COMPUTER TELEPHONE INTERFACE

by:

MOHSEN BANAN

A thesis submitted in partial fulfillment

of the requirement for the degree of

Master of Science in Electrical Engineering

University of Washington

August 1982

Approved by:____
(Chairperson of Supervisory Committee)

Program Authorized
to offer degree ____ ELECTRICAL ENGINEERING __________

Date ____

Master’s Thesis

Signature_______________

Date____________________

This is the electronic reproduction of Mohsen Banan's Thesis in 1982. The original document was written on the word processor of Hewlett-Packard Emulator and printed using the dot matrix printer.

The HP tape was retrieved on November 4, 1998. The text was converted into LaTex format. Every effort was made to keep this copy as close as possible to the original.

This document is available in the following format:
PDF PS

University of Washington

ABSTRACT

COMPUTER TELEPHONE INTERFACE

By Mohsen Banan

Chairperson of the Supervisory Committee:

Dr. William E. Moritz

Department of Electrical Engineering

Speech synthesis, the production of intelligible speech from computer data, is no longer a novelty. Text-to-speech products with unlimted vocabulary are available. Soon, these products will be incorporated into telephone based inquiry systems that will give users access to data-bases from any phone. These systems would enable ordinary telephone sets to function as limited computer terminals. To accomplish this, in addition to the ability of sending information to the user in the form of computer generated speech, the system should be capable of accepting the data transmitted by the telephone (entries on the keypad of a Touch-Tone telephone set).

The aim of this project has been to study possible approaches leading to implementation of a system that would enable a computer to fully use an ordinary telephone set. The six major functions associated with using the phone are:

1. Picking-up the receiver
2. Dialing a number
3. Talking (transmitting information)
4. Listening (receiving information)
5. Hanging-up
6. Recognizing the ring.

Lifting and replacing the receiver is accomplished by using a solenoid that mechanically controls the cradle switch. Dialing is implemented by generating DTMF signals on the voice channel. A Votrax SC-01 phoneme synthesizer enables the system to talk. A MSD 3201 CMOS DTMF decoder/receiver can provide the ability to accept information (listen). Ring recognition may be accomplished through the use of a sound switch.

By using the above mentioned functions, a computer could allow for the use of an ordinary telephone set as a limited terminal. Although not in the form of a final product and totaly operational, the outcome of this project demonstrates the feasability and the practicality of implementing means of interaction between a computer and an ordinary telephone set.

Contents

List of Figures

PREFACE

This document is aimed at a technically oriented reader. The author has tried to provide sufficient background on the material presented. The interested reader is invited to further investigate his interests among the references.

Fundings for this project came from a variety of resources. The Votrax SC-01 phoneme synthesizer was purchased by the speech recognition budget number 63-4183. The telephone adapter was borrowed from Dr. R. L. Turner of Seattle University. Remaining parts were gathered by the authors personal funds.

A Hewlett Pacard 64000 Logic Development System (LDS) was used to develop the software. Various features of this system such as, Editor, Assembler, Linker, In-Circuit-Emulator and EPROM programer facilitated the implementation of this project.

I want to express my deepest appreciation to Prof. William E. Moritz for the use of his resources, for much needed advice and his demand for perfection. I also want to thank Prof. Alistair D. C. Holden for his encouragement and Miss Rita M. Hilton for her patience.

Chapter 1

INTRODUCTION

1.1 INTRODUCTION:

A computer terminal may be defined as: ”A device capable of transmitting and receiving data to and from a computer”. An ordinary telephone set can easily meet these requirements and may be considered a computer terminal. The already existing keypad on the ordinary Touch-Tone telephone set provides a limited means of transmitting data. The voice channel is available for receiving data.

The most common computer terminal in use is probably the the Video Display Terminal (VDT). A VDT consists of a typewriter style keyboard by which the user is able to transmit data to the computer in a rather comfortable manner, and a Cathode Ray Tube (CRT) through which the user visually accepts information from the computer. VDTs are rather expensive ($500 is a typical price), and they are bulky and heavy. In some applications it is desirable that the output of the computer be permanently preserved. In these applications, often the output is printed paper. A modified version of the VDT which has a printer as a means of output is suitable for these type of applications.

In some cases an extensive interaction between the user and the computer is not required and a limited means of Input/Output (I/O) is sufficient. For example, in the case of a microprocessor board, a limited keyboard which might consist of a 5 by 4 keypad is sufficient for input and a few 7 segment Light Emitting Diodes (LED) which display digits are sufficient for output.

Man-machine interaction thus far mentioned has been limited to two human senses: touch and vision. Lately a great deal of effort has been devoted to making use of the human sense of hearing and the ability to speak, to facilitate man-machine interaction. The area of speech recognition is not very developed yet, but the area of speech synthesis is quite advanced and several electronic speech synthesizers are presently available in the market.

