GESTURE VOCALIZER FOR AUDIO AND SPEECH IMPAIRED

Project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

in

Computer Science and Engineering

By

Hardik Kumar (121244)
Lakshay Arora (121260)

Under the supervision of

Mrs. Sanjana Singh

to

Department of Computer Science & Engineering and Information Technology
Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

Candidate’s Declaration

I hereby declare that the work presented in this report entitled “**Gesture Vocalizer for Audio and Speech Impaired**” in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering, submitted in the department of Computer Science & Engineering and Information Technology,
Jaypee University of Information Technology, Waknaghat is an authentic record of my own work carried out over a period from August 2015 to May 2016 under the supervision of Mrs. Sanjana Singh .
The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Hardik Kumar, 121244

Lakshay Arora, 121260

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Mrs. Sanjana Singh
Assistant Professor
Department of Computer Science and Engineering
Dated : 20-12-2015

Acknowledgement

I would like to thank everyone that has contributed to the development of this project, which is the final chapter of our Bachelor education in Information Technology at Jaypee University of Information Technology, Waknaghat, Solan.

Thanks to my supervisor Mrs. Sanjana Singh, for her guidance and valuable advice during this ongoing development of the project. We would also like to thank our parents who provided us with the opportunity to study in this university and enlightened by life and career.

Date : Hardik Kumar(121244)

Lakshay Arora(121260)

Table of Contents

List of Figures

ABSTRACT

Computers and technologies are becoming an integral and important part of lives for most people, including those with disabilities. Improving and enhancing text entry and interaction with computers for disabled users has been investigated for many years, with many systems and interface devices proposed to facilitate and simplify text input process.

This project focuses on the difficulties faced by the audio and speech-impaired while communicating with others.

Hence, the project aims to provide an interface for Speech-to-Gesture conversion via a Speech-to-Text and Gesture-to-Speech module and a Gesture-to-Text interface via a Gesture-to-Text module.

Keywords – Hand Gestures, Text-to-Gesture, Gesture-to-Text, Speech-Recognition

CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION

People who are audio and speech-impaired, or in simple words, deaf and dumb face a lot of difficulty in communicating with others and expressing themselves . Hence, they adopt the method of hand gestures or sign language in order to talk or communicate. But, people who are gifted enough to not suffer from these disabilities, find it difficult to comprehend as they are unfamiliar with these signs and gestures, hence resulting in a communication gap between the impaired and the normal ones.

1.2 PROBLEM STATEMENT

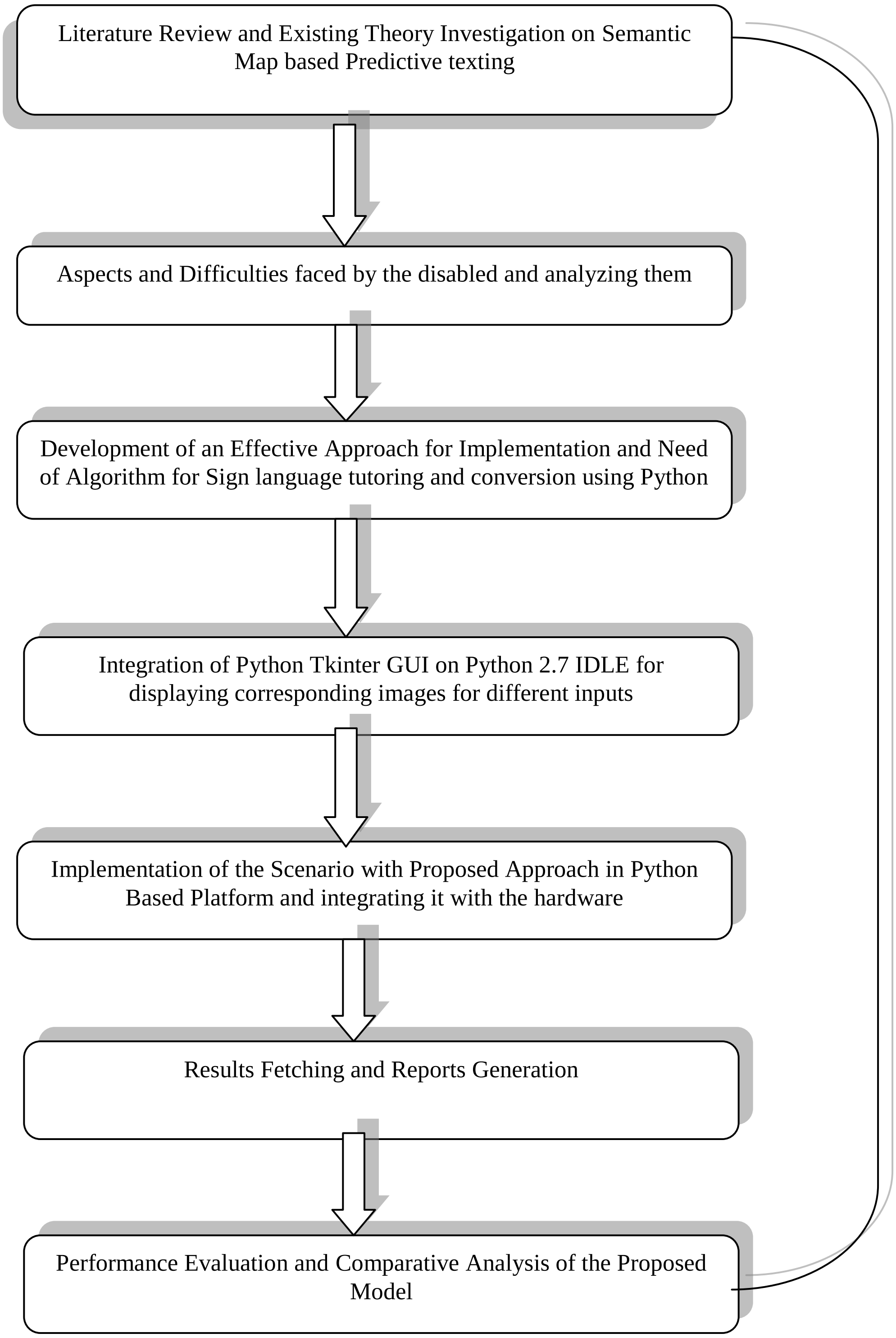
Developing a gesture vocalizer glove based on hardware (Arduino UNO) that bridges the communication gap between the normal people and the impaired through the use of sign language .

1.3 **OBJECTIVES**

1. To perform the deep investigation of the difficulties faced by the disabled i.e. deaf and dumb during initiation of a conversation.
2. To perform a comparative analysis of various designs used by various texting and conversion applications.
3. To propose and implement an effective algorithm and work that is a Python based Application in association with Tkinter GUI module for proper display of sign language and their conversion to text and speech.
4. The keywords frequency and its effect on overall system shall be analyzed. The parameters of analysis shall be
    - Keywords Frequency
    - Performance and Efficiency
    - Cost Factor
5. Creation of Python Based GUI for fetching the inputs
    a. Integration of Tkinter GUI
        i. Keyword
        ii. Images for different sign languages

6. Implementation of a gesture vocalizer
    a. Use of Arduino UNO Development Board.
    b. Integration of hardware with the software implemented

1.4 METHODOLOGY



1.5 **ORGANIZATION**

Chapter 1 highlights and underlines the assorted vulnerabilities and problems to be overcome . In this chapter, the introduction to various concepts and techniques used in the implementation are covered.

Chapter 2 provides the detailed literature review from the research paper, books, journals and conferences are done in. In this chapter, the extracts from assorted research papers on various situations are taken.

Chapter 3 covers the system development which is the key aspect of this work. In this chapter, the proposed model, algorithm and related parameters are emphasized.

Chapter 4 shows the simulation of implementation results with the relative performance analysis .In this chapter, the simulation results and screenshots are revealed to depict and defend the proposed work.

Chapter 5 ends with the detailed conclusion and scope of the future work which guides the upcoming students and research scholars to enhance the current work with higher efficiency and effectiveness.

CHAPTER 2

LITERATURE SURVEY

For completion, justification and solving the problem definition, a number of research papers, magazines, journals and online links are investigated in details.

In this chapter, the details of research papers and journals are specified from where we have analyzed the content and formulated the problem.

A number of research scholars and scientists has written a number of research papers and found excellent results. This section underlines all those research papers and their extracts

Title	A Predictive Text Completion Software in Python
Author	Wong Jiang Fung
Abstract	Predictive text completion is a technology that extends the traditional auto completion and text replacement techniques. It helps to reduce key strokes needed in text input and serves as a more affordable assistive technology for computer users who are dyslexic or has learning/reading difficulty/disability, as compared to more expensive speech-to-text technology or special input devices. Python is used for prototyping, rapid R&D and testing of advanced features, while AutoHotKey scripting language can be used to code the regular stable release. The problems are around reading written text and writing ideas down. Even if the words can be read correctly, it is often so slow that comprehension is lost. In writing the difficulties are compounded since one has to consider mapping out the ideas, arranging them into a formal form, writing, spelling and grammar. All of these are difficult to the dyslexic individual. However there is no problem in speaking their thoughts. In the digital age, computers are often essential for work and personal life. That means a dyslexic individual faces more challenges than ever. The problem of writing ideas down on a paper is now transformed to typing and writing them down on computers. There are assistive technology developed to aid dyslexic computer users[3] such as : 1. <u>Text-to-speech software</u> – so that they can hear the text instead of reading it. 2<u>. Concept mapping</u> – so that they can jot down thoughts on a map which is a representation of relationships between objects and ideas. 3. <u>Speech-to-text software</u> – Alternative input method, speech recognition is expensive and difficult to develop. Current offerings are available in only around six languages, for commercial reasons. 4. <u>Special input devices</u> – created to suit the needs of dyslexic users, are also expensive and not commonly found. A low-cost alternative is a software solution that predicts what the user is typing,presents suggestions and completes the word or phrase for the user. Usually it also comes with writing support features such as spell-checking and grammar checking.

