PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)

Electronic City, Bengaluru – 560 100, Karnataka, India

Report on

"SIGN LANGUAGE TO TEXT AND SPEECH"

Submitted by

BASIREDDY KISHAN REDDY	-	**PES2201800554**
DHANUSH M	-	**PES2UG19EC808**
SIRI RACHAPPA JARMALE	-	**PES2201800662**
TEJAS RAJ G R	-	**PES2201800304**

August – December 2021

Under the guidance of

Internal Guide

Prof. UDOSHI BASAVARAJ

Department of ECE

PES University

Bangalore-560034

FACULTY OF ENGINEERING

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

PROGRAM B.TECH

CERTIFICATE

This is to certify that the report entitled

"Sign Language to Text and Speech Conversion"

Is a bonafide work carried out by

BASIREDDY KISHAN REDDY (PES2201800554)

DHANUSH M (PES2UG19EC808)

SIRI RACHAPPA JARMALE (PES2201800662)

TEJAS RAJ G R (PES2201800304)

In partial fulfillment for the completion of 7th semester course work in the Program of Study B.Tech in Electronics and Communication Engineering, under rules and regulations of PES University, Bengaluru during the period **January to May** and **June to December 2021**. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report. The report has been approved as it satisfies the 7th semester academic requirements in respect of Capstone project work.

Signature with date & Seal	*Signature with date & Seal*	*Signature with date & Seal*
Prof. Udoshi Basavaraj	***Dr. Ajey SNR***	***Dr. B K Keshavan***
Internal Guide	*Chairperson*	*Dean – Faculty Engg. & Tech.*

Name and signature of the examiners:

1. ……..

2. ……..

DECLARATION

We, ***Basireddy Kishan Reddy, Dhanush M, Siri Rachappa Jarmale,and Tejas Raj G R,*** hereby declare that the report entitled, *'**Sign Language to Text and Speech Conversion',*** is an original work done by us under the guidance of ***Prof. Udoshi Basavaraj***, *Designation*, ECE Department and is being submitted in partial fulfilment of the requirements for completion of 7th Semester course work in the Program of Study, B.Tech in Electronics and Communication Engineering.

PLACE:

DATE:

NAME AND SIGNATURE OF THE CANDIDATES

1. **BASIREDDY KISHAN REDDY**

2. **DHANUSH M**

3. **SIRI RACHAPPA JARMALE**

4. **TEJAS RAJ G R**

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to all the lecturers and staff of the department of electronic and communication engineering for extending their help and guidance towards our project.

We would like to thank the college management and express our sincere gratitude to Dr. Subhash Kulkarni, Principal of PESU EC have given me the opportunity for the completion of this project.

We would like thank Dr.Ajey S.N.R Head of Department Electronics and communication PESU EC for given us the support and encouragement that was necessary for the completion of this report.

We convey our thanks to the project coordinator, Dr Bajarangbali &
Prof. Muralidhar For providing us the needful information and facilities which was of a great help to complete this project successfully.

We would like to thank our guide Prof. Udoshi Basavaraj Assistant Professor, Department of Electronics and Communication Engineering, PESU EC for providing us required assistance, encouragement and constant support which helped us a lot.

Finally, we would like to express our gratefulness to our parents and friends, whose suggestions and support helped us to realize the completion of this project.

ABSTRACT

People interact with each other to communicate their thoughts, ideas, and experiences to those around them. However, this is not the case for deaf and mute people. Sign language paved the way for communication for the deaf. Deaf people can communicate without a voice using sign language. The objective of this work was to develop a sign language recognition system that would allow people with speech impairments to communicate with the public and reduce the communication gap between them. Hand gestures play an important role in being able to express a user's point of view in a shorter time than other gestures (arms, face, head, body). In the present study, we developed a flex sensor-based gesture recognition module that can recognize letters and some words and convert that text to speech using the Bluetooth Text to Speech app.

CONTENTS

FIGURES

CHAPTER 1

INTRODUCTION

1.1. BACKGROUND

Communication between the speech impaired community and the general public has always been an issue. The learning of sign language is a proposed solution that will require the awareness of every single individual. These efforts of creating awareness of sign language can be overcome with the help of technology. Technology plays an important role in every single aspect of our lives, and it can certainly help the deaf community as well. The sole purpose of this project is to bridge the gap between the communication of individuals and the Deaf-community by creating a glove that translates gestures to speech and text. This can also create a welcoming environment for them to be more communicative, confident, and interactive in the outside world.

1.2. MOTIVATION / PROBLEM STATEMENT

In the fast and developing world, the main challenge for the deaf community is to be able to communicate with everyone around them. The main intent of our project is to demonstrate the development of a Glove that can detect gestures that represent the alphabets of the sign language which is then converted to speech and text that any individual can understand. The project attempts to detect and convert signs in an easy, fast, and convenient way that provides a solution to the above-mentionedreal-world real world problem.

1.3. SIGN LANGUAGE GESTURES

Sign language is basically a visual means of communication through gestures, body language, and hand movements.

It is the most important form of communication for the Deaf and Hard-of-Hearing community. People with disabilities such as Autism, Apraxia of speech, Cerebral Palsy, and Down Syndrome also can use sign language to communicate.

There are several sign languages currently in use all over the globe. Similar to spoken language, even sign languages have developed regionally in groups of people interacting with each other, explaining the number of sign languages present. There are around 138 -300 different sign languages currently in practice.

One can note that several countries share the same vocal language yet do not have the same sign language.

One such example is English has three varieties in Sign Language itself:

- American Sign Language (ASL)
- British Sign Language (BSL)
- Australian Sign Language (Auslan)

In general one may start learning sign language by learning sign form equivalent to each alphabet. The use of the hand signs to represent each letter of a word is called 'fingerspelling'.

Sign language may represent alphabets in one hand, for example, ASL and French Sign Language, or may use both hands for example in BSL or Auslan. There are several similarities between each of the alphabets but each sign language has its style.

American Sign Language (ASL)

ASL is one of the most popularly used sign languages and hence we have chosen it for our project. Our project can be programmed for any sign language with very minor changes since the structure, logic, software, and hardware remains the same. ASL has the same alphabet as English, ASL is not a subset of the English language. American Sign Language was created independently and it has its linguistic structure. (It is descended from Old French Sign Language.)

Signs are also not expressed in the same order as words are in English. This is due to the unique grammar and visual nature of sign language.

Fig 1.1 American Sign Language

Chinese Sign Language (CSL)

Chinese Sign Language is one more very popularly used sign language. This language started developing in the 1950s. It uses hand gestures to make a visual sign of the written Chinese characters.

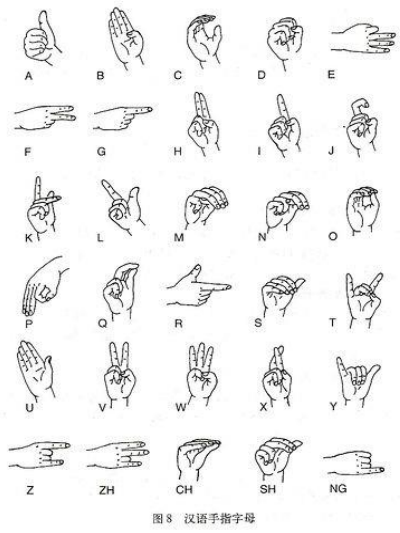

Fig 1.2 Chinese Sign Language

We have used a combination of the hand gestures from the ASL and CSL to create hand gestures that can be determined accurately with the system we have used.

CHAPTER 2

LITERATURE SURVEY

LITERATURE SURVEY

Paper [1]: *S. A. Mehdi and Y. N. Khan, "Sign language recognition using sensor gloves," Proceedings of the 9th International Conference on Neural Information Processing, 2002. ICONIP '02., 2002, pp. 2204-2206 vol.5, doi: 10.1109/ICONIP.2002.1201884.*

This paper examines the opportunity of spotting sign language gestures with the use of sensor gloves. Previously sensor gloves had been utilized in video games or applications with custom gestures. This paper explores their use in Sign Language popularity. This is done with the aid of implementing a venture referred to as “Talking Hands”, and reading the results. The challenge makes use of a sensor glove to seize the signs and symptoms of American Sign Language performed by a consumer and translates them into sentences of English language. Artificial neural networks are used to recognize the sensor values coming from the sensor glove. These values are then classified in 24 alphabets of the English language and two punctuation symbols delivered by way of the writer. So, mute people can write complete sentences the use of this utility. One problem that changed into confronted inside the challenge changed into that one of the alphabets concerned dynamic gestures

These may not be diagnosed using this glove. So these have been left out from the area of the assignment.

Paper[2]: *V. Y. Flamenco, P. M. Yanik, R. D. Adams, and M. L. Tanaka, "An Automated Method for Evaluating the Accuracy of ASL Static Gestures," 2014 International Conference on Computational Science and Computational Intelligence, 2014, pp. 173-177, doi: 10.1109/CSCI.2014.36.*

This paper has offered a brand new approach in correcting the location of static American Sign Language (ASL) gestures the usage of present algorithms and characteristic reputation strategies. In ASL the position of the fingers is very crucial thinking about a moderate misplacement conveys a totally one of a kind phrase, letter, or meaning.

Paper [3]: A. Kannan, A. Ramesh, L. Srinivasan and V. Vijayaraghavan, "Low-cost static gesture recognition system using MEMS accelerometers," 2017 Global Internet of Things Summit (GIoTS), 2017, pp. 1-6, doi: 10.1109/GIOTS.2017.8016217.

The number one objective of this paper is to assemble and check a low-cost, minimally supervised gesture popularity device thatmentioned realtatic gestures efficaciously and as they should be. The proposed machine makes use of ADXL335 accelerometer sensors which detect the gestures and these sensors are interfaced with an Arduino ATMega 2560 microcontroller for data processing and gesture popularity. The overall performance of the system is classed the use of static gestures within the alphabets of the American Sign Language. (ASL)

Paper [4]: *O. Sidek and M. Abdul Hadi, "Wireless gesture recognition system using MEMS accelerometer," 2014 International Symposium on Technology Management and Emerging Technologies, 2014, pp. 444-447, doi: 10.1109/ISTMET.2014.69365.*

This paper provides the development of a wi-fi Bluetooth hand gesture reputation gadget the usage of six three-axis accelerometers embedded in a glove and a database gadget in a pc. This machine can recognize any sampled information saved inside the database at the same time as selling maximum portability and mobility to the person via wi-fi Bluetooth era. Analyses along with static statistics, dynamic records, and average recognition fees relationships are mentioned in this paper.

