அரசினர் பொறியியற் கல்லூரி – போடிநாயக்கனூர்

GOVERNMENT COLLEGE OF ENGINEERING, BODINAYAKKANUR – 625582
(A Tamil Nadu Government Education Institution Affiliated to Anna University, Chennai)

Prototype for Sign Language Recognition and Conversion Into Text Format for Deaf and Mute Individuals

A PROJECT REPORT

Submitted by

1. **ASWIN RAJ S (923320106005)**
2. **LOGESH A (923320106024)**
3. **RAGUL M (923320106034)**

Under the Guidance of

Prof. N.Sugan, M.E.,(Ph.D),

In partial fulfillment for the award of the

degreeOf

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

GOVERNMENT COLLEGE OF ENGINEERING, BODINAYAKKANUR

ANNA UNIVERSITY: CHENNAI 600 025

May 2024

BONAFIDE CERTIFICATE

This is to certify that this project report entitled **“Prototype for Sign Language Recognition and Conversion into Text Format for Deaf and Mute Individuals”** is the Bonafide work of **ASWIN RAJ S (Reg.No:923320106005), LOGESH A (Reg.No:923320106024), RAGUL M (Reg.No:923320106005)** Who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr. D. MARY SUGANTHARATHNAM, M.E., Ph.D,	**Prof. N.SUGAN,M.E.,(Ph.D),**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
Department of Electronics and	Department of Electronics and
Communication Engineering,	Communication Engineering,
Government College of Engineering,	Government College of Engineering,
Bodinayakkanur-625 582	Bodiyakkanur-625 582

Submitted for PROJECT WORK Viva Voce Examination held at Government College of Engineering, Bodinayakkanur conducted on…………...

INTERNAL EXAMINER	**EXTERNAL EXAMINER**

DECLARATION

We, hereby jointly declare that the project work titled **" Prototype for Sign Language Recognition and Conversion Into Text Format for Deaf and Mute Individuals" Submitted to the Anna University Project viva - voice –**MAY **2024** to " **Bachelor of Engineering in Electronics and Communication Engineering",** is the report of project work done By us under the guidance **Prof. N. SUGAN, M.E.,(Ph.D.),** Department Of Electronics and communication Engineering, Government College of Engineering, Bodinayakkanur.

NAME	SIGNATURE
ASWIN RAJ S	____________________
LOGESH A	____________________
RAGUL M	____________________

I certify that the declaration made by the above candidates is true.

Supervisor,

Prof. N. Sugan, M.E., (Ph.D),

Electronics and Communication Engineering,
Government College of Engineering,
Bodinayakkanur – 625 582

ACKNOWLEDGEMENT

We take this to acknowledge with great pleasure deep satisfaction and convey our gratitude the contribution of many individuals in the successful completion of this project.

We express our profound gratitude to **Prof. Dr.C.VASANTHANAYAKI, M.E., Ph.D., Principal, Government College of Engineering, Bodinayakkanur,** for all the support and encouragement given to us throughout our project.

We thank **Dr. D. MARY SUGANTHARATHNAM M.E., Ph.D. Head of the Department Electronics and Communication Engineering,** Government College of Engineering, Bodinayakkanur, for his valuable suggestions throughout our project.

We express our gratitude to our guide **Prof. N. SUGAN M.E., (Ph.D.), Assistant Professor of Electronics and Communication Engineering,** Government College of Engineering, Bodinayakkanur, for his guidance and help in doing this project.

Finally, we thank our parents for their financial and moral support and also our friends who helped us in completing our project work successfully.

ABSTRACT

A Deaf and Dumb person throughout the world uses sign language for the communication. Deaf and Dumb people are specially trained to use sign language, but normal people do not understand what the dumb and deaf people are trying to convey. The advancement in embedded system can provide a space to design and develop a translation system to convert the sign language into speech. Now a days embedded system has become an important trend in all application. The main aim and the work presented in this paper may remove the communication gap between the impaired and ordinary people.

KEYWORDS: Embedded system, Flex sensors, AVR microcontroller, ASL (American Sign Language), Android application.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATION

AREF	Analogue Reference
EEPROM	Electrically Erasable Programmable Read Only Memory
HC	High Current
12C	Inter Integrated Circuit
IDE	Integrated development organization
LCD	Liquid Crystal Display
LED	Light Emitting Diode
PWM	Pulse Width Modulation
SPI	Serial Peripheral Modulation
SRAM	Static Random Access Memory
UART	Universal Asynchronous Receiver and Transmitter
SVM	Support Vector Machine
ML	Machine Learning

CHAPTER 1
INTRODUCTION

This project aims to develop a prototype system for real-time sign language recognition and conversion into text format. Sign language is a vital mode of communication for the Deaf and Hard of Hearing community. However, there is a gap in bridging the communication divide between sign language users and those who do not understand sign language. The proposed system will utilize computer vision techniques and machine learning to recognize signs made by a user wearing a wearable device equipped with 5 Flex sensors and Combined 3 axis Gyroscope and Accelerometer. The recognized signs will then be converted into text format and displayed on a screen using Support Vector Machine Algorithm. This technology has the potential to facilitate better communication and inclusivity for sign language users.

CHAPTER 2

LITERATURE SURVEY

[1] Give Voice to the Voiceless Using Microcontroller and Digital Gloves Glove-based systems represent one of the most important efforts aimed at acquiring hand movement data. Generally dumb people use sign language for communication but they find difficulty in communicating with others who do not Understand sign language. It is based on the need of developing an electronic device that can translate sign language into speech in order to make the communication take place between the mute communities with the general public possible, a Wireless data gloves is used which normal cloth is driving gloves fitted with flex sensors along the length of each finger and the thumb Mute people can use the gloves to perforin hand gesture and it will be converted into speech so that normal people can understand their expression. This paper provides the map for developing such a digital glove. It also analyses the characteristics of the device and discusses future work. A foremost goal of this paper is to provide readers with a basis for understanding glove system technology.

[2] Image based and Sensor-Based Approaches to Arabic Sign Language recognition. In this paper we propose an Image based system for Arabic sign language recognition. A Gaussian skin color model is used to detect the signer face. The centroid of the detected face is then used as a reference to track the hands movement using region growing from the sequence of images comprising the signs.

[3] Sign Language to Speech Translation System Using PIC Microcontroller. The advancement in embedded system, provides a space to design . and develop a sign language translator system to assist the dump people This paper mainly addresses to facilitate dumb person's lifestyle. Dumb people throughout the world use sign language to communicate with others, this is possible for those who has undergone special trainings. Common people also face difficult to understand the gesture language. To overcome these real time issues, this system is developed. Whenever the proposed system senses any sign language, it plays corresponding recorded voice. This reduces the communication gap between dumb and ordinary people. This proposed model consists of four modules, they are sensing unit, processing unit, voice storage unit and wireless communication unit. It is achieved by integrating flux sensor and APR9600 with PIC16F877A.

[4] Sign Language Recognition System for Deaf and Dumb People. The Sign language is very important for people who have hearing and speaking deficiency generally called Deaf and Mute. It is the only mode of communication for such people to convey their messages and it becomes very important for people to understand their language. This paper proposes the method or algorithm for an application which would help in recognizing the different signs which is called Indian Sign Language. The images are of the palm side of right and left hand and are loaded at runtime. The method has been developed with respect to single user. The real time images will be captured first and then stored in directory and on recently captured image and feature extraction will take place to identify which sign has been articulated by the user through SIFT (scale invariance Fourier transform).

CHAPTER 3
OVERVIEW OF THE PROJECT

3.1 EXISTING SYSTEM

Give Voice to the Voiceless Using Microcontroller and Digital Gloves Glove-based systems represent one of the most important efforts aimed at acquiring hand movement data. Generally dumb people use sign language for communication but they find difficulty in communicating with others who do not Understand sign language. It is based on the need of developing an electronic device that can translate sign language into speech in order to make the communication take place between the mute communities with the general public possible, a Wireless data gloves is used which normal cloth is driving gloves fitted with flex sensors along the length of each finger and the thumb Mute people can use the gloves to perforin hand gesture and it will be converted into speech so that normal people can understand their expression. This paper provides the map for developing such a digital glove. It also analyses the characteristics of the device and discusses future work. A foremost goal of this paper is to provide readers with a basis for understanding glove system technology

3.2 PROPOSED SYSTEM

The proposed system will utilize computer vision techniques and machine learning to recognize signs made by a user wearing a wearable device equipped with 5 Flex sensors and Combined 3 axis Gyroscope and Accelerometer. The recognized signs will then be converted into text format and displayed on a screen using Support Vector Machine Algorithm. This technology has the potential to facilitate better communication and inclusivity for sign language users.

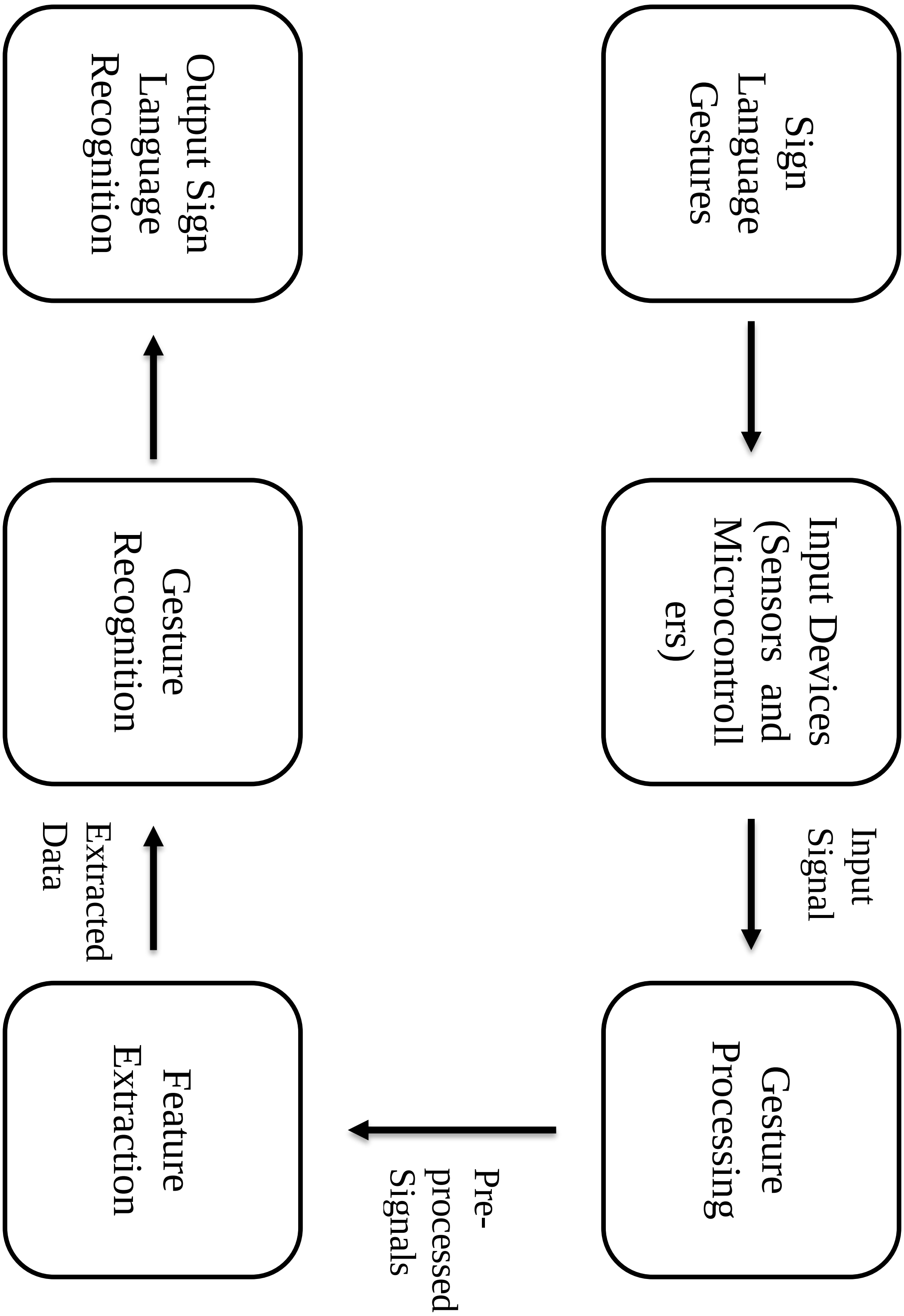
3.3 WORKING OF HAND TALK ASSISTING SYSTEM

In general, deaf people have difficulty in communicating with others who don't understand sign language. Even those who do speak aloud typically have a deaf voice" of which they are self-conscious and that can make the reticent. The Hand Talk glove is a normal, cloth driving gloves fitted with Accelerometer and Flex Sensor along the length of each finger and the thumb.

