MULTI-PURPOSE VOICE ACTIVATED CONTROLLER

By

LEE WAN NEE

FINAL PROJECT REPORT

Submitted to the Electrical & Electronics Engineering Program
in Partial Fulfillment of the Requirements
for the Degree
Bachelor of Engineering (HONS)
(Electrical & Electronics Engineering)

Universiti Teknologi Petronas
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

MULTI-PURPOSE VOICE ACTIVATED CONTROLLER

by

Lee Wan Nee

A project dissertation submitted to the
Electrical & Electronics Engineering Program
Universiti Teknologi PETRONAS
in partial fulfillment of the requirement for the
Bachelor of Engineering (HONS)
(Electrical & Electronics Engineering)

Approved:

Dr. Mumtaj Begam
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK
June 2007

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

LEE WAN NEE

ABSTRACT

This project, a multipurpose voice-activated controller is a device that can be activated using human voice instead of the conventional switch. Devices which can be voice activated brings more convenience and allows a certain safety measure as it allows user to not be in physical contact with the appliance. This project is divided mainly into the hardware and software part. The software part is implemented first. Software part includes MATLAB. MATLAB programming receives and processes audio command from microphone and is used to implement the speech recognition and sending output to trigger the microcontroller. PCB development involves use of EAGLE software. The hardware part involves the HM2007 IC utilized in the speech recognition circuit as well as an interfacing circuit to control the external appliances. This project will be implemented on table lamp and doorbell.

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my utmost gratitude to my supervisor Dr. Mumtaj Begam for her guidance and supervision throughout my Final Year Project. I would also like to thank UTP technicians who've greatly aided and tipped us in circuit building and components search. Lastly, a special appreciation to all lecturers and friends who've in one way or another aided me in completing my Final Year Project.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 Background Study

Voice recognition is beginning to bring impact on our technologies nowadays to further simplify our daily routines. It is a very convenient technology that is becoming common with all appliances now. This project, multipurpose voice-activated controller are devices that can be activated using human voice instead of the conventional switch. This technology is now widely used and HomeVoice is the latest technology that allows the control of home appliances using voice recognition.

In this project, the focus will be on voice activated doorbell and voice activated light. The doorbell should be able to be activated by voice and then ring (just as how it would when a button is pushed) to notify the person inside the house or office to attend to the door. The light is also the same, except it has to be able to recognize two verbal commands that will be able to activate it, to turn ON and another to deactivate it to turn OFF.

Of late, speech recognition is a very popular technology in research lately. However, to date, there is still no perfect speech recognition application, thus the evolvement of this technology continues on to improve it from discrete dictation mode, where a pause is needed between spoken words till continuous dictation which is yet to be perfectly achieved today.

Speech recognition is also divided into speaker dependent and speaker independent. Speaker dependent recognition will recognize the pitch of each individual, while speaker independent recognition only recognizes the words spoken irrespective of who spoke it, also known as word speech recognition.

While speaker dependent speech recognition is currently more wide-spread, my project will not be utilizing it. This project is utilizing speaker independent speech recognition which is still in the research stage and not very established yet as it has error rates.

Microprocessor approach to this objective has proven to be more reliable and performs more consistently.

HM2007 is quite a recognized IC in the speech recognition field. It however does not have the high performance to be able to be utilized for security purposes, but is relatively reliable enough for simple controls of common appliances.

Voice activated controllers can be applied to many common appliances such as air condition, doorbell, table lamp, television, ceiling fan, radio and many more. In fact, it can be utilized in future smart home or in a vehicle. This will allow physically disadvantaged persons; such has the handicapped or those with height disadvantage to no longer face the problem of not able to reach the switch. Moreover, applying this to a radio / air condition / window controller and many others switches within a vehicle allow the driver to have both his hands and concentration dedicated fully to his driving.

1.2 Problem Statement

To see a clutter of switches in one corner of a house is definitely not an unfamiliar sight, especially with the increment of luxury device that are becoming more available and affordable these days. This will either become a sore sight when they are all concentrated in a certain corner, or it will become of tedious inconvenience to turn on and off all these appliances if they are spread out all over the corners. Either way, it brings us to one conclusion, discomfort and inconvenience.

Besides, the advancement in technology has considerably increased workload and burdens to one's daily life. Schedules are usually very hectic as well. It is more often than not that one usually leaves their house in a hurry, forgetting to turn off at least one or two switches that are out of sight, and out of reach. This brings two disadvantages; firstly it is a waste of electric bill and secondly it may be hazardous to leave some appliances on for long hours.

Most importantly, there are some group of people who are physically impaired or may have height disadvantage to reach for the controllers or switches of these devices. For example, a person on a wheelchair may not be able to reach the switch for a light or the button for the doorbell as they are usually fabricated at a human's standing height.

With voice activated implemented for lights, a considerable number of switches can be reduced, saving a sore sight. Voice activated lights / doorbells also allow the minority but existing disadvantaged group to 'reach' these devices and activate them without difficulty. For others, it may as well mean more convenience and time saving as they do not need to move to the switch but merely project their voice within the sensor's range. Also, since voice activated controllers does not require physical contact, it keeps user safe if there happens to current leakage.

The problem currently encountered with this project is the MATLAB implementation of the voice recognition. The function is limited, and needs to be train with the words it was to recognize. It will recognize the spoken word, disregard of weather or not it really existed in the database as it is using the dynamic time warping technique and measures the shortest distance to be the most accurate word spoken and then recognized from the database.

Since the last of MATLAB programming, I've been working on microprocessor approach instead. Selecting HM2007 as my IC, HM2007 proves to be more reliable to a certain extent, yet at the same time has a certain percentage of error as well. In order to overcome this, more samples of speech must be collected to widen the variety of how a speech is spoken can be recognized by the circuit.

1.3 Objective and Scope of Study

This project aims to implement a voice activated doorbell and light for the convenience of all. However, both these devices are limited to only sense a distance of 2 meters to avoid capturing unintended verbal commands.

Voice recognition used here has to be speaker independent and will be able to filter out background noise to capture and recognize the audio command given.

The scope of study for the voice recognition covers on how audio signal is processed using the Dynamic Time Warping technique. In depth discussion will be covered in Literature review and Methodology.

The later part of project also implements HM2007 IC into speech recognition circuit. The study of HM 2007 workability is further discussed in the literature review and results.

Studies are also made on how to construct the interfacing circuit from the output of HM2007 speech recognition circuit and then to the external circuit (the voice controlled appliance).

CHAPTER 2
LITERATURE REVIEW & THEORY

This chapter will cover on the background study of speech recognition technology which will be widely utilized in this project. In the software development, MATLAB code is used, the method utilized is the Dynamic Time warping. Meanwhile, the microprocessor approach will be utilizing the MN2007 IC.

2.1 Speech Recognition

In this project, speech recognition is the main concern and can be divided into speaker dependent and speaker independent. Speech recognition is a complicated process that undergoes many sub-processing before the computer can actually recognize what is spoken of human voice.

Speaker dependent recognition will recognize the unique pitch of each individual's voice, while speaker independent recognition recognizes the words spoken irrespective of who spoke it, also known as word speech recognition.

The basic steps in speech recognition are:

1. Determining the phonemes that are spoken.
2. Convert the phonemes into words.
3. Transform the Pulse Code Modulation digital audio [1]

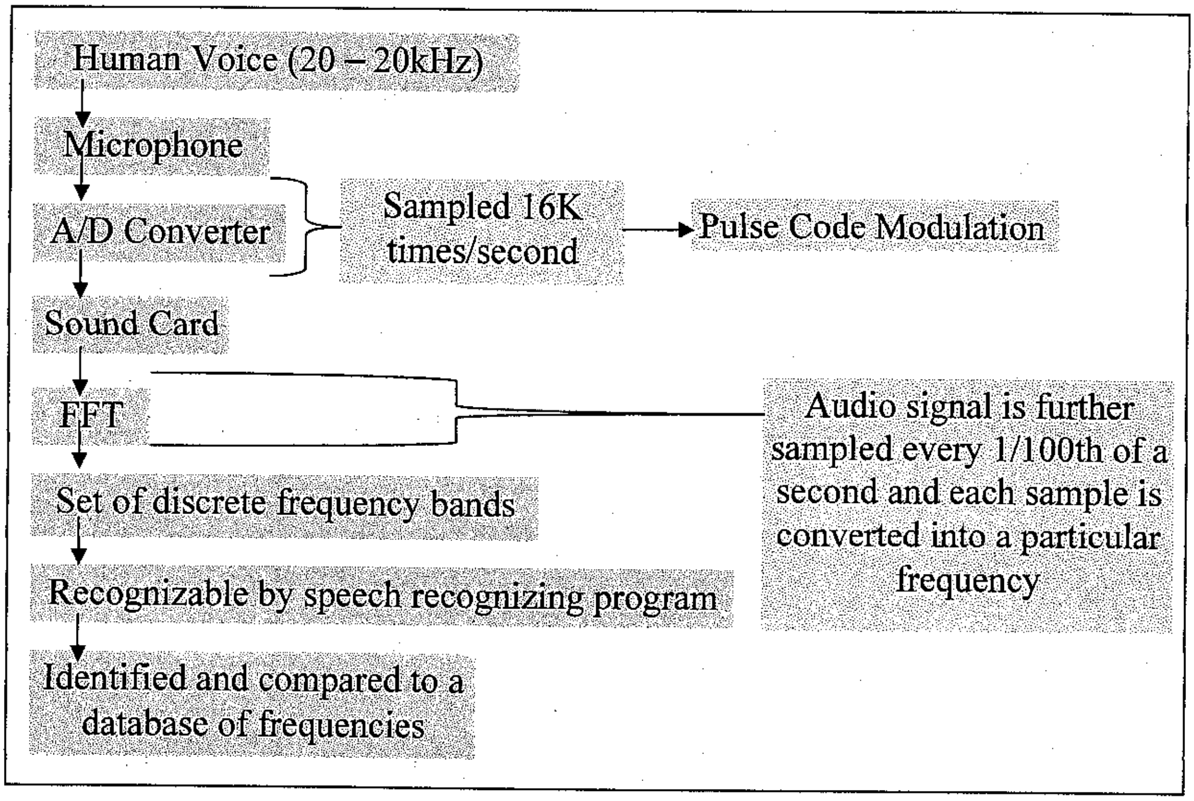

Figure 1: Flow chart of Voice recognition system

As illustrated in Figure 1, the raw input (human voice) is initially fed to the system through a receiver. In this case, a microphone is used as the receiver. The human voice then goes to the sound card. Human voice is of analog signal. Therefore, it is necessary to be converted by the analog to digital converter into the form that the computer can comprehend – a stream of digital data (digital audio). To achieve this, it is firstly sampled 16,000 times a second. Secondly, the PCM (Pulse Code Modulation) is utilized to convert it into digital waveform. This waveform is then translated into a set of discrete frequency bands using the technique Windowed Fast Fourier Transform (FFT). In Windowed Fast Fourier Transform (FFT), the audio signal is further sampled every 1/100th of a second. Later, each sample is converted into a particular frequency. The input stream is now a set of discrete frequency bands. This set of discrete frequency bands is the form which can finally be processed by speech recognizing program.

