Data visualization and pre processing

Kedar Potdar[1], Gauri Nagavkar [2]

U.G. Student, Department of Computer Engineering, Watumull Institute of Electronics Engineering and Computer Technology, Mumbai, Maharashtra, India[1]

U.G. Student, Department of Electronics and Telecommunication Engineering, Watumull Institute of Electronics Engineering and Computer Technology, Mumbai, Maharashtra, India[2]

ABSTRACT: This project aims to aid the deaf-mute by creation of a new system that helps convert sign language to text and speech for easier communication with audience. The system consists of a gesture recognizer hand-glove which converts gestures into electrical signals using flex sensors. These electrical signals are then processed using an Arduino microcontroller and a Python-based backend for text-to-speech conversion. The glove includes two modes of operation – phrase fetch mode and letter fetch mode. The phrase fetch mode speaks out words at once, while the letter fetch mode speaks out individual letters. This project forms a base infrastructure which can later be augmented with addition of different Sign Languages and integrating with other hearing impaired aid systems.

KEYWORDS: Gesture recognition, Flex sensor, Arduino, Test to speech, Sign language.

I. Introduction

Gesture is a non-verbal means of communication. It refers to expressing an idea using position, orientation or movement of a body part.Gesture recognition is the mathematical interpretation of orientation or motion of human body by a computational system. In this project, the words expressed by hand gestures by the speech and hearing impaired are converted into verbal means of communication. The translated output is displayed on a screen and "spoken" on a speaker.

Sign Language is the well-structured code, which uses hand gestures instead of sound to convey meaning, simultaneously combining hand shapes, orientations and movement of the hands. Communicative hand glove is an electronic device that can translate sign language into speech and text in order to make the communication possible between the deaf and/or mute with the general public. This technology has been used in a variety of application areas, which demands accurate interpretation of sign language. In this project, the words/letters conveyed by the disabled person are displayed on a screen and also spoken on a speaker.

The project is divided into two parts: 1) Data acquisition from the flex sensors 2) Processing the acquired data and giving corresponding output on the screen and speaker. Data acquisition is done using Flex sensors mounted on the Hand glove. Next, the analog signals obtained from the flex sensors are converted into digital. The digital signals are processed and compared with the predefined values. If the values match, the corresponding letter is returned.

The paper is organized as follows. The previous projects and papers related to this paper are described in section II. Description of the hardware and software components used in the project is done in section III. Section IV has the system architecture of the project. Section V gives the operation and flowcharts of the main function and the two

II. RELATED WORK

About 70 million people in the world are speech and hearing impaired.[4] They communicate using hand gestures (the sign language). There have been many projects till date which aim at bridging the communication gap between these specially abled people and the ones who cannot understand the sign language.

In [1], S.Upendran and Thamizharasi A. present a new method to recognize ASL alphabets from an image input. The system extracts Principal Component Analysis (PCA) features from the image of the ASL hand pose. The k-NN (k-Nearest Neighbour) classifier is used to distinguish between various gestures and their meanings.

AnbarasiRajamohan, Hemavathy R., Dhanalakshmi M developed Deaf-Mute Communication Interpreterwhich uses flex sensors and accelerometer to convert signs into nomal language. The text to speech conversion was done using TTS256 SpeakJet, which is a TTS module. Successful interpretation of only 10 letters (A, B, C, D, F, I, L, O, M, N, T, S, W) was achieved. [2]

In [3], Dumb Person by PallaviVerma , Shimi S. L., RichaPriyadarshani have used a PIC microcontroller with flex sensors for gesture recognition, to only display the letters on a LCD screen (no audio output).

III. Introduction To Tools

A. HARDWARE COMPONENTS

1. Flex Sensor

Flex Sensor (Fig 1) uses technology which is based on resistive carbon elements. Flex sensor acts as a variable resistor. Flex Sensors achieve a great form-factor on a thin flexible substrate. When a flex sensor is bent, it produces an output resistance corresponding to the bend radius. Smaller the radius, the higher the resistance value as shown in figure 2.

