To Design Voice Control Keyboard System using Speech Application Programming Interface

Md. Sipon Miah[1], and Tapan Kumar Godder[2]

[1] Department of Information and Communication Engineering, Islamic University, Kushtia, 7003, Bangladesh.

[2] Department of Information and Communication Engineering, Islamic University, Kushtia, 7003, Bangladesh.

Abstract

With the passing of days men are more dependent on electronic devices. The Main objective of this project is to design and develop a voice Control Keyboard Systems, fully controlled by a computer, and display output on the display device with predefined time. So this project will work as a helping system for those person who has small knowledge about computer system even those person who are illiterate they can operate computer system. We can implement this developed system in other system for example voice control car system. This project voice is the input, sound application programming interface(SAPI 4.1) recognize this voice transfer the command to the microprocessor according to the programming code and display device displays the output.

Keywords: PC with Pentium microprocessor, a microphone, HMM and sound application software (SAPI).

1. Introduction

Day by day our life becomes busy. We have to do a lot of work everyday. For these purpose we use many kinds of computer control system. If we control the Computer system using voice then we can save enough time to do other sophisticated work. By doing this we can makes our work easier and faster.

There are several different possibilities for environment control for physically disabled persons. In many cases speech is the most convenient and easy-to-learn alternative. In this study, we wanted to explore the possibility of voice control without being involved in major hardware system development.

Thus, our guiding principle was that the system should connect easily to standard products for environmental control, be expandable and use standard equipment to the largest possible extent.

The person that volunteered to try out this application already had a sip-and-puff operated system for limited environmental control through infra-red (IR) techniques. Rather than replacing this system, we decided to expand it with voice control and new possibilities. In this way, we could use the old system as back-up in case of system break-downs and we also got an immediate evaluation of relative drawbacks and merits of the two control modes. In developed countries most of the sectors are computerized but in Bangladesh, computer is used mostly in education, office work and printing purpose. If we can control our computer using voice, we lead a smart and faster life. So we have to try to use this advantage. Because the Voice-control is trained to recognize the individual voice pattern, operation by a third person is only possible by means of command keys. Speaking a command is the same as selecting a command with the scroll key and confirming it with the OK key on the pilot or on an external keyboard. By connecting special operating peripherals (suck/blow switch, foot switch, head switch,...) to the keyboard interface, the Voice-control can also be operated by people with a speech impediment. The Voice-control is delivered together with configuration software, allowing the Voice-control to be adjusted to suit the individual. The software can be run on a standard PC under MS-Windows. (CPU Intel 80386 or higher, RAM 4 MB, MS-Windows V3.1 or higher) The selected commands are trained on the PC with the help of the Voice-control. Each word is spoken several times so that a common voice pattern can be analyzed. The voice pattern of the word is part of a neural network that allows speech to be recognized in operation. The voice patterns are stored in the pilot. Individual words can be retrained on the Voice-control even without a PC and configuration software.

2. Voice Control System

Voice-controlled systems have become more and more popular in the last few years, but in many cases the focus has been on what is technically possible to do, more than what people really want from their systems. We will focus on the human aspect, and try to figure out what the most intuitive way of communicating with a voice-

controlled system is. We are also interested in finding out how people adapt the way they talk when they are talking to a computer. Perhaps people want a shorter reply from the system than in other cases and also want to express themselves in shorter sentences? In order to build human friendly systems in the future we need to find out how people want their systems to work and perform instead of building more and more technically advanced systems that nobody asked for. We want to explore how different people talk when they talk to a computer compared with when they talk to a human. How does the communication differ with respect to syntax, pragmatics, phonetics and semantics, depending on whether they talk to a computer system or a human? Does a voice-controlled computer system have to be able to handle everything that a human can understand? Do people talk with longer sentences or do they choose to give short commands? What seems to be the intuitive way to speak? focus on voice control would be to find out how people really talk with the systems instead of how the systems work.

2.1 Dialogue Systems

A dialogue is a spoken, typed or written interaction in natural language between two or more agents. It is divided into units called turns, in which a single speaker has temporary control of the dialogue and speaks/writes for some period of time. Within a turn, the speaker may produce several spoken or typed utterances1. A turn can consist of more than one move. A move is a speech act in the sense that it is an act that a speaker performs when making an utterance, such as questions, warnings, statements etc2. It has a functional relation to the conversation of which it is a part. A dialogue system is a system that allows a human, the user, to use natural language in the interaction with the computer. In the same way, the computer replies with natural language. The natural language can be either spoken or written, either complete sentences or fragments of sentences. An example of a dialogue system is the SJ3 system.