	Text completion software Text completion software is designed for reducing key strokes needed in text input and therefore increasing the typing speed of dyslexic users and making them productive. Different types of text completion software are available in the market, but they are not suitable for dyslexic users because they are not created specifically to address the needs of dyslexic users.
Conclusion	The productivity of Python and the convenience of AutoHotKey help make an excellent software project that benefits many dyslexic computer users and helps them to become productive. Further on-going development is needed to improve the software and attract more developers to work on the project.

Title	A Novel Model for Speech to Text Conversion
Author	Deepa V.Jose, Alfateh Mustafa, Sharan R
Abstract	Gaining fluency in any language is a cumbersome task. To make it easy so many software's are existing now a days. Our aim is to develop a software that enhances the user's way of speech through correctness of pronunciation following the English phonetics. This software allows one to learn, judge and recognize their potential in English language. It also facilitates an extra add-on feature which nourishes the user's communication skills by an option of text to speech conversion conversion also. Enhancing the existing algorithms of speech to text to improve the quality of the output is under consideration. The main constituents of the research : --> TTS(Text To Speech) is the process wherein the computer is made to speak . Speech synthesiser converts audio input into text form and processes the text for further learning modules . -->System based on smart intelligence, voice processing and speech recognition . Software would increase one's communication skill to a higher level . -->Smart Recognition : The voice recording process is achieved through different recognition modus operandi like smart recognition or minimal recognition. Through smart recognition, the background sound can be filtered and only the efficient matching of words is processed. -->Cryptography : The user account consists of various field details which includes the result of progression in each dictionary. Cryptography is imbibed to make sure that the changes in the progress are acquired through genuine learning and not through fake updates. --> Confluence of separate modules : Merging the two different modules i.e. speech recognition module and text to speech module together onto the same platform, facilitating the user to acquire the benefits of both simultaneously. -->MCQ and Tests : Learning overnight for an exam and forgetting everything just after it finishes is never a suggested way of learning. In the same way, learning new words or increasing the vocabulary, requires evaluation at every levels or intervals to ensure efficiency and firmness. issue.

Conclusion	This project aims to provide an easy platform to learn and master the English language with modern ways of technology. It includes the correctness of spelling and meaning with end results of achieving excellence in pronunciation. In future we are planning to improve the pronunciation i.e. sound accuracy by incorporating appropriate filtering techniques. Comparative study of the existing TTS and STT algorithms are performed and work has to be done to improve the performance & improve the quality of the output. The project has been planned to be designed in a way that it is the complete course learning process for the betterment of pronunciation of the users struggling to achieve at convenience.

Title	Real Time Speech to Text Converter for Mobile Users
Author	Ms. Anuja Jadhav , Prof. Arvind Patil
Abstract	Mobile phone usage in World is spreading rapidly and has gone through great changes due to new developments and innovations in mobile phone technology. This project based on evaluating voice versus keypad as a means for entry and editing of texts. In other words, messages can be voice/speech typed. The project will make use of a dictating-machine prototype for the English language, which recognizes in real time natural-language sentences built from a 2000 word vocabulary. A speech to text converter is developed to send SMS .It is found that large vocabulary speech recognition can offer a very competitive alternative to traditional text entry. This project converts speech into text. At the run time speech data is quire from microphone and converted into text speech frames the speech frames are then pass for preprocessing and after preprocessing of the sample frames HMM-based training is applied on speech frame. Functionally the project divided into three modules : 1.-->Speech Acquisition: In this phase speech samples are obtained from speaker at real time and stored for preprocessing. Speech acquisition require microphone to receive voice speech signals, Speech acquisition can be easily done by the microphone present in the mobile phone, In the acquisition phase the different M/C is depends upon the its own configuration, hence there is need to store the sample of different users to make system more compatible to any type of voice. To recognize the speech HMM-based automatic recognition was conducted. For continuous phoneme recognition, an 86% phoneme correct was achieved for the normal-hearing.To achieve speech preprocessing sphinx frame work is used this is the best tool found to acquiesce speech signals. Sphinx is design with high flexibility modularity.The Linguist translates any type of standard language model, along with pronunciation information from the Dictionary and structural information from one or more sets of Acoustic Models, into a Search Graph. The Search Manager in the Decoder uses the Features from the Front End and the SearchGraph from the Linguist to perform the actual decoding, generating results. At any time prior to or during the recognition process, the application can issue Controls to each of the modules, effectively becoming a partner in the recognition process.

2.-->Speech Preprocessing

The speech signals consist of background noise that need to be removed. The preprocessing reduces the amount of efforts in next stages. Input to the speech preprocessing is speech signals which then converted into speech frames and gives unique sample.

Steps:

1. The system must identify useful or significant samples from the speech signal. To accomplish this goal, the system divides the speech samples into overlapped frames.
2. The system performs checks for the voice activity using endpoint detection and energy threshold calculations.
3. The speech samples are then passed through a pre-emphasis filter.
4. The frames with voice activity are passed through a Hamming window. The system performs autocorrelation analysis on each frame.
6. The system finds linear predictive coding (LPC) coefficients using the Levinson and Durbin algorithm.
7. From the LPC coefficients, the system determines the cepstral coefficients and weighs them using a tapered window. The cepstral coefficients serve as feature vectors.

3.-->HMM Training

Training involves creating a pattern representative of the features of a class using one or more test patterns that correspond to speech sounds of the same class. A model commonly used for speech recognition is the HMM, which is a statistical model used for modeling an unknown system using an observed output sequence. The system trains the HMM for each digit in the vocabulary using the Baum-Welch algorithm. The codebook index created during preprocessing is the observation vector for the HMM model.

i)Temporal Cepstral Derivative

We can obtain improved feature vectors for the speech frames using temporal cepstral derivatives. We use them with the cepstral derivative if the cepstral coefficients do not have acceptable recognition accuracy.

ii) Vector Quantization

A codebook of size 128 is obtained by vector quantizing the weighted cepstral coefficients of all reference digits generated by all users. The advantages of vector quantization are:

-> Reduced storage for spectral analysis information.

-> Reduced computation for determining the similarity of spectral analysis vectors.

-> Discrete representation of speech sounds. By associating phonetic label(s) with each codebook

	4.-->HMM Recognition Recognition or pattern classification is the process of comparing the unknown test pattern with each sound class reference pattern and computing a measure of similarity(distance) between the test pattern and each reference pattern . The digit is recognized using a maximum likelihood estimate, such as the Viterbi decoding algorithm, which implies that the digit whose model has the maximum probability is the spoken digit.Preprocessing, feature vector extraction, and codebook generation are same as in HMM training. The input speech sample is preprocessed and the feature vector is extracted. then, the index of the nearest codebook vector for each frame is sent to all digit models. The model with the maximum probability is chosen as the recognized digit.
Conclusion	In this paper, isolated speech recognition and continuous text generation for users, Hidden Markov Model is used here to perform HMM-based automatic recognition With this software the mobile phone usage and communication among the mobile users increases. Call routers become easier for users, since they don't need to know how to spell a name in order to say it. It becomes easier for users who are driving or otherwise incapable of looking at keypads to interact with a system.

Title	QuickSuggest: Character Prediction on Web Appliances
Author	Ullas Gargi , Rich Gossweiler
Abstract	As traditional media and information devices integrate with the web, they must suddenly support a vastly larger database of relevant items. Many devices use remote controls with on-screen keyboards which are not well suited for text entry but are difficult to displace. We introduce a text entry method which significantly improves text entry speed for on-screen keyboards using the same simple Up/Down/Left/Right/Enter interface common to remote controls and gaming devices used to enter text. The paper describes QuickSuggest's novel adaptive user interface, demonstrates quantitative improvements from simulation results on millions of user queries and shows ease of use and efficiency with no learning curve in user experiments. With internet appliances there are many situations where a standard or touch keyboard is not available – internet capable televisions, gaming consoles etc. Often the surrogate is an Up-Down-Left-Right (UDLR) keypad and an on-screen keyboard. The difficulty of entering text restricts the fluid dialog between the device and the person. As devices provide more services and more content (e.g. televisions with access to the WWW and online video content), the suddenly vastly larger vocabulary for search exacerbates the text-entry task. -><u>Adaptive Text Input Interface</u> : Given a function that accepts a string prefix and returns an ordered list of next-occurring characters by probability, we developed a prediction-ring user interface. The goal was to move the most likely characters closer (reducing physical effort) without introducing significantly greater cognitive or visual search effort. Consider the navigation path required to enter the term ”BRADY BUNCH” on an alphabetic on-screen keyboard. It takes 57 clicks for the 11 character term, as shown in Figure 1. With the predictive ring overlay, the click count was reduced to 25 clicks (the minimum possible is 2 clicks per character, 22 clicks). The goal of this experiment was to obtain initial, informal results from people using the system. Ten subjects, five males and five females were asked to participate in a simple within-subject experiment. Half of the subjects owned a DVR.02. The subjects were given a remote control and sat in front of a 23-inch LCD screen. Half were asked to enter five shows using the standard TiVo layout and then the same five terms using the predictive ring layout and the other half did the predictive model first. We timed their button click rate, overall term entry speed and interviewed them about the two methods. The experiment presented a show term (e.g. ”LOST”) and an onscreen keypad that they could navigate with the remote control. They were asked to enter the show. When they completed the show, they pressed ”OK” on the remote and then went on to the next term.

