قفاز ذكي يمكن هؤلاء الأشخاص من التخاطب مع الأشخاص العاديون ،وهذا القفاز يقوم وبتحويل لغة الإشارة إلى نص وصوت يكافئ الإشارة المطلوبة ،وهذا اقفاز مكون من حساس مرونة لقياس زاوية الأصابع الخمسة،وحساس لقياس إتجاه اليد ،ووحدة صوت بها قارئ للذاكرة وذاكرة تخذين وشاشة وسماعة ،يتم قراءة علامة اليد عن طريق حساسات المرونة وحساس إتجاه اليد ويتم تغذية المتحكم بقراءات الحساسات ليقوم بمعالجة الإشارات وتحويلها للعلامة وبعد تحويل القراءات إلى العلامة يتم مقارنتها بالعلامات المحفوظة مسبقا،إذا وجد أنه لايوجد تطابق يتم تجاهلها والذهاب لقرءاة العلامة الجديدة وإذا وجد تطابق يتم تشغيل الصوت في السماعة المحفوظ مسبقا الذي يطابق العلامة وفي نفس اللحظة يتم عرض النص المطابق للعلامة في الشاشة وبهذا يتم مساعدة ذوي الإحتياجات الخاصة وتمكنهم من تحويل علاماتهم إلى كلام ويتمكنون من التخاطب مع الغير وجعل مستقبلهم أفضل .

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ADC	Analog to Digital Converter
AVR	Advance Virtual RISC
H	Hertz
I/O	Input/output
LCD	Liquid Crystal Display
USB	Universal Serial Board

CHAPTER ONE

INTRODUCTION

1.1 Overview

Communication is the only medium by which we can share our thoughts or convey the message but for a person with disability (deaf and dumb) faces difficulty in communication with normal person. Because of this, a person who lacks in hearing and speaking ability is not able to stand in race with normal person.

Communication for a person who cannot hear is visual, not auditory. Generally dumb people use sign language (which depend on hand gesture and movement) for communication but they find difficulty in communicating with others who don't understand sign language. [1]

Sign language uses for communication used by Deaf and Dump person which is not known to normal person for communication they require a translator physically which is not always convenient to arrange, So, there is a barrier in communication between normal person and Deaf and Dump person[2].

To overcome this problem using smart glove to help communication between people and people with disability. the smart glove is a device which can translate the sign hand gestures into voice and text which a normal person can understand, by which smart glove pick up the gesture of hand and translate it into speech and text.

The smart glove consists of sensors fitted to it to interpret the gesture. The glove uses flex sensors gather data on each finger’s position, MPU6050 sensor to detect the direction of hand the data gathering by sensors is faded to Arduino Microcontroller to process it and determine the gesture after gesture was known the equivalent audio run and simultaneously text displayed.

1.2 Problem statement

When a speech impaired person speaks to a normal person, the normal person finds it difficult to understand and asks the deaf-dumb person to show gestures for his/her needs.

1.3 Propose solution

To minimize this barrier, design a device which can convert their hand gestures into text and voice which a normal person can understand.

1.4 Aim and Objectives

The main aim is to proposed system which can give voice and text to voiceless person with the help of Smart Glove.

- To proposed system that convert gesture into voice and text.
- To simulate proposed system.
- To practical implementation of the proposed system.
- test the circuit for data analysis and performance evaluation.

1.5 Scope

This research covers the area of control specially Arduino Microcontroller and voice analysis system such Micro SD Adapter, Speaker along with Flex sensor, MPU6050 sensor and LCD.

1.6 Methodology

The purpose of the project is to covert the hand gesture which is capture by the glove into the audio signal and text ,the glove is consist of Arduino Nano, five Flex sensor , MPU6050 sensor ,Micro SD Adapter and Speaker , 5-flex sensor is attached to the back of glove for each five finger and is used to capture the angle of the finger from straight (0 degree) to bed (90 degree) , MPU6050 is used to attached to the back of glove to capture the direction of the hand , the output of five Flex sensor and MPU6050 sensor is fed to analog input of Arduino ,Micro SD Adapter along with micro SD to store the audio files that to be run by Speaker and LCD display used to monitor the equivalent text to the gesture ,according the gesture capture Arduino loaded the file audio from Micro SD Adapter and then run it at Speaker and display equivalent text at LCD.

1.7 Thesis outline

The thesis will be reported in five chapters:

Chapter one Introduction include.

Chapter two: Literature Review.

Chapter three: System Design.

Chapter four: System Simulation and Result.

Chapter five: Conclusion and Recommendation.

CHAPTER TWO

LITERATURE REVIEW

2.1 Overview

In order to share thoughts and to communicate with person with disability (dumb and deaf) the communication is the only medium so that they can convey the message to others. But there are lots of issues in communication with the person with disability.

Therefore, a person with disability is not able to stand in the race with normal person. As we know that communication for a person who cannot hear is visual, not auditory. Usually dumb people use sign language for communication but they find difficulty in communicating with others who don't understand sign languages. So, this creates barrier in the communication between these two communities. In order to reduce the barrier in communication with normal person is by develop a cost-effective system which can give voice to voiceless person with the help of Smart Gloves. It means that using smart gloves communication will not be barrier between two different communities and they will be able to communicate easily with the normal person. Use of smart glove by person with disability makes nation grow and also, they will not differ themselves from the normal people[2].

2.2 Previous Work

In 2016, Rastogi [3] presented A Review Paper on Smart Glove – Converts Gestures into Speech and Text paper, propose glove to translates the sign language gestures according to the American Sign Language Standard. This glove has been implemented with the help of flex sensors, accelerometer, microcontroller (Arduino Leonardo) and the Bluetooth chip. It Is a wireless data glove which is normal cloth driving glove fitted with flex sensors along the length of each finger.

In 2018, Bagade [4] presented Smart Hand Gloves for Disable People paper, introduce Smart Hand Gloves help disable people to live with normal people. As dumb person cannot speak then the smart gloves helps him to convert his hand gesture into text and pre-recorded voice. And also help normal person to understand what he is trying say and reply accordingly. The Smart Gloves has facility of Home Appliance control from which a physically impaired person become independent to live. The main objective of the implemented project is to develop a reliable, easy to use, light weight smart hand gloves system which can minimizes the obstacles for disable people where they can stand with the race.

In 2016, Fale [5] presented Smart Glove: Gesture Vocalizer for Deaf and Dumb People paper, propose a device which can convert their hand gestures into voice which a normal person can understand. The device consists of a Wireless Glove, consisting of flex sensors and accelerometer. These sensors sense the movement of hands and fingers. The system consists of a speech synthesizer circuit which converts these movements of hand into real time speech output and a display will give the text for the corresponding gesture. The text and voice output being in English. So, this device provides efficient way of communication for both deaf-dumb and normal people.

In 2017 Manware presented [2] Smart Gloves as a Communication tool for the Speech Impaired and Hearing Impaired paper, develop a cost-effective system which can give voice to voiceless person with the help of Smart Gloves. It means that using smart gloves communication will not be barrier between two different communities and they will be able to communicate easily with the normal person. Use of smart glove by person with disability makes nation grow and also, they will not differ themselves from the normal people.

In2017 Mishra and Malhotra presented [6] Design of Hand Glove for Wireless Gesture Control of Robot paper, propose work to control of wheel-based robots by hand gestures has been done with the use of accelerometer, flex sensors and metal contacts between fingers of the glove, and then controlling the robot wirelessly using radio frequency module. The device designed in the present work can sense up to four gestures for various operations of the wheel-based robot. Application of the device in robotics can vary from pick & drop to bomb dismantling.

In 2014 present Verma et al presented [1] Design of Smart Gloves paper, develop a cost-effective system which can give voice to voiceless person with the help of Smart Gloves. project data glove is implemented to capture the hand

gestures of a user Smart gloves having sensors in it captures the movement of user and converts analog input into digital output utilizing voltage divider rule. Then movement is given to microcontroller for further processing. Now gesture array is transmitted using RF transmitter and receiver. Recognized gestures are matched with prefaded data and if it matches given to speaker using voice section.

In 2016 Soni et al [7] presented Missing sense a Smart Glove for Sign to Speech/text Conversion paper, develop a low-cost wearable device which gives voice to a voiceless person, the aim of project is to develop a wearable electronic device which recognizes the hand gestures of the sign language using Flex sensors and translate into an audio output. This will help the integration of deaf and dumb communities into the mainstream society by allowing communication with larger demographic and hence providing greater opportunities.

2.3 Sign Language

The sign language is an important and only method of communication for deaf-dumb persons. As sign language is a formal language employing a system of hand gesture for communication (by the deaf and dumb) [1]. Using here American Sign Language Standard and the Sign Language symbol[8] is shown in figure 2.1 next page.

2.4 Sensors

The sensors were needed in order to pick up the gesture of hand which consist of degree bending of fingers and the direction of hand.

2.4.1 Bend Sensor

A bend sensor consists of three components: a flexible tube, an infrared sensitive (photo diode) and an infrared diode. Infra red-light incident on a

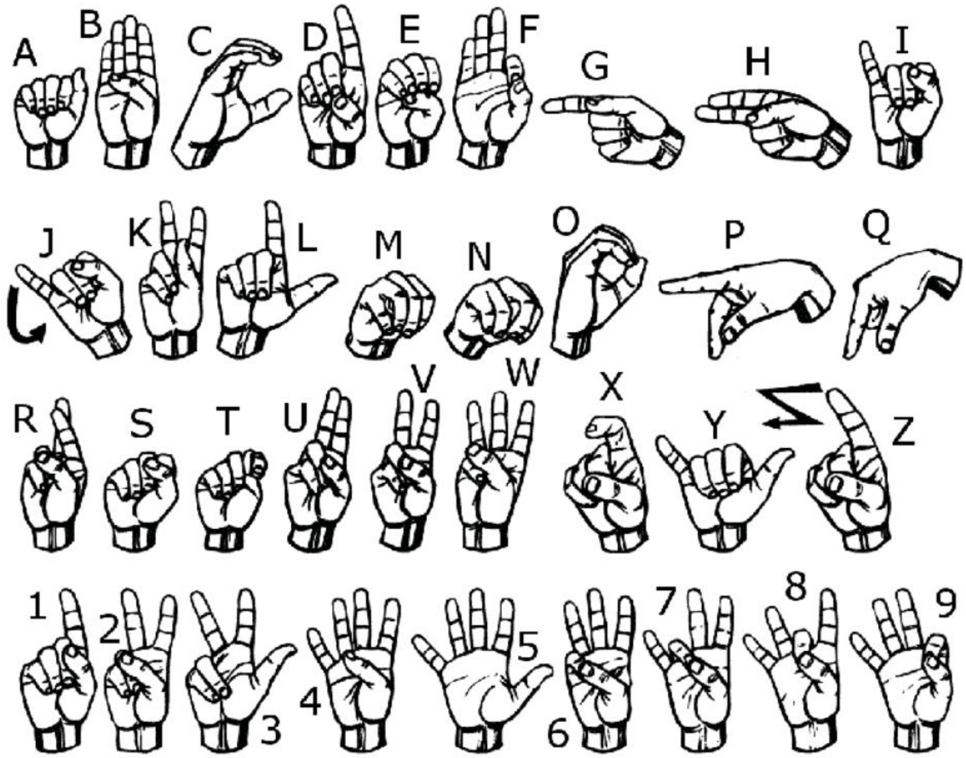

Figure 2.1: sign language symbols[8]

photodiode induces a reverse bias current. This reverse saturation current is directly proportional to the intensity of light incident on the photo diode. The infra-red diode and the photo diode are inserted into a small 5cm pipe from either end, maintaining a certain minimum distance between them. A small incision is made at the center of the pipe to facilitate flexible movement of the bend sensor arrangement show figure 2.2[9].

Figure 2.2: A sample of the bend sensor[9]

2.4.2 Flex Sensor

Flex sensors (or flexion or flection sensors, from the Latin *flectere*, ‘to bend’, also known as bend, bendable, angular displacement or flexible angular sensors).

A flex sensor measures the amount of deflection caused by bending the sensor. Developed from late 80‟s, there are three kinds of flex sensors. Initially optical flex sensors, were created and later conductive ink-based flex sensors and capacitive flex sensors, were developed as alternates to prior, by different people. Although used for sensing „deflection‟, each of the type of flex sensor is different in both construction and working principle.

a) Optical flex sensor

consists of a flexible tube having two ends, a reflective interior wall within the flexible tube and a light source placed within one end and a photosensitive detector placed within the other end of the flexible tube to detect a combination of direct light rays when the flexible tube is bend. convert the change in bend to an electrical resistance variation, that is, they convert a physical energy into an electrical one.

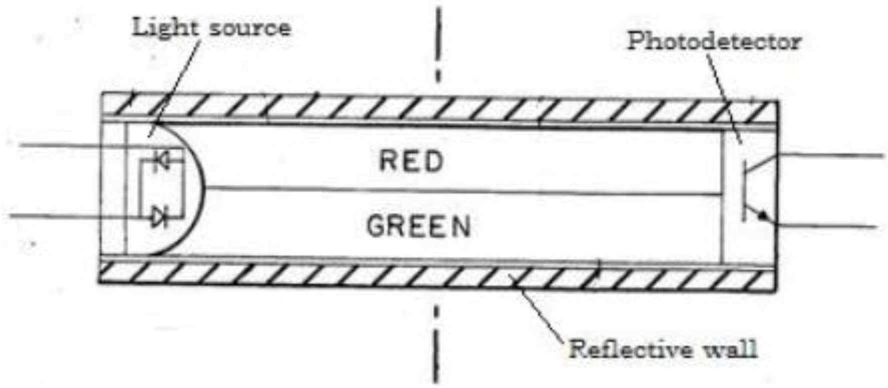

Figure 2.3: Optical flex sensor[10]

b) Conductive ink-based flex sensor

consists of a phenolic resin substrate with conductive ink deposited and thereon a segmented conductor is placed on top to form a flexible potentiometer in which resistance changes upon deflection.

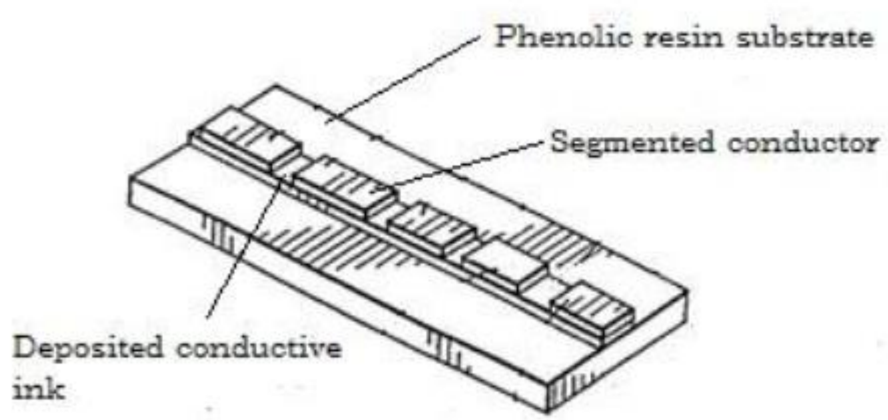

Figure 2.4: Conductive ink-based flex sensor [10]

c) Capacitive flex sensor

has two conductive layers of metals separated by dielectric material in between conductive layers and thereby reducing resistance between them that change in relation with deflection[10].

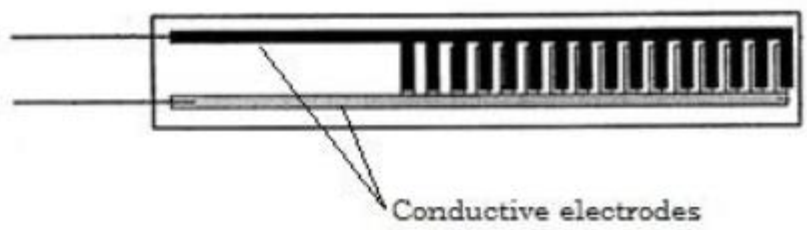

Figure 2.5: Capacitive flex sensor [10]

Among flex sensors, a special role is played by the passive resistive ones. They are made of electrically conductive patterns, engineered on top of or within a flexible substrate (schematized in figure 2.2) that is able to tolerate bending, vibration, thermal shock (within normal thermal excursion of usage) and stretching, without concern for electromagnetic interference or sensor occlusion. is dedicated to some fundamental applications concerning biometric measurements of upper limbs (finger, hand, wrist, elbow and arm), head and torso (neck, lung, larynx, chest and back), muscle shape and lower limbs (knee, leg, ankle and foot). These applications are intended for medical purposes (rehabilitation, kinesiology, diagnostic, fitness), for human–machine interfaces, for gesture modeling and recognition. RFSs can be particularly useful in developing systems for interaction with and/or interfacing virtual/ augmented reality ambience and for human–computer interaction possibilities [11].

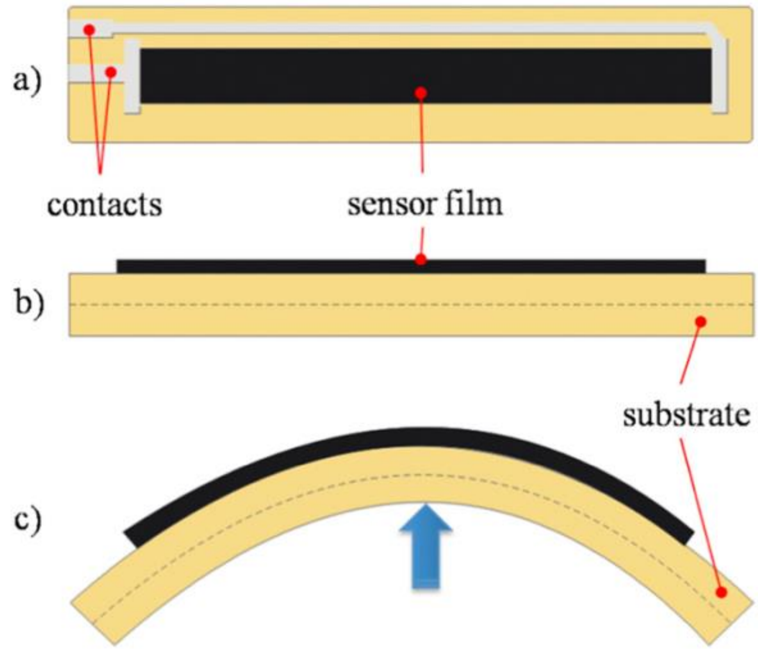

Figure 2.6: resistive flex sensor [11]

Figure 2.2 resistive flex sensor. (a) Top view: electrical contacts in grey, conductive film in black. (b) Lateral view: conductive film, in black, on top of a substrate, in a lighter color. (c) Bending the substrate causes a mechanical stress of the conductive pattern that leads to a change in its electrical resistance.

