Acknowledgement:

This was a crucial project for us as it was the first time we were exposed to fusion of data and control practically and hence in this regard we thank our project staff Smt. B Sujatha Kumari and Mr. T Srikanth of Department of Electronics and Communication, SJCE to have given us this crucial opportunity to work on the mini project.

ABSTARCT:

The project concept is that the turning of the pages by using the voice commands like ‘next’ and ‘previous’ to turn the respective next and previous pages. The turning of pages is done with the usage of motors which creates an air gap and an arm like robot which will turn the pages according to the voice commands given by the reader.

List of figures

Chapter 1

INTRODUCTION

1.1 Motivation:

The physically disabled people cannot turn the pages so seeing their difficulties we came up with an idea to turn the pages automatically using voice commands so that it can be of help to them to read the books without anyone's help.

1.2 Problem Definition:

To design a voice controlled page rotation system. This is implemented using the microcontroller and dc motors which rotate the pages. The microcontroller is programmed to rotate then pages.

1.3 Literature Survey:

1. IEEE paper abstract by Gunnar Bolmsjö

Robotics in rehabilitation provide considerable opportunities to improve the quality of life for physically disabled people. However, practical results limited, mainly due to the need for developing new robotics concepts where people are where they are working in separated areas. This paper reveals some of the developments needed and presents two projects currently underway at Lund University. The first one is concerned with end-effector design for a robotic workstation for office based tasks, while the second is concerned with a mobile robotic system for use in medical an chemical laboratories by disabled people. Both projects show promising results. There is also a need for further research in developing new robotic systems for use in rehabilitation with new mechanical .

2. The project objective is to build one system that can recognise human voice and then activate the electric appliance at home . This system called speech operated system for home appliances; It is software designed using Matlab. System is designed to help people with disabilities to do their work at home . Two main parts in this system is voice train process and voice recognition process . Main objective of the process is to design a GUI and source code that can recognise human voice . This system used speaker independent method ; any user who have traines their voice into the system can use the system . This is because system is built to recognise speakers speech not his identity .

By HairailAmri Bin Jaffer
University Tenikal Malaysia

1.4 Block diagram:

Fig 1. Shows the simple block diagram of the page turner which consists of the voice processor which takes voice commands as input signal and process the command, the processed signal is given to the micro controller unit as shown, which is programmed in such way that according to the voice commands the motors and arms should be run in the manner that is needed for the tuning of the pages.

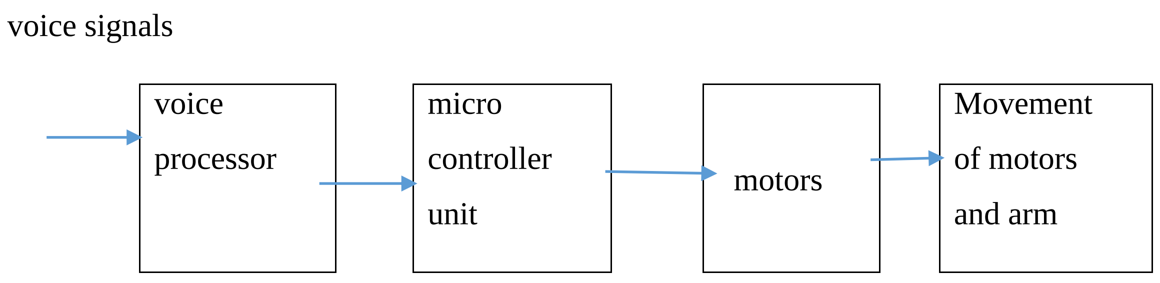

Fig 1: Block diagram of simple page turner using voice commands

Chapter 2

MICROCONTROLLER

2.1 Application

The microcontroller is used to drive the DC motors which in turn rotate the arms in order to turn the page.

2.2 Atmega 16

The ATmega16 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega16 achieves throughputs approaching 1 MIPS per MHz allowing the system designed to optimize power consumption versus processing speed.