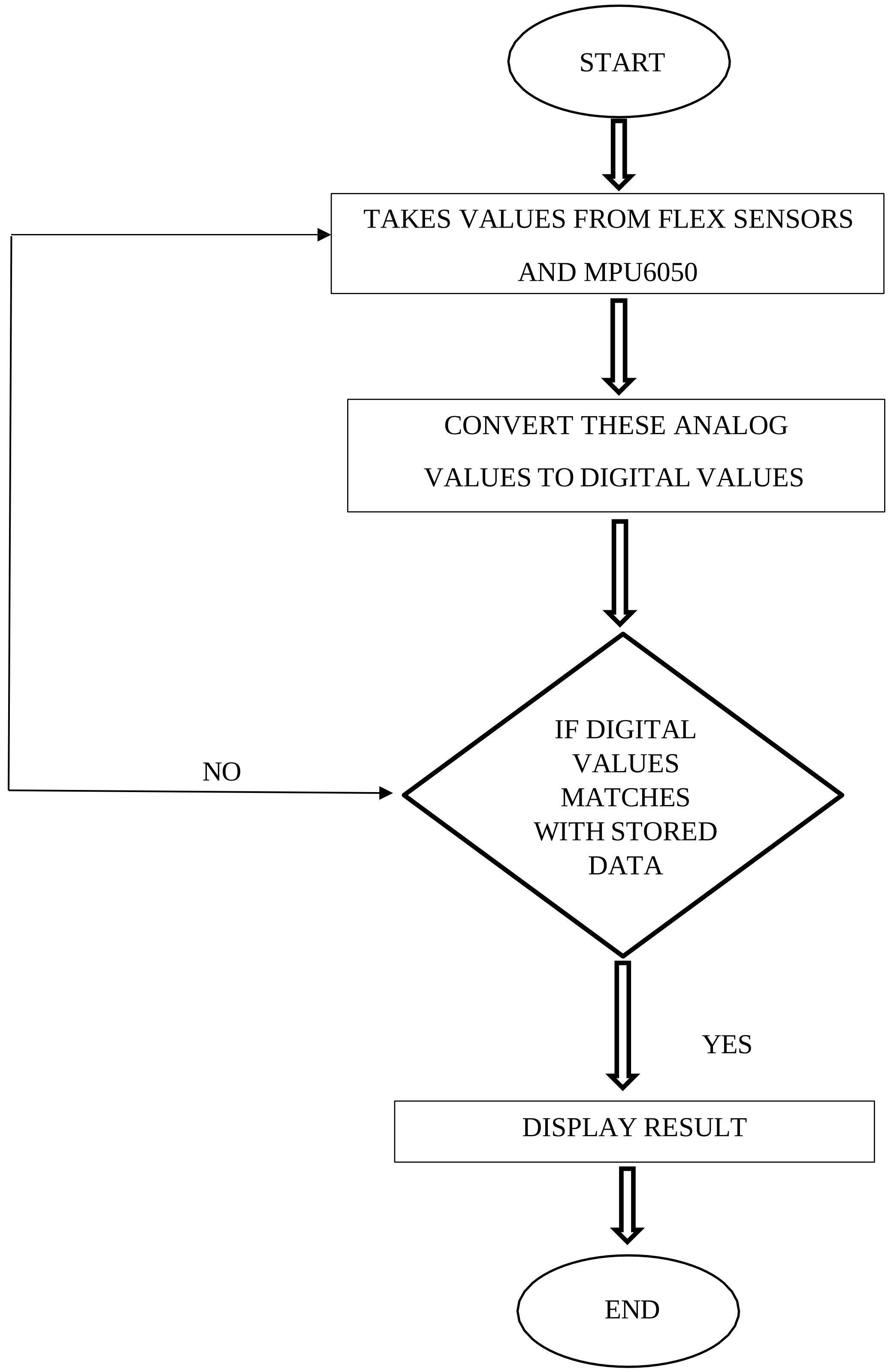
The sensors output a stream of data that varies with degree of bend. The output from the sensor is analog values it is converted to digital and processed by using microcontroller and then it will be transmitted through Wired Communication and processed using responds in texts. In this project Accelerometer and Flex sensor plays the major role, Flex sensors are sensors that change in resistance depending on the amount of bend on the sensor.

They convert the change in bend to electrical resistance - the more the bend, the more the resistance value. They are usually in the form of a thin strip from 1'-5" long that vary in resistance from approximately 10 to 50 kilo ohms. They are often used in gloves to sense finger movement, Flex sensors are analog resistors. They work as variable analog voltage dividers. Inside the Flex Sensors are carbon resistive elements within a thin flexible substrate. More carbon means less resistance. When the substrate is bent the sensor produces a resistance output relative to the bend radius.

3.4. BLOCK DIAGRAM



3.4.1 FLOW CHART



CHAPTER 4

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 electronic 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 microcontrollers that are manufactured by Atmel or a 32-bit Atmel ARM, these Microcontrollers can be programmed easily using the C or CH 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 electrons 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 simple 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++. With so many Arduino boards available in the market, selecting a particular development board needs a variety survey done with respect to their specifications and capabilities

4.1 ARDUINO NANO CONTROLLER

The Arduino Nano is an open-source breadboard-friendly microcontroller board based on the Microchip ATmega328P microcontroller (MCU) and developed by Arduino.cc and initially released in 2008. It offers the same connectivity and specs of the Arduino Uno board in a smaller form factor.

The Arduino Nano is equipped with 30 male I/O headers, in a DIP-30-like configuration, which can be programmed using the Arduino Software integrated development environment (IDE), which is common to all Arduino boards and running both online and offline. The board can be powered through a type-B mini-USB cable or from a 9 V battery.

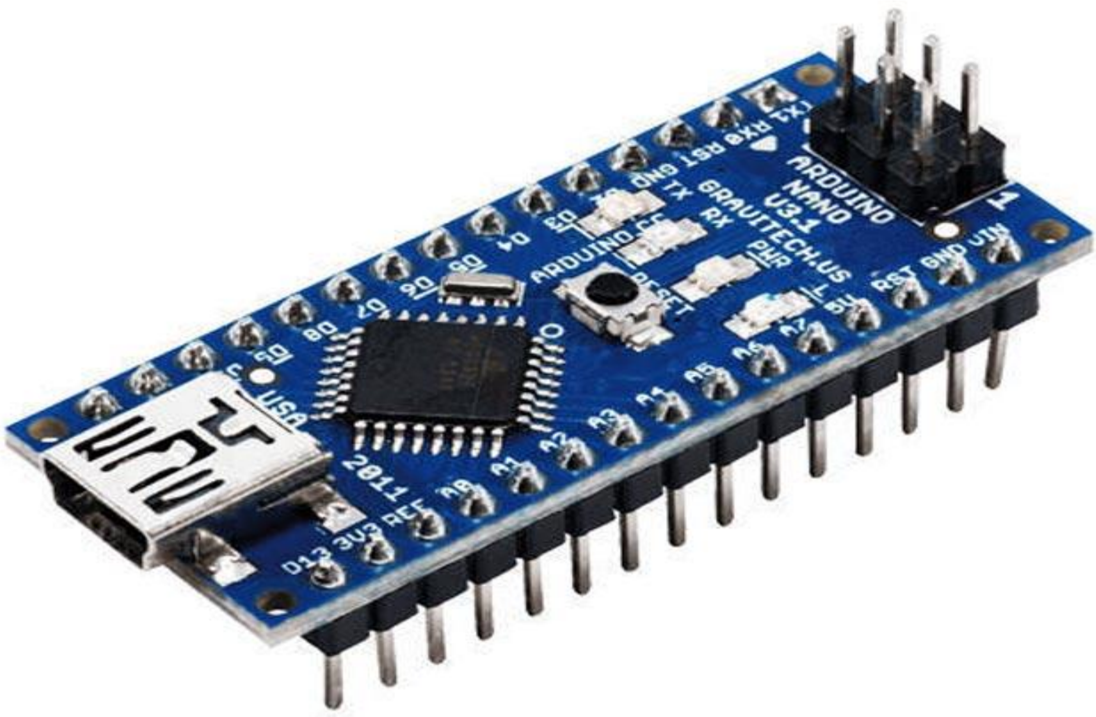
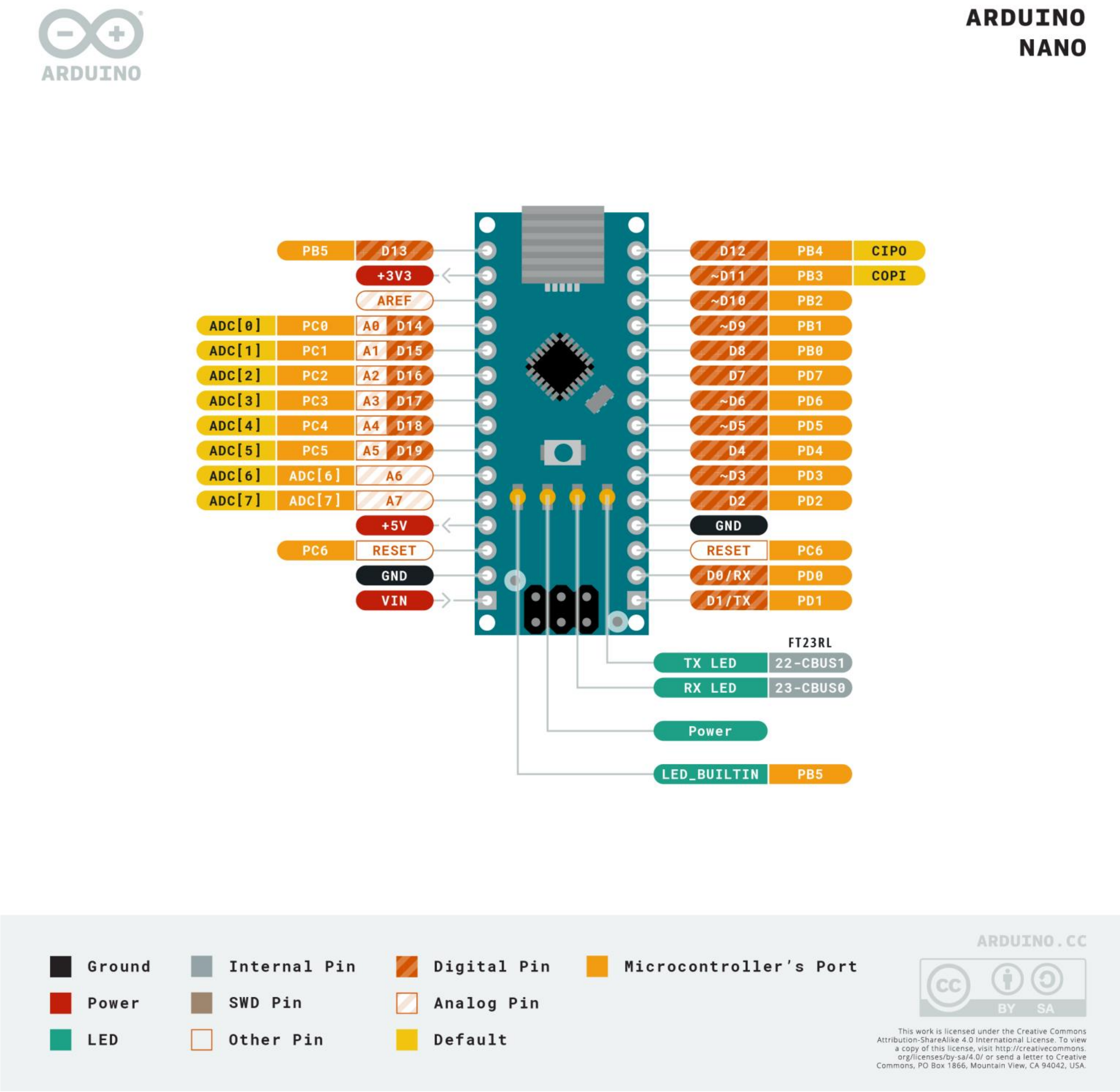


Fig4.1 Arduino Nano Controller

4.1.1 PIN DIAGRAM



4.2 SPECIFICATION

Microcontroller	ATmega328P – 8-bit AVR family microcontroller
Operating Voltage	5V
Recommended Input Voltage for Vin pin	7-12V
Analog Input Pins	6 (A0 – A5)
Digital I/O Pins	14 (Out of which 6 provide PWM output)
DC Current on I/O Pins	40 mA
DC Current on 3.3V Pin	50 mA
Flash Memory	32 KB (2 KB is used for Bootloader)
SRAM	2 KB
EEPROM	1 KB
Frequency (Clock Speed)	16 MHz
Communication	IIC, SPI, USART

Table.1: Arduino UNO specification

4.3 MEMORY:

The Atmega328 has 32 KB of flash memory for storing code (of which 0,5 KB is used for the boot loader); It has also 2 KB of SRAM and 1 KB of EEPROM

4.4 Pin Description:

Each There are total 14 digital Pins and 8 Analog pins on your Nano board. The digital pins can be used to interface sensors by using them as input pins or drive loads by using them as output pins. A simple function like **pinMode()** and **digitalWrite()** can be used to control their operation. The operating voltage is 0V and 5V for digital pins. The analog pins can measure analog voltage from 0V to 5V using any of the 8 Analog pins using a simple function like **analogRead().**

- **Serial Pins 0 (Rx) and 1 (Tx):** Rx and Tx pins are used to receive and transmit TTL serial data. They are connected with the corresponding ATmega328P USB to TTL serial chip.
- **External Interrupt Pins 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.
- **PWM Pins 3, 5, 6, 9 and 11:** These pins provide an 8-bit PWM output by using **analogWrite()** function.
- **SPI Pins 10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK):** These pins are used for SPI communication.
- **In-built LED Pin 13:** This pin is connected with a built-in LED. When pin 13 is HIGH – LED is on and when pin 13 is LOW, it is off.
- **I2C A4 (SDA) and A5 (SCA):** Used for IIC communication using Wire library.
- **AREF:** Used to provide reference voltage for analog inputs with **analogReference()** function.
- **Reset Pin:** Making this pin LOW, resets the microcontroller.