2.2 Speech Recognition

Speech recognizers are computer systems that process human speech into something a computer can recognize and act on. There are several advantages in using speech in an application. For example, you can enter data when no keyboard or screen is available. It is also very convenient to use speech when hands or eyes, or both, are busy, and in difficult environments such as darkness or cold. Some areas where speech recognition is useful are: help for functional disabled people who are not able to type using their hands, telephone services where you have a very limited keyboard and when you need free hands, for example talking in your cell phone while driving speech recognizers into different categories. First divide them into speaker dependent or speaker independent. Natural-language speech recognition refers to computer systems that recognize and act on unconstrained speech. That is, the user does not need to know a predefined set of command words in order to use the system successfully (Boyce, 2000). Good speech recognition can be quite hard to achieve. This makes it difficult to find the word boundaries, compared to finding them in written text, which makes the words difficult to recognize. There is a great variability in speech between speakers depending on age, sex and dialect and speech within a speaker depending on mood, stress etc. External conditions such as background and recording device also make a difference.

2.3 Speech Synthesis

Speech synthesis is when the computer transforms textual or phonetic description into speech. A Text-To-Speech (TTS) synthesizer is a computer-based system that should be able to read any text aloud. There is a fundamental difference between this kind of system and any other talking machine (a cassette-player for example) because of the automatic production of new sentences that a TTS can perform.

2.4 Existing Voice-Controled Systems

Most car companies do some kind of research about voice-controlled systems today. BMW for example has voice-control in most of its vehicles7. Volvo is another car company that works with voice-controlled solutions in the car.

2.5 Syntactic Analysis

We did these transcriptions in order to be able to classify everything our test persons said into the different functions they wanted performed with their utterances, or the functions their utterances were related to. An utterance is a string of speech found between breath and pauses. Every test person's utterances were divided into stereo- and address book functions. The utterances which were related to the stereo, were divided into these categories:

- Change the volume
- Change tune/CD
- Turn on/turn off

The utterances which were related to the address book, were divided into these categories:

- Missed calls
- Check address book

- Add to address book
- Delete
- Change

We also divided the utterances to the computer system and the utterances to the human system into different columns. We were mostly interested in the utterances made to the computer system, since this is what we believe to be what a voice controlled system has to handle. This, we believe, is also very likely to be the way people will interact with a voice-controlled computer system. We were also interested in comparing the utterances to the computer system with the utterances to the human system.

As can be seen from the comments to Table No.2 the utterances from the computer system and the test leader are very similar.

Table 1: An example of utterances from a test person divided into respective functions

	Computer System	Human System
Stereo		
Change the volume	*-höj* (turn up)	*-kan du sänka lite* (can you turn down the volume a little)
Change tune/CD	*-byt låt* (change tune)	*-byt till Madonnas skiva* (change to Madonnas CD)
Turn on/turn off	*-sätt på stereon* (turn on the stereo)	*-vi kan ju börja med att sätta på stereon då* (we can start by turning the stereo on then)
Telephone		
Missed calls	*-finns det några missade samtal* (are there any missed calls)	*-kan du kolla om jag har några missade telefonsamtal* (can you check if I have any missed calls)
Check address book	*-sök efter Kalles telefonnummer* (check for Kalle's phone number)	*-kan du söka upp Karins telefonnummer* (can you check for Karin's phone number)
Add to address book	-lägg till telefonnummer (add phone number)	-kan du ändra Lisas nummer (2) *Lisas nya nummer är 031-50 50 50* (can you change Lisa's number (2) Lisa's new number is 031-50 50 50)
Delete	*-ta bort Karins telefonnummer* (delete Karin's phone number)	*-kan du ta bort Pelle från adresslistan* (can you delete Pelle from the address list)
Change	*-ändra Kalles nummer* (1) *031-60 60 60* (change Kalle's number (1) 031-60 60 60)	*-och så kan du ändra Karins nummer till 031-30 30 30* (and then you can change Karin's number to 031-30 30 30)

1 The computer answers *"vad vill du ändra numret till"* (what would you like to change the number to).
2 The test leader answers *"vad vill du ändra numret till"* (what would you like to change the number to).