2.3Features:

High-performance, Low-power AVR® 8- bit Microcontroller

- Advanced RISC Architecture
  - 131 Powerful Instructions – Most Single-clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 16 MIPS Throughput at 16 MHz
  - On-chip 2-cycle Multiplier

- High Endurance Non-volatile Memory segments
  - 16K Bytes of In-System Self-programmable Flash program memory
  - 512 Bytes EEPROM
  - 1 K Byte Internal SRAM
  - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - Data retention: 20 years at 85°C/100 years at 25°C
  - Optional Boot Code Section with Independent Lock Bits In-System Programming by

  On-chip Boot Program

  True Read-While-Write Operation

- – Programming Lock for Software Security

- JTAG (IEEE std. 1149.1 Compliant) Interface
  - Boundary-scan Capabilities According to the JTAG Standard
  - Extensive On-chip Debug Support
  - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescalars and Compare Modes
  - One 16-bit Timer/Counter with Separate Prescalar, Compare Mode, and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Four PWM Channels
  - 8- channel, 10-bit ADC

    8 Single-ended Channels

    7 Differential Channels in TQFP Package Only

    2 Differential Channels with Programmable Gain at 1x, 10x, or 200x
  - Byte-oriented Two-wire Serial Interface
  - Programmable Serial USART
  - Master/Slave SPI Serial Interface
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated RC Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby
- I/O and Packages
  - 32 Programmable I/O Lines
  - 40-pin PDIP, 44-lead TQFP, and 44-pad QFN/MLF
- Operating Voltages

- 2.7 - 5.5V for ATmega16L
- 4.5 - 5.5V for ATmega 16

• Speed Grades
- 0 - 8 MHz for ATmega16L
- 0 - 16 MHz for ATmega 16

• Power Consumption @ 1 MHz, 3V, and 25·C for ATmega16L
- Active: 1.1 mA
- Idle Mode: 0.35 mA
- Power-down Mode: < 1 µA

2.4 Pin diagram:

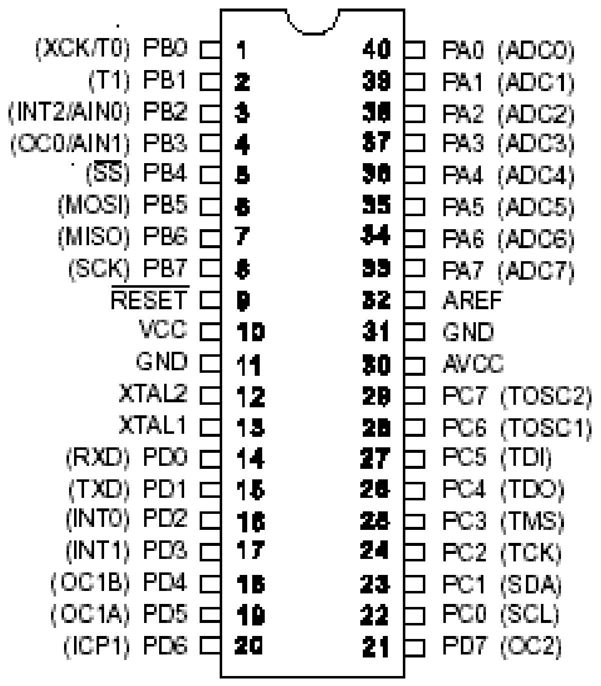

Fig 2: ATMega16 pin diagram

Chapter 3

APPROACH AND ANALYSIS

3.1 MFCC

In sound processing, the mel-frequency cepstrum (MFC) is a representation of the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency.

Mel-frequency cepstral coefficients (MFCCs) are coefficients that collectively make up an MFC. They are derived from a type of cepstral representation of the audio clip (a nonlinear "spectrum-of-a-spectrum"). The difference between the cepstrum and the mel-frequency cepstrum is that in the MFC, the frequency bands are equally spaced on themel scale, which approximates the human auditory system's response more closely than the linearly-spaced frequency bands used in the normal cepstrum. This frequency warping can allow for better representation of sound, for example, in audio compression.

MFCCs are commonly derived as follows:

1. Take the Fourier transform of (a windowed excerpt of) a signal.
2. Map the powers of the spectrum obtained above onto the mel scale, using triangular overlapping windows.
3. Take the logs of the powers at each of the mel frequencies.
4. Take the discrete cosine transform of the list of mel log powers, as if it were a signal.
5. The MFCCs are the amplitudes of the resulting spectrum.

3.2 Vector Quantization

Vector quantization (VQ) is a classical quantization technique from signal processing which allows the modeling of probability density functions by the distribution of prototype vectors. It was originally used for data compression. It works by dividing a large set of points (vectors) into groups having approximately the same number of points closest to them. Each group is represented by its centroid point, as in k-means and someOther clustering algorithms.