Next, these bands of frequencies will have to be identified and compared to a database of frequencies that are actually 'phonemes' for a match according to their audio frequency bands. Phoneme is the smallest unit of speech. Every phoneme has a feature number that can be assigned to the compatible input stream. These feature numbers will be matched to the set of frequency bands converted from the raw input earlier on.

However, due to the variation of pronunciation, even for a single phoneme, the speech recognition program needs to be 'trained' to recognize these variations through probability and statistics. As the stream is sampled at 1/100th a second, the duration of a phoneme is therefore lengthy enough to have many frequency bands passing through the speech recognizer where they are respectively assigned the appropriate feature number. Statistical method is used to analyze probability of the feature numbers assigned to that phoneme, where the highest probability feature number will be chosen to match the phoneme fed in the system.

To enable the speech recognizer to realize the beginning and the ending of a certain phoneme, the HMM (Hidden Markov Model) – a mathematical model that utilizes statistics, technique is used. Here, silent phonemes assigned with feature numbers as used. Some phonemes depend on their precedent as well. For this case, tri-phones are used. Meanwhile, pruning, where the software generates hypotheses of what might have been spoken, is another alternative. Scored of each hypothesis is generated and the highest scored one is chosen, while others are 'pruned' out. [1]

2.2 Dynamic Time Warping

Speech is a time-dependent process. The same word spoken in the same duration will still differ in the middle due to the different segments – the phonemes, of the words uttered in different rates. Thus, time alignment is performed to obtain the global distance between two speech patterns, which is represented as a sequence of vectors. This methodology is known as Dynamic Programming. When template is used for this speech recognition, it is known as Dynamic Time Warping (DTW). DP finds the lowest distance path through the matrix but minimize the computation as well. DP algorithm operates in a time synchronous manner. [2]

Each column of the time-time matrix is considered in succession (processing input frame by frame), so that only a maximum N number of paths considered for a certain N-length template. [2]

This method converts test data into templates and then matches them with the stored templates. The template with the lowest distance measured from the input pattern is recognized as the word. The best match is based upon dynamic programming, known as the Dynamic Time Warping (DTW) word recognizer.

In DTW:

Features – Information in each signal needs to be represented in some manner

Distances – some form of metric is used in order to obtain a match path.

Euclidean Distance Metric

Euclidean distance metric is used to measure the distance between two feature vectors. [2]

$$d(x,y) = \sqrt{\sum_j (x_j - y_j)^2}$$

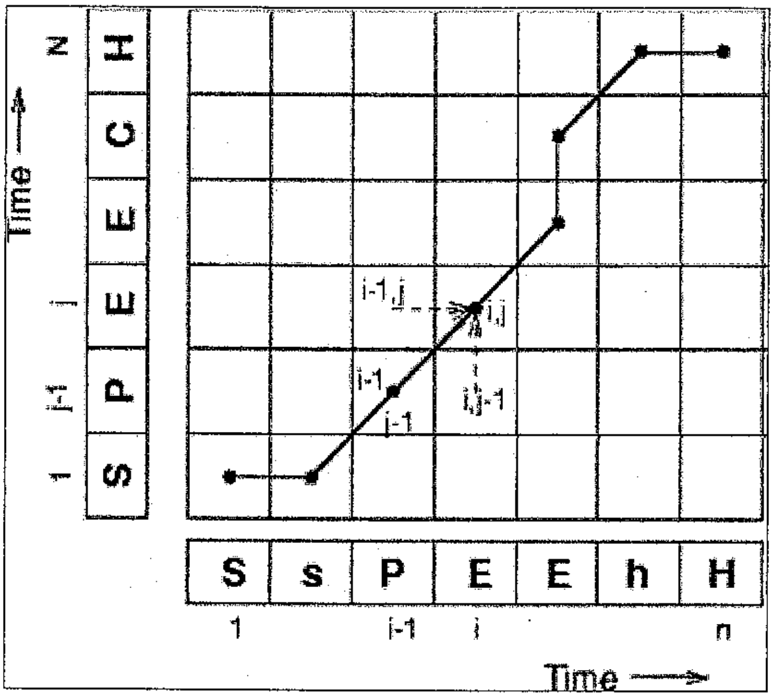

Figure 2: Dynamic Time Warping [2]

Figure 2 illustrates the implementation of time alignment between the test and the training pattern.

$$d(x,y) = \sqrt{\sum_i (x_i - y_i)^2}$$

The formula above measures the distance between a point X (X1, X2, etc.) and a point Y (Y1, Y2, etc.) just as illustrated in Figure 2. Euclidean metric is the "ordinary" distance between two points that one would measure with a ruler, which can be proven by repeated application of the Pythagorean theorem. By using this formula as distance, Euclidean space becomes a metric space. Older literature refers to this metric as Pythagorean metric. [12]

2.3 Integrated Circuit

There are several speech recognizing ICs that can help to enhance the speech recognition process in this project. Microprocessors choices available are such as the Philips manufactured Hello IC and also a speech recognition IC, HM2007.

Hello IC

Hello IC's performance is able to interpret up to 100 words, with up to 50 words active at a time, and also functions for continuous speech recognition instead of the usually discrete word recognition. In other words it will be able to recognize the speech when user speaks in a sentence instead of one word at a time. Hello IC utilizes VoCon as its speech recognition software which enables it to work well even with background noise. However, the Hello IC is not easily found in the current market, and thus is not selected as choice of microprocessor utilized in this project.

HM2007 IC

Meanwhile, HM2007 is flexible to either recognize 40 words of 0.96 seconds length or 20 words but of 1.92 seconds length. This chip is able to work under a host computer as it does not occupy the existing CPU operation time. When the HM2007 recognizes a command it can signal an interrupt to the host CPU and then relay the command code. The HM2007 chip can be cascaded to provide a larger word recognition library. HM2007 IC is chosen to be utilized in this project because currently it is the most popular speech recognition IC in the market (through internet). Therefore, it can also be easily purchased.

CHAPTER 3
METHODOLOGY

In this section, the procedure that is carried out throughout this project will be discussed. Early stages of this project includes in depth research and understanding of speech recognition as it is the main and most complicated part of this project. Further understanding of MATLAB is also necessary as voice recognition is to be implemented with MATLAB.

3.1 The MATLAB Approach

Initially, I have taken the MATLAB approach for the voice recognition part in my project. The code I utilize for my project with aids from Internet is utilizing the Dynamic Time Warping methodology.

Later, the MATLAB codes are modified and edited to enhance and troubleshoot it for better adaptation to the project's functionality. So far the tools/software used for this project is only MATLAB.

Meanwhile, the output of data from MATLAB was planned to be connected to microcontroller for the control of the external circuit.

A common table lamp will be used, but a carbon microphone will be integrated so that it can detect voice commands from user and function respectively to the commands (On / Off).

The microphone will initially detect the command and will be connected to MATLAB for speech recognition. If the input matches with any of the desired templates, MATLAB will trigger the microcontroller to activate the external circuit to either on or off the table lamp.

3.2 Studying of speech processing complication

In efforts to overcome the speaker dependent speech recognizing program, I need to record several templates of speaking of the same word, in several ambience noise (quiet, with ambience noise, others talking, etc.)

The following MATLAB code is used for speech word recognition:

```
% Constant Recording
clc;clear;close all; %CLC clears the command window and homes the
cursor, CLEAR  Clear variables and functions from memory,

chos=0;
possibility=4;
while chos~=possibility,
    chos=menu('Constant Recording','On', ...
        'Database Info','Delete database','Exit'); %MENU   Generate
a menu of choices for user input.
    %-------------------------------------------------------------------
-------
    % On
    while chos==1
        any_key=input('Press any key to begin or press "q" to
quit..\nYou have 2 seconds to speak into the microphone after you
have pressed any key\n\n>>','s');
        if (any_key~='q' | any_key=='')

            load('sound_database.dat','-mat');
            Fs = samplingfrequency;
            durata          = 5
            micrecorder =
audiorecorder(samplingfrequency,samplingbits,1);
            disp('Appliance activated');
            record(micrecorder,durata);
            while (isrecording(micrecorder)==1)
                disp('Recording...');
                pause(0.5);
            end
            disp('Recording stopped.');
            y = getaudiodata(micrecorder, 'uint8');
            %----- code for speech recognition -------
            vettore_pesi = zeros(sound_number,1);
            D1 = specgram(y,512,Fs,512,384);
            disp('Database scanning...');
            for ii=1:sound_number
                D2                =
specgram(data{ii,1},512,Fs,512,384);
                SM                = simmx(abs(D1),abs(D2));
                %[p,q,C]           = dp(1-SM);
                [p,q,C]           = dpfast(1-SM);
                peso              = C(size(C,1),size(C,2));
                vettore_pesi(ii) = peso;
                message=strcat('Sound #',num2str(ii));
                disp(message);
            end
            [min_value,min_index] = min(vettore_pesi);
            speech_id              = data{min_index,2};
            %------------------------------------------
```

```
            disp('Matching sound:');
            message=strcat('File:',data{min_index,4});
            disp(message);
            message=strcat('Location:',data{min_index,3});
            disp(message);
            message = strcat('Recognized speech ID:
',num2str(speech_id));
            disp(message);
            msgbox(message,'Matching result','help');
    end
              %-------------------------------------------------------
------------------
    % Database Info
    if chos==2
        if (exist('sound_database.dat')==2)
            load('sound_database.dat','-mat');
            message=strcat('Database has
#',num2str(sound_number),'words:');
            disp(message);
            disp(' ');
            for ii=1:sound_number
                message=strcat('File:',data{ii,4});
                disp(message);
                message=strcat('Location:',data{ii,3});
                disp(message);
                message=strcat('Sound ID:',num2str(data{ii,2}));
                disp(message);
                disp('-');
            end
        else
            warndlg('Database is empty.',' Warning ')
        end

    else

    fprintf('Program aborted..\n\n');
    break

end
    end

    %----------------------------------------------------------------
--------
```

```
    % Delete database
    if chos==3
        clc;
        close all;
        if (exist('sound_database.dat')==2)
            button = questdlg('Do you really want to remove the
Database?');
            if strcmp(button,'Yes')
                delete('sound_database.dat');
                msgbox('Database was succesfully removed from the
current directory.','Database removed','help');
            end
        else
            warndlg('Database is empty.',' Warning ')
        end
    end
    %-----------------------------------------------------------------
--------
end
```

3.3 Debugging of MATLAB code

The code above has been modified to continuously scan for audio command input from the microphone with the interval of 5 seconds in between. All audio input will be performed DTW, and the matching template will be chosen. Circuit will be activated accordingly. Scanning and template matching will be continuous until user inputs 'q'.