2.6 Syntax of sentences uttered to computer system

46 NP ({PP, NP, FV NP})
39 FV PP ({NP, PP, AdvP, PP NP})
36 FV NP ({NP, AP, PP, AdvP, NP NP, NP PP}) 121
11 FV (AdvP) NP
7 FV AdvP ({NP, NP PP})
5 FV NP PP (NP)
4 AdvP NFV NP FV (AdvP) PP
4 AdvP FV NP ({AdvP, NP AdvP})
3 AdvP NFV NP NFV FV {AdvP, PP, AdvP NP}
3 NP FV {AdvP, NP, AdvP NP }
2 NFV NP FV (AdvP)
2 InterjP
2 NP NFV NFV FV {NP PP, PP}
2 FV NP (NP) PP NP NP NFV NP
1 AdvP NFV NP FV NP NP NFV PP
1 AdvP FV NP AdvP NP PP NP NP NP NFV NP
1 AdvP FV NP PP
1 AdvP FV PP NP NP NFV NP PP
1 NFV FV AdvP NP PP
1 NFV FV NP
1 NP NFV AdvP NFV FV PP
1 NP NFV NFV FV AdvP NP NP
1 NP NFV FV PP
1 FV NP PP NP NP PP NP NP
1 FV PP NP NP NFV NP 56

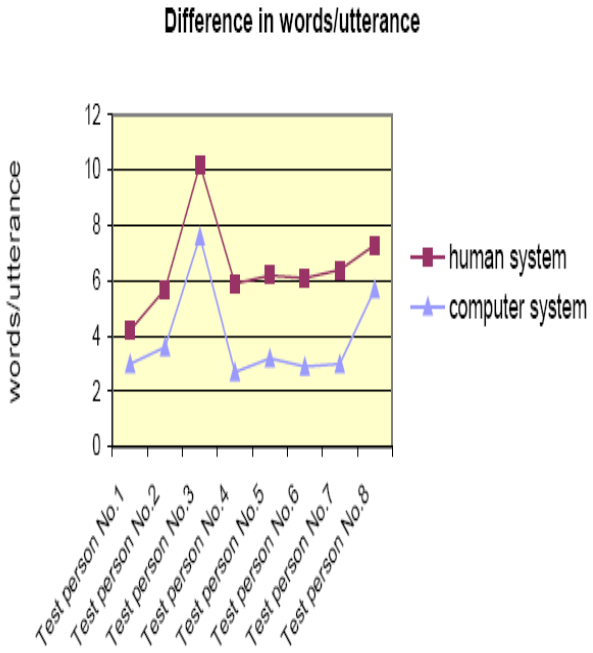

Fig. 1 The individual difference in words/utterance between the two tests.

3. Speech Recognition

Speech recognition has a history of more than 50 years. With the emerging of powerful Computers and advanced algorithms, speech recognition has undergone a great amount of progress over the last 25 years. The earliest attempts to build systems for automatic Speech recognition (ASR) was made in 1950s based on acoustic

phonetics. These systems relied on spectral measurements, using spectrum analysis and pattern matching to make Recognition decisions, on tasks such as vowel recognition [1]. Filter bank analysis was also utilized in some systems to provide spectral information. In the 1960s, several basic ideas in speech recognition emerged.

Zero-crossing analysis and speech segmentation were used, and dynamic time aligning and tracking ideas were proposed [2]. In the 1970s, speech recognition research achieved major milestones. Tasks such as isolated word recognition became possible using Dynamic Time Warping (DTW). Linear Predictive Coding (LPC) was extended from speech coding to speech recognition systems based on LPC spectral parameters. IBM initiated the effort of large vocabulary speech recognition in the 70s [3], which turned out to be highly successful and had a great impact in speech recognition research. Also, AT&T Bell Labs began making truly speaker-independent speech recognition systems by studying clustering algorithms for creating speaker-independent patterns [4]. In the 1980s, connected word recognition systems were devised based on algorithms that concatenated isolated words for recognition. The most important direction was a transition of approaches from template-based to statistical modeling especially the Hidden Markov Model (HMM) approach [5]. HMMs were not widely used in speech application until the mid-1980s. From then on, almost all speech research has involved using the HMM technique. In the late 1980s, neural networks were also introduced to problems in speech recognition as a signal classification technique. Recent focus is on large vocabulary, continuous speech recognition systems.

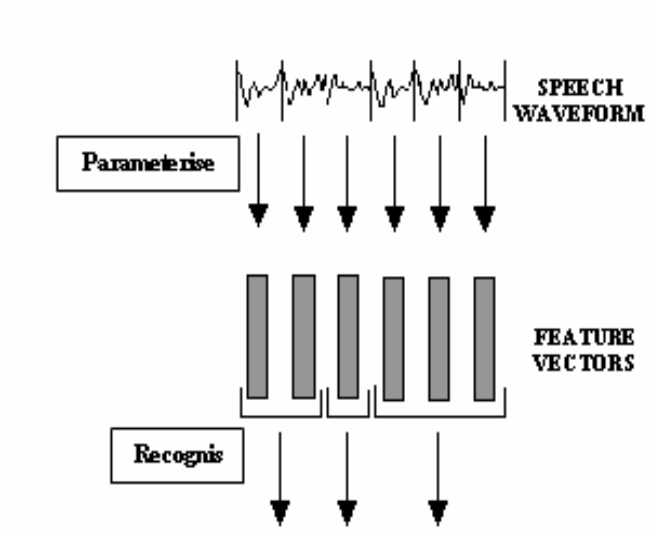

Fig. 2 Messages Encoding and Decoding in a ASR

3.1 Acoustic Model by Hidden Markov Model

Any acoustic unit, such as word, syllable, diphone, triphones or phone can be modeled by Hidden Markov Models (HMMs). An HMM is a finite state machine which can be viewed as a generator of random observation sequences according to probability density functions. The model changes state once at each time step and at time t a state j is entered, a speech vector o_t is generated from the probability density a_{ij}. The values of aij should satisfy