CHAPTER 3

TECHNOLOGY FOR CAPTURING GESTURE

3.1 Mechanics of the Sign Language

Gestures used in the ASL are a significant movement of the hand to convey some information. The recognition of these gestures involves the technology of tracking the hands by interpreting the various orientations and movements. We have proposed a design that is a microcontroller-based hybrid system to identifies hand gestures using flex sensors and an accelerator. These sensors can be worn by an individual which can detect his gestures and finger movements. The system we have introduced will continuously read these signals and process them to produce a speech and text

conversion which is done using a wireless connection over bluetooth. The values detected by our hardware system are processed by the software where the values are compared with predetermined values to map the gestures to particular alphabets. Each flex sensor's output varies with each bend angle of every finger, hence we are able to map each angle combination to different alphabets. The accelerometer we have used also helps us differentiate between different alphabets of the sign language.

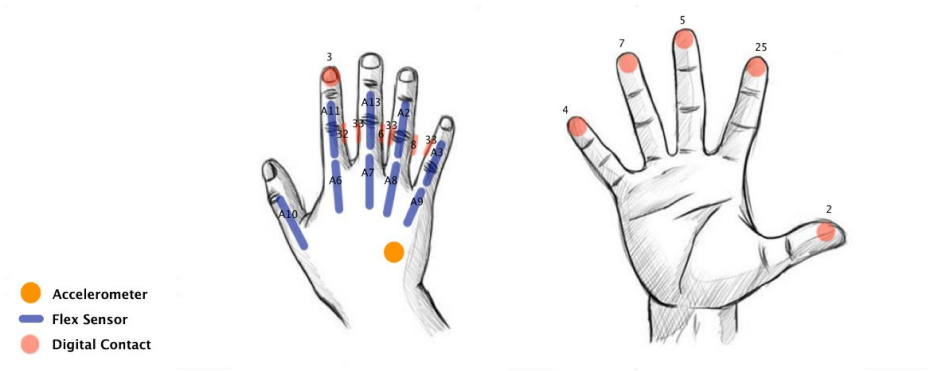

Fig 3.1 Mechanics of hand gestures

3.2 Communication between hearing and non-hearing

Gesturing is the main medium of communication between hearing and non-hearing humans to Sign Language Interpretation can be proven to be very useful as a tool for the deaf and mute people to be a part of the community with no obstacles. In this paper we thus introduce a smart system that provides a wearable hand glove that can detect gestures, transmit the signal and convert that to text and speech. Sensor data is collected and processed by the software and sent to identify each alphabet. It is further sent over Bluetooth to a mobile application that converts the processed alphabet output to text and speech.

CHAPTER 4

PROPOSED DESIGN AND SPECIFICATIONS

4.1 User Requirements and Design Specifications

Easiness of Usage: There should be minimal problems faced by user while using the glove in their daily lives. Delay of the translation should also be taken care of as it is an important feature in usage. There should be minimal tasks given to the user while using the glove in order to process the sign alphabet

Cost Efficiency: For these devices to be accessible to the general public, cost is a very important consideration for the designers of the glove.

Lightweight, compact & convenient- The physical aspects of the glove also define how easy it is to use. The weight and dimension should be taken into consideration inorder to provide maximum usability.

Accuracy- This is the most important factor. The accuracy of the device defines how many users will continue using such a device

4.2 Design Outline

Our design outline specifies a simplification of our entire model to understand the basic working principle of our design. It consists of a smart glove with 5 flex sensors and one accelerometer connected to a microcontroller that can process the signals and is programmed using software. The flex angle values are then used to identify the letter, which is sent via a bluetooth signal to a smartphone. The bluetooth ensures minimal delay and hence we get an instant text and speech conversion that can be heard and read on our smartphone. market appeal, and accuracy.

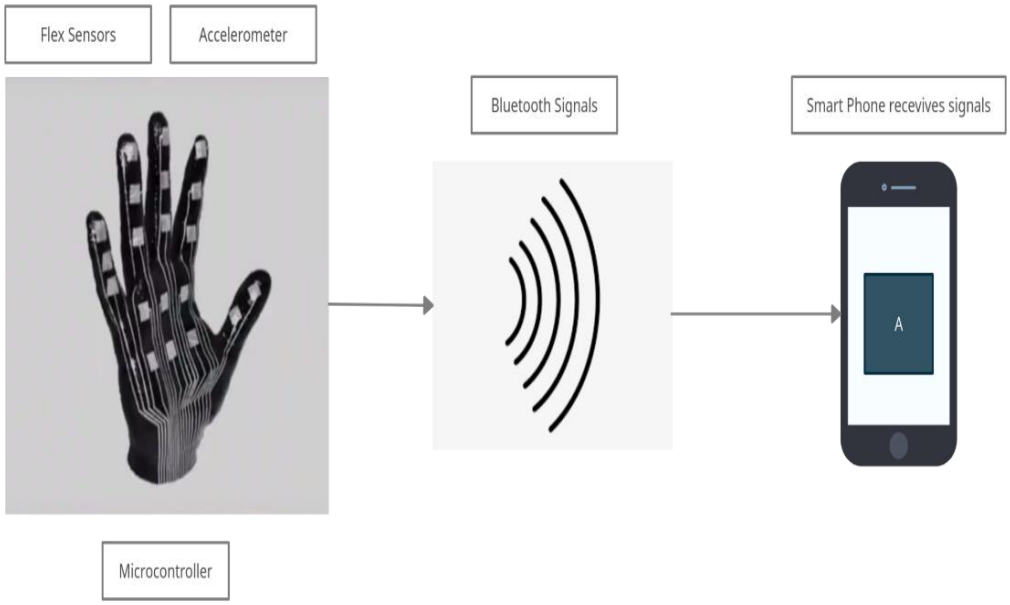

Figure 4.1 Flowchart of the Design outline

4.3 Design Selection

We have assessed the model requirements in detail in order to select the appropriate design elements. The elements that were required to be selected were

Flex Sensors

The requirements of an efficient flex sensor is that they should be accurate in detecting the bend angle of each finger. The project basically creates a letter recognition unit with sensors to track fingers movements. The accuracy is the most crucial in regards to this particular element. The design of a flex sensor is done by the use of materials like

plastic and carbon. The carbon surface is placed on a plastic strip such that when the sensor is turned aside its resistance will change. This bend and change of resistance is proportional to each other and hence accuracy will be maintained with such a mechanism.

Accelerometer

Some specifications to consider while selecting an accelerometer is:

Sensitivity (mV/g or LSB/g):

Sensitivity is a measure of the minimal detectable sign of the trade in output electric sign in line with a change in input mechanical alternation. This is legitimate in one frequency simplest.

Frequency response (Hz):

This is the frequency range targeted with a tolerance band (±5%, and so forth) for which the sensor will locate movement and report a real output. The detailed band tolerance we could calculate how a whole lot of the tool's sensitivity deviates from the reference sensitivity at any frequency inside its exact frequency range.

Bandwidth (Hz):

The bandwidth of a sensor shows the variety of vibration frequencies to which the accelerometer responds or how regularly a dependable studying may be taken. Humans cannot create movement much beyond the range of 10Hz to12Hz. For this purpose, a bandwidth of 40Hz to 60Hz is alright for identifying human movement.

Voltage noise density (µg/SQRT Hz):

The change of voltage noise with the inverse rectangular root of the bandwidth. The faster that we examine accelerometer adjustments, the more severe accuracy we get. Noise has a higher impact on the overall performance of the accelerometers whilst working at decreasing conditions with a smaller output sign.

Zero-g voltage:

This term specifies the variety of voltages that can be predicted on the output underneath 0g of acceleration.

Dynamic range (g):

This is the variety between the smallest detectable amplitude that the accelerometer can degree to the most important amplitude before distorting or clipping the output signal

Microcontroller

The new generation has a wide variety of microcontrollers available. We can use highly integrated microcontrollers with

- Built in switches
- Low-noise receiver amplifiers
- Filters
- Power management modules
- minimal PCB requirements

- system that can perform as a slave device to a host MCU or a complete standalone system
- WiFi- or Bluetooth interface to interact with several different devices conveniently

.

CHAPTER 5

HARDWARE DEVELOPMENT

5.1 Flex sensors

5.1.1 Features

- Angle Displacement Measurement
- Bends and Flexes physically with motion device
- Possible Uses - Robotics
- Gaming (Virtual Motion)
- Medical Devices - Computer Peripherals
- Musical Instruments
- Physical Therapy
- Simple Construction
- Low Profile

5.1.2 Mechanical specifications

- Life Cycle: >1 million
- Height: 0.43mm (0.017")
- Temperature Range: -35°C to +80°C
-

5.1.3 Electrical specifications

- Flat Resistance: 10K Ohms
- Resistance Tolerance: ±30%
- Bend Resistance Range: 60K to 110K Ohms
- Power Rating : 0.50 Watts continuous. 1 Watt Peak