	We presented the following shows: LOST, BRADY BUNCH, ENTOURAGE, FAMILY GUY, and HOUSE. The first show, LOST, was a practice term and was discarded.When the person was done, we asked them three questions to elicit feedback: Did they own a DVR? What were their thoughts on the two methods? Which would they prefer to have?
Conclusion	Based on both the informal user studies and large-scale statistical experiments on real user queries, QuickSuggest's predictive model shows merit as a light-weight "short cut" mechanism for character based entry when using web-enabled devices . Many appliances may want to have a simple input device rather than a full keyboard but still provide access to a large corpus of content. The predictive ring model helps balance visual search costs while reducing the distance to the target to reduce physical effort (Fitt's law)

Title	Joining Hands : Developing a Sign Language Machine Translation System with and for the Deaf Community
Author	Sara Morrissey , Andy Way
Abstract	This paper discusses the development of an automatic machine translation (MT) system for translating spoken language text into signed languages (SLs). The motivation for our work is the improvement of accessibility to airport information announcements for D/deaf and hard of hearing people. This paper demonstrates the involvement of Deaf colleagues and members of the D/deaf community in Ireland in three areas of our research: the choice of a domain for automatic translation that has a practical use for the D/deaf community; the human translation of English text into Irish Sign Language (ISL) as well as advice on ISL grammar and linguistics; and the importance of native ISL signers as manual evaluators of our translated output. In this paper, we discuss a data-driven approach to Sign Language Machine Translation (SLMT) for translating English text into ISL. We use this work as a vehicle to acknowledge and demonstrate the role members of the deaf community play in the research of accessibility aids. The remainder of the paper is constructed as follows. In section 2 we give a brief overview of ISL, the primary SL used in our work. Section 3 outlines the general SLMT process and overviews previous and current research in this area. A description of the choice of domain and the data processing is given in section 4 and our own system is described in section 5. In section 6 we discuss the experiments we have carried out, their evaluation and results. We conclude the paper in section 7 and outline the future direction of our work. **Sign Language Machine Translation :** SLs worldwide lack political recognition (Gordon, 2005) and are poorly resourced in comparison to their spoken language counterparts. This is evident in the area of SLMT research with the earliest papers in this area dating back only 18 years. Fewer than 10 groups within this time have attempted SLMT and for the most part these projects have been short-lived with varying degrees of success. In general, SLMT has followed the trend of mainstream MT towards data-driven approaches over rule-based or more linguistic approaches. Essentially, an SLMT system for translating spoken language text into an SL will take a sentence as input, run it through the system looking for the most likely translation based on pre-described linguistic rules or the best statistical match, for example, and reproduce the sentence in a textual format of the SL. Some systems stop at this point of the process, focussing mainly on the translation process; others fit an avatar to the text output that will sign the translated sentence in real SL **Data Selection and Processing** : A data-driven approach such as ours necessitates a corpus of text in the source and target language. Having chosen our domain, we found a suitable base corpus in the ATIS (Hemphill et al., 1990) dataset, a corpus that is frequently used in NLP and was derived from a speech dialogue system of air traffic queries and

	responses (e.g. “What flights are there from Cork to Dublin?”). Our translation methodology requires a bilingual dataset. As the ATIS corpus is an English corpus it was necessary for us to translate the original datasets into ISL. To ensure the authenticity of our data, we liaised with the Irish Deaf Academy to employ two native Deaf ISL signers for translation and consultation work. During this process, the signers were encouraged to translate the sentence into an authentic ISL sentence irrespective of the choice of English words and grammar. In order to ensure fluency and consistency, each translation and signed sentence was discussed between the signers. The lack of a formalised writing system for SLs leads to the issue of how to represent them during the translation process. Having considered methods such as Stokoe Notation (Stokoe, 1960), HamNoSys (Prillwitz, 1989) and SignWriting5, we have chosen to use manual gloss annotation to transcribe the ISL video data of the ATIS corpus for its adaptability. For this we used the ELAN video annotation toolkit6 to transcribe a semantic representation of the ISL in the videos
Conclusion	Our collaboration with members of the Deaf community as both consultants and facilitators has allowed us to effectively channel our research in the area of SLMT towards a practical goal, namely aiding D/deaf and HOH people in the accessibility of airport information announcements. To date, our research has primarily focused on the development and improvement of MT processes with the translated output being produced in annotated format. For the system to be of practical use to its intended users a signing avatar is required. With a view to this, it is intended to expand the annotation to include descriptive phonetic features of the signs which could feed into Poser 4 software shown in Figure 1 in section 4.1 to create on-the-fly signed sentences. The use of such a human-like mannequin allows for real ISL to be produced as similar to the natural language as possible. Ideally, in fully functioning software for airport use, announcements would be appearing on the screen in both text and avatar cover the preferences of the Deaf, deaf and HOH communities. This final stage necessitates manual analysis by native ISL signers. For this, we propose the use of formalised accuracy and fluency scales for evaluating the translated output. This will allow us to assess the performance of the system in terms of complete translation but also to gauge the practical usability of our work as the evaluators are from the intended user group.

Title	Text-To-Sign Language Synthesis Tool
Author	Maria Papadogiorgaki, Grammalidis Nikos, Dimitrios Tzovaras
Abstract	This document presents an approach for generating VRML animation sequences from Sign Language notation, based on MPEG-4 Face and Body Animation. The proposed application aims in providing a computer-based sign-language synthesis output for the deaf and the hearing impaired. Moreover the application may be used as a teaching tool for relatives of deaf people as well as people interested in learning the sign language. The application receives text sentences as input and provides as output 3D animated VRML sequences able to be visualised in any VRML compliant browser. The SignWriting system is a writing system for deaf sign languages developed by Valerie Sutton for the Center of Sutton Movement Writing, in 1974 [1]. Almost all international sign languages, including the American Sign Language (ASL) and the Brazilian Sign Language (LIBRAS), can be represented in the SignWriting system. Each sign-box (basic sign) consists of a set of graphical and schematic symbols that are highly intuitive (e.g. denoting specific head, hand or body postures, movements or even facial expressions). The rules for combining symbols are also simple, thus this system provides a simple and effective way for common people with hearing disabilities that have no special training in sign language linguistics, to write in sign languages. An efficient representation of these graphical symbols in a computer system should facilitate tasks as storage, processing and even indexing of sign language notation. For this purpose, the SignWriting Markup Language (SWML), an XML-based format, has recently been proposed [6]. An online converter is currently available, allowing the conversion of sign-boxes in SignWriting format (produced by SignWriter, a popular SignWriting editor) to SWML format. Another important problem, which is the main focus of this paper, is the visualization of the actual gestures and body movements that correspond to the sign language notation. Traditionally, dictionaries of sign language notation contain videos (or images) describing each sign-box, however the production of these videos is a tedious procedure and has significant storage requirements. On the other hand, recent developments in computer graphics and virtual reality, such as the new Humanoid Animation (H-Anim) [8] and MPEG-4 SNHC [3] standards, allow the fast conversion of sign language notation to Virtual Reality animation sequences, which can be easily visualized using any VRML-enabled Web browser. In this document, we present the design, implementation details and preliminary results of a system for performing such a visualization of sign-boxes, available in SWML. After the application of the proposed technique, the resulting sequences of MPEG-4 Face and Body Animation Parameters can be used to animate

	any H-anim-compliant VRML avatar using MPEG-4 SNHC BAP and FAP players, provided by EPFL [4]. The proposed technique has significant advantages: • Web- (and Internet-) friendly visualization of signs. No special software has to be installed except a VRML plugin to a Web browser, • Allows almost real-time visualization of sign language notation, thus enabling interactive applications, • Avatars can easily be included in any virtual environment created using VRML, which is useful for a number of envisaged applications, such as TV newscasts, automatic translation systems for the deaf, etc. • Efficient storage and communication of animation sequences, using MPEG-4 coding techniques for BAP/FAP sequences. Significant similar work for producing VRML animations from signs represented in the HamNoSys transcription system to VRML has been carried out by the EC IST ViSiCAST project[10], and its follow-up project “E-Sign[11]. Current extensions of HamNoSys are able to transcribe all possible body postures, movements and facial expressions [12] and significant work towards supporting MPEG-4 BAPs has been made. The main contribution of the proposed approach in this paper is the attempt to work towards the same direction for the most common and popular representation of Sign Languages, which is the SignWriting notation system. **System Evaluation** The system was evaluated using the proposed on-line system. This experimental Web application has already allowed us to identify problems with the synthesis of static and dynamic gestures, which have to be solved in the future, e.g. when contacts and complex movements are involved. A major problem that has to be solved occurs when the sign-box contains contact symbols. In that case the touch between the hands, or the hand and the face is difficult to be achieved. Problems may also occur for complex movements, when the inclinations of the hand joints, which have been estimated in each key frame, are not accurate enough for the exact description of the movement. In the future, improved reproduction of difficult movements (e.g. touching) will be made using inverse kinematics techniques as in [5]. There was no systematic evaluation of the system with real users till now. Further evaluation is planned for the future, using Greek and International SignWriting users, and attempts will be made to solve possible problems in the reproduction of specific signs.