CHAPTER 5

MPU 6050

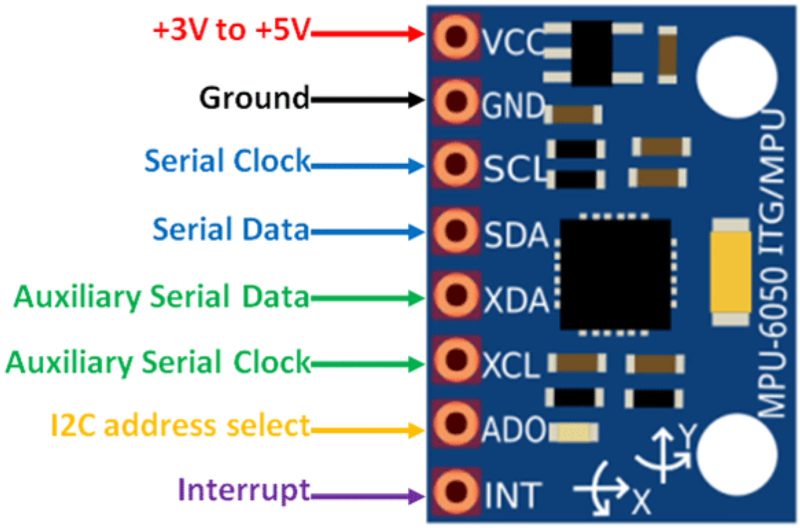
5.1 GENERAL

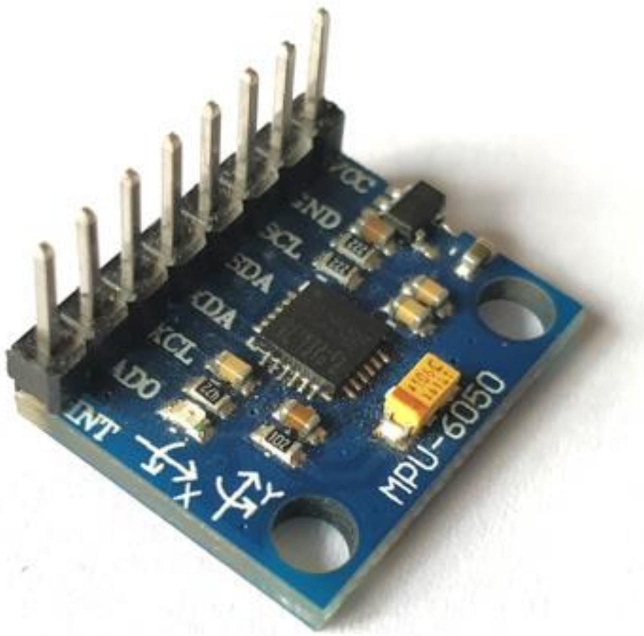
The MPU6050 module is a Micro Electro-Mechanical Systems (MEMS) which consists of a 3-axis Accelerometer and 3-axis Gyroscope inside it. This helps us to measure acceleration, velocity, orientation, displacement and many other motion related parameter of a system or object. This module also has a (DMP) Digital Motion Processor inside it which is powerful enough to perform complex calculation and thus free up the work for Microcontroller.

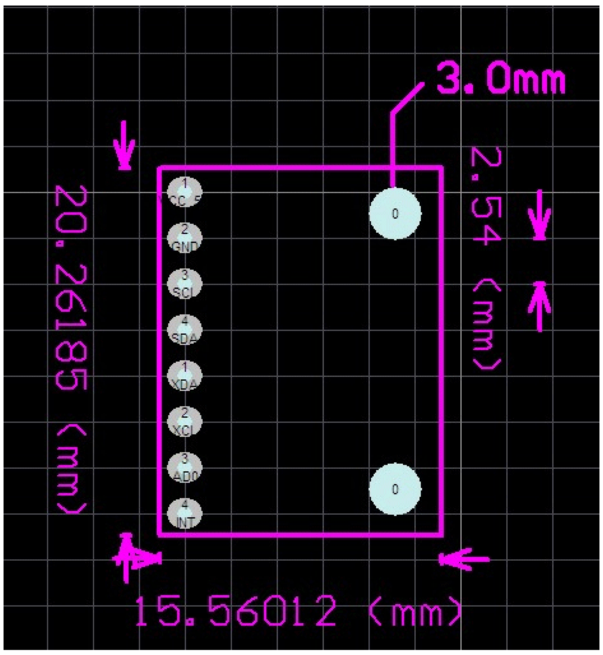
The module also have two auxiliary pins which can be used to interface external IIC modules like an magnetometer, however it is optional. Since the IIC address of the module is configurable more than one MPU6050 sensor can be interfaced to a Microcontroller using the AD0 pin. This module also has well documented and revised libraries available hence it’s very easy to use with famous platforms like Arduino. So, if you are looking for a sensor to control motion for your RC Car, Drone, Self balancing Robot, Humanoid, Biped or something like that then this sensor might be the right choice for you.

The hardware of the module is very simple, it actually comprises of the MPU6050 as the main components as shown above. Since the module works on 3.3V, a voltage regulator is also used. The IIC lines are pulled high using a 4.7k resistor and the interrupt pin is pulled down using another 4.7k resistor.

The MPU6050 module allows us to read data from it through the IIC bus. Any change in motion will be reflected on the mechanical system which will in turn vary the voltage. Then the IC has a 16-bit ADC which it uses to accurately read these changes in voltage and stores it in the FIFO buffer and makes the INT (interrupt) pin to go high. This means that the data is ready to be read, so we use a MCU to read the data from this FIFO buffer through IIC communication. As easy as it might sound, you may face some problem while actually trying to make sense of the data. However there are lots of platforms like Arduino using which you can start using this module in no time by utilizing the readily available libraries.







CHAPTER 6

FLEX SENSOR

6.1 GENERAL

Flex sensors are usually available in two sizes. One is **2.2 inch** and another is **4.5 inch**. Although the sizes are different the basic function remains the same. They are also divided based on resistance. There are LOW resistance, MEDIUM resistance and HIGH resistance types. Choose the appropriate type depending on requirement. Here we are going to discuss 2.2inch Flex sensor that is **FS-L-0055**.

FLEX SENSOR function

FLEX SENSOR terminal resistance changes when it is bent.

FLEX SENSOR Pin Configuration

Flex sensor is a two-terminal device. The Flex sensor does not have polarized terminals like diode. So, there is no positive and negative.

Pin Number	**Description**
P1	Usually connected to positive of power source.
P2	Usually connected to ground.

FLEX SENSOR Features and Specifications

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

Where to Use FLEX SENSOR

For understanding the use of FLEX SENSOR consider:

Case1: Where you want to check whether the surface of a device or thing is levelled or not. Say you want a device to check whether a window or door is open or not. At that time a Flex sensor could be used. The sensor could be fixed at door edge and when the door opens the Flex sensor gets flexed. With the sensor being flexed its parameters changes which could be designed to provide an alert.

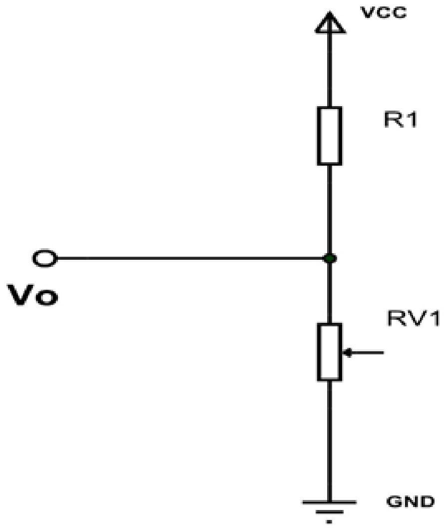
Case2: Where you want to measure the FLEX or BENT or ANGLE change of any instrument or device. The **FLEX SENSOR internal resistance** changes almost linearly with its flex angle. So, by sticking the sensor to the instrument, we can have the flex angle in electrical parameter of resistance.

How to Use FLEX SENSOR

As mentioned earlier, **FLEX SENSOR** is basically a **VARIABLE RESISTOR** whose terminal resistance increases when the sensor is bent. So, this sensor resistance increases depends on surface linearity. So, it is usually used to sense the changes in linearity.

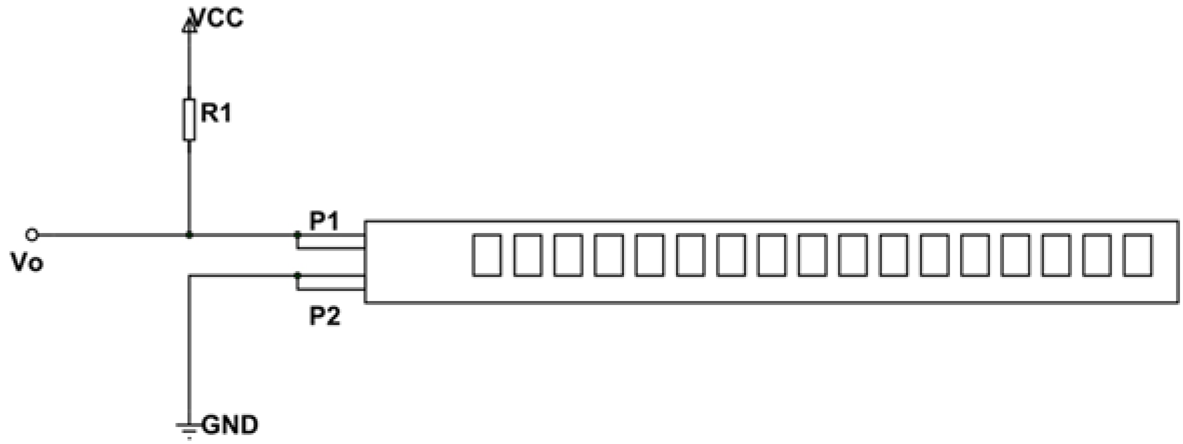
As shown in the above figure, when the surface of FLEX SENSOR is completely linear it will be having its nominal resistance. When it is bent 45º angle the FLEX SENSOR resistance increases too twice as before. And when the bent is 90º the resistance could go as high as four times the nominal resistance. So, the resistance across the terminals rises linearly with bent angle. So, in a sense the FLEX sensor converts flex angle to RESISTANCE parameter.

For convenience we convert this RESISTANCE parameter to VOLTAGE parameter. For that we are going to use **VOLTAGE DIVIDER circuit**. A typical **VOLTAGE DIVIDER circuit** is shown below.



In this resistive network we have two resistances. One is constant resistance (R1) and other is variable resistance (RV1). Vo is the voltage at midpoint of VOLTAGE DIVIDER circuit and is also the output voltage. Vo is also the voltage across the variable resistance (RV1). So, when the resistance value of RV1 is changed the output voltage Vo also changes. So, we will have resistance change in voltage change with VOLTAGE DIVIDER circuit.

Here we will replace the variable resistance (RV1) with FLEX SENSOR. The circuit will be as below.



As shown in figure, R1 here is a constant resistance and **FLEX SENSOR** which acts as a variable resistance. Vo being output voltage and also the voltage across the FLEX SENSOR.

Here,

Vo=VCC(Rx/(R1+Rx)).

Rx - FLEX SENSOR resistance

Now, when the FLEX SENSOR is bent the terminal resistance increases. This increase also appears in VOLTAGE DIVIDER circuit. With that the drop across the FLEX SENSOR increases so is Vo. So, with increase in bent of FLEX sensor Vo voltage increases linearly. With that we have VOLTAGE parameter representing the flex.

