SIGN LANGUAGE TRANSLATOR FOR DEAF AND DUMB USING GLOVE

PROJECT REPORT

Submitted in partial fulfillment of the requirements
for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

Submitted by

G. SANTHOSH KUMAR	15JN1A0521
P. TANVI SATHWIK	15JN1A0557
A. KRISHNA PRANAV	15JN1A0503
K. RAJESH	15JN1A0538

Under the Guidance of
Mr.B.ASHOK KUMAR, M.Tech
Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SREE VENKATESWARA COLLEGE OF ENGINEERING
KODAVALURU, NELLORE DISTRICT, AP-524316

Jawaharlal Nehru Technological University, Ananthapur, AP, India

APRIL 2019

SREE VENKATESWARA COLLEGE OF ENGINEERING

NELLORE

BONAFIDE CERTIFICATE

Certified that this project report "**SIGN LANGUAGE TRANSLATOR FOR DEAF AND DUMB USING GLOVE**" is the bonafide work done by "**G.SANTHOSH KUMAR (15JN1A0521), P.TANVI SATHWIK (15JN1A0557), A.KRISHNA PRANAV (15JN1A0503), K.RAJESH (15JN1A0538)**", who carried out the project under my guidance during the year 2018-19, towards partial fulfillment of the requirements of the Degree of Bachelor of Technology in Computer Science & Engineering from Jawaharlal Nehru Technological University, Anantapur. The results embodied in this report have not been submitted to any other University for the award of any degree.

HEAD OF THE DEPARTMENT
Department of CSE

PROJECT GUIDE
Department of CSE

External Viva voce conducted on _____

Internal Examiner

External Examiner

SREE VENKATESWARA COLLEGE OF ENGINEERING

NELLORE

CERTIFICATE OF AUTHENTICATION

We solemnly declare that this project report **"SIGN LANGUAGE TRANSLATOR FOR DEAF AND DUMB USING GLOVE"** is the bonafide work done purely by us, carried out under the supervision of Mr. **B. ASHOK KUMAR**, towards partial fulfillment of the requirements of the Degree of **BACHELOR OF TECHNOLOGY** in **Computer Science & Engineering** from Jawaharlal Nehru Technological University, Ananthapur during the year 2018-19.

It is further certified that this work has not been submitted, either in part of in full, to any other department of the Jawaharlal Nehru Technological University, or any other University, institution or elsewhere, or for publication in any form.

Signature of the Students

Date:

1. G. SANTHOSH KUMAR (15JN1A0521)
2. P. TANVI SATHWIK (15JN1A0557)
3. A. KRISHNA PRANAV (15JN1A0503)
4. K. RAJESH (15JN1A0538)

ACKNOWLEDGMENT

Our most and sincere acknowledgements to **Dr. P.BABU NAIDU,** Chairman who took keen interest and encouraged us in every effort throughout this course.

We owe our gratitude to the **Dr.S.V.PADMAJA RANI**, Principal, **SREE VENKATESWARA COLLEGE OF ENGINEERING,** NELLORE for giving us the opportunity to fulfill our aspirations and become engineers.

We would like to extend ardent thanks to **Mr. T.DAYAKAR**, Head of the Department, Computer Science & Engineering, for endowing a practical environment in the institute.

We take this opportunity to express my sincere deep sense of gratitude to our Coordinator **Mr. M.KIRAN KUMAR**, Assistant Professor, Department of Computer Science & Engineering for his significant suggestions and help in every respect to accomplish the seminar report.

Mr. B.ASHOK KUMAR, Assistant Professor, Department of Computer Science Engineering, his persisting encouragement, everlasting patience and keen interest in discussions have benefited us to an extent that cannot be spanned by words. It is more than words which speak the way her involvement has generated interest and improved our confidence to face the challenges encountered in completing the seminar before the course of duration.

We are thankful to the technical and non technical staff of **SREE VENKATESWARA COLLEGE OF ENGINEERING**, NELLORE, and also our parents, friends and all my well wishers for their assistance in finishing the project successfully.

PROJECT ASSOCIATES

1. G. SANTHOSH KUMAR (15JN1A0521)
2. P. TANVI SATHWIK (15JN1A0557)
3. A. KRISHNA PRANAV (15JN1A0503)
4. K. RAJESH (15JN1A0538)

ABSTRACT

Communication between speakers and non-speakers of Sign Language can be problematic, inconvenient, and expensive. Sign Language translator attempts to bridge the communication gap by designing a portable glove that captures the user's gestures and outputs the translated text on a smartphone. Glove is equipped with flex sensors to measure the flexion of the fingers of the hand. The glove's Arduino Nano microcontroller analyses the sensor readings to identify the gesture from a library of learned gestures. The Bluetooth module transmits the gesture to a smartphone. Using this device, one day speakers of able to communicate with others in an affordable and convenient way.

LIST OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Today, it is estimated that the number of users Sign Language falls between 250,000 and 500,000 Americans. There are only about 250 certified sign language interpreters in India, translating for a deaf population of 7 million. (The wide range in population estimates exists because the Indian census doesn't track the number of deaf people — instead, it documents an aggregate number of people with disabilities.)On a large scale, a truly international sign language does not exist; however, American Sign Language (ASL) is one of the most popular sign languages in the world. Despite such a large population of people who speak ASL, there are few people outside of native speakers who can understand it .The ASL community not only shares a common signed language, but also its own unique culture . Unfortunately, this language and culture often do not transcend into the hearing world, causing a lack of deeper communication and understanding between ASL users and non-users.

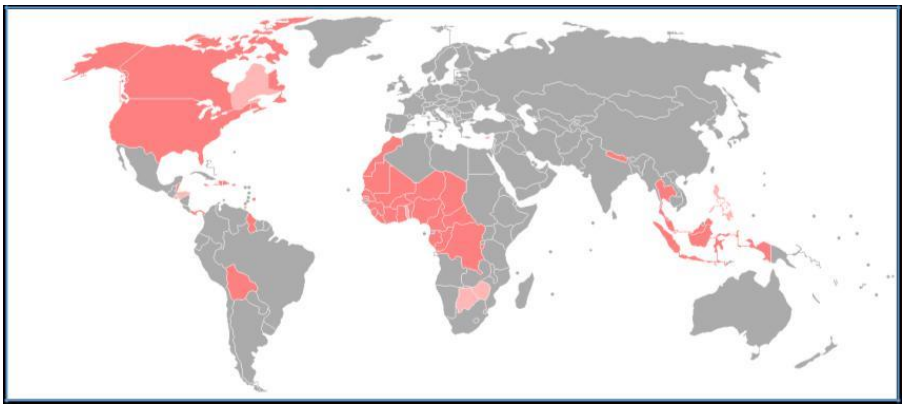

Figure 1.1: Statistics of World

Sign language is a way by which the gestures made by the user are used for communication. Human gestures are an efficient and powerful way of interaction. These are sometimes used to express ourselves.

Sign language translator focus on developing a help for disabled people using this gesture recognition technique. The gestures are converted into text messages and sound for communication. The basic concept involves the use of data gloves worn by disabled people. These gloves are designed using Flex sensors. The flex sensors are normally attached to the glove.

The team developed a sensor network that differentiates various types of hand movement used in American Sign Language (ASL), interfaced the sensor network with a microcontroller, designed a Circuit Board (CB) that serves as the bridge between hardware and software components, developed a program that recognizes sensor signals and stores those that correspond to a designated ASL sign into a code library, and successfully outputted two ASL signs through a Bluetooth module to a smartphone. A high level diagram of the system is illustrated below in Figure.

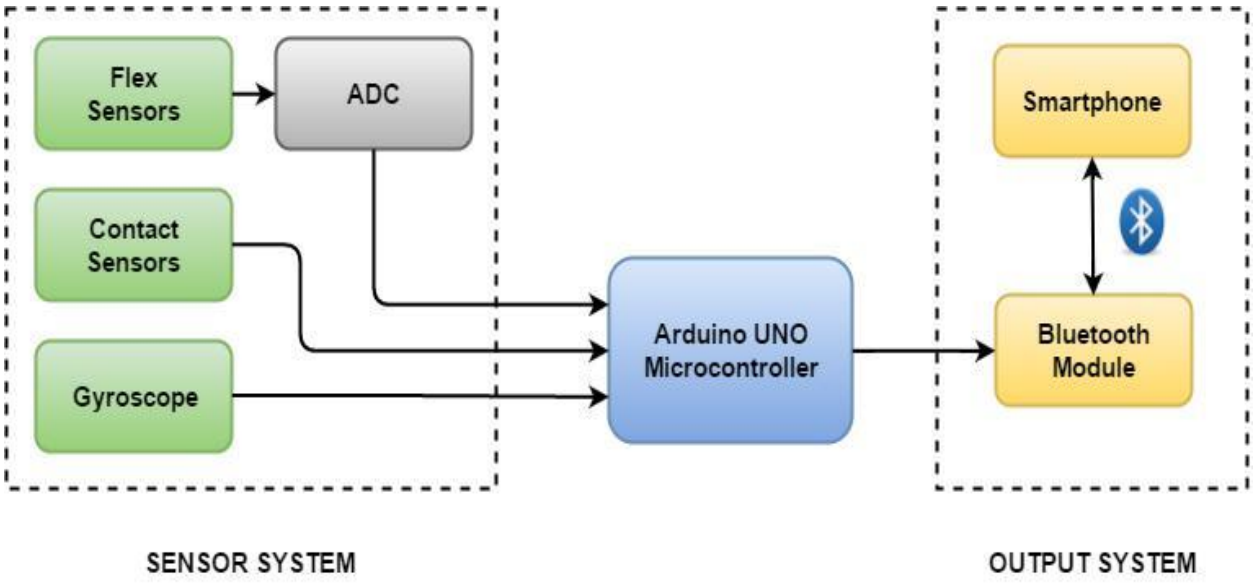

Figure 1.2: High-level diagram of glove system

After having completed the design of the glove, the team ran performance tests to ensure that it met our preliminary design specifications and development objectives. The average runtime of the classification algorithm is calculated to be 0.506 milliseconds.

Bluetooth to a smartphone application, making the glove portable and not dependent on any external devices. This realization accomplishes one of the project's primary goals, which is to design a portable and compact glove that can be used on a daily basis. All essential components to both signal communication once the Arduino software and glove hardware were able to communicate and exhibit the desired functionality, the team was able to test them for accuracy.

In the future, there are several changes that can be made to improve the glove into a well-rounded product. To help shape the device into something more slim and comfortable, the team suggests a few improvements to the prototype. Primarily, the team suggests designing a smaller and more compact PCB, which will combine the components of the system together more efficiently. The team also recommends using conductive fabric to replace, the contact sensors, the flex sensors, and/or the wiring to allow for a more lightweight glove that is less bulky and more easily conforms to hand movement. Another valued improvement to the prototype would be replacement batteries which are slimmer but can still provide the required voltage of 5V and enough power to last for at least few hours.

Research and emerging technologies are attempting to bridge this communication gap between the hearing and non-hearing, in order to make every day interaction in public places more effective. Such projects include the EnableTalk Project and the Sign Language Glove from Cornell. These devices, however, are inadequate in their ability to incorporate the full range of motion that the language requires, including wrist and arm movements. They instead limit their interpretation abilities to just the ASL alphabet, requiring users to spell out each word letter by letter. Such a method is both cumbersome and impractical, especially since the language is based on signing topics and words instead of individual signs for every letter in a word.

This project explores a solution through the development of a prototype device to help practically bridge the communication gap between hearing and non-hearing people in a way that improves on the methods of pre-existing designs.