Flex means “bend “or “curve”, the suitable sensor for measure amount of bend a finger is Flex sensor employed for each finger, Flex sensor basically variable resistance which terminal resistance will increase when sensor bent.

When sensor placed in gloves is bent, it produces a resistance output correlated to the bend radius- the smaller the radius, the higher the resistance value the Electrical Specifications shown on table 2.1.

Electrical characterization

An RFS is a passive device which does not require any bias or source power to work. When the sensor is flexed the substrate is consequently compressed and the conductive layer stretches, thereby increasing the overall resistance up to a maximum value corresponding to the maximum measurable angle of deflection[11].

Table 2.1 Electrical Specifications for Flex Sensor

Operating voltage	0-5V
can operate on LOW voltage	-
Power Rating	0.50 Watts continuous. 1-Watt Peak.
Life Cycle	>1 million
Height	0.43mm (0.017")
Temperature Range	-35°C to +80°C
Flat Resistance	25K Ohms
Resistance Tolerance	±30%
Length	15 cm

The figure 2.7 below show the equivalent change of flex sensor resistance according to bend radius.

basic flex sensor circuit consist of two or three sensors are connected. The output from the flex sensors are inputted into op-amps and used a non-inverted style setup to amplify their voltage, the greater the degree of bending the lower the output voltage. By voltage divider rule, output voltage is determined and given by.

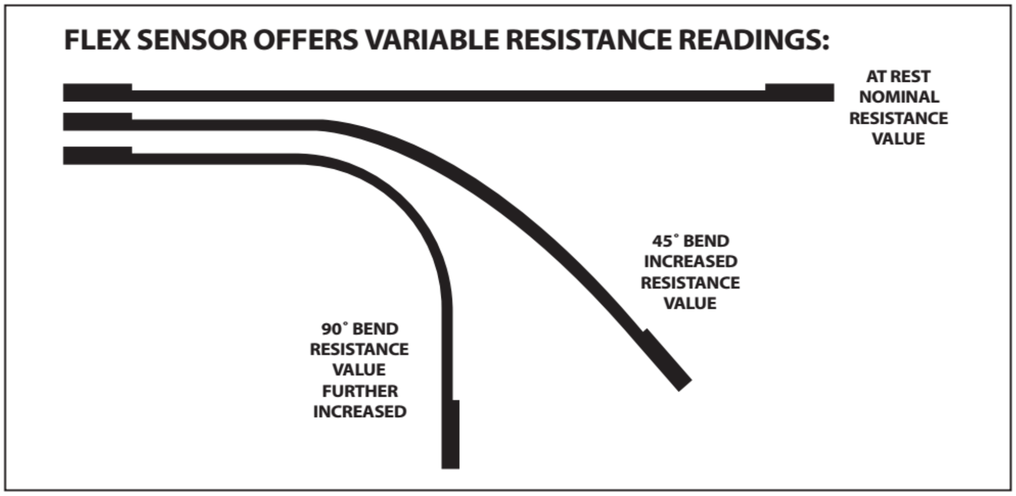

Figure 2.7: Flex Sensor offers variable resistance readings[1]

Vout = Vin *R1 / (R1 + R2) …... (2.1)).

Vout: output.

Vin: input.

R: resister.

Where R1 is the other input resistor to the non-inverting terminal[1] ,show figure2.8.

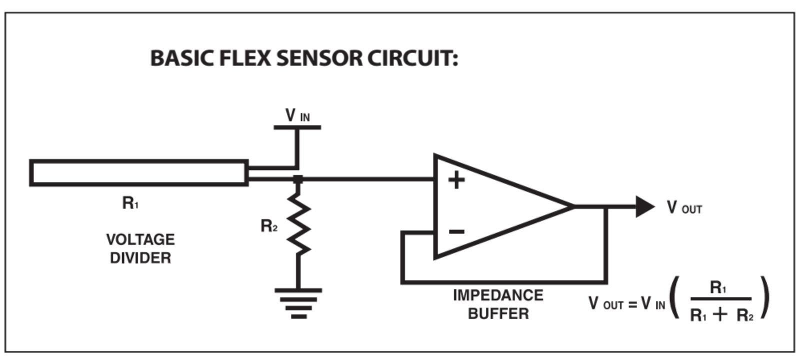

figure2.8: basic Flex Sensor circuit[1]

figure 2.5 show the flex sensor.

Figure 2.9: Flex sensor

2.4.3 Accelerometer Sensor

The ADXL335 is a small, thin, low power 3-axis accelerometer which measures acceleration with a minimum full-scale range of ±3g along with measurement of the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration resulting from motion, shock, or vibration. Tilting an accelerometer along its measured axis, gives the gravitational force relative to the amount of tilt.

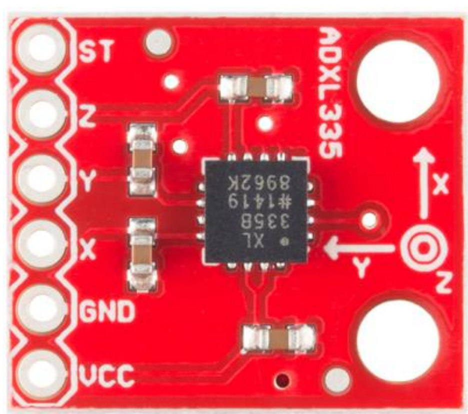

Figure.2.10: Accelerometer [12]

As shown in Figure.2.10, It measures ±3 g in three orthogonal axes labeled the X, Y and Z direction. It can read in the range of 0.5 Hz to 1600 Hz for the X and Y axis while the Z axis has a range of 0.5 Hz to 550 Hz. However, the Spark Fun model comes mounted with 0.1 μF capacitors that acts as a low-pass filter and limits the lower bandwidth of each axis to 50Hz. to operate the accelerometer, it needs between 1.8 V to 3.6 V so we can't use the 5 V output on the Arduino and have to use the 3 V. This also means that simply plugging one of the axes into one of the Arduinos analog input pins will lead to complications since it expects a maximum value to be 5 V. To solve this, we connect the supply voltage to the AREF pin on the Arduino as well as to the accelerometer and in the software tell the Arduino to use this voltage as a reference instead of the default 5 V.

The ADXL335 is a so-called MEMS (Micro Electro Mechanical System) accelerometer. The sensor in the ADXL335 is a polysilicon surface micromachined sensor that is built on a silicon wafer. In the sensor there is a proof mass called a seismic mass that is tethered to deflectable plates. When subjected to acceleration the plates are deflected by the mass and this deflection is measured by a differential capacitor. The differential capacitor is made of independently fixed plates and the plates that are connected to the seismic mass. The fixed plates are driven by 180*!* out of phase square waves and when the plates are deflected the differential capacitor gets unbalanced and gives an output signal of a square wave whose amplitude is proportional to the acceleration.

By using demodulation techniques that are sensitive to the phase-magnitude and direction of the acceleration can be determined. The signal is then amplified and taken through a 32-kW resistor and now one signal for each axis is available. Each signal is then taken through a 0.1 μF capacitor that as was mentioned earlier acts as a low-pass filter. The ADXL335 uses one structure for the X, Y and Z axis which gives the axis high orthogonality that in turn leads to little cross-axis sensitivity[12].

2.4.5 MPU6050

MPU6050 is a 6-axis-axis motion sensor with 3 axis gyroscopes and a 3 axis accelerometer. And an extensible digital motion processor DMP, IIC or SPI interface can be connected to other sensors, the output is a 9-axis signal. Angle detection in industrial production and practice has a lot of applications, such as the detection of the object's horizontal angle or tilt angle, can make the balance car or four rotor aircraft, to detect the object's attitude, that is, the spatial angle.

According to the angle, a certain control algorithm is adopted, and the PID algorithm is used to balance the object. Angle detection can also be used to detect the elderly fall, this year, with the development of society, China has gradually entered the aging, the elderly population continues to increase. Old people's motor function decline, so the detection of the fall of electronic products produced. The front end of the sensor is MPU6050.

The MPU6050 pin diagram is shown in figure 2.11. Figure VDD supply pin is supplied by 3.3V, CS is a chip select signal. Data transfer using SPI or IIC interface. Figure 2.8 is a three-dimensional view of MPU6050 for angle detection.

MPU6050 has three 16-bit ADC respectively, to collect the acceleration value of the 3 axis or gyro value, which is converted to digital output. The range of gyroscopes measurement is plus or minus 250 degrees, plus or minus 500 degrees, plus or minus 1000 degrees, plus or minus 2000 degrees, the accelerometer measurement range is + 2G, + 4G, + 8g, + 16g. on chip 1MB FIFO, can be used for data cache show figure 2.12. The serial communication interface, IIC rate can reach 400K, SPI rate is up to 1M., digital motion inside the DMP gyroscope SPI interface can reduce the complexity of the data fusion, the accurate output value of the angle[13].

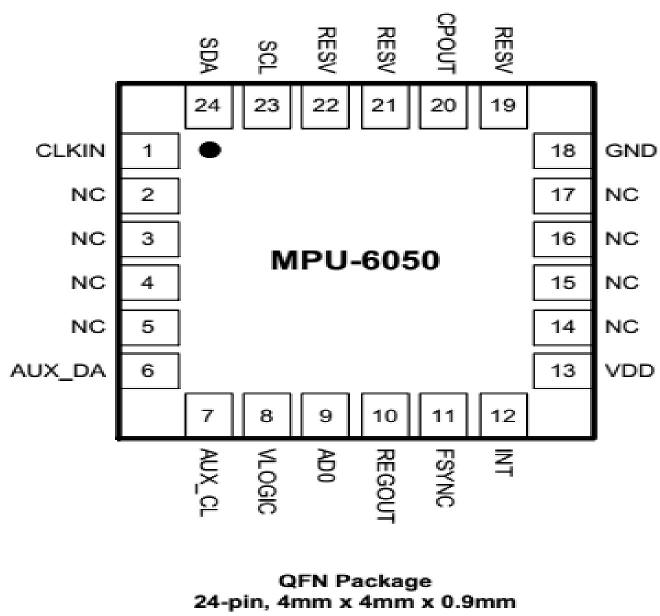

Figure 2.11: MPU6050 Pins [13]

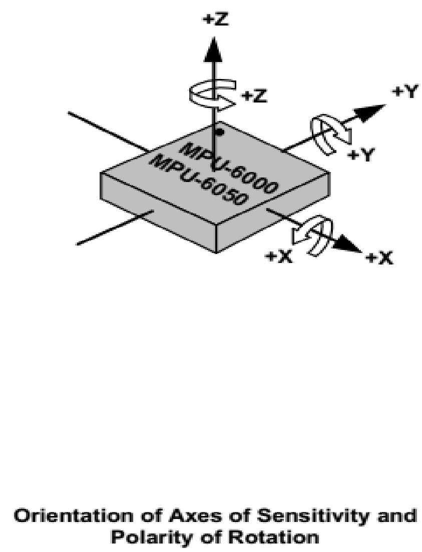

Figure 2.12: Three axis test chart [13]

The dimensions of MPU6050 are 4 mm x 4 mm x 0.9 mm QFN package. The MPU-6050 sensor has run-time calibration facility. It is provided with the calibration firmware so it avoids the costly and complex system level integration of the devices which are required for calibration procedures. Due to this, inertial navigation system gives good performance show figure 2.13[14].

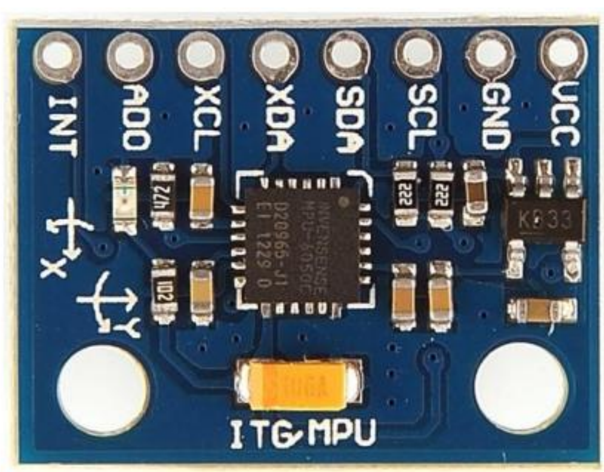

Figure 2.13: MPU 6050 [14]

Table 2.2 MPU6050 Pin Configuration

Pin Number	Pin Name	Description
1	Vcc	Provides power for the module, can be +3V to +5V. Typically +5V is used
2	Ground	Connected to Ground of system
3	Serial Clock (SCL)	Used for providing clock pulse for I2C Communication
4	Serial Data (SDA)	Used for transferring Data through I2C communication
5	Auxiliary Serial Data (XDA)	Can be used to interface other I2C modules with MPU6050. It is optional
6	Auxiliary Serial Clock (XCL)	Can be used to interface other I2C modules with MPU6050. It is optional
7	AD0	If more than one MPU6050 is used a single MCU, then this pin can be used to vary the address
8	Interrupt (INT)	Interrupt pin to indicate that data is available for MCU to read.

2.4.6 Micro SD Card Adapter

Micro SD Card Micro SDHC Mini TF Card Adapter Reader Module for Arduino show figures 2.14 and 2.15 below.

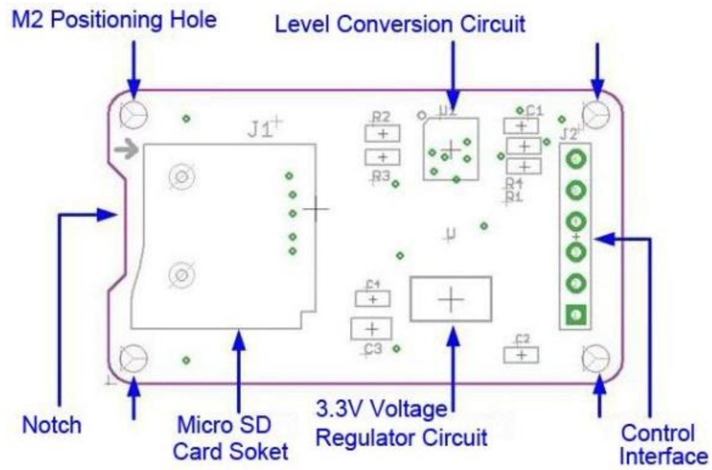

Figure 2.14: Micro SD Adapter pin [15]

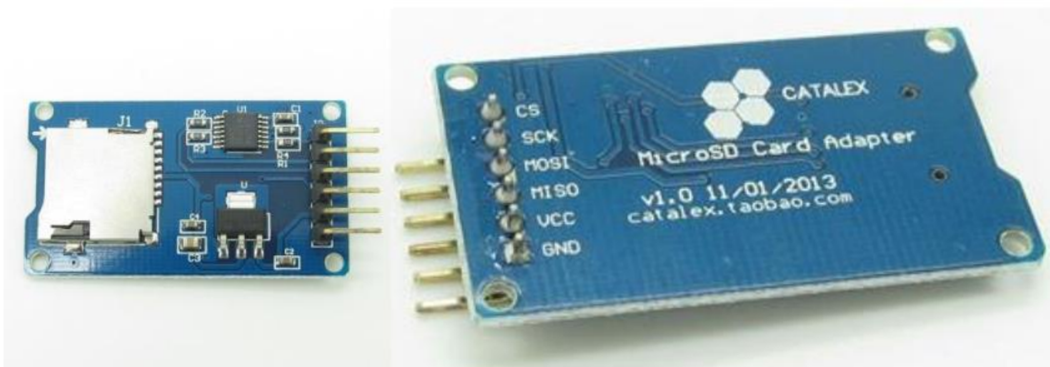

Figure 2.15: Micro SD Adapter [15]

Description

• The module (MicroSD Card Adapter) is a Micro SD card reader module for reading and writing through the file system and the SPI interface driver, SCM system can be completed within a file MicroSD card.

• Support Micro SD Card, Micro SDHC card (high speed card).

• Level conversion circuit board that can interface level is 5V or 3.3V.

• Power supply is 4.5V ~ 5.5V, 3.3V voltage regulator circuit board.

• Communications interface is a standard SPI interface.

• 4 M2 screws positioning holes for easy installation.

• Control Interface: A total of six pins (GND, VCC, MISO, MOSI, SCK, CS), GND to ground, VCC is the power supply, MISO, MOSI, SCK for SPI bus, CS is the chip select signal pin.

• 3.3V regulator circuit: LDO regulator output 3.3V for level conversion chip, Micro SD card supply.

• Level conversion circuit: Micro SD card to signal the direction of converts 3.3V, MicroSD card interface to control the direction of the MISO signal is also converted to 3.3V, general AVR microcontroller systems can read the signal.

• Micro SD card connector: self-bomb deck, easy card insertion.

• Positioning holes: 4 M2 screws positioning holes with a diameter of 2.2mm, so the module is easy to install positioning, to achieve inter-module combination[15].