CHAPTER 4

SOFTWARE AND HARDWARE USED

4.1 AVR Studio

AVR Studio was created by Atmel in order to help developers to create applications for AVR microcontrollers using C/C++ programming languages. This piece of software comes with a large number of tutorials, which allow the users to get familiar with the application. The program stands as a complete pack for programmers that use C++ and other programming languages.

4.2 Khazama

We have used this software to burn the c programme into the microcontroller.

4.3 Matlab

MATLAB is a high-level language and interactive environment for numerical computation, visualization, and programming. Using MATLAB, you can analyze data, develop algorithms, and create models and applications. The language, tools, and built-in math functions enable you to explore multiple approaches and reach a solution faster than with spreadsheets or traditional programming languages, such as C/C++ or Java.

4.4 Motor Driver IC

L293D is a dual H-bridge motor driver integrated circuit (IC). Motor drivers act as current amplifiers since they take a low-current control signal and provide a higher-current signal. This higher current signal is used to drive the motors.

L293D contains two inbuilt H-bridge driver circuits. In its common mode of operation, two DC motors can be driven simultaneously, both in forward and reverse direction. The motor operations of two motors can be controlled by input logic at pins 2 & 7 and 10 & 15. Input logic 00 or 11 will stop the corresponding motor. Logic 01 and 10 will rotate it in clockwise and anticlockwise directions, respectively.

Enable pins 1 and 9 (corresponding to the two motors) must be high for motors to start operating. When an enable input is high, the associated driver gets enabled. As a result, the outputs become active and work in phase with their inputs. Similarly, when the enable input is low, that driver is disabled, and their outputs are off and in the high-impedance state.

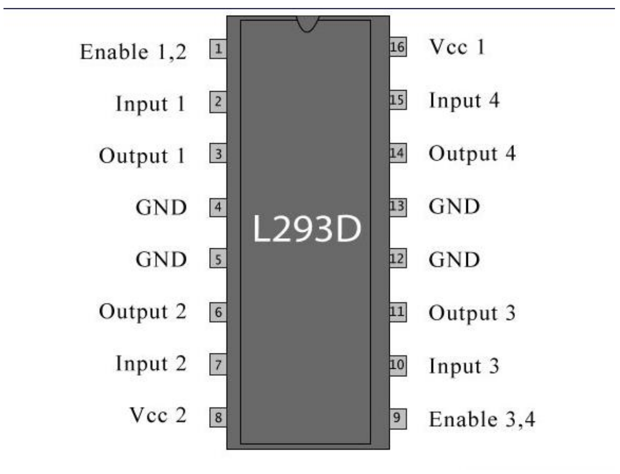

Fig 3: Pin Diagram Of L293D

4.5 MAX232

The MAX232 IC is used to convert the TTL/CMOS logic levels to RS232 logic levels during serial communication of microcontrollers with PC. The controller operates at TTL logic level (0-5V) whereas the serial communication in PC works on RS232 standards (-25 V to + 25V). This makes it difficult to establish a direct link between them to communicate with each other.

The intermediate link is provided through MAX232. It is a dual driver/receiver that includes a capacitive voltage generator to supply RS232 voltage levels from a single 5V supply. Each receiver converts RS232 inputs to 5V TTL/CMOS levels. These receivers (R_1 & R_2) can accept ±30V inputs. The drivers (T_1 & T_2), also called transmitters, convert the TTL/CMOS input level into RS232 level.

The transmitters take input from controller’s serial transmission pin and send the output to RS232’s receiver. The receivers, on the other hand, take input from transmission pin of RS232 serial port and give serial output to microcontroller’s receiver pin. MAX232 needs four external capacitors whose value ranges from 1µF to 22µF.