Fig. 1 Flex Sensor [13]

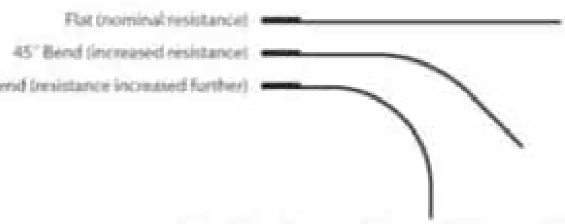

Fig. 2 Resistance Change Diagram [11]

2. Arduino Uno

Arduino Uno is a microcontroller based on the ATmega328. It has 14 digital I/O pins, 6 analog inputs pins, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button. It can be powered via the USB connection or with an external power supply (5 to 20 volts).

3. Hand Glove

A hand glove, mounted with flex sensors for all the five fingers, is worn by the user. The hand glove must be necessarily thin, allowing for ease of movement of the sensors.

B. SOFTWARE SPECIFICATIONS

1. Arduino IDE

The Arduino microcontroller board used for the project needs to be programmed using the Arduino Integrated Development Environment (IDE). It contains contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them. Fig. 3 shows the Arduino IDE User Interface and the components.

Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension .ino. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom right-hand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

Fig. 3 Arduino IDE[12]

2. Python Programming Language

[7] Python is a widely used high-level programming language for general-purpose programming, created by Guido van Rossum and first released in 1991. Being an interpreted language, it focuses on code readability.

Python has a large standard library, which allows developers to install packages from its web server to suite their development needs.

For development of this project, we installed the following Python libraries/dependencies:

1. PySerial – for Arduino Serial Port communication
2. PyWin32 – for interfacing the TTS library with our Windows based system
3. Speech – a TTS library to convert text received from Arduino Uno to speech.

IV. SYSTEM ARCHITECTURE

This section describes the architecture of the system in depth. The system is divided into three modules (Fig. 4) – the flex sensor assembly on glove, the Arduino microcontroller circuit and the Python back end.

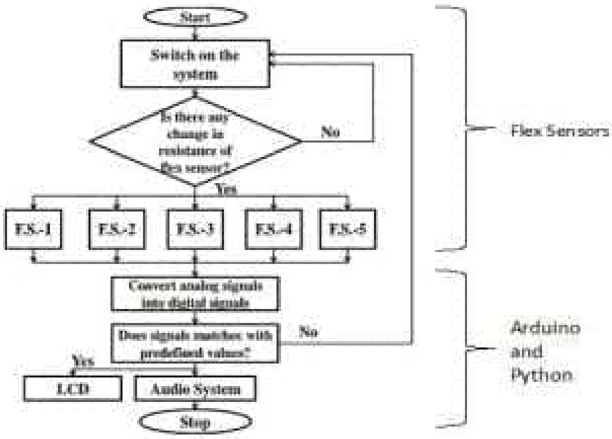

Fig. 4 System Architecture

A. FLEX SENSOR ASSEMBLY

Flex sensors are mounted on top of the hand glove as shown in Fig. 5. Each of the five sensors are then connected to power supply and Analog input pins on the Arduino board.

Fig. 5 Flex Sensors mounted on a Hand glove

B. ARDUINO UNO CIRCUIT

The Arduino Uno Circuit includes additional assembly to help capture gestures for recognition and transmit the corresponding letter to the computer.

Two push buttons are used in the circuit. One for capturing the gesture, i.e. conversion of current gesture to corresponding letter from sign language, and one for printing the End-Of-Line character (EOL) through the Serial Port.

V. WORKING

A. SENSOR DATA RETRIEVAL AND PROCESSING

The function of this unit is to read the gesture and fetch the corresponding letter. A voltage divider circuit is used between the Arduino Uno board and the sensors. It brings down the voltage range to lower values. The stepped down voltage is then supplied to the flex sensors mounted on each finger. The flex sensors act as variable resistors. As the sensor is flexed, the resistance across it increases, giving rise to different values of voltage for different orientations of the fingers. The Arduino Uno acts as ADC and converts these analog signals into corresponding digital signals. (These values are mapped into a higher range for the convenience of classification between the letters and their finger orientations.)