5.1.4 How it works

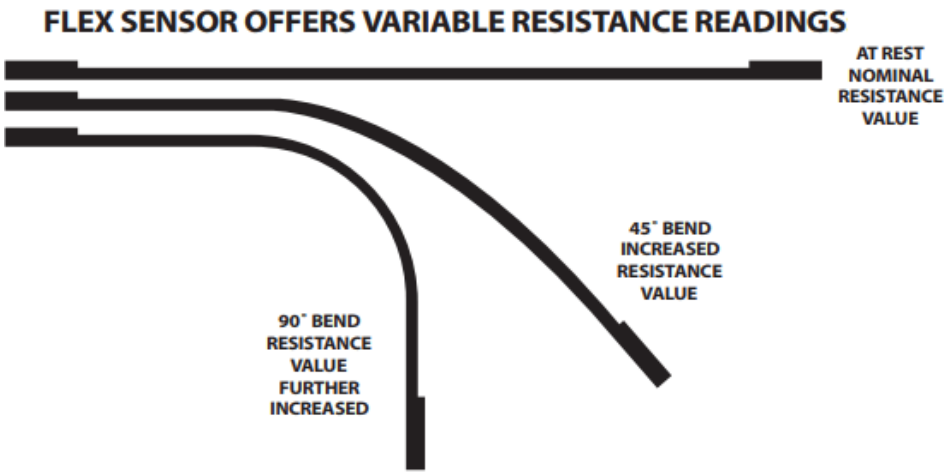

Fig 5.1 working of flex sensor

5.1.5 Schematics

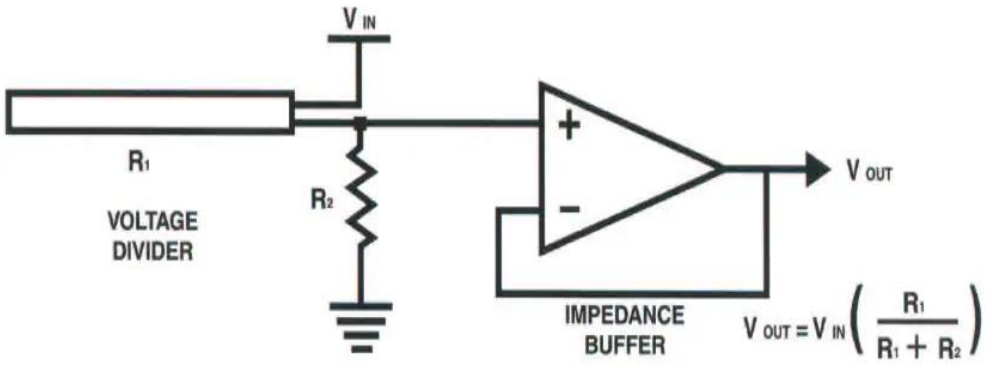

Fig 5.2 basic flex sensor circuit

- When using a flex sensor as a voltage divider, the output voltage rises as the sensor bends.
- The impedance buffer in the circuit decreases the error caused by the flex sensor's source impedance acting as a voltage divider.
- The LM358 or LM324 are recommended op amps.
- We can also use an easy test to test our flex sensor and skip the op-amp.

5.2 Accelerometer

5.2.1 Features

- Measurement Range
- Accelerometer Sensitivity
- Nonlinearity
- Package Alignment Error
- (Orthogonal) Alignment Error
- Cross-Axis Sensitivity
- Accelerometer Noise Density

5.2.2 How it works

Accelerometers are electromechanical instruments that detect static or dynamic acceleration forces. Gravity is a static force, whereas vibrations and movement are dynamic forces.

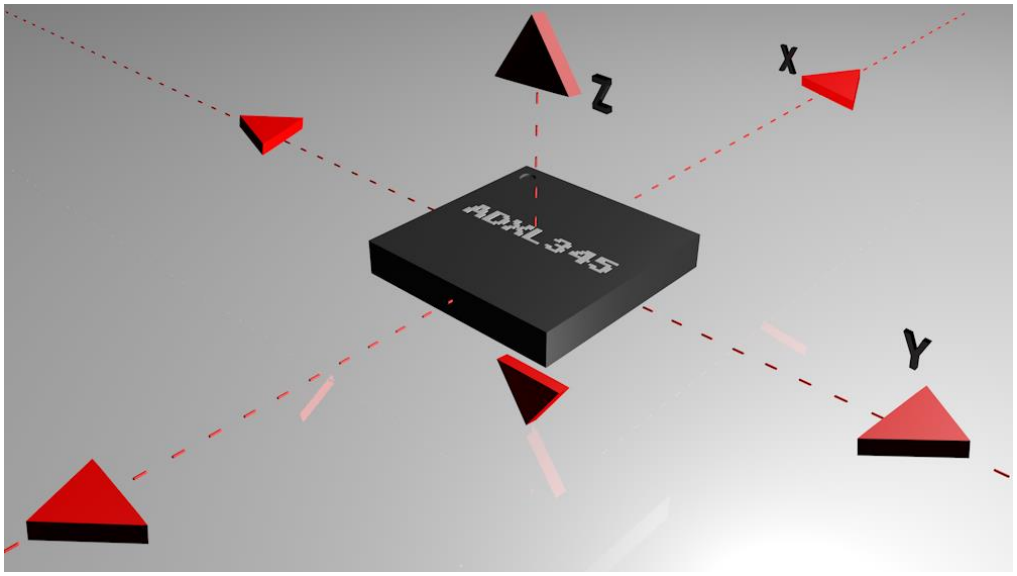

Fig 5.3 Axes of measurement for an accelerometer

Accelerometers can detect movement on one, two, or three axes. As the cost of developing 3-axis machines falls, they are becoming more popular.

Internally, most accelerometers have capacitive plates. Some are stationary, while others are attached to microscopic springs that move internally while the sensor is subjected to acceleration forces. The capacitance between these plates changes as they move in regard to each other. The acceleration may be calculated using these variations in capacitance.

Other types of accelerometers are based on piezoelectric materials. When mechanical tension is applied to these small crystal formations, they produce an electrical charge.

5.2.3 Benefits

- Every step you take is accurately counted by the accelerometer. Some are the size of a quarter and can be clipped to your shoe, so you don't even notice they're there.
- The computer application shows you not only how many steps you've taken, but also when you took them. It's crucial to figure out when you're the most active.
- You can check the pace of your walking or running steps thanks to the accelerometer.
- The programme retains track of past weeks and months because fitness goals should always be long-term. Understanding the successes and possible areas for growth requires the ability to see trends over time.
- Knowing how many calories you burn in a day can help you determine the amount of calories you need to lose weight.

5.3 ESP 32

• 5.3.1 Features

1. Processors:
   - CPU: Xtensa dual-core (or single-core) 32-bit LX6 microprocessor, operating at 160 or 240 MHz and performing at up to 600 DMIPS
   - Ultra low power (ULP) co-processor
2. Memory: 320 KiB RAM, 448 KiB ROM
3. Wireless connectivity:
   - Wi-Fi: 802.11 b/g/n
   - Bluetooth: v4.2 BR/EDR and BLE (shares the radio with Wi-Fi)
4. Peripheral interfaces:
   - 34 × programmable GPIOs
   - 12-bit SAR ADC up to 18 channels
   - 2 × 8-bit DACs
   - 10 × touch sensors (capacitive sensing GPIOs)
   - 4 × SPI
   - 2 × I²S interfaces
   - 2 × I²C interfaces
   - 3 × UART
   - SD/SDIO/CE-ATA/MMC/eMMC host controller
   - SDIO/SPI slave controller
   - Ethernet MAC interface with dedicated DMA and IEEE 1588 Precision Time Protocol support
   - CAN bus 2.0
   - Infrared remote controller (TX/RX, up to 8 channels)

   - Motor PWM
   - LED PWM (up to 16 channels)
   - Hall effect sensor

- Ultra low power analog preamplifier

5. Security:

- IEEE 802.11 standard security features all supported, including WPA, WPA2, WPA3 (depending on version)[4] and WAPI
- Secure boot
- Flash encryption
- 1024-bit OTP, up to 768-bit for customers
- Cryptographic hardware acceleration: AES, SHA-2, RSA, elliptic curve cryptography (ECC), random number generator (RNG)

6. Power management:

- Internal low-dropout regulator
- Individual power domain for RTC
- 5 μA deep sleep current
- Wake up from GPIO interrupt, timer, ADC measurements, capacitive touch sensor interrupt.

5.3.2 Pin configuration

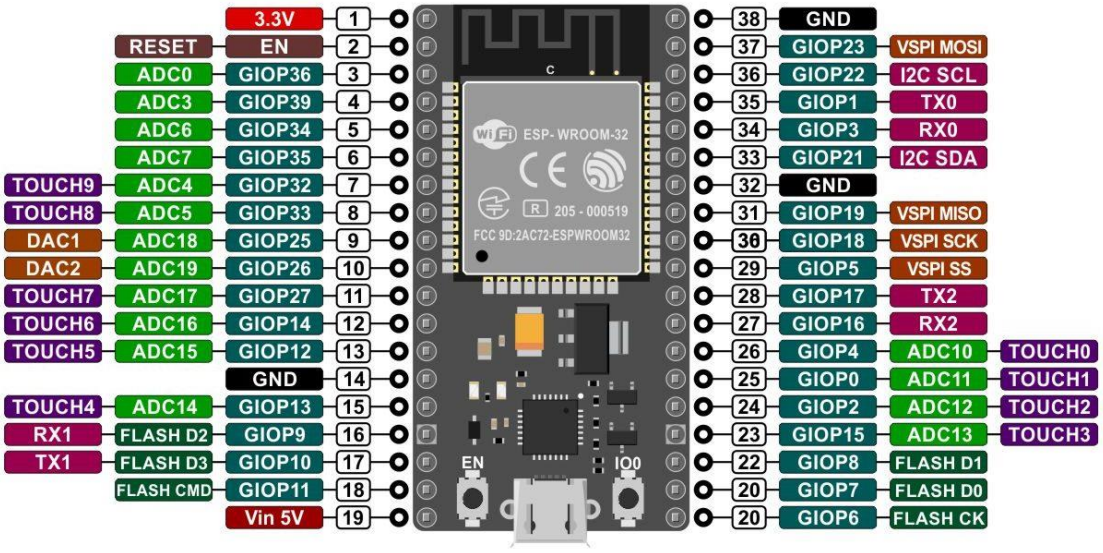

Fig 5.4 Pinout diagram of Esp32

The 30-pin variant of the ESP32 Development Board seen above is one of the most common today. It includes the ESP-WROOM-32 as the baseboard, as well as a few pins and components for interacting with the ESP32.

As you can see from the illustration, each pin has multiple roles. Before using a pin for a certain activity, double-check its alternative functions.

A 36-pin version of the ESP32 DevKit Board is also available. This model isn't as popular as the 30-pin model. However, if you have a 36-pin ESP32 board, the pinout below will come in handy.

Except for a few pins at the bottom, the pinout of both the 30-pin and 36-pin variants of the ESP32 Development Boards are nearly similar. SPI Flash IC uses 6 GPIO pins

(GPIO6 to GPIO11) in the 36-pin version. As a result, they should not be utilised for anything else. Finally, one extra pin (GPIO0 – Pin 23) is provided.