The sound database will already have been configured to have the templates of commands.

Flow Chart of Voice Activated Table Lamp

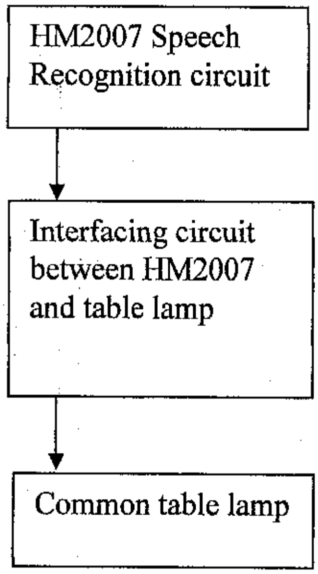

Figure 3: Flow chart of speech recognition utilizing HM2007 IC

Figure 3 illustrates the flow of the entire voice activated circuit controlling the appliance (table lamp). The main circuit (HM2007 speech recognition circuit) trains the HM2007 IC to recognize the commands that will be used to control the external voice activated circuit. It will be able to output through the ten pin right angle header, connected to the interfacing circuit. The command output by the speech recognition circuit will be recognized by the interfacing circuit and will activate the corresponding switch to turn on / off the external appliance connected to it.

Summary of work done:

1. Searching and studying the best IC to be integrated in speech recognition circuit.
2. Selecting HM2007 as the IC and purchasing it from a local company.
3. Learning the workability of PIC16F877 (microprocessor to control external circuit) while awaiting HM2007 to arrive from company.
4. Learning C programming on for PIC16F877 to be used in interfacing circuit.
5. Receiving HM2007 speech recognition circuit. Learn the practical workability and reliability of speech recognition.

6. Studying the interfacing of HM2007 speech recognition circuit to an external circuit (to control appliance).

3.4 Studying of speech recognition IC

I've done some research work on the possible ICs to be integrated into the circuit to enhance the speech recognition. My most beneficial find would be the Speech Recognition IC, Hello IC, which is a product of Philips Semiconductors. It is claimed to be the most economical yet high performance speech recognition chip for command and control application, which I believe should be functional for my project.

Although the elaborations of Philips Semiconductor focused on how it is implemented in cars initially, it also emphasized that it can be used at home – which I assume that it can be implemented on the 2 appliances I'm working on, doorbell and light. Unlike other speech recognizer, the Hello IC astoundingly has an accuracy of up to 95% even without the need to train it first!

Hello IC's performance is able to interpret up to 100 words, with up to 50 words active at a time, and also functions for continuous speech recognition instead of the usually discrete word recognition. In other words it will be able to recognize the speech when user speaks in a sentence instead of one word at a time. Hello IC utilizes VoCon as its speech recognition software which enables it to work well even with background noise.

This product is claimed to be first utilized in appliances since end of 2001 and has become less popular at the current market.

I have also found out about the HM2007 speech recognition IC which will be able to recognize speech independent of the speaker. While HM2007 was not ensured of the high performance rate as Hello IC, it is more widely found and can be obtained at a less hassled method.

3.5 Structures of HM2007

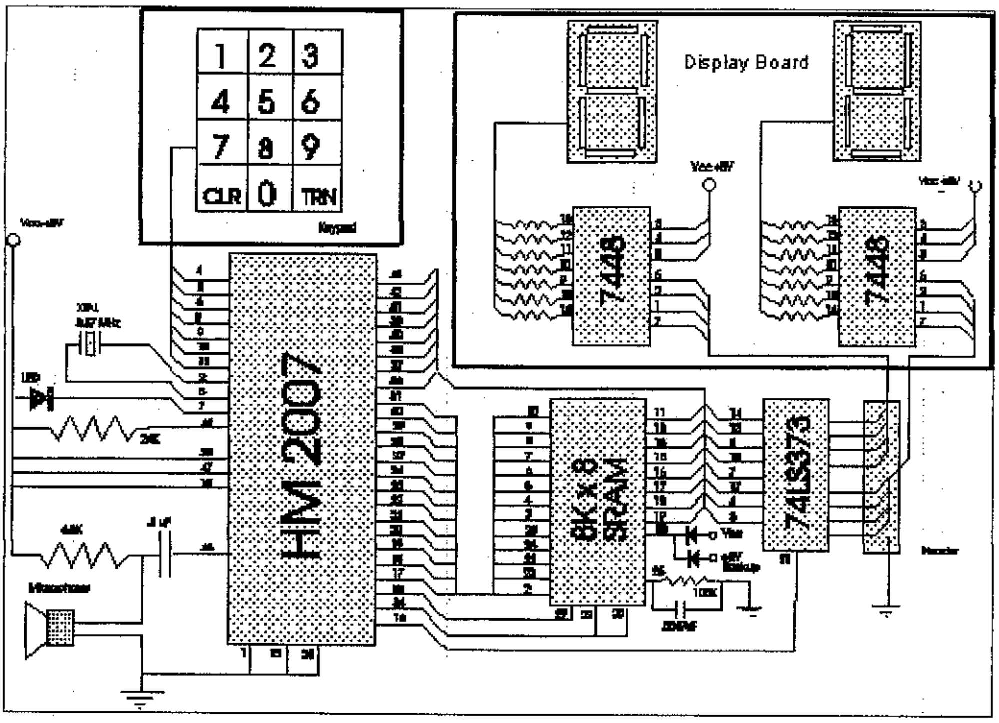

Figure 6: HM2007 integrated into a speech recognition circuit

HM2007 speech recognizer circuit as illustrated in Figure 6 is constructed by mounting and soldering 3 IC sockets. The HM2007 uses a 48 pin socket identified on the pcb as U1. The 8K static RAM uses a 28 pin socket identified as U2. The 74LS373 uses a 20 pin socket identified as U3.

A 6.8K resistor, R1 is mounted and soldered, and is followed by 22K resistor, R2, and later the 100K resistor, R3, and last but not least, 330ohms resistor R4. The rest of the components needed to construct the complete speech recognizing circuit is also mounted and soldered: the diodes, D1 and D2, the 3.57 MHz crystal, the red LED, the capacitors; C1 and C5 of 10-22p value, C2 - 1μF, C3, 47 to 100μF and C4 is a 0.0047μF, 7805 voltage regulator, the on-off slide switch, microphone jack, button batter holder and 9V battery cap, 10 pin right angle header, 7 pin right angle header. Integrated circuits are then installed appropriately into their respective sockets.

3.6 Structures of interfacing circuit

The interface circuit consists of two 4 to 16 line decoder. Both these IC's will receive output A1, B1, C1, D1 and A2, B2, C2, and D2 from the ten pin output of the main circuit respectively. Four 1k ohm resistors are also used to decrease the 5V source from the main board, to 1.2V to abide the specifications of the optocoupler IC. 4 relays, 2 single pole double throw (SPDT) and 2 DPDT relay 24vdc coil are used to control and latch the desired output for the control of the external appliance.

Initial Interface circuit (not used in the final result of this project)

This is the initial interface circuit implemented for the project. However, as it does not work perfectly, it is set aside. The construction of the circuit is as stated below:

The topside of the printed circuit board has white silk screened component drawings. The components are mounted on the top (silk screen side) of the pc board and soldering the components on the opposite side. After soldering the component excess wire is clipped off.

The construction done by mounting and soldering the needed components: ten 100K ohm resistors, ten 1 K ohm resistors, 10K ohm resistor, 15K ohm resistor, 330 ohm resistor, the 1N4007 diodes, the 1C sockets for the 74154,16F84, and 4011, the 10-position right angle female header, 4.0 MHz ceramic resonator, the LM339, on-off pc mounted switch, DC Power Jack, the LM2940 voltage, ten 2N3906 transistors, the ten relays, three screw terminal connectors, five units together using the tongue and groves before placing them onto the pc board, the second line of screw , capacitor Cl (100 uf) and bridge rectifier.

CHAPTER 4

RESULTS AND DISCUSSION

This chapter will discuss on the outcomes of the methodologies attempted in this project. This includes the attempt of utilizing MATLAB coding, and the switch to the approach of HM2007 microprocessor for the speech recognizing.

4.1 The MATLAB Approach

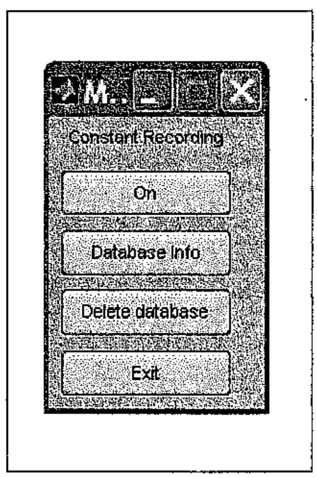

Figure 4: GUI for continuous audio command detecting and speech recognition

Figure 4 shows the interface for easy use of the MATLAB speech recognition code. The first option 'On', is to start the audio scanning and voice recognition. Database Info allows user to view the number of templates available. Delete database allows user to delete the available templates. This option will be eliminated later because user will not have the option to add their own or delete templates. Exit will terminate the program immediately.

```
Press any key to begin or press "q" to quit..
You have 2 seconds to speak into the microphone after you have pressed any key

>>d

durata =

     5

Appliance activated
Recording...
Recording...
Recording...
Recording...
Recording...
Recording...
Recording...
Recording...
Recording...
Recording...
Recording stopped.
Database scanning...
Sound #1
Sound #2
Sound #3
Sound #4
Sound #5
Matching sound:
File:Microphone
Location:Microphone
Recognized speech ID:0
>>
```

Figure 5: Program running to record and recognize the template

Figure 5 illustrates when the MATLAB code is activated by pressing the 'On' button as seen in Figure 4. The recording begins by prompting the user for the duration of time to record the input speech. Input speech is obtained from microphone connected to the CPU microphone input jack. The program will then start recording for 5 seconds. After 5 seconds, the program compares the set of frequency bands (after speech processing) to the templates that have been trained into the system beforehand. The matching ID speech is output. In this case the matching ID is 0.

