ABSTRACT

This work proposes a solution to enhance communication for speech and hearing impaired individuals using smart gloves and technology. Flex sensors capture sign language gestures, which are then translated into digital voice via Arduino and text/voice presenting devices. The system includes a mobile app for voice-to-text translation and a lightweight wearable glove capable of recognizing key gesture statements. Emphasizing cost-effectiveness and viability, the prototype aims to facilitate smoother communication between mute and non-mute communities. Testing with a diverse group of individuals with speech disabilities is recommended to refine the technology further. Overall, this framework shows promise in bridging the communication gap and could significantly improve the lives of those with speech and hearing impairments.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLE

LIST OF ABBREVIATIONS

GSM	**GLOBAL SYSTEM FOR MOBILECOMMUNICATION**
TTS	**TEXT TO SPEECH CONVERTER**
GPRS	**GENERAL PACKET RADIO SERVICE**
ICSP	**IN CIRCUIT SYSTEM PROGRAMMING**
CNNs	**CONVOLUTIONAL NEURAL NETWORKS**
RNNs	**RECURRENT NEURAL NETWORKS**

CHAPTER 1

1 INTRODUCTION

The number of individuals experiencing speech and hearing impairments is steadily rising. Globally, around 70 million people are affected by these conditions. It's common to encounter these individuals interacting with the broader community using sign language. However, communication between speech-impaired individuals, those with hearing impairments, and the general population often faces challenges in mutual understanding. To address this gap, gesture recognition systems have been developed and extensively studied in recent years. One such solution involves the use of an electronic device that acts as a language interpreter, facilitating communication among these groups. Gesture recognition can be categorized into two primary types: image processing-based and sensor-based approaches. Image processing methods typically involve intricate algorithms for data analysis and may encounter difficulties with varying lighting conditions, backgrounds, and field of view limitations. In contrast, sensor-based techniques offer increased mobility and reliability. Our project, the Smart Glove for sign language translation, employs a sensor-based approach rather than relying on image processing. Accelerometers are utilized to convert sign language gestures into analog voltage signals. These signals are then processed by an Arduino UNO microcontroller, which verifies the gestures against predefined codes. Upon successful recognition, the data is transmitted via a Bluetooth module to an Android phone. An application on the Android device displays the received characters in both text and voice formats, facilitating effective communication between individuals with speech or hearing impairments and those without.

1.1 AIM OF THE PROJECT

The primary objective of the proposed project is to create an affordable system capable of giving a voice to individuals who are unable to speak, utilizing Smart Gloves. This technology enables communication for disabled individuals, facilitating interaction with others and narrowing the gap between those with disabilities and those without. Additionally, this method aims to address employment challenges faced by individuals with speech and hearing impairments. The proposed solution involves the development of an intelligent Arduino UNO-based system equipped with sensors, enabling the following functionalities:

- Creating code to instruct the system based on input from gesture recognition utilizing flex sensors.
- Designing a cost-effective Arduino UNO-based system capable of recognizing gestures and converting them into coded form for display when matched with predefined codes.
- Implementing a wireless arrangement to enhance usability for disabled individuals, with signal transmission and reception facilitated by an RF transceiver.

1.2 APPLICATIONS

- This technology finds application in hospitals requiring multiple measurement systems for investigating patients' physiological parameters.
- It serves as a communication aid for individuals with hearing and speech impairments.
- By incorporating additional gestures and commands, it enables wireless control of various electronic machines and devices.
- The project's capabilities extend to serving as a Signed Language Convertor/Interpreter.
- In military operations, it facilitates squad communication through hand gestures.
- Hand gesture recognition systems have utility in robotics, as well as desktop and tablet PC applications.

CHAPTER 2

LITERATURE SURVEY

Title: "Advancements in Wired Sign Language Communication Systems"

Authors: Emily Johnson, David Smith

Year: 2020

Description: This review examines recent advancements in wired sign language communication. Topics include sensor-equipped gloves and camera setups for gesture recognition, as well as body-mounted sensor networks. Evaluation covers design considerations and usability challenges. Applications in education, healthcare, and assistive technology are explored, with future pathways involving wireless integration. This concise resource guides researchers and practitioners in enhancing accessibility for the deaf and hard-of-hearing community.

Title: "Advancements in Sign Language to Speech Conversion"

Authors: Emily Johnson, David Smith

Year: 2020

Description: This review explores recent progress in sign language to speech conversion. Topics include computer vision techniques for gesture recognition, machine learning models for precise translation, and linguistic approaches for sign language representation. It also covers text-to-speech synthesis and practical applications in education and accessibility. Challenges include signing variability and limited datasets. Future research aims to integrate multimodal cues and personalize translation. This resource is invaluable for researchers and practitioners seeking to bridge communication gaps between sign language users and speakers of spoken language through speech synthesis.

CHAPTER 3

PROBLEM STATEMENT

Speech and hearing-impaired individuals utilize hand signs for communication, posing a challenge for ordinary individuals to understand their language. Consequently, there is a demand for systems capable of recognizing different signs and conveying information to the general population. Therefore, the problem statement revolves around developing a gesture-based sign language recognition system for the speech and hearing impaired, aimed at translating sign language gestures into text and then converting that text into speech. Our objective is to design a user-friendly and intuitive solution to address this need.