5.3.3 Internal block diagram

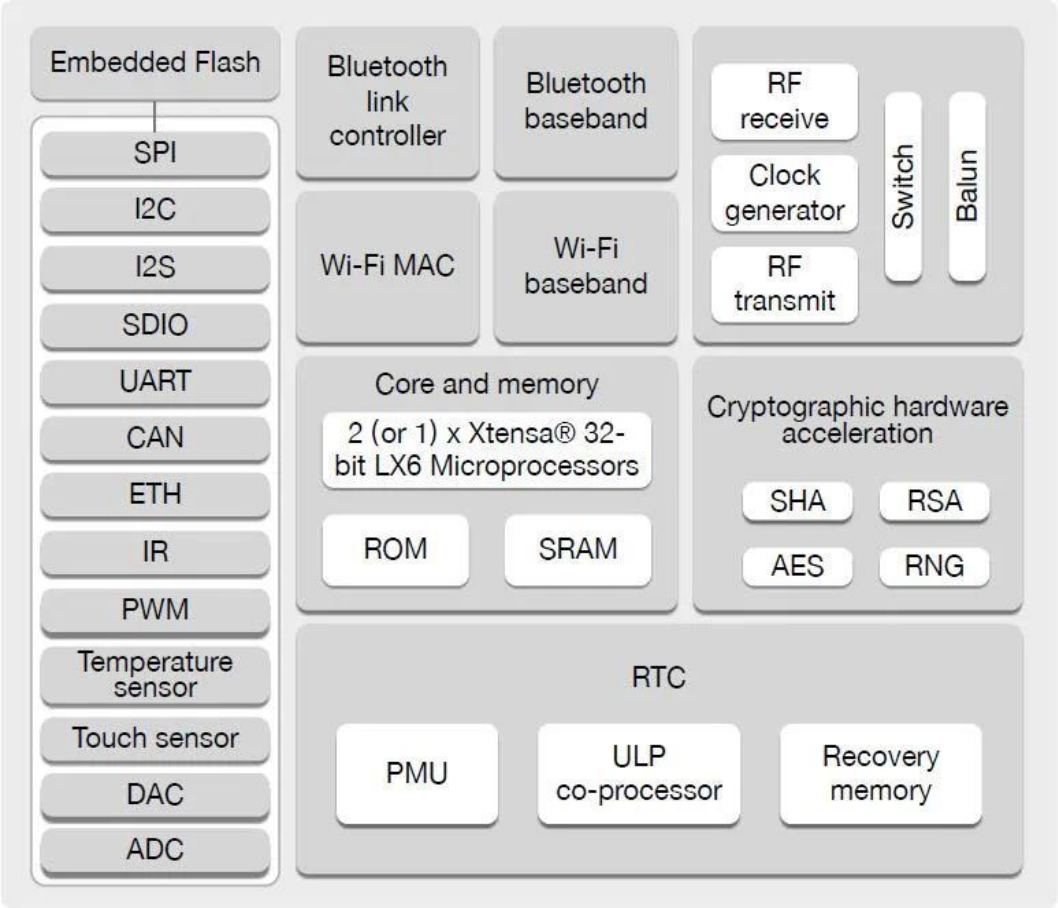

Fig 5.5 Block diagram of esp32

The ESP32 contains a dual core, a chip region that controls WiFi, and a chip area that controls Bluetooth, as seen in this diagram. It also supports hardware encryption acceleration, allowing it to connect to LoRa, a long-distance network that can link up to 15 kilometres using an antenna. We also look at the clock generator, real-time clock, and other components like PWM, ADC, DAC, UART, SDIO, and SPI, to name a few. All of this adds up to a device that is quite comprehensive and functional.

CHAPTER 6

SOFTWARE REQUIREMENTS

SOFTWARE REQUIREMENTS

We required a software program that can read from the flex sensor, accelerometer, and output the corresponding gesture. The goal of this code is to read from each sensor and compare their combined values to a library of signs.

To use the flex sensors, and accelerometer together seamlessly, the best coding program to use would be Arduino IDE, as the program is both flexible and user friendly.

We require an app that can convert Arduino output into speech , and can connect to the board using Bluetooth. The best app to use would be Arduino Bluetooth Text to Speech, as it is available on play store and user friendly.

6.1 ARDUINO IDE

The Arduino Integrated Development Environment (IDE) is a cross-platform application written in the Java programming language (for Windows, macOS, Linux). It originated from the IDE for cabling and processing languages. It includes a code editor with functions such as cut and paste text, find and replace text, automatic indentation, bracket matching, and syntax highlighting, and provides a simple one-click mechanism for compiling and loading programs onto the Arduino board. It also contains a message area, a text console, a toolbar with buttons for frequently used functions, and a hierarchical structure of operating menus. The source code for the IDE is released under the second edition of the GNU General Public License. Arduino IDE uses special code structure rules to support C and C ++ languages. The Arduino IDE provides a software library from the Wiring project, which provides many common input and output programs. User-written code only needs two basic functions to start the sketch and loop of the main program. They are compiled and linked with the main () program stub to form a looping executable program with the GNU toolchain, which is also included on the IDE launch media. The Arduino IDE uses the Arduino program to convert the executable code into a hexadecimal

encoded text file, which the loader in the board firmware loads onto the Arduino board.

Programs written with the Arduino IDE are called sketches.

A minimal Arduino C / C ++ program contains only two functions:

- setup(): This function is called once when a sketch starts after power-up or reset. It is 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.

6.2 ARDUINO BLUETOOTH TEXT TO SPEECH

This smartphone app receives text messages via Bluetooth from a paired Bluetooth device, i.e., our Arduino board with an CH-05 module. It then converts the received text into speech.

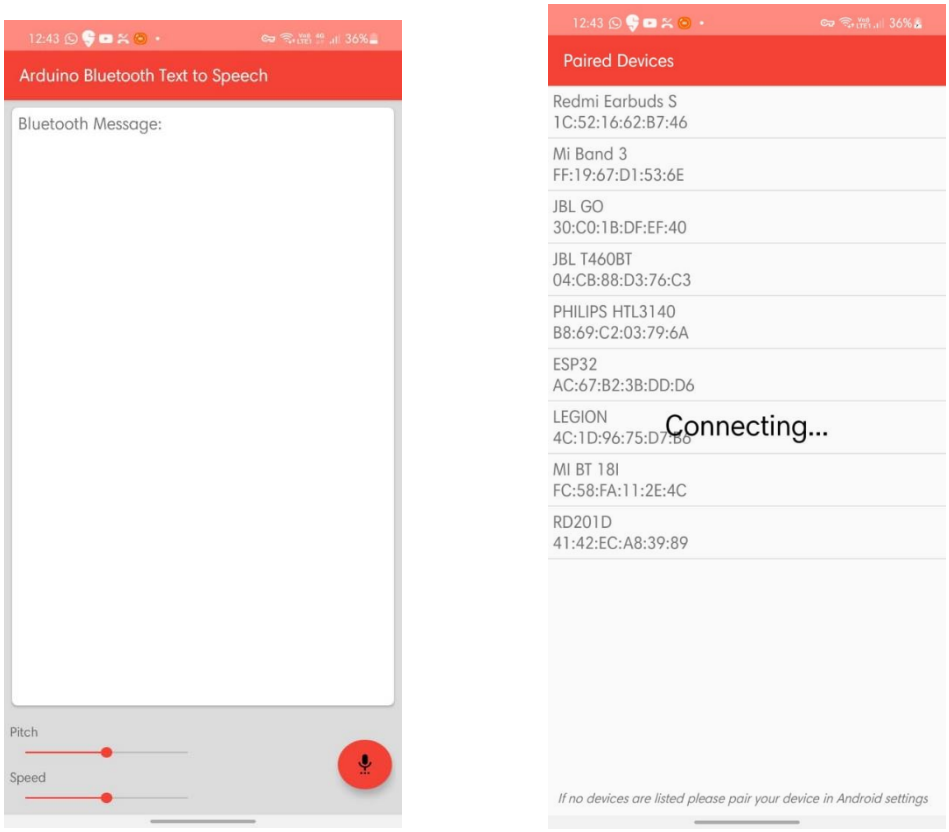

Fig 6.1 Arduino Bluetooth text to speech app interface

CHAPTER 7

DESIGN OF THE SMART GLOVE

7.1. BLOCK DIAGRAM OF SMART GLOVE

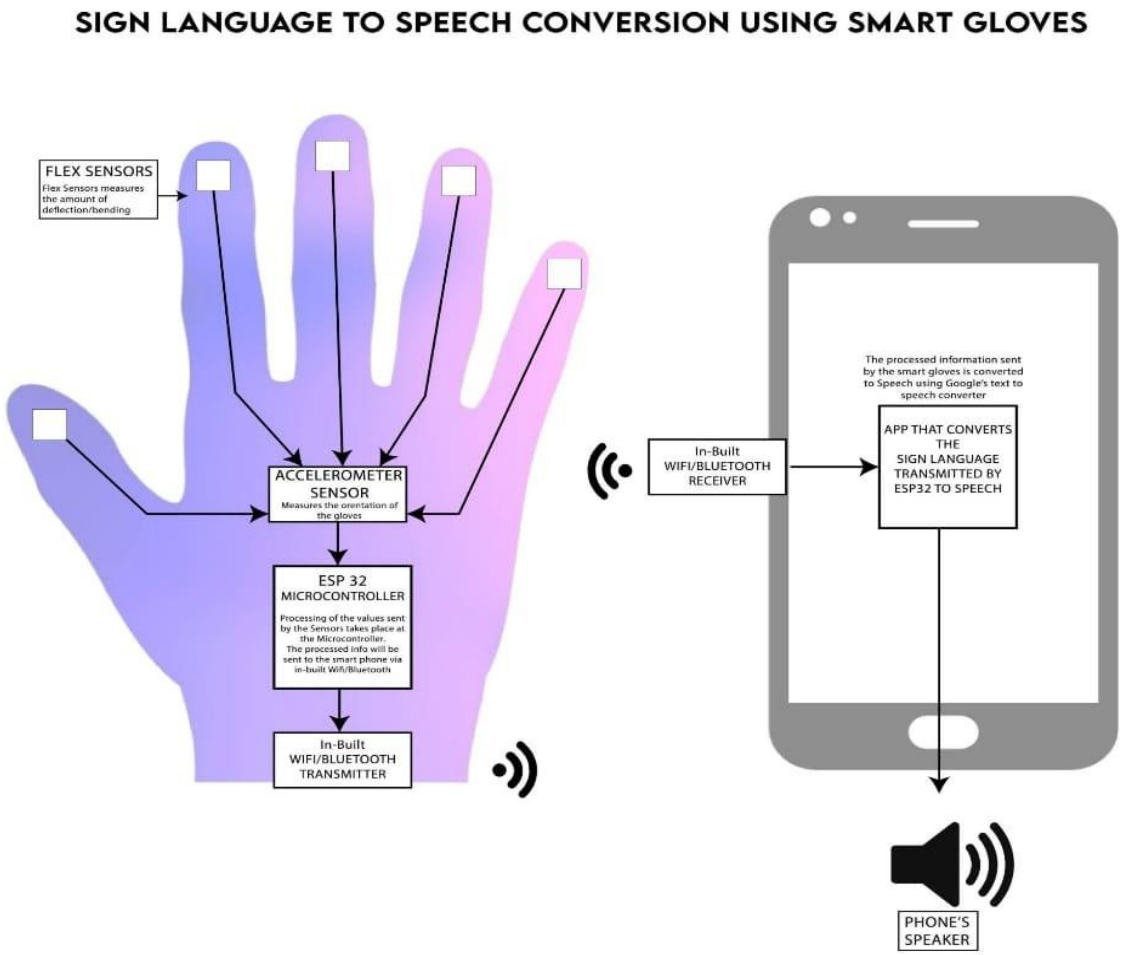

Fig 7.1 Block diagram of the smart glove

The electrical quantity from the flex sensor, and accelerometer processed in the microcontroller for each standard sign gesture. It is then converted to text and then sent to arduino bluetooth text speech app and it is converted into speech.