The team developed a glove with various electronic sensors to sense the flexing, rotation, and contact of various parts of the hands and wrist, as used in ASL signs. The meaning of each sign will be communicated in a way that is perceivable to a person who does not understand ASL. The creation of this device involves an understanding of both electrical engineering (wiring, circuitry, and understanding of sensor implementation and voltage signals) and computer engineering (sensor signal processing as well as computation and output of the device's microcontroller).

Speech synthesis is the artificial production of human speech. A computer system used for this purpose is called a speech computer or speech synthesizer, and can be implemented in software or hardware products. A **text-to-speech** (**TTS**) system converts normal language text into speech, other systems render symbolic linguistic representations like phonetic transcriptions into speech.

There seems to be other devices and patents that aim to do the same process of translation of sign language into speech. In most of these cases it seems that the inventors augment the device with several more sensors than what we use, but we hold no claim of originality.

A sign language editing apparatus includes a glove-type sensor for converting movement of fingers in the sign language into an electrical signal to produce time series data of sign language words, a sign language word data editing device for adding predetermined additional data to the time series data inputted from the glove-type sensor to process the time series data, a sign language sentence data editing device for reading out the time series data of sign language words stored in the sign language word dictionary in accordance with the predetermined characters inputted from the input unit and adding predetermined additional information to the time series data of sign language words to produce time series data of sign language sentence, a sign language animation synthesizing device inputted with the time series data of sign language sentence produced by the sign language sentence data editing device to produce sign language animation expressing movement of the sign language.

When it comes to the problem of translating the sign language gestures, it is intuitive that researchers looked to gloves, as sign languages involve the extensive use of hand movement . Patent 5,047,952 in 1981 uses the concept of an “instrumented glove”, but it was not used for sign language. Almost two decades later, the idea of an instrumented glove was designed for sign language translation.

Although this patent was filed in 1999, no further information can be found on the implementation of the patent. A few other patents from some of the same patent holders are on record, which suggests other applications for the technology, such as for video learning software to learn finger spelling. The predecessor to this was patent, No. 4414537 from 1981 which covers much of the same design of an “instrumented glove”, except with more technical detail. However this predecessor was not specifically designed as a communication device.

The Enable Talk project was a student project completed for the Microsoft Imagine Cup in 2012. Its system consisted of a mobile device and a pair of gloves which translate the ASL alphabet into text on a mobile device. Each glove was outfitted with contact sensors, flex sensors, and a microcontroller.

Although the website hints at further development, it has not been updated in several years .As far as can be found, this product is not on the market for ASL users and does not have a purchasable price. The glove demonstrates that the use of a glove outfitted with sensors can be used for ASL translation.

AUS Patent filed under the title “Communication system for deaf, deaf-blind, or non-vocal individuals using instrumented glove” is one of the first attempts found to use sensors in a glove to facilitate communication between hearing and non-hearing people.

2. SYSTEM ANALYSIS

2.1 Technologies Used

2.1.1 Arduino

Arduino is an open-source hardware and software company, project and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices and interactive objects that can sense and control both physically and digitally. Permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially in preassembled form or as do-it-yourself (DIY) kits.

Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards or breadboards (shields) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers. The microcontrollers are typically programmed using a dialect of features from the programming languages C and C++. In addition to using traditional compiler tool chains, the Arduino project provides an integrated development environment (IDE) based on the Processing language project.

2.1.2 C++ Language

C++ is a general-purpose programming language that was developed as an extension of the C language, or "C with Classes". It has imperative, object-oriented and generic programming features, while also providing facilities for low-level memory manipulation. It is almost always implemented as a compiled language, and many vendors provide C++ compilers, including the Free Software Foundation, Microsoft, Intel, and IBM, so it is available on many platforms.

It was designed with a bias toward system programming and embedded, resource-constrained software and large systems, with performance, efficiency and flexibility of use as its design highlights. C++ has also been found useful in many other contexts, with key strengths being software infrastructure and resource-constrained applications, including desktop applications, servers (e.g. e-commerce, Web search or SQL servers), and performance-critical applications

2.2 Existing System

2.2.1 Gesture recognition

It is a topic in computer science and language technology with the goal of interpreting human gestures via mathematical algorithms. Gestures can originate from any bodily motion or state but commonly originate from the face or hand. Current focuses in the field include emotion recognition from face and hand gesture recognition. Users can use simple gestures to control or interact with devices without physically touching them. Many approaches have been made using cameras and computer vision algorithms to interpret sign language. However, the identification and recognition of posture, gait, proxemics, and human behaviors is also the subject of gesture recognition techniques. Gesture recognition can be seen as a way for computers to begin to understand human body language, thus building a richer bridge between machines and humans than primitive text user interfaces or even GUIs (graphical user interfaces), which still limit the majority of input to keyboard and mouse and interact naturally without any mechanical devices. Using the concept of gesture recognition, it is possible to point a finger at this point will move accordingly. This could make conventional input on devices such and even redundant.

Input Devices:

The ability to track a person's movements and determine what gestures they may be performing can be achieved through various tools. The kinetic user interfaces (KUIs) are an emerging type of user interfaces that allow users to interact with computing devices through the motion of objects and bodies. Examples of KUIs include tangible user interfaces and motion-aware games such as Wii and Microsoft's Kinect, and other interactive projects.

Although there is a large amount of research done in image/video based gesture recognition, there is some variation within the tools and environments used between implementations.

- **Wired gloves**. These can provide input to the computer about the position and rotation of the hands using magnetic or inertial tracking devices. Furthermore, some gloves can detect finger bending with a high degree of accuracy (5-10 degrees), or even provide haptic feedback to the user, which is a simulation of the sense of touch. The first commercially available hand-tracking glove-type device was the DataGlove,a glove-type device which could detect hand position, movement and finger bending.

This uses fiber optic cables running down the back of the hand. Light pulses are created and when the fingers are bent, light leaks through small cracks and the loss is registered, giving an approximation of the hand pose.

- **Depth-aware cameras**. Using specialized cameras such as structured light or time-of-flight cameras, one can generate a depth map of what is being seen through the camera at a short range, and use this data to approximate a 3d representation of what is being seen. These can be effective for detection of hand gestures due to their short range capabilities.
- **Stereo cameras.** Using two cameras whose relations to one another are known, a 3d representation can be approximated by the output of the cameras. To get the cameras' relations, one can use a positioning reference such as a lexian-stripe or infrared emitters.[In combination with direct motion measurement (6D-Vision) gestures can directly be detected.
- **Gesture-based controllers.** These controllers act as an extension of the body so that when gestures are performed, some of their motion can be conveniently captured by software. An example of emerging gesture-based motion capture is through skeletal hand tracking, which is being developed for virtual reality and augmented reality applications.
- An example of this technology is shown by tracking companies uSens and Gestigon, which allow users to interact with their surrounding without controllers.

Another example of this is mouse gesture trackings, where the motion of the mouse is correlated to a symbol being drawn by a person's hand, as is the Wii Remote or the Myo armband or the mForce Wizard wristband, which can study changes in acceleration over time to represent gestures. Devices such as the LG Electronics Magic Wand, the Loop and the Scoop use Hillcrest Labs Free space technology, which uses MEMS accelerometers, gyroscopes and other sensors to translate gestures into cursor movement. The software also compensates for human tremor and inadvertent movement. The sensors of these smart light emitting cubes can be used to sense hands and fingers as well as other objects nearby, and can be used to process data. Most applications are in music and sound synthesis, but can be applied to other fields.

2.2.2 Speech synthesis

It is the artificial production of human speech. A computer system used for this purpose is called a speech computer or speech synthesizer, and can be implemented in software or hardware products. A text-to-speech (TTS) system converts normal language text into speech; other systems render symbolic linguistic representations like phonetic transcriptions into speech.

The quality of a speech synthesizer is judged by its similarity to the human voice and by its ability to be understood clearly. An intelligible text-to-speech program allows people with visual impairments or reading disabilities to listen to written words on a home computer. Many computer operating systems have included speech synthesizers since the early 1990s.

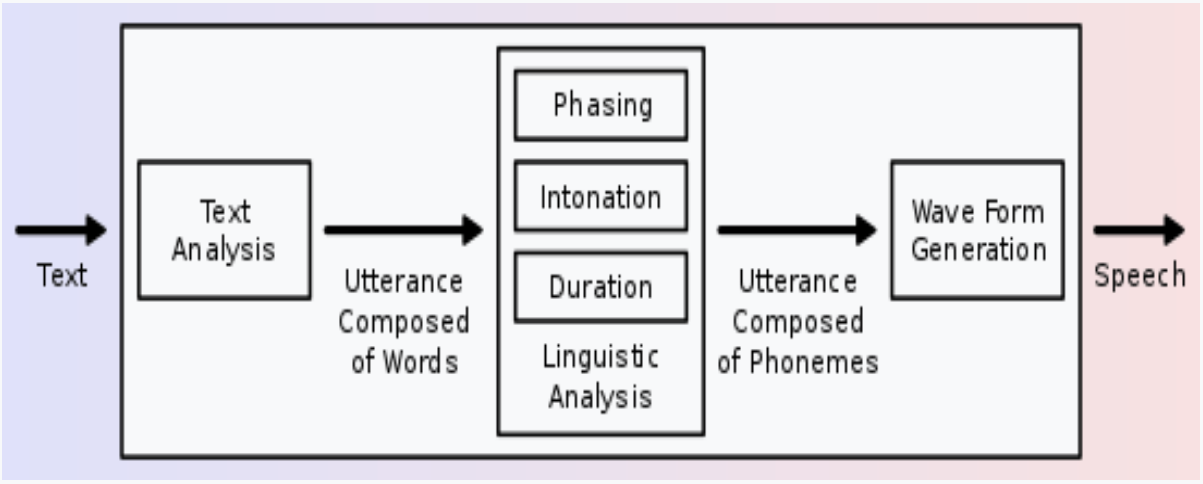

Figure 2.1: Speech Synthesis

A text-to-speech system (or "engine") is composed of two parts a front-end and a back-end. The front-end has two major tasks. First, it converts raw text containing symbols like numbers and abbreviations into the equivalent of written-out words. This process is often called text normalization, pre-processing, or tokenization. The front-end then assigns phonetic transcriptions to each word, and divides and marks the text into prosodic units, like phrases, clauses, and sentences.

The process of assigning phonetic transcriptions to words is called text-to-phoneme or grapheme-to-phoneme conversion. Phonetic transcriptions and prosody information together make up the symbolic linguistic representation that is output by the front-end. The back-end—often referred to as the synthesizer—then converts the symbolic linguistic representation into sound. In certain systems, this part includes the computation of the target prosody (pitch contour, phoneme durations), which is then imposed on the output speech.

2.2.3 Drawbacks

- Ineffective Gesture Recognition.
- Less number of gestures can be used.
- Text Overriding.