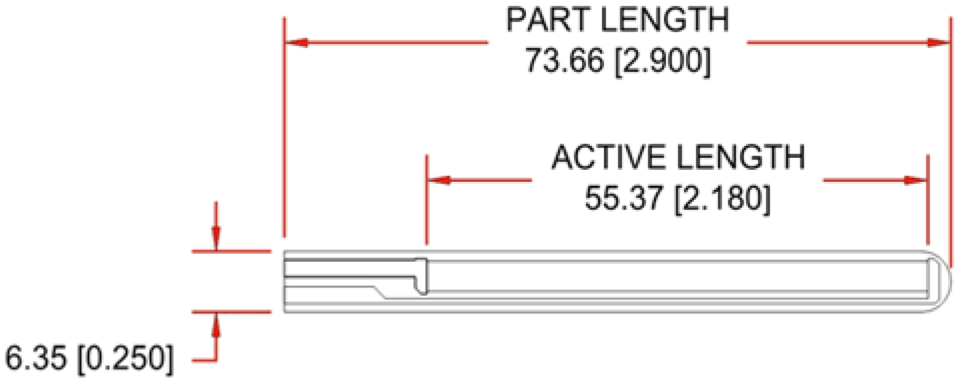
We can take this VOLTAGE parameter and feed it to ADC to get the digital value which can be used conveniently.

Applications

- Robotics
- Gaming (Virtual Motion)
- Medical Devices
- Computer Peripherals
- Musical Instruments
- Physical Therapy

2D-Model and Dimensions

Measurements in millimetre (inches)



CHAPTER 7

CLASSIFIERS

7.1 Overview to Classifiers

Classifiers are algorithms used in machine learning to categorize or label data points based on their features. The process of classification involves training a model on a dataset where each data point is associated with a class label. The goal is for the model to learn the relationship between the features and the labels so that it can predict the correct label for new, unseen data.

7.2 Working of Classifiers:

Training Phase: During this phase, the classifier is provided with a dataset containing input features and corresponding labels. The classifier learns patterns in the data to differentiate between different classes.

Testing Phase: After training, the classifier is tested on a separate dataset that it hasn't seen before. This allows us to evaluate its performance and see how well it generalizes to new, unseen data.

Prediction Phase: Once trained and evaluated, the classifier can be used to predict the class labels of new data points.

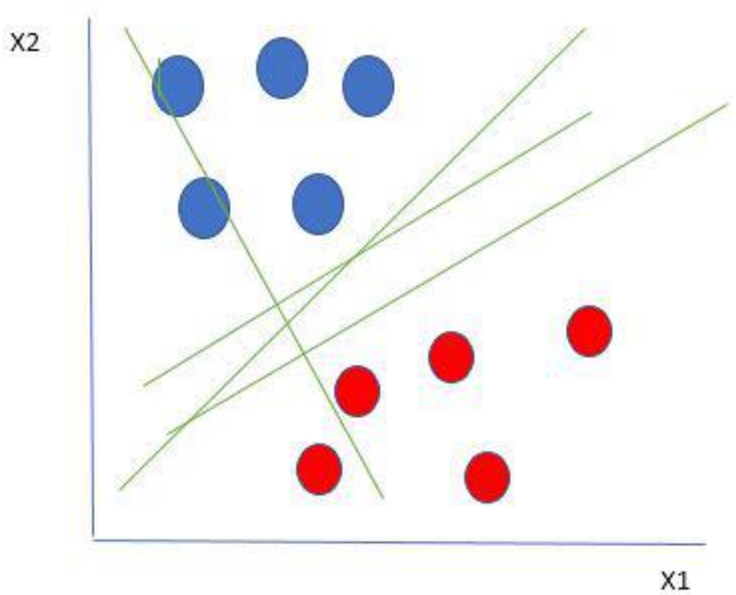
In this project we are using,

- **Support Vector Machine**
- **Neural Network**

7.3 SUPPORT VECTOR MACHINE

Support Vector Machine (SVM) is a supervised machine learning algorithm used for both classification and regression. Though we say regression problems as well it's best suited for classification. The main objective of the SVM algorithm is to find the optimal hyperplane in an N-dimensional space that can separate the data points in different classes in the feature space. The hyperplane tries that the margin between the closest points of different classes should be as maximum as possible. The dimension of the hyperplane depends upon the number of features. If the number of input features is two, then the hyperplane is just a line. If the number of input features is three, then the hyperplane becomes a 2-D plane. It becomes difficult to imagine when the number of features exceeds three.

Let's consider two independent variables x1, x2, and one dependent variable which is either a blue circle or a red circle.



From the figure above it's very clear that there are multiple lines (our hyperplane here is a line because we are considering only two input features x1, x2) that segregate our data points or do a classification between red and blue circles.

7.4 NEURAL NETWORK

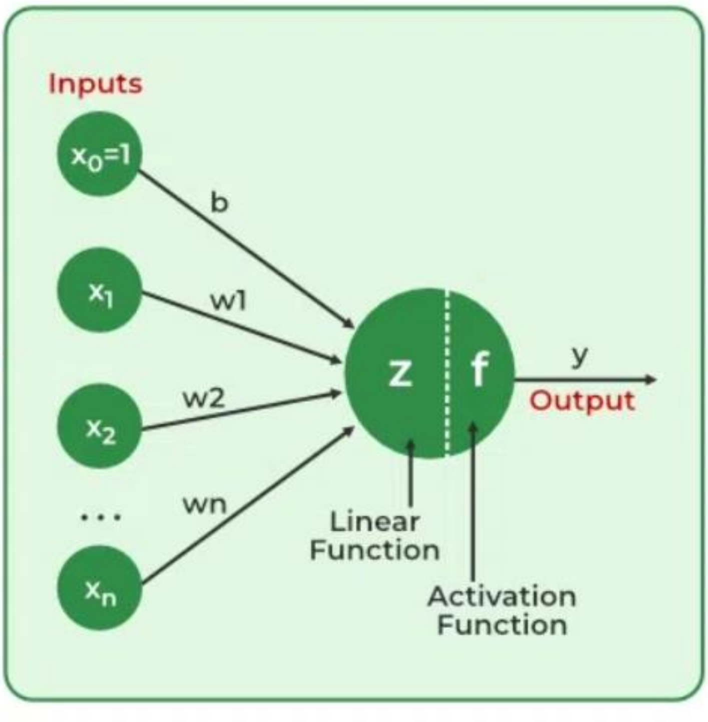
Neural networks extract identifying features from data, lacking pre-programmed understanding. Network components include neurons, connections, weights, biases, propagation functions, and a learning rule. Neurons receive inputs, governed by thresholds and activation functions. Connections involve weights and biases regulating information transfer. Learning, adjusting weights and biases, occurs in three stages: input computation, output generation, and iterative refinement enhancing the network's proficiency in diverse tasks.

These include:

1.The neural network is simulated by a new environment.

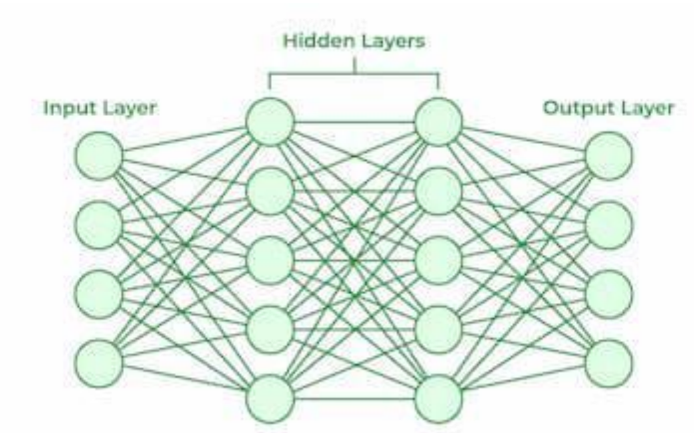
2.Then the free parameters of the neural network are changed as a result of this simulation.

3.The neural network then responds in a new way to the environment because of the changes in its free parameters.



7.5 WORKING OF A NEURAL NETWORK

Neural networks are complex systems that mimic some features of the functioning of the human brain. It is composed of an input layer, one or more hidden layers, and an output layer made up of layers of artificial neurons that are coupled. The two stages of the basic process are called backpropagation and forward propagation.



7.6 WORKING OF A NEURAL NETWORK

Forward Propagation

Input Layer: Each feature in the input layer is represented by a node on the network, which receives input data.

Weights and Connections: The weight of each neuronal connection indicates how strong the connection is. Throughout training, these weights are changed.

Hidden Layers: Each hidden layer neuron processes inputs by multiplying them by weights, adding them up, and then passing them through an activation function. By doing this, non-linearity is introduced, enabling the network to recognize intricate patterns.

Output: The final result is produced by repeating the process until the output layer is reached.

Backpropagation

Loss Calculation: The network’s output is evaluated against the real goal values, and a loss function is used to compute the difference. For a regression problem, the Mean Squared Error (MSE) is commonly used as the cost function.

Loss Function: $MSE = \frac{1}{N}\sum_{i=1}^{N}\left(y(i) - \overline{y(\iota)}\right)^2$

Gradient Descent: Gradient descent is then used by the network to reduce the loss. To lower the inaccuracy, weights are changed based on the derivative of the loss with respect to each weight.

Adjusting weights: The weights are adjusted at each connection by applying this iterative process, or backpropagation, backward across the network.

Training: During training with different data samples, the entire process of forward propagation, loss calculation, and backpropagation is done iteratively, enabling the network to adapt and learn patterns from the data.

Activation Functions: Model non-linearity is introduced by activation functions like the rectified linear unit (REL) or sigmoid. Their decision on whether to “fire” a neuron is based on the whole weighted input.

CHAPTER 8

SIGN LANGUAGE RECOGNITION SYSTEM MODULE

8.1 GENERAL

To establish a communication or interaction with Deaf and Mute people is utter importance nowadays. These people interact through hand gestures or is Gestures are basically the physical action form performed by a person to some meaningful information.

Gestures are a powerful means of communication among humans. In fact, gesturing is so deeply rooted in our communication that people often continue gesturing when speaking on the telephone. There are various signs which express complex meanings and recognizing them is a challenging task for people who have no understanding for that language.

It becomes difficult finding a well experienced and educated translator for the sign language every time and everywhere but human-computer interaction system for this can be installed anywhere possible.

The motivation for developing such helpful application came from the fact that it would prove to be of utmost importance for socially aiding people and how it would help increasingly for social awareness as well.

The remarkable ability of the human vision is the gesture recognition, it is noticeable mainly in deaf people when they communicating with each other via sign language and with hearing people as well. In this paper we take up one of the social challenges to give this set of mass a permanent solution in communicating with normal human beings.

CHAPTER 9
SOFTWARE REQUIREMENT

9.1 ARDUINO IDE

A minimal Arduino embedded C sketch, as seen by the Arduino IDE programmer, consist of only two functions:

Setup: This function is called once when a sketch starts after power-up or reset. It is used to initialize variables, input and output pin modes, and other libraries needed in the sketch

Loop: After setup has been called, function loop is executed repeatedly in the main program. It controls the board until the board is powered off or is reset.

9.2 PYTHON

Python is a high-level, general-purpose, and very popular programming language. Python programming language (latest Python 3) is being used in web development, and Machine Learning applications, along with all cutting-edge technology in Software Industry. Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

Python is a versatile and widely-used programming language with a vast ecosystem. Here are some areas where Python is commonly used:

Web Development: Python is used to build web applications using frameworks like Django, Flask, and Pyramid. These frameworks provide tools and libraries for handling web requests, managing databases, and more.

Data Science and Machine Learning: Python is popular in data science and machine learning due to libraries like NumPy, pandas, Matplotlib, and scikit-learn. These libraries provide tools for data manipulation, analysis, visualization, and machine learning algorithms.

9.3 MATLAB

MATLAB, short for "MATrix LABoratory," is a high-level programming language and environment designed for numerical computing, data analysis, and visualization. It provides a wide range of tools and functions that allow users to perform various computational tasks efficiently.

Here are some key features and uses of MATLAB:

Matrix Operations: MATLAB is optimized for matrix operations, making it particularly useful for linear algebra computations. Matrices are fundamental data structures in MATLAB, and many mathematical operations are performed on them.

Numerical Analysis: MATLAB provides a vast array of built-in functions for numerical analysis, including solving differential equations, optimization, interpolation, and numerical integration.

Data Analysis and Visualization: MATLAB offers powerful tools for analyzing and visualizing data. It supports various plotting functions for creating 2D and 3D plots, histograms, scatter plots, and more.

Algorithm Development: MATLAB is widely used for developing and testing algorithms due to its extensive library of built-in functions and tools for algorithm design.