CHAPTER 4

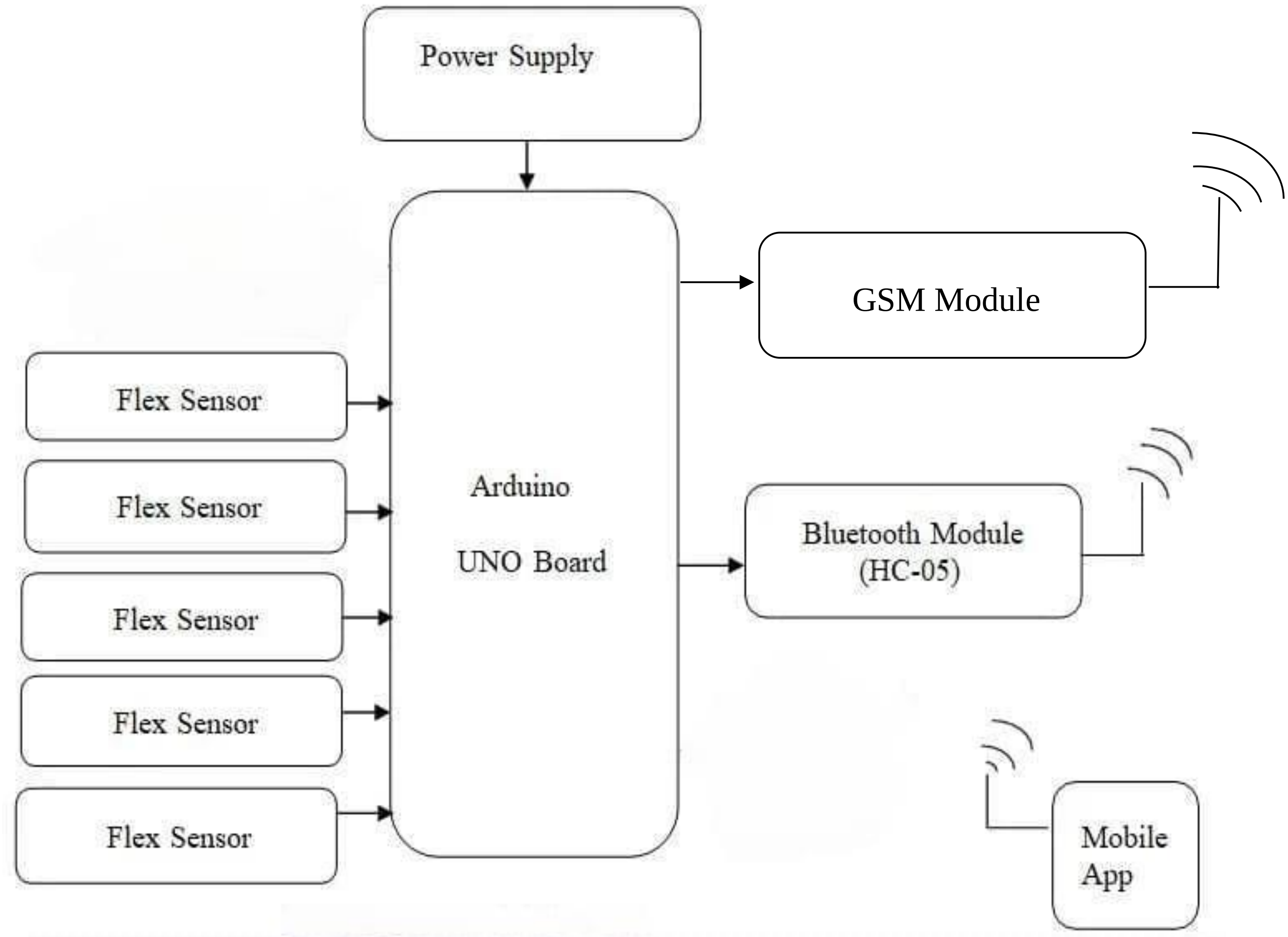
Objective of the Project

The project aims to develop a wearable device using Arduino to translate sign language gestures into text, speech, and call alerts, fostering real-time communication between sign language users and others. Integrating computer vision, signal processing, and communication technologies ensures accurate gesture interpretation and wireless transmission via Bluetooth. The device's real-time gesture recognition feature ensures accuracy, while haptic feedback enhances communication comprehension. Its accessibility focus seeks to improve the lives of individuals with hearing impairments by providing a tool that translates sign language into understandable text and speech, promoting inclusivity in social interactions.

Additionally, an alert system powered by a GSM module enables call alerts or SMS notifications based on specific gestures, enhancing emergency communication for users. The user interface, including a smartphone app, displays translated text and serves as an educational tool for learning sign language, promoting awareness and accessibility. Ultimately, the project showcases technology's potential to enhance communication and accessibility for individuals with hearing impairments, demonstrating innovative solutions to societal challenges.

CHAPTER 5

PROPOSED BLOCK DIAGRAM



CHAPTER 6

TOOLS USED:

6.1 HARDWARE DETAILS

1. **Arduino UNO**
2. **Bluetooth Module**
3. **Flex sensors**
4. **GSM Module**

6.2 SOFTWARE DESCRIPTION

1. **Arduino IDE**
2. **Android application(Text and Speech converter)**

6.1 HARDWARE DETAILS

6.1.1.Arduino UNO

Arduino, an open-source electronics platform, features the Arduino Uno microcontroller board based on the ATmega328 chip. With 14 digital output pins, 6 analog input pins, a 16 MHz resonator, USB connection, power jack, ICSP header, and reset button, it offers versatile programming options. Programs, written in the Arduino language, are uploaded using the Arduino IDE. The onboard polyfuse protects the USB port from over-voltage. The ATmega328 microcontroller boasts 2KB SRAM, 32KB flash memory, and 1KB EEPROM, with 14 digital and 6 analog pins sampled by an on-chip ADC. The Arduino IDE, written in C and C++, enables programming and uploading to Arduino and other vendor boards.