4.2 The HM2007 IC approach

HM2007 recognizes the 3 main methods of spoken words as follow:

Isolated

Isolated speech is when each and every word is spoken discretely. The speech recognition circuit is set up to identify isolated words of .96 second lengths.

Connected

Connected speech is the condition when words are spoken in between the likeliness of isolated and continuous speech. This thus allows users to speak a line of multiple words where it can recognize up to a phrase of 1.92 seconds in length, but however reduces the memory size to only 20 words.

Continuous

Continuous speech is the common method we speak in. This method is still being researched and improved from time to time in hopes to come up with future perfect continuous speech recognition compared to the current development which still contain high error rates.

HM2007 Circuit diagram

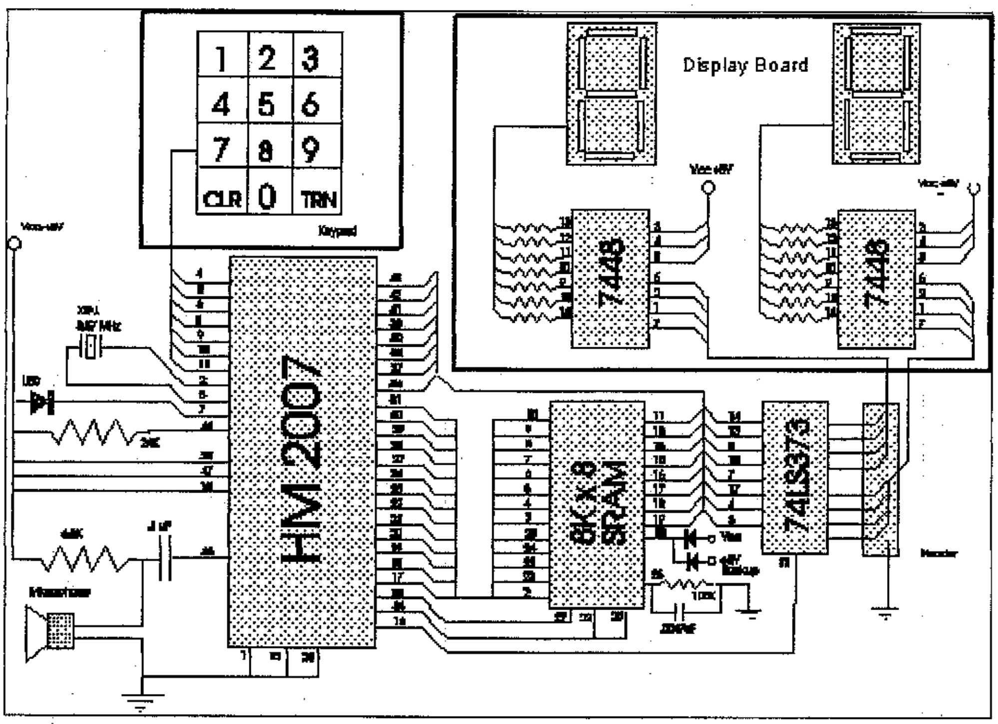

Figure 7: HM2007 integrated into a speech recognition circuit

4.3 Practical workability and reliability of speech recognition circuit

Parts of main speech recognizing circuit as illustrated in Figure 7:

- HM2007 IC for speech recognition - recognizing either forty 0.96s words / twenty 1.92s words
- Memory - 8K X 8 static RAM (store trained words)
- 74LS373 Octal Transparent latch - for address decoding (of trained words)
- 7448 - BCD to 7-segment decoder with ripple-blank input (active high outputs)

HM2007 IC

The HM2007 IC is capable of recognizing either up to 20 words of approximately 2 seconds long or 40 words of approximately 1 second long. This IC is the heart of the main HM2007 speech recognition circuit. This IC is trained with the command instructions that are meant to control the switching of the external appliance.

8k x 8 SRAM

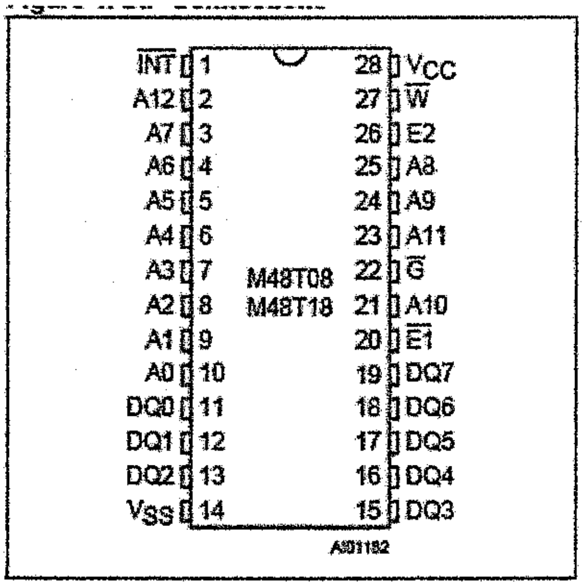

Figure 8: Structure of 8k x 8 SRAM

An 8K x 8 non-volatile static RAM shown in Figure 8 is used to store the commands input from the HM2007 IC.

The M48T08/18/08Y is in the WRITE Mode whenever W, E1, and E2 are active. This is when the commands trained using HM2007 is stored into the memory.

The M48T08/18 is in the READ Mode whenever W (WRITE Enable) is high, E1 (Chip Enable 1) is low, and E2 (Chip Enable 2) is high. This is when the commands trained and stored using HM2007 is read from the memory.

E1 or W must return high or E2 low for a minimum of tE1HAX or tE2LAX from Chip Enable or tWHAX from WRITE Enable prior to the initiation of another READ or WRITE Cycle.

74LS373 Octal Transparent latch

74LS373 IC's, are used as an interface between a system data bus and various system components for configuration and status information. Here, it functions to decode the address at where the command is stored inside the SRAM.

7448 BCD to 7-segment decoder with ripple-blank input

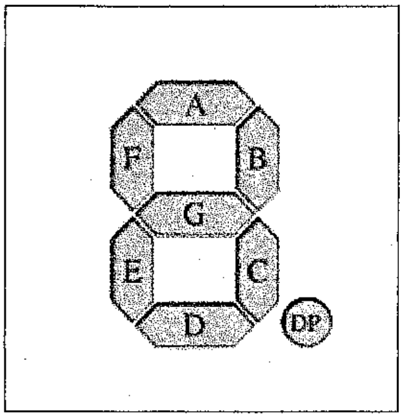

Figure 9: Seven segment display [7]

The seven segments are arranged as a rectangle of two vertical segments on each side with one horizontal segment on the top and bottom as illustrated in Figure 9. The seventh segment bisects the rectangle horizontally.

The HM2007 speech recognition circuit's performance is no doubt better than the MATLAB code that I developed in the past, but however, seemed to tend have a certain percentage of reliability. HM2007 is not able to recognize the speech 100% perfectly (sometimes it tends to recognize the wrong word, and sometimes it does not recognize the speech even if it has been trained to), and it tends to be speaker dependent as well. To reduce this error, more samples of speech need to be trained into the HM2007.

To make the recognition system simulate speaker independent, I need to collect various types of speaking (pronunciation) a certain control word (speech). As HM 2007 is able to recognize up to 40 words, the address for the first word being 01, second word, 02, and so on. Thus we make use of this double digit address to enable it recognize for more types of pronunciation. Training 01, 11, 21, 31 to be 4 different types of speaking 'Table lamp', and the output needs only to recognize the least significant bit to be '1' to activate the table lamp circuit.

The HM2007 will lose all the training once the main circuit is turned off. Thus to overcome this, a coin battery must be added to the circuit to supply the circuit with 3V backup for the SRAM, allowing the word patterns to be retained in the memory when the main circuit is turned off.

4.4 Interfacing HM2007 speech recognition circuit to an external circuit (to control appliance)

The 10 pin output from the HM2007 voice recognition circuit is connected to the interface circuit using a ribbon wire. The output is connected to two 74HCT154, to decode the 4 bit output from the main board. The 74HCT154 will output low voltage at the rightful output pin (Y_0 to Y_{15}) for the respective combinations.

In this interface circuit, we are only concerned with one bit, for the output from 1 to 4 (0001, 0010, 0011, 0100). Therefore, output is obtained from pin Y_1 to Y_4. Output is then connected to the negative terminals at the optocouple board. When a low output is received, this will create a 5V potential difference with the positive terminal. This is because the positive terminal is connected to the 5V source from pin 9 and 10 from the main board output. From the switches, 1k ohm resistors are used to reduce the 5V to

1.2V. This is because the optocoupler IC only needs 1.2V to function. The optocoupler acts and outputs like a transistor, therefore acting as a switch in this circuit.