$$\sum_{j=1}^{N} a_{ij} = 1 \qquad (1)$$

where N is the number of states. They provide the temporal information in the HMM [6].

The quantity P(Y/W) is the probability of an acoustic vector sequence Y given a word sequence W to find the most probable word sequence. A simplistic approach to achieve this would be to obtain several samples of each possible word sequence, convert each sample to the corresponding acoustic vector sequence and compute a statistical similarity metric for the given acoustic vector sequence Y to the set of known samples. For large vocabulary speech recognition this is not feasible because the set of possible word sequences is very large. Instead words may be represented as sequences of basic sounds. Knowing the statistical correspondence between the basic sounds and acoustic vectors, the required probability can be computed.

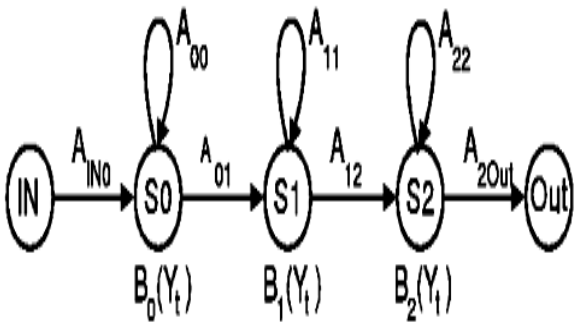

Fig. 3 Trip hones HMM

3.2 Deccoder or Recognizer

A decoder is a searching algorithm, which finds the corresponding word sequence. We given the maximum a posteriori probability P (W |O) for a spoken utterance O and its corresponding word string W. In the HMM based

recognition system, decoding is controlled by a recognition network. A recognition network consists of a word-level network, a dictionary and a set of HMMs. A word network describes the sequence of words that can be recognized and, for the case of sub-word systems, a dictionary describes the sequence of HMMs that constitute each word. A word-level network will typically represent either a finite-state Task Grammar which defines all of the legal word sequences explicitly or a Word Loop which simply puts all words of the vocabulary in a loop and therefore allows any word to follow any other word. Word-loop networks are often augmented by a stochastic language model (LM). A recognition network ultimately consists of HMM states connected by transitions.

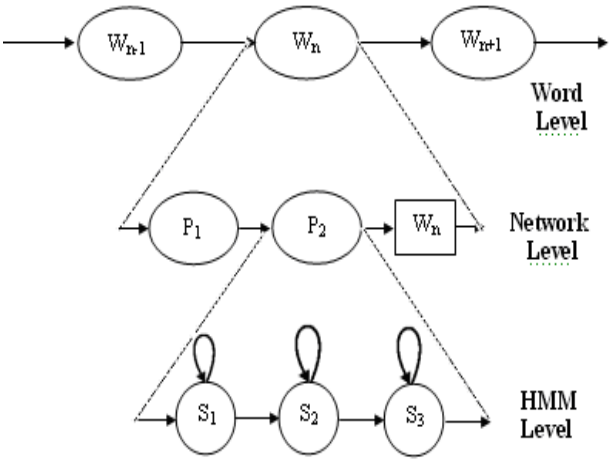

Fig. 4 Recognition Network Model

3.3 Parameterization

For input into a digital computer the continuous speech signal must first be converted into discrete samples which are then converted into a set of representative feature vectors. This parameterization process is often called the front-end of the speech recognition system.
The steps involve in converting speech signal into a set of parameters are shown in Figure 5.