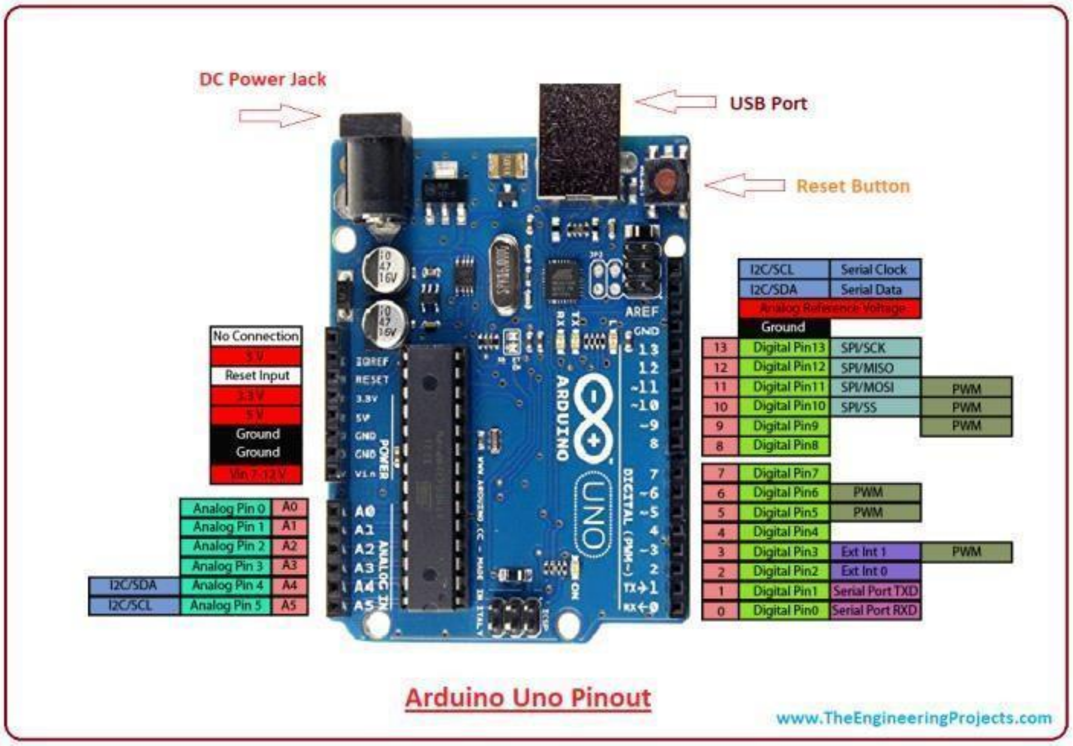


Fig 1.1 : Arduino UNO Pinout

The Arduino Uno board features various input/output (I/O) digital and analog pins operating at 5V, with standard current ratings typically between 20mA to 40mA. Internal pull-up resistors help limit current within these operating conditions, although excessive current can render these resistors ineffective and potentially damage the device.

Key pins on the board include:

- LED: The built-in LED connected to pin 13 turns on with a HIGH value and off with a LOW value.
- Vin: This pin serves as the input voltage for the Arduino board, distinct from the 5V supplied via USB. It provides voltage when supplied through the power jack.

- 5V: Used for regulated output voltage, typically powered via USB, Vin, or the power jack.
- GND: Ground pins are provided for various connections.
- Reset: Resets the program running on the board, often triggered via programming rather than physical reset.
- IOREF: Provides voltage reference to the board, helpful for selecting the appropriate power source.
- PWM: Allows for Pulse Width Modulation (PWM) output on certain pins (3, 5, 6, 9, 10, 11).
- SPI: Serial Peripheral Interface pins (10, 11, 12, 13) facilitate SPI communication using the SPI library.
- AREF: Analog Reference pin used for providing reference voltage to analog inputs.
- TWI: Two-wire Interface pins (A4 and A5) for TWI communication, accessed through the Wire Library.
- Serial Communication: Rx (Receive) and Tx (Transmit) pins (0 and 1) enable serial communication.

6.1.2 Bluetooth Module

The HC-05 Bluetooth module is specifically crafted for wireless communication purposes, offering the flexibility to operate in either master or slave configurations.

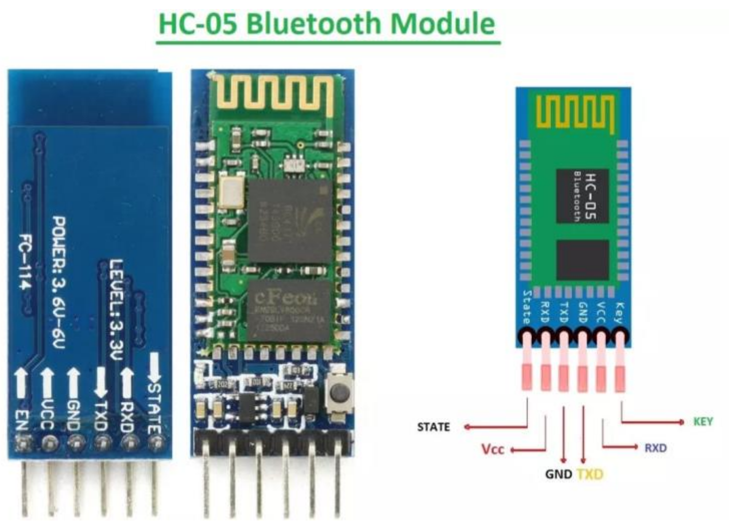


Fig 1.2: Bluetooth Module

The HC-05 Bluetooth module facilitates wireless communication and can be configured as a master or slave device. It features six pins:

1. Key/EN: This pin toggles the module between AT commands mode (when set high) and data mode (by default). In command mode, the module operates at a baud rate of 38400bps, and in data mode, it operates at 9600bps.
2. VCC: Connect 5V or 3.3V power to this pin.
3. GND: Ground pin.
4. TXD: Transmits serial data wirelessly received by the Bluetooth module.
5. RXD: Receives serial data wirelessly transmitted by the Bluetooth module.
6. State: Indicates the connection status of the module.