7.2. Working Principle of the Smart Glove

The working of the speaking glove is as follows:

- Initially, the five flex sensors are attached to five fingers, and an accelerometer is connected to detect the tilt . The flex sensors are a part of the voltage divider circuit, so whenever there is a change in the position of the finger(bend), resistance of the corresponding flex sensors change. This results in change in the voltage drop across the corresponding flex sensors.
- These values along with the values measured by the accelerometer are sent to the ESP32 microcontroller.
- The microcontroller detects that particular sign by analysing the values sent by the sensors.
- The sign detected is sent to the user's mobile phone via in-built bluetooth.
- The phone receives the sign language sent by the microcontroller and the text-to-speech converter app on the phone reads out that particular sign through its speaker.

7.3 Construction of the hand glove

The construction of the hand glove consists of the following steps:

- First of all, a glove of free size is obtained and on which switching can be done easily.
- Flex sensors are put on the glove and glued so that it stays with the glove .
- A voltage divider circuit is designed for the flex sensor connection. The resistor used for the voltage divider is 10 k ohms.
- Connections of the flex sensor are given to the voltage divider circuit and the resulting connections from the circuit from the circuit is taken.

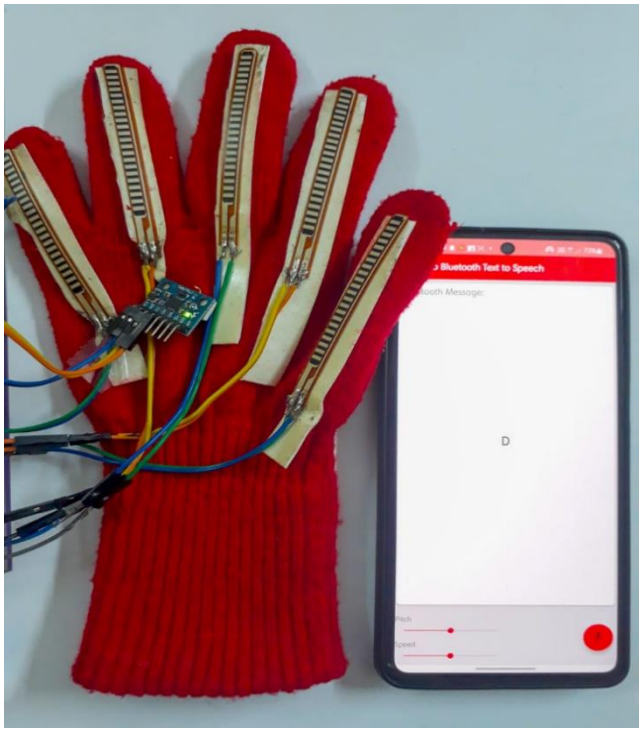

Fig 7.2 The construction of the smart glove.