Table 2.3 Interface Parameters for Micro SD Adapter

Item	Min	Typical	Max	Unit
Power voltage VCC	4.5	5	5.5	V
Current	.2	80	200	mA
Interface Electrical Potential	3.3 or 5			V
Support Card Type	Micro SD Card(<=2G), Micro SDHC (<=32 G)			-
Size	24X24X12			Mm
Weight	5			G

2.4.7 Micro SD

Using micro SD 2GB for storing voices show figure 2.16**.**

Figure 2.16: Micro SD

2.4.8 Speaker

This 0.5W, 16 Ohm speaker is used as loud speaker the voice is being hearing by it, has two terminal one for signal the other for ground show figure2.17.

Table 2.4: Electrical and Mechanical Characteristics for speaker[16].

No	Item	Specification
1	Impedance (at 2kHz)	16±15%Ω
2	Rated Input Power	0.5W
3	Maximum Input Power	0.8W
4	Resonance Frequency	850±20%Hz
5	Frequency Response	F0～20KHz
6	Output SPL	90±3dB/0.1W0.1m at0.8,1.0,1.2,1.5KHz Average
7	Distortion at 1kHz,0.5W）	≤5%
8	Buzzes & Rattles	Must be normal at sine wave 2.83V
9	Operating Temperature	-20～+70°C
10	Storage Temperature	-40～+85°C

Figure: 2.17 Speaker

2.4.9 Liquid Crystal Display (LCD)

LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. These modules are preferred over seven segments and other multi segment LEDs. The reasons being: LCDs are economical; easily programmable; have no limitation of displaying special & even custom characters (unlike in seven segments), animations and so on.

A 16x2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command and Data. The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD[17]. Show figure 2.14 and table 2.5 below.

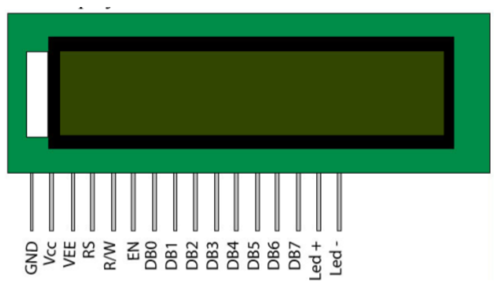

Figure 2.18: pin diagram of 16x2 LCD [17]

2.5 Microcontroller

Table 2.5. Interface PIN Description of LCD

Pin No	Function	Name
1	Ground	0V -Vss
2	Supply voltage	5V -VDD
3	Contrast adjustment	Vo
4	Register Select	Rs
5	Read / Write signal	R / W
6	Data read / write Enable signal	E
7	Low order data bus used for data transfer between microcontroller and LCD display	D0 – D3
8		
9		
10		
11	High order data bus used for data transfer between microcontroller and LCD display	D4 – D7
12		
13		
14		
15	Backlight 5V	LED +
16	Backlight 0V	LED -

A microcontroller is a small and low-cost computer built for the purpose of dealing with specific tasks, such as displaying information on seven segment display at railway platform or receiving information from a television's remote control. Microcontrollers are mainly used in products that require a degree of control to be exerted by the user.

Today various types of microcontrollers are available in market with different word lengths such as 8bit, 16bit, 32bit, and microcontrollers.
Microcontroller is a compressed microcomputer manufactured to control the functions of embedded systems in office machines, robots, home appliances, motor vehicles, and a number of other gadgets. Therefore, in today's technological world lot of things done with the help of Microcontroller.

2.5.1 Classifications of Microcontrollers

The microcontrollers are characterized regarding bus-width, instruction set, and memory structure. For the same family, there may be different forms with different sources. the basic types of the Microcontroller that newer users may not know about.

A) Classification according to number of bits:

1) The 8-bit microcontroller: Means CPU or ALU can process 8-bit data at a time. The examples of 8-bit microcontrollers are Intel 8031/8051. These are used in position control, speed control applications.

2) The 16-bit microcontroller: It performs greater precision and performance as compared to 8-bit. These are developed for the purpose of high speed applications such as servo control system, Robotics etc. Some examples of 16-bit microcontroller are 16-bit MCUs are extended Intel 8096 and Motorola MC68HC12 families.

3) 32-bit microcontroller: It uses the 32-bit instructions to perform the arithmetic and logic operations These are developed for the purpose of very high-speed application in Image processing, Telecommunications, Intelligent control system etc. Some examples are Intel/Atmel 251 family, PIC3x, ARM.

B. Classification according to memory devices:

1) Embedded memory microcontroller: When an embedded system has a microcontroller unit that has all the functional blocks available on a chip is called an embedded microcontroller. For example, 8051 having program & data memory, I/O ports, serial communication, counters and timers and interrupts on the chip is an embedded microcontroller.

2) External Memory Microcontroller: When an embedded system has a microcontroller unit that has not all the functional blocks available on a chip is called an external memory microcontroller. For example, 8031 has no program memory on the chip is an external memory microcontroller.

C. Classification according to instruction set:

1) CISC architecture: CISC means complex instruction set computer, it allows the user to apply 1 instruction as an alternative to many simple instructions.

2) RISC architecture: RISC means Reduced Instruction Set Computers. RISC reduces the operation time by shortening the clock cycle per instruction. The RISC gives a better execution than the CISC.

D) Classification according to memory architecture:

1) Harvard Memory Architecture Microcontroller: The point when a microcontroller unit has a dissimilar memory address space for the program and data memory, the microcontroller has Harvard memory architecture in the processor. The RISC gives a better execution than the CISC.

2)Princeton Memory Architecture Microcontroller: The point when a microcontroller has a common memory address for the program memory and data memory, the microcontroller has Princeton memory architecture in the processor.

2.5.2 Types of Microcontroller

A. 8051 Microcontroller

8051 microcontrollers are an eight-bit microcontroller invented in 1981 by Intel Corporation. It is available in 40 pin DIP i.e. dual in line package. This is the basic Microcontroller but still many companies are manufacturing such types of Microcontroller. The older types of 8051 have 12 clocks per instruction that make it sluggish whereas the recent 8051 have 6 clocks per instruction. The 8051 microcontrollers do not have an in-built memory bus and A/D converters and such Microcontrollers are CISC processors, also 8051 uses Von Neuman architecture show figure 2.19.

Figure2.19: Intel 8051 Microcontroller [18]

Various features of 8051 microcontroller are given as follows.

1) 8-bit CPU

2) 16-bit Program Counter

3) 8-bit Processor Status Word (PSW)

4) 8-bit Stack Pointer

5) 4K bytes internal ROM (program memory).

6) 128 bytes internal RAM (data memory).

7) Special Function Registers (SFRs) of 128 bytes

8) 32 I/O pins arranged as four 8-bit ports (P0 – P3)

9) Two 16-bit timer/counters: T0 and T1

10) Two external and three internal vectored interrupts

11) One full duplex serial I/O.

B. PIC Microcontroller

Peripheral interface controller is a family of Microcontrollers by Microchip technology USA with Harvard architecture. Originally this was developed as supporting device for PDP (program data processor) computers to support for its peripheral devices and therefore named as PIC. PIC Microcontrollers are RISC processors. An interesting thing about PIC is that its machine cycle consists of only 4 clock pulses in contrast with 12 clock pulses in Intel 8051 Microcontroller. PIC microcontrollers are finding their way into new applications like smart phones, audio accessories, video gaming peripherals and advanced medical devices show figure 2.20.

Fig2.20: PIC 16F877 [18]

Various features of PIC 16F877 microcontroller are given as follows

1) High-performance RISC CPU

2) Up to 8K x 14 words of FLASH program memory

3) 35 Instructions (fixed length encoding-14-bit)

4) 368×8 static RAM based data memory

5) Up to 256 x 8 bytes of EEPROM data memory

6) Interrupt capability (up to 14 sources)

7) Three addressing modes (direct, indirect, relative)

8) Power-on reset (POR)

9) Harvard architecture memory

10) Power saving SLEEP mode

11) Wide operating voltage range: 2.0V to 5.5V

12) High sink / source current: 25mA

13) Accumulator based machine.

C. ARM Microcontroller

ARM is 32-bit Microcontroller whose core is designed by ARM Limited with RISC architecture. ARM has von Neumann architecture (program and RAM in the same space). ARM Microcontrollers are extremely used in power saving and operate in very low power consumption. ARM Microcontrollers Widely used in modern handset for mobile communications. These are also used in various other embedded system likes iPod, hand held gaming unit, disk driver and so on. 8051 and PIC need multiple clock cycles per instruction. AVR and ARM execute most instructions in a single clock cycle show figure 2.21.

Fig2.21: ARM microcontroller [18]

Various features of ARM microcontroller are given as follows:

1) Maximum single cycle functioning

2) Constant 16×32-bit register file.

3) Load or store architecture.

4) Preset instruction width of 32 bits so as to simplify pipe-lining and decoding, at minimized code density.

5) For misaligned memory access there is no support.

D. AVR Microcontroller

The AVR is a modified Harvard RISC architecture 8-bit RISC single-chip microcontroller, which is developed by Atmel in 1996. The AVR is stands for Alf-Egil Bogen and Vegard Wollan's RISC processor. AVR takes only one clock per instruction shpw figure 2.18.

AVR Microcontrollers are classified into three types:

a) TinyAVR – Less memory, small size, suitable only for simpler applications.

b) MegaAVR – These are the most popular ones having good amount of memory (up to 256 KB), higher number of inbuilt peripherals and suitable for moderate to complex applications.

c) XmegaAVR – Used commercially for complex applications, which require large program memory and high speed.

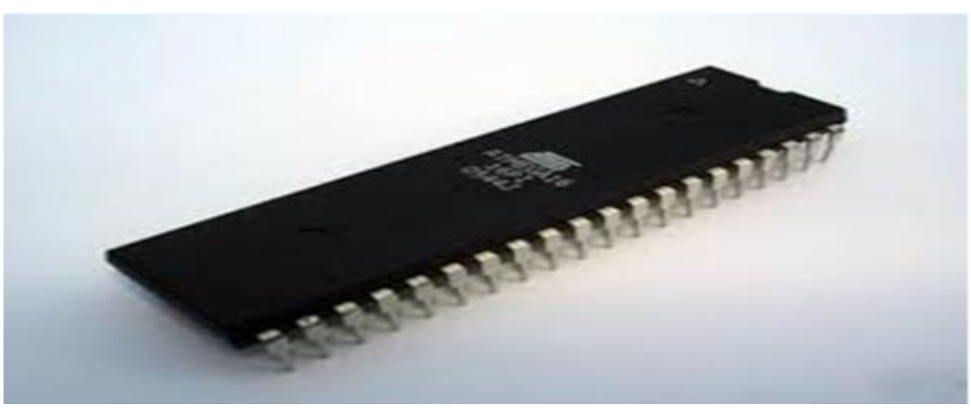

Fig2.22 ATMEGA 16 [18]

Some of the features of Atmega16 are:

a) 16KB of Flash memory

b) 1KB of SRAM

c) 512 Bytes of EEPROM

d) Available in 40-Pin DIP

e) 8-Channel 10-bit ADC

f) Two 8-bit Timers/Counters

g) One 16-bit Timer/Counter

h) 4 PWM Channels

i) In System Programmer (ISP)

j) Serial USART

k) SPI Interface

l) Digital to Analog Comparator[18].

2.6 Arduino

Arduino is an open source microcontroller which can be easily programmed, erased and reprogrammed at any instant of time. Introduced in 2005 the Arduino platform was designed to provide an inexpensive and easy way for hobbyists, students and professionals to create devices that interact with their environment using sensors and actuators. Based on simple microcontroller boards, it is an open source computing platform that is used for constructing and programming electronic devices. It is also capable of acting as a mini computer just like other microcontrollers by taking inputs and controlling the outputs for a variety of electronics devices.

It is also capable of receiving and sending information over the internet with the help of various Arduino shields, which are discussed in this paper. Arduino uses a hardware known as the Arduino development board and software for developing the code known as the Arduino IDE (Integrated Development Environment). Built up with the 8-bit Atmel AVR microcontroller's that are manufactured by Atmel or a 32-bit Atmel ARM, these microcontrollers can be programmed easily using the C or C++ language in the Arduino IDE.

Unlike the other microcontroller boards in India, the Arduino boards entered the electronic market only a couple of years ago, and were restricted to small scale projects only. People associated with electronics are now gradually coming up and accepting the role of Arduino for their own projects. This development board can also be used to burn (upload) a new code to the board by simply using a USB cable to upload. The Arduino IDE provides a simplified integrated platform which can run on regular personal computers and allows users to write programs for Arduino using C or C++.

The elements of an Arduino Board can be done into two categories:

•Hardware

• Software

2.6.1 Hardware

The Arduino Development Board consists of many components that together makes it work. Here are some of those main component blocks that help in its functioning:

• Microcontroller**:** This is the heart of the development board, which works as a mini computer and can receive as well as send information or command to the peripheral devices connected to it. The microcontroller used differs from board to board; it also has its own various specifications.

• External Power Supply: This power supply is used to power the Arduino development board with a regulated voltage ranging from 9 – 12 volts.

• USB plug: This plug is a very important port in this board. It is used to upload (burn) a program to the microcontroller using a USB cable. It also has a regulated power of 5V which also powers the Arduino board in cases when the External Power Supply is absent.

• Internal Programmer: The developed software code can be uploaded to the microcontroller via USB port, without an external programmer.

• Reset button: This button is present on the board and can be used to resets the Arduino microcontroller.

• Analog Pins: There are some analog input pins ranging from A0 – A*7 (typical)*. These pins are used for the analog input / output. The no. of analog pins also varies from board to board.

• Digital I/O Pins: There are some digital input pins also ranging from 2 to *16 (typical)*. These pins are used for the digital input / output. The no. of these digital pins also varies from board to board.

• Power and GND Pins: There are pins on the development board that provide 3.3, 5 volts and ground through them.

2.6.2 Software

The program code written for Arduino is known as a sketch. The software used for developing such sketches for an Arduino is commonly known as the Arduino IDE. This IDE contains the following parts in it:

•Text editor: This is where the simplified code can be written using a simplified version of C++ programming language.

• Message area: It displays error and also gives a feedback on saving and exporting the code.

• Text: The console displays text output by the Arduino environment including complete error messages and other information.

• Console Toolbar: This toolbar contains various buttons like Verify, Upload, New, Open, Save and Serial Monitor. On the bottom right hand corner of the window there displays the Development Board and the Serial Port in use.

2.6.3 Features of Arduino IDE

• The project file or the sketches for a project are saved with the file extension. ino.

• Features such as cut / copy / paste are supported in this IDE.

• There also is a facility for finding a particular word and replacing it with another by pressing the Ctrl + F buttons on the keyboard.

• The most basic part or the skeleton of all Arduino code will have two functions

2.6.4 Programming Basics

Now we'll discuss about the programming techniques of Arduino sketch in the Arduino IDE. There are two main parts every sketch will always have, they are:

•void setup ()

• void loop ()

1) void setup ()

This is the first routine that begins when the Arduino starts functioning. This function is executed only once throughout the entire program functioning.

The setup function contains the initialization of every pin we intend use in our project for input or output.

2) void loop ()

This function is the next important function in the Sketch. It consists of that part of the code that needs to be continuously executed unlike the part of the code written in the setup function.

2.6.5 Types of Arduino Board

A) Arduino Uno

Processor: ATmega328 (8- bit CPU, 16MHz clock speed, 2KB SRAM, 32KB flash storage).

Features: 14 digital I/O pins, 6 analog input pins, removable microcontroller.

Advantage: Microcontroller can be removed and replaced from the socket in case of breakdown.

Limitation: Doesn’t have a lot of SRAM or flash memory that limits the kinds of programs you can load on the chip.

B) Arduino Leonardo

Processor: ATmega32u4 (8- bit CPU, 16MHz clock speed, 2.5KB SRAM, 32KB flash storage).

Features: 20 digital I/O pins, 12 of which is used as analog inputs, native USB support Advantage: ATmega32u4 has built -in USB communication (compatibility) eliminating the need for secondary processor.

Leonardo to interface with PC, which sees it as a generic mouse or keyboard. It also has a few extra analog input pins.

Limitation: Still has a few bugs that need ironing out and isn’t quite as beginner friendly as the Uno.

C)Arduino Due

Processor: Atmel SAM3X8E ARM Cortex-M3 (32-bit CPU, 84MHz clock speed, 96KB SRAM, 512KB flash storage).

Features: 54 digital I/O pins, native USB port, 12 analog input pins, 2 analog output pins.

Advantage: The Due is primarily for more complicated projects that can make use of its muscular processor, otherwise that needs more I/O pins than are found on the smaller Arduino boards.

Limitation: It operates at 3.3 volts that limits the add-on hardware that's compatible with the Arduino Due-if an add-on board tries to send a 5-volt signal to the Due's I/O pins, it would damage the microcontroller.

D)Arduino Micro

Processor: ATmega32u4 (8- bit CPU, 16MHz clock speed, 2.5KB SRAM, 32KB flash storage).

Features: 20 digital I/O pins, 12 of which is used as analog inputs, native USB support.

Advantage: Includes all of the power and functionality of a full- sized Arduino Leonardo board in a much smaller form factor. It is designed to easily slot into a bread board, for faster prototyping.

Limitation: Due to the small form factor, Arduino Micro will not work with many add-on boards[19].

E) Arduino ESPLORA

Processor: ATmega32u4 (8- bit CPU, 16MHz clock speed, 2.5KB SRAM, 32KB flash storage).

Features: Lots of built-in input and output hardware.

Advantage: A whole bunch of I/O hardware soldered directly to the board. On input side you get a joystick, four buttons, a linear potentiometer (slider), a microphone, a light sensor, a temperature sensor and a three-axis accelerometer. For outputs, you get an RGB led, a buzzer and a TFT display connector to attach an LCD screen (not included).

Limitation: The tradeoff is that you do not get the standard set of digital and analog I/O pins, which allows you to wire up all sorts hardware to your Arduino board.

f) Arduino Yun

Processor: ATmega32u4 (8- bit CPU, 16MHz clock speed, 2.5KB SRAM, 32KB flash storage), Atheros AR9331 system on the chip.