Conclusion	A demonstrator for generating VRML animation sequences from Sign Language notation, based on MPEG-4 Body Animation has been developed. The system is able to convert almost all hand symbols as well as the associated movement, contact and movement dynamics symbols contained in any ASL sign-box. Furthermore, most facial expression and animation symbols are also supported, while torso movements will be also supported in the near future. Some facial expressions, e.g. cheek wrinkles, have not been implemented, since no FAPs exist to produce such movements. Results are currently being evaluated by SignWriting users and experts so that problems associated with specific Sign-Writing symbols are identified and solved. In the future, improved reproduction of difficult movements (e.g. touching) will be made possible using inverse kinematics techniques. A short-term goal is to design other practical applications of the proposed system, either as a "plug-in" to existing applications (e.g. sign language dictionaries) or as a stand-alone tool for creating animations for TV newscacts (e.g. weather reports). Particular emphasis will be given in applications that can be used and evaluated by national Sign Language communities (e.g. the Greek Sign language), thus many dictionaries of Sign languages, in SignWriter notation, are planned to be supported in the future.

Title	Intralingual speech-to-text-conversion in real-time: Challenges and Opportunities
Author	Susanne Wagner (Halle)
Abstract	Intralingual speech-to-text-conversion is a useful tool for integrating people with hearing impairments in oral communication settings, e. g. councelling interviews or conferences. However, the transfer of speech into written language in real time requires special techniques as it must be very fast and almost 100% correct to be understandable. The paper introduces and discusses different techniques for intralingual speech-to-text-conversion. **The need for real-time speech-to-text conversion** Language is a very fast and effective way of communicating. To use language means to express an unlimited amount of ideas, thoughts and practical information by combining a limited amount of words with the help of a limited amount of grammatical rules. The result of language production processes are series of words and structure. Series of words are produced – i.e. spoken or signed – in a very rapid and effective way. Any person can follow such language production processes and understand what the person wants to express if two preconditions are fulfilled the recipients must: 1. know the words and grammatical rules the speaker uses and 2. be able to receive and process the physical signal. **The challenges of speech-to-text-conversion in real-time** Real-time speech-to-text-conversion aims at transferring spoken language into written text (almost) simultaneously. This gives people with a hearing impairment, access to the contents of spoken language in a way that they e.g. become able to take part in a conversation within the normal time frame of conversational turn taking. Another scenario for real-time speech-to-text-transfer is a live broadcast of a football match where the spoken comments of the reporter are so rapidly transferred into subtitles that they still correspond to the scene the reporter comments on. An example from the hearing world would be a parliamentary debate which ends with the electronic delivery of the exact word protocol presented to the journalists immediately after the end of the debate. (cf. Eugeni, forthcoming) **Message Transfer** The main aim of speech-to-text transfer is to give people access to spoken words and auditory events almost simultaneously with the realization of the original sound event. However, for people with limited access to spoken language at a young age, 1:1 transfer of spoken words into written text may sometimes not be very helpful. If children are not sufficiently exposed to spoken language, their oral language system may develop

	more slowly and less effectively compared with their peers. As a result, many people with an early hearing impairment are less used to the grammatical rules applied in oral language as adults and have a less elaborated **Computer-assisted note taking (CAN)** With computer-assisted note taking (CAN), a person writes into an ordinary computer what a speaker says. However, as was discussed earlier, even professional writing speed is not sufficient to write down every word of a speech. To enhance writing speed, abbreviation systems are used in computer-assisted note taking which minimize the amount of key strokes per word. The note taking person types abbreviations or a mixture of abbreviations and long forms. An abbreviation-to-long-form dictionary translates the abbreviations immediately into the corresponding long form. On the screen, every word appears in its long form.
Conclusion	Real-time speech-to-text transfer is already a powerful tool which provides people with a hearing impairment access to oral communication. However, elaborated dictionaries as they are needed for efficient CAN- or CART-systems are not yet developed for many languages. Without those dictionaries, the systems can not be used. Linguistic research has to find easy but efficient strategies for the real-time adaptation of the wording in order to make a message understandable also for an audience with limited language proficiency. Finally, the optimal presentation of moving text to an audience with diverging reading abilities is a fascinating research field not only for real-time speech-to-text services but with respect to the presentation of movable text in general.

Summary of the Literature Reviewed

The literature reviewed basically helped us in getting acquainted with the various techniques through which our two phases – Speech-to-Text and Text-to-Speech could be made to work efficiently. We looked at various tools that have been developed already for the deaf community for sign language conversion. Also, the use of Higher Markov Model for the purpose of training and recognition has been suggested. Apart from this, the need and challenges in real time conversion of speech-to-text were also studied.

As a part of our future work on the project, various research papers on Predictive Texting and Text Completion were also looked upon so that the rate of communication could be increased and the process of communication between the impaired and normal ones could be made more efficient.

CHAPTER 3
SYSTEM DEVELOPMENT

System development specifies the details of the application or the implemented model proposed in the project . It specifies whether the proposed model works on the concept or guidelines of *analysis, computation, experiments* or *mathematic results* . It displays sequentially the analysis done in the project, its basic outline or design, the development features and the algorithm used to successfully implement the idea practically

PROPOSED MODEL AND PROJECT DESIGN

- The project aims at bridging the communication gap between the impaired and the normal ones and working on the
- It includes a module for the conversion of speech to text and further sign language, for the audio impaired .
- Next, it comprises of another module for the conversion of sign language to text and further to speech, for the speech impaired .
- It also includes Google Web Speech Recognition module, which provides a web interface for speech input .

Hence the model proposed works under the *computational guidelines* and comprises of two procedural phases :

.1. Speech to Gesture

2. Gesture to Speech

1.**<u>Speech To Gesture</u>**

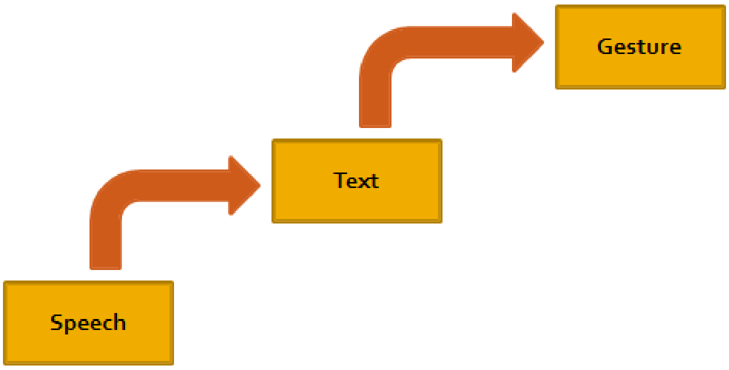


Fig.1 Phase-1

Phase-I includes a speech recognizer web interface which uses **Google Web Speech Recognition Module** which converts the input speech through a microphone, into text , which further feeds the converted text to the Python module in order to convert the text into corresponding sign language images, acting as a sign language tutor and convertor as well .

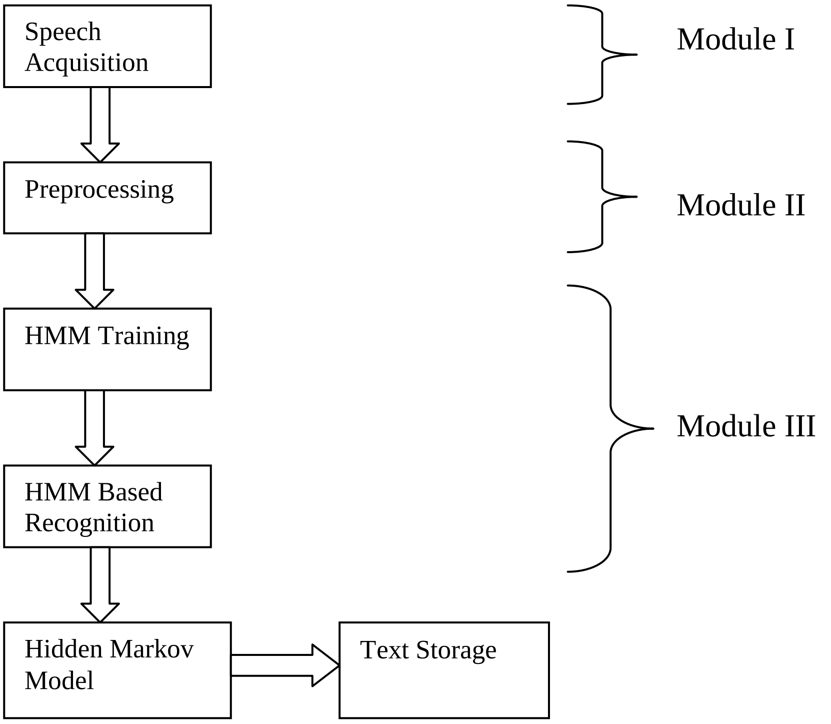


Fig.2 Speech to Text Conversion System ;it is divided into 3 modules

2.**<u>Gesture to Speech</u>**

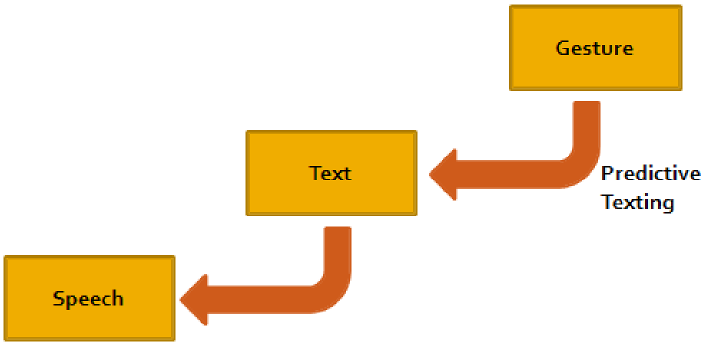


Fig.3 Phase-II

Phase-II includes the implementation of a gesture vocalizer i.e. a hand glove , combined with flex sensors and the whole setup is made into a device and integrated with the Arduino UNO Development Board .