7.3. Flow charts

7.3.1. Flow chart of the smart glove

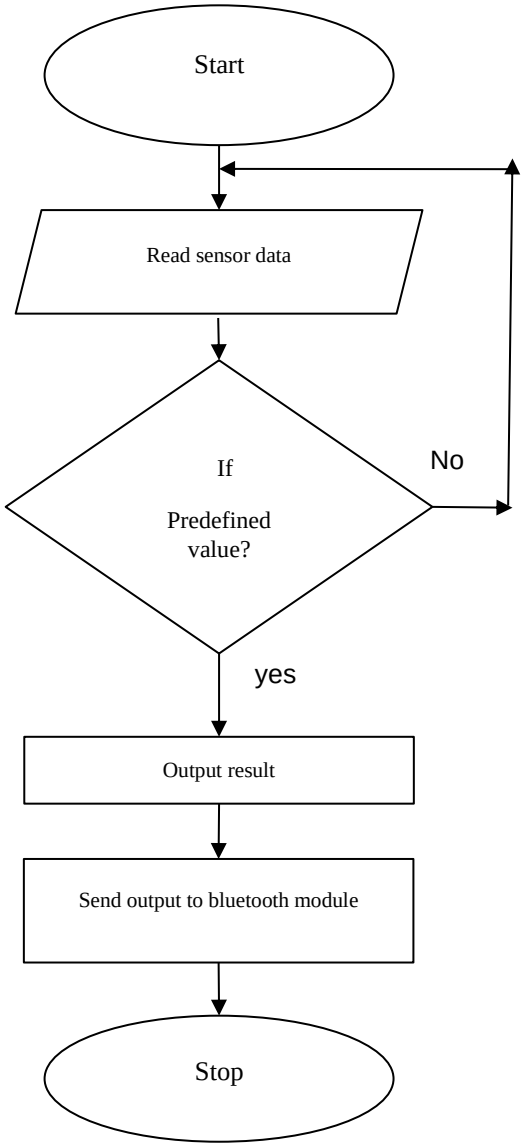

Fig 7.3 Flow chart of the smart glove

7.3.2. Flow chart for Android Application

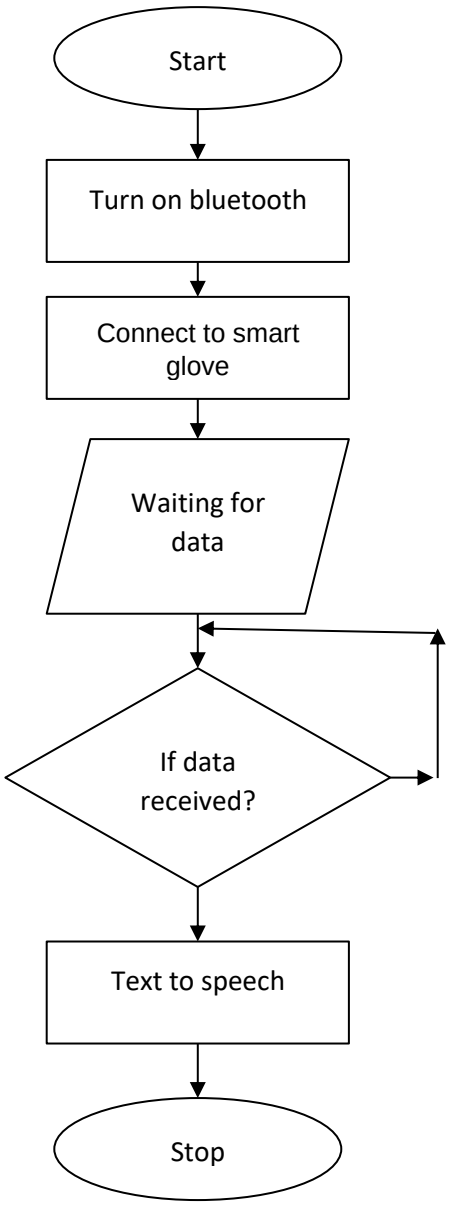

Fig 7.4 Flow chart of android application

7.4 Code :

```
#include "BluetoothSerial.h"

#if !defined(CONFIG_BT_ENABLED)
!defined(CONFIG_BLUEDROID_ENABLED)

#error Bluetooth is not enabled! Please run `make menuconfig` to and enable it

#endif

#include <Adafruit_MPU6050.h>

#include <Adafruit_Sensor.h>

#include <Wire.h>

Adafruit_MPU6050 mpu;

BluetoothSerial SerialBT;

int received;

const int flexPin1 = A6; // Pin connected to voltage divider output

const int flexPin2 = A7;

const int flexPin3 = A4;

const int flexPin4 = A5;

const int flexPin5 = A18;

// Change these constants according to your project's design

const float VCC = 3.3; // voltage at Ardunio 5V line

const float R_DIV = 2000.0; // resistor used to create a voltage divider
```

```
const float flatResistance = 3070.0; // resistance when flat

const float bendResistance = 5000.0; // resistance at 90 deg

void setup() {

Serial.begin(115200);

while (!Serial)

delay(10); // will pause Zero, Leonardo, etc until serial console opens

Serial.println("Adafruit MPU6050 test!");

// Try to initialize!

if (!mpu.begin()) {

Serial.println("Failed to find MPU6050 chip");

while (1) {

delay(10);

}

}

Serial.println("MPU6050 Found!");

mpu.setAccelerometerRange(MPU6050_RANGE_8_G);

Serial.print("Accelerometer range set to: ");

switch (mpu.getAccelerometerRange()) {

case MPU6050_RANGE_2_G:

Serial.println("+-2G");

break;

case MPU6050_RANGE_4_G:
```

```
Serial.println("+-4G");
break;
case MPU6050_RANGE_8_G:
Serial.println("+-8G");
break;
case MPU6050_RANGE_16_G:
Serial.println("+-16G");
break;
}
mpu.setGyroRange(MPU6050_RANGE_500_DEG);
Serial.print("Gyro range set to: ");
switch (mpu.getGyroRange()) {
case MPU6050_RANGE_250_DEG:
Serial.println("+- 250 deg/s");
break;
case MPU6050_RANGE_500_DEG:
Serial.println("+- 500 deg/s");
break;
case MPU6050_RANGE_1000_DEG:
Serial.println("+- 1000 deg/s");
break;
case MPU6050_RANGE_2000_DEG:
```

```
Serial.println("+- 2000 deg/s");
break;
}
mpu.setFilterBandwidth(MPU6050_BAND_21_HZ);
Serial.print("Filter bandwidth set to: ");
switch (mpu.getFilterBandwidth()) {
case MPU6050_BAND_260_HZ:
Serial.println("260 Hz");
break;
case MPU6050_BAND_184_HZ:
Serial.println("184 Hz");
break;
case MPU6050_BAND_94_HZ:
Serial.println("94 Hz");
break;
case MPU6050_BAND_44_HZ:
Serial.println("44 Hz");
break;
case MPU6050_BAND_21_HZ:
Serial.println("21 Hz");
break;
case MPU6050_BAND_10_HZ:
```

```
Serial.println("10 Hz");
break;
case MPU6050_BAND_5_HZ:
Serial.println("5 Hz");
break;
}
Serial.println("");
delay(100);
SerialBT.begin("ESP32");
Serial.println("Start pairing!");
pinMode(flexPin1, INPUT);
pinMode(flexPin2, INPUT);
pinMode(flexPin3, INPUT);
pinMode(flexPin4, INPUT);
pinMode(flexPin5, INPUT);
}
void loop() {
 /* Get new sensor events with the readings */
sensors_event_t a, g,;
mpu.getEvent(&a, &g);
/* Print out the values */
Serial.print("Acceleration X: ");
```

```
Serial.print(a.acceleration.x);

delay(500);

// Read the ADC, and calculate voltage and resistance from it

int ADCflex1 = analogRead(flexPin1);

float Vflex1 = ADCflex1 * VCC / 1023.0;

float Rflex1= -R_DIV * (VCC / Vflex1 - 1.0);

//Serial.println("Resistance: " + String(Rflex) + " ohms");

//delay(1000);

// Use the calculated resistance to estimate the sensor's bend angle:

float angle1 = map(Rflex1, flatResistance, bendResistance, 0, 90.0);

//Serial.println("Bend1: " + String(angle1) + " degrees");

/* if(int(angle1)<=-189 && int(angle1)>=-194)

{

// SerialBT.println("HELP");

}*/

Serial.println();

if (Serial.available()) {

SerialBT.write(ADCflex1);

if (SerialBT.available()) {

SerialBT.println(angle1);

}

}
```

```
 // delay(2000);
 int ADCflex2 = analogRead(flexPin2);
float Vflex2 = ADCflex2 * VCC / 1023.0;
float Rflex2 = -R_DIV * (VCC / Vflex2 - 1.0);
//Serial.println("Resistance: " + String(Rflex2) + " ohms");
 // Use the calculated resistance to estimate the sensor's bend angle:(-200 to -300)
float angle2 = map(Rflex2, flatResistance, bendResistance, 0, 90.0);
//Serial.println("Bend2: " + String(angle2) + " degrees");
//Serial.println();
/*if(int(angle2)<=-200 && int(angle2)>=-300)
{
SerialBT.println("FOOD");
}*/
Serial.println();
if (Serial.available()) {
SerialBT.write(ADCflex2);
if (SerialBT.available()) {
SerialBT.println(angle2);
}
}
int ADCflex3 = analogRead(flexPin3);
```

```
float Vflex3 = ADCflex3 * VCC / 1023.0;

float Rflex3 = -R_DIV * (VCC / Vflex3 - 1.0);

//Serial.println("Resistance: " + String(Rflex3) + " ohms");

//

// // Use the calculated resistance to estimate the sensor's bend angle:(-100 to -150)

float angle3 = map(Rflex3, flatResistance, bendResistance, 0, 90.0);

//Serial.println("Bend3: " + String(angle3) + " degrees");

//Serial.println();

/* if(int(angle3)<=-100 && int(angle3)>=-150)

{

}*/

Serial.println();

if (Serial.available()) {

SerialBT.write(ADCflex3);

if (SerialBT.available()) {

SerialBT.println(angle3);

}

}

// delay(500);

int ADCflex4 = analogRead(flexPin4);

float Vflex4 = ADCflex4 * VCC / 1023.0;

float Rflex4 = -R_DIV * (VCC / Vflex4 - 1.0);
```

```
//Serial.println("Resistance: " + String(Rflex4) + " ohms");

//

// // Use the calculated resistance to estimate the sensor's bend angle:

float angle4 = map(Rflex4, flatResistance, bendResistance, 0, 90.0);

//Serial.println("Bend4: " + String(angle4) + " degrees");

Serial.println();

//

// delay(500);

int ADCflex5 = analogRead(flexPin5);

float Vflex5 = ADCflex5 * VCC / 1023.0;

float Rflex5 = -R_DIV * (VCC / Vflex5 - 1.0);

// Serial.println("Resistance: " + String(Rflex5) + " ohms");

//

// // Use the calculated resistance to estimate the sensor's bend angle:(-90 to -120)

float angle5 = map(Rflex5, flatResistance, bendResistance, 0, 90.0);

// Serial.println("Bend5: " + String(angle5) + " degrees");

//Serial.println();

/* if(int(angle5)<=-90 && int(angle5)>=-120)

{

SerialBT.println("WATER");

}*/

Serial.println();
```

```
if (Serial.available()) {

SerialBT.write(ADCflex5);

if (SerialBT.available()) {

SerialBT.println(angle5);

}

}

if(int(a.acceleration.x)> 7)

{

if(int (angle1)<-110 && int(angle1)>-160 && int (angle2)<-190 && int (angle3)<-180
&& int (angle4)<-150 && int (angle5)<-150 )

{

SerialBT.println("A");

delay(500);

}

else if(int (angle1)<-170 && int (angle2)<-125 && int(angle2)>-210 && int
(angle3)<-120 && int (angle3)>-160  && int (angle4)<-115 && int (angle4)>-140 &&
int (angle5)<-115 && int (angle5)>-145)

{

SerialBT.println("B");

delay(500);

}

else if(int (angle1)<-120 && int (angle2)<-150 && int (angle3)<-150 && int
(angle4)<-135 && int (angle5)<-120)
```

```
{
SerialBT.println("C");
delay(500);
}
else if(int (angle1)<-140 && int (angle2)<-120 && int (angle3)<-160 && int (angle4)<-140 && int (angle5)<-140 )
{
SerialBT.println("D");
delay(500);
}
/*else if(int (angle1)<-160  && int (angle2)<-200 && int (angle3)<-170 && int (angle4)<-150  && int (angle5)<-170)
{
SerialBT.println("E");
delay(500);
}
*/
else if(int (angle1)<-164 && int (angle2)<-220 && int (angle3)<-129 && int (angle4)<-120 && int (angle5)<-114)
{
SerialBT.println("F");
delay(500);
}
```

```
/*else if(int (angle1)<-119 && int (angle2)<-130 && int (angle3)<-200 && int
(angle4)<-170 && int (angle5)<-170)

{

SerialBT.println("G");

delay(500);

}

*/

else if(int (angle1)<-145 && int (angle2)<-120 && int (angle3)<-124 && int
(angle4)<-170 && int (angle5)<-143)

{

SerialBT.println("V");

delay(500);

}

else if(int (angle1)<-160 && int (angle2)<-197 && int (angle3)<-185 && int
(angle4)<-162 && int (angle5)<-110)

{

SerialBT.println("I");

delay(500);

}

else if(int (angle1)<-130 && int (angle2)<-121 && int (angle3)<-124 && int
(angle4)<-165 && int (angle5)<-145)

{

SerialBT.println("K");
```

```
delay(500);

}

else if(int (angle1)<-115 && int (angle2)<-127 && int (angle3)<-124 && int
(angle4)<-170 && int (angle5)<-130)

{

SerialBT.println("L");

delay(500);

}

else if(int (angle1)<-152 && int (angle2)<-127 && int (angle3)<-125 && int
(angle4)<-177 && int (angle5)<-119)

{

SerialBT.println("M");//ZH chinese

delay(500);

}

else if(int (angle1)<-174 && int (angle2)<-123 && int (angle3)<-205 && int
(angle4)<-168 && int (angle5)<-120)

{

SerialBT.println("T");//chinese

delay(500);

}

else if(int (angle1)<-138 && int (angle2)<-120 && int (angle3)<-120 && int
(angle4)<-125 && int (angle5)<-140)

{
```

```
SerialBT.println("W");

delay(500);

}

else if(int (angle1)<-171 && int (angle2)<-148 && int (angle3)<-202 && int
(angle4)<-175 && int (angle5)<-174)

{

SerialBT.println("X");

delay(500);

}

else if(int (angle1)<-108 && int (angle2)<-189 && int (angle3)<-200 && int
(angle4)<-160 && int (angle5)<-110)

{

SerialBT.println("Y");//

delay(500);

}

if(int (angle1)<-160  && int (angle2)<-200 && int (angle3)<-170 && int (angle4)<-
150  && int (angle5)<-170)

{

SerialBT.println("S");

delay(500);

}

}

//horizontal
```

```
else if(int(a.acceleration.x)<3 )

{

if(int (angle1)<-160  && int (angle2)<-200 && int (angle3)<-170 && int (angle4)<-
150  && int (angle5)<-170)

{

SerialBT.println("E");

delay(500);

}

else if(int (angle1)<-115 && int (angle2)<-127 && int(angle2)>-147 && int
(angle3)<-124 && int (angle4)<-170 && int (angle5)<-130)

{

SerialBT.println("G");

delay(500);

}

else if(int (angle1)<-145 && int (angle2)<-120 && int (angle3)<-124 && int
(angle4)<-170 && int (angle5)<-143)

{

SerialBT.println("H");

delay(500);

}

else if(int (angle1)<-160 && int (angle2)<-197 && int (angle3)<-185 && int
(angle4)<-162 && int (angle5)<-110)

{
```

```
SerialBT.println("I");

delay(500);

}

else if(int (angle1)<-174 && int (angle2)<-123 && int (angle3)<-205 && int
(angle4)<-168 && int (angle5)<-120)

{

SerialBT.println("U");//chinese ,horizontal of T

delay(500);

}

else if(int (angle1)<-130 && int (angle2)<-121 && int (angle3)<-124 && int
(angle4)<-165 && int (angle5)<-145)

{

SerialBT.println("P");

delay(500);

}

else if(int (angle1)<-164 && int (angle2)<-220 && int (angle3)<-129 && int
(angle4)<-120 && int (angle5)<-114)

{

SerialBT.println("Q");//horizontal of F

delay(500);

}

/*else if(int (angle1)<-140 && int (angle2)<-120 && int (angle3)<-160 && int
(angle4)<-140 && int (angle5)<-140 )
```

```
{
SerialBT.println("Z");
delay(500);
}
*/
else if(int (angle1)<-152 && int (angle2)<-127 && int (angle3)<-125 && int (angle4)<-177 && int (angle5)<-119)
{
SerialBT.println("N");//ZH chinese
delay(500);
}
else if(int (angle1)<-108 && int (angle2)<-189 && int (angle3)<-200 && int (angle4)<-160 && int (angle5)<-110)
{
SerialBT.println("J");//
delay(500);
}
}
}
```

CHAPTER 8

RESULT ANALYSIS

8. RESULT ANALYSIS

After having completed the design of the glove, we ran tests on it to ensure that it met our preliminary design specifications and development objectives. The preliminary specifications included ease of use, portability, affordability, reliability, and aesthetics. The development objectives were followed and achieved in the construction of the glove.