Features: Wi-Fi enabled Linux based system on a chip, 14 digital, analog I/O pins, and 12 of which can be used as analog inputs. Native USB.

Advantage: It is easier to connect to cloud-based services from the Arduino platform. It features a separate Linux-based system-on-a chip on the motherboard.

Limitation: The low-bandwidth, low memory, microcontrollers have a hard time handling the verbose protocols used to access those services.

g) Arduino Pro mini

Processor: ATmega168, Clock speed 8MHz (3.3v model) or 16MHz (5v model), SRAM-1 KB, Flash Memory-16 KB.

Features: Operating voltage-3.3V or 5V, Input Voltage-3.3V- 12V, Digital I/O pins-14, Analog Input pins-8.

Advantage: Rather than requiring a physical press of the reset button before an upload, the Arduino Pro Mini is designed in a way that allows it to be reset by software running on a connected computer.
Limitation: The Arduino pro mini is compact in size. Its size is about 1.3*0.70".

h) Arduino Robot

Processor: 2 x ATmega32u4 (8-bit CPU, 16MHz clock speed, 2.5KB SRAM, 32KB flash storage).
Features: Wheels, 8 analogue input pins, 6 digital I/O pins, LCD screen.
Advantage: A little robot composed of two separate boards (a control board and a motor board) that each feature the Leonardo's ATmega32u4 processor. Though it's designed with room to add your own custom hardware
Limitation: More expensive than other Arduino boards on account of having two separate boards.

i) Lilypad Arduino

Processor: ATmega328 (8- bit CPU, 16MHz clock speed, 2KB SRAM, 32KB flash storage).
Features: 14 digital I/O pins, 6 analog input pins
Advantage: Basically, designed for wearable's and e -textiles (fabric-based projects).
Limitation: 2 x ATmega32u4 (8-bit CPU, 16MHz clock speed, 2.5KB SRAM, 32KB flash storage)[19].

j) Arduino Nano

Processor: ATmega328, Architecture AVR, Operating Voltage 5 V, Flash

Memory32 KB of which 2 KB used by bootloader, SRAM2 KB Clock Speed16 MHz.

Features: Digital I/O Pins14 (of which 6 provide PWM output), Analog Input Pins 8.

From the types of Arduino boards, we are choosing Arduino Nano. Figure 2.23 show types of Arduino boards.

2.6.6 ARDUINO NANO

The Arduino Nano is a small, complete, and breadboard-friendly board based on the ATmega328 (Arduino Nano 3.0) or ATmega168 (Arduino Nano 2.x). It has more or less the same functionality of the Arduino Duemilanove, but in a different package. It lacks only a DC power jack, and works with a Mini-B USB cable instead of a standard one. The Nano was designed and is being produced by Gravitech. show figure 2.24.

Power: The Arduino Nano can be powered via the Mini-B USB connection, 6-20V unregulated external power supply (pin 30), or 5V regulated external power supply (pin 27). The power source is automatically selected to the highest voltage source.

The FTDI FT232RL chip on the Nano is only powered if the board is being powered over USB. As a result, when running on external (non-USB) power, the 3.3V output (which is supplied by the FTDI chip) is not available and the RX and TX LEDs will flicker if digital pins 0 or 1 are high.

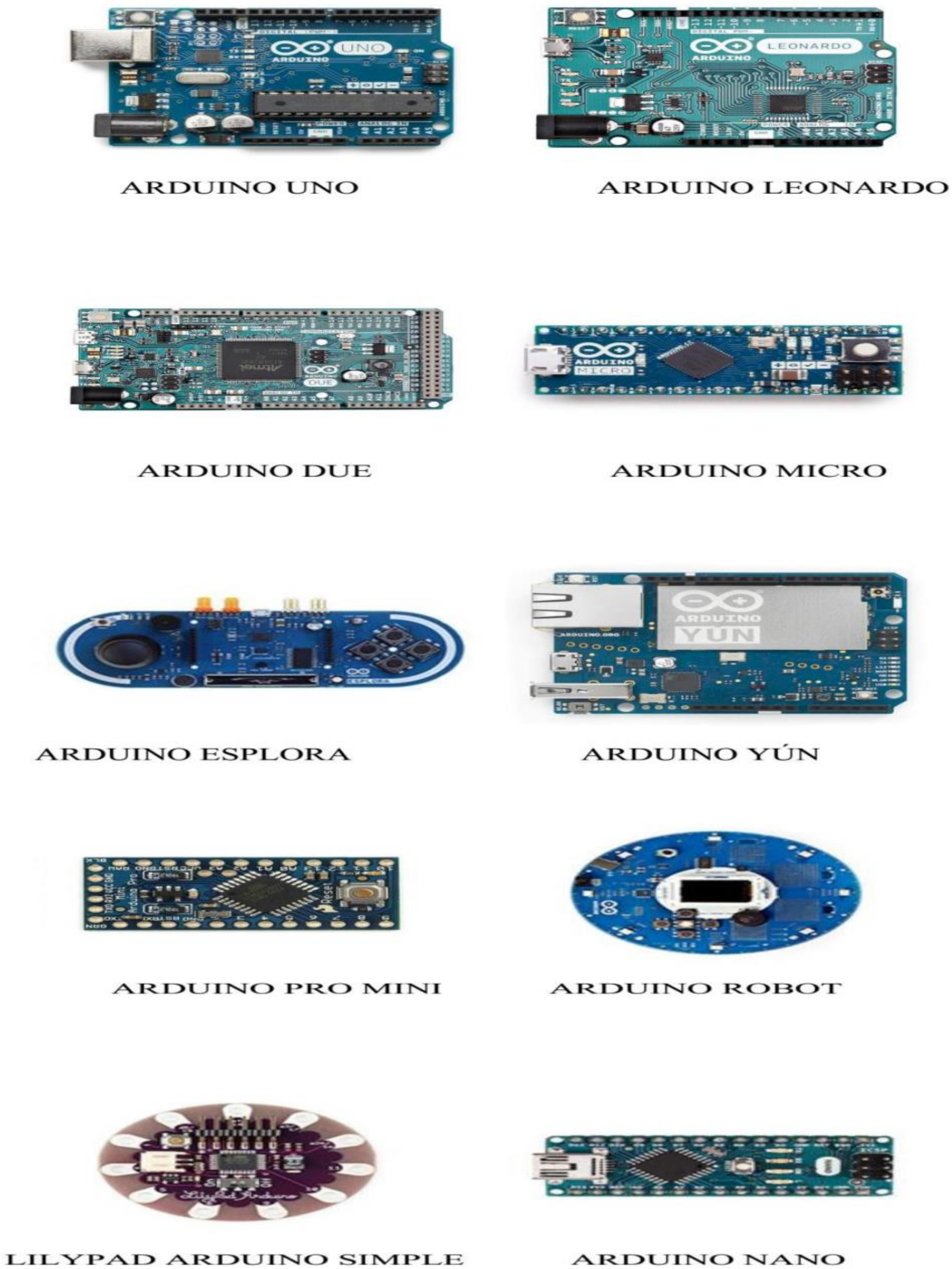

Figure2.23: Arduino Boards [19]

Memory: The ATmega168 has 16 KB of flash memory for storing code (of which 2 KB is used for the bootloader); the ATmega328 has 32 KB, (also with 2 KB used for the bootloader). The ATmega168 has 1 KB of SRAM and 512 bytes of EEPROM (which can be read and written with the EEPROM library); the ATmega328 has 2 KB of SRAM and 1 KB of EEPROM.

Input and Output: Each of the 14 digital pins on the Nano can be used as an input or output, using pin Mode (), digital Write (), and digital Read () functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 k Ohms.

Specifications:

Table 2.6: show Specifications for Arduino Nano

Microcontroller	Atmel ATmega168 or ATmega328
Operating Voltage (logic level)	5 V
Input Voltage (recommended)	7-12 V
Input Voltage (limits)	6-20 V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	8
DC Current per I/O Pin	40 mA
Flash Memory	16 KB (ATmega168) or 32 KB (ATmega328) of which 2 KB used by bootloader
SRAM	1 KB (ATmega168) or 2 KB (ATmega328)
EEPROM	512 bytes (ATmega168) or 1 K (ATmega328)
Clock Speed	16 MHz
Dimensions	0.73" x 1.70"

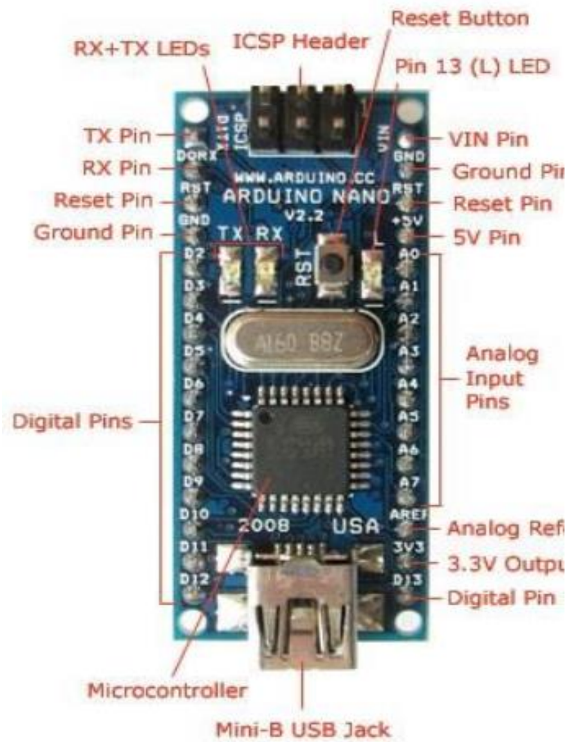

Figure 2.24 pins of Arduino Nano [20]

In addition, some pins have specialized functions:

Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the FTDI USB-to-TTL Serial chip.

External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attach Interrupt () function for details.

PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analog Write () function.

SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.

LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off. The Nano has 8 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the analogReference() function. Additionally, some pins have specialized functionality:

I2C: 4 (SDA) and 5 (SCL). Support I2C (TWI) communication using the Wire library (documentation on the Wiring website).

There are a couple of other pins on the board:

AREF. Reference voltage for the analog inputs. Used with analogReference().

Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Communication: The Arduino Nano has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega168 and ATmega328 provide UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An FTDI FT232RL on the board channels this serial communication over USB and the FTDI drivers (included with the Arduino software) provide a virtual com port to software on the computer. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the FTDI chip and USB connection to the computer (but not for serial communication on pins 0 and 1)[20].

CHAPTER THREE

SYSTEM DESIGN

3.1 Overview

The project consist of Arduino Nano , five flex sensor , accelerometer ,micro SD adapter, SD card , speaker and16X2 LCD module, when the Glove worn by the user which mounted by flex sensors , accelerometer ,Arduino Nano , the flex sensors give their output in the form of change in resistance according to the bend angle, the output from the flex sensors is connected to voltage divider circuit then connected to the analog input of Arduino Nano, the accelerometer gives output signals in terms of analog voltages that are proportional to acceleration , the output from the accelerometer is given to analog input of Arduino ,the gesture is defined from the output of flex sensors and accelerometer and fed to Arduino to recognition ,Arduino compare it with recorded data and when there is conformity with gesture then loaded the determine audio then run the voice by speaker and display the determined text intended by gesture .

The system hardware is shown by figure 3.1:

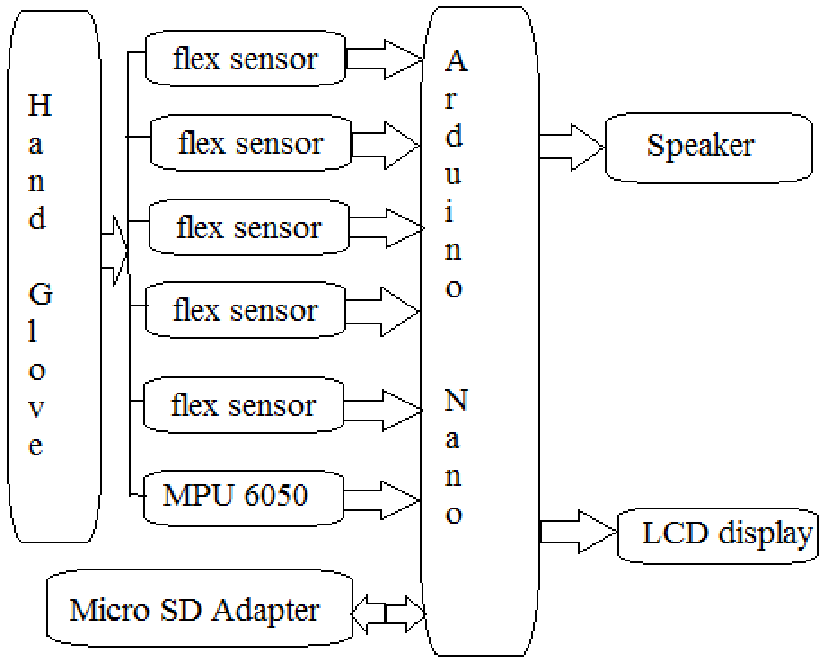

Figure 3.1: block diagram of the system

3.2 Flex Sensor Circuit

Using 5-flex sensor to recognized hand gestures these sensors are attached to the five fingers when we bend the finger the value of the sensor gets changed. flex sensors are similar to potentiometer, i.e. (variable resistor) the resistance of flex sensor change according to amount of its bending (from straight to bend) while bending the sensor the resistance of sensor will increase.

One end of flex sensor is connected to Analog Input (A0) of Arduino between A0 and +5V connected 47KΩ resistor, this connection means that the flex sensor and the 47KΩ resistor for a voltage divider. While the other end is connected to GND, show figure 3.2.

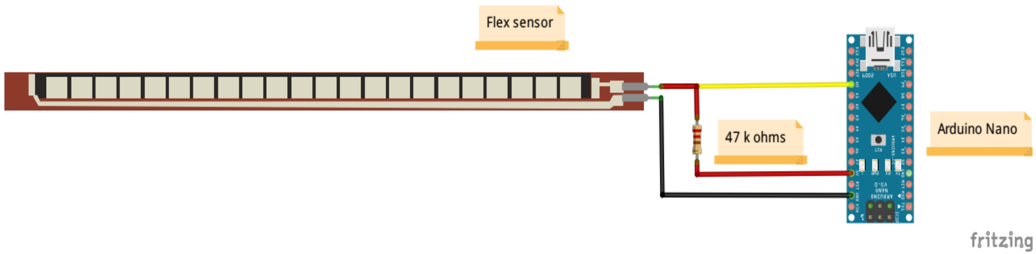

Figure 3.2: flex sensor with voltage divider

The basic concept of Flex Sensor is a variable resistor, it configured in a voltage divider fashion along by paired with resistors of a static value (47 K-ohm),and when paired each flex sensor with resistors of a static value(47 K-ohm) this means each change in resistance is the result of change in flex sensor resistor (express as bending the sensor from straight and bend each change in bend had equivalent resistor) show figure 3.3, and the change of flex sensor resistor can be measured by the Arduino through its analog inputs by sensed the change in voltage between the resistors (flex sensor and 47 k-ohm resistor) show figure 3.3.

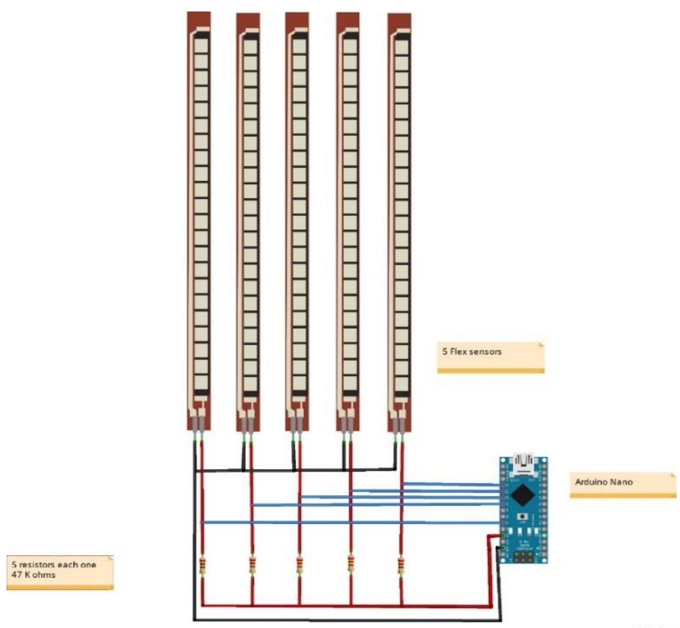

Figure 3.3: Flex sensors circuit

From figure 3.3 attached red wire (positive voltage) is +5V from the Arduino, black is (negative)GND wire, is connected to all the individual GND wires from the sensors, gets plugged into the Arduino's GND, and blue wire (send to Arduino the data of each flex sensor) gets plugged into an analog input pin to Arduino (A0, A1, A2, A3, A6).

Upon moving the flex sensor from straight to bend, it gives equivalent voltage values from 5 to 0. Now when we move the sensor from the straight (0 degree) the output from the sensor in the analog input of Arduino give 1023. and at bend (90 degree) the output from the sensor in the analog input of Arduino give 0 and map these values from 0 to 1023 to get equivalent angle from 0 to 90 to. When we move the move the sensor in the downward direction, then the output of the sensor will be 0.

3.3 MPU6050

The structure of the accelerometer is a mass attached to a spring which has fixed outer plates and moves along one direction. The capacitance between the plates will change whenever acceleration is applied. This change in capacitance will be measured and will correspond to an acceleration value.