[10] In the Arduino code, we have set predefined threshold values of the digital signals for interpretation of the gestures. For fetching a letter, the user needs to press the fetch button. If the values from the flex sensor orientation match one of the predefined values, the letter will be returned and stored in a buffer. The next letter will be concatenated to the previous letter(s) until the EOL button is pressed.

B. DATA TRANSMISSION AND TEXT TO SPEECH CONVERSION

Text data is sent from Arduino Uno board to the computer using Serial port. This data is parsed by Python's PySerial module.

Fig. 6 shows the flowchart of the entire system. The user can choose between either the phrase mode or letter mode at the startup of the system. Depending on the mode chosen, the letters received at the computer are either spoken out immediately or the computer buffered into a character array and then spoken out using the Python Speech module.

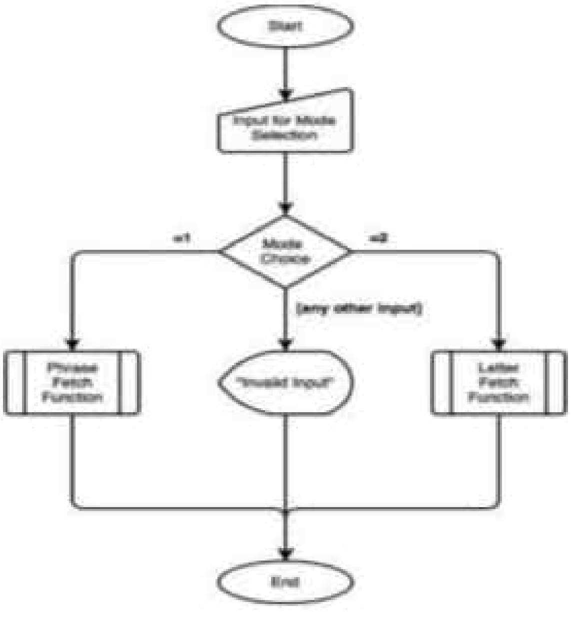

Fig. 6 System Flowchart

V. WORKING

A. SENSOR DATA RETRIEVAL AND PROCESSING

The function of this unit is to read the gesture and fetch the corresponding letter. A voltage divider circuit is used between the Arduino Uno board and the sensors. It brings down the voltage range to lower values. The stepped down voltage is then supplied to the flex sensors mounted on each finger. The flex sensors act as variable resistors. As the sensor is flexed, the resistance across it increases, giving rise to different values of voltage for different orientations of the fingers. The Arduino Uno acts as ADC and converts these analog signals into corresponding digital signals. (These values are mapped into a higher range for the convenience of classification between the letters and their finger orientations.)

[10] In the Arduino code, we have set predefined threshold values of the digital signals for interpretation of the gestures. For fetching a letter, the user needs to press the fetch button. If the values from the flex sensor orientation match one of the predefined values, the letter will be returned and stored in a buffer. The next letter will be concatenated to the previous letter(s) until the EOL button is pressed.

B. DATA TRANSMISSION AND TEXT TO SPEECH CONVERSION

Text data is sent from Arduino Uno board to the computer using Serial port. This data is parsed by Python's PySerial module.

Fig. 6 shows the flowchart of the entire system. The user can choose between either the phrase mode or letter mode at the startup of the system. Depending on the mode chosen, the letters received at the computer are either spoken out immediately or the computer buffered into a character array and then spoken out using the Python Speech module.