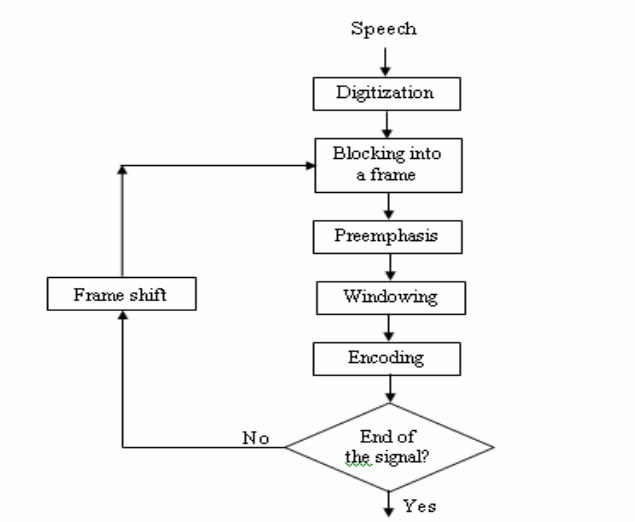

Fig. 5 Steps Sequence in Converting a speech signal into a set parameters suitable for ASR

The main purpose of the digitization process is to produce a sampled data representation of the speech signal with as high a Signal to Noise ratio (SNR) as possible [11].

The process of grouping digitalized speech into a set of samples, called frame, typically represented between 20 and 30 ms of speech. The digitized speech signal is blocked into overlapping frames [1]-[20] as shown in Fig. 10. The overlap decreases problems that might otherwise occur due to signal data discontinuity.

A one coefficient digital filter, known as a Preemphasis filter[11]. This stage spectrally flattens the frame using a first order filter. The transformation may be described as[12]:

$$Y_0[n] = x[n] - \alpha x[n-1], \quad 0.9 \le \alpha \le 1, \quad 0 < n < Samples\,per\,frame$$

Here, **X[n]** refers to the n^{th} speech sample in the frame. Sphinx uses **α= 0.97** and the sampling rate is typically 8K or 16K 16-bit samples per second.

Windows are functions defined across the time record which are periodic in the time record. They start and stop at zero and are smooth functions in between. When the time record is windowed, its points are multiplied by the window function, time bin by time bin, and the resulting time record is by definition periodic. It may not be identical from record to record, but it will be periodic (zero at each end). In the frequency domain, a window acts like a filter. The amplitude of each frequency bin is determined by centering this filter on each bin and measuring how much of the signal falls within the filter. If the filter is narrow, then only frequencies near the bin will contribute to the bin. A narrow filter is called a selective window, it selects a

small range of frequencies around each bin. However, since the filter is narrow, it falls off from center rapidly. This means that even frequencies close to the bin may be attenuated somewhat. If the filter is wide, then frequencies far from the bin will contribute to the bin amplitude but those close by will probably not be attenuated much.

The net result of windowing is to reduce the amount of smearing in the spectrum from signals not exactly periodic with the time record. The different types of windows trade off selectivity, amplitude accuracy, and noise floor.

Today, in speech recognition, the Hamming window is almost exclusively used. The Hamming window is a specific case of the Hanning window [11]. In this stage a Hamming window is applied to the frame to minimize the effect of discontinuities at the edges of the frame during FFT. The transformation is[12]:

$$Y_1[n] = x[n] \times H[n], \quad 0 < n < Frame\,size$$

The vector **H[n]** is computed using the following equation[12].

$$H[n] = 0.54 - 0.46 \times cos(\frac{2\pi n}{Frame\,size - 1})$$

The constants used in the **H[n]** transform were obtained from the Sphinx source code. In practice, it is desirable to normalize the window so that the power in the signal after windowing is approximately equal to the power of the signal before windowing. The purpose of the window is to weight, or favor, samples towards the center of the window [11].

Speech coding is the compression of speech into a code using audio signal processing and speech processing techniques. To encode the speech signal into a suitable set of parameters three basic classes of techniques are being used:

- Fourier transformations
- Filtering through digital filter-banks
- Linear prediction

Since the speech signal is not stationary, speech analysis for encoding must be performed on short-term windowed segments, usually, a duration of 20 to 30 ms with a frame period of 10 to 15 ms. Therefore, for short period of time (~10ms) the speech signal is quasi-stationary and this allows us to represent the signal over this period by a single feature vector.

4. Keyboard Interface

The PS/2 interface is a bit serial interface with two signals Data and Clock. Both signals are bi-directional and logic 1 is electrically represented by 5 V and logic 0 is represented by 0 V (digital ground). Whenever the Data and Clock line is not used, i.e. is idle, both the Data and Clock lines are left floating, that is the host and the device both set the outputs in high impedance. Externally, at the PCB, large (about 5 k) pull-up resistors keep the idle lines at 5V (logic 1).

The FPGA/keyboard interface is shown in figure 7. When the FPGA “reads” the Data or Clock inputs both PS2Data_out and PS2Clk_out are kept low which puts the tri-state buffers in high impedance mode. When the FPGA "writes" a logic 0 on an output, the corresponding x_out (x = PS2Data or PS2Clk) signal is set high which pulls the line low. When “writing” logic 1 the FPGA simply sets the x_out signal low.