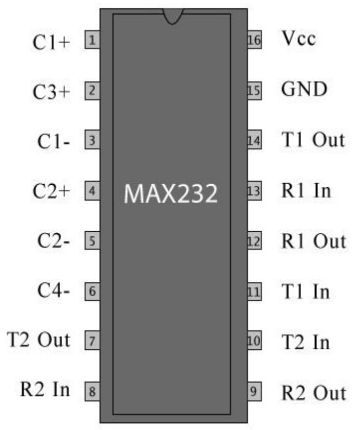

Fig 4: Pin Diagram Of MAX232

Chapter 5

DESIGN & IMPLEMENTATION

5.1 Circuit Diagrams:

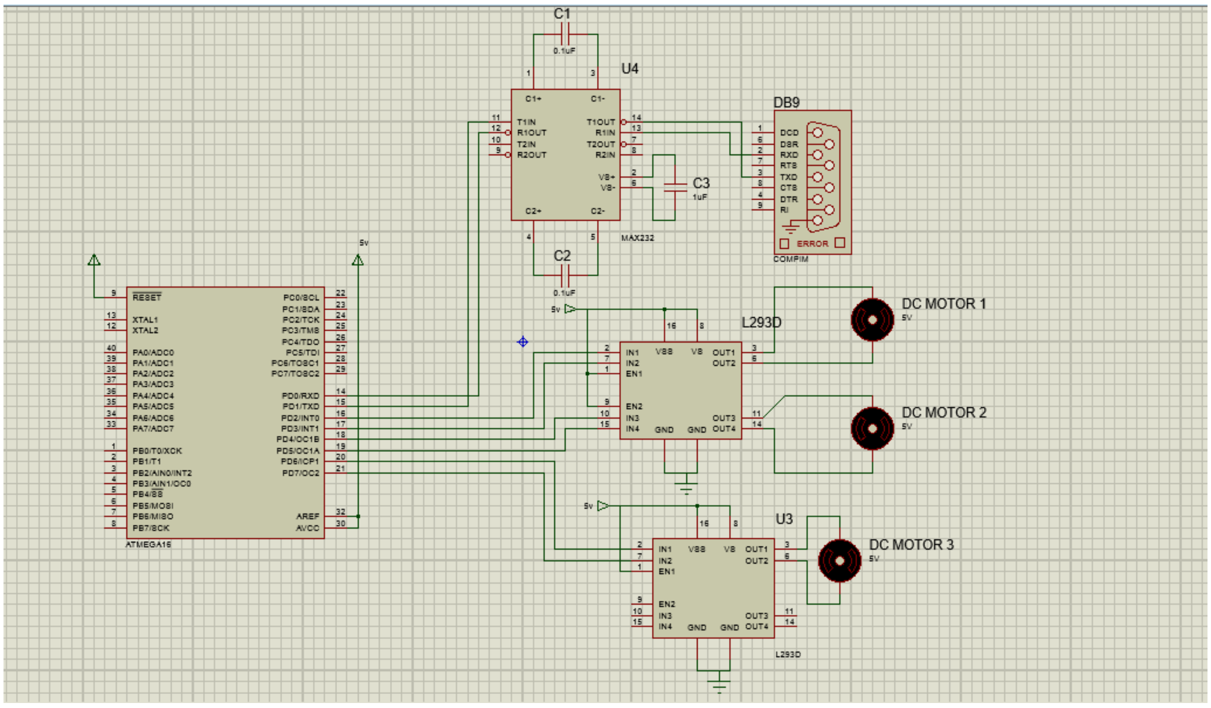

Fig5: Interfacing DC motors and serial communication(DB9) to ATMEGA16

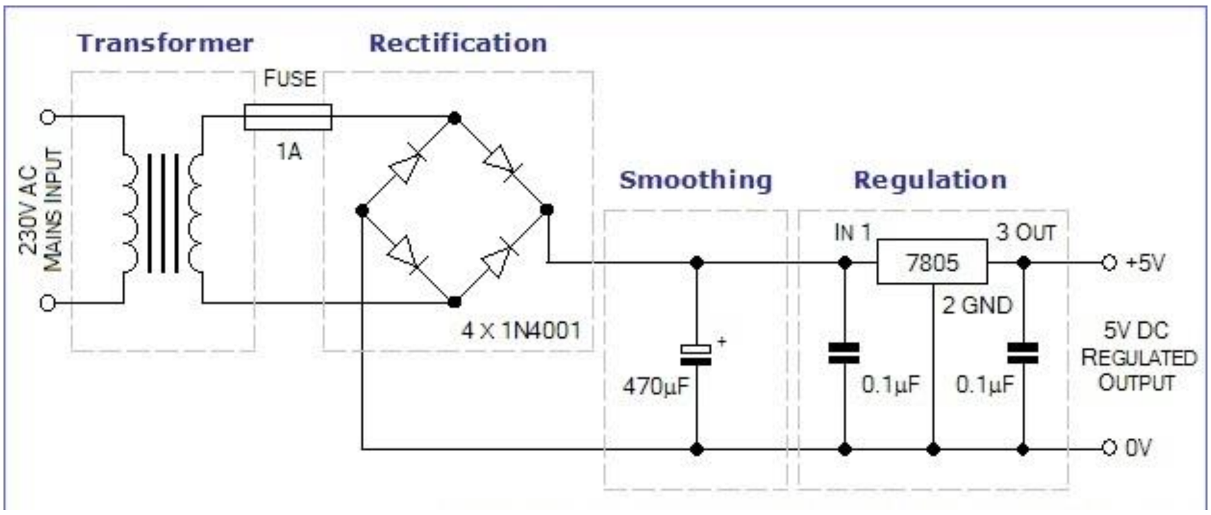

Fig 6: 5v Power supply

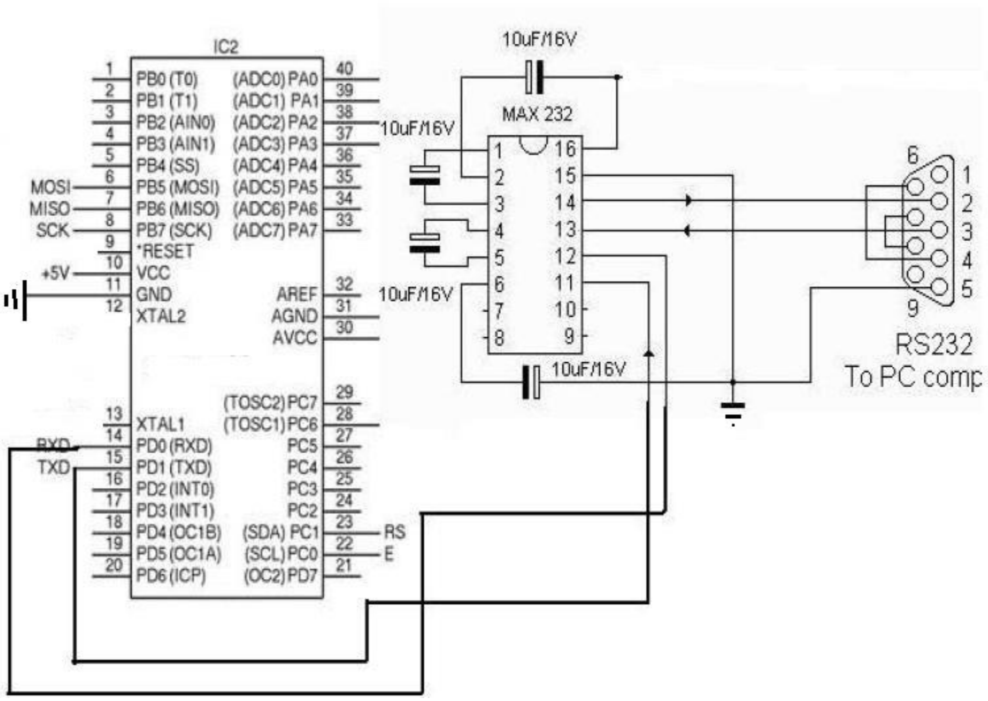

Fig 7: Interfacing db-9 connector to ATmega16 via Max 232

5.2 Procedure:

Voice samples were collected from different people for the words 'Next' and 'Previous' . For each person about 10 samples were collected using a microphone in a noiseless environment using Praat software . The samples were recorded at a particular sampling frequency . The samples were made noise free by selecting the required portions of the sample and stored .

A matlab code(appendix) has been written in order to extract the MFCC from the noiseless samples of all the persons . The program creates the database of all the samples and extracts mfcc from each of the and forms a reference vector by vector quantization . The sample to be recognised is loaded in the program . It decodes the sample and provides different code for different samples . It produces 1 code for next and a different code for previous . Thos code produced will be sent to the microcontroller via the serial communication . Code for serial communication is in appendix .

The microcontroller receives the code from the matlab via serial communication . It then proceeds depending on the code received . If next is received it will turn to the next page . If previous is received it will turn back to the previous page . The code for this is in appendix .

The arms which are controlled by the DC motors which are driven by microcontroller depending on word .