Further operations of the circuit can be understood from the ladder diagram below:

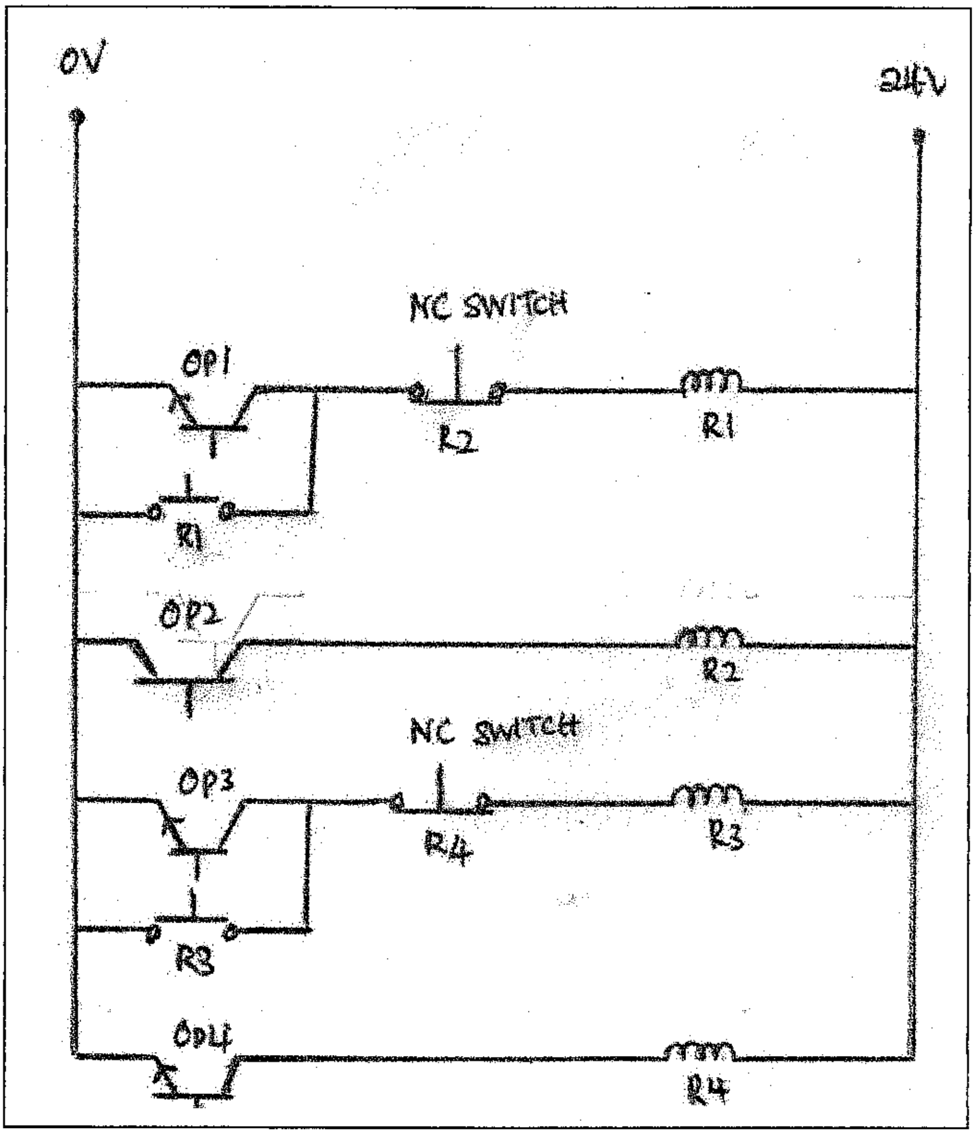

Figure 10: Ladder diagram of interface circuit operation

According to Figure 10 when OP1 is on, the output will be high. R1 is used to latch the output even when OP1 is no longer on. When R2 is on, the NC switch will open the first loop, thus causing the first output to be low (OFF). Thus, this can be summarized as R1 is used to ON the output of the first loop, while R2, is used to OFF the output of first loop.

The same logic is applied for R2 and R4 to control the second external appliance.

74HCT154, 4 to 16 line decoder / demultiplexer

The 74HC/HCT154 are high-speed Si-gate CMOS devices and are pin compatible with low power Schottky TTL (LSTTL). They are specified in compliance with JEDEC standard no. 7A.

The 74HC/HCT154 decoders accept four active HIGH binary address inputs and provide 16 mutually exclusive active LOW outputs.

In the interface circuit, we are only concerned of one bit from the output of the main board, from 1 – 4 (0001 to 0100). The respective combination of inputs and outputs are highlighted shaded in Figure 11.

FUNCTION TABLE

		INPUTS				OUTPUTS															
$\overline{E}_0$	$\overline{E}_1$	A_0	A_1	A_2	A_3	$\overline{Y}_0$	$\overline{Y}_1$	$\overline{Y}_2$	$\overline{Y}_3$	$\overline{Y}_4$	$\overline{Y}_5$	$\overline{Y}_6$	$\overline{Y}_7$	$\overline{Y}_8$	$\overline{Y}_9$	$\overline{Y}_{10}$	$\overline{Y}_{11}$	$\overline{Y}_{12}$	$\overline{Y}_{13}$	$\overline{Y}_{14}$	$\overline{Y}_{15}$
H	H	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
H	L	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	H	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	H	L	L	L	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	L	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	H	H	L	L	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	H	L	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H
L	L	H	L	H	L	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	L	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H
L	L	H	H	H	L	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H
L	L	L	L	L	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H
L	L	H	L	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H
L	L	L	H	L	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H
L	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H
L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H
L	L	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H
L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L

Note

1. H = HIGH voltage level
   L = LOW voltage level
   X = don't care

Figure 11: Input and Output combination for 74LS154

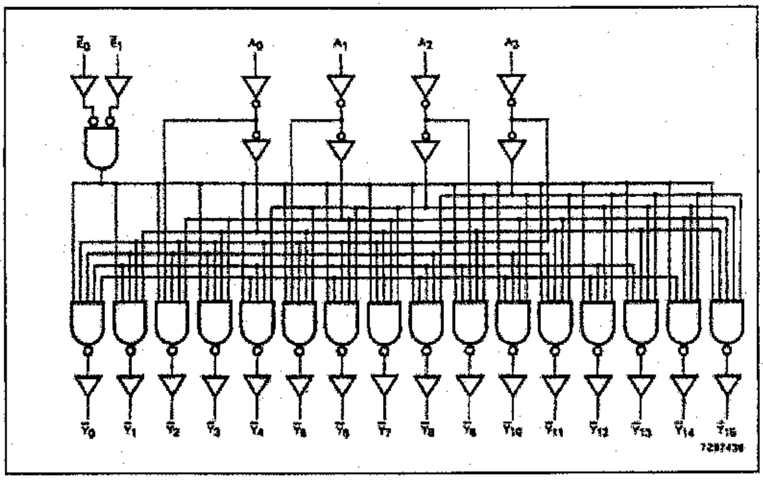

Figure 12: Logic circuit of 74LS154

4N32 - 6 pin Photodarlington optocouplers

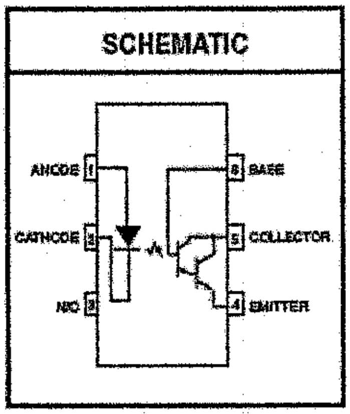

Figure 13: Structures of optocouplers

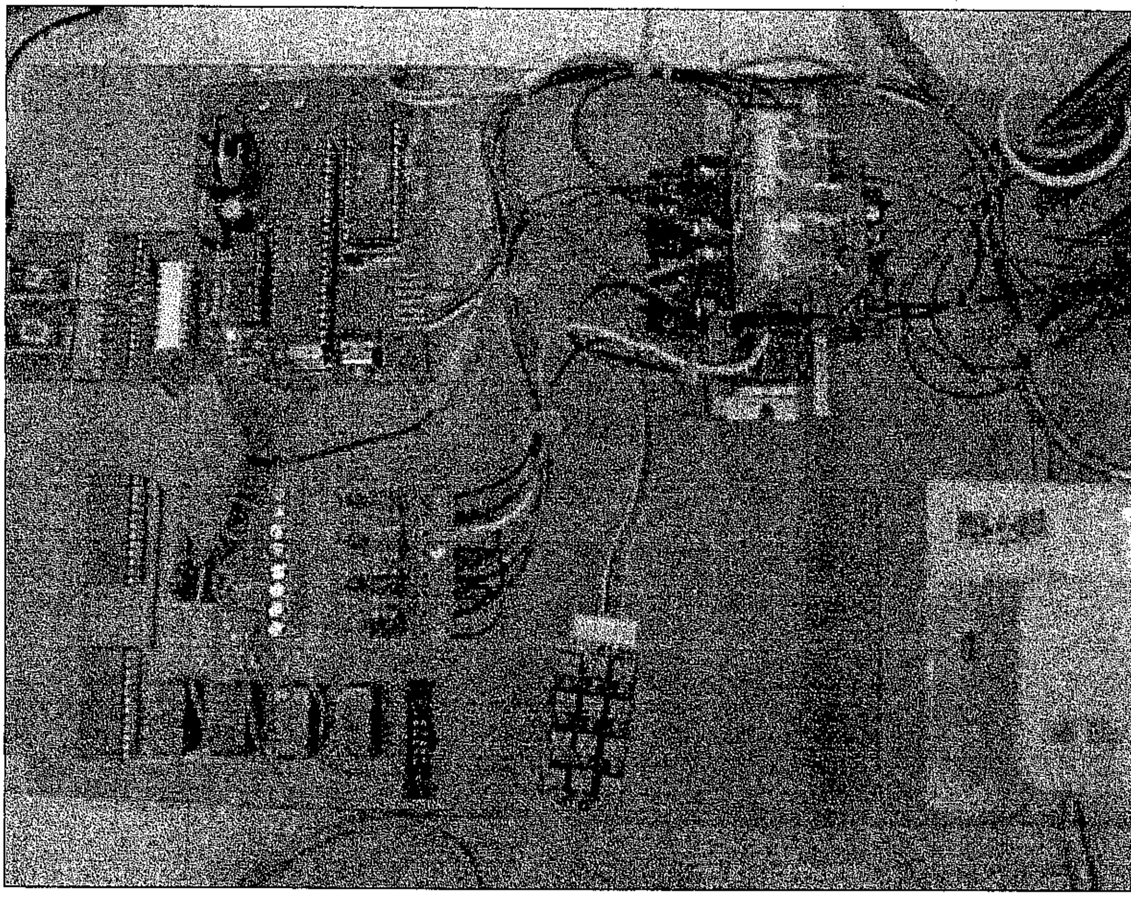

Figure 14: Completed circuit, with main board, interface circuit and external appliance.

From Figure 14, output 04 on LED, 0100 from 74LS154 will cause the second loop to output low voltage and thus OFF the second external circuit connected to switch 2.