Now considering the various scenarios wherein the system could be useful :

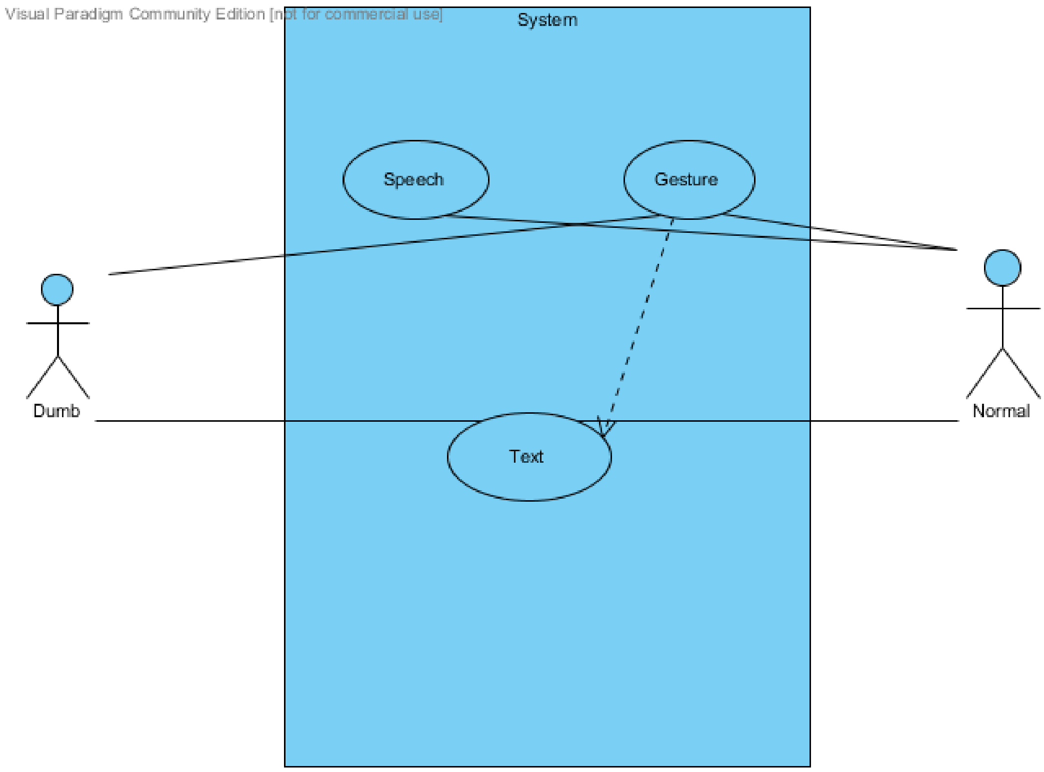


Fig.4 Conversation between speech impaired and normal people

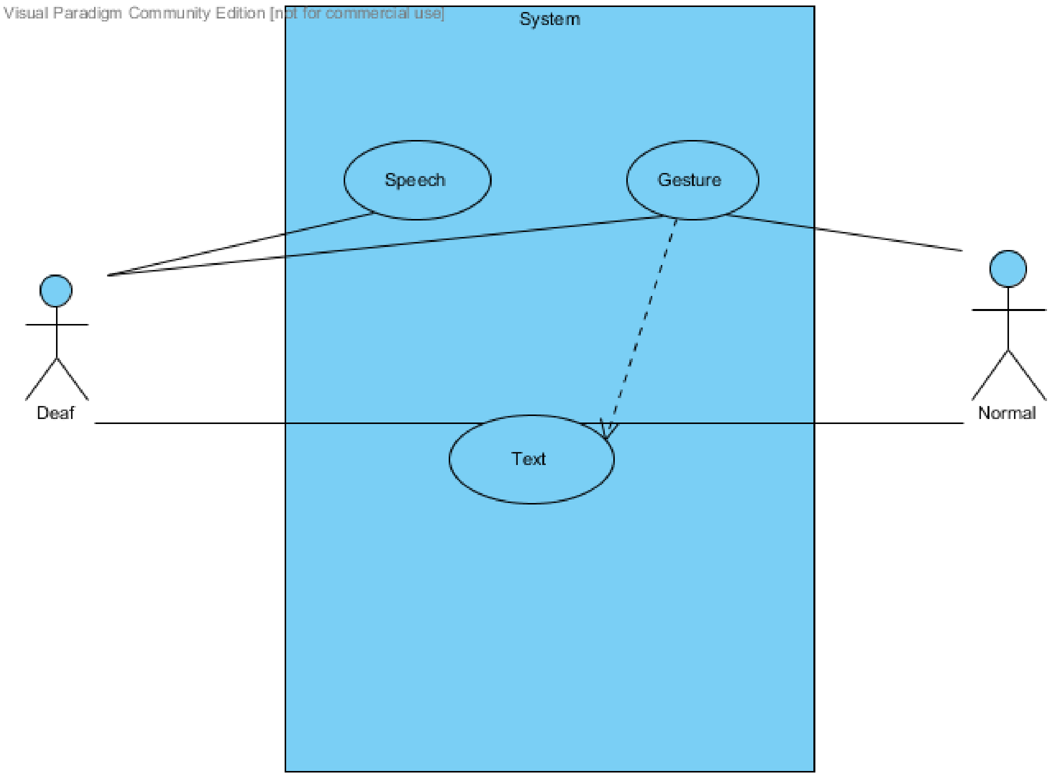


Fig.5 Conversation between audio impaired and normal people

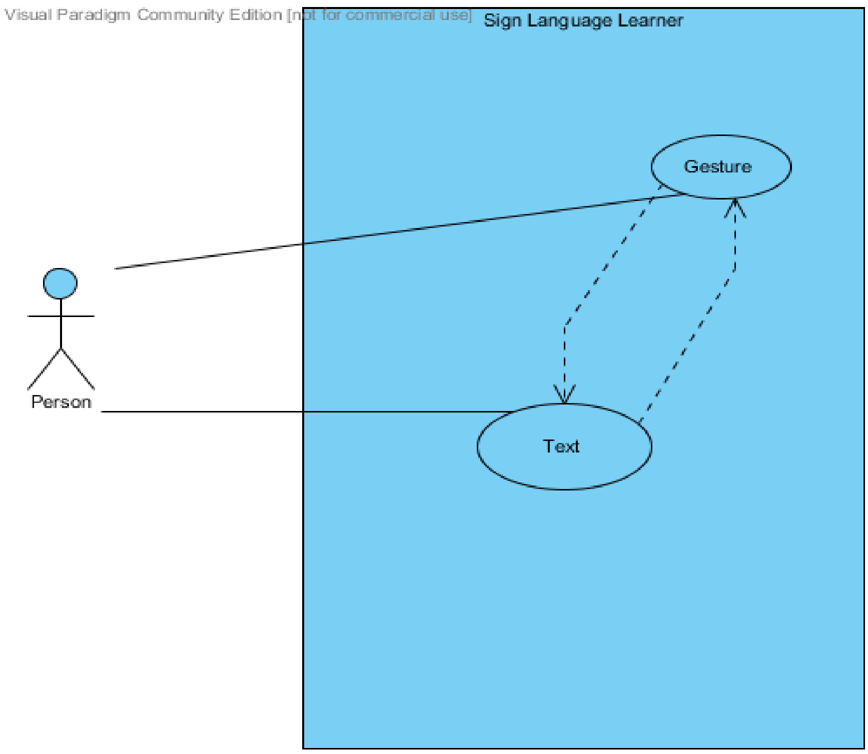


Fig.6 Use of the system as a sign-language tutor

HARDWARE REQUIREMENTS

The minimum requirements needed to perform operations are

- Arduino UNO Development Board
- Flex Sensors(Bend Sensors)
- Gesture Glove(Gesture vocalizer)
- Bluetooth Sensor(HC-06)
- LCD screen

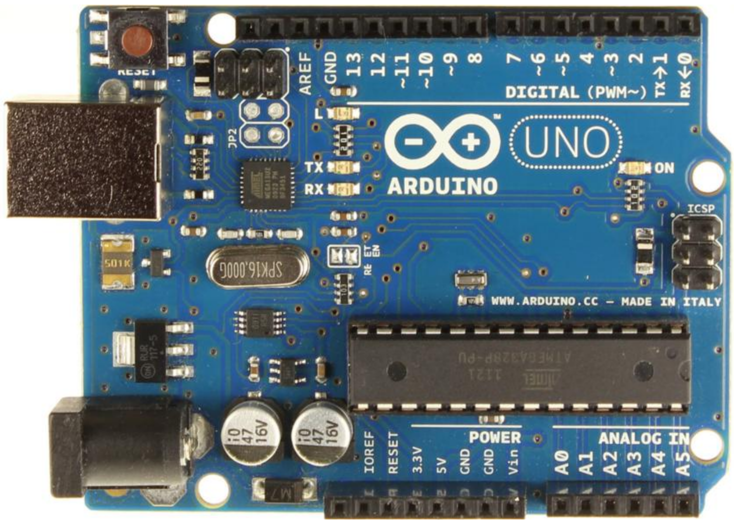


Fig.7 Arduino UNO

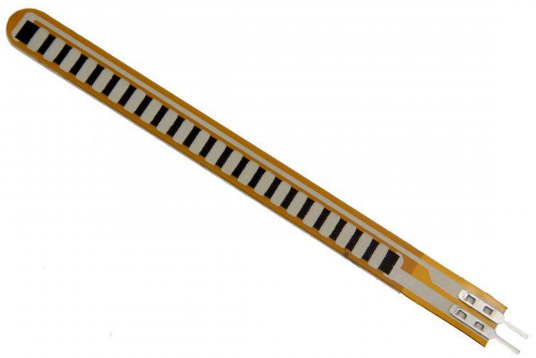


Fig.8 Flex Sensor

Fig.9 HC-06 Bluetooth module

Fig.10. 16X2 LCD Screen

SOFTWARE REQUIREMENTS

The software required to perform the implementation are

- Windows or Linux Operating System (Ubuntu)
- Python 2.7 or 3.0
- Python libraries like Tkinter, Pyspeech, Image
- Postimage.org
- Notepad++
- Python IDLE
- Google Web Speech Recognition
- ArduiDroid mobile application
- Arduino 1.6.1 IDE

CHAPTER 4
PERFORMANCE ANALYSIS

Performance Analysis is the provision of objective feedback to the performers or the developers undertaking or implementing a project trying to get a positive change in **performance**.