Using MPU6050 as accelerometer to get X, Y and Z directions and connect the MPU 6050 sensor to the Arduino are showing by figure 3.4:

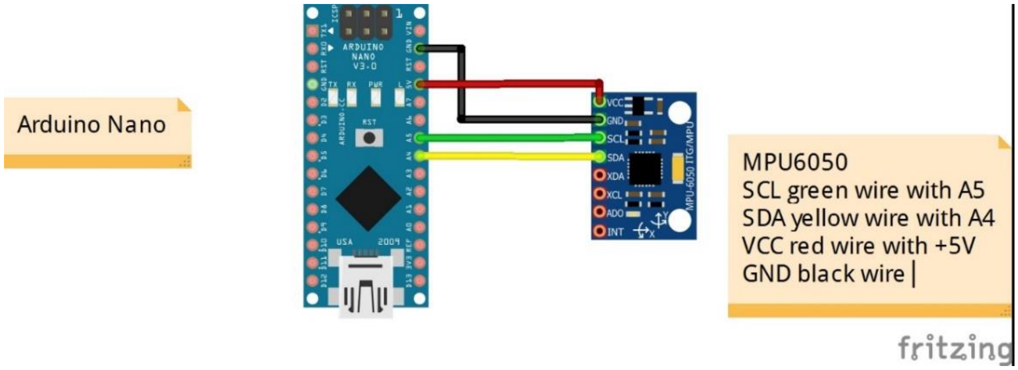

Figure3.4: MPU6050

VCC (red wire) on the MPU6050 connected to the 5V pin on the Arduino.

GND (black)on the MPU6050 connect to the GND on the Arduino.

SCL (green wire) on the MPU6050 connect to theA5 on the Arduino.

SDA (yellow wire) on the MPU6050 connect to the A4 on the Arduino.

Connected SCL and SDA on the MPU6050 to A5 and A4 on the Arduino because The A4 and A5 pins on the Arduino are for SPI communication.

Upon moving the Mpu6050 in the upward or downward direction, it gives values of accelerometer equivalent of direction. Then according to this value can determine the values reading from senor to upward and downward and put the values in range to accurate determine the value up or down.

3.4 Voice Unit

Component:

- SD Card Module
- SD Card (2GB).
- Speaker

The SD Card Module is used for transferring data to and from the memory card. and here using it to play audio files with your Arduino in decent quality from SD card (which audio file was stored in it to be run audio file is stored of form of wave file) show figure 3.5.

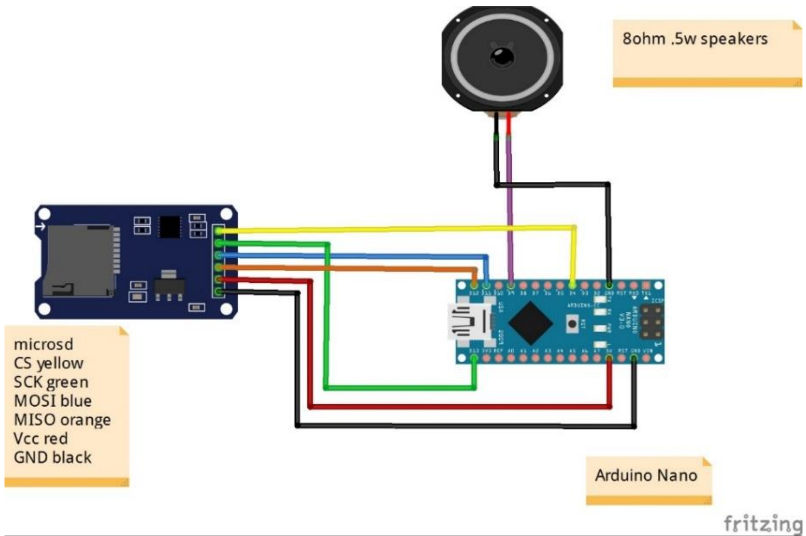

Figure 3.5: voice unit

VCC (red wire) to 5V of Arduino Nano.

GND (black wire) to GND of Arduino Nano.

CS (yellow wire) to pin 4 of Arduino Nano.

SCK (green wire) to pin 13 of Arduino Nano.

MOSI (blue wire) to pin 11 of Arduino Nano.

MISO (orange wire) to pin 12 of Arduino Nano.

5V (purple wire) of speaker to pin 9 of Arduino Nano.

GND of speaker (black wire) to GND of Arduino Nano.

The concept of play audio is by using Arduino to loads the .wav files from the micro-SD card. then generates a signal and outputs it through the speaker connected to digital pin 9. This allows the speaker to create sounds and play audio.

SD card:

Sd card is put into Micro SD Adapter and used to store audio files as form of .wav that wanted to be run, first must format it before used by used SD formatter program after download it and installed is showing by figure 3.6:

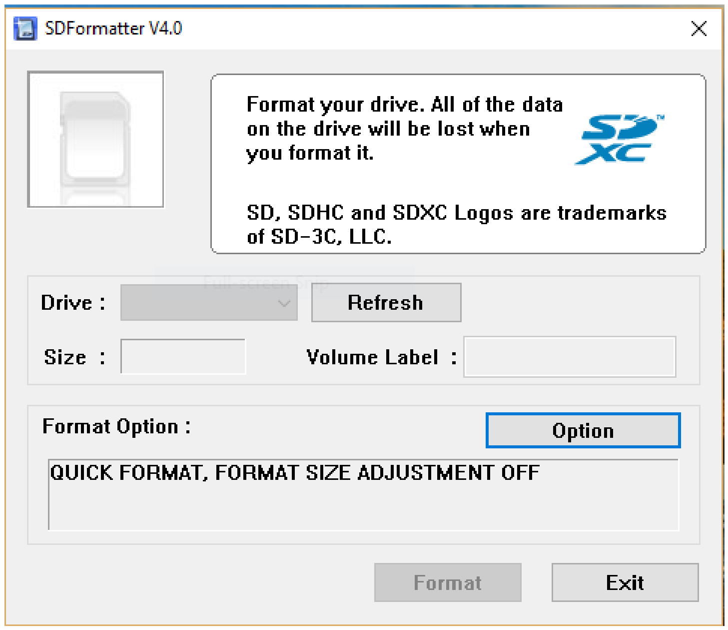

Figure 3.6: SD Formatter

Then after formatted it then down load into it audio files as form of .wav which had some specification to be compatible to Arduino to be run:

Bit Resolution 8 Bit

Sampling Rate 16000 Hz

Audio Channel Mono

PCM format PCM unsigned 8-bit

To get .wav audio using following step:

1.go to online music converter .

2. choose the position of music file and Upload.

3.In optional settings, change bit resolution to 8 bits.

4.Change sampling rate to 16000 Hz.

5.Change audio channels to Mono.

6.Click on "Show advanced options".

7.Set the PCM format as PCM unsigned 8-bit.

8.Click on "Convert", and the files are converted to .wav.

9. Then downloaded the audio file show figure 3.7:

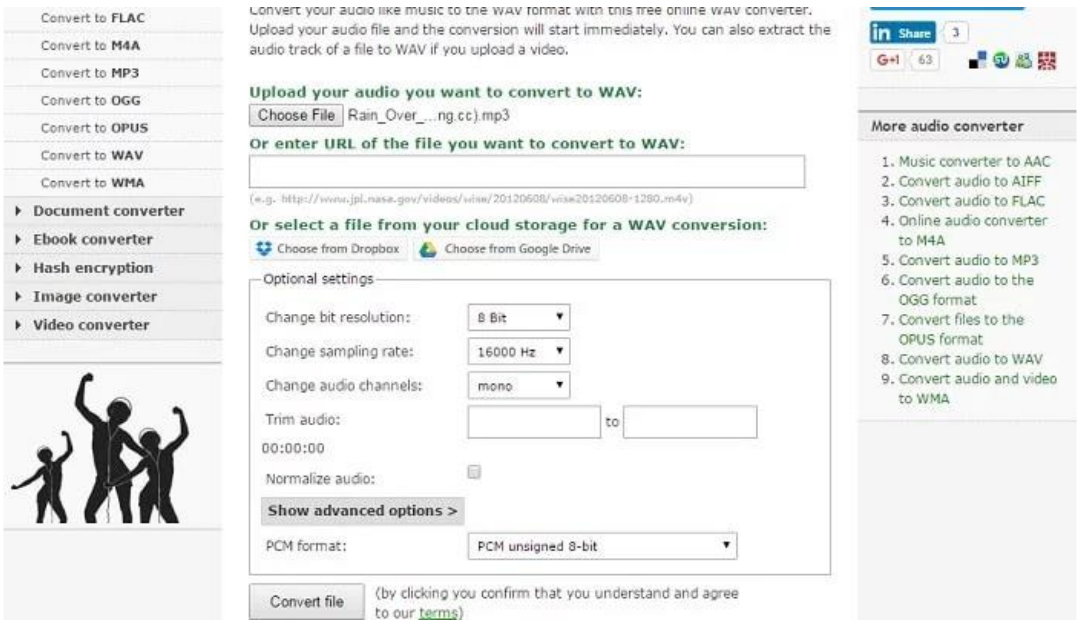

Figure3.7: step of convert audio file [21]

3.5 Complete design circuit

The Arduino Nano Capture the gesture by gets the value of five flex sensor and MPU6050 which attached to the back of glove, the values are compared with preserved gesture if it mismatches ignore it and capture next gesture, if it matched with preserved gesture then Arduino run the equivalent audio and display the equivalent text of gesture show figure 3.8.

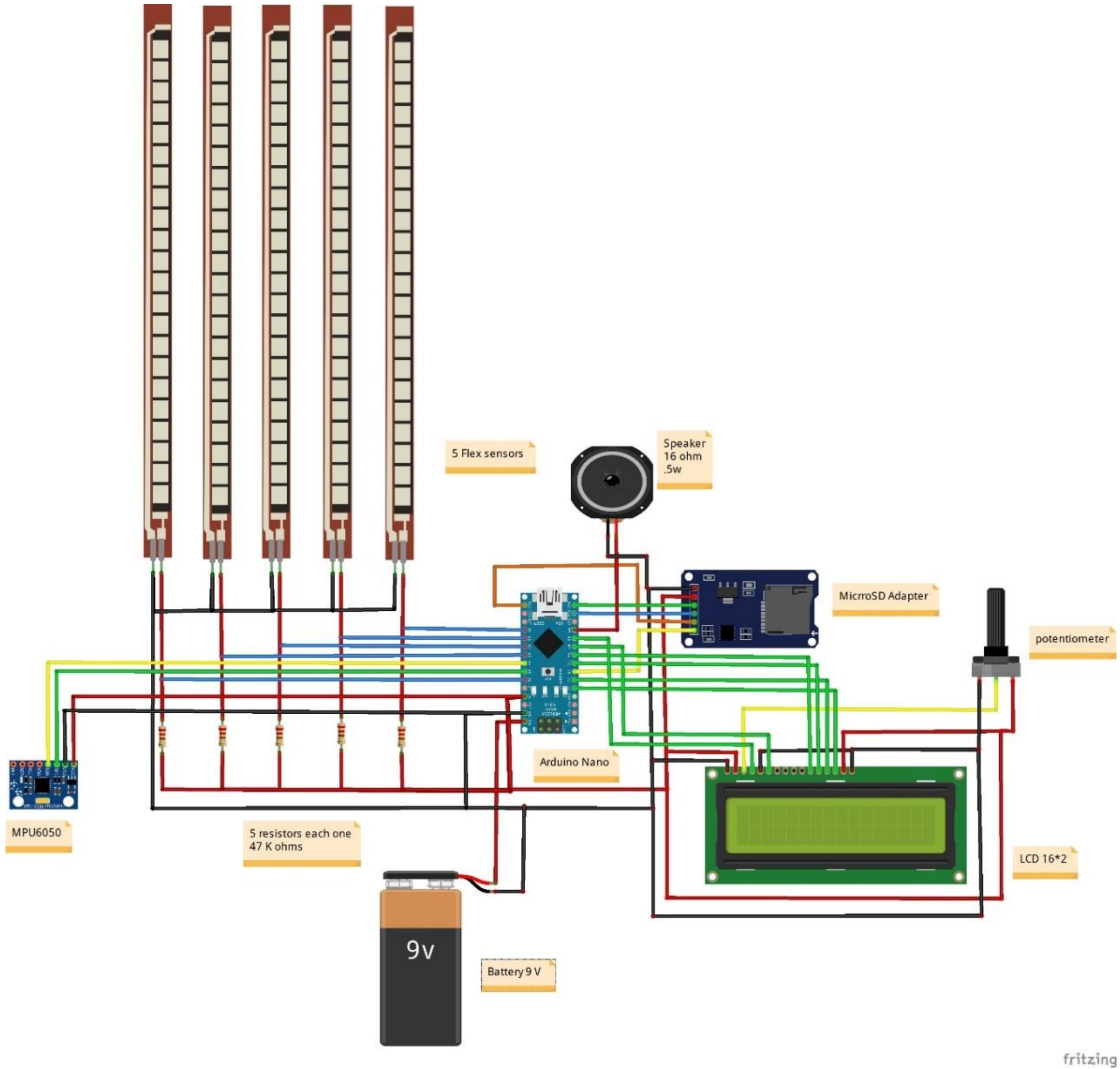

Figure 3.8: full design circuit

Using 9v battery to fed the Arduino connect the battery to Vin and ground to Arduino.

3.6 System Steps

Figure below describing the design of Gesture Sensing Glove.

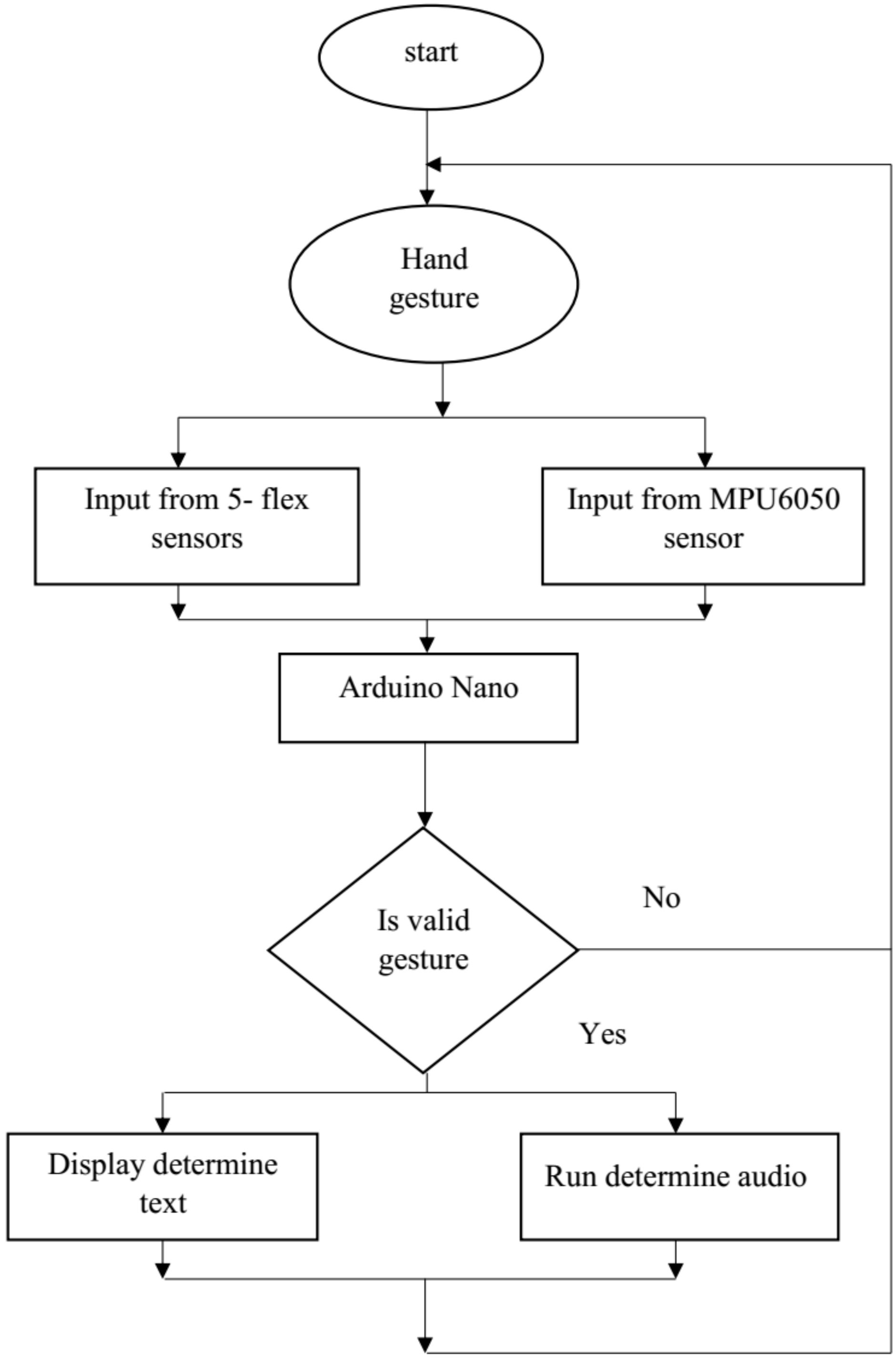

Figure 3.9: System Steps

CHAPTER FOUR

SYSTEM SIMULATION AND RESULT

4.1 System simulation

Flex sensors and MPU6050 are simulated as voltage divider, voice unit can't simulate it.

4.1.2 Flex Sensors Circuit

As say before flex sensor is like variable resistor to simulate it using flex sensor along with constant resistor (47 K Ohm) to form voltage divider then taken value of sensor between resistor and flex sensor to test the bending of sensor is it straight or at which angle degree (angle from 0 to 90 degree), then the output of five flex sensor to analog input of Arduino A0, A1, A2, A3, A6 and that represent the value of each five finger.

the value of fingers input from 0 to 1023 and it represent the bending of finger when the value it 1023 that mean it straight and the value decrease along with bending the finger until 90 degree to get value 0 ,by mapping this value 0 to 1023 to get the angle of finger from 0 to 90 degree, and by trier the value of each finger and find that each angle less than 38 degree consider it straight and each value greater than 40 degree consider it bend this consideration for all finger except thumb finger the consideration for it each value less than 20 degree consider it straight and each value greater than 25 consider it bend and using virtual terminal to display the value of sensor .the figure 4.1show simulation.