CHAPTER 6

SUMMARY

6.1 IMPLEMENTATION RESULTS:

The circuit was switched on. The required word (previous or next) is selected from the database and loaded into the program. The matlab differentiates between the words and performs the action of turning the page depending on the word loaded into the program.

6.2 ADVANTAGES

Pages of the book can be turned without using hands. This can be used by physically challenged people.

6.3 DISADVANTAGES:

1. Roller material's should be selected such that good friction must exist between roller and pages or else the page may not turn or many pages will be turned at once.
2. If pages are too delicate they may be damaged.
3. Voice recognition must be trained to one's particular voice.

6.4 APPLICATIONS:

This system can be used by physically disabled people to read books. May be used by lazy people also.

6.5 FUTURE SCOPE:

1. The speech recognition system can be implemented in real mode.
2. We can scan the text on the page and read it out for the blind people.

6.6 Cost of the project:

Commodity	Quantity	Cost(Rs)
ATMega16	1	200
Roller and Arms	1	300
DC Motors	3	300
Db-9 connector	1	200
Dc motor driver IC’s	2	100
Power supply circuit	1	200
Others		200
TOTAL $\longrightarrow$		1500

APPENDIX:

Programs:

1. **Matlab code for speech recognition**

```
clc;
clear all;
close all;
% LOAD THE TEST SIGNAL
test = wavread('C:\Users\brijesh2693\Desktop\speech samples\test');
i=1;
D = struct ;
%---------------------- DATABASE LOADING --------------------------------
%-------------------------next------------------------------------------
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\bnext1');
D(i).B=D(i).B(:,1);
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\bnext2');
D(i).B=D(i).B(:,1);
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\bnext3');
D(i).B=D(i).B(:,1);
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\bnext4');
D(i).B=D(i).B(:,1);
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\bnext5');
D(i).B=D(i).B(:,1);
i=i+1;

D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\bnext6');
D(i).B=D(i).B(:,1);
i=i+1;
```

```
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\bnext7');
D(i).B=D(i).B(:,1);
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\bnext8');
D(i).B=D(i).B(:,1);
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\bnext9');
D(i).B=D(i).B(:,1);
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\bnext10');
D(i).B=D(i).B(:,1);
i=i+1;
 D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\dnext1');
D(i).B=D(i).B(:,1);
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\dnext2');
D(i).B=D(i).B(:,1);
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\dnext3');
D(i).B=D(i).B(:,1);
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\dnext4');
D(i).B=D(i).B(:,1);
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\dnext5');
D(i).B=D(i).B(:,1);
i=i+1;

D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\dnext6');
D(i).B=D(i).B(:,1);
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\dnext7');
```

```
D(i).B=D(i).B(:,1);
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\dnext8');
D(i).B=D(i).B(:,1);
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\dnext9');
D(i).B=D(i).B(:,1);
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\dnext10');
D(i).B=D(i).B(:,1);
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\naganext1');
D(i).B=D(i).B(:,1);
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\naganext2');
D(i).B=D(i).B(:,1);
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\naganext3');
D(i).B=D(i).B(:,1);
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\naganext4');
D(i).B=D(i).B(:,1);
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\naganext5');
D(i).B=D(i).B(:,1);
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\naganext6');
D(i).B=D(i).B(:,1);
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\naganext7');
D(i).B=D(i).B(:,1);
i=i+1;
```

```
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\naganext8');
D(i).B=D(i).B(:,1);
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\naganext9');
D(i).B=D(i).B(:,1);
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\naganext10');
D(i).B=D(i).B(:,1);
i=i+1;
        %----------previous-----------------------
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\bprev1');
D(i).B=D(i).B(:,1);
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\bprev2');
D(i).B=D(i).B(:,1);
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\bprev3');
D(i).B=D(i).B(:,1);
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\bprev4');
D(i).B=D(i).B(:,1);
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\bprev5');
D(i).B=D(i).B(:,1);
i=i+1;


D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\bprev6');
D(i).B=D(i).B(:,1);
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\bprev7');
D(i).B=D(i).B(:,1);
```

```
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\bprev8');
D(i).B=D(i).B(:,1);
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\bprev9');
D(i).B=D(i).B(:,1);
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\bprev10');
D(i).B=D(i).B(:,1);
i=i+1;



D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\dprev1');
D(i).B=D(i).B(:,1);
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\dprev2');
D(i).B=D(i).B(:,1);
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\dprev3');
D(i).B=D(i).B(:,1);
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\dprev4');
D(i).B=D(i).B(:,1);
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\dprev5');
D(i).B=D(i).B(:,1);
i=i+1;

D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\dprev6');
D(i).B=D(i).B(:,1);
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\dprev7');
```

```
D(i).B=D(i).B(:,1);
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\dprev8');
D(i).B=D(i).B(:,1);
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\dprev9');
D(i).B=D(i).B(:,1);
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\dprev10');
D(i).B=D(i).B(:,1);
i=i+1;

D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\nagaprev1');
D(i).B=D(i).B(:,1);
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\nagaprev2');
D(i).B=D(i).B(:,1);
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\nagaprev3');
D(i).B=D(i).B(:,1);
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\nagaprev4');
D(i).B=D(i).B(:,1);
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\nagaprev5');
D(i).B=D(i).B(:,1);
i=i+1;

D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\nagaprev6');
D(i).B=D(i).B(:,1);
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\nagaprev7');
```

```
D(i).B=D(i).B(:,1);
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\nagaprev8');
D(i).B=D(i).B(:,1);
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\nagaprev9');
D(i).B=D(i).B(:,1);
i=i+1;
D(i).B=wavread('C:\Users\brijesh2693\Desktop\noisless\nagaprev10');
D(i).B=D(i).B(:,1);
i=i+1;
disp('Program Running......................');
disp('Please wait for few seconds..........');
len = i-1;
for j=1:len
    xx=D(j).B;
    xx1=test;
   xx=vad(xx);
   xx1=vad(xx1);
    [x,x1]=sampeq(xx,xx1);      % sampling rate equalizing
    [cax,cdx]=dwt(x,'db4');
    [cax1,cdx1]=dwt(x1,'db4');
    mm=MFCC(cax);             % feature extraction
    mm1=MFCC(cax1);
    v=vqlbg(mm,1);            % vector quantization
    v1=vqlbg(mm1,1);
    ed=disteu(v,v1);          % euclidean distance
    edarr(j)=ed;
       end
[edmin,pos]=min(edarr);
if edmin < 100
      disp('The uttered word is:');
```

```
     wavplay(D(44).B,44100);
     subplot(2,1,1);
     plot(D(pos).B);

    if pos>=1 && pos<=30
       disp('next');
       wavplay(test,44100);
       disp(pos);
       subplot(2,1,2);
        plot(test);

s=serial('COM5','BaudRate',9600);
   fopen(s);
   fprintf(s,'a');
   delete(s);

    elseif pos>=31 && pos<=60
       disp('previous');
        wavplay(test,44100);
        disp(pos);
        subplot(2,1,2);
        plot(test);
```