2.3 Proposed Systems

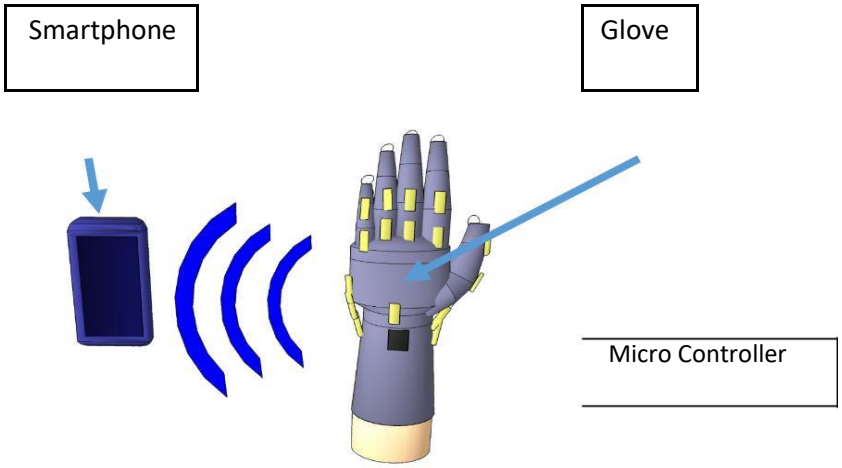

Figure 2.2: Proposed Model

This project is implemented based on the current existing systems like Speech synthesis, Gesture recognisation, Arduino etc. By using these systems this project is done by Gesture recognisation to detect the gesture based upon the input the data is converted to Speech using Text-to-Speech.

Easy to Use - Any complications in its user interface would inhibit the glove's use in everyday life. The user should be able to begin translation without much difficulty or delay. Each translation should be done without any unnecessary button-pressing or other interfacing.

Portable - The system should not be dependent on a computer or other attached system. It should be able to be brought almost anywhere the user goes, with the possible exception for underwater.

Affordable - Not much financial aid is available for assistive devices. This device should be accessible by the average person by practical and affordable means.

Reliable - There is a certain degree of accuracy that the device should maintain. This threshold has been decided by our team to be of about 90% accuracy. If the device does not accurately and consistently translate signs, then the user will resort to time consuming alternatives such as writing on pen and paper and the device will have no use.

Aesthetically Pleasing - For marketability purposes, the device shall be aesthetically pleasing and easily wearable without any factors that hinder convenience during extended periods of time. This includes a smooth, professional appearance without any components that irritate, cut, bruise, or otherwise cause discomfort for the user.

2.3.1 Advantages

- This Device eliminate the barrier in communication between the mute community and the normal people by discarding the need for an interpreter.
- It facilitates effective real time communication.
- The Device is Portable which is easy to carry.
- Fast processing and output of data.

3. REQUIREMENTS SPECIFICATIONS

3.1 Hardware Requirements:

Arduino Nano:

Figure 3.1: Arduino Nano

Arduino Nano is a small, compatible, flexible friendly Microcontroller board which is used for programming with different sensors, developed by Arduino.cc. This board doesn't use standard USB for connection with a computer, instead, it comes with Mini USB support

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.

Memory

The ATmega328 has 32 KB, (also with 2 KB used for the bootloader. The ATmega328 has 2 KB of SRAM and 1 KB of EEPROM.

Input and Output

Each of the 14 digital pins and 7 analog pins on the Nano can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA.

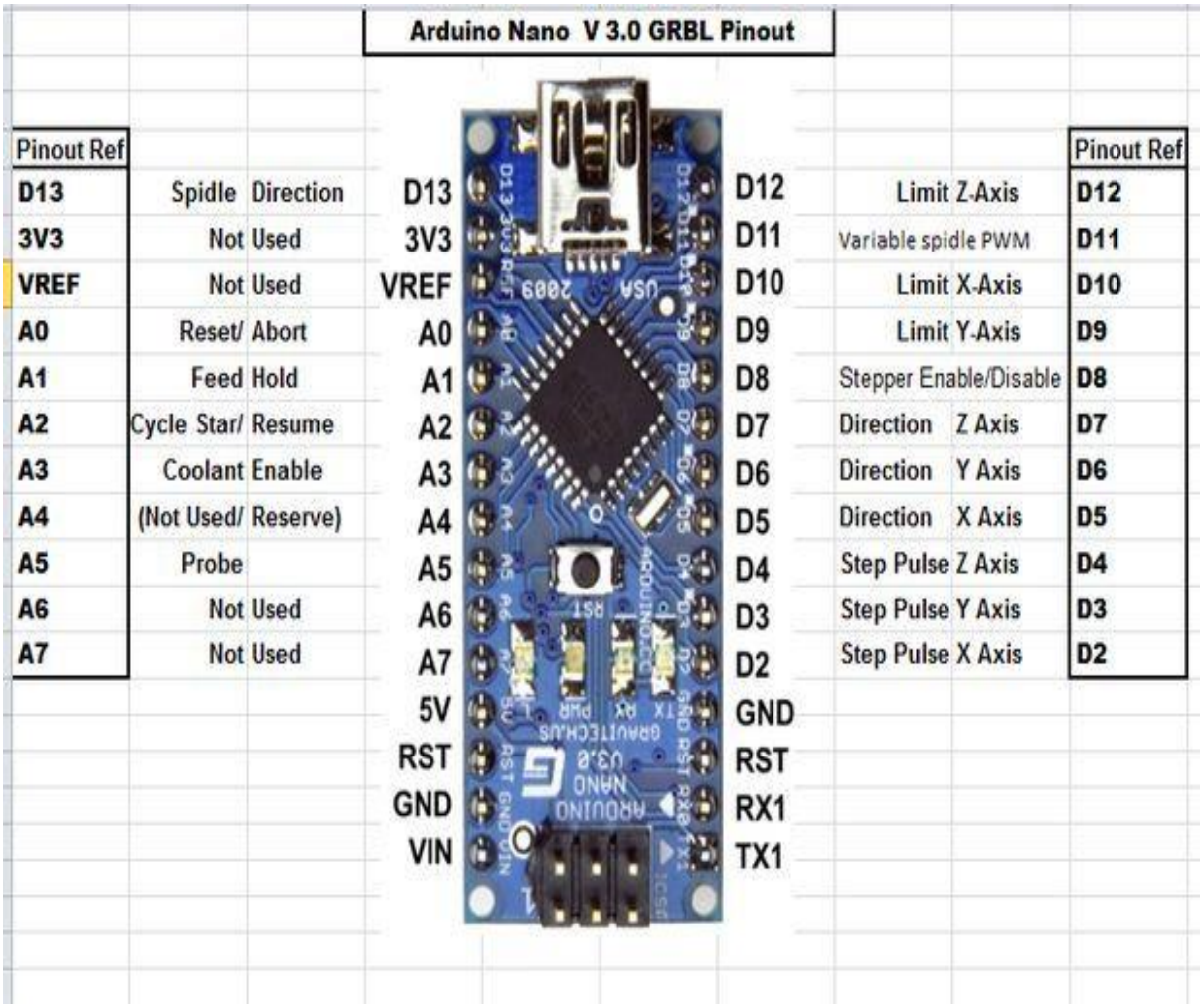

Figure 3.2: Pins of Arduino Nano

Flex Sensors

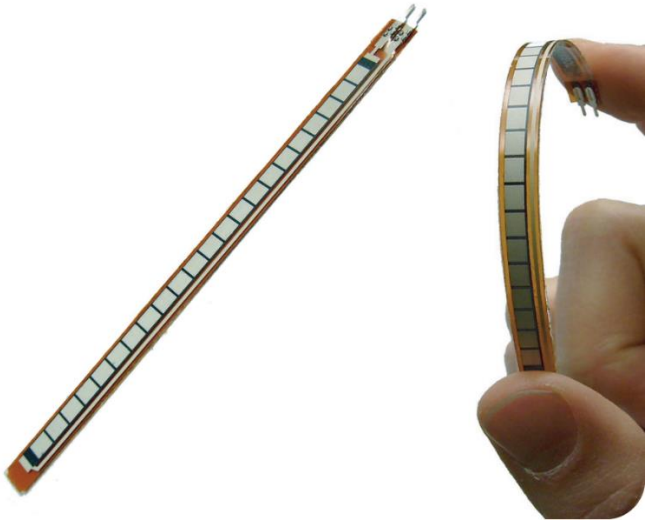

Figure 3.3: Flex Sensor

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

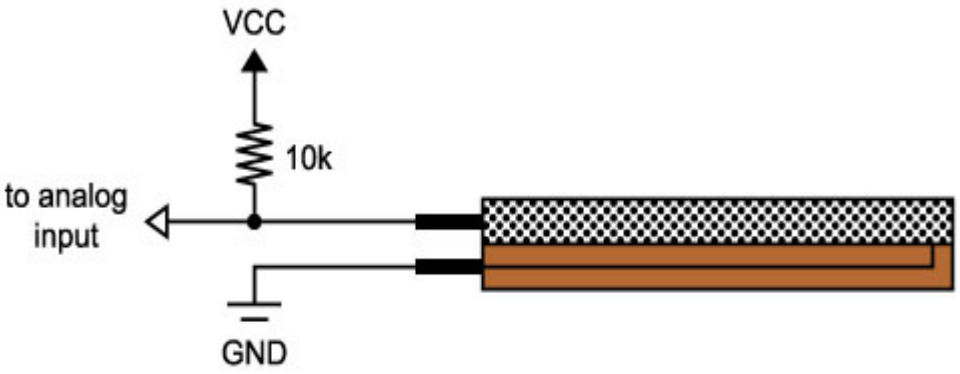

Figure 3.4: Schematics of Flex Sensor

A Flex sensor is resistor, since resistors doesn't have polarity VCC and GND can be connected in any direction. Flex Sensor Values changes according to the resistor used and formulas used.

Bluetooth (HC-05)

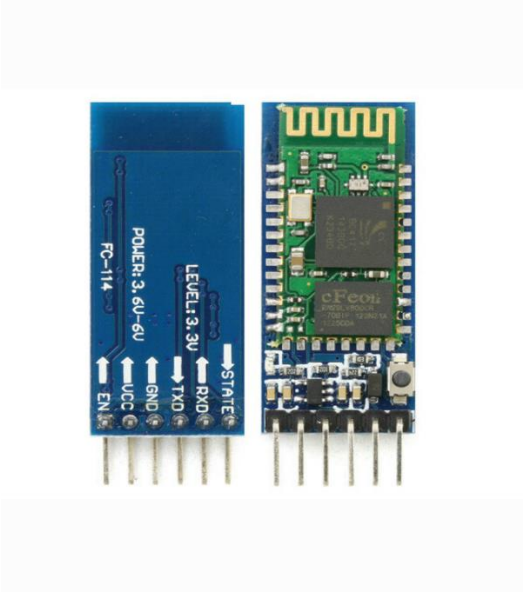

Figure 3.5: Bluetooth

Bluetooth is a technology for wireless communication. It is designed to replace cable connections. It uses serial communication to communicate with devices. It communicates with microcontroller using serial port (USART). Usually, it connects small devices like mobile phones, PDAs and TVs using a short-range wireless connection to exchange documents. It uses the 2.45GHz frequency band. The connection can be point-to-point or multi-point where the maximum range is 10 meters. The transfer rate of the data is 1Mbps.

HC-05 Bluetooth module provides switching mode between master and slave mode which means it able to use neither receiving nor transmitting data.

Comparing it to the HC-06 module, which can only be set as a Slave, the HC-05 can be set as Master as well which enables making a communication between two separate Arduino Boards.

Resistors

Resistors are used to reduce current flow, adjust signal levels, to divide voltages, bias active elements, and terminate transmission lines, among other uses. Resistors doesn't have polarity which it can be connected in any direction.

Based on requirement we use different resistors for this project "**10k resistor"** is used, which have color code of brown, black, orange, gold.