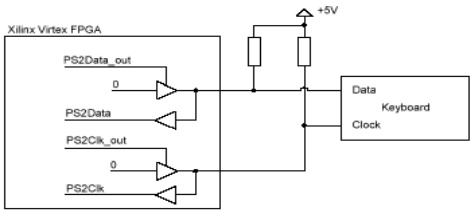

Fig. 7 FPGA/ Keyboard Interface

4.1 Protocol for receiving data from the keyboard

Data is received from the keyboard as illustrated in Figure 8.

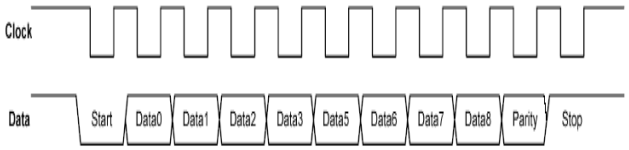

Fig. 8 PS/2 protocol

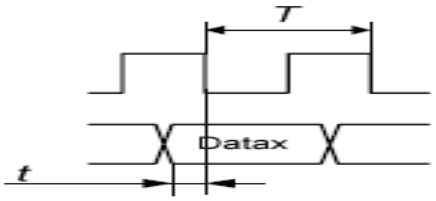

Fig. 9 PS/2 Timing

Data is sent in bit serially. The first bit is always a start bit, logic 0. Then 8 bits are sent with the least significant bit first. The data is padded with a parity bit (odd parity). The parity bit is set if there is an even

number of 1's in the data bits and reset (logic 0) if there is an odd number of 1's in the data bits. The number of 1's in the data bits plus the parity bit always add up to an odd number (odd parity.) This is used for error detection. A stop bit (logic 1) indicates the end of the data stream.

4.2 The keyboard scan-codes

The keyboard sends packets of data, scan codes, to the host indicating which key has been pressed. When a key is pressed or held down a make code is transmitted. When a key is released a break code is transmitted. Every key is assigned a unique make and break code so that the host can determine exactly what has happened. There are three different scan code sets, but all PC keyboards use Scan Code Set 2. A sample of this scan code set is listed in table 2. Please refer to the lab homepage for the full scan code set.

Table 2: Scan Code Set 2 (sample)

KEY	MAKE	BREAK
A	1C	FO, 1C
B	32	FO, 32
C	21	FO, 21
D	23	FO, 23
E	24	FO, 24
F	2B	FO, 2B
G	34	FO, 34
H	33	FO, 33
I	43	FO, 43
J	3B	FO, 3B
K	42	FO, 42
L	4B	FO, 4B
M	3A	FO, 3A
N	31	FO, 31
O	44	FO, 44
P	4D	FO, 4D
Q	15	FO, 15
R	1B	FO, 2D
S	1B	FO, 1B
T	2C	FO, 2C
U	3C	FO, 3C
V	2A	FO, 2A
W	1D	FO, 1D
X	22	FO, 22
Y	35	FO, 35
Z	1A	FO, 1A

4.3 Displaying scan codes

Receive the scan codes from the keyboard and display the corresponding code in hexadecimal format on the XSV board digit LEDs. The LEDs should be updated at a rate of approximately 1 Hz.

Following we will partition this problem into more manageable pieces. We will partition the design into a data path and a control path. A block diagram of the complete design is shown in figure 10.