2. MFCC function definition

```
function ccc = MFCC(x,Fs,nM,nC,nfft,fl,fh,ovlp)
 if nargin < 2   Fs = 8000;       end
if nargin < 3   nM = 12;    end
if nargin < 4    nC = 24;          end
if nargin < 5    nfft = 256;       end
if nargin < 6    fl = 0;     end
if nargin < 7    fh = 0.5;          end
if nargin < 8    ovlp = 0.5;       end
x = x(:);
bank=melbankm(nC,nfft,Fs,fl,fh);
%figure;plot(bank);title('Melbank');
bank = full(bank);
bank = bank / max(bank(:));
%figure;plot(bank);title('Melbank');
j = 0:nC-1;
for k = 1:nM
    dctcoef(k,:) = cos(k * (j+0.5) * pi / nC);
end



w = 1 + (nM/2) * sin(pi*[1:nM]./nM);
w = w / max(w);
% pre-emphasis filtering
% x = double(x);
% x = filter([1 -0.95],1,x);
 % ----- Calculate MFCC coefficients of each frame -----
L_x = length(x);
nOvlp = floor(nfft*ovlp);
i = 0;
while ((i*nOvlp+nfft) <= L_x)
    in = x(i*nOvlp+1:i*nOvlp+nfft);
    in = in ./ max(abs(in));
    in  = in .* hamming(nfft);

    % --- Calculate the energy spectrum ---
    s = abs(fft(in));
    t  = s.^2;
    t  = t+2*realmin;
    t = t(1:nfft/2+1);
    % --- Calculate the energy in each channel ---
    t1 = bank * t;

    % --- Calculate MFCC ---
    t1 = log10(t1);
    c1 = dctcoef * t1;
    c2 = c1.*w';
    m(i+1,:) = c2';

     i = i + 1;
end

ccc = m;
       s=serial('COM5','BaudRate',9600);
       fopen(s);
       fprintf(s,'b');
       delete(s);
  end
else
    disp('No match found');
end
```