Figure 3.6: Resistor

Dot Board

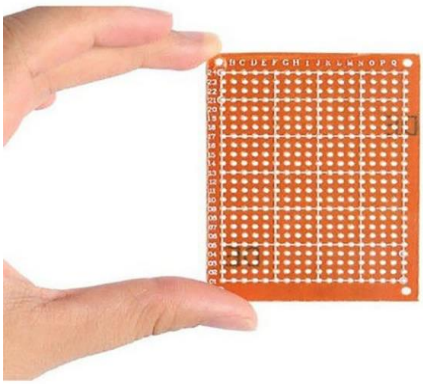

Figure 3.7: Dot Board

Dot Board is used to solder the components as a small circuit board. Which components are soldered using soldering kit. By this we get a hassle free circuit.

Connecting Wires

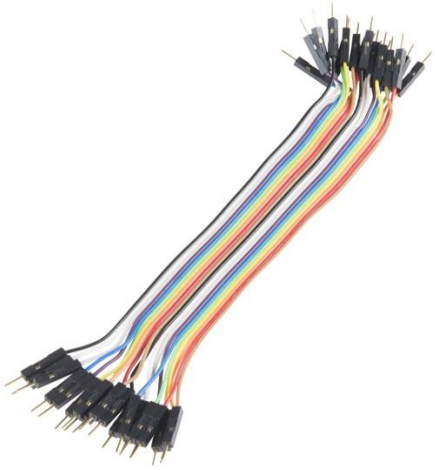

Figure 3.8: Jumper Wires

Silk wires and Jumper wires are used connect components each other.

Glove

To attach sensors a wearable glove is used. Which it is easy to carry and efficient or record hand movements.

Figure 3.9 Gloves

Power Supply

Power supply is given to Arduino Nano through mini USB cable with a Battery or Power bank which require minimum 5V.

3.2 Software Requirements

Arduino IDE

The Arduino IDE is incredibly minimalistic, yet it provides a near-complete environment for most Arduino-based projects. The top menu bar has the standard options, including “File” (new, load save, etc.), “Edit” (font, copy, paste, etc.), “Sketch” (for compiling and programming), “Tools” (useful options for testing projects), and “Help”. The middle section of the IDE is a simple text editor that where you can enter the program code. The bottom section of the IDE is dedicated to an output window that is used to see the status of the compilation, how much memory has been used, any errors that were found in the program, and various other useful messages.

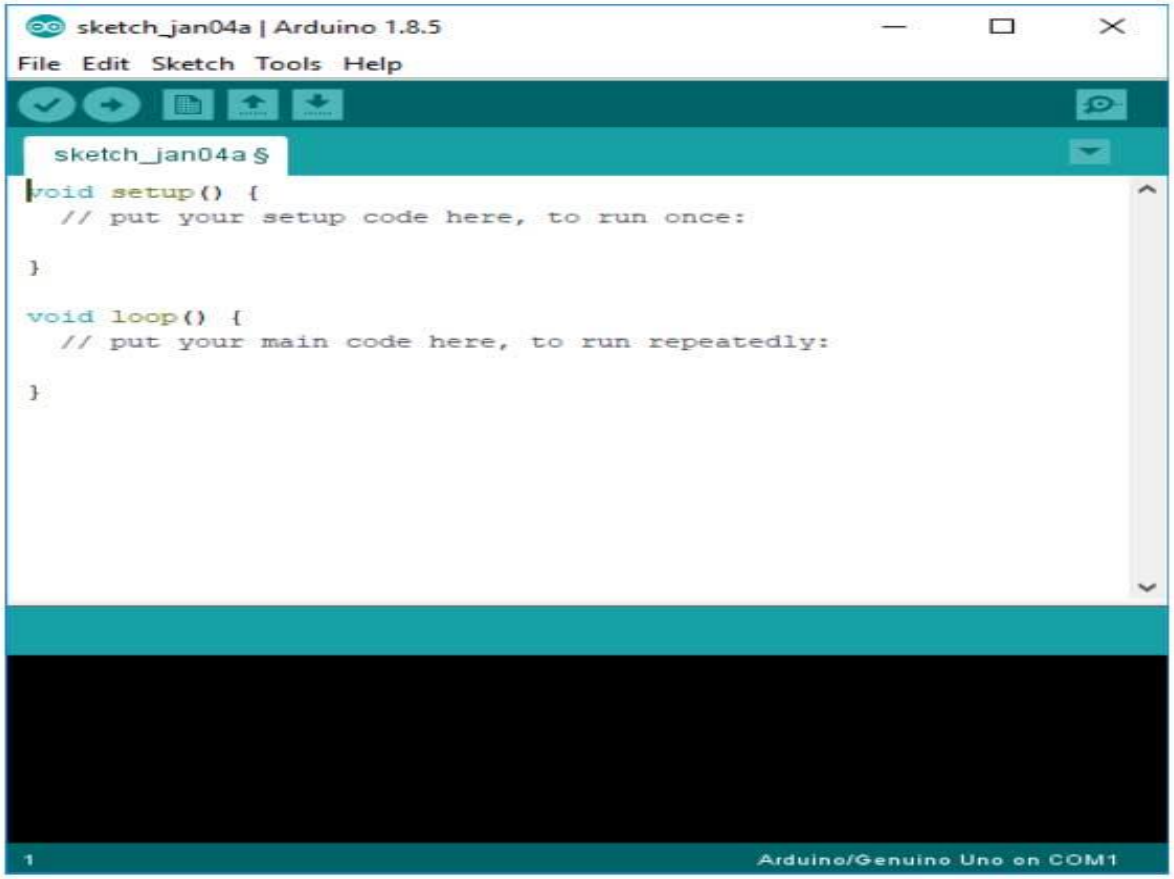

Figure 3.10: Arduino IDE in its default state

Projects made using the Arduino are called sketches, and such sketches are usually written in a cut-down version of C++ (a number of C++ features are not included). Because programming a microcontroller is somewhat different from programming a computer, there are a number of device-specific libraries (e.g., changing pin modes, output data on pins, reading analog values, and timers). This sometimes confuses users who think Arduino is programmed in an "Arduino language." However, the Arduino is, in fact, programmed in C++. It just uses unique libraries for the device.

Arduino Bluetooth Text-to-Speech(TTS) App

Arduino processed data is sent through to Bluetooth hc-05 to this Android App, which app receives the text from arduino and converts to speech which is Text-to-Speech, by this output sound heard through this app. This app is available on Google Play store.

Figure 3.11: Arduino Bluetooth TTS App

4. SYSTEM DESIGN

The basic goal of system design is to plan a solution for the problem. This phase is composed of several systems. This phase focuses on the detailed implementation of the feasible system. It emphasis on translating design specifications to performance specification. System design has two phases of development logical and physical design. During logical design phase the analyst describes inputs (sources), outputs (destinations), databases (data stores) and procedures (data flows) all in a format that meets the user requirements. The analyst also specifies the user needs and at a level that virtually determines the information flow into and out of the system and the data resources. Here the logical design is done through data flow diagram and database design. The logical design is followed by physical design or coding. Physical design produces the working system by defining the design specifications, which tell the programmers exactly what the candidate system must do. The programmers write the necessary programs that accept input from the user, perform necessary processing on accepted data through call and produce the required report on a hard copy or display it on the screen.

4.1. Section Overview

4.1.1. Logical Design

Logical design of an information system shows the major features and also how they are related to one another. The first step of the system design is to design logical design elements. This is the most creative and challenging phase and important too. Design of proposed system produces the details of the state how the system will meet the requirements identified during the system analysis that is, in the design phase we have to find how to solve the difficulties faced by the existing system. The logical design of the proposed system should include the details that contain how the solutions can be implemented. It also specifies how the database is to be built for storing and retrieving data, what kind of reports are to be created and what are the inputs to be given to the system. The logical design includes input design, output design, and database design and physical design.

4.1.2. Input Design

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data into a usable form for processing data entry. The activity of putting data into the computer for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling errors, avoiding delay, avoiding extra steps and keeping the process simple. The system needs the data regarding the asset items, depreciation rates, asset transfer, physical verification for various validation, checking, calculation and report generation.. The error raising method is also included in the software, which helps to raise error message while wrong entry of input is done. So in input design the following things are considered.

a. What data should be given as input
b. How the data should be arranged or coded
c. The dialogue to guide the operating personnel in providing input.
d. Methods for preparing input validations and steps to follow when error occur
e. The samples of screen layout are given in the appendix.

4.1.3. Output Design

Computer output is the most important and direct information source to the user. Output design is a process that involves designing necessary outputs in the form of reports that should be given to the users according to the requirements. Efficient, intelligible output design should improve the system's relationship with the user and help in decision making. Since the reports are directing referred by the management for taking decisions and to draw conclusions they must be designed with almost care and the details in the reports must be simple, descriptive and clear to the user. So while designing output the following things are to be considered.

4.1.4. Physical Design

The process of developing the program software is referred to as physical design. We have to design the process by identifying reports and the other outputs the system will produce. Coding the program for each module with its logic is performed in this step. Proper software specification is also done in this step.

4.2. UML DIAGRAMS

The Unified Modelling Language allows the software engineer to express an analysis model using the modelling notation that is governed by a set of syntactic semantic and pragmatic rules. A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows.

4.2.1. User Model View

This view represents the system from the users perspective. The analysis representation describes a usage scenario from the end-users perspective.

4.2.2. Structural Model View

In this model the data and functionality are arrived from inside the system. This model view models the static structure.

4.2.3. Behavioural Model View

It represents the dynamic of behavioural as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

4.2.4. Implementation Model View:

In this the structural and behavioural as parts of the system are represented as they are to be built.

4.2.5 Environmental Model View:

In this the structural and behavioural aspects of the environment in which the system is to be implemented are represented.

CLASS DIAGRAM:

Class diagrams are the main building blocks of every object oriented methods. The class diagram can be used to show the classes, relationships, interface, association, and collaboration. UML is standardized in class diagrams. Since classes are the building block of an application that is based on OOPs, so as the class diagram has appropriate structure to represent the classes, inheritance, relationships, and everything that OOPs have in its context. It describes various kinds of objects and the static relationship in between them.

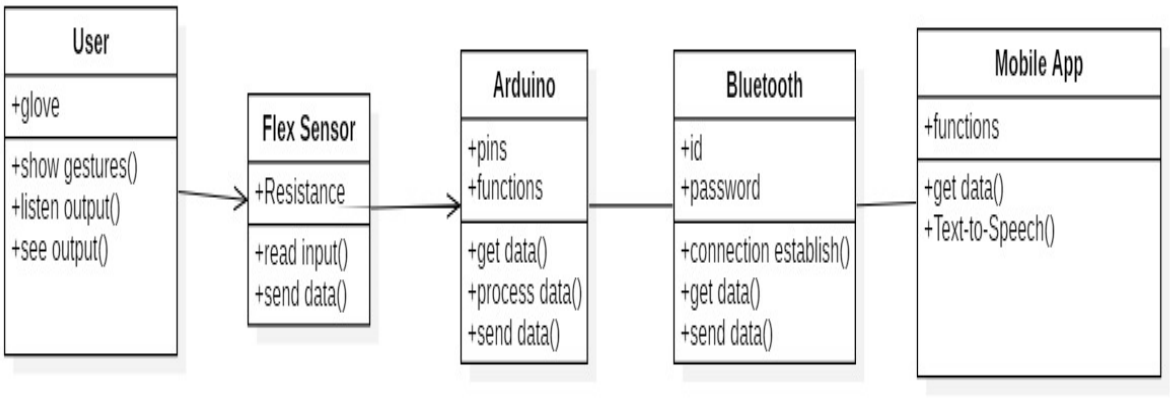

Figure 4.1: Class Diagram

USECASE DIAGRAM:

A use case is a set of scenarios that describing an interaction between a user and a system. A use case diagram displays the relationship among actors and use cases. The two main components of a use case diagram are use cases and actors.