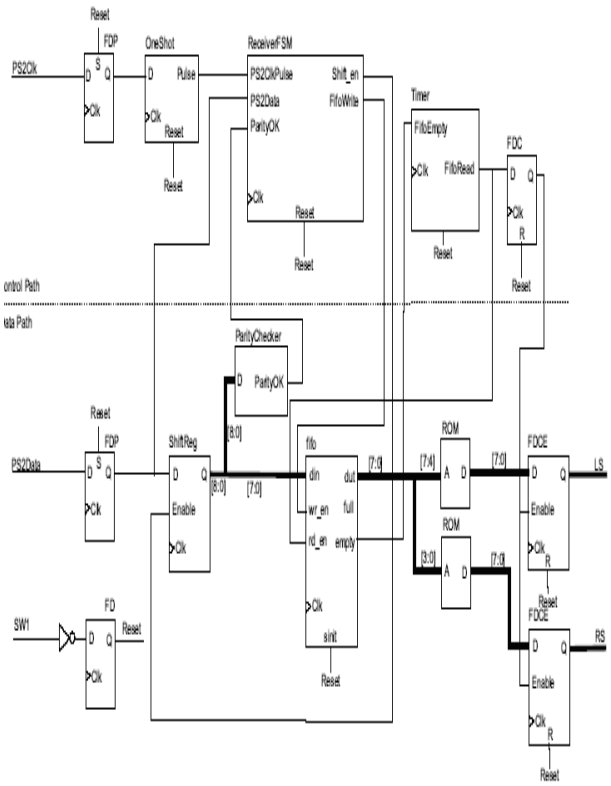

Fig. 10 A block diagram of a PS/2 Keyboard Interface

All flip-flops in the design are clocked with a 20 MHz clock and the rising edge is the active edge. Four types of positive edge-triggered D-flip-flops are shown in figure 4.FD No reset/set FDC Asynchronous reset FDP Asynchronous set FDCE Clock enable and asynchronous reset The PS2Data and PS2Clk signals are sampled with FDP flip-flops. The PS2Data is fed into a shift register. When 8 data bits (scan code) and one parity bit has been shifted into the shift register the scan code is written to a synchronous FIFO. When the FIFO is not empty the output is read in intervals of approximately 1 second. The FIFO output is connected to two identical ROMs. The ROMs decode the scan code data so that a character "0" – "F" is displayed on each digit LED.A short description of

each module in the data and control path follows. The keyboard scanning process is illustrated in Figure 8. First the column result is scanned. The three lower bits (column part) of the scan code is incremented until the low column line is found. Then, the port directions are inverted and the row result is scanned. The row part of the scan code is incremented until the low row bit is found. Finally, the scan code processing function is called.

5. Speech Application Programming Interface

The Speech Application Programming Interface (SAPI) is defined by Microsoft. Version 4 was published in the beginning of 1998 and its successor, version 5, was published in the beginning of 2001. Maybe the biggest weakness in SAPI 4 is the fact that there is no centralized control panel for speech synthesizer parameters. This means that user is required to select his or her favorite voice and adjust it individually for every speaking application. SAPI 5 adds the Speech item to the control panel. This control panel item is used by the user to define his or her favourite voice and other speech parameters SAPI (Speech Application Programming Interface) was first introduced to Windows 95. This API provides a unified interface for dynamic speech synthesis and recognition. Over the years new versions were developed and now it is version 4.0 with WinXP. Unfortunately the API wasn't really maturated and supported only C++ (later Visual Basic and COM), so it was quite widely used. Microsoft redesigned the version 5.0 from scratch and changed critical parts of the interface. However the latest stable version 5.1 is still a native code DLL, but with the next one, which is considered to be part of Windows Vista (a complete redesign again), A full support for managed .NET code will be expected [7]. Right now it is only possible to take advantage of the current SAPI interface via C# by using COM Interop, which is .NET technique to use native COM object Programming interfaces expose the full power of complex software engines. SAPI 5 is Microsoft's fifth iteration towards a speech application programming interface to enable programmers to build speech-enabled applications for Microsoft Windows platforms. It incorporates lessons that Microsoft has learned over many years and many APIs, in an effort to make the best comprehensive API possible. Even so, most programmers should do at least their prototyping and in most cases their product using a tool suite that does most of the SAPI 5 programming for them. Why? As always, to improve value and reduce costs. Microsoft SAPI is a little known speech recognition and synthesis engine that is supported in all versions of Windows after Windows 95. The Microsoft Speech SDK (System Development Kit) version 5.1 is available for free download from the Microsoft Speech Technologies Website. Speech SDK 5.1 is compatible with a number of programming languages, but most of the documentation focuses on examples written in C++. In general, any programming environment that supports OLE automation will work for writing SAPI applications.

6. Result and Discussion

In our project work, an attempt has been made to develop voice control computer system. Here computer is the central device. Microphone is the input device and voice is the input. For this system a microphone is connected to the pc via sound cord and then software was developed to accept the input processed by sound application programming interface(SAPI) software. Input was processed by microprocessor and display this.

We do believe that a voice-controlled system must be able to handle some kinds of disturbances in the utterance. The system must not interpret them as speech or as the ending of an utterance. This is especially important when the system is going to be used in a car since this is an environment where these disturbances can often occur since your attention is on the traffic and everything around you. When something happens that catches your attention you might, for example, make "unmotivated" pauses in your utterance to the computer, which the computer system does not have a clue why you are doing and probably will interpret it as if you are finished with your utterance. To actually build a voice-controlled system on the basis of our communication model and evaluate it would also be a great challenge. Thereafter, to do a study with a large group of test persons to find out how our system would work, would have been challenging. To do a study with a large group of test persons would also be interesting since we could see if the results could be statistically secure. When we created our communication model we worked on the basis of the specific environment of the car. Therefore our solution is custom-made for the car. However, it might be possible to use it for other areas where you use service gateways. For example when you want to use voice-controlled computers in your home environment.