Simulink: Simulink is an extension of MATLAB that provides a graphical environment for modeling, simulating, and analyzing dynamic systems. It is commonly used in control systems engineering, robotics, and signal processing.

Integration and Deployment: MATLAB can be integrated with other programming languages and tools, such as C/C++, Python, and Java. It also provides options for deploying MATLAB applications as standalone executables, web applications, or as shared libraries.

CHAPTER 10

RESULT

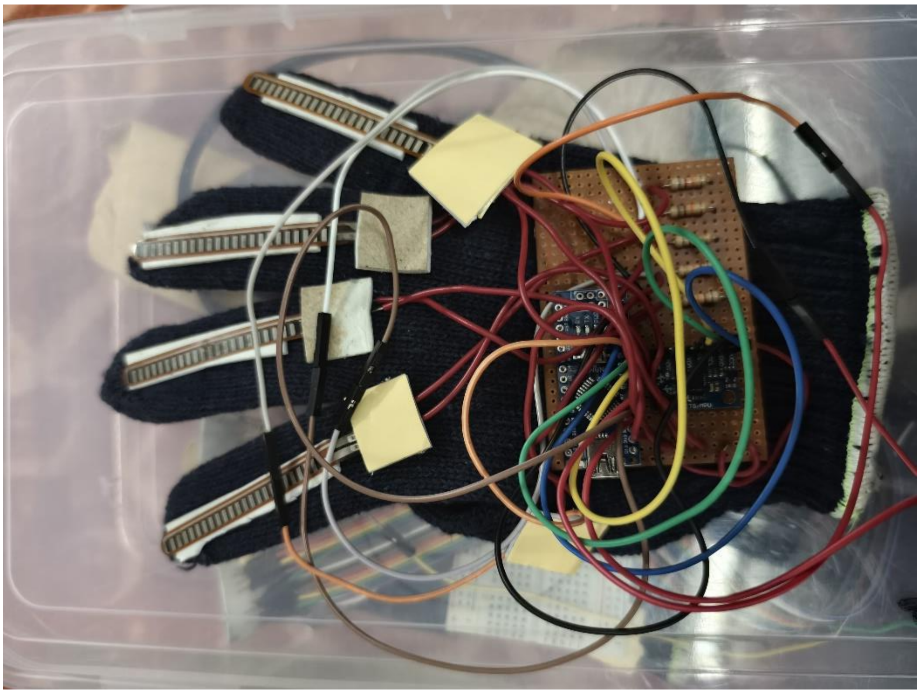
Sign language use gestures instead of sound to convey information. In this flex sensor and MPU6050 plays the major role. Arduino microcontroller is used to take system flexes out from flex sensors and then this analog data is converted to digital form. the data from microcontroller is sent to android phone connected to loud speak corresponding character which has hen sensed.

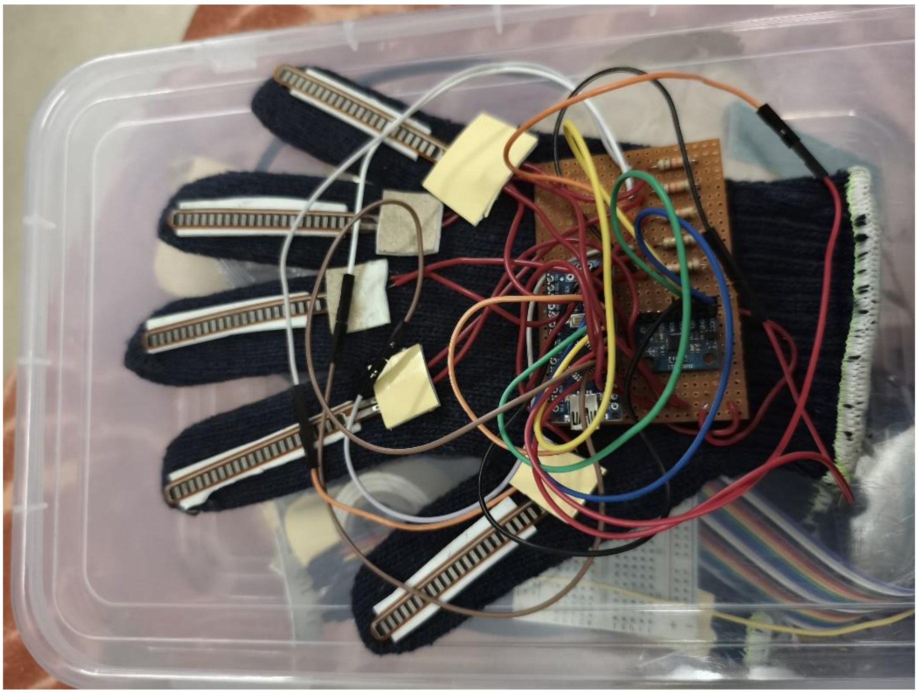
Step 1

Accelerometer and flex sensor are attached to the gloves of the speech challenged person hand

Step 2

The output of the accelerometer is based on the movement made by the glove.



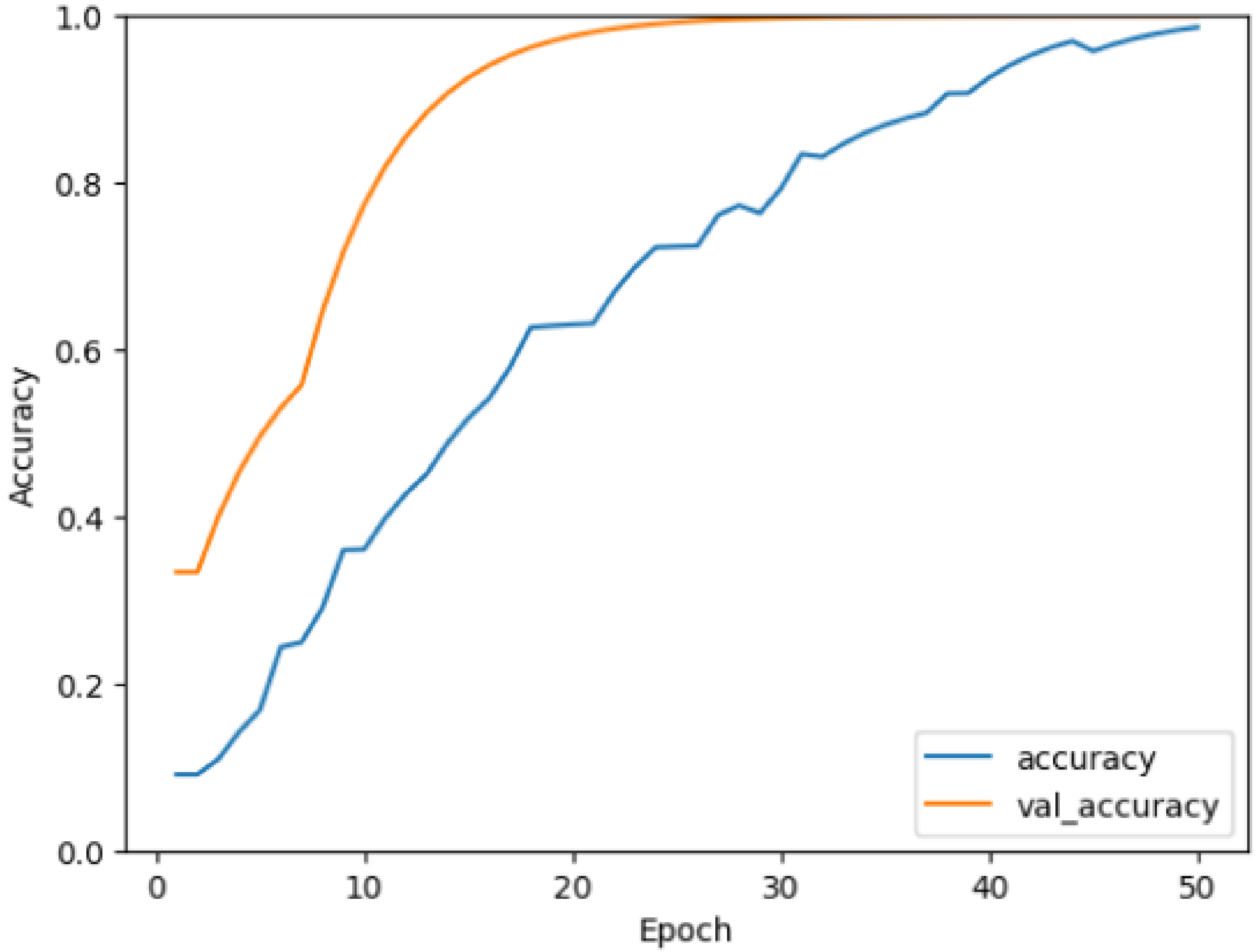


Step 3

ROC GRAPH

NEURAL NETWORK OUTPUT

Accuracy Graph:



Accuracy Obtained:

```
Test Loss: 0.7734860181808472
Test Accuracy: 0.5
```

EPOCH RUNNING RATE

Epoch 1/50 1/1 [==============================] - 1s 1s/step - loss: 1.9578 - accuracy: 0.0909 - val_loss: 1.6631 - val_accuracy: 0.3333 Epoch 2/50 1/1 [==============================] - 0s 62ms/step - loss: 1.9870 - accuracy: 0.0909 - val_loss: 1.6395 - val_accuracy: 0.3333 Epoch 3/50 1/1 [==============================] - 0s 39ms/step - loss: 1.7929 - accuracy: 0.1818 - val_loss: 1.6154 - val_accuracy: 0.6667 Epoch 4/50 1/1 [==============================] - 0s 40ms/step - loss: 1.6785 - accuracy: 0.2727 - val_loss: 1.5895 - val_accuracy: 0.6667 Epoch 5/50 1/1 [==============================] - 0s 37ms/step - loss: 1.6103 - accuracy: 0.2727 - val_loss: 1.5644 - val_accuracy: 0.6667 Epoch 6/50 1/1 [==============================] - 0s 39ms/step - loss: 1.4508 - accuracy: 0.5455 - val_loss: 1.5383 - val_accuracy: 0.6667 Epoch 7/50 1/1 [==============================] - 0s 39ms/step - loss: 1.5730 - accuracy: 0.2727 - val_loss: 1.5136 - val_accuracy: 0.6667 Epoch 8/50 1/1 [==============================] - 0s 42ms/step - loss: 1.5310 - accuracy: 0.4545 - val_loss: 1.4872 - val_accuracy: 1.0000 Epoch 9/50 1/1 [==============================] - 0s 39ms/step - loss: 1.5315 - accuracy: 0.6364 - val_loss: 1.4585 - val_accuracy: 1.0000 Epoch 10/50 1/1 [==============================] - 0s 40ms/step - loss: 1.4491 - accuracy: 0.3636 - val_loss: 1.4287 - val_accuracy: 1.0000 Epoch 11/50 1/1 [==============================] - 0s 38ms/step - loss: 1.2397 - accuracy: 0.5455 - val_loss: 1.3984 - val_accuracy: 1.0000 Epoch 12/50 1/1 [==============================] - 0s 38ms/step - loss: 1.2941 - accuracy: 0.5455 - val_loss: 1.3665 - val_accuracy: 1.0000 Epoch 13/50 1/1 [==============================] - 0s 56ms/step - loss: 1.1494 - accuracy: 0.5455 - val_loss: 1.3344 - val_accuracy: 1.0000 Epoch 14/50 1/1 [==============================] - 0s 41ms/step - loss: 1.0972 - accuracy: 0.6364 - val_loss: 1.3024 - val_accuracy: 1.0000 Epoch 15/50 1/1 [==============================] - 0s 42ms/step - loss: 1.0570 - accuracy: 0.6364 - val_loss: 1.2717 - val_accuracy: 1.0000 Epoch 16/50 1/1 [==============================] - 0s 43ms/step - loss: 1.0931 - accuracy: 0.6364 - val_loss: 1.2419 - val_accuracy: 1.0000 Epoch 17/50 1/1 [==============================] - 0s 38ms/step - loss: 0.8441 - accuracy: 0.7273 - val_loss: 1.2125 - val_accuracy: 1.0000 Epoch 18/50 1/1 [==============================] - 0s 37ms/step - loss: 0.9656 - accuracy: 0.8182 - val_loss: 1.1834 - val_accuracy: 1.0000 Epoch 19/50 1/1 [==============================] - 0s 39ms/step - loss: 0.8989 - accuracy: 0.6364 - val_loss: 1.1522 - val_accuracy: 1.0000 Epoch 20/50 1/1 [==============================] - 0s 39ms/step - loss: 0.8679 - accuracy: 0.6364 - val_loss: 1.1190 - val_accuracy: 1.0000 Epoch 21/50 1/1 [==============================] - 0s 40ms/step - loss: 0.9726 - accuracy: 0.6364 - val_loss: 1.0847 - val_accuracy: 1.0000 Epoch 22/50 1/1 [==============================] - 0s 38ms/step - loss: 0.6782 -