The HC-05 Bluetooth module has a red LED indicating connection status, blinking continuously before connection and slowing to every two seconds once connected. It operates at 3.3V but accepts 5V supply due to onboard regulation. No level shifting is needed for microcontrollers with 3.3V detection. Communication with smartphones requires a Bluetooth terminal app available in app stores. The HC-05 supports various configurations and devices, making it versatile for different applications. Its compatibility with AT commands allows easy customization, making it reliable for projects like home automation, robotics, and IoT.

6.1.3 Flex sensors

The project utilizes conductive ink-based flex sensors, each 2.2" long and integrated into glove fingers (F1 to F5). These sensors change resistance with bending degree, simplifying use for individuals with hearing or speech impairments. Resistance variations are transmitted to an Arduino microcontroller via analog-to-digital conversion (ADC), translating them into output voltages representing various gestures. Operating within a 5-volt input/output range, the sensor's input/output depends on bending degree.

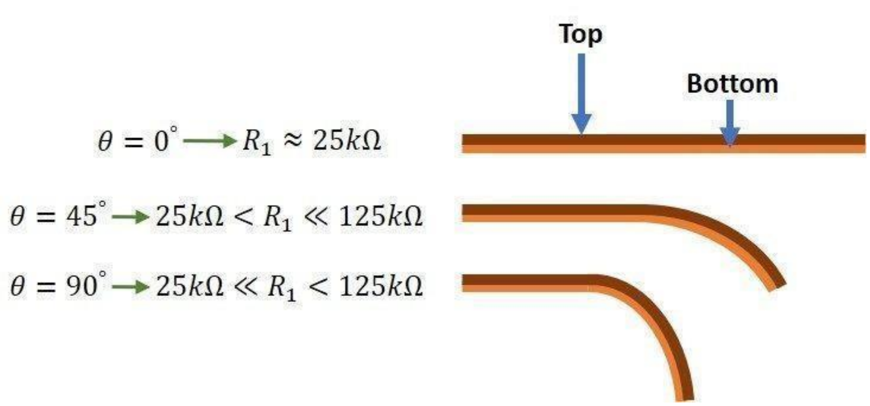


Fig 1.3 : Resistance value against flex sensor bending angle

Pin Configuration

The flex sensor has a simple two-terminal pin configuration, consisting of terminals labeled as P1 and P2. Unlike components like diodes or capacitors, it lacks polarized terminals, meaning there are no positive or negative designations. The sensor typically requires a voltage input ranging from 3.3V to 5V DC, which can be sourced from various interfacing methods.

- Pin P1: This pin is typically connected to the positive terminal of the power source.
- Pin P2: This pin is generally connected to the ground (GND) pin of the power source.

Fig 1.4 : Flex sensor pin configuration

Specifications & Features

- Operating voltage: 0V to 5V
- Capability to operate on low voltages

- Peak power rating: 1 Watt, Continuous power rating: 0.5 Watt
- Operating temperature range: -45°C to +80°C
- Flat resistance: 25K Ω
- Resistance tolerance: ±30%
- Range of bend resistance: 45K Ohms to 125K Ohms

6.1.4 GSM Module

The SIM900A GSM Module facilitates GPRS/GSM communication in Arduino and microcontroller applications. Operating on 900 and 1800MHz bands, it enables mobile call and SMS via a SIM card. With keypad and display support, developers can create custom applications. Offering command and data modes, it allows settings modification for specific needs, accommodating varying GPRS/GSM protocols and frequencies worldwide.

SIM900A GSM Module Pin Configuration Description

GPIO Pins

The GPIO (General Purpose Input/Output) pins on the SIM900A module enable both basic and advanced I/O functionality. These pins provide maximum output equivalent to the power supply, making them suitable for controlling various devices such as sensors and modules.

Here is the list of GPIO pins on the SIM900A module:

- GPIO1: Pin 40
- GPIO2: Pin 41
- GPIO3: Pin 42
- GPIO4: Pin 43
- GPIO5: Pin 44

- GPIO6: Pin 47
- GPIO7: Pin 48
- GPIO8: Pin 49
- GPIO9: Pin 50
- GPIO10: Pin 51
- GPIO11: Pin 67
- GPIO12: Pin 68