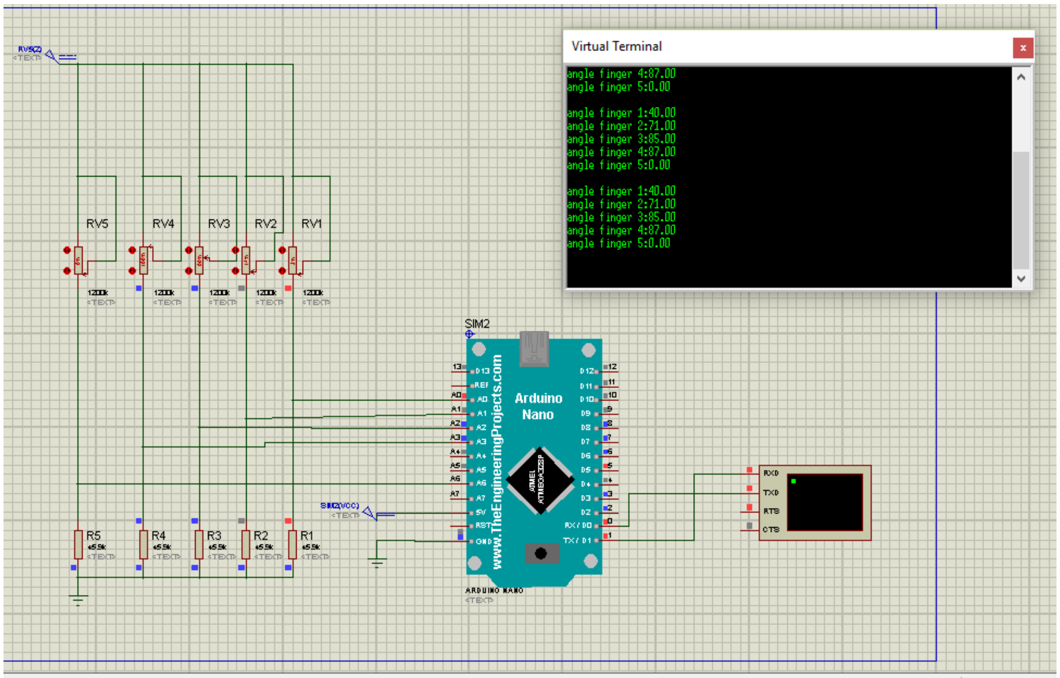

Figure 4.1: show the five-flex sensor simulation

4.1.2 MPU60 50 circuit

Using MPU6050 sensor to gets the value of acceleration (direction of hand), and MPU 6050 gets this value of form of voltage for three variables x, y, z, to gets this value connecting MPU 6050 to analog input of Arduino in exactly to A4, A5, and then gets the value from sensor that represent the three axis direction, and then here need to determine the direction of hand its up or at side direction, and according to the direction of hand save all variables that represent each direction to use them as test hand direction.

For the up direction of hand finding that the value of x direction is all value that <=0.39, the value of y direction is all value that < -0.80. show figure4.2 below:

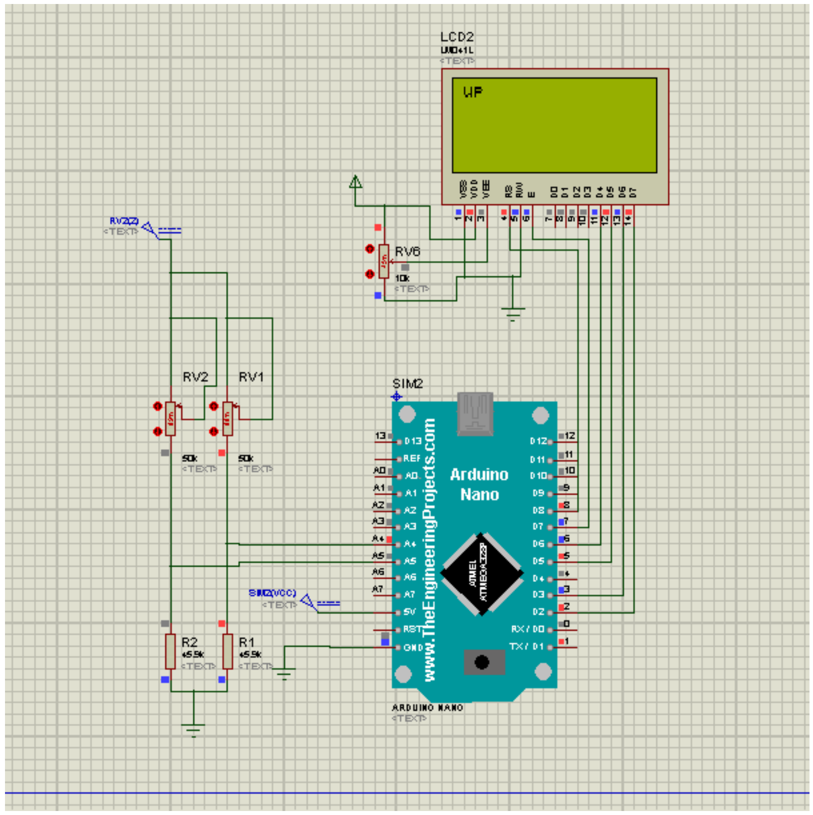

Figure4.2: the up direction of hand

For the side direction of hand finding that the value of x direction is all value that >0.49, the value of y direction is all value that > -0.70. show figure 4.3.

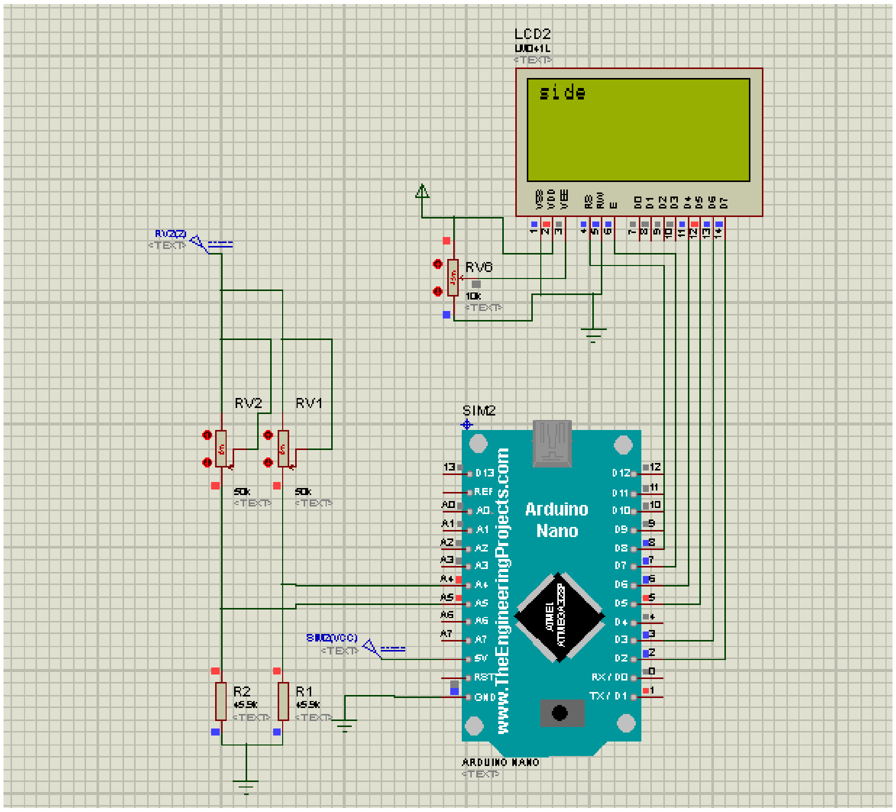

Figure4.3: the side direction of hand

4.1.3 Hand gesture circuit

The circuit here is include five flex sensors and MPU6050 circuit, using flex sensor to determine the amount of bending of finger and MPU6050 sensor to determine the direction of hand.as example using some gesture represented show figure 4.4.

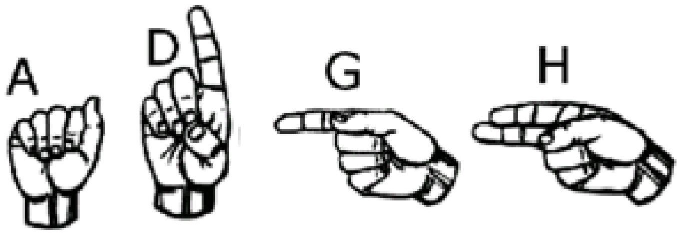

Figure 4.4: gesture "A","D","G","H" [8]

Figure 4.5 show the simulation for gesture "A"(show figure 4.4) which express as all finger is bending with 90 degree except thumb finger is straight with 0 degree and the direction of the hand is up.

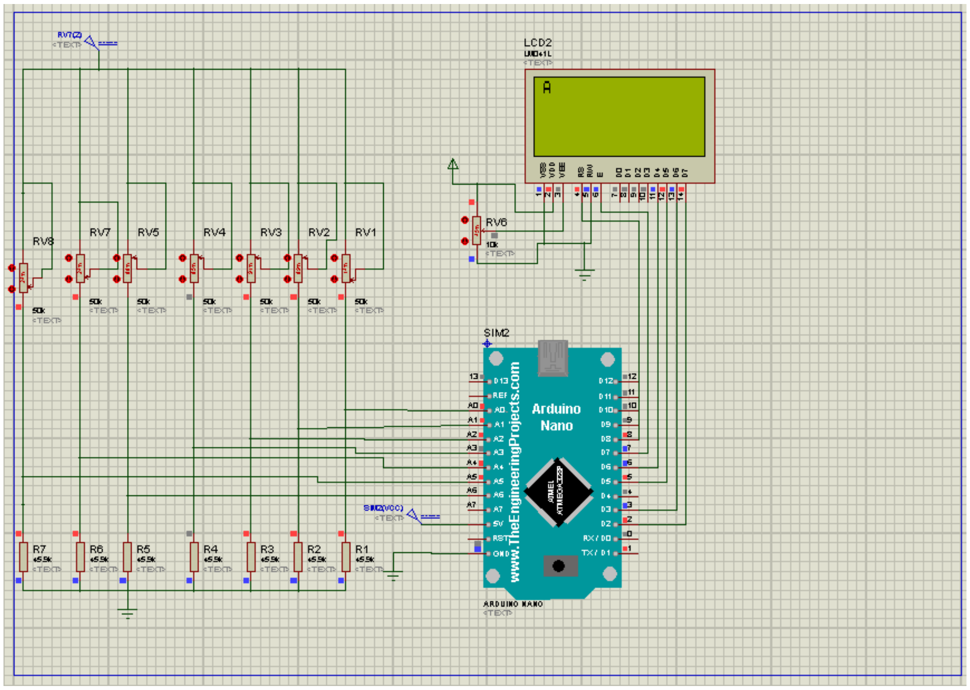

Figure 4.5: gesture "A"

Figure 4.6 show the simulation for gesture "D"(show figure 4.4) which express as all finger is bending with 90 degree except index finger is straight with 0 degree and the direction of the hand is up.

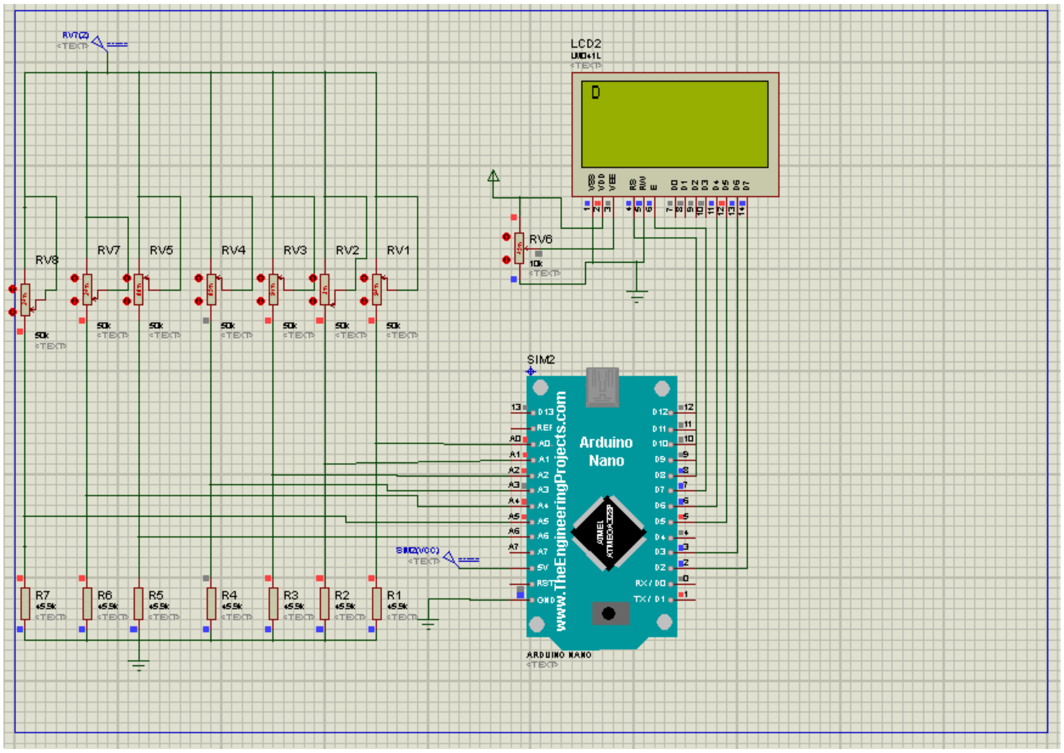

Figure 4.6: gesture "D"

Figure4.6 show the simulation for gesture "G"(show figure4.4) which express as all finger are bend with 90 degree except thumb and index fingers are straight with 0 degree and the direction of the hand is side.

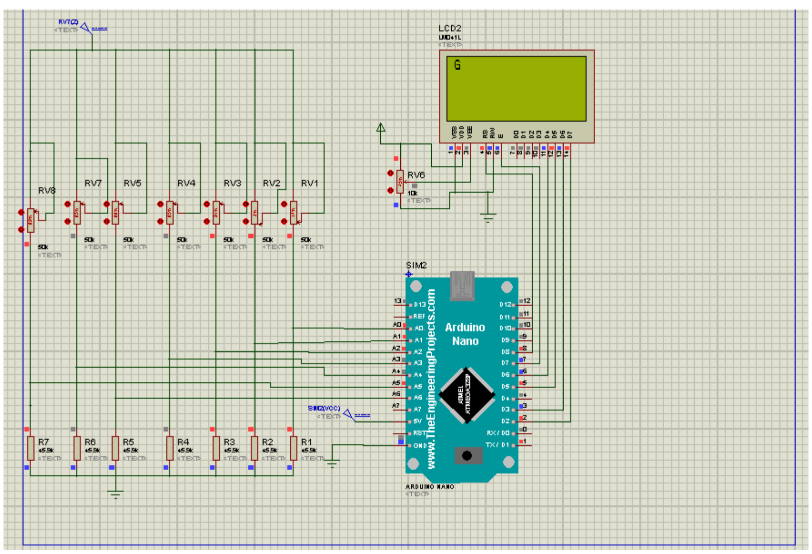

Figure 4.7: gesture "G"

Figure4.8 show the simulation for gesture "H"(show figure4.4) which express as all finger is straight with 0 degree except ring and pinky fingers are bend with 90 degree and the direction of the hand is side.

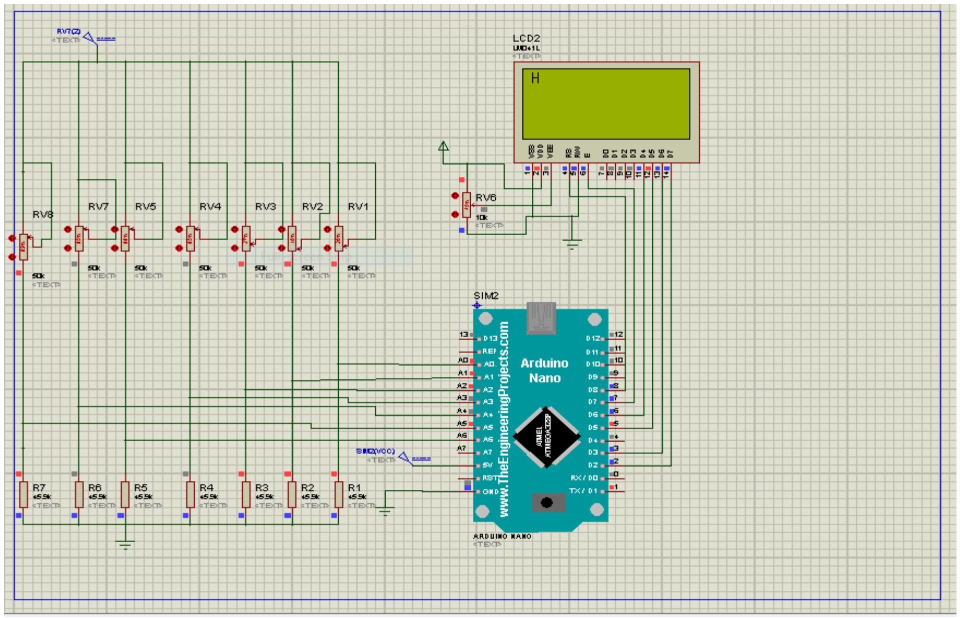

Figure4.8: gesture "H"

4.2 Hardware Implementation

The way the circuit works start form wearing the glove show figure 4.9 (5-flex sensors and MPU6050 attached to it) which will determine which gesture is chooses, after pick up gesture it will be compared with preserved gestures is it mismatch with any gesture just ignore it and next gesture will pick up is it matches with preserved gesture then the equivalent preserved voice will run at speaker after loaded from Micro SD Adapter (included Micro SD) which voices was stored as form as .wave file and text will displayed at LCD show figure 4.9 .