8.1. Ease to use

Ease of use is a basic concept that describes how easily users can use a glove . We developed the glove in such a way that users can begin translation without much difficulty or delay. The glove we made is soft and easy to wear and user-friendly

8.2. Portability

From the outset of this project, we wanted to ensure that the glove was portable so that it could be used in real-life situations. The glove can be considered portable because it is light in weight and compact.

All of the flex sensors and components are almost as slim as paper. Since the glove is a compact product, it can be easily stored and transported by the user, making it indeed portable.

8.3. Reliability

The most important design specification that had to be met for the completion of this glove was its reliability. Not only was it important that the glove had

someway of outputting a translated sign somewhere, it also had to consistently translate a sign correctly. We made the glove as reliable as possible.

8.4. Aesthetics

If this glove were to be developed into a market product, it would be desirable to have it be aesthetically pleasing so that people would be more readily inclined to purchase it. Ideally, it would look like any other glove. In addition, part of the aesthetics of this glove involves its wearability. This means it should not have any components that hinder the movement of the glove during extended use. The glove is very lightweight and easy to wear, therefore allowing it to be considered.

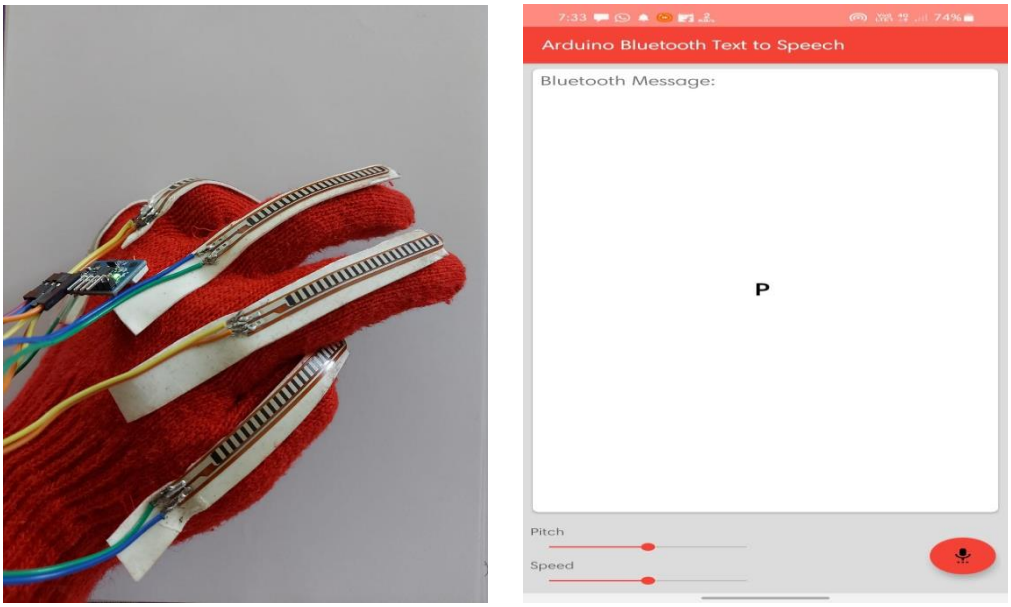

Fig 8.1 Gesture for "P" and the corresponding text is displayed

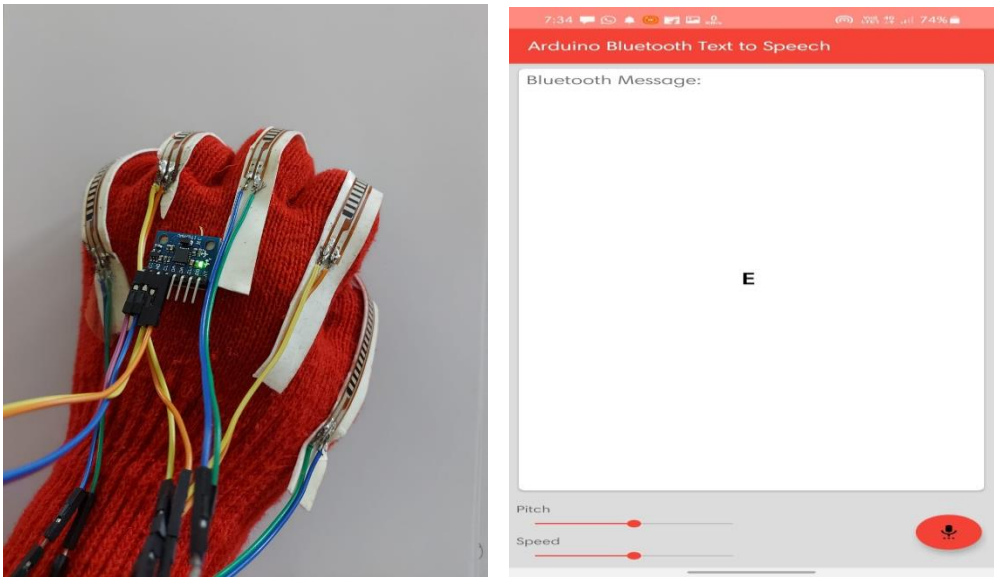

Fig 8.2 Gesture for "E" and the corresponding text is displayed

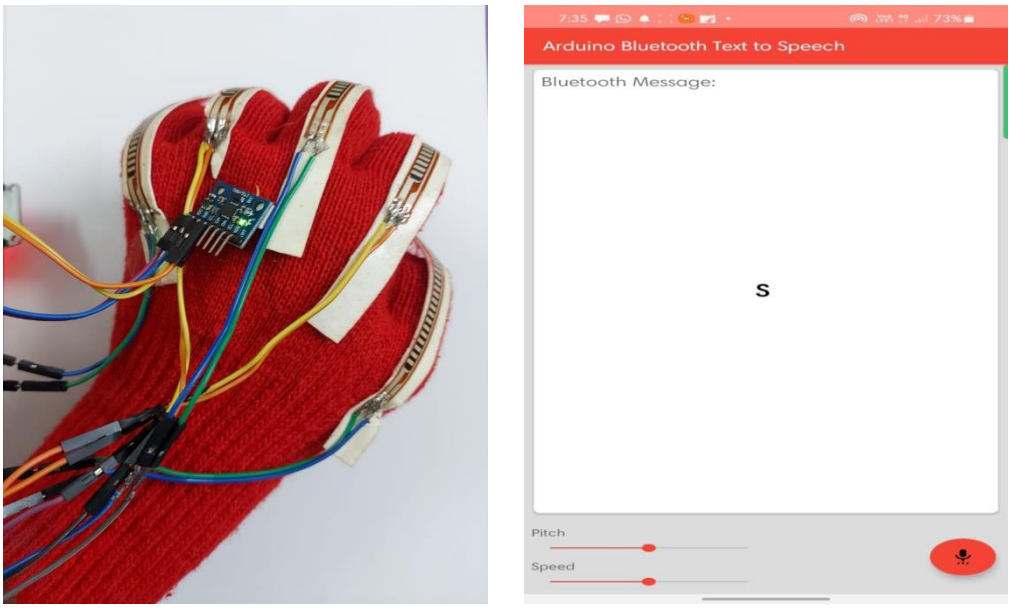

Fig 8.3 Gesture for "S" and the corresponding text is displayed

Note: Gestures for " E" and "S" look the same but the orientation is different.

- For "E" orientation used is horizontal orientation.
- For "S" orientation used is a vertical orientation.

CHAPTER 9

CONCLUSION AND FUTURE SCOPE

CONCLUSION

Most importantly we aim to create a working model that will be able to convert almost all the 26 English letters. We have 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.

We are excited about the progress made in the development of the glove prototype. Although it is not a finished product, it shows that using a glove outfitted with sensors, a microcontroller, and wireless communications can be used to translate ASL signs. It satisfies all of the major requirements put forth by us, and it may lead to further developments in translation devices. With increased attention to the challenge of ASL translation, the team hopes the communication gap between ASL users and the hearing may soon be diminished.

FUTURE SCOPE

If this basic model is complete the future scope of this would be to convert different signs for full words, common actions used in general daily life, and increase the dataset that the model is working on.

We can research and create a vaster dataset and implement it.

10. REFERENCES

1. P Vijayalakshmi, M Aarthi. (2016). " *Sign language to speech conversion* " **Published in:** 2016 IEEE 5th International conference on recent trends in information technology. **INSPEC Accession Number:** 16320657. **DOI:** 10.1109/ICRTIT.2016.7569545.
2. K Abhijit Baskaran , Anoop G Nair , K Deepak Ram , Krishnan Ananth Narayanan . (2017) ."*Smart glove for hand gesture recognition: sign language to speech conversion system*" **Published in:** 2016 International Conference on Robotics and Automation for Humanitarian Applications (RAHA). **INSPEC Accession Number:** 16320657. **DOI:** 10.1109/RAHA.2016.7931887.
3. T. Humphries, An Introduction to the Culture of Deaf People in the United States: Content Notes amp; Reference Manuel for Teachers, Sign Language Studies, vol. 72, no. 10, pp. 209 240, 1991.
4. C. Padden. (2009, September 29). Sign Language Geography [Online]. Available: http://pages.ucsd.edu/ cpadden/files/Padden20SLm] 20Geography.pdf
5. R. Mitchell et al. (2005, February 21). How Many People Use ASL in the United States? Why Estimates Need Updating [Online]. Available: http://research.gallaudet.edu/Publications/ASLusers.pdf
6. Sun robotronics technologies . (2020) . Gesture vocalizer | sign language to speech conversation for deaf and dumb | using arduino Uno . https://youtu.be/60ch5FFG5nI

11. APPENDIX

11.1. GANTT CHART

The Gantt Chart below shows how work was carried out.