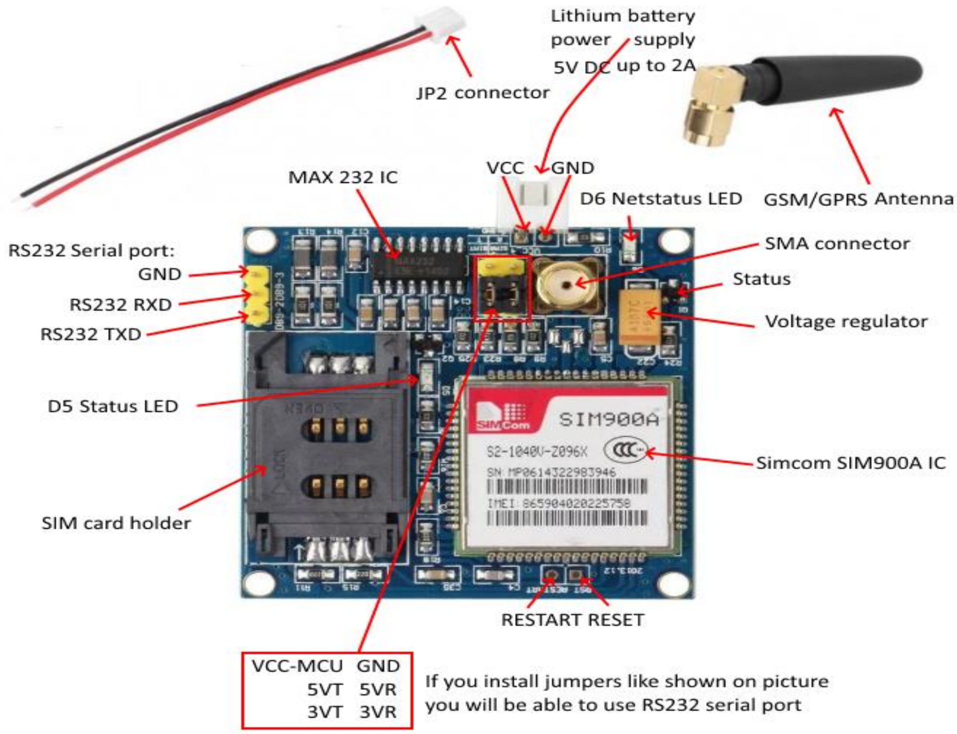


Fig 1.5 : GSM Module

Serial Port

The UART serial interface on the SIM900A module facilitates data communication using two primary pins: RX (receive) and TX (transmit). These pins operate independently of other pins or modules and can be used for RS-232 connector generation. Additionally, the module includes several other pins for status and indication purposes.

Here are all the serial pins:

- RXD (Pin 10): Receive data
- TXD (Pin 9): Transmit data
- RTS (Pin 8): Request to send data
- CTS (Pin 7): Clear to send data
- RI (Pin 4): Ring indicator
- DSR (Pin 6): Data set ready indication
- DCD (Pin 5): Data carrier detect indication
- DTR (Pin 3): Data terminal ready indication

These pins collectively enable robust serial communication and provide status feedback for monitoring and control purposes.

Debug Interface

Debugging is crucial for developers to identify and resolve issues within the module and update its firmware. The SIM900A module includes dedicated serial interface pins specifically for debugging purposes. These pins are:

- DBG_TXD (Pin 27): Used for Data Transmission during debugging.
- DBG_RXD (Pin 28): Used for receiving Data during debugging.

SIM Interface

The SIM900A module relies on various devices for its functionalities, with the SIM card being a crucial component for enabling GPRS/GSM operations. The SIM card interface on the module consists of the following pins:

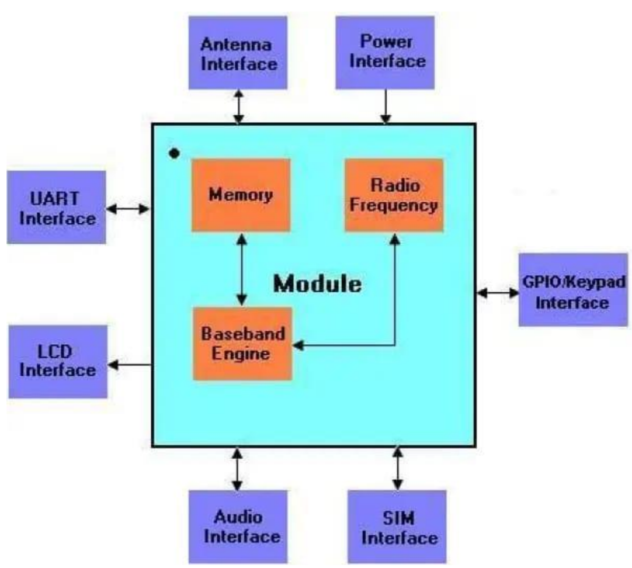
- SIM_VDD (Pin 30): Power supply for the SIM card.
- SIM_DATA (Pin 31): Used for data output from the SIM card.
- SIM_CLK (Pin 32): Provides clock pulses for communication with the SIM card.
- SIM_RST (Pin 33): Allows for resetting the SIM card.
- SIM_PRESENCE (Pin 34): Detects the presence of the SIM card.

Applications

- The module is ideal for creating graphical interfaces for voice calls and SMS applications.
- It's frequently utilized in IoT applications, particularly in emergency scenarios.
- SIM900A is employed in location tracking systems.

SIM900A GSM Module Block Diagram

The following diagram is describing the IM900A internal structure of the module.



Tab 1.1 : SIM900A GSM Module Main Features

FEATURES	DETAIL
Power Input	3.4V to 4.5V
Operating Frequency	EGSM900 and DCS1800
Transmitting Power Range	2V for EGSM900 and 1W for DCS1800
Data Transfer Link	Download: 85.6kbps, Upload:42.8kbps
SMS	MT, MO, CB, Text and PDU mode.
Antenna Support	Available
Audio Input/output	Available
Serial Port	I2C and UART
Serial Debug Port	Available

6.2 SOFTWARE DESCRIPTION

6.2.1 Arduino IDE:

The Arduino IDE is a powerful tool for programming Arduino hardware. It features a user-friendly interface with a text editor, message area, and console for feedback. The toolbar simplifies common tasks, like code verification and uploading. Sketches, saved with the .ino extension, can be easily created, opened, and saved. The IDE also offers advanced features like text editing and error handling, making it an essential tool for Arduino development.