3. Disteu function definition

```
function d = disteu(x, y)
[M, N] = size(x);
[M2, P] = size(y);

if (M ~= M2)
    error('Matrix dimensions do not match.')
end
```

```
d = zeros(N, P);

if (N < P)
    copies = zeros(1,P);
    for n = 1:N
        d(n,:) = sum((x(:, n+copies) - y) .^2, 1);
    end
else
    copies = zeros(1,N);
    for p = 1:P
        d(:,p) = sum((x - y(:, p+copies)) .^2, 1)';
    end
end
d = d.^0.5;
```

4. Serial communication

```
clc;
s=serial('COM9','BaudRate',9600);
fopen(s);
fprintf(s,'b');
delete(s);
```

5. Microcontroller code

```
#include <avr/io.h>
#include<util/delay.h>
#define USART_BAUDRATE 9600
#define BAUD_PRESCALE (((16000000 / (USART_BAUDRATE * 16UL))) - 1)
int main (void)
{
   char cv,pv;
   pv=0b01100001;

DDRD=0b11111111;
UCSRB |= (1 << RXEN) | (1 << TXEN);
UCSRC |= (1 << URSEL) | (1 << UCSZ0) | (1 << UCSZ1);
UBRRL = BAUD_PRESCALE;
UBRRH = (BAUD_PRESCALE >> 8);
for (;;)
   {
      while ((UCSRA & (1 << RXC)) == 0) {};
cv = UDR;
              if(cv==pv)
              {
      if(cv==(0b01100001))  //character 'a'
      {
        PORTD=0x10;
              _delay_ms(30000);
      PORTD=0x00;
      _delay_ms(6000);

               PORTD=0x40;
                         _delay_ms(6000);
                        PORTD=0x40;
                         _delay_ms(3000);
        PORTD=0x80;
                        PORTD=0x00;

      }



else if(cv==(0b01100010)) // char 'b'
```

```
     {
       PORTD=0x20;
                        _delay_ms(30000);
                       PORTD=0x00;
                        _delay_ms(6000);



                       PORTD=0x80;
                        _delay_ms(6000);
                       PORTD=0x80;
                        _delay_ms(6000);
                       PORTD=0x00;;
}

   }

            else
            {

            if(cv==(0b01100001))  //character 'a'
     {
        PORTD=0x08;
                        _delay_ms(180000);
                        PORTD=0x08;
                 _delay_ms(180000);
                 PORTD=0x08;
                 _delay_ms(180000);
                 PORTD=0x08;
                 _delay_ms(180000);
                         PORTD=0x08;
                  _delay_ms(180000);
                  PORTD=0x08;
```

```
                _delay_ms(180000);
                      PORTD=0x00;
        _delay_ms(6000);

                      PORTD=0x10;
        _delay_ms(30000);;
                      PORTD=0x00;
        _delay_ms(6000);

                      PORTD=0x40;
                      _delay_ms(6000);
                     PORTD=0x40;
                      _delay_ms(6000);
                      PORTD=0x80;
                     PORTD=0x00;
          }

else if(cv==(0b01100010)) // char 'b'

 {
      PORTD=0x04;
             _delay_ms(180000);
             PORTD=0x04;
             _delay_ms(180000);
             PORTD=0x04;
             _delay_ms(180000);
             PORTD=0x04;
             _delay_ms(180000);
              PORTD=0x04;
             _delay_ms(180000);
              PORTD=0x04;
             _delay_ms(180000);
```

```
                PORTD=0x00;
        _delay_ms(6000);

                PORTD=0x20;
                          _delay_ms(30000);
                         PORTD=0x00;
                          _delay_ms(6000);

                         PORTD=0x80;
                          _delay_ms(6000);
                         PORTD=0x80;
                          _delay_ms(6000);
         PORTD=0x40;
                         PORTD=0x00;
    }
              }
              pv=cv;
               }
}
```

REFERENCES:

1. www.engineersgarage.com/
2. www.mathworks.in/
3. www.electroons.com