INTERACTION BETWEEN HEARING, SPEECH IMPAIRED AND A NORMAL PERSON VIA WEB APP

Vignesh S Upadhya¹, Sripad Arunkumar Wagdarikar², Rishik A Sunadoli³, Sumukha Bangalore Lokesh⁴, Dr. A R Aswatha⁵

¹₂₃₄Student, Department of Telecommunication Engineering, Dayananda Sagar College of Engineering, Bangalore, Karnataka, India.
⁵Professor and Head, Department of Telecommunication Engineering, Dayananda Sagar College of Engineering, Bangalore, Karnataka, India.

Abstract: Human life has become increasingly pleasant and simpler due to the advances in science and innovation. In the planet, there are almost 300 million individuals who are having hearing disability and almost 1 million who are speechless. It is difficult for a normal person to have an efficient interaction with these disabled individuals. Either he/she must learn the sign language, or he/she can communicate by writing the message which is inefficient and time consuming. There is a need of alternative way of communication among these individuals. This prototype provides efficient means of communication in which a normal person sends his input message through an URL (Uniform Resource Locator) web app to the AWS (Amazon Web Services) cloud and a BeagleBone Black is used as a receiver which retrieves the input message from the cloud and displays it on the display screen.

Key Words - Web URL app, Node-RED, AWS, BeagleBone Black.

I. INTRODUCTION

Interaction between deaf, dumb and deaf-dumb and a normal individual is difficult as a normal person can express his views naturally but for people with hearing and speech disabilities, it is not possible to express their views naturally and they cannot even understand the normal way of communication. They either need sign language or they need to see printed or written version of the message. So, the normal individual needs to learn and understand the sign language in order to communicate with these people.

For instance, let’s consider a normal individual wants to communicate with an individual with hearing disability. The normal person will not be able to communicate with the deaf person naturally. He needs to know the sign language in order to communicate or he must give the message in written format which is difficult and time consuming. So, there is a need for an alternative method for efficient communication.

The proposed module gives an efficient way of communication between these individuals through an URL based web app [1][2]. The Web app is developed using Node-RED, a flow-based visual programming platform specifically developed by IBM [3]. There are two ends to the web app namely the front end and the back end. The message input given by a normal individual through the URL web app is sent to the Amazon Web Services (AWS) cloud and the input is stored there [4]. The BeagleBone Black is burnt with a python code which is used to communicate with the AWS and the retrieve the message stored in it [5]. The message retrieved by the BeagleBone Black is displayed on a display screen.

II. METHODOLOGY

Fig1. Block diagram featuring the flow of message

As shown in the Fig1, the proposed work consists of a web app which takes inputs from the normal person and transmits it to the AWS cloud. The BeagleBone Black which is burnt with the python code which retrieves the message from the cloud and displays it on the display screen.

This proposed work makes use of Amazon Web Services (AWS). Amazon Web Services (AWS) is an Amazon company that offers on-demand cloud computing services and APIs on a metered pay-as-you-go basis to individuals, enterprises, and governments. In AWS,
an account is created which gives a public IP address. Then through GoDaddy.com, which is both a website hosting provider and a domain registrar — and has the resources to help create a website. It means that the tool can register the domain name, build a website for it and get it on the net-all in one place, the public IP address is registered and given a domain name as embetics.com. The message input given by the normal individual is stored in the AWS cloud.

A normal person gives his input through web app. The web app consists of two ends namely the back end and the front end. The web app is developed using the tool Node-RED, a flow-based visual programming platform specifically developed by IBM. In the back-end development, the Node-RED writes the code in JavaScript. here there is no need to learn the coding, as Node-RED does the coding by itself. The user just needs to select the function blocks and link them according to the need. The software based on the blocks connected, writes the code by itself in the Node.js platform. In the front end, the user can use the web app by entering the URL in the address bar and can give his input message. The input message can contain any number of characters.

The BeagleBone Black which acts as a receiver is installed with Linux operating system with ubuntu flavor. The python 3.4 version is installed in the BeagleBone Black to work with python programming. The BeagleBone Black does not have in-built Wi-Fi module. Therefore, it is configured to work with a dongle which provides internet connection. The BeagleBone Black is installed with http client package which is used to request the message from the cloud. A python program is written in the BeagleBone Black which uses the module to fetch the URL and request the message from the cloud and displays it on the display screen.

III. EXPERIMENTAL RESULTS

Fig2. Front end User Interface of Web app

Fig3. Message displayed on Desktop/PC screen

The Fig2 describes the front end user interface of the web app. Here the user gives text input. Once the BeagleBone Black requests the message, it can be seen in the command prompt window of a Desktop/PC as shown in Fig3.

IV. CONCLUSION

The described prototype provides an efficient way of communication between normal, deaf and dumb individuals. The proposed work makes technological access easier which was difficult to fetch earlier. It is portable, flexible and is also inexpensive. The software can be upgraded and installed easily and backups can be stored in the cloud as well. Some of the advantages are that the proposed model cutbacks on system hardware and a tailored integration process.
REFERENCES