Many networks and apps have their own interface that programmers can work with. These interfaces are called APIs (short for Application Programming Interface). Here also, a web interface is provided for speech input (through a microphone) and also, Arduino 1.6 software is used, which provides an interface to the user to handle or feed input to the hardware.

Following are the results and outputs after implementing **Phase-I** procedurally:

- Setting up the database i.e. finding appropriate sign language images and resizing and restructuring them according to the canvas .

Fig.11 Image database

- Proper retrieval of sign language images according to the corresponding voice input fed into the **Google Web Speech Input** through a microphone, further fed into the software implemented in Python 2.7 . Proper output is shown on the Python 2.7.5 Shell and appropriate images corresponding to the text are shown on Tkinter **tk** canvas .

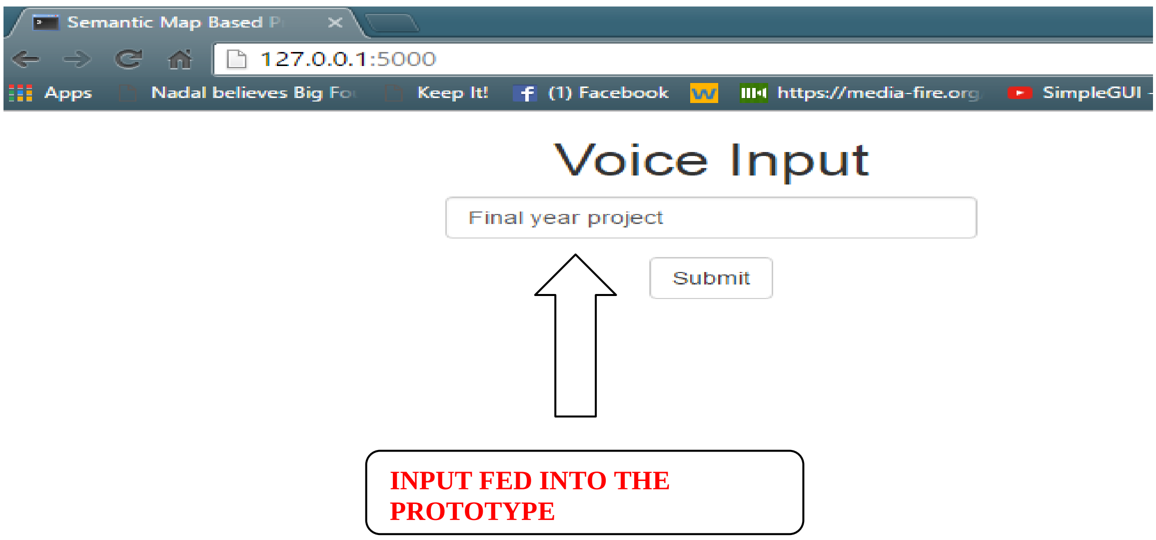


Fig.12 Input on the local server (Google Web Speech)

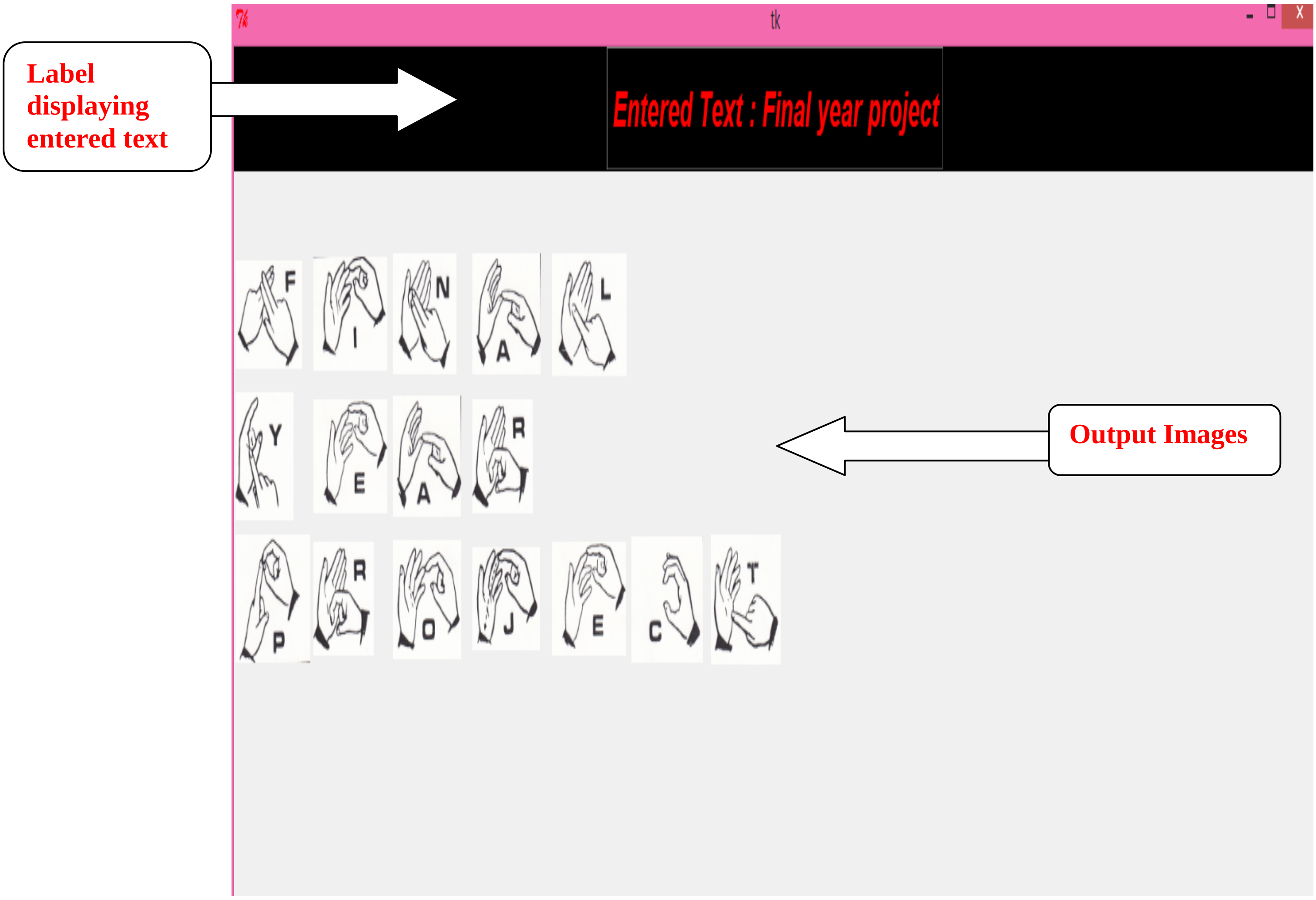


Fig.13 Corresponding image output on the 'tk' canvas

Following are the results and outputs after implementing **Phase-II** procedurally:

- Making up the **gesture vocalizer glove** i.e. attaching the flex sensors on the four fingers of the glove and attaching them to the Arduino board through jumper wires and making up the necessary circuit on the breadboard .
- Coding done (in Embedded C)in **Arduino 1.6.1** to assign values or letters according to the varying voltage spikes, caused by the bending of flex sensors attached on the fingers .
- Transmission of result messages or converted text(from gestures) from the Arduino to the mobile phone through the **HC-06 Bluetooth module**, and the messages are received and read through **ArduDroid** mobile application.