An actor is represents a user or another system that will interact with the system you are modeling. A use case is an external view of the system that represents some action the user might perform in order to complete a task.

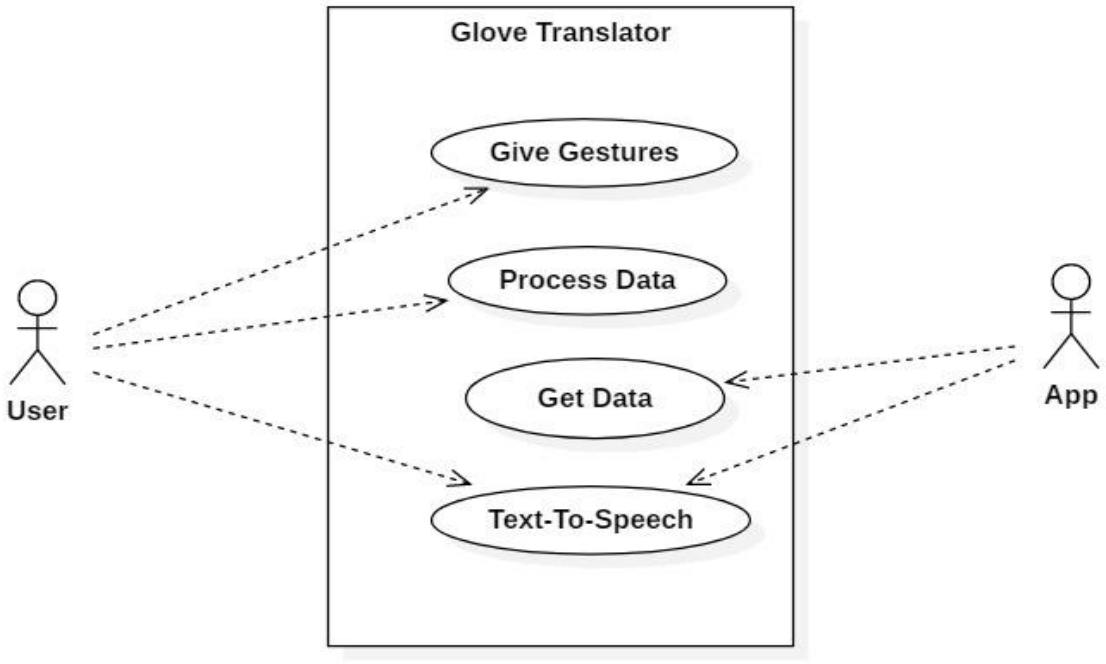

Figure 4.2: Usecase Diagram

ACTIVITY DIAGRAM:

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc

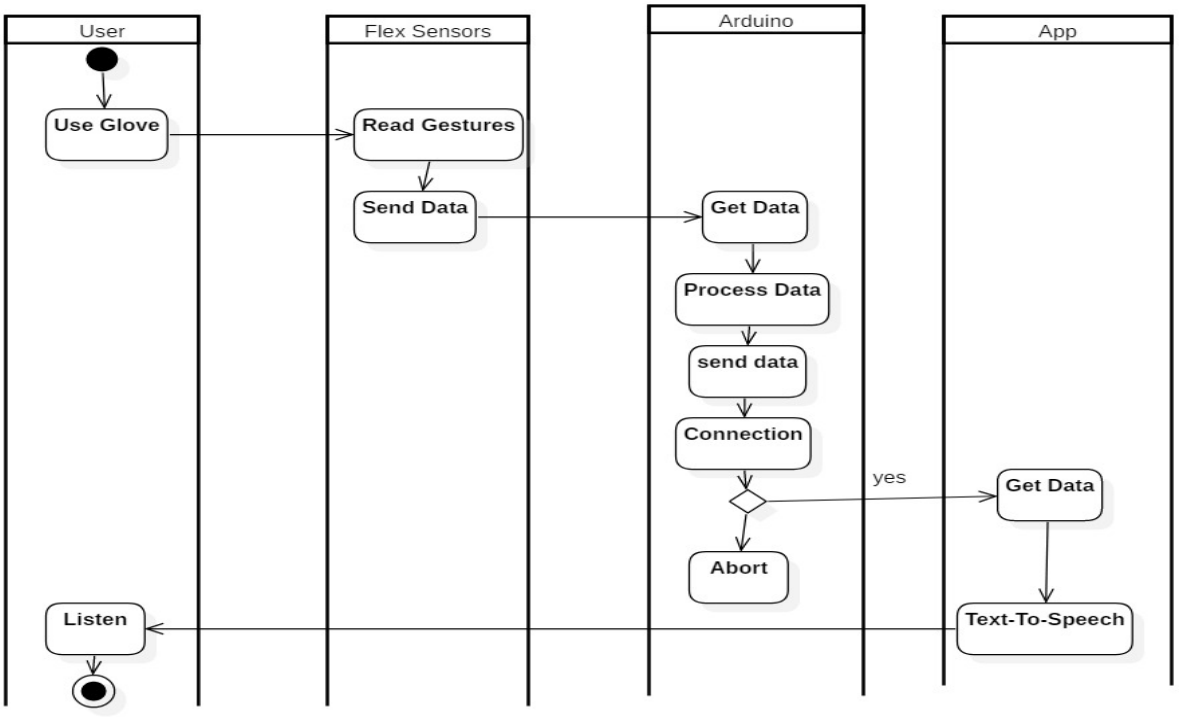

Figure 4.3: Activity Diagram

SEQUENCE DIAGRAM:

A Sequence diagram is an interaction diagram that shows how processes operate with one another and what is their order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

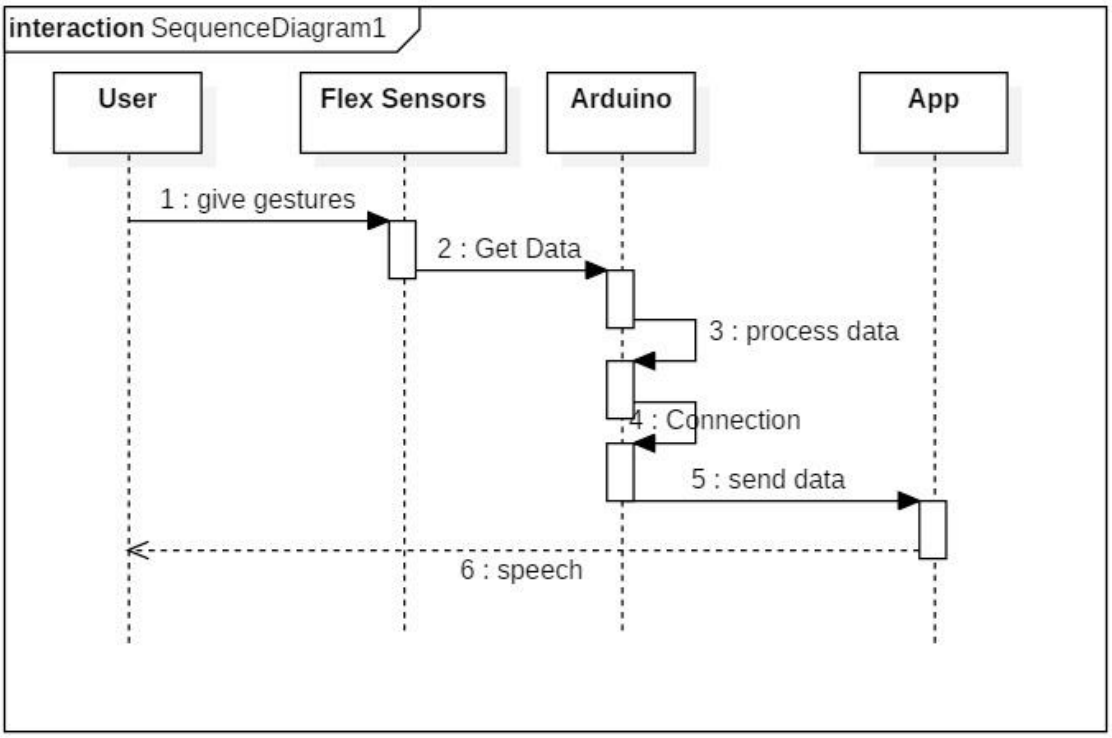

Figure 4.4: Sequence Diagram

COMPONENT DIAGRAM:

A component diagram shows the internal parts, connectors, and ports that implement a component. When the component is instantiated, copies of its internal parts are also instantiated. The UML component diagram shows how a software system will be composed of a set of deployable components—dynamic-link library (DLL) files, executable files, or web services—that interact through well-defined interfaces and which have their internal details hidden.

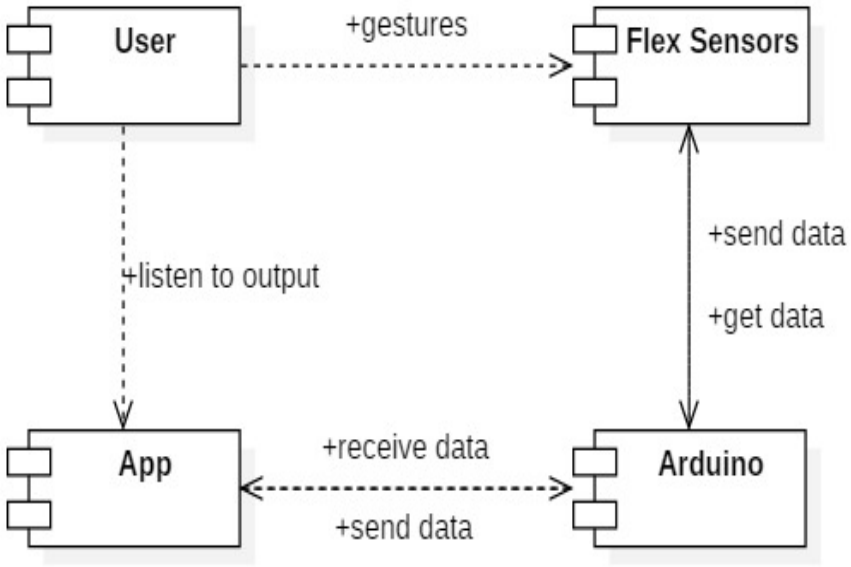

Figure 4.5: Component Diagram

DEPLOYMENT DIAGRAM:

Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed. So deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships.

Deployment diagrams are useful for system engineers. An efficient deployment diagram is very important because it controls the following parameters

- Performance
- Scalability
- Maintainability
- Portability

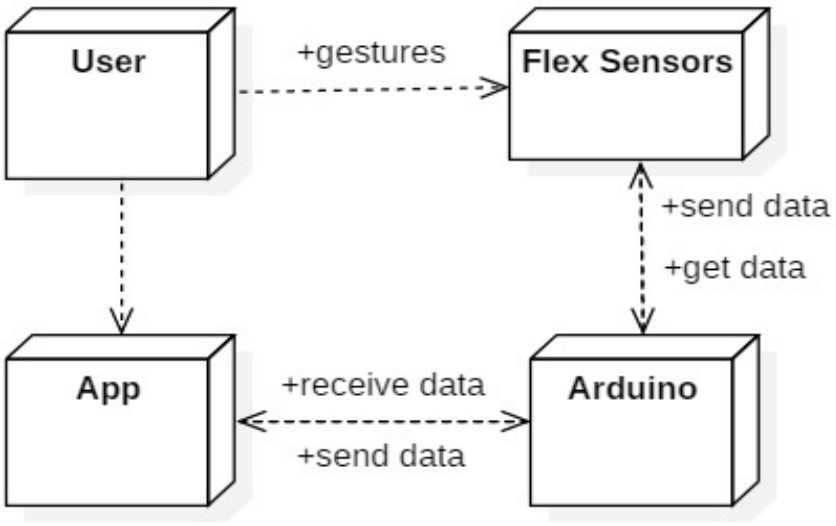

Figure 4.6: Deployment Diagram

5. SYSTEM IMPLEMENTATION

5.1 Gathering and Connecting Components

Components Required:

- Flex Sensors
- Arduino Nano
- 10k Resistors
- Bluetooth Module
- Connecting Wires
- Dot Board or Bread board

All the components required to the project are gathered and connected according to the below circuit.