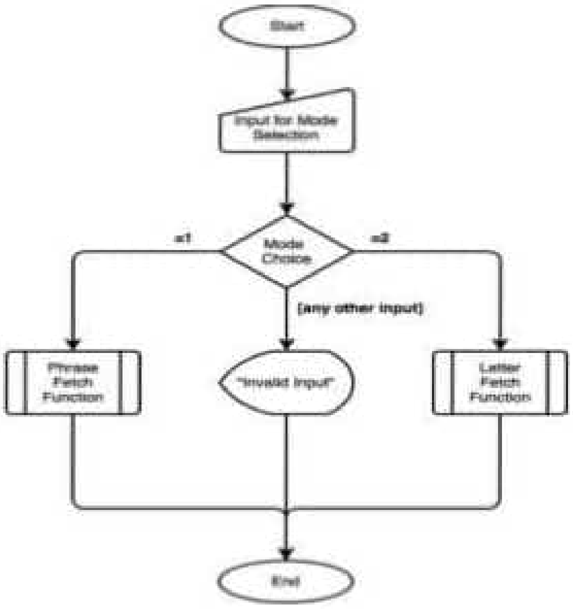

Fig. 6 System Flowchart

1. Phrase Mode

In this mode, as the name suggests, the system prints and speaks out entire words at once. Fig. 7. shows the flowchart for the phrase mode method in the Python program. Each letter transmitted to the computer over serial port is buffered into a character array. Once the system receives EOL character, the buffered string is printed and spoken out using the TTS system.

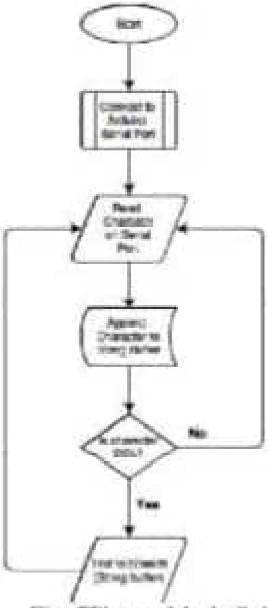

Fig. 7Phrase Mode Subroutine

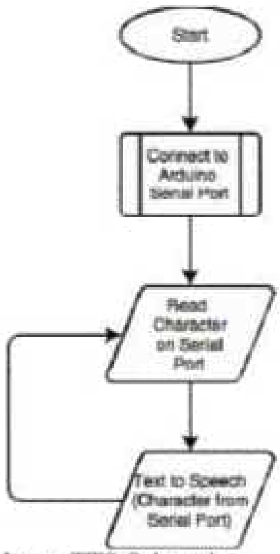

Fig. 8 Letter TTS Subroutine

2. Letter Mode

In Letter Mode, the computer does not buffer characters received at the Serial port, instead prints and speaks out the letters as soon as it receives those as shown in Fig. 8.

VI. LIMITATIONS

The limitation of the system is that employs it a logical mechanism for classification of letters based on sensor values. Implementation of a Machine Learning algorithm like Artificial Neural Networks for classification of letters can help clear the boundaries between similar gestures. [6]

Another limitation is the lack of portability as a Windows-based computer is needed for the backend.

VII. FUTURE SCOPE

The system forms the base infrastructure for a complete communicational aid system for the deaf and mute. To expand its capabilities, more languages can be easily added by adjusting sensor values.

Further, reliance on a dedicated computer system to enable the TTS functionality can be eliminated by adding a portable computer like the Raspberry Pi, which can handle the TTS while retaining portability of such a system. [14]

1. Phrase Mode

In this mode, as the name suggests, the system prints and speaks out entire words at once. Fig. 7. shows the flowchart for the phrase mode method in the Python program. Each letter transmitted to the computer over serial port is buffered into a character array. Once the system receives EOL character, the buffered string is printed and spoken out using the TTS system.