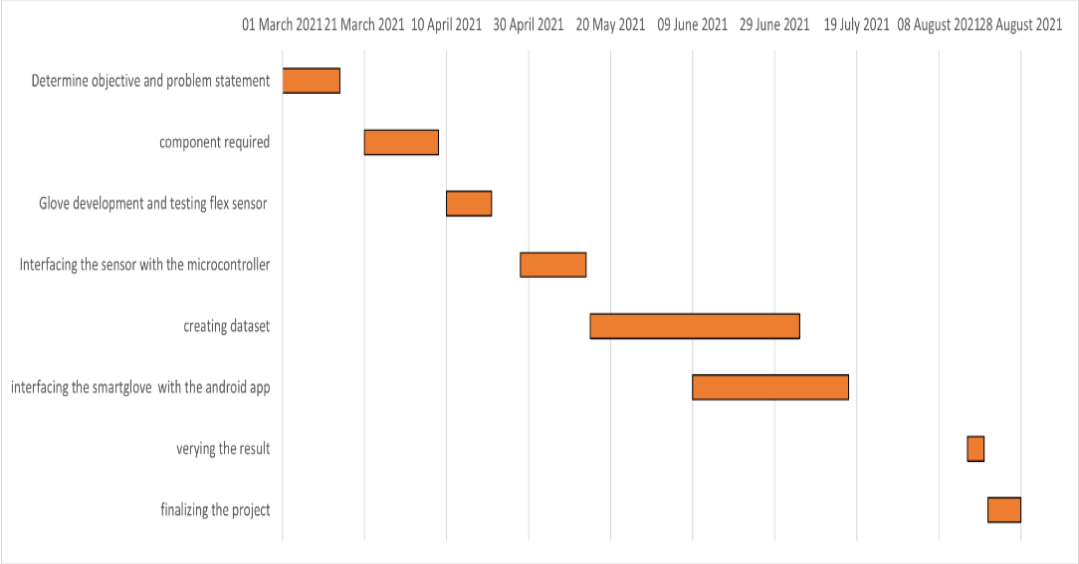

Fig 11.1 Gantt chart

11.2. DATASHEET

11.2.1. ESP32 Pin diagram

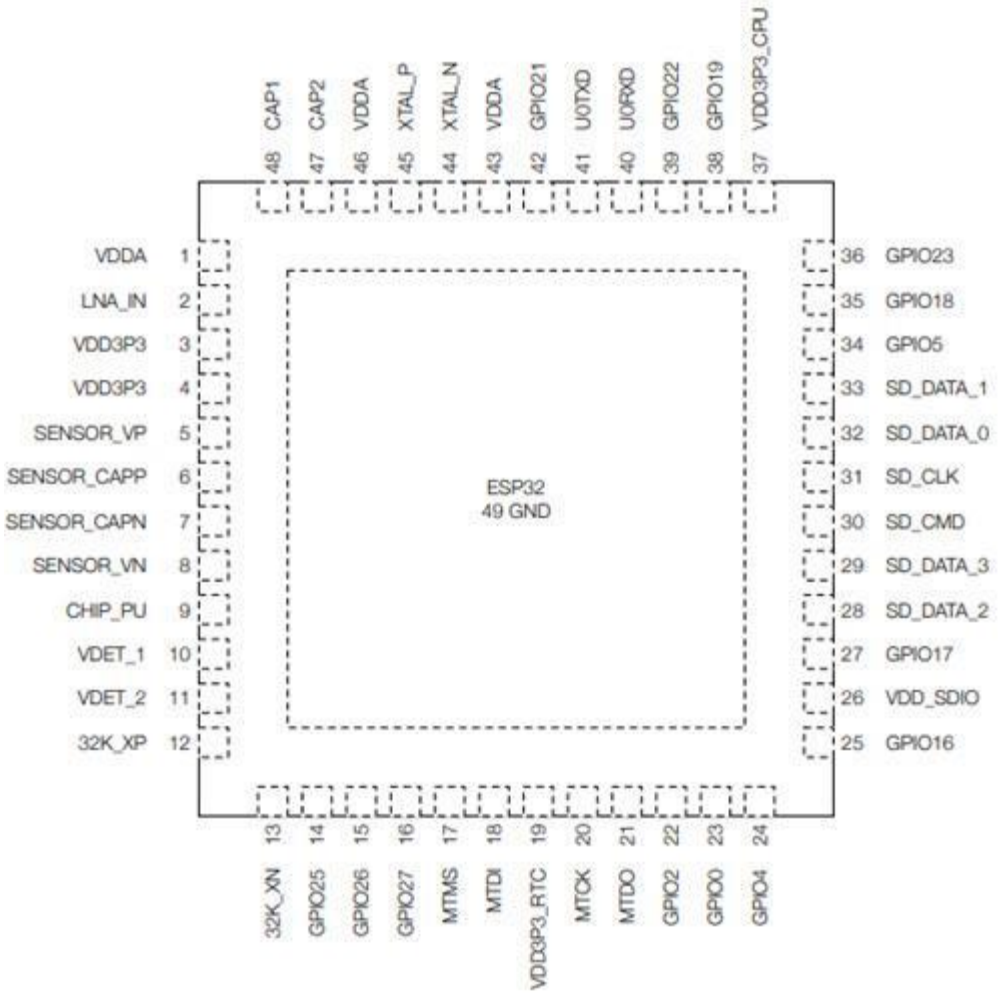

Fig 11.2 Pinout diagram of ESP32

Features

Categories	Items	Specifications
Certification	RF certification	FCC/CE-RED/IC/TELEC/KCC/SRRC/NCC
	Bluetooth certification	BQB
	Green certification	RoHS, REACH
Test	Reliablity	HTOL/HTSL/uHAST/TCT/ESD
Wi-Fi	Protocols	802.11 b/g/n (802.11n up to 150 Mbps)
		A-MPDU and A-MSDU aggregation and 0.4 μs guard interval support
	Frequency range	2.4 GHz ~ 2.5 GHz
Bluetooth	Protocols	Bluetooth v4.2 BR/EDR and BLE specification
	Radio	NZIF receiver with –97 dBm sensitivity
		Class-1, class-2 and class-3 transmitter
		AFH
	Audio	CVSD and SBC
Hardware	Module interfaces	SD card, UART, SPI, SDIO, I^2C, LED PWM, Motor PWM, I^2S, IR, pulse counter, GPIO, capacitive touch sensor, ADC, DAC
	On-chip sensor	Hall sensor
	Integrated crystal	40 MHz crystal
	Operating voltage/Power supply	2.7 ~ 3.6V
	Minimum current delivered by power supply	500 mA
	Recommended operating temperature range	–40°C ~ 65°C
	Package size	(18.00±0.10) mm x (31.40±0.10) mm x (3.30±0.10) mm

Fig 11.4 Features of ESP32

11.2.2. Flex sensor

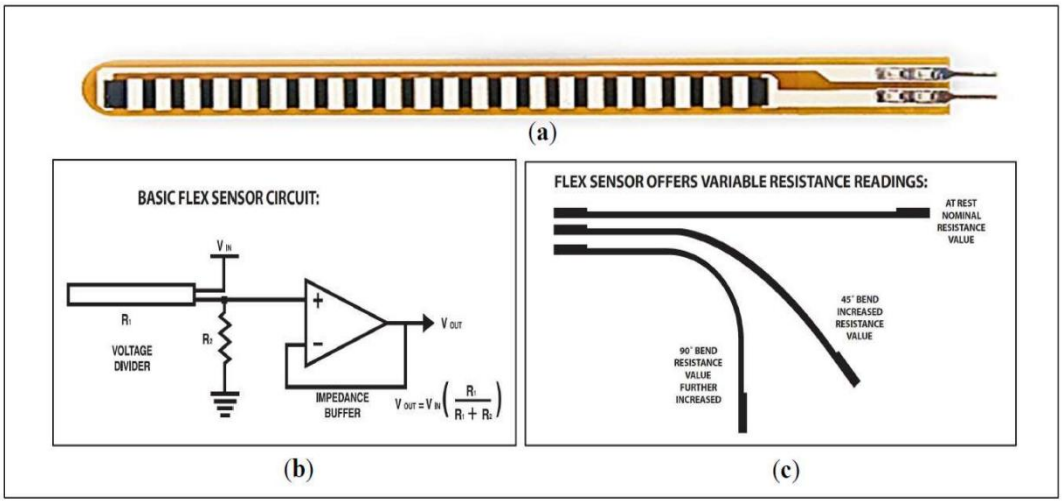

Fig 11.4 Flex sensor circuit

Features

- Angle Displacement Measurement
- Bends and Flexes physically with motion device
- Possible Uses - Robotics
- Gaming (Virtual Motion)
- Medical Devices - Computer Peripherals
- Musical Instruments
- Physical Therapy
- Simple Construction
- Low Profile

Mechanical specifications

- Life Cycle: >1 million
- Height: 0.43mm (0.017")
- Temperature Range: -35°C to +80°C

Electrical specifications

- Flat Resistance: 10K Ohms
- Resistance Tolerance: ±30%
- Bend Resistance Range: 60K to 110K Ohms
- Power Rating : 0.50 Watts continuous. 1 Watt Peak

12. Plagiarism Report

ORIGINALITY REPORT

6%	3%	4%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	P Vijayalakshmi, M Aarthi. "Sign language to speech conversion", 2016 International Conference on Recent Trends in Information Technology (ICRTIT), 2016 Publication	2%
2	ieeexplore.ieee.org Internet Source	1%
3	digitalcommons.wpi.edu Internet Source	1%
4	trepo.tuni.fi Internet Source	1%
5	Veronica Yenquenida Flamenco. "An Automated Method for Evaluating the Accuracy of ASL Static Gestures", 2014 International Conference on Computational Science and Computational Intelligence, 2014 Publication	<1%
6	Dong Dai, Wei Zhuang, Yixian Shen, Lu Li, Haiyan Wang. "Chapter 10 Design of Intelligent Mobile Robot Control System	<1%

	Based on Gesture Recognition", Springer Science and Business Media LLC, 2020 Publication	
7	www.ritec.com Internet Source	<1 %
8	Vivek Kumar, Vineet Shekhar, Vikrant Verma. "An intelligent wearable to aid speech impaired people by detection of specific hand gestures using flex sensors", AIP Publishing, 2019 Publication	<1 %
9	docplayer.net Internet Source	<1 %
10	Mohamed Aktham Ahmed, Bilal Bahaa Zaidan, Aws Alaa Zaidan, Mahmood Maher Salih, Muhammad Modi bin Lakulu. "A Review on Systems-Based Sensory Gloves for Sign Language Recognition State of the Art between 2007 and 2017", Sensors, 2018 Publication	<1 %
11	S Yarisha Heera, Madhuri K Murthy, V S Sravanti, Sanket Salvi. "Talking hands — An Indian sign language to speech translating gloves", 2017 International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), 2017 Publication	<1 %