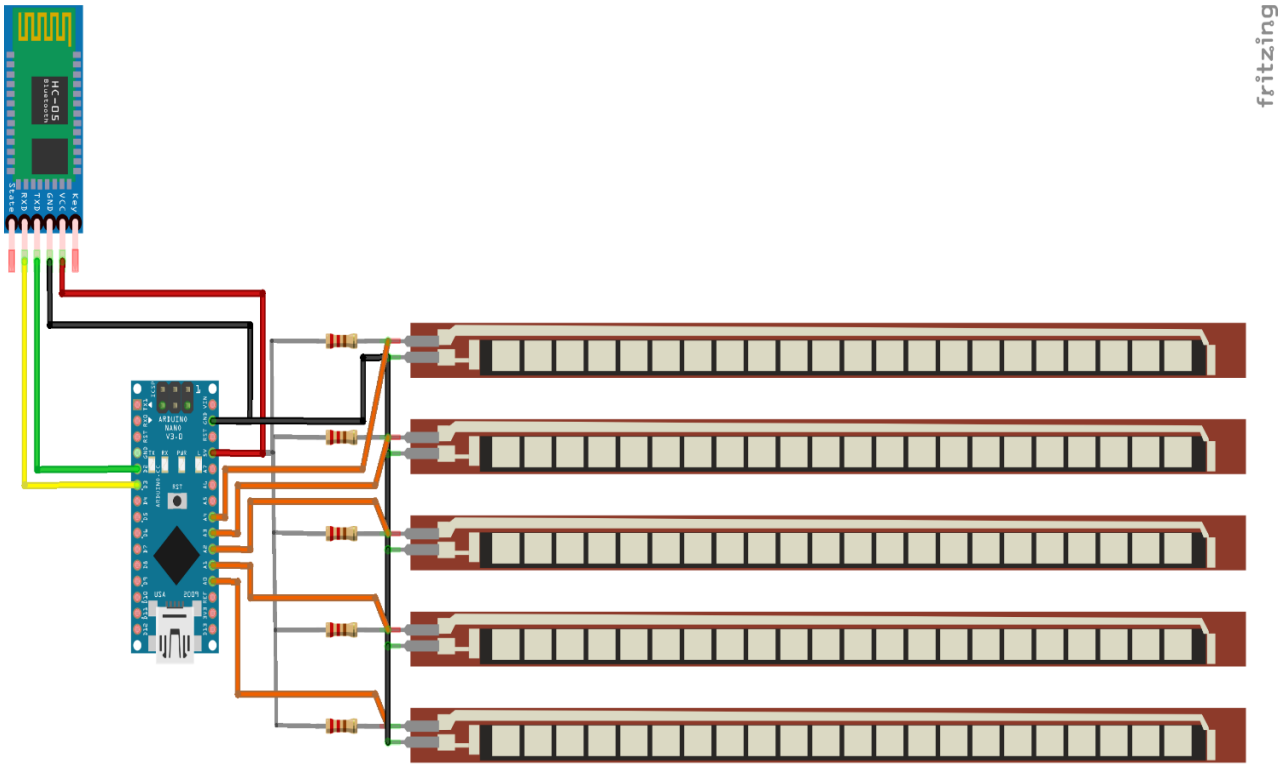

Figure 5.1: Circuit Diagram

5.2 Connecting Flex Sensors to Arduino Nano

- Connect one end of sensor to the Vcc with 10k resistor.
- Other end to respective analog pin and in series to Gnd.
- Repeat the same process for all flex sensors.

5.3 Connecting Bluetooth Hc-05 with Arduino Nano

1. Connect Power Supply (based on datasheet of modules) for Bluetooth and Microcontroller which you are using.
2. Connect TXD pin of HC-05 Bluetooth module to RXD pin of Microcontroller.
3. Connect RXD pin of HC-05 Bluetooth module to TXD pin of Microcontroller.
4. Common grounding should be needed for both modules.

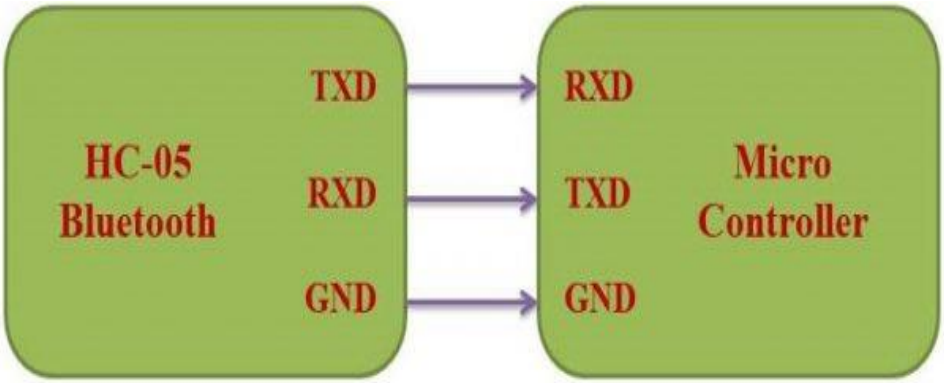

Figure 5.2: Bluetooth Connections

5.4 Algorithms

5.4.1 Algorithm for checking values

Step 1: Start Loop

Step 2: Declare Variables Thumb, index, middle, ring, little

Step 3: Read Values of Thumb, index, middle, ring, little

Step 4: Display Values in Serial Monitor

Step 5: Repeat Loop

5.4.2 Algorithm for Output

Step 1: Start Loop

Step 2: Declare Variables Thumb, index, middle, ring, little

Step 3: Read Values of Thumb, index, middle, ring, little

Step 4: if(finger>value)

Display data

Step 5: Repeat Loop

5.5 Functions Used

Operations like void setup(),void loop(),analogRead(), bt.println() and conditional statements are used.

void setup()-

The setup() function is called when a sketch starts. Use it to initialize variables, pin modes, start using libraries, etc. The setup() function will only run once, after each powerup or reset of the Arduino board.

void loop()-

After creating a setup() function, which initializes and sets the initial values, the loop() function does precisely what its name suggests, and loops consecutively, allowing your program to change and respond. Use it to actively control the Arduino board.

analogRead()-

Reads the value from the specified analog pin. Arduino boards contain a multichannel, 10-bit analog to digital converter. This means that it will map input voltages between 0 and the operating voltage(5V or 3.3V) into integer values between 0 and 1023.

bt.println()-

Prints text or data to Bluetooth device, which ‘bt’ is a object create from SoftwareSerial. println prints the statements in next line. This Device Captures the motion of hand through analog values given from flex sensor and processed and output is sent to Bluetooth to the App.

5.6 Analyse The Readings

Arduino gives out analog readings which is raw data. Either raw data can be used for processing or else offset values and corrections can also be made. After connecting the device, the readings of 5 fingers are studies and noted according to their respective positions. Which by analysis the data words or sentences are set to particular gestures.

Need for much analysis for better accuracy of gesture recognisation. Each finger their combinations are also need to be studied.

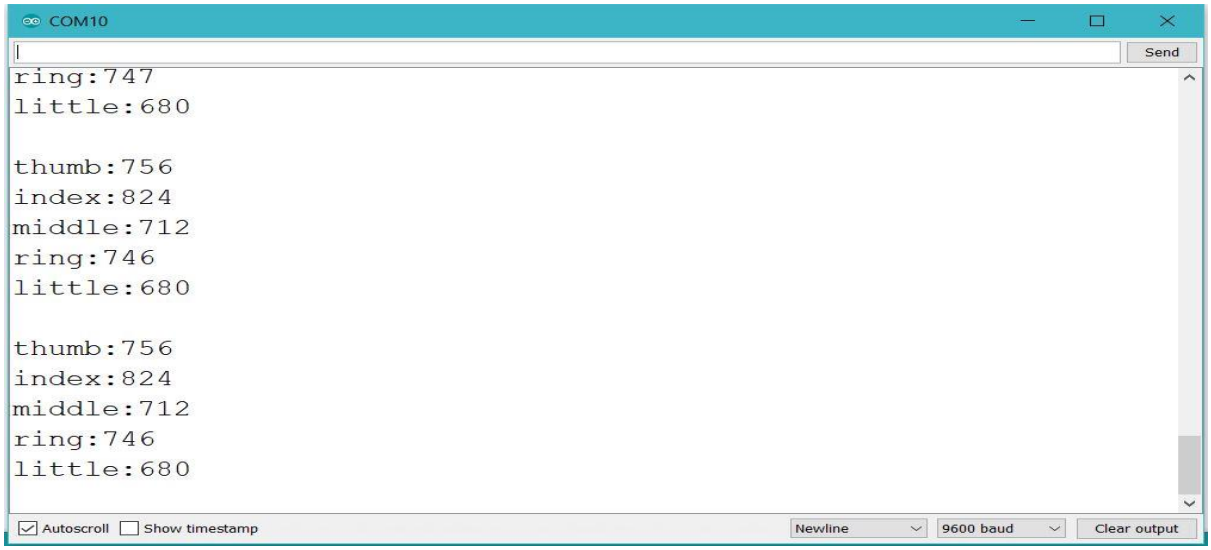

Figure 5.3: Readings of Sensor 1

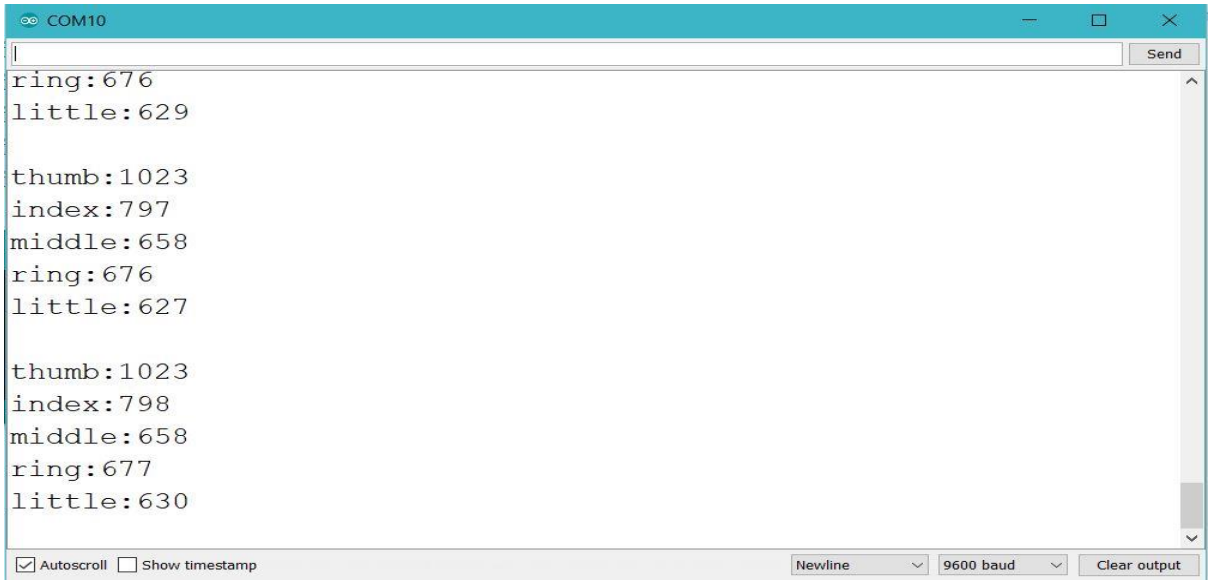

Figure 5.4: Readings of Sensor 2

5.7 Reading the Gesture Values

Code:

```
const int a = A0;
const int b=A1;
const int c=A2;
const int d=A3;
const int e=A4;

int thumb;
int index;
int middle;
int ring;
int little;

void setup()
{
Serial.begin(9600);
}

void loop()
{
thumb=analogRead(a);
Serial.print("thumb:");
Serial.println(thumb);
index=analogRead(b);
Serial.print("index:");
Serial.println(index);
middle=analogRead(c);
Serial.print("middle:");
Serial.println(middle);
ring=analogRead(d);
Serial.print("ring:");
```

```
Serial.println(ring);
little=analogRead(e);
Serial.print("little:");
Serial.println(little);

delay(1000);
Serial.println("");
}
```

5.8 Setting Data for Appropriate Positions

After analysis of data, values for every position is marked and data is set to for respective gestures.