Initial Interface circuit (not used)

Parts of Interface Circuit

- 4028 – 4-bit BCD to 1-of-10 active HIGH decoder; 10 outputs (high given based on input from 74LS373)
- The 10 relays controls ten external circuit with NO/NC switches – activated one at a time only (SPDT switch 125 VAC/0.5A or 24VDC/1A)
- Each relay corresponds to trained word location (1 to 10)
- Input 11 deactivates all relays on interface circuit
- 0.25 seconds of delay from the word recognition to relay activation

4028

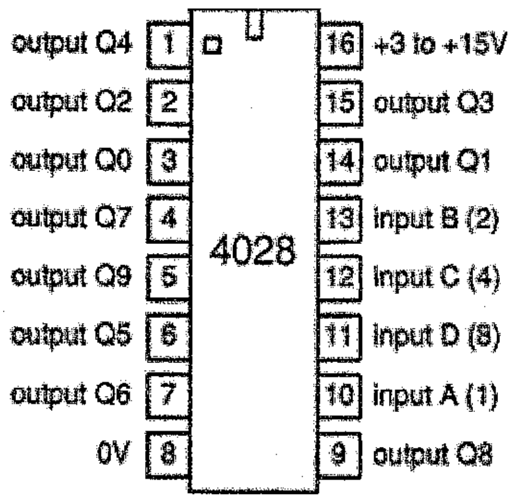

Figure 15: 4028

The HEF4028B is a 4-bit BCD to 1-of-10 active HIGH decoder. Here, it decodes the value fed from the main circuit. The value also indicates the location of the command. The location (number) of command corresponds to the number of the switch to be activated in the interfacing circuit.

Relays

The 10 relays control ten external circuit with NO/NC switches. Only one relay can be activated at time only (SPDT switch 125 VAC/0.5A or 24VDC/1A). The activated relay will cause NO switch to close and NC switch to open.

After training the HM2007 speech recognition circuit, the speech circuit can be connected to an interface circuit instead of the LCD display. The 10 position display header is plugged into the speech recognition 10 position femaled header.

There are 10 relays on this interface circuit, where the first trained word activated the relay 1 on the speech interface, the second word activates the second relay and so on. Only one relay can be activated at a time. To control the appliance, the control word is trained into the corresponding relay the appliance is connected to. On each relay, it contains 2 switches, the normally open and normally closed. When relay is activated, the open switch closes and the closed switched opens. The appliance can be deactivated by giving an input of '11' to the interface circuit, thus the control command to stop it should be trained at that address. There is approximately 0.25 seconds of delay from the word recognition to relay activation. The circuit automatically detects and discards the three possible error codes, 55, 66, and 77 from the speech recognition circuit.

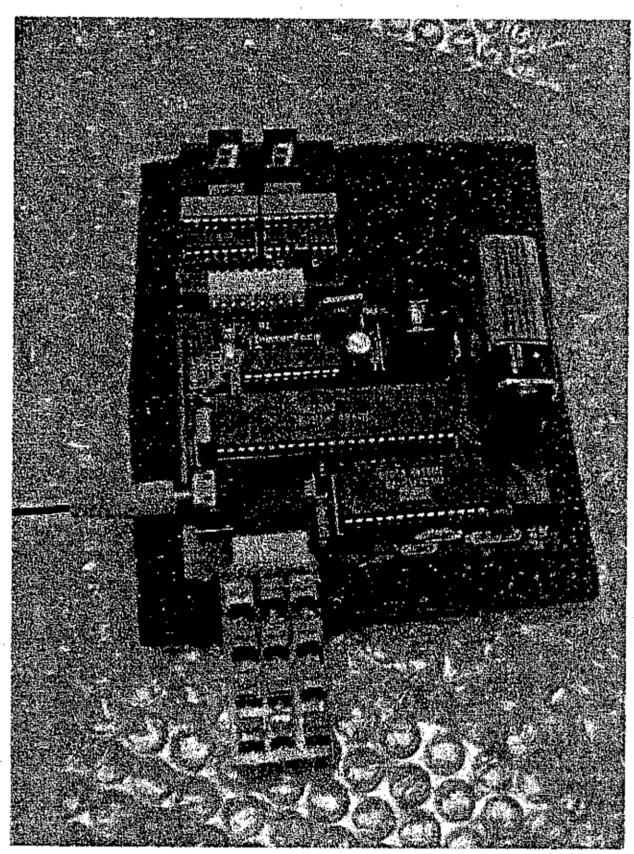

Figure 16a: Main HM 2007 speech recognition circuit - training mode. '77' on display indicates the speech spoken is not recognized by the circuit.

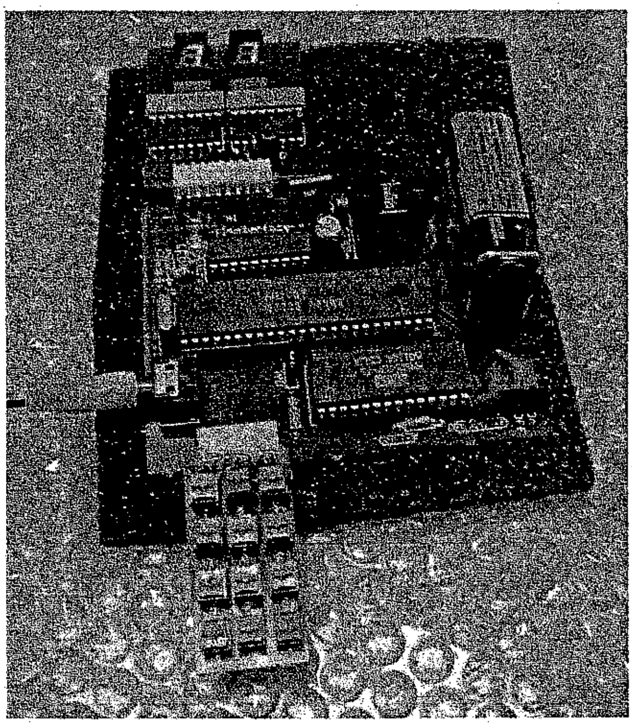

Figure 16b: Main HM 2007 speech recognition circuit – closer view

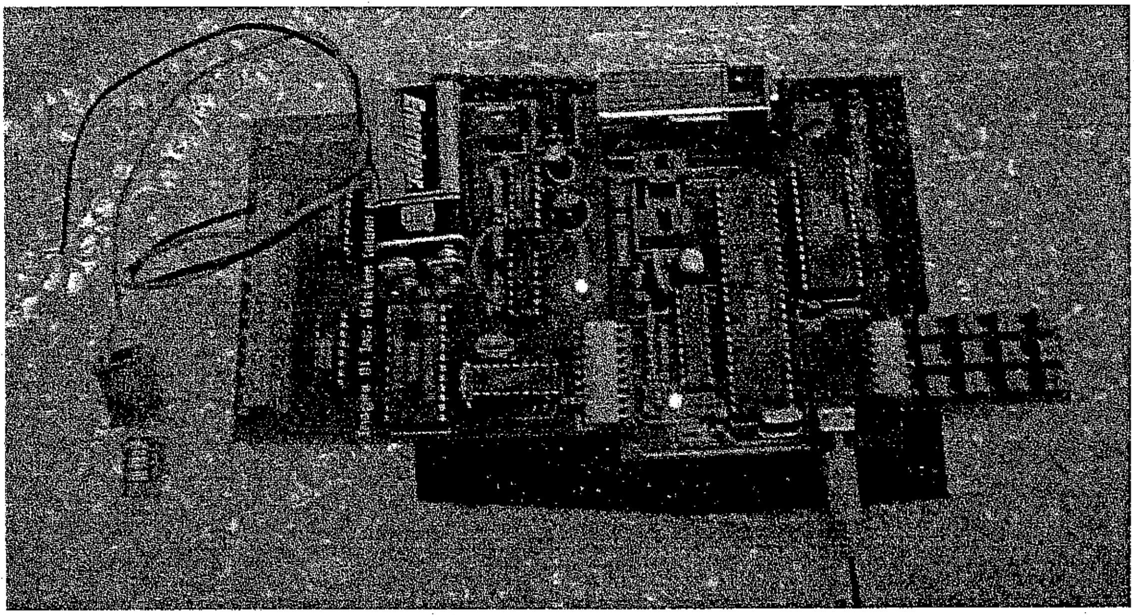

Figure 17a: Testing the trained HM 2007 speech recognition circuit with the interfacing circuit and the external appliance (the light bulb represents the table lamp).

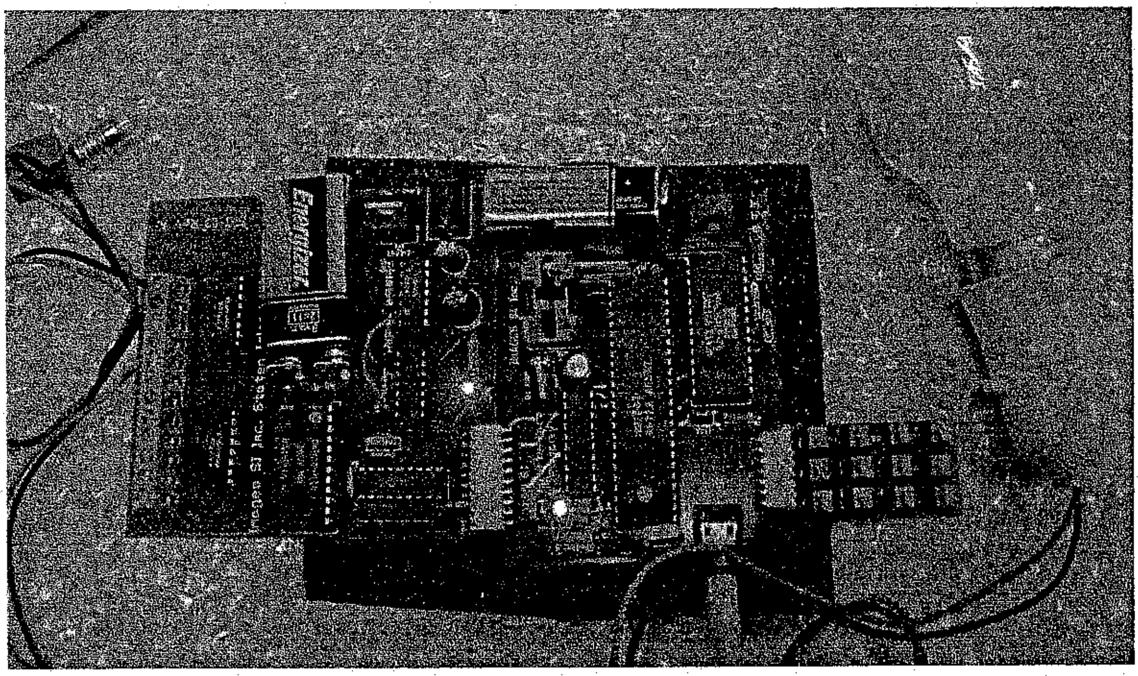

Figure 17b: Testing the trained HM 2007 speech recognition circuit with the interfacing circuit and the external appliance (another light bulb at the second relay)

Figure 16a and 16b shows the observation on the seven segment display when the HM2007 speech recognition circuit is being trained with commands. Both display shows '77' which is an error returned. The error indicates the speech spoken into the microphone is too long to be recognized by the circuit.