accuracy: 0.8182 - val_loss: 1.0507 - val_accuracy: 1.0000 Epoch 23/50 1/1 [==============================] - 0s 37ms/step - loss: 0.8522 - accuracy: 0.8182 - val_loss: 1.0160 - val_accuracy: 1.0000 Epoch 24/50 1/1 [==============================] - 0s 56ms/step - loss: 0.6650 - accuracy: 0.8182 - val_loss: 0.9806 - val_accuracy: 1.0000 Epoch 25/50 1/1 [==============================] - 0s 49ms/step - loss: 0.8491 - accuracy: 0.7273 - val_loss: 0.9458 - val_accuracy: 1.0000 Epoch 26/50 1/1 [==============================] - 0s 37ms/step - loss: 0.6908 - accuracy: 0.7273 - val_loss: 0.9098 - val_accuracy: 1.0000 Epoch 27/50 1/1 [==============================] - 0s 37ms/step - loss: 0.7644 - accuracy: 0.9091 - val_loss: 0.8751 - val_accuracy: 1.0000 Epoch 28/50 1/1 [==============================] - 0s 40ms/step - loss: 0.6049 - accuracy: 0.8182 - val_loss: 0.8410 - val_accuracy: 1.0000 Epoch 29/50 1/1 [==============================] - 0s 39ms/step - loss: 0.7421 - accuracy: 0.7273 - val_loss: 0.8065 - val_accuracy: 1.0000 Epoch 30/50 1/1 [==============================] - 0s 38ms/step - loss: 0.5314 - accuracy: 0.9091 - val_loss: 0.7730 - val_accuracy: 1.0000 Epoch 31/50 1/1 [==============================] - 0s 37ms/step - loss: 0.4172 - accuracy: 1.0000 - val_loss: 0.7401 - val_accuracy: 1.0000 Epoch 32/50 1/1 [==============================] - 0s 37ms/step - loss: 0.5964 - accuracy: 0.8182 - val_loss: 0.7069 - val_accuracy: 1.0000 Epoch 33/50 1/1 [==============================] - 0s 58ms/step - loss: 0.5381 - accuracy: 0.9091 - val_loss: 0.6755 - val_accuracy: 1.0000 Epoch 34/50 1/1 [==============================] - 0s 37ms/step - loss: 0.4044 - accuracy: 0.9091 - val_loss: 0.6445 - val_accuracy: 1.0000 Epoch 35/50 1/1 [==============================] - 0s 54ms/step - loss: 0.3823 - accuracy: 0.9091 - val_loss: 0.6147 - val_accuracy: 1.0000 Epoch 36/50 1/1 [==============================] - 0s 39ms/step - loss: 0.4358 - accuracy: 0.9091 - val_loss: 0.5850 - val_accuracy: 1.0000 Epoch 37/50 1/1 [==============================] - 0s 37ms/step - loss: 0.3298 - accuracy: 0.9091 - val_loss: 0.5569 - val_accuracy: 1.0000 Epoch 38/50 1/1 [==============================] - 0s 40ms/step - loss: 0.2120 - accuracy: 1.0000 - val_loss: 0.5300 - val_accuracy: 1.0000 Epoch 39/50 1/1 [==============================] - 0s 40ms/step - loss: 0.3963 - accuracy: 0.9091 - val_loss: 0.5041 - val_accuracy: 1.0000 Epoch 40/50 1/1 [==============================] - 0s 39ms/step - loss: 0.2639 - accuracy: 1.0000 - val_loss: 0.4797 - val_accuracy: 1.0000 Epoch 41/50 1/1 [==============================] - 0s 40ms/step - loss: 0.3008 - accuracy: 1.0000 - val_loss: 0.4564 - val_accuracy: 1.0000 Epoch 42/50 1/1 [==============================] - 0s 41ms/step - loss: 0.2477 - accuracy: 1.0000 - val_loss: 0.4338 - val_accuracy: 1.0000 Epoch 43/50 1/1 [==============================] - 0s 39ms/step - loss: 0.1529 - accuracy: 1.0000 - val_loss: 0.4120 - val_accuracy: 1.0000 Epoch 44/50 1/1 [==============================] - 0s 38ms/step - loss: 0.2814 - accuracy: 1.0000 - val_loss: 0.3908 - val_accuracy: 1.0000 Epoch 45/50 1/1

[==============================] - 0s 42ms/step - loss: 0.2698 - accuracy: 0.9091 - val_loss: 0.3718 - val_accuracy: 1.0000 Epoch 46/50 1/1 [==============================] - 0s 40ms/step - loss: 0.1126 - accuracy: 1.0000 - val_loss: 0.3538 - val_accuracy: 1.0000 Epoch 47/50 1/1 [==============================] - 0s 42ms/step - loss: 0.2244 - accuracy: 1.0000 - val_loss: 0.3348 - val_accuracy: 1.0000 Epoch 48/50 1/1 [==============================] - 0s 47ms/step - loss: 0.1770 - accuracy: 1.0000 - val_loss: 0.3173 - val_accuracy: 1.0000 Epoch 49/50 1/1 [==============================] - 0s 41ms/step - loss: 0.1410 - accuracy: 1.0000 - val_loss: 0.3005 - val_accuracy: 1.0000 Epoch 50/50 1/1 [==============================] - 0s 38ms/step - loss: 0.1444 - accuracy: 1.0000 - val_loss: 0.2850 - val_accuracy: 1.0000

DATA SET

Flex 1	Flex 2	Flex 3	Flex 4	Flex 5	Flex Resistance 1	Flex Resistance 2	Flex Resistance 3	Flex Resistance 4	Flex Resistance 5	Flex Bending 1	Flex Bending 2	Flex Bending 3	Flex Bending 4	Flex Bending 5	X A	Y A	Z A	X G	Y G	Z A	Gesture
672	679	693	778	756	13058.04	12665.69	11904.76	7872.75	8829.36	34	35	37	49	46	9408	13900	2012	-208	230	-6	A
678	682	698	778	744	12721.24	12500	11640.4	7872.75	9375	35	36	38	49	45	9436	14320	2308	597	-676	-223	A
686	684	698	778	745	12281.34	12390.35	11640.4	7872.75	9328.86	36	36	38	49	45	8768	15092	1548	-10	-50	301	A
757	752	781	782	709	8784.68	9009.31	7746.48	7704.6	11071.93	46	46	49	50	40	2352	12740	-11512	-238	16	106	B
757	751	780	782	708	8784.68	9054.6	7788.46	7704.6	11122.88	46	46	49	50	40	2748	14060	-10032	-757	52	69	B
760	744	771	783	705	8651.32	9375	8171.21	7662.84	11276.6	47	45	48	50	39	3692	16168	-3688	-94	23	243	B
744	693	711	780	737	9375	11904.76	10970.46	7788.46	9701.49	45	37	40	49	44	4320	15928	-2924	281	119	82	C
729	706	727	781	757	10082.3	11225.21	10178.82	7746.48	8784.68	43	39	42	49	46	1808	15472	-7152	121	374	19	C
729	712	734	781	753	10082.3	10919.94	9843.33	7746.48	8964.14	43	40	43	49	46	3476	16340	-2408	-138	-91	284	C
686	695	691	782	697	12281.34	11798.56	12011.58	7704.6	11692.97	36	38	37	50	38	5280	15552	3224	-39	39	296	D
694	704	702	783	725	11851.58	11328.12	11431.62	7662.84	10275.86	38	39	39	50	42	3240	15628	2800	-235	355	420	D
696	709	708	783	714	11745.69	11071.93	11122.68	7662.84	10819.33	38	40	40	50	41	2448	15732	3844	-119	205	226	D
686	669	639	773	693	12281.34	13228.7	15023.47	8085.38	11904.76	36	34	28	48	37	8984	12868	4984	139	44	238	E
694	680	651	774	701	11851.58	12610.3	14285.71	8042.64	11483.59	38	35	31	49	39	6664	15144	3184	-25	-34	244	E
704	686	661	775	703	11328.12	12281.34	13691.38	8000	11379.8	39	36	32	49	39	5252	15916	528	-71	-12	197	E
764	752	781	776	736	8475.13	9009.31	7746.48	7957.47	9748.64	47	46	49	49	44	9612	13604	2924	-80	-349	311	F
760	751	778	777	734	8651.32	9054.6	7872.75	7915.06	9843.33	47	46	49	49	43	4692	15952	684	-29	-44	292	F
759	752	777	777	737	8695.65	9009.31	7915.06	7915.06	9701.49	47	46	49	49	44	4060	16272	-700	-521	11	487	F

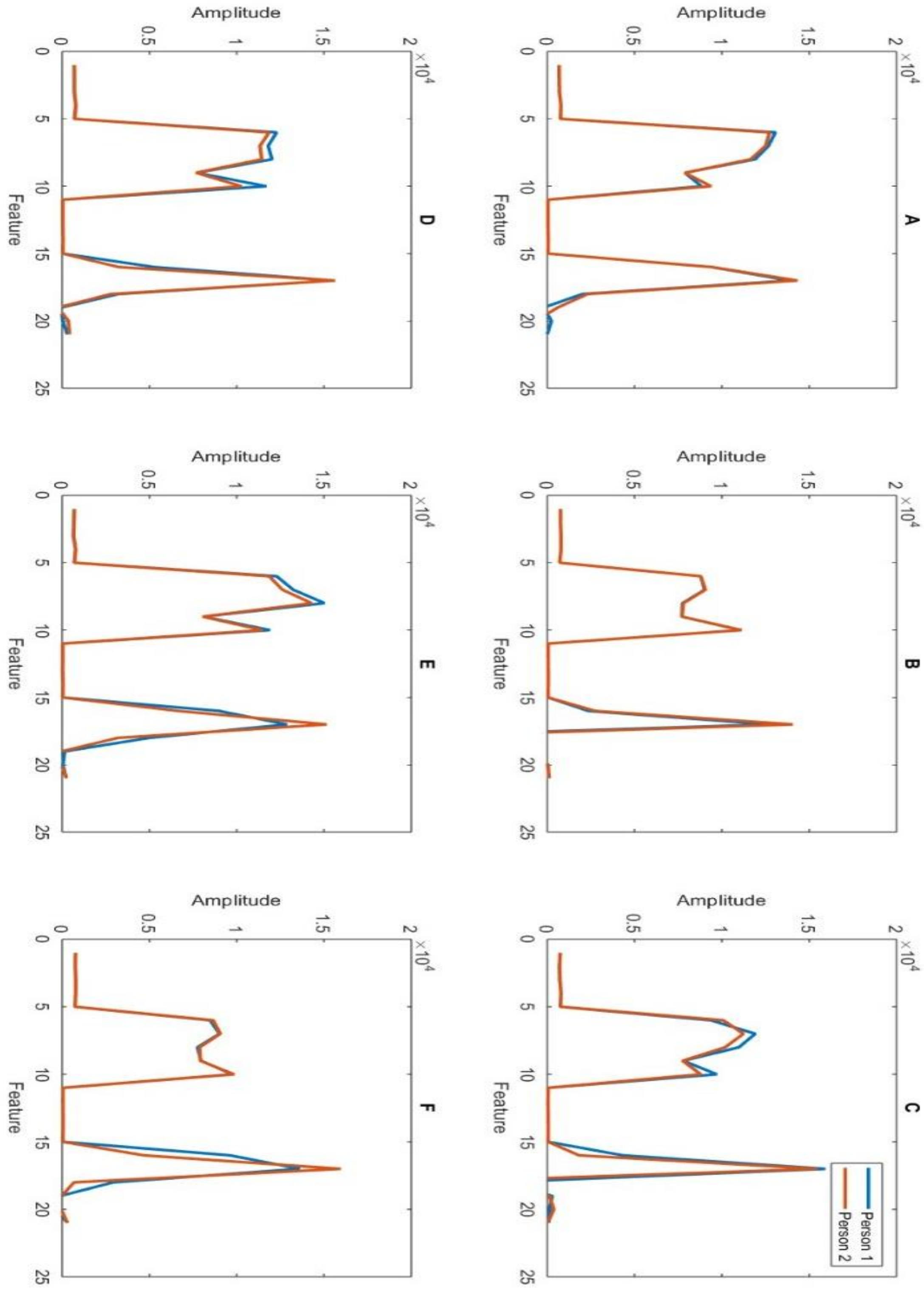
Flex 1	Flex 2	Flex 3	Flex 4	Flex 5	Flex Resistance 1	Flex Resistance 2	Flex Resistance 3	Flex Resistanc e 4	Flex Resistanc e 5	Flex Bending 1	Flex Bending 2	Flex Bending 3	Flex Bending 4	Flex Bending 5	X A	Y A	Z A	X G	Y G	Z A	Gesture
693	679	672	778	756	13058.04	12665.69	11904.76	7872.75	8829.36	34	35	37	49	46	9408	13900	2012	-208	230	-6	G
698	682	678	778	744	12721.24	12500	11640.4	7872.75	9375	35	36	38	49	45	9436	14320	2308	597	-676	-223	G
698	684	686	778	745	12281.34	12390.35	11640.4	7872.75	9328.86	36	36	38	49	45	8768	15092	1548	-10	-50	301	G
781	752	757	782	709	8784.68	9009.31	7746.48	7704.6	11071.93	46	46	49	50	40	2352	12740	-11512	-238	16	106	H
780	751	757	782	708	8784.68	9054.6	7788.46	7704.6	11122.88	46	46	49	50	40	2748	14060	-10032	-757	52	69	H
771	744	760	783	705	8651.32	9375	8171.21	7662.84	11276.6	47	45	48	50	39	3692	16168	-3688	-94	23	243	H
711	693	744	780	737	9375	11904.76	10970.46	7788.46	9701.49	45	37	40	49	44	4320	15928	-2924	281	119	82	I
727	706	729	781	757	10082.3	11225.21	10178.82	7746.48	8784.68	43	39	42	49	46	1808	15472	-7152	121	374	19	I
734	712	729	781	753	10082.3	10919.94	9843.33	7746.48	8964.14	43	40	43	49	46	3476	16340	-2408	-138	-91	284	I
691	695	686	782	697	12281.34	11798.56	12011.58	7704.6	11692.97	36	38	37	50	38	5280	15552	3224	-39	39	296	J
702	704	694	783	725	11851.58	11328.12	11431.62	7662.84	10275.86	38	39	39	50	42	3240	15628	2800	-235	355	420	J
708	709	696	783	714	11745.69	11071.93	11122.68	7662.84	10819.33	38	40	40	50	41	2448	15732	3844	-119	205	226	J
639	669	686	773	693	12281.34	13228.7	15023.47	8085.38	11904.76	36	34	28	48	37	8984	12868	4984	139	44	238	K
651	680	694	774	701	11851.58	12610.3	14285.71	8042.64	11483.59	38	35	31	49	39	6664	15144	3184	-25	-34	244	K
661	686	704	775	703	11328.12	12281.34	13691.38	8000	11379.8	39	36	32	49	39	5252	15916	528	-71	-12	197	K
781	752	764	776	736	8475.13	9009.31	7746.48	7957.47	9748.64	47	46	49	49	44	9612	13604	2924	-80	-349	311	L
778	751	760	777	734	8651.32	9054.6	7872.75	7915.06	9843.33	47	46	49	49	43	4692	15952	684	-29	-44	292	L
777	752	759	777	737	8695.65	9009.31	7915.06	7915.06	9701.49	47	46	49	49	44	4060	16272	-700	-521	11	487	L