Code:

```
#include<SoftwareSerial.h>

const int val0=A0;
const int val1=A1;
const int val2=A2;
const int val3=A3;
const int val4=A4;
int count=0;
int power;

SoftwareSerial bt(2,3 ); /* (Rx,Tx) */

void setup()
{
  bt.begin(9600);
  Serial.begin(9600);
}

void loop()
{
  int thumb = analogRead(val0);
  int index = analogRead(val1);
  int middle = analogRead(val2);
  int ring = analogRead(val3);
  int little = analogRead(val4);

  Serial.println(thumb);
  Serial.println(index);
```

```
  Serial.println(middle);
  Serial.println(ring);
  Serial.println(little);
   Serial.println();
  delay(1000);

  if (bt.available())
  {
    Serial.write(bt.read());
  }


if((thumb<800)&&(little<800)){
   count++;
}

if(count%2==0)
{
  int power=0;
}
else{
  power=1;
}

  if(power=1)
 {
  bt.println("device on");
if((thumb>580)&&(thumb<630))
{
  bt.println("how are you");
}
```

```
if((thumb>770)&&(thumb<840) && (index>640)&& (index<680))
{
  bt.println("you look beautiful");
}
if((thumb>820)&&(thumb<900))
{
bt.println("Look at me");
}
 delay(1500);
}
else
{
  bt.println("device off");
}
}
```

6. SYSTEM TESTING

6.1 Testing objective

Software Testing is the process used to help identify the correctness, completeness, security, and quality of developed computer software. Testing includes, but is not limited to, the process of executing a program or application with the intent of finding errors. Quality is not an absolute; it is value to some person. With that in mind, testing can never completely establish the correctness of arbitrary computer software. Testing furnishes a criticism or comparison that compares the state and behaviour of the product against a specification. An important point is that software testing should be distinguished from the separate discipline of Software Quality Assurance (SQA), which encompasses all business process areas, not just testing.

6.2 Test Case Design

Testing is a process of executing a program to find out errors. If testing is conducted successfully, it will uncover all the errors in the software. Any testing can be done basing on two ways:

6.2.1 White Box Testing

It is a test case design method that uses the control structures of the procedural design to derive test cases. Using this testing a software Engineer can derive the following test cases: Exercise all the logical decisions on either true or false sides. Execute all loops at their boundaries and within their operational boundaries. Exercise the internal data structures to assure their validity.

We test the loops and conditional statements in the code to check its working.

6.2.2 Black Box Testing

It is a test case design method used on the functional requirements of the software.

6.3 Unit Testing

Unit testing verification efforts on the smallest unit of software design, module. This is known as “Module Testing”.

Here we test working of every individual module. First we test arduino nano with an example program and check whether it is working or not. In the same way we test 5 flex sensors working individually.

Bluetooth module is also tested whether it is connected to phone and transmission and receiver is working. It’s better to check all connecting wires are working properly.

6.4 Integration Testing

Integration testing is a systematic technique for constructing tests to uncover error associated within the interface. We integrate all modules like sensors, Bluetooth are connected with power supply then tested by this we can find errors in interfaces.

6.5 Acceptance Testing

Acceptance Testing is also called as “Beta Testing”, Application Testing, End user “Testing”, it is a phase of software development in which software is tested in the real world by the intended audience.

This model is given to end user to check its accuracy and comfort for the user, based on the user’s review we make changed to the project.

6.6 Test Cases

FUNCTION	EXPECTED REUSLTS	ACTUAL RESULTS	LOW PRIORITY	HIGH PRIORITY
Check all the components are connected	Device gives an error or doesn't work properly	device works properly		yes
Check app is Connected with Bluetooth of device	If not connected output will not be displayed on app	It displays output and generates sound to the text.		Yes
Check the Correct board and port is selected in IDE	If not connected properly code will not upload	Code uploads into lilypad		yes
Check the syntax and semantics of the code	There will be errors and bugs	Code executes successfully		yes
Give gesture (index finger)	If not relevant word is not shown	It displays and speaks out 'You' as output	Yes	
Give gesture (all fingers)	If not relevant word is not shown	It displays and speaks out 'Hi' as output	Yes	