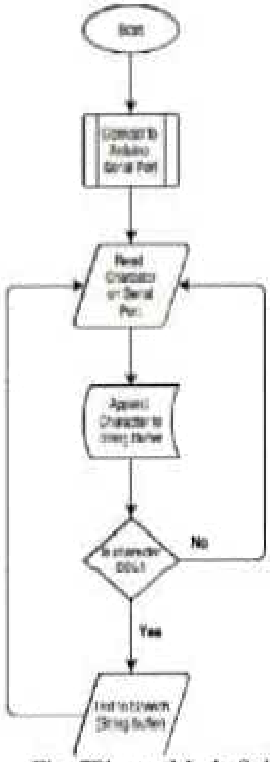

Fig. 7Phrase Mode Subroutine

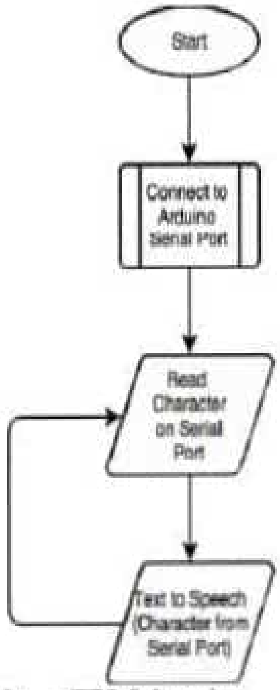

Fig. 8 Letter TTS Subroutine

2. Letter Mode

In Letter Mode, the computer does not buffer characters received at the Serial port, instead prints and speaks out the letters as soon as it receives those as shown in Fig. 8.

VI. Limitations

The limitation of the system is that employs it a logical mechanism for classification of letters based on sensor values. Implementation of a Machine Learning algorithm like Artificial Neural Networks for classification of letters can help clear the boundaries between similar gestures. [6]

Another limitation is the lack of portability as a Windows-based computer is needed for the backend.

VII. Future Scope

The system forms the base infrastructure for a complete communicational aid system for the deaf and mute. To expand its capabilities, more languages can be easily added by adjusting sensor values.

Further, reliance on a dedicated computer system to enable the TTS functionality can be eliminated by adding a portable computer like the Raspberry Pi, which can handle the TTS while retaining portability of such a system. [14]

VIII.EXPERIMENTAL RESULTS

In the Arduino code, R2, R4, R6, R8, R10 are the variables assigned to the thumb, index finger, middle finger, ring finger and little finger respectively.

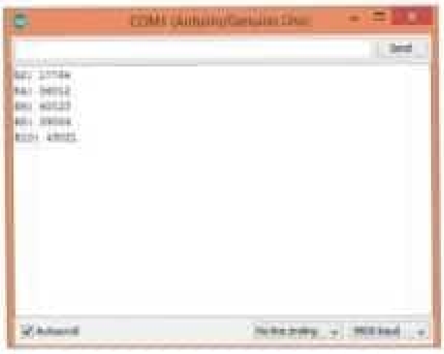

Fig. 8 Serial Output: Variable values

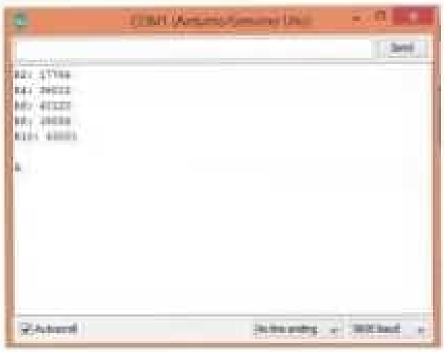

Fig. 9 Serial Output: Letter A

Figure 8 shows the values of R2, R4, R6, R8, R10 obtained at the serial output for a random orientation of the hand. Figure 9 shows the values of R2- R10 for the sign of the letter "A" and the corresponding output, ie "A".

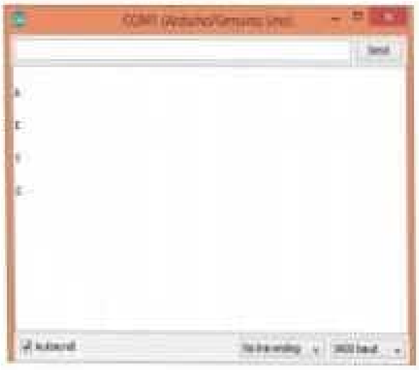

Fig. 10 Serial Output: different letters

Fig. 11 Serial Output: One word

Figure 10 shows letters 'A', 'E', 'Y', 'G' as output on Arduino serial monitor according to the hand gestures made by the user.

Figure 11 shows the word 'GAURI' and the EOL character. This output has been obtained by concatenating the letters 'G', 'A', 'U', 'R', 'I' using fetch button and then pressing the space button for EOL character.