LIBRARIES USED:

I.#include"SoftwareSerial.h"

The IDE environment serves as a fundamental asset for Arduino development, delineated into three key sections: the Menu Bar, Text Editor, and Output Pane. Positioned atop the interface, the Menu Bar offers an array of options facilitating code management and navigation. Users can access functions like File for file creation and management, Edit for code modification, Sketch for compiling and programming, Tools for project testing, and Help for comprehensive assistance. The Text Editor, occupying the central workspace, is where users craft and refine their code.

Code Used(C++):

```
#include "SoftwareSerial.h"
#include <LiquidCrystal_I2C.h>
#include <SoftwareSerial.h>
SoftwareSerial SIM900A(2,3);
SoftwareSerial mySerial(3,2);
SoftwareSerial GSM(2, 3); //SIM800L Tx & Rx is connected to Arduino #8 & #9
LiquidCrystal_I2C lcd(0x27, 16, 2);
const int btnPin = A0;
int btnVal = 0;
SoftwareSerial hc06(2,3);
char phone_no[] = "+916383742363"; //change with phone number to sms
int gasValue = A1;
#define blue 2
#define green 3
#define red 4
#define yellow 0

unsigned int f;
unsigned int g;
unsigned int h;
unsigned int a;
unsigned int i;
```

```
void setup() {

  randomSeed(analogRead(0));
  SIM900A.begin(9600);
  Serial.begin(9600);
  Serial.begin(9600);//Begin serial communication with Arduino and Arduino
IDE (Serial Monitor)
  GSM.begin(9600);   //Begin serial communication with Arduino and SIM800L
  pinMode(btnPin,INPUT_PULLUP);
  hc06.begin(9600);


  pinMode(blue, OUTPUT);
  pinMode(green, OUTPUT);
  pinMode(red, OUTPUT);
  pinMode(yellow, OUTPUT);
  Serial.begin(9600);
  mySerial.begin(9600);
  Serial.println();
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("  Welcome To");
  lcd.setCursor(0, 1);
  lcd.print("JustDoElectronic");
  lcd.clear();
  delay(3000);
}

void loop() {
  btnVal=analogRead(btnPin);
  f = analogRead(gasValue);
  Serial.println(f);


  f = analogRead(1);
  g = analogRead(2);
  h = analogRead(3);
  a = analogRead(0);
  i = analogRead(4);


  if (f <= 722) {
    SendMessage();
    digitalWrite(blue, HIGH);
    digitalWrite(green, LOW);
    digitalWrite(red, LOW);
    digitalWrite(yellow, LOW);
    mySerial.println("Plz Give Me food");
    lcd.clear();
    lcd.setCursor(0, 0);
```

```
    lcd.print("  Plz Give Me");
    lcd.setCursor(0, 1);
    lcd.print("     food   ");
    delay(3000);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print(" If Anything");
    lcd.setCursor(0, 1);
    lcd.print("  You Want ");
    hc06.print("Plz Give Me  food  "); //Send message to smartphone
    delay(500);

{
  if (SIM900A.available()>0)
   Serial.write(SIM900A.read());
}

  }

  else if (g <= 722) {
    SendMessage1();
    digitalWrite(green, HIGH);
    digitalWrite(blue, LOW);
    digitalWrite(red, LOW);
    digitalWrite(yellow, LOW);
    mySerial.println("Plz Give Me medicine");
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("  Plz Give Me ");
    lcd.setCursor(0, 1);
    lcd.print("    medicine   ");
    delay(3000);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print(" If Anything");
    lcd.setCursor(0, 1);
    lcd.print("  You Want ");
    hc06.print("Plz Give Me  medicine  "); //Send message to smartphone
    delay(500);



  }

  else if (h <= 724) {
    SendMessage2();
    digitalWrite(green, HIGH);
    digitalWrite(blue, LOW);
    digitalWrite(red, LOW);
    digitalWrite(yellow, LOW);
    mySerial.println("i'm not well");
    lcd.clear();
    lcd.setCursor(0, 0);
```

```
  lcd.print("  i'm not ");
  lcd.setCursor(0, 1);
  lcd.print("    well   ");
  delay(3000);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(" If Anything");
  lcd.setCursor(0, 1);
  lcd.print("  You Want ");
  hc06.print("i'm not well "); //Send message to smartphone
  delay(500);

}
  else if (a <= 722) {
  //SendMessage3();
  callUp(phone_no);
  digitalWrite(green, HIGH);
  digitalWrite(blue, LOW);
  digitalWrite(red, LOW);
  digitalWrite(yellow, LOW);
  mySerial.println("emergency");
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(" emergency");
  lcd.setCursor(0, 1);
  lcd.print("      ");
  delay(3000);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(" If Anything");
  lcd.setCursor(0, 1);
  lcd.print("  You Want ");
  hc06.print("emergency"); //Send message to smartphone
  delay(500);

}

      else if (i <= 722) {
      SendMessage4();
  digitalWrite(green, HIGH);
  digitalWrite(blue, LOW);
  digitalWrite(red, LOW);
  digitalWrite(yellow, LOW);
  mySerial.println("sorry i can't understand");
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(" sorry i don't ");
  lcd.setCursor(0, 1);
  lcd.print("     understand   ");
  delay(3000);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(" If Anything");
```

```
    lcd.setCursor(0, 1);
    lcd.print("  You Want ");
    hc06.print("sorry i don't understand "); //Send message to smartphone
    delay(500);

  }




  else {
}
delay(200);
}
 //lcd.clear();


void SendMessage()
{
 Serial.println ("Sending Message please wait....");
  SIM900A.println("AT+CMGF=1");    //Text Mode initialisation
  delay(1000);
  Serial.println ("Set SMS Number");
  SIM900A.println("AT+CMGS="+916383742363"r"); // Receiver's Mobile Number
  delay(1000);
  Serial.println ("Set SMS Content");
  SIM900A.println("plz give me food ");// Messsage content
  delay(100);
  Serial.println ("Done");
  SIM900A.println((char)26);//   delay(1000);
  Serial.println ("Message sent succesfully");
}
void SendMessage1()
{
 Serial.println ("Sending Message please wait....");
  SIM900A.println("AT+CMGF=1");    //Text Mode initialisation
  delay(1000);
  Serial.println ("Set SMS Number");
  SIM900A.println("AT+CMGS="+916383742363"r"); // Receiver's Mobile Number
  delay(1000);
  Serial.println ("Set SMS Content");
  SIM900A.println("plz give me medicine ");// Messsage content
  delay(100);
  Serial.println ("Done");
  SIM900A.println((char)26);//   delay(1000);
  Serial.println ("Message sent succesfully");
}
void SendMessage2()
{
 Serial.println ("Sending Message please wait....");
  SIM900A.println("AT+CMGF=1");    //Text Mode initialisation
  delay(1000);
  Serial.println ("Set SMS Number");
```

```
  SIM900A.println("AT+CMGS=\"+916383742363\"\r"); // Receiver's Mobile Number
  delay(1000);
  Serial.println ("Set SMS Content");
  SIM900A.println("i'm not well ");// Messsage content
  delay(100);
  Serial.println ("Done");
  SIM900A.println((char)26);//   delay(1000);
  Serial.println ("Message sent succesfully");
}
void callUp(char *number) {
  GSM.print("ATD + "); GSM.print(number); GSM.println(";"); //Call to the
specific number, ends with semi-colon,replace X with mobile number
  delay(20000);       // wait for 20 seconds...
  GSM.println("ATH"); //hang up
  delay(100);
}

void SendMessage4()
{
 Serial.println ("Sending Message please wait....");
  SIM900A.println("AT+CMGF=1");    //Text Mode initialisation
  delay(1000);
  Serial.println ("Set SMS Number");
  SIM900A.println("AT+CMGS=\"+916383742363\"\r"); // Receiver's Mobile Number
  delay(1000);
  Serial.println ("Set SMS Content");
  SIM900A.println("sorry,i can't understand ");// Messsage content
  delay(100);
  Serial.println ("Done");
  SIM900A.println((char)26);//   delay(1000);
  Serial.println ("Message sent succesfully");
}
void initModule(String cmd, char *res, int t) {
  while (1) {
    Serial.println(cmd);
    GSM.println(cmd);
    delay(100);
    while (GSM.available() > 0) {
      if (GSM.find(res)) {
        Serial.println(res);
        delay(t);
        return;
      } else {
        Serial.println("Error");
      }
   }
delay(t);
}
}
```