Table 1: Test Cases

7. SCREENSHOTS

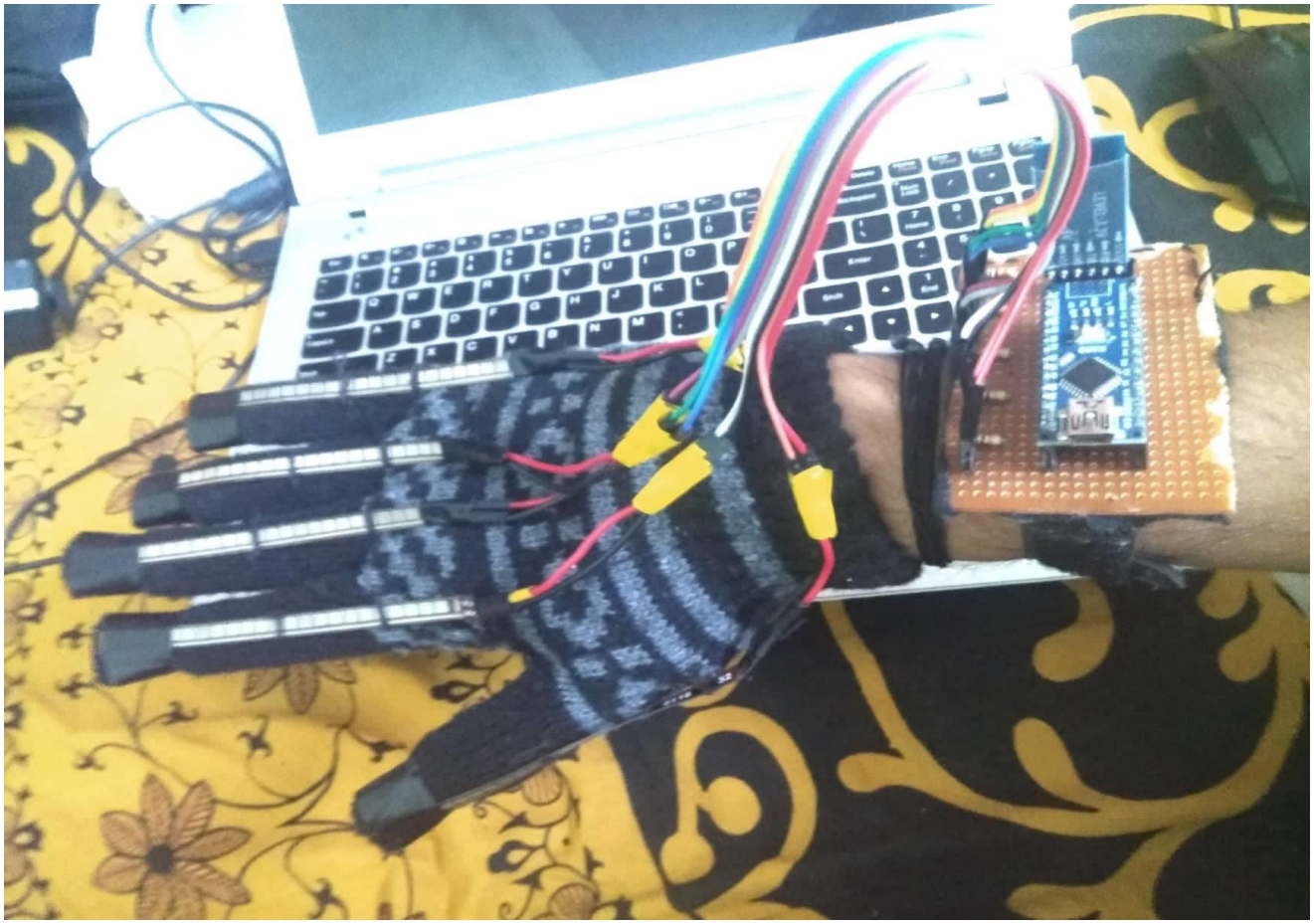

Figure 7.1: View of Model 1

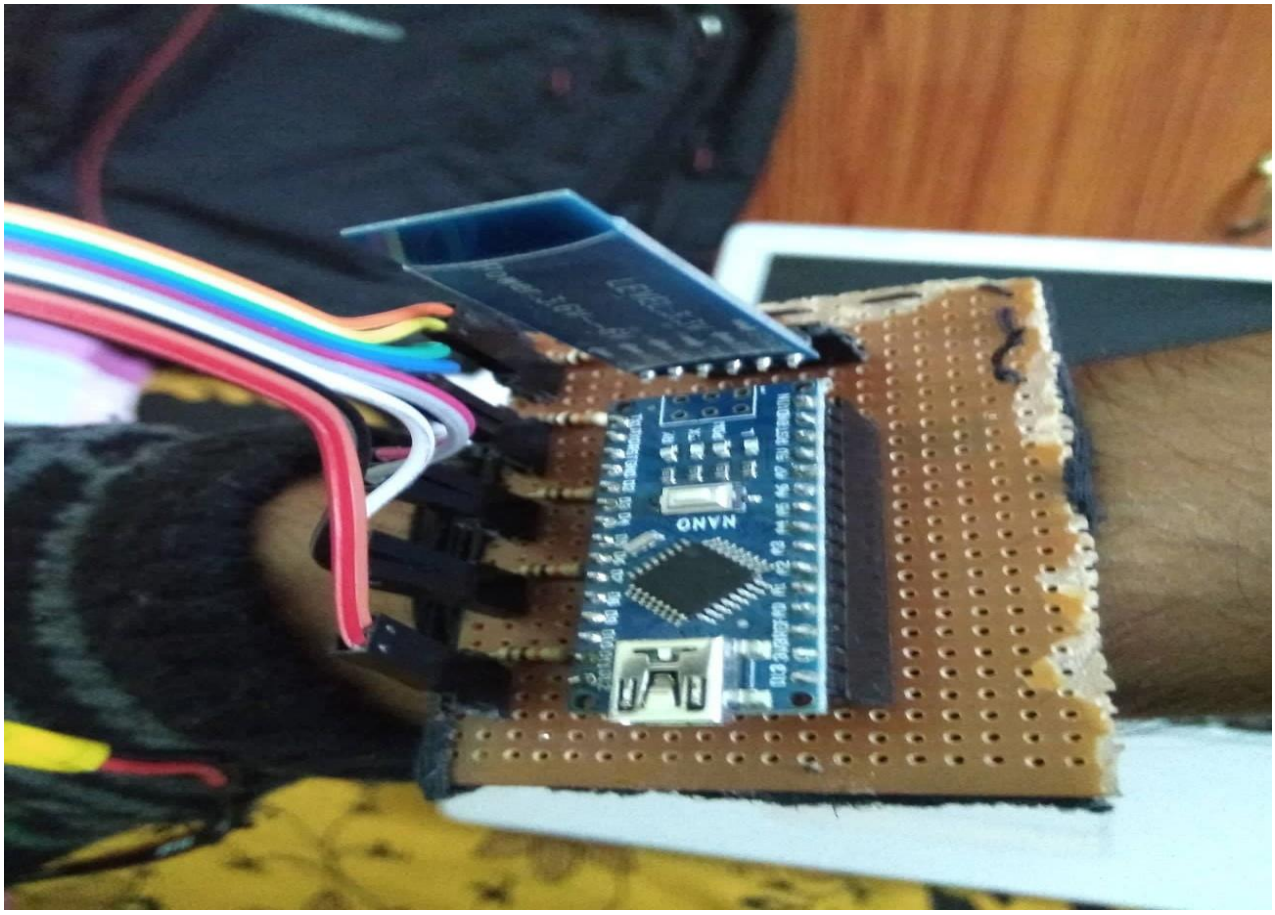

Figure 7.2: View of Model 2

Figure 7.3: Gesture for “ok fine”

By this hand gesture it gives out sound and displays “ok fine” in the App.

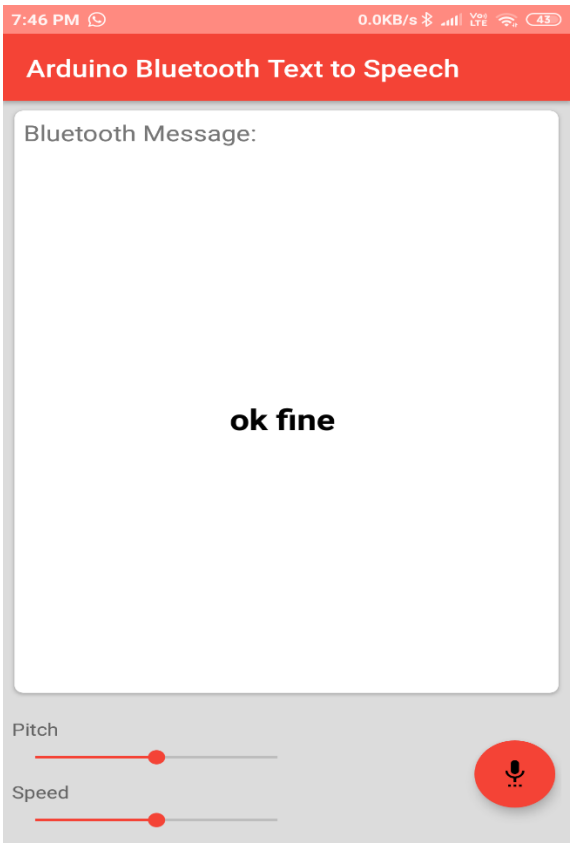

Figure 7.4: Output in App

Figure 7.5: Gesture for “thank you”

By this hand gesture it gives out sound and displays “thank you” in the App.

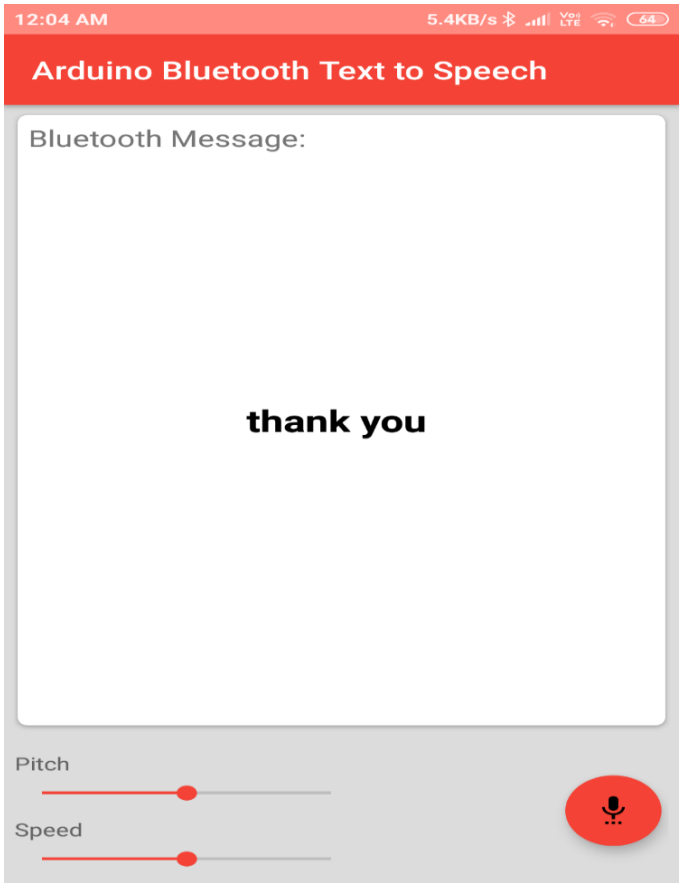

Figure 7.6: Output in App

Figure 7.7: Gesture for “how are you”

By this hand gesture it gives out sound and displays “How are you” in the App.

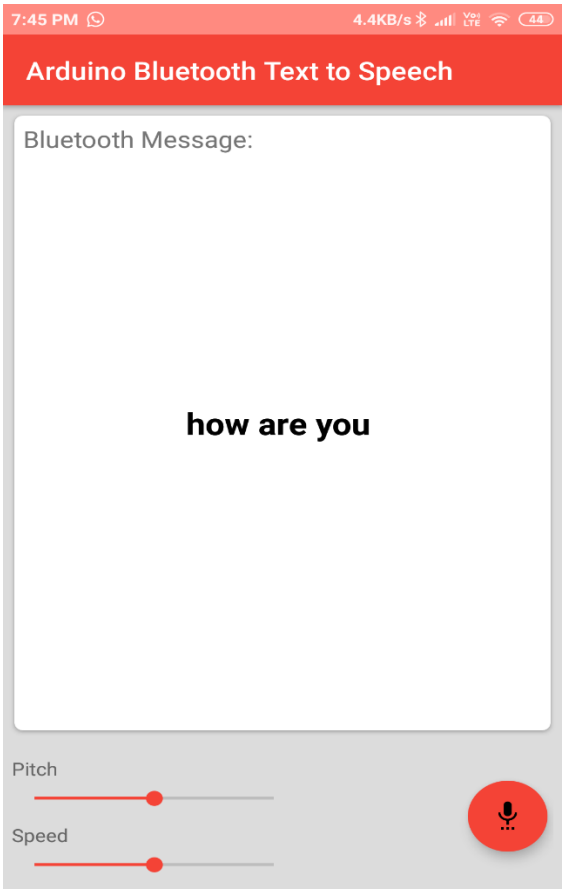

Figure 7.8: Output in App

8. CONCLUSION AND FUTURE ENHANCEMENTS

The team has succeeded in building all of the necessary hardware for the project. However, more can be done to improve the device's accuracy, form, and other important specifications.

To help shape the device into something more slim and comfortable, the team suggests a few improvements to the prototype. Primarily, the team suggests designing a smaller and more compact PCB, which will more efficiently combine the components of the system. The team also recommends using conductive fabric to possibly replace the wiring, the contact sensors, or the flex sensors to allow for less bulky wiring and more conforming, lightweight connections. Slimmer battery supplies which may provide the required voltage and hopefully enough power to last for many hours would also be a valued improvement to the prototype.

The software used in the algorithm has not been finely tuned. There are many optimizations which may be possible. An even greater improvement in its function may be iterative debugging using signs, which will be possible once the new prototype is steadily constructed. The current software does not use real tolerances or multiple samples per gesture. In addition, completion of the Android app would increase the ease of use through text-to-speech output, the number of gestures understood, and the accuracy of the glove since the smartphone has more computational resources. Testing with actual Sign Language users to determine how well the user requirements and design specifications were followed would allow for a more accurate library of gestures and a more practical product.

Hand rotations can also be captured using gyroscope which detects x,y,z axis. Much improvements can be done using Machine Learning, which learns movements for a word without need of coding with much accuracy by giving the test data.

Implementing Speaker and Smaller PCB Board on the Glove itself, without need of App or mobile to translate it. Adding multiple language support for different regional users. The below figure show the proposed model.

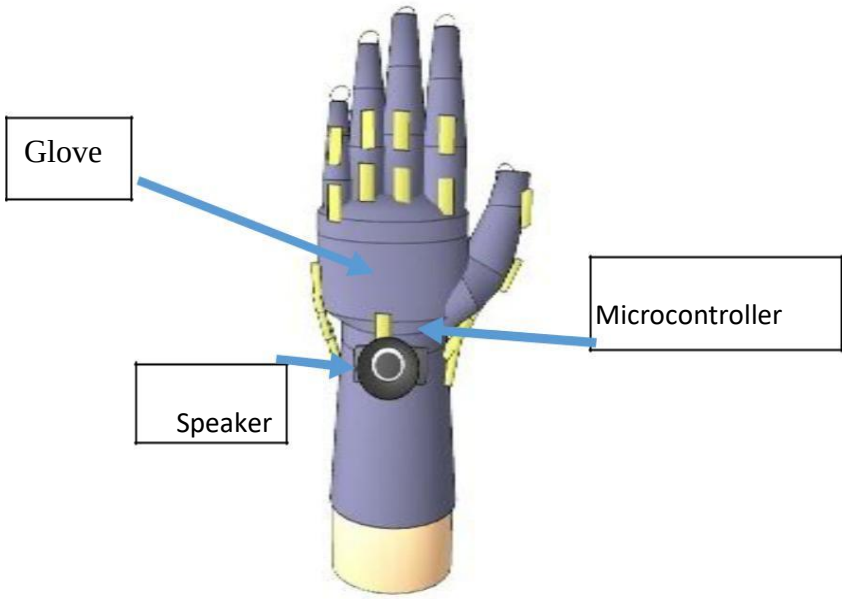

Figure 8.1: Proposed Speaker System

9. BIBLIOGRAPHY

1. https://play.google.com/store/apps/details?id=com.zaptha.bluetooth.bluetoothsensor&hl=en
2. https://ieeexplore.ieee.org/document/7002401
3. https://www.theengineeringprojects.com/2018/06/introduction-to-arduino-nano.html
4. https://learn.sparkfun.com/tutorials/flex-sensor-hookup-guide/all
5. https://en.wikipedia.org/wiki/Speech_synthesis
6. https://www.nidcd.nih.gov/health/american-sign-language
7. https://www.smithsonianmag.com/innovation/wearable-device-translates-sign-language-english-180956827/