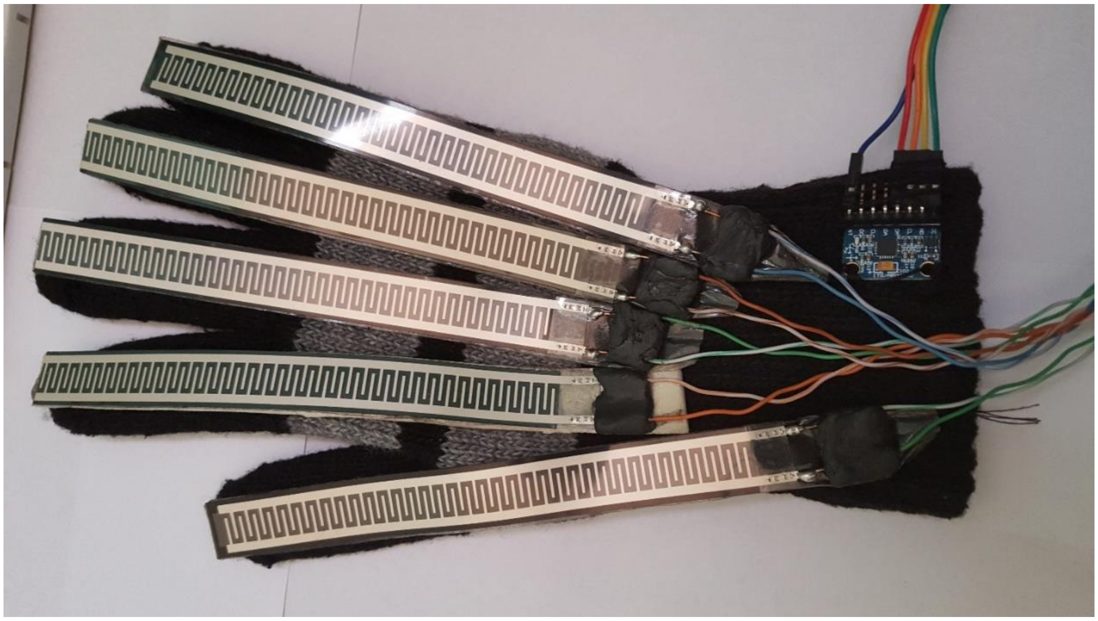

Figure4.9: 5-flex sensors and MPU6050 sensor

Figure 4.9 show the 5- flex sensors and MPU6050 sensor attached to the back of the get value of 5-finger degree and direction of hand.

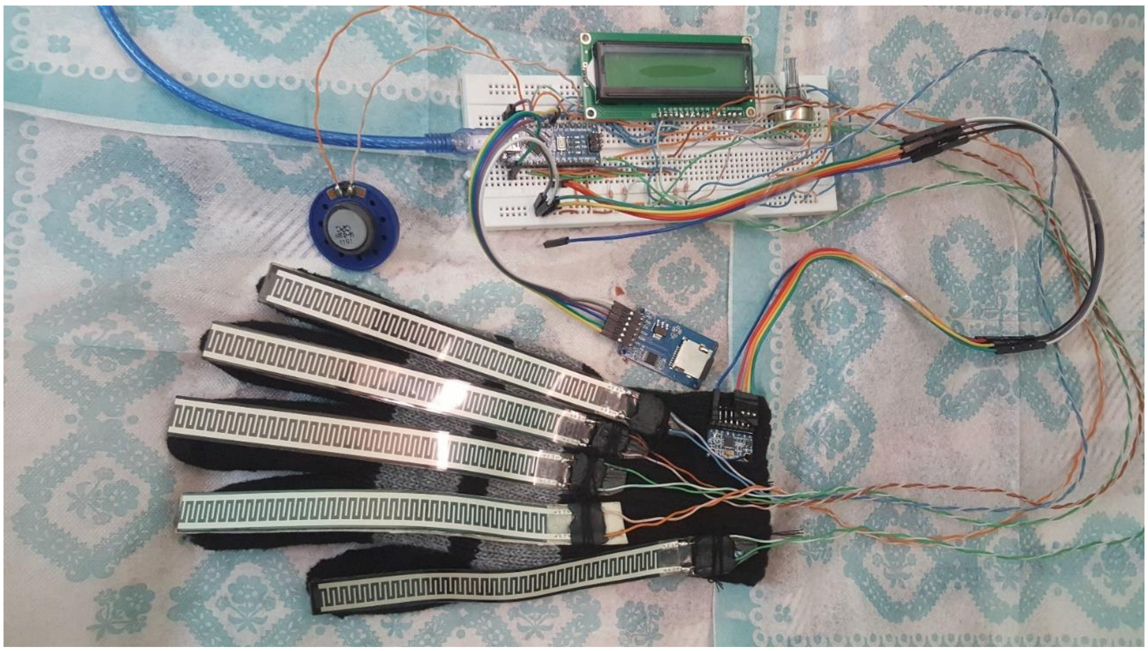

Figure 4.10: complete design circuit implementation

Figure 4.10 the complete design circuit implemented of smart glove which include 5-flex sensor, MPU6050 sensor, speaker, Micro SD Adapter, LCD and Arduino Nano.

For hardware implementation test using following gesture "A", "B", "D", "K", "H" show figure 4.13:

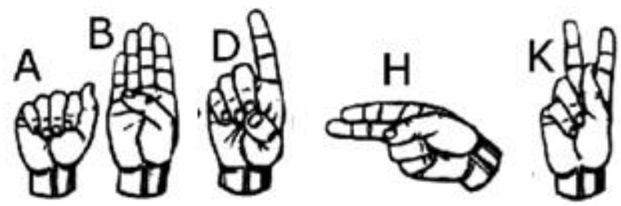

Figure 4.13: show gesture of hardware implantation test

user can use feature desired gestures and then recorded voice can be played and preserved text displayed simultaneously.

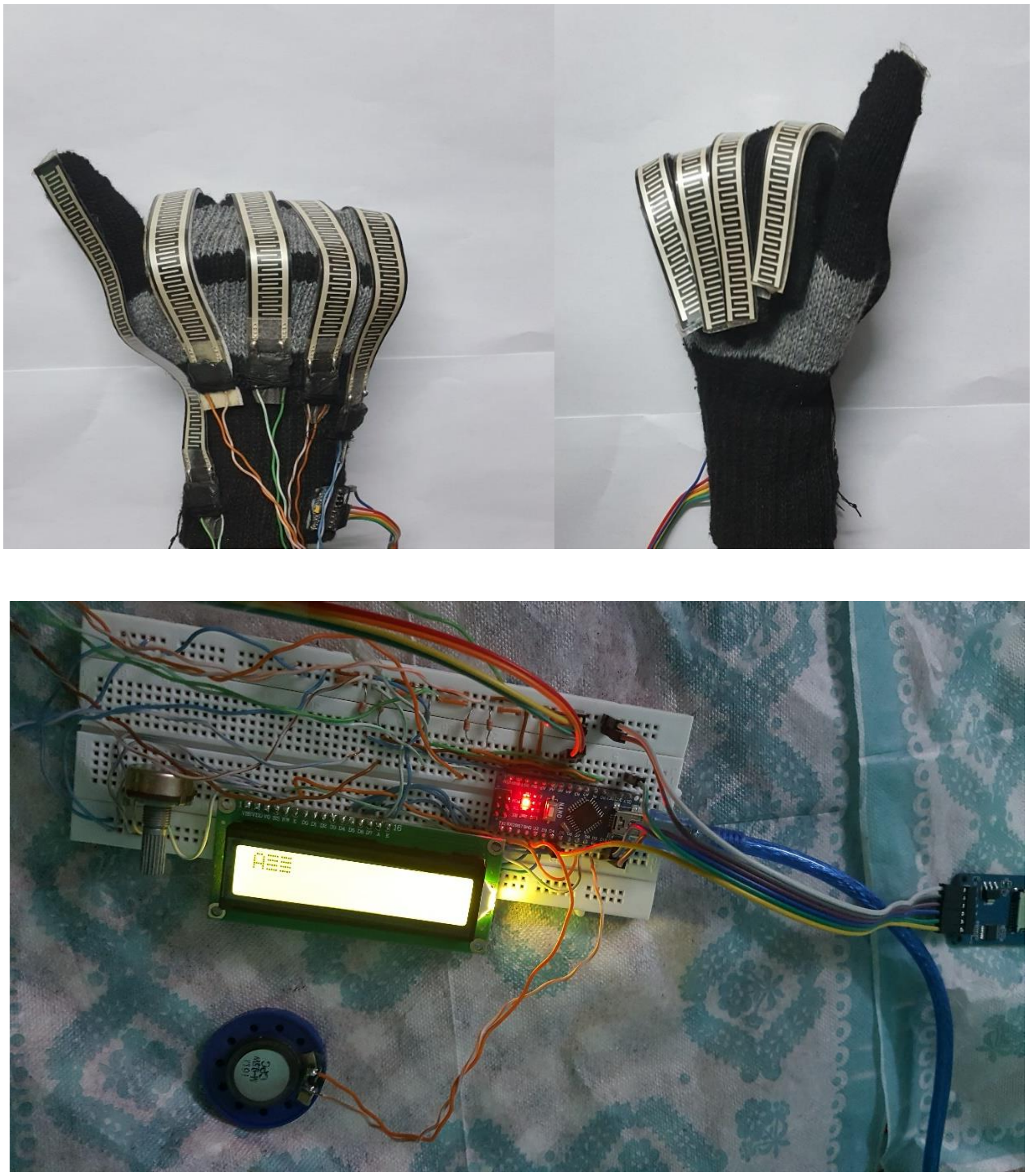

Figure 4.14: gesture "A" of hardware implantation test

In figure 4.14 when the user chooses desired gesture in this case gesture 'A' the equivalent text displayed and at same time equivalent audio is run.

Which express as all finger is bending with 90 degree except index finger is straight with 0 degree and the direction of the hand is up.

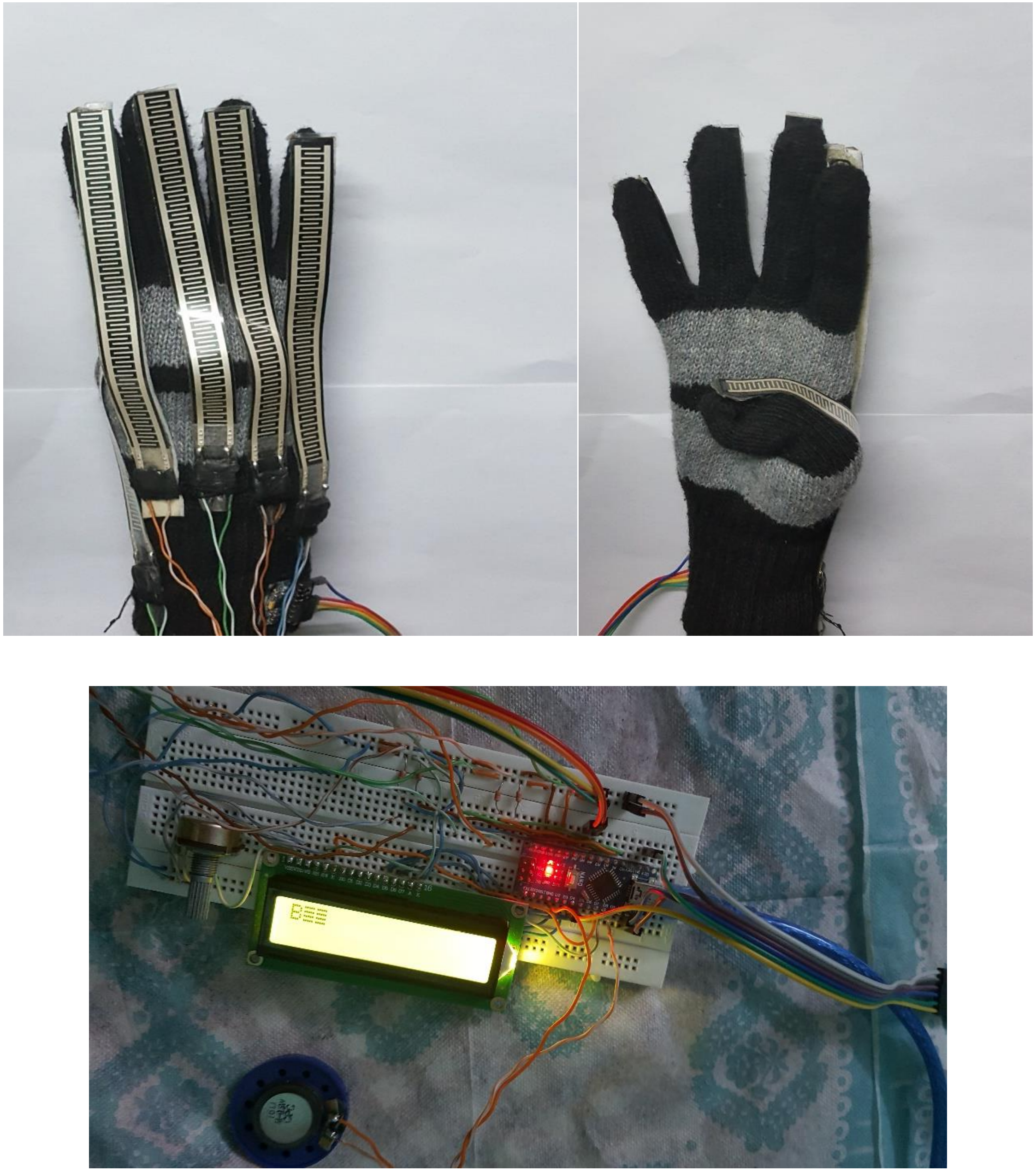

Figure 4.15: gesture "B" of hardware implantation test

In figure 4.14 when the user chooses desired gesture in this case gesture 'B' the equivalent text displayed and at same time equivalent audio is run.

Which express as all finger are straight with 0 degree except index finger is bend with 90 degree and the direction of the hand direction is up.

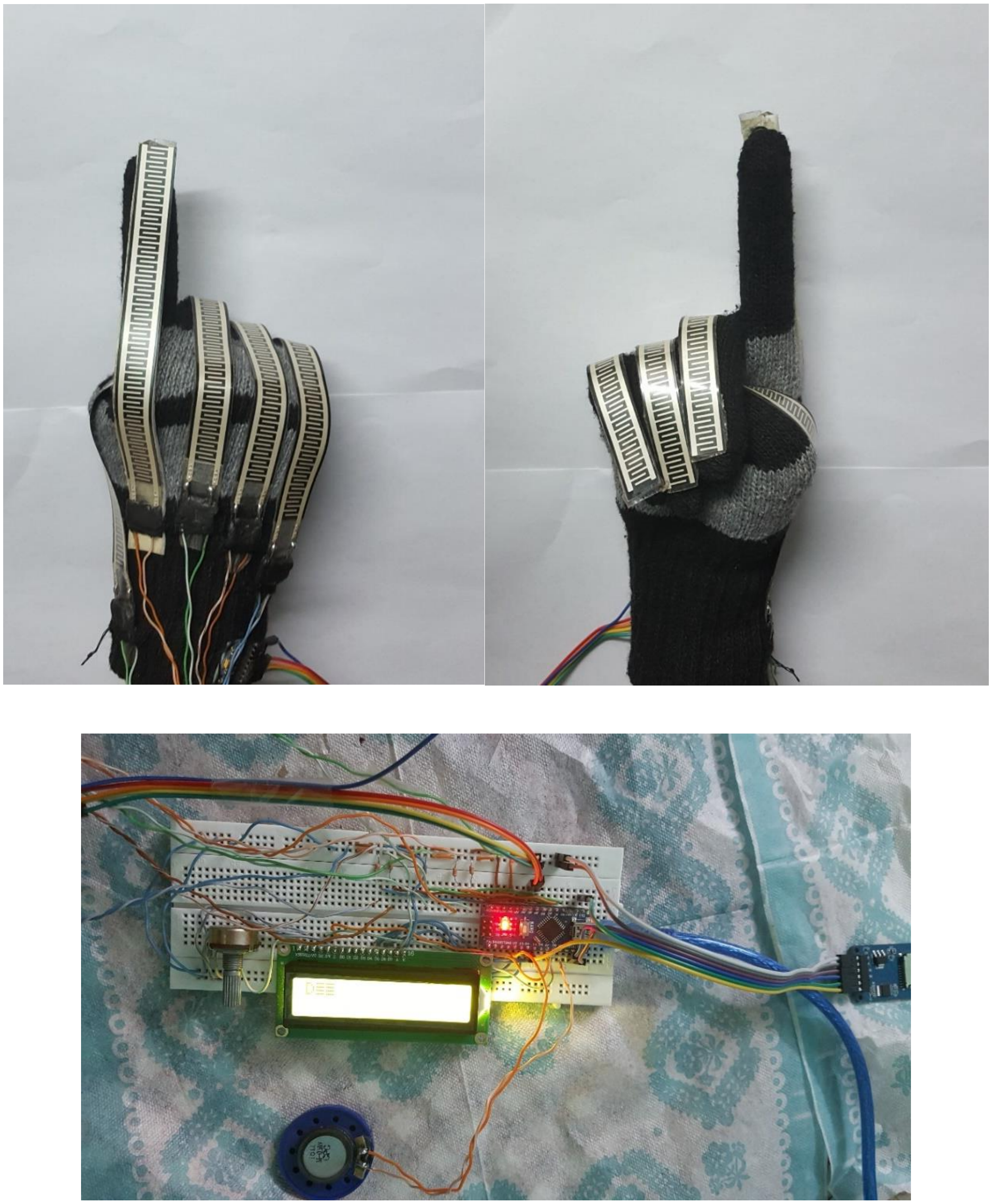

Figure 4.16: gesture "D" of hardware implantation test

In figure 4.14 when the user chooses desired gesture in this case gesture 'D' the equivalent text displayed and at same time equivalent audio is run.