Figure 17a and 17b shows the trained circuit connected to the external appliance such as light bulbs in place of table lamp (to represent the table lamp).

CHAPTER 5
CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

The speech recognition part in this project is able to achieve accuracy of up to 95%.

Improvement can be done, by providing larger library with cascade of HM2007.

Accuracy enhancement by collecting more samples of speech.

Among the obstacles came over while implementing this project is the wide scope of coverage. It not only covers the control part but also the signal processing part. A lot of background research needs to be done to fully understand the theoretical part before programming can be done.

Also, much time has been spent to debug the program, using trial and error, thus the hardware implementation of the project was much delayed.

5.2 Recommendations

For future works, the HM2007 IC can be cascaded to have a larger word library which will also enable us to collect more samples of a certain word spoken thus increasing the reliability. The accuracy enhancement can be achieved by collecting more samples of voices spoken. This can be achieved when a larger library is available.

REFERENCES

[1] Infotech, Indiatimes, 11 June 2002, How Speech Recognition Works URL http://infotech.indiatimes.com/articleshow/12650262.cms

[2] John G Harris
URL http://www.cnel.ufl.edu/~kkale/dtw.html

[3] Connected Earth
URL http://www.connected-earth.com/FunandGames/Gadgets/Howdoesacarbonmicrophonework/index.htm

[4] Adan Galvan, 25 February 2004
URL http://cnx.org/content/m10821/latest/

[5] Free Patents Online, 14 March 1989, Weight Lifting Apparatus
URL http://www.freepatentsonline.com/4811946.html

[6] Wikipedia, 21 August 2001, Binary Code Decimal
URL http://en.wikipedia.org/wiki/Binary-coded_decimal

[7] Wikipedia, 23 March 2004, Seven Segment Display
URL http://en.wikipedia.org/wiki/7-segment_display

[8] Jose J Amador, 28 November 1995, Biphase Bus Monitor
URL http://www.patentstorm.us/patents/5471462-description.html

[9] Philips Semiconductors, January 1995, Decoders / Demultiplexers
URL http://www.digchip.com/datasheets/parts/datasheet/364/HEF4028B.php

[10] John Hewes, 4000 Series CMOS Logic ICs
URL http://www.kpsec.freeuk.com/components/cmos.htm#4028

[11] Mark Young, January 2004, Euclidean and Euclidean Squared
URL
http://www.improvedoutcomes.com/docs/WebSiteDocs/Clustering/Clustering_Parameters/Euclidean_and_Euclidean_Squared_Distance_Metrics.htm

APPENDIX 1: FULL MATLAB CODES

```
% Constant Recording
clc;clear;close all; %CLC clears the command window and homes the cursor, CLEAR
Clear variables and functions from memory,

chos=0;
possibility=4;
while chos~=possibility,
  chos=menu('Constant Recording','On', ...
    'Database Info','Delete database','Exit'); %MENU   Generate a menu of choices for
user input.
  %-------------------------------------------------------------------
  % On
  while chos==1
    any_key=input('Press any key to begin or press "q" to quit..\nYou have 2 seconds to
speak into the microphone after you have pressed any key\n\n>>','s');
    if (any_key~='q' | any_key=='')

      load('sound_database.dat','-mat');
      Fs = samplingfrequency;
      durata        = 5
      micrecorder = audiorecorder(samplingfrequency,samplingbits,1);
      disp('Appliance activated');
      record(micrecorder,durata);
      while (isrecording(micrecorder)==1)
        disp('Recording...');
        pause(0.5);
      end
      disp('Recording stopped.');
```

```
    y = getaudiodata(micrecorder, 'uint8');
    %----- code for speech recognition -------
    vettore_pesi = zeros(sound_number,1);
    D1 = specgram(y,512,Fs,512,384);
    disp('Database scanning...');
    for ii=1:sound_number
      D2           = specgram(data{ii,1},512,Fs,512,384);
      SM           = simmx(abs(D1),abs(D2));
      %[p,q,C]       = dp(1-SM);
      [p,q,C]      = dpfast(1-SM);
      peso         = C(size(C,1),size(C,2));
      vettore_pesi(ii) = peso;
      message=strcat('Sound #',num2str(ii));
      disp(message);
    end
    [min_value,min_index] = min(vettore_pesi);
    speech_id        = data{min_index,2};
    %-----------------------------------------
    disp('Matching sound:');
    message=strcat('File:',data{min_index,4});
    disp(message);
    message=strcat('Location:',data{min_index,3});
    disp(message);
    message = strcat('Recognized speech ID: ',num2str(speech_id));
    disp(message);
    msgbox(message,'Matching result','help');
end
     %------------------------------------------------------------------
% Database Info
if chos==2
  if (exist('sound_database.dat')==2)
```

```
        load('sound_database.dat','-mat');
        message=strcat('Database has #',num2str(sound_number),'words:');
        disp(message);
        disp(' ');
        for ii=1:sound_number
          message=strcat('File:',data{ii,4});
          disp(message);
          message=strcat('Location:',data{ii,3});
          disp(message);
          message=strcat('Sound ID:',num2str(data{ii,2}));
          disp(message);
          disp('-');
        end
      else
        warndlg('Database is empty.',' Warning ')
      end

    else

    fprintf('Program aborted..\n\n');
    break

  end
    end

    %------------------------------------------------------------------
    % Delete database
    if chos==3
      clc;
      close all;
      if (exist('sound_database.dat')==2)
```

```
        button = questdlg('Do you really want to remove the Database?');
        if strcmp(button,'Yes')
           delete('sound_database.dat');
           msgbox('Database was succesfully removed from the current directory.','Database removed','help');
        end
     else
        warndlg('Database is empty.',' Warning ')
     end
  end
  %--------------------------------------------------------------------
end
```

APPENDIX 2: BINARY CODED DECIMAL

The encoding section is shown in FIG. 7. This section takes the signal from the information stored in IC chip 74 LS 273 (FIG. 6) and encodes this stored information to a "Binary-to-Decimal" (BCD) coding format in a manner well known to those or ordinary skill in the circuit design art. This permits the signals to be used in a BCD-to-7-segment decoder (FIG. 8A). The BCD signals are then sent to the BCD-to-7-segment decoder shown in FIG. 8A at 122. BCD-to-7-segment decoder 122 is a commercial unit comprised of three IC chips 7448. Each IC chip 7448 converts the BCD into a 7-segment display signal. Next, the 7-segment display signals are sent from decoder 122 to a 7-segment driver (FIG. 9) and finally to a digital readout 116 for a visual display of the selection made. The 7-segment driver of FIG. 9 is also a known commercial unit such as transistor ECG 916. Digital readout 116 is shown schematically in FIG. 8B. Each digital display has a respective pin socket 124, 126 and 128 (FIG. 8C) in a known manner with pin sockets 124, 126 and 128 corresponding respectively to the hundreds, tens and single units of the digital display 116 shown in FIG. 8B.

In computing and electronic systems, **Binary-coded decimal (BCD)** is an encoding for decimal numbers in which each digit is represented by its own binary sequence. Its main virtue is that it allows easy conversion to decimal digits for printing or display and faster decimal calculations. Its drawbacks are the increased complexity of circuits needed to implement mathematical operations and a relatively inefficient encoding – 6 wasted patterns per digit. Even though the importance of BCD has diminished, it is still widely used in financial, commercial, and industrial applications.

In BCD, a digit is usually represented by four bits which, in general, represent the values/digits/characters 0-9. Other combinations are sometimes used for sign or other indications.

Basics

To BCD-encode a decimal number using the common encoding, each decimal digit is stored in a four-bit nibble.

Decimal: 0 1 2 3 4 5 6 7 8 9
BCD: 0000 0001 0010 0011 0100 0101 0110 0111 1000 1001

Thus, the BCD encoding for the number 127 would be:

0001 0010 0111

Since most computers store data in eight-bit bytes, there are two common ways of storing four-bit BCD digits in those bytes:

- each digit is stored in one byte, and the other four bits are then set to all zeros, all ones (as in the EBCDIC code), or to 0011 (as in the ASCII code)
- two digits are stored in each byte.

Unlike binary encoded numbers, BCD encoded numbers can easily be displayed by mapping each of the nibbles to a different character. Converting a binary encoded number to decimal for display is much harder involving integer multiplication or divide operations. The BIOS in many PCs keeps the date and time in BCD format, probably for historical reasons (it avoided the need for binary to ASCII conversion).

BCD in electronics

BCD is very common in electronic systems where a numeric value is to be displayed, especially in systems consisting solely of digital logic, and not containing a microprocessor. By utilising BCD, the manipulation of numerical data for display can be greatly simplified by treating each digit as a separate single sub-circuit. This matches much more closely the physical reality of display hardware—a designer might choose to use a series of separate identical 7-segment displays to build a metering circuit, for example. If the numeric quantity were stored and manipulated as pure binary, interfacing to such a display would require complex circuitry. Therefore, in cases where the calculations are relatively simple working throughout with BCD can lead to a simpler overall system than converting to 'pure' binary.

The same argument applies when hardware of this type uses an embedded microcontroller or other small processor. Often, smaller code results when representing numbers internally in BCD format, since a conversion from or to binary representation can be expensive on such limited processors. For these applications, some small processors feature BCD arithmetic modes, which assist when writing routines that manipulate BCD quantities

Concept and visual structure

A seven-segment display (abbreviation: "7-seg(ment) display"), less commonly known as a seven-segment indicator, is a form of display device that is an alternative to the more complex dot-matrix displays. Seven-segment displays are commonly used in electronics as a method of displaying decimal numeric feedback on the internal operations of devices.