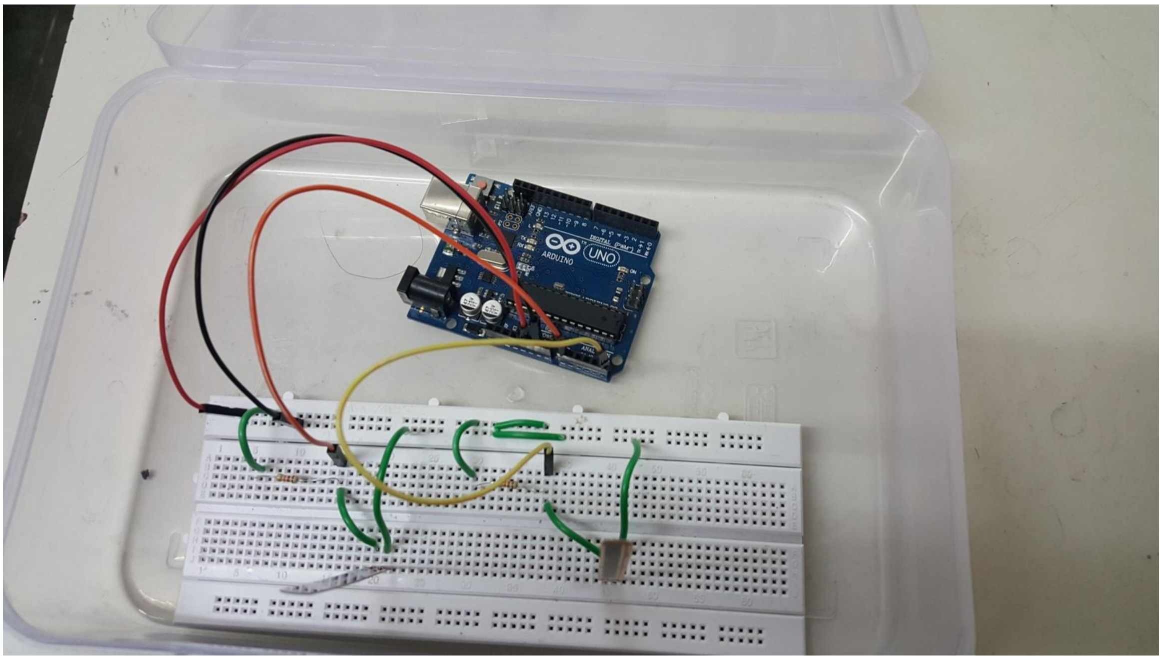


Fig.14 Overhead view of the circuit

LIMITATIONS OF SOLUTION

The proposed model provides an efficient tool for bridging the communication gap between the audio impaired and the normal people. The two phases of the model – Gesture-to-Text and Text-to-Gesture provide accurate conversions between the two forms.
Given these facts, there still are certain areas where more work needs to be done.

- The glove is fitted with sensors and uses a chip for recognition and conversion of gestures. Thus, making this tool wearable for regular use is something that is still a challenge.
- Moreover, the process of conversion from one form to the other takes some time, that is a bit extra than the normal communication rate. This does affect the efficiency of the mode for now. As a solution to this problem, addition of a Predictive Texting or Text Completion module to the model would help improve the communication rate and thus achieve the desired efficiency.
- Another limitation is the use of particular hardware equipments like flex sensors and bluetooth module, which are not very easily and abundantly available
- Conversion of gesture converted text into sound through an IC and speaker is also a limitation, which could not be done due to unavailability of hardware in the nearby proximity . It is considered in the future scope of the project.
- Also, the range of voltage spikes is not very high with corresponding bend in fingers . So, voltage range provided for each alphabet is quite less and hence, a bit prone to errors .
- The Flex sensors used in the project are even quite expensive as compared to other sensors preferrably used in other hardware based projects. Hence, cost factor is also an important constraint.

CHAPTER 5
CONCLUSION

5.1 **FINDINGS AND OBSERVATIONS**

The whole project consisted of integration of both hardware and software, and also was carried out in two phases. Hence, a lot of observations and findings have been recorded. Also, a lot of modifications needed to be done due to various factors like time, hardware availability, efficiency, accuracy etc.

Following are the findings and the modifications done and observed in the whole project timeline :

Phase I

- Earlier, **pyspeech** module was used and imported in the Python code, which connects with **Windows Ease of Access Speech Recognition** for integrating the speech recognition in the software.
- Windows Speech Recognition took in voice input from the microphone and displayed the text grasped from the microphone, and further sign images .
- . It was noticed or found that this approach was not completely efficient as the voice synthesizer was not very accurate. Also, the speech recognition servicer needed to be trained a lot.
- Hence, a web interface is now being used which inculcates in it **Google Web Speech Recognition**. It takes in voice input from the microphone similarly like previous module, translates it into text and feeds the text to the python code or python built software. This webpage works on the local server.
- Google Web Speech Recognition has been adopted as it is much more efficient, fast and much more accurate. Also, it is synced with google and hence, through certain inbuilt algorithms, continuously self trains itself, making it quite accurate and reliable.

Phase II

- **Flex Sensors,** also known as bend sensors are used as they show different resistance values according to the amount of bend. More the sensors are bent, more will be the resistance.
- These sensors are used as they are quite efficient and appropriate for the purpose of the project. According to different resistance values, different voltage values will be recorded as output. Hence, different alphabets or strings will be mapped to different values.
- Earlier, **Raspberry Pi (Model B)** was used in the project. This is because it is a small computer and is capable of computing a lot of functions quite efficiently, is quite preferred in hardware projects, and is easily programmable in Python 2.7.
- Pi didn't prove out to be successful as all the GPIO pins(General Purpose Input Output) are digital in Pi. And the flex sensors used in the project give analog output. Hence, analog to digital converter was needed for Pi(MCP 3008) which is quite expensive and not easily available.
- Hence, the project is shifted to **Arduino UNO Development Board.** This hardware is much cheaper than Raspberry Pi. Also, it has both analog and digital pins and inbuilt analog-to-digital converting pins. Hence it is preferred as it fulfills all the requirements of the project.

5.2 **CONCLUSION**

We presented an assistive technique for people with disabilities to communicate in a better manner. The proposed method will allow for more accurate communication between the impaired and the normal ones via the two suggested modules. It also bridges the gap between them and can be used as a wearable device by the disabled people for their convenience and ease .

Also, it can be used as a Sign Language learner to get the basic signs for different alphabets right and thus reducing the communication gap and increasing the rate of communication . It helps the normal people also in a way that they also easily grasp and comprehend what the disabled person wants to convey to them, resulting in efficient communication .

5.2 **PROPOSED FUTURE SCOPE**

- Integration of audio processing for conversion of text to speech using audio devices or audio processing ICs like **APR 33A3** for audio processing and playback sound . Also, microphone and speaker to record and play the audio message .

- Testing and improvement of the web interface, making it more interactive for users and making the system fast .

- Implementation of Predictive Texting with the help of **N-gram models**, **NLTK (Natural Language Toolkit)** and semantic mapping and integrating it with the hardware and shifting the whole project to the hardware device. This is done in order to reduce the number of keystrokes or reducing the amount of signs or gestures done by the impaired, making it more efficient.

- Improvement in the appearance of the hardware device and focussing on making it wearable and easily usable.

REFERENCES

[1] Al-Mubaid, Hisham, and Ping Chen. "Application of word prediction and disambiguation to improve text entry for people with physical disabilities (assistive technology)." *International Journal of Social and Humanistic Computing* 1.1 (2008): 10-27.

[2] Tom Stocky, Alexander Faaborg, Henry Lieberman. (2004**)** "A Commonsense Approach to Predictive Text Entry.*" Proceedings of Conference on Human Factors in Computing Systems. April 24-29, Vienna, Austria.*

[3] Dunlop, Mark D., and Michelle Montgomery Masters. "Investigating five key predictive text entry with combined distance and keystroke modelling."*Personal and Ubiquitous Computing* 12.8 (2008): 589-598.

[4] Schmidt, Maik, and Jaquelyn Carter. “Raspberry Pi: A Quick-start Guide”. Pragmatic Programmers, 2014.

[5]Morrissey, Sara, and Andy Way. "Joining hands: Developing a sign language machine translation system with and for the deaf community." (2007).

[6] Al-Mubaid, Hisham, and Ping Chen. "Application of word prediction and disambiguation to improve text entry for people with physical disabilities (assistive technology)." *International Journal of Social and Humanistic Computing* 1.1 (2008): 10-27.

[7] Gargi, Ullas, and Rich Gossweiler. "QuickSuggest: character prediction on web appliances." *Proceedings of the 19th international conference on World wide web*. ACM, 2010.

[8] Fung, Wong Jiang. "A Predictive Text Completion Software in Python."*Python Papers Monograph* 2 (2010).

[9] Murugan, MT Bala, M. Balaji, and B. Venkataramani. "SOPC-Based Speech-to-Text Conversion." *National Institute of Technology, Trichy* (2006).

[10] Deepa V.Jose, Alfateh Mustafa, Sharan R. "A Novel Model for Speech to Text Conversion." *International Referred Journal of Engineering and Science*