Which express as all finger are bend with 90 degree except thumb and index fingers are straight with 0 degree and the direction of the hand direction is side

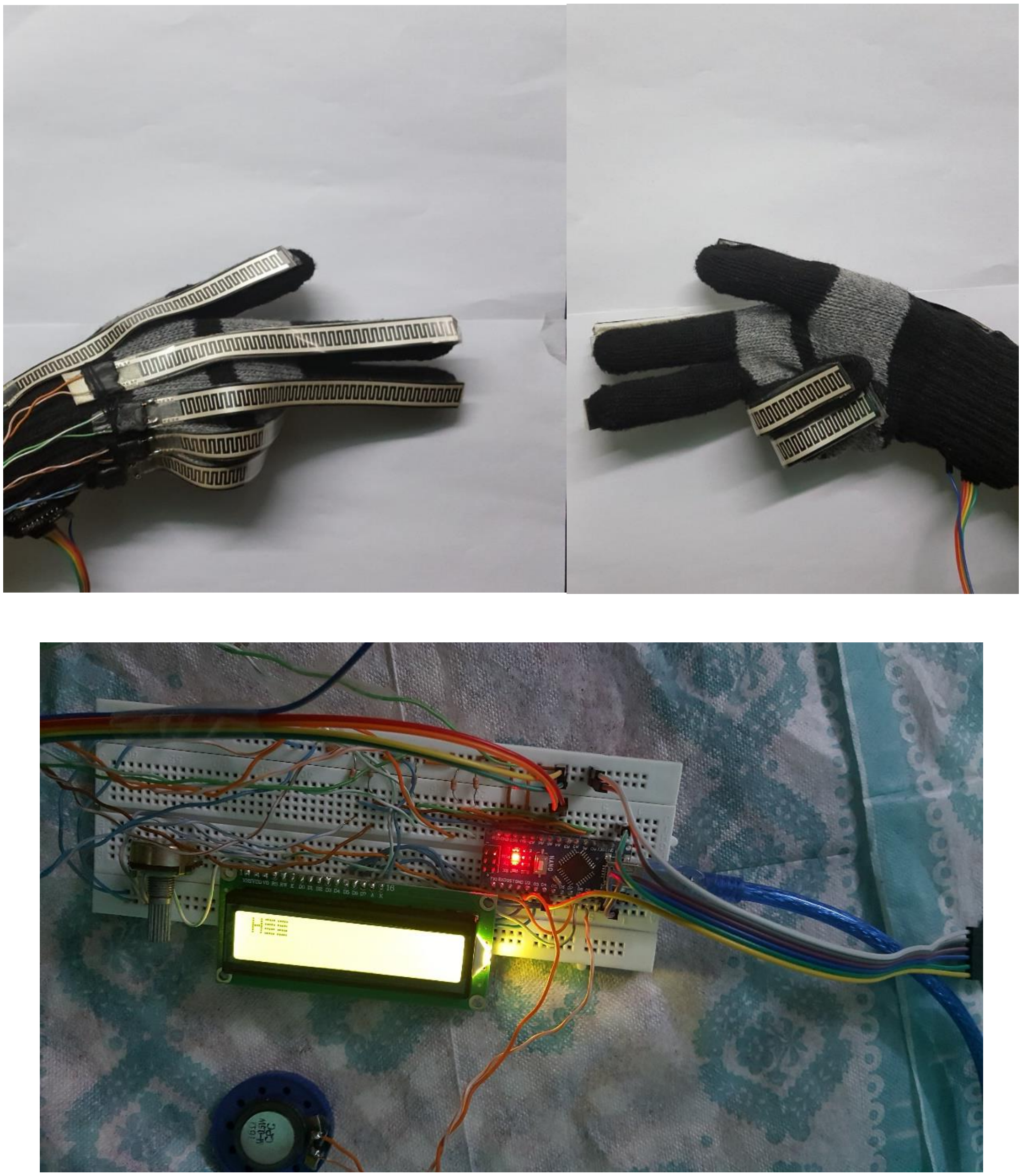

Figure 4.17: gesture "H" of hardware implantation test

In figure 4.17 when the user chooses desired gesture in this case gesture 'H' the equivalent text displayed and at same time equivalent audio is run.

Which express as all finger is straight with 0 degree except ring and pinky fingers are bend with 90 degree and the direction of the hand is side.

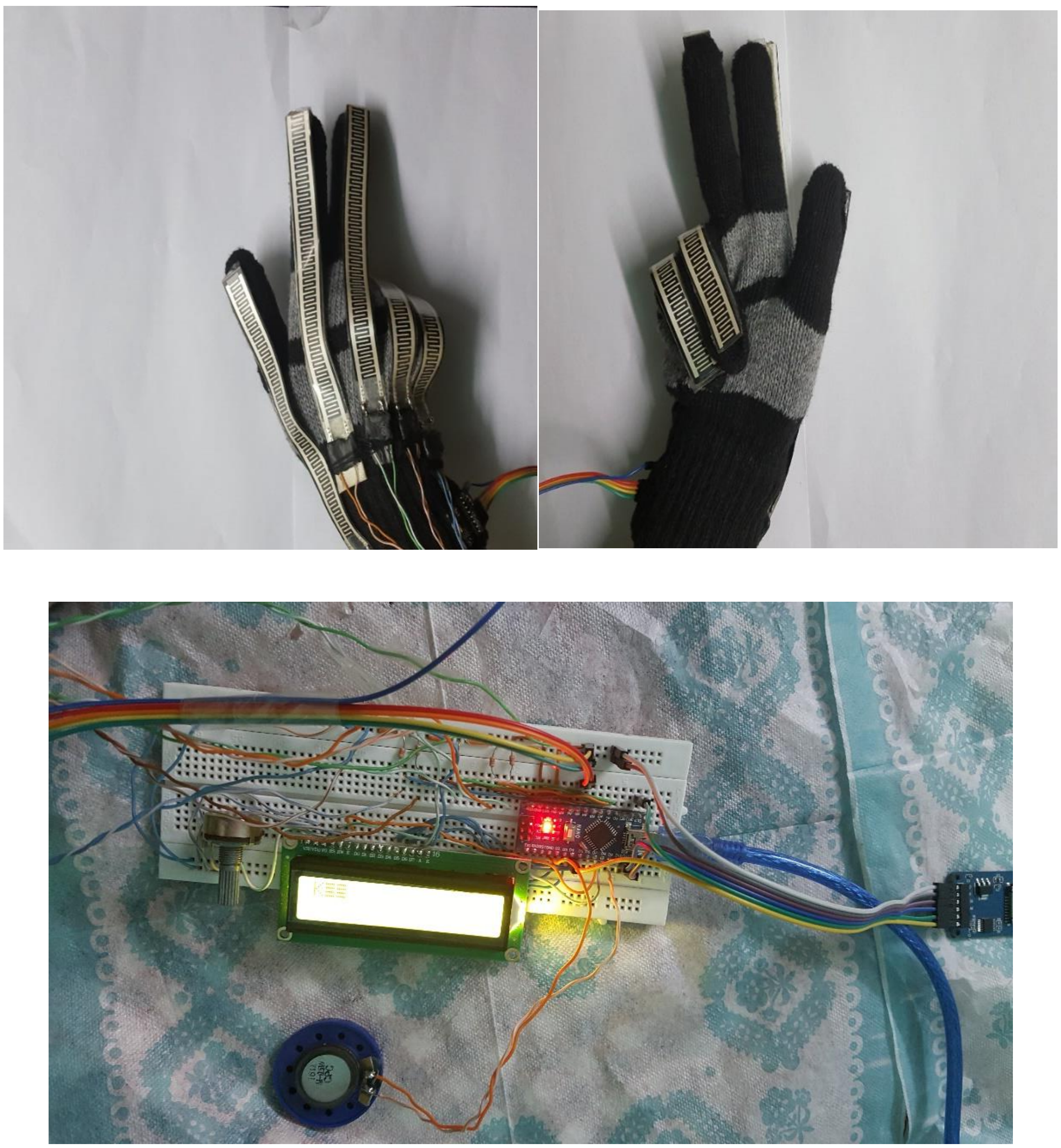

Figure 4.18: gesture "K" of hardware implantation test

In figure 4.18 when the user chooses desired gesture in this case gesture 'K' the equivalent text displayed and at same time equivalent audio is run.

Which express as all finger is straight with 0 degree except ring and pinky fingers are bend with 90 degree and the direction of the hand direction is up. notice that the voice is run while the gesture is displayed in LCD

4.3 Result

The person who wears the glove must wait for 3 seconds in order to initialization the MPU6050 sensor, then gesture can be detected, the gesture is consist of direction of hand and bending degree of five finger , which gets for gesture value by detect the hand and bending degree of five finger by MPU6050 and 5-flex sensor respectively, the value of these sensor is fed to ADC of Arduino Nano, with each variable of value of sensors either bending the finger or change the hand direction the sensors give values for each changes, the gesture then be define as specific values determined for each sensor.
When determine gesture was pick up it will be translation into text at LCD and speech at speaker.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

Disable people using sign language (gesture) to communicate with others, but normal person cannot understand what disable people try to say by their sign language and needed to translation to be understood and to communicate with them, to make a communication bridge glove was designed for deaf and dumb people. The glove is capable of translating their sign language gestures into speech and text, so, that communication is not limited between disable people only they can communicate with normal people, and make their future better.

Then as the result the glove is capable to translate sign language into text and speech, but there is some limitation more gesture needing to be added by increasing more sensor, and needing to differentiate some gesture from others as there is some similarity of them and all these problems can solve by added more sensors to make the gesture more precisely.

5.2 Recommendations

To improve system performance in future development, some suggestion below.

- Using two glove and connected them wirelessly in order to increase the number of gestures.

- Make glove depend also on motion of hand in order to increase number of gesture and at sometimes gets precisely some gesture.
- To precisely increase recognition of finger gesture (as different between gesture "V" and "U"), more sensors need to be added to detect gab between finger.
- To increase recognition of gesture, more sensors need to be added to detect the other part of the body such as arm, elbow, shoulder.
- Using an application to translate gesture into text and speech.

References:

[1] P. Verma, S. L. Shimi, and S. Chatterji, “Design of Smart Gloves,” vol. 3, no. 11, pp. 210–215, 2014.

[2] A. Manware, R. Raj, A. Kumar, and T. Pawar, “SMART GLOVES AS A COMMUNICATION TOOL FOR THE SPEECH IMPAIRED AND HEARING,” vol. 4, no. 06, pp. 78–82, 2017.

[3] K. Rastogi, “A Review Paper on Smart Glove - Converts Gestures into Speech and Text,” no. May, pp. 92–94, 2016.

[4] P. A. P. Bagade, “SMART HAND GLOVES FOR DISABLE PEOPLE,” pp. 1423–1426, 2018.

[5] K. V Fale, A. Phalke, P. Chaudhari, and P. Jadhav, “Smart Glove : Gesture Vocalizer for Deaf and,” pp. 6800–6806, 2016.

[6] A. Mishra and S. Malhotra, “Design of Hand Glove for Wireless Gesture Control of Robot,” vol. 114, no. 8, pp. 69–79, 2017.

[7] A. Soni, A. Kardak, and K. Gokul, “Missing sense a smart glove for sign to speech / text conversion,” vol. 3, no. 7, pp. 327–329, 2016.

[8] N. Kawale, P. Kaspate, H. Vanjari, and P. P. Sarode, “Implementation Paper on Sign Language,” pp. 2623–2629, 2018.

[9] T. Chouhan, A. Panse, A. K. Voona, and S. M. Sameer, “Smart Glove With Gesture Recognition Ability For The Hearing And Speech Impaired,” no. September 2014, 2015.

[10] A. Sreejan and Y. S. Narayan, “A Review on Applications of Flex Sensors,” no. August, 2017.

[11] G. Saggio, F. Riillo, L. Sbernini, and L. R. Quitadamo, “Resistive flex sensors : A survey Resistive flex sensors : a survey,” *Smart Mater. Struct.*, vol. 25, no. 1, p. 13001, 2016.

[12] A. Hjort, “Department of Physics and Astronomy Measuring mechanical vibrations using an Arduino as a slave I / O to an EPICS control system as a slave I / O to an EPICS control system,” 2015.

[13] “Design of Angle Detection System Based on MPU6050 Jian Huang,” vol. 73, no. Icemc, pp. 7–9, 2017.

[14] S. P. Ambildhok and N. B. Hulle, “Development of Inertial Navigation System based on Accelerometer and Gyroscope,” vol. 7, no. 5, pp. 12149–12151, 2017.

[15] Data Sheet, “Micro SD Card Micro SDHC Mini TF Card Adapter Reader Module for Arduino,” 2019.

[16] B. M. data sheet and I. Date, “SPECIFICATION Approval :,” no. 199, 2019.

[17] A. Harshika, “Real Time Based Temperature Control Using Arduino,” vol. 8, no. 2, pp. 209–216, 2017.

[18] R. Khadse, N. Gawai, and B. M. Faruk, “Overview and Comparative Study of Different Microcontrollers,” vol. 2, no. Xii, pp. 311–315, 2014.

[19] C. Rajan, B. Megala, A. Nandhini, and C. R. Priya, “A Review : Comparative Analysis of Arduino Micro Controllers in Robotic Car,” no. January, 2015.

[20] Data Sheet, “Arduino Nano,” vol. 328, pp. 3–5, 2019.

[21] Http://audio.online-convert.com/convert-to-wav, “No Title.”

Appendix A

```
#include <pcmConfig.h>
#include <pcmRF.h>
#include <TMRpcm.h>
#include <SPI.h>
#include <SD.h>
#include <MPU6050_tockn.h>
#include <Wire.h>
 #include <LiquidCrystal.h>
 #define SD_ChipSelectPin 4 //Chip select is pin number 4
  LiquidCrystal lcd(8, 7, 6, 5, 3, 2);
MPU6050 mpu6050(Wire);
  //Constants:
//Variables:
//const int flexPin0 = A0; //pin A0 to read analog input
int flexSensor0; //save analog value
int flexSensor1;
int flexSensor2;
int flexSensor3;
int flexSensor4;
void setup(){
 pinMode(A0,INPUT);
  pinMode(A1,INPUT);
  pinMode(A2,INPUT);
   pinMode(A3,INPUT);
   pinMode(A6,INPUT);
```

```
  Serial.begin(9600);       //Begin serial communication
  Wire.begin();
   mpu6050.begin();
  mpu6050.calcGyroOffsets(true);
  lcd.begin(16, 2);
if (!SD.begin(SD_ChipSelectPin)) {
Serial.println("SD fail");
return;}}
void loop(){
  lcd.noCursor();
   lcd.clear();
  lcd.display();
lcd.home();
TMRpcm music;
music.setVolume(5);    //   0 to 7. Set volume level
music.quality(1);
music.speakerPin = 9; //Auido out on pin 9
  mpu6050.update();
 flexSensor0 = analogRead(A0);             //Read and save analog value from
potentiometer
  Serial.println(flexSensor0);              //Print value
delay (500);
 flexSensor1 = analogRead(A1);             //Read and save analog value from
potentiometer
  Serial.println(flexSensor1);              //Print value
delay (500);
```

```
flexSensor2 = analogRead(A2);           //Read and save analog value from
potentiometer
  Serial.println(flexSensor2);            //Print value
delay (500);
flexSensor3 = analogRead(A3);           //Read and save analog value from
potentiometer
 Serial.println(flexSensor2);            //Print value
delay (500);
flexSensor4 = analogRead(A6);           //Read and save analog value from
potentiometer
  Serial.println(flexSensor4);            //Print value
delay (500);
float angle = map(flexSensor0, 0.0, 1023.0, 90.0, 0.0);
float angle1 = map(flexSensor1, 0.0, 1023.0, 90.0, 0.0);
float angle2 = map(flexSensor2, 0.0, 1023.0, 90.0, 0.0);
Serial.println(angle2);

float angle3 = map(flexSensor3, 0.0, 1023.0, 90.0, 0.0);
float angle4 = map(flexSensor4, 0.0, 1023.0, 90.0, 0.0);
   mpu6050.update();
Serial.println();
if(angle<=20 &&angle1<=38 &&angle2<=38 &&angle3>=40 &&angle4>=40
&& mpu6050.getAccX()<=0.39 && mpu6050.getAccY()< -0.80 ){
Serial.println("k");
music.play("k.wav");
lcd.println("K");
delay(3000);}
```

```
if(angle<=20 &&angle1<=38 &&angle2<=38 &&angle3>=40 &&angle4>=40
&&mpu6050.getAccX()>0.49 && mpu6050.getAccY() > -0.70 ){
Serial.println("H");
music.play("h.wav");
lcd.println("H");
delay(3000);}
if(angle>=25 &&angle1<=38 &&angle2>=40 &&angle3>=40
&&angle4>=40&& mpu6050.getAccX()<=0.39 && mpu6050.getAccY()< -
0.80 ){
Serial.println("D");
music.play("d.wav");
lcd.println("D");
delay(3000);}
if(angle<=20 &&angle1<=38 &&angle2>=40 &&angle3>=40 &&angle4>=40
&&mpu6050.getAccX()>0.49 && mpu6050.getAccY()> -0.70 ){
Serial.println("G");
music.play("g.wav");
lcd.println("G");
delay(3000);}
else if(angle>=25 &&angle1<=38 &&angle2<=38&&angle3<=38
&&angle4<=38 && mpu6050.getAccX()<=0.39 && mpu6050.getAccY()< -
0.80 ){
Serial.println("B");
music.play("b.wav");
lcd.println("B");
delay(3000);}
```

```
else if(angle<20 &&angle1>=40 &&angle2>=40 &&angle3>=40 &&angle4>=40 && mpu6050.getAccX()<=0.39 && mpu6050.getAccY()< -0.80  ){
Serial.println("A");
music.play("a.wav");
lcd.println("A");
delay(3000);}
//else if(angle>=40 &&angle1<=38 &&angle2<=38 &&angle3>=40 &&angle4>=40 ){
//Serial.println("aa");
//delay(5000);}
  delay(1000);
  delay(5000);   }
```