APPENDIX 3: SEVEN SEGMENT DISPLAY

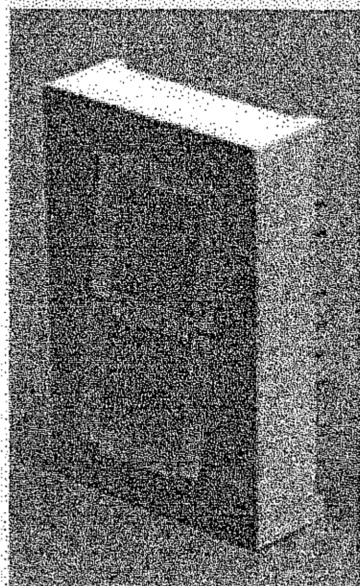

Figure 18: A typical 7-segment LED display component, with decimal point.

A seven segment display, as its name indicates, is composed of seven elements. Individually on or off, they can be combined to produce simplified representations of the Hindu-Arabic numerals. Each of the numbers 0, 6, 7 and 9 may be represented by two or more different glyphs on seven-segment displays.

The seven segments are arranged as a rectangle of two vertical segments on each side with one horizontal segment on the top and bottom. Additionally, the seventh segment bisects the rectangle horizontally. There are also fourteen-segment displays and sixteen-segment displays (for full alphanumerics); however, these have mostly been replaced by dot-matrix displays.

Often the seven segments are arranged in an *oblique*, or *italic*, arrangement, which aids readability.

The segments of a 7-segment display are referred to by the letters A to G, as follows:

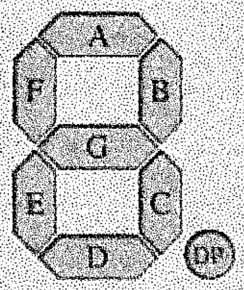

Figure 19: Structure of 7 segment display

where the optional DP decimal point (an "eighth segment") is used for the display of non-integer numbers.

APPENDIX 4: 74LS373 IC Octal Transparent Latch

74LS373 IC's, are used as an interface between a system data bus and various system components for configuration and status information. Flip-flops 318 and 319 are used to set bits for interrupt signals. Flip-flop 318 outputs an interrupt signal INTD on signal line 317 for the data FIFO. Flip-flop 319 outputs status register information. Finally, two IDT7202 FIFO's 320 and 322 provide data storage. FIGS. 8 and 9 are provide a more convenient representation of the inputs/outputs associated with conversion chips 230 and 231.

APPENDIX 5: 4028 IC

INTEGRATED CIRCUITS

DATA SHEET

For a complete data sheet, please also download:

· The IC04 LOCMOS HE4000B Logic Family Specifications HEF, HEC · The IC04 LOCMOS HE4000B Logic Package Outlines/Information HEF, HEC

HEF4028B MSI 1-of-10 decoder

Product specification File under Integrated Circuits, IC04 January 1995

Philips Semiconductors

Product specification

1-of-10 decoder

DESCRIPTION The HEF4028B is a 4-bit BCD to 1-of-10 active HIGH decoder. A 1-2-4-8 BCD code applied to inputs A0 to A3 causes the selected output to be HIGH, the other nine will be LOW. If desired, the device may be used as a 1-of-8 decoder with enable; 3-bit octal inputs are applied to inputs A0, A1 and A2 selecting an output O0 to O7. Input A3 then becomes an active LOW enable, forcing the selected output LOW when A3 is HIGH. The HEF4028B may also be used as an 8-output (O0 to O7) demultiplexer with A0 to A2 as address inputs and A3 as an active LOW data input. The outputs are fully buffered for best performance.

HEF4028B MSI

Fig.1 Functional diagram.

HEF4028BP(N): HEF4028BD(F): HEF4028BT(D):

16-lead DIL; plastic (SOT38-1) 16-lead DIL; ceramic (cerdip) (SOT74) 16-lead SO; plastic (SOT109-1)

( ): Package Designator North America PINNING Fig.2 Pinning diagram. A0 to A3 O0 to O9 address inputs, 1-2-4-8 BCD outputs (active HIGH)

4028 BCD to decimal (1 of 10) decoder

The appropriate **output Q0-9** becomes high in response to the BCD (binary coded decimal) input. For example an input of binary 0101 (=5) will make output Q5 high and all other outputs low.

The 4028 is a BCD (binary coded decimal) decoder intended for input values 0 to 9 (0000 to 1001 in binary). With inputs from 10 to 15 (1010 to 1111 in binary) all outputs are low.

Note that the 4028 can be used as a **1-of-8 decoder** if input D is held low.

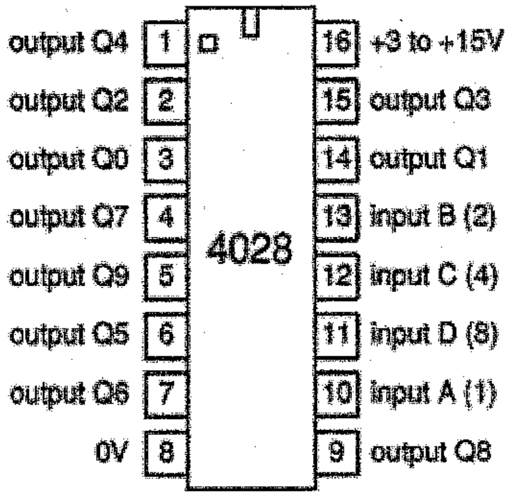

Figure 20: 4028

APPENDIX 6: 74HC/HCT154 – 4 to 16 line decoder

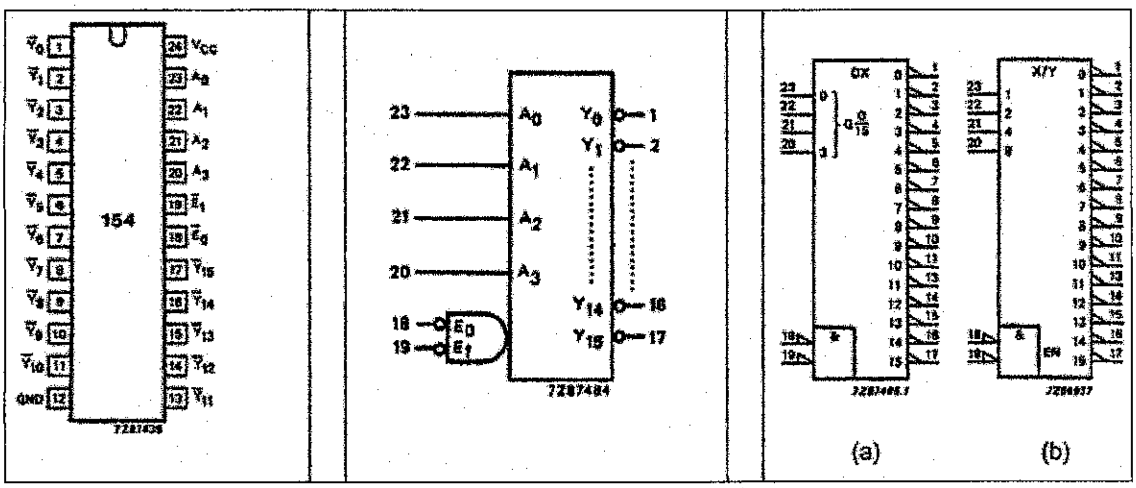

Figure 21: 4 to 16 line decoder: Pin configuration, logic symbol and IEC logic symbol

FEATURES

- 16-line demultiplexing capability
- Decodes 4 binary-coded inputs into one of 16 mutually
exclusive outputs
- 2-input enable gate for strobing or expansion
- Output capability: standard
- ICC category: MSI

GENERAL DESCRIPTION

The 74HC/HCT154 are high-speed Si-gate CMOS devices and are pin compatible with low power Schottky TTL (LSTTL). They are specified in compliance with JEDEC standard no. 7A.

The 74HC/HCT154 decoders accept four active HIGH binary address inputs and provide 16 mutually exclusive active LOW outputs.The 2-input enable gate can be used to strobe the decoder to eliminate the normal decoding "glitches" on the outputs, or it can be used for the expansion of the decoder.The enable gate has two AND'ed inputs which must be LOW to enable the outputs.

The “154” can be used as a 1-to-16 demultiplexer by using one of the enable inputs as the multiplexed data input. When the other enable is LOW, the addressed output will follow the state of the applied data.

APPENDIX 7: General Purpose 6-pin Photodarlinton Optocouplers

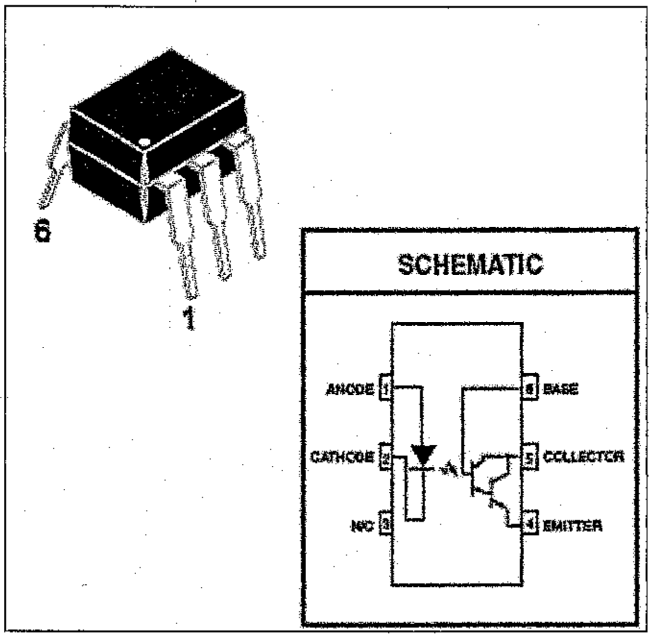

Figure 22: Photodarlington Optocouplers

DESCRIPTION
The 4N29, 4N30, 4N31, 4N32, 4N33 have a gallium arsenide infrared emitter optically coupled to a silicon planar photodarlington.

FEATURES
- High sensitivity to low input drive current
- Meets or exceeds all JEDEC Registered Specifications
- VDE 0884 approval available as a test option

-add option .300. (e.g., 4N29.300)

APPLICATIONS
- Low power logic circuits
- Telecommunications equipment
- Portable electronics
- Solid state relays
- Interfacing coupling systems of different potentials and impedances.