System performance totally depends on the output of the system. The percentage of success rate and failure rate has been calculated using the following equations:

Success:

$$\text{Rate} = \frac{Total\ number\ of\ success}{Total\ number\ of\ Test} \times 100\%$$

Failure:

$$\text{Rate}=\frac{Total\ number\ of\ failure}{Total\ number\ of\ Test}\times 100\%$$

The performance is related to success rate and failure rate. If the success is high then the performance of the system is good. Success rate and Failure rate are contradiction of each other. So when success rate is high then failure rate is low. In the two terms the performance of the system is depended.

7. Conclusions

We have demonstrated a voice control system, for physically disabled persons, that operates with a minimum of specialized hardware. Some of the chosen solutions are dictated by the wish to build on the equipment already used by our test subject. For a more mobile person, the same solution could be used in connection with a portable computer. The telephone capability could be given a less expensive and more flexible solution through a standard modem. The use of a standard PC gives potential access to a wide variety of programs. Future expansions include improved software for text processing.

Acknowledgments

The authors would like to thank student Pulack Chakma [M.Sc.(Tech.)] for her assistance during the computer simulations and Md. Farzan Ali [M.Sc. (Mat.)] for the language checking.

References

[1] "Visual Basic v6 (Part 1-Database and Multimedia Programming)" By Mahabubur Rahaman
[2] "Visual Basic v6 (Part 2- Database and Multimedia Programming)" By Mahabubur Rahaman
[3] "Mastering Visual Basic 6 " By Michael j. Young
[4] "Mastering Visual Basic 6" By Evangelos petroutsos
[5] "Microprocessor and interfacing programming and hardware" By Douglas Hall,Tata McGraw-Hill Edition
[6] "Microprocessors and Microprocessors based system design" By Dr. M.Rafiquzzaman
[7] Spectral contrast normalization and other techniques for speech recognition in noise, . C. Bateman, et. Al
[8] "Speech enhancement based on masking properties of the auditory system"
[9] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," The Journal f the Acoustical Society of America, vol. 87, no. 4, pp. 17-29, 1987
[10] B. Atal and M. Schroeder, "Predictive coding of speech signals", Proc. of 6th International Congress on Acoustics, Tokyo, pp. 21-28, 1968
[11] H. Matsumoto and M. Moroto, "Evaluation of Mel-LPC cepstrum in a large ocabulary continuous speech recognition," Proc. ICASSP, pp.117-120, 2001
[12] P. H. Lindsay and D. A. Norman, "Human information processing: An introduction to psychology", 2nd Ed., pp. 163, Academic Press, 1977
[13] S. F. Boll, "Suppression of acoustic noise in speech using spectral subtraction," IEEE Trans. Acoust., Speech and Signal Processing vol. 27, no. 2, pp. 113-120
[14] H. W. Strube, "Linear prediction on a warped frequency scalle", J. Acoust. Soc. Am., vol. 68, no. 4, pp. 1071-1076, 1980
[15] H. Matsumoto, Y. Nakatoh and Y. Furuhata, "An efficient Mel-LPC analysis method for speech recognition",Proc. ICSLP '98, pp. 1051-1054, 1998
[16] H. Matsumoto and M. Moroto, "Evaluation of Mel-LPC cepstrum in a large vocabulary continuous speech recognition," Proc. ICASSP, pp.117-120, 2001
[17] Sub Auditory Speech Recognition" by Kim Binsted and Charles Jorgensen
[18] "Sub Auditory Speech Recognition Base on EMG/EPG Signals" by Chuck Jorgensen, Diana D. Lee, and Shane Gabon
[19] S. Itahashi and S. Yokoyama, "A formant extraction method utilizing mel scaleand equal loudness contour", Speech Transmission Lab.-Quarterly Progress and Status Report (Stockholm) (4), pp. 17-29, 1987
[20] Various websites On Internet

Md. Sipon Miah received the Bachelor's and Master's degree in Information and Communicatio Engineering from Islamic University, Kushtia, 2006 and 2007, respectively. He is currently Lecturer in the department of Information and Communication Engineering, Islamic University, Kushtia,Bangladesh.Science 2003, he has been a Research Scientist at the Communication Research Laboratory, the department of ICE, Islamic University, Kushtia, where he belongs to the spread-spectrum research group.He is pursing research in the area of internetworking in wireless communication. He has three published papers in international and national journals in the same area but some papers under process. His areas of include database system, interfacing, programming, optical fiber communication and wireless communications.

Tapan Kumar Godder received the Bachelor's, Master's and MPh degree in Applied Physics & Electronics from Rajshahi University, Rajshahi. In 1994, 1995 and 2007, respectively. He is courrently Associate Professor in the department of ICE, Islamic University, Kushtia-7003, Bangladesh. He has seventeen published papers in international and national journals. His areas of interest include internetworking, AI & mobile communication.