Flex 1	Flex 2	Flex 3	Flex 4	Flex 5	Flex Resistance 1	Flex Resistance 2	Flex Resistance 3	Flex Resistanc e 4	Flex Resistanc e 5	Flex Bending 1	Flex Bending 2	Flex Bending 3	Flex Bending 4	Flex Bending 5	X A	Y A	Z A	X G	Y G	Z A	Gesture
672	679	693	778	756	13058.04	12665.69	11904.76	7872.75	8829.36	34	35	37	49	46	9408	13900	2012	-208	230	-6	M
678	682	698	778	744	12721.24	12500	11640.4	7872.75	9375	35	36	38	49	45	9436	14320	2308	597	-676	-223	M
686	684	698	778	745	12281.34	12390.35	11640.4	7872.75	9328.86	36	36	38	49	45	8768	15092	1548	-10	-50	301	M
757	752	781	782	709	8784.68	9009.31	7746.48	7704.6	11071.93	46	46	49	50	40	2352	12740	-11512	-238	16	106	N
757	751	780	782	708	8784.68	9054.6	7788.46	7704.6	11122.88	46	46	49	50	40	2748	14060	-10032	-757	52	69	N
760	744	771	783	705	8651.32	9375	8171.21	7662.84	11276.6	47	45	48	50	39	3692	16168	-3688	-94	23	243	N
744	693	711	780	737	9375	11904.76	10970.46	7788.46	9701.49	45	37	40	49	44	4320	15928	-2924	281	119	82	O
729	706	727	781	757	10082.3	11225.21	10178.82	7746.48	8784.68	43	39	42	49	46	1808	15472	-7152	121	374	19	O
729	712	734	781	753	10082.3	10919.94	9843.33	7746.48	8964.14	43	40	43	49	46	3476	16340	-2408	-138	-91	284	O
686	695	691	782	697	12281.34	11798.56	12011.58	7704.6	11692.97	36	38	37	50	38	5280	15552	3224	-39	39	296	P
694	704	702	783	725	11851.58	11328.12	11431.62	7662.84	10275.86	38	39	39	50	42	3240	15628	2800	-235	355	420	P
696	709	708	783	714	11745.69	11071.93	11122.68	7662.84	10819.33	38	40	40	50	41	2448	15732	3844	-119	205	226	P
686	669	639	773	693	12281.34	13228.7	15023.47	8085.38	11904.76	36	34	28	48	37	8984	12868	4984	139	44	238	Q
694	680	651	774	701	11851.58	12610.3	14285.71	8042.64	11483.59	38	35	31	49	39	6664	15144	3184	-25	-34	244	Q
704	686	661	775	703	11328.12	12281.34	13691.38	8000	11379.8	39	36	32	49	39	5252	15916	528	-71	-12	197	Q
764	752	781	776	736	8475.13	9009.31	7746.48	7957.47	9748.64	47	46	49	49	44	9612	13604	2924	-80	-349	311	R
760	751	778	777	734	8651.32	9054.6	7872.75	7915.06	9843.33	47	46	49	49	43	4692	15952	684	-29	-44	292	R
759	752	777	777	737	8695.65	9009.31	7915.06	7915.06	9701.49	47	46	49	49	44	4060	16272	-700	-521	11	487	R

Flex 1	Flex 2	Flex 3	Flex 4	Flex 5	Flex Resistance 1	Flex Resistance 2	Flex Resistance 3	Flex Resistance 4	Flex Resistance 5	Flex Bending 1	Flex Bending 2	Flex Bending 3	Flex Bending 4	Flex Bending 5	X A	Y A	Z A	X G	Y G	Z A	Gesture
672	679	693	778	756	13058.04	12665.69	11904.76	7872.75	8829.36	34	35	37	49	46	9408	13900	2012	-208	230	-6	S
678	682	698	778	744	12721.24	12500	11640.4	7872.75	9375	35	36	38	49	45	9436	14320	2308	597	-676	-223	S
686	684	698	778	745	12281.34	12390.35	11640.4	7872.75	9328.86	36	36	38	49	45	8768	15092	1548	-10	-50	301	S
757	752	781	782	709	8784.68	9009.31	7746.48	7704.6	11071.93	46	46	49	50	40	2352	12740	-11512	-238	16	106	T
757	751	780	782	708	8784.68	9054.6	7788.46	7704.6	11122.88	46	46	49	50	40	2748	14060	-10032	-757	52	69	T
760	744	771	783	705	8651.32	9375	8171.21	7662.84	11276.6	47	45	48	50	39	3692	16168	-3688	-94	23	243	T
744	693	711	780	737	9375	11904.76	10970.46	7788.46	9701.49	45	37	40	49	44	4320	15928	-2924	281	119	82	U
729	706	727	781	757	10082.3	11225.21	10178.82	7746.48	8784.68	43	39	42	49	46	1808	15472	-7152	121	374	19	U
729	712	734	781	753	10082.3	10919.94	9843.33	7746.48	8964.14	43	40	43	49	46	3476	16340	-2408	-138	-91	284	U
686	695	691	782	697	12281.34	11798.56	12011.58	7704.6	11692.97	36	38	37	50	38	5280	15552	3224	-39	39	296	V
694	704	702	783	725	11851.58	11328.12	11431.62	7662.84	10275.86	38	39	39	50	42	3240	15628	2800	-235	355	420	V
696	709	708	783	714	11745.69	11071.93	11122.68	7662.84	10819.33	38	40	40	50	41	2448	15732	3844	-119	205	226	V
686	669	639	773	693	12281.34	13228.7	15023.47	8085.38	11904.76	36	34	28	48	37	8984	12868	4984	139	44	238	W
694	680	651	774	701	11851.58	12610.3	14285.71	8042.64	11483.59	38	35	31	49	39	6664	15144	3184	-25	-34	244	W
704	686	661	775	703	11328.12	12281.34	13691.38	8000	11379.8	39	36	32	49	39	5252	15916	528	-71	-12	197	W
764	752	781	776	736	8475.13	9009.31	7746.48	7957.47	9748.64	47	46	49	49	44	9612	13604	2924	-80	-349	311	X
760	751	778	777	734	8651.32	9054.6	7872.75	7915.06	9843.33	47	46	49	49	43	4692	15952	684	-29	-44	292	X
759	752	777	777	737	8695.65	9009.31	7915.06	7915.06	9701.49	47	46	49	49	44	4060	16272	-700	-521	11	487	X

Flex 1	Flex 2	Flex 3	Flex 4	Flex 5	Flex Resistance 1	Flex Resistance 2	Flex Resistance 3	Flex Resistance 4	Flex Resistance 5	Flex Bending 1	Flex Bending 2	Flex Bending 3	Flex Bending 4	Flex Bending 5	X A	Y A	Z A	X G	Y G	Z A	Gesture
672	679	693	778	756	13058.04	12665.69	11904.76	7872.75	8829.36	34	35	37	49	46	9408	13900	2012	-208	230	-6	Y
678	682	698	778	744	12721.24	12500	11640.4	7872.75	9375	35	36	38	49	45	9436	14320	2308	597	-676	-223	Y
686	684	698	778	745	12281.34	12390.35	11640.4	7872.75	9328.86	36	36	38	49	45	8768	15092	1548	-10	-50	301	Y
757	752	781	782	709	8784.68	9009.31	7746.48	7704.6	11071.93	46	46	49	50	40	2352	12740	-11512	-238	16	106	Z
757	751	780	782	708	8784.68	9054.6	7788.46	7704.6	11122.88	46	46	49	50	40	2748	14060	-10032	-757	52	69	Z
760	744	771	783	705	8651.32	9375	8171.21	7662.84	11276.6	47	45	48	50	39	3692	16168	-3688	-94	23	243	Z

EXTRACTED FEATURES OF THE GESTURES



CHAPTER 11

CONCLUSION

The project on HAND TALK ASSISTING SYSTEM FOR SPEECH CHALLENGED PEOPLE is working fine, getting the parameter envisaged during the conceptual stage.

During the design, as well as during the construction, greater care has been put into avoid hiccups at the final stage. The PCB layout were prepared with almost care to incorporate the circuits in a modular manner. The circuit is made as simple as to our knowledge. Also, components were selected keeping in mind their availability and cost.

It was a very interesting process of developing the prototype, stage by stage and testing the same. We have to go through fairly large pages of data related to the components etc. It was a useful and fulfilling assignment to get the project completed in time. This gave us a sense of satisfaction and accomplishment.

CHAPTER 12
REFERENCES

1. Hand Talk Assistive System A.A.Markad1 , A.L.Borkar2 1 Student, 2 Asst Professor, Department of Electronics and Telecommunication, M.S.S.C.E.T., Jalna, Maharashtra, (India)
2. Design And Implementation of a Sign-To-Speech/Text System For Deaf and Dump People. Dalal Abdulla, Shahrazad Abdulla and Rameesa Manaf, University of Sharjah.
3. Sensor Based Hand Gesture Recognition System for English Alphabets Used in Sign Language of Deafmute People Abhishek B. Jani1 (Member, Ieee), Nishith A. Kotak2 (Member, Ieee) And Anil K. Roy1 (Senior Member, Ieee) 1 Dhirubhai Ambani Institute of Information and Communication Technology, Gandhinagar 2 Marwadi University, Rajkot Gujarat, India.
4. Embedded Sign Language Interpreter System for Deaf And Dumb People .Geethu G Nath Department of Electronics and Communication Sree Buddha College of Engineering, Pattoor Kerala, India
5. E. Torunski, A. E. Saddik and E. Petriu, "Gesture Recognition on A Mobile Device for Remote Event Generation", Proc. Ieee Int. Conf. Multimedia and Expo (Icme), Pp. 1-6, Jul. 2011.
6. X. Zhang, X. Chen, Y. Li, V. Lantz, K. Wang and J. Yang, "A Framework for Hand Gesture Recognition Based on Accelerometer And Emg Sensors", Ieee Transactions on Systems Man and Cybernetics Part A, Vol. 41, No. 6, Pp. 1064-1076, 2011.
7. R. Hamzah and M. I. M. Fadzil, "Voice4blind: The Talking Braille Keyboard to Assist the Visual Impaired Users in Text Messaging", 2016 4th International Conference on User Science and Engineering (I-User), Pp. 265-270, 2016.
8. Karthick, Pratibha, V.B.Rekha, S.Thanalaxmi, "Transforming Indian Sign Language into Text Using Leap Motion", International Journal of Innovative

Research in Science, Engineering and Technology, Pp. 10906-10908,2014

9. Leigh Ellen Potter, Jake Araullo,Lewis Carter,” The Leap Motion Controller: A View on Sign Language”, Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collabration.Pp.1-4,2013

10. Paulo Trigueiros, Ferando Ribeiro, Luis Paulo Reis, “ Vision Based Portuguese Sign Language Recognition System”, New Perspectives in Information System and Technologies,Vollume 1,Advances in Intelligent System and Computing, Springer International Publishing Switzerland,Pp-605-608,2014

11. Cao Dong, Ming C.Leu, Zhaozheng Yin, “ American Sign Language Alphabet Recognition Using Microsoft Kinect”, Computer Vision and Pattern Recognition Workshop, Ieee Conference, Pp: ,2015.