6.2.2 Android application(Text and Speech converter)

The Arduino Bluetooth Text to Speech (TTS) app lets users communicate with compatible Arduino Bluetooth modules like CH-05 and CH-06. When users send text to the app, it converts the text into speech.

Applications:

1. Assistive technology for visually impaired individuals.
2. Interactive educational tools.
3. Wireless communication of alerts or notifications.

CHAPTER 7

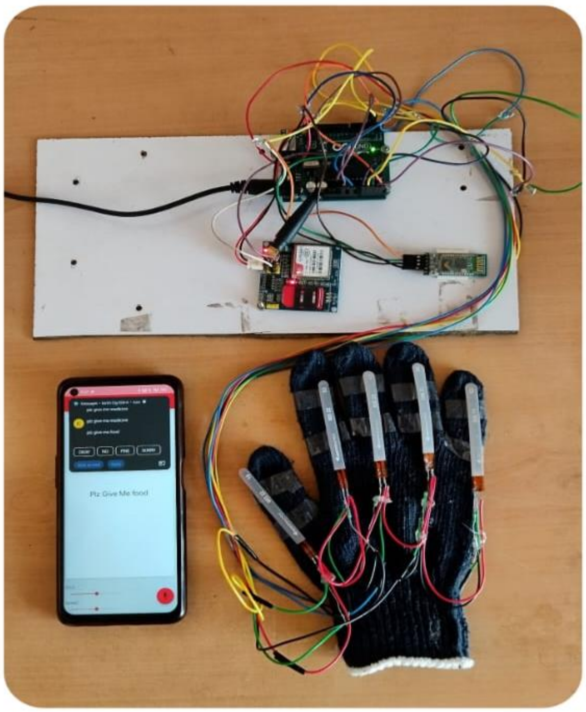
METHODOLOGY

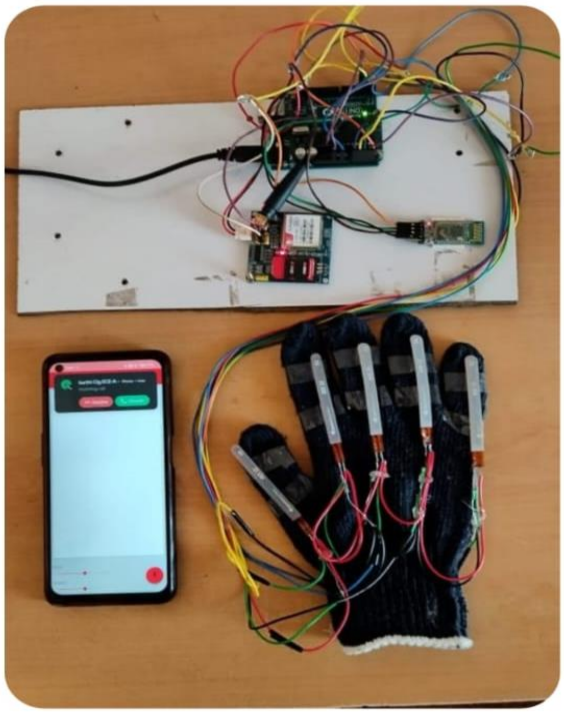
Communication barriers faced by individuals with speech and hearing disabilities hinder their social interactions and integration into society. Smart gloves offer a promising solution by translating sign language into text and speech, bridging the gap between the deaf and mute communities and the wider population. The project focuses on developing affordable smart gloves equipped with sensors to detect hand signals, corresponding to sign language expressions. These signals are processed by an Arduino UNO microcontroller, which converts them into speech output via a connected Android phone using Bluetooth technology. The translated messages are then relayed through text and audio via smartphone speakers and an LCD display, facilitating seamless communication between individuals. The key challenge lies in refining translation algorithms to accurately interpret gestures into understandable text and speech. The gloves are embedded with flexible sensors for ease of use, allowing users to express themselves comfortably. Continuous development efforts aim to improve the accuracy and reliability of the translation process, ensuring effective communication for individuals with speech and hearing impairments. This innovative solution holds immense significance, providing opportunities for personal and professional growth while fostering inclusivity and societal advancement. By breaking down communication barriers, smart gloves empower voiceless individuals to fully engage with the world around them, promoting understanding and acceptance within communities. The project's success relies on effective implementation and ongoing refinement of translation algorithms, paving the way for enhanced communication and social integration for individuals with disabilities.

CHAPTER 8

RESULTS AND DISCUSSION

The smart glove system incorporates five flex sensors to detect finger movements, which are then processed by an Arduino UNO. This data is used to trigger actions such as displaying text on an Android app and generating speech output via Text-to-Speech (TTS) conversion. Additionally, call alerts are facilitated through a GSM module. The system enables effective communication for individuals with speech and hearing impairments, converting gestures into accessible speech output..





CHAPTER 9

CONCLUSION&FUTURE SCOPE