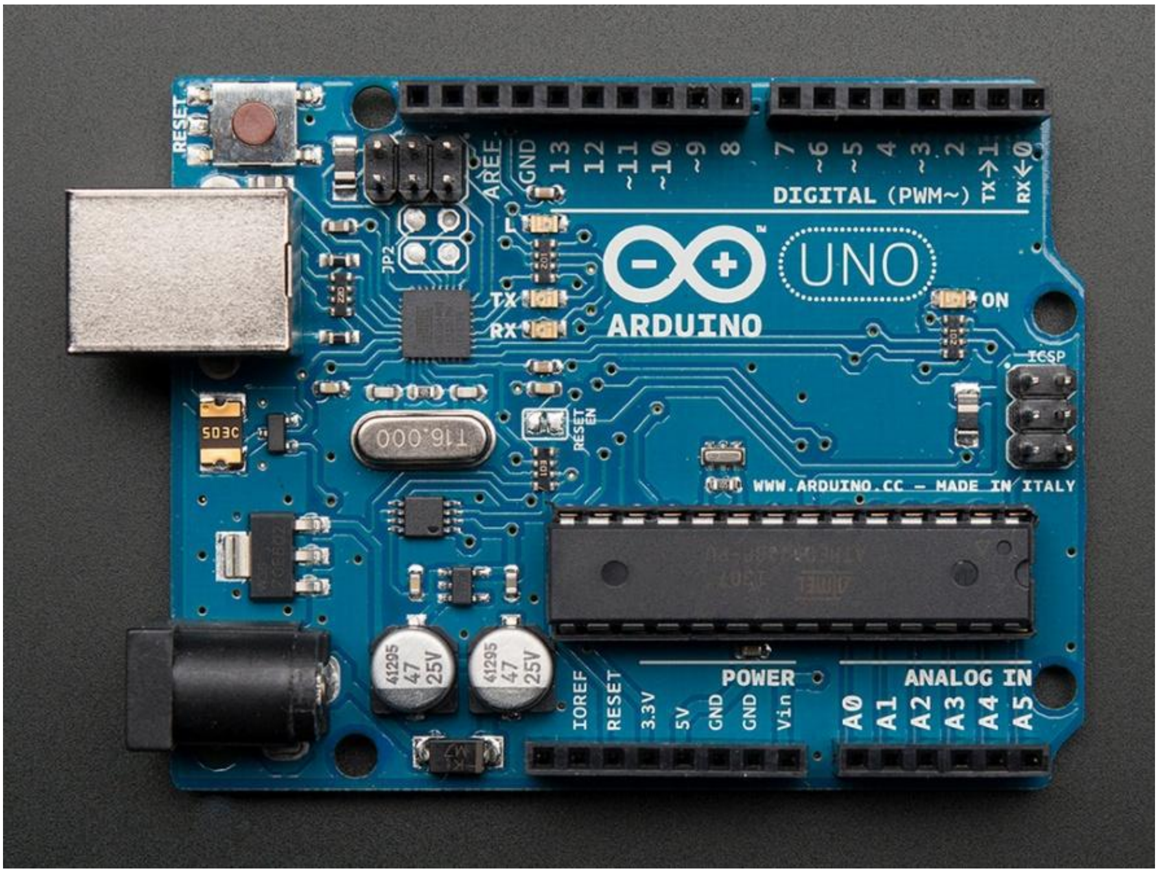
[11]Ms. Anuja Jadhav, Prof. Arvind Patil."Real Time Speech to Text Converter for Mobile Users" *National Conference on Innovative Paradigms in Engineering & Technology*.

[12] S.Mandeep , "Arduino and Its Working" ,2015,Online
Available : https://www.arduino.cc/en/main/arduinoBoardUno

[13] M.Rawashed ,"Arduino and Bluetooth Connectivity ",2015,Online
Available: http://www.instructables.com/id/Arduino-AND-Bluetooth-HC-05

APPENDIX

ARDUINO UNO



The Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller. "Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform.

Power

- Vin: The input voltage to the Uno board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

- 5V: This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.

- 3V: A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50mA.

- GND: Ground pins.

- IOREF: This pin on the Uno board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs to work with the 5V or 3.3V.

Memory

The ATmega328 has 32 KB (with 0.5 KB occupied by the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

- Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive 20 mA as recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50k ohm. A maximum of 40mA is the value that must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller.
- In addition, some pins have specialized functions:

- Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
- PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.
- LED: 13. There is a built-in LED driven by digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off
- TWI: A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library.

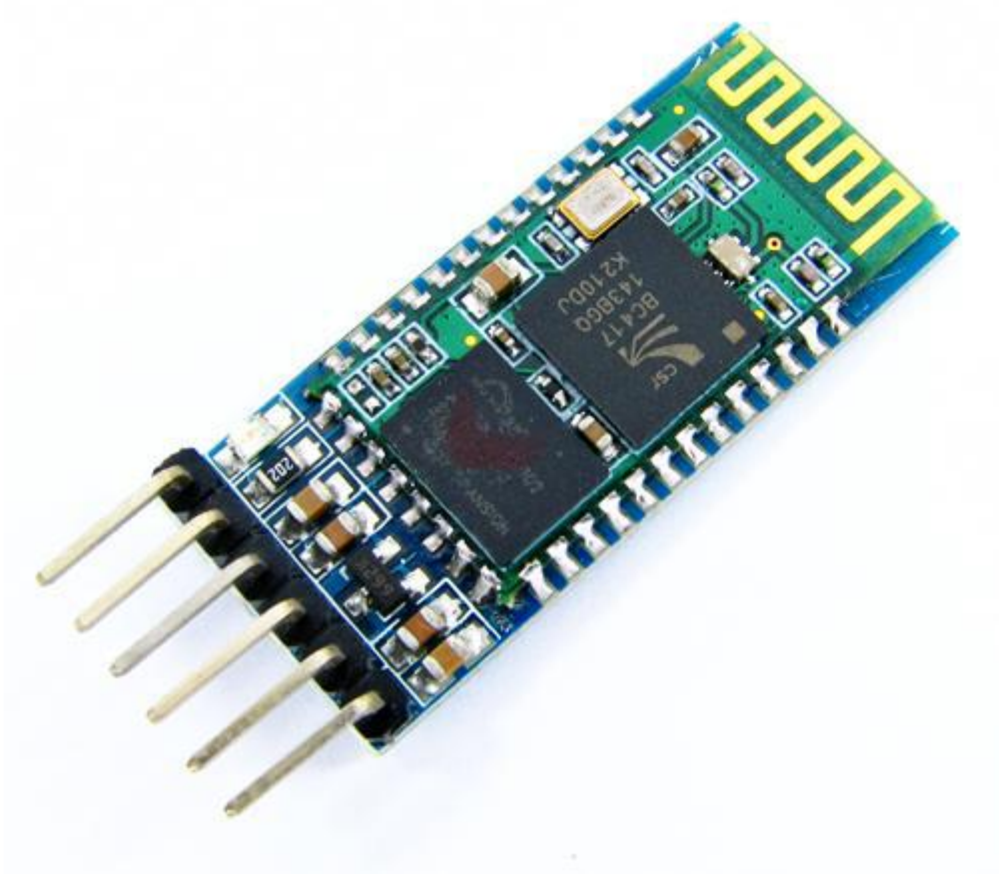
The Uno has 6 analog inputs, labelled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function. There are a couple of other pins on the board :

- AREF. Reference voltage for the analog inputs. Used with analogReference().
- Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Communication

The Uno has a number of facilities for communicating with a computer, another Uno board, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The 16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required.
The Arduino Software (IDE) includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).A Software Serial library allows serial communication on any of the Uno's digital pins.The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino Software (IDE) includes a Wire library to simplify use of the I2C bus.

BLUETOOTH MODULE V2.0



HC-05 module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup.
Serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation with complete 2.4GHz radio transceiver and baseband.It uses CSR Blue core 04-External single chip Bluetooth system with CMOS technology and with Adaptive Frequency Hopping Feature. It has the footprint as small as 12.7mmx27mm.

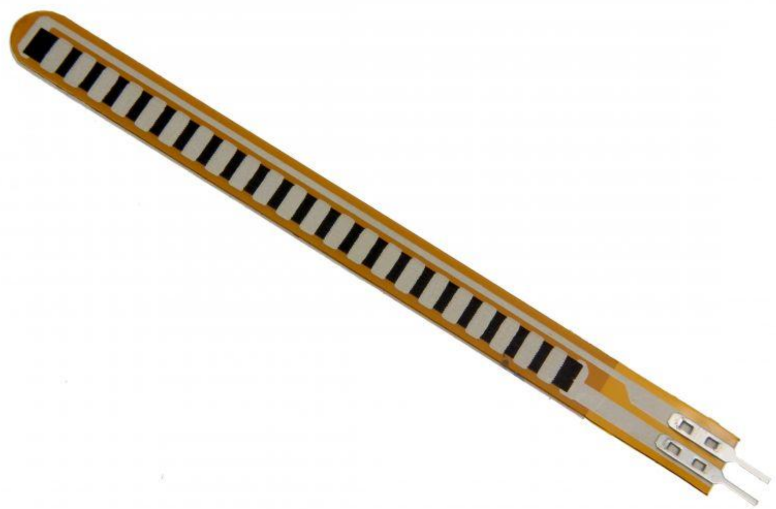
Hardware Features

- Typical -80dBm sensitivity
- Up to +4dBm RF transmit power
- Low Power 1.8V Operation ,1.8 to 3.6V I/O
- PIO control
- UART interface with programmable baud rate
- Integrated antenna
- Edge connector

Software Features

- Default Baud rate: 38400, Data bits:8, Stop bit:1,Parity:No parity,
- Data control: has supported baud rate: 9600, 19200, 38400, 57600, 115200, etc.
- Given a rising pulse in PIO0, device will be disconnected.
- Status instruction port PIO1: low-disconnected, high-connected;
- PIO10 and PIO11 can be connected to red and blue led separately.
- When master and slave are paired, red and blue led blinks 1time/2s in interval, while disconnected only blue led blinks 2 times/s.
- Auto-connect to the last device on power as default.
- Permit pairing device to connect as default.
- Auto-pairing PINCODE:"0000" as default
- Auto-reconnect in 30 min when disconnected as a result of beyond the range of connection.

FLEX SENSORS



The Flex Sensor patented technology is based on resistive carbon elements. As a variable printed resistor, the Flex Sensor achieves great form-factor on a thin flexible substrate. When the substrate is bent, the sensor produces a resistance output correlated to the bend radius—the smaller the radius, the higher the resistance value.

Spectra Symbol has used this technology in supplying Flex Sensors for the Nintendo Power Glove, the P5 gaming glove, and the below applications: Automotive controls, Medical devices, Industrial controls, Computer peripherals, Fitness products, Musical instruments, Measuring devices, Virtual reality games, Consumer products, Physical therapy. Spectra Symbol Designers can vary the actual nominal resistance of the Flex Sensors to meet customer's needs. We can produce our Flex Sensors on a variety of substrates, for example, we can use Dupont's Kapton material if you require high temperature operations.

Attributes :

Custom designed to match customer specs High level of reliability, consistency, repeatability Harsh temperature resistance Variety of flexible or stationary surfaces for mounting Infinite number of resistance possibilities and bend ratios