12. Aarthi M And Vijayalakshmi P, “Sign Language to Speech Conversion”, Fifth International Conference on Recent Trends in Information Technology 2016,India.

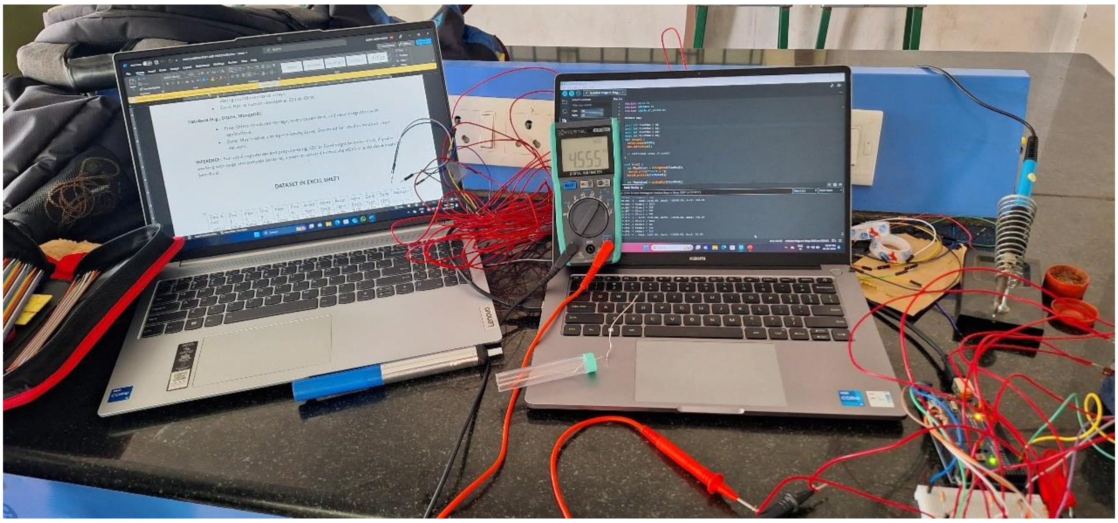
13. Sarth Pham The Hai, Huynh Chau Thinh, Bui Van Phuc, Ha Hoang Kha, "Automatic Feature Extraction for Vietnamese Sign Language Recognition Using Support Vector Machine", Recent Advances in Signal Processing Telecommunications & Computing (Sigtelcom) 2018 2nd International Conference On, Pp. 146-151, 2018

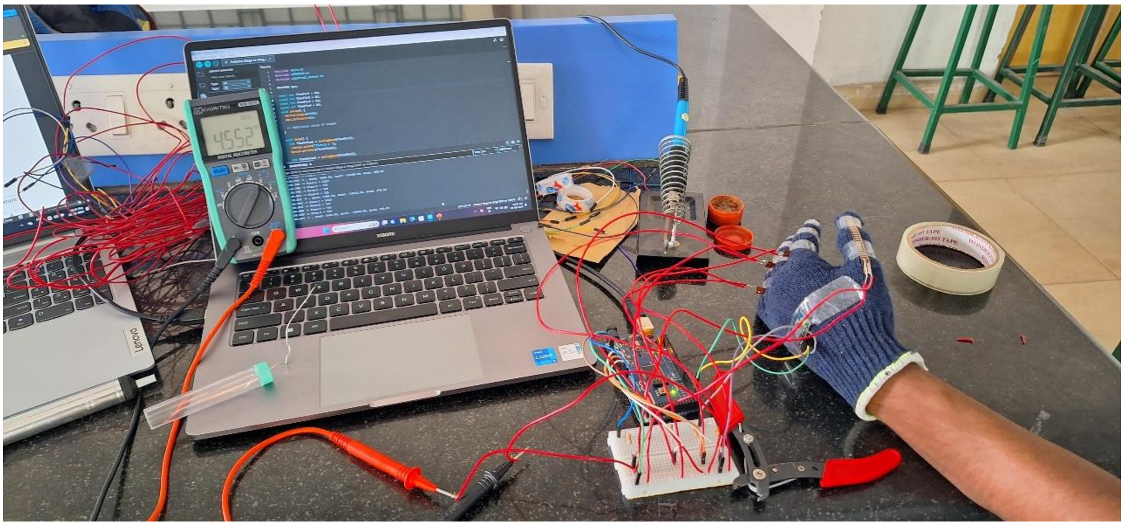
14. ‘‘Malay Sign Language Gesture Recognition System ”Tan Tian Swee, Sh-Hussain Salleh, A.K. Ariff, Chee-Ming Ting, Siew Kean Seng, And Leong Seng Huat, ‘‘International Conference on Intelligent and Advanced Systems 2007”

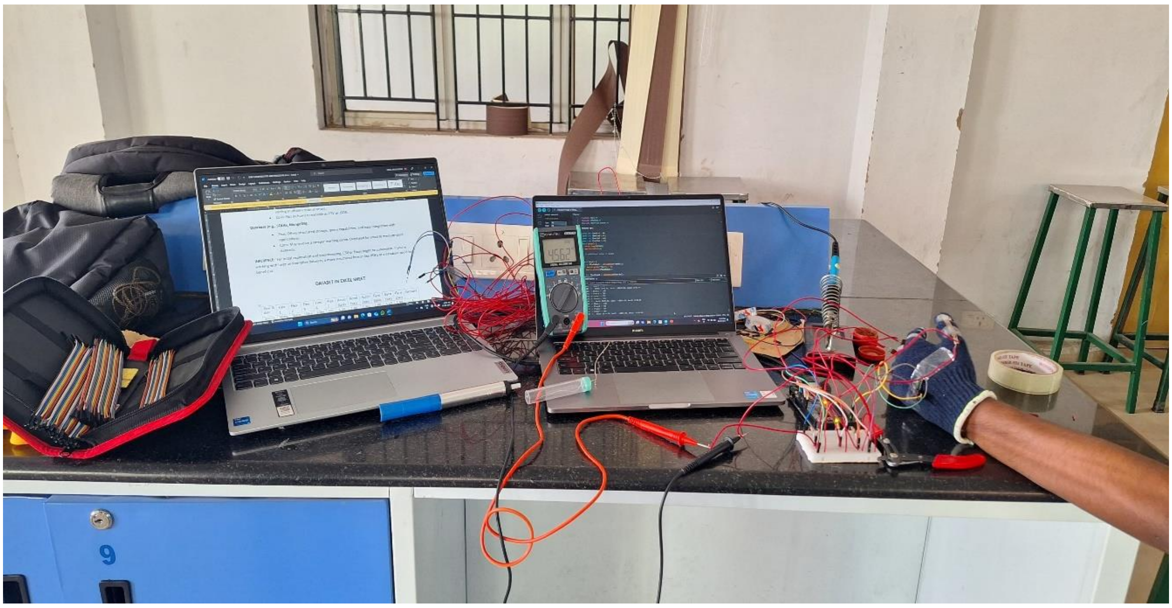
15. Sign languages to speech conversion prototype using the SVM classifier MM Chandra, S Rajkumar, LS Kumar - TENCON 2019-2019 IEEE Region 10 Conference …, 2019

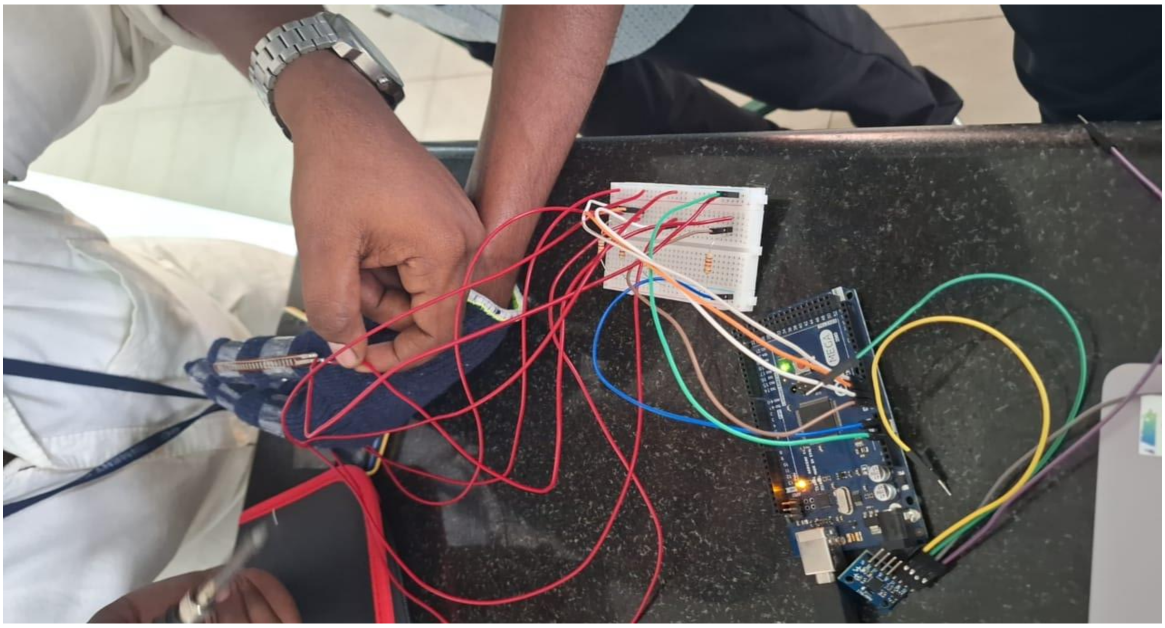
CHAPTER 13

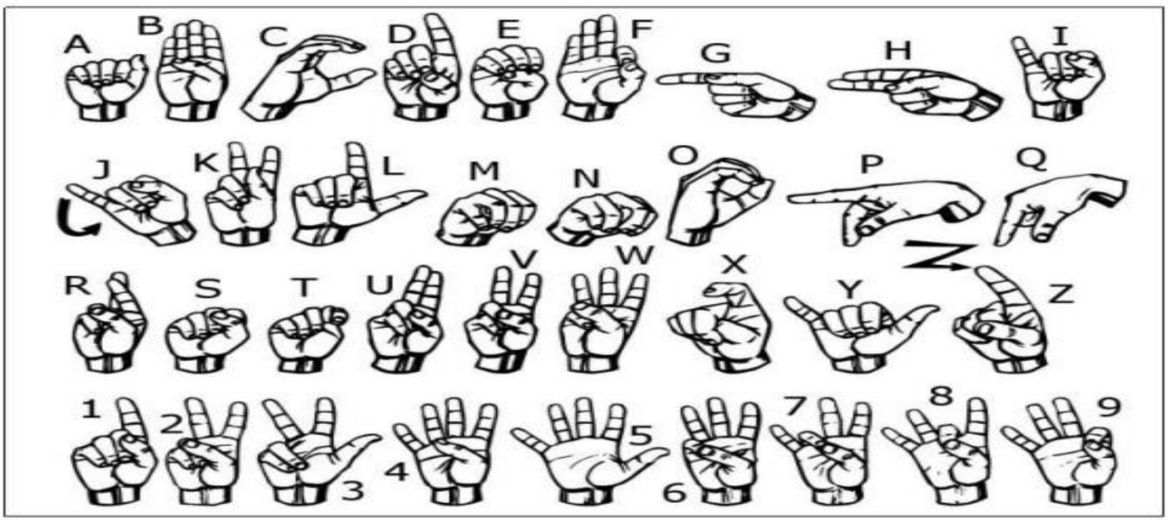
GALLERY










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 Z
1 2 3 4 5 6 7 8 9