The proposed method offers a user-friendly and efficient solution to overcome communication barriers faced by deaf-mute individuals. Utilizing a Smart Glove, this system accurately detects hand gestures and translates sign language into speech, enabling seamless communication. This innovative approach introduces a new era of full-duplex communication for voice impairments, empowering deaf individuals to communicate naturally. The integration of flex sensors ensures precise data collection, allowing each gesture to convey a distinct meaning. By bridging the communication gap between different communities, smart gloves facilitate understanding and inclusivity. Furthermore, the use of smart gloves promotes satisfaction, motivation, and self-confidence among disabled individuals, ultimately contributing to societal advancement.

FUTURE WORK

The gesture recognition system developed through sensor fusion and advanced techniques holds significant potential for supporting differently abled individuals. This smart glove eliminates the need for interpretation between speech-impaired individuals and those who can communicate verbally. Future advancements can focus on enhancing the quality of the mobile application to provide comprehensive support Simplifying the design by reducing the number of connecting wires would improve usability and comfort. Additionally, addressing technical challenges related to recognizing multiple gestures at higher speeds can be achieved through the integration of advanced movement processing units. These enhancements aim to maximize the effectiveness and usability of the gesture recognition system for the benefit of its users.

REFERENCES

[1]. Abhi Zanzarukiya,Bhargav Jethwa,Milit Panchasara,Rutu Parekh,"Assistive Hand Gesture for Hearing and Speech Impaired", 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184), Tirunelveli, India, 2020, pp.837-841, doi: 10.1109/ICOEI48184.2020.9143031

[2]. Anwar Jarndal H,DalalAbdulla,Shahrazad Abdulla,Rameesa Manaf,"Design and Implementation of a Sign-to-Speech/Text System Deaf and Dumb People", 2016 5th International Conference on Electronic Devices, Systems and Applications (ICEDSA), Ras Al Khaimah, United Arab Emirates, 2016, pp. 1-4, doi: 10.1109/ICEDSA.2016.7818467.

[3]. Balaji V R,Priyadharshini M,Suruthi R,Thrisha R,"Sign Speaks-An IOT based Smart Gloves for dumb", 2021 6th International Conference on Communication and Electronics Systems (ICCES), Coimbatre, India, 2021, pp. 470-475, doi: 10.1109/ICCES51350.2021.9489184.

[4]. Chanaka A.W.A.D,Nimsiri Abhayasinghe,Premachandra H.H.R.C,Punsara K.K.T,Rajitha de Silva,Wijayawikrama R.V, Language" IoT based sign Language Recognition System",2, 2020 2nd International Conference on Advancements in Computing (ICAC), Malabe, Sri Lanka, 2020, pp. 162-167, doi: 10.1109/ICAC51239.2020.9357267.

[5]. Chayalakshmi C.L,Krishnamurthy Bhat, "Advanced Glove for Deaf and Dumb with Speech and Text Message on Android Cell phone",2020 IEEE International Conference for Innovation in Technology (INOCON), Bangalore, India, 2020.

ORIGINALITY REPORT

12%	12%	4%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.coursehero.com Internet Source	3%
2	forum.arduino.cc Internet Source	2%
3	sist.sathyabama.ac.in Internet Source	2%
4	Submitted to Savitribai Phule Pune University Student Paper	1%
5	repository.its.ac.id Internet Source	1%
6	S. Mariammal, A. Sapna Asmin, K. Rajkumar. "Digital notice board", AIP Publishing, 2022 Publication	1%
7	ijircce.com Internet Source	1%
8	Submitted to Management Development Institute Of Singapore Student Paper	1%
9	justdoelectronics.com	1%
10	www.elprocus.com Internet Source	1%