Sometimes it is desirable to be able to communicate with a computer over a long distance. Requesting information of a data base from a remote location is one example. Several techniques are available to allow for long distance communication. Some techniques worthy of mention are satellite communications, fiber optics communication and special digital data channels. The extensive telephone network is an already existing means of communication between distant locations. The voice channel on the telephone network was designed for analog signals and is not directly suitable for exchange of data in the digital form. The use of a MODEM or MOdulator-DEModulator, is the common way for a computer to take advantage of the telephone network. A MODEM is a device that transforms digital data to analog signals suitable for transmission on the telephone voice channel and vice versa. MODEMs that are capable of functioning at high speeds are expensive and complex.

The fact that ordinary Touch-Tone telephone sets are in common use all around the U.S. and that they are capable of functioning as limited computer terminals, make the implementation of a system that allows for a telephone set to become a computer terminal very attractive. Such a system should be capable of:

1. Recognizing the received data (keypad entries on the telephone).

2. Transmitting data to the user in a convenient and comprehensible manner (digital speech).

The most convenient, natural, comprehensible and practical way of transmitting data to a user over a telephone line is speech. Although not in common use yet, good electronic voice synthesizers are readily available.

Several techniques are used to implement voice output. The standard reference in the area of speech generation is a book by Rabiner and Schafer , ”Digital Processing of Speech Signals”. Some interesting articles on this subject are also available in references 2 and 4.

The quality of generated voice is very dependent on the complexity of the method used. To a large extent, it is the nature of the application that dictates the desired speech quality. A computer room console voice that says, ”Printer number two is jammed” need not speak as beautifully as Orson Welles, because with any luck the printer will not jam with great regularity. Simple intelligibility is what is desired. In an application like training, however, extended exposure to unnatural voice will cause fatigue and frustration.

The amount of information to be transimitted to the user over the telephone line in most applications is not very extensive. For example in a Stock Price Quotation System, the length of the information to be transmitted may be as short as : ” American Telephone and Telegraph, 62-and 3/8, up 1/4” .

In those applications where the amount of information to be transferred is not large the quality of voice is not very crucial and a slight mechanical quality may be tolerable. Since in most applications the information to be transferred is not static (e.g. a data base is very often updated), the vocabulary needed to utter this dynamic information should be very large. Ideally an unlimitted vocabulary is desirable.

There are a limited number of sounds that make up words in the English language. In linguistics these sounds are referred to as phonemes. Any word in the English language may be created by by a sequence of phonemes. I chose the Phoneme Synthesis Technique to implement an unlimited vocabulary voice synthesizer. (Details of the speech synthesis technique are covered later.)

The user would input information to the computer using the Touch-Tone keypad which generates Dual Tone Multiple Frequency (DTMF) signals on the voice channel. Entries on the keypad of a regular Touch-Tone telephone set could generate this information. For example in the case of the Stock Price Quotation System, the user may make an inquiry about his desired stock by entering the market abbreviation followed by a terminator, say, A-T-T-*.

Associated with some keys on the telephone key-pad are 3 members of the alphabet set. This already existing characterstic could be used to make the use of Touch-Tone telephone set as a means of transmitting data even more attractive. The 12 keys on the telephone set therefore constitute the outgoing vocabulary of the telephone set.

There is more that a computer and a telephone set can do together. The computer can also dial desired numbers, recognize a ring and answer an incoming call.

There are 6 major functions associated with using the phone;

1. Picking up the receiver
2. Dialing a number
3. Talking
4. Listening
5. Hanging up
6. Recognizing the ring

In addition to these it is necessary to recognize the Dial Tone and the Busy Signal.

It is the aim of this project to explore possible approaches leading to the creation of convenient means of interaction between a computer and an ordinary telephone set while eliminating human intervention on the computer site. It is important to consider that the interfacing should be made as simple as possible so that most computers and telephones can be used.

The outcome of this project will not be a final product that enables most computers to fully interact with most telephone sets. Instead it represents an effort, limited by the available resources, to demonstrate the feasiblity and practicality of this basic concept.

Before going any further, ”Some Basic Telephone Principles” will be reviewed.

1.2 Some Basic Telephone Principles:

In the United States you can dial a phone number using two completely independent methods: Tones and Dial Pulses. The information presented below has been gathered from Refs. 1 and 3. Since in this project the existance of DTMF facilities is being assumed and is the method used, more emphasis will be placed on the Tone method.

TONES:

Each time you hold down a key on your push botton telephone set a pair of audio frequency signals is transmitted over the telephone voice channel. Central-office switching facilities decode these tones and connect the desired circuits based on the sequence of tone pairs received. Each tone must last long enough and there must be adequate separation between them. A tone pair duration of about 150 ms and a separation of about 75 ms works.

Each of these tones is composed of two pure sine waves of different frequencies superimposed on each other. These two frequencies explicitly represent one of the digits on the telephone keypad.

The telephone keypad can be thought of as a 4 row by 3 column matrix. Associated with each row is a specific frequency belonging to the low group (697 to 941 HZ) and corresponding to each column is a unique frequency of the high group (1209 to 1633 HZ). All the keys in a given row or column have one tone in common (see Table 1.1). For example, pressing the digit ”9” (row 3 and column 3) produces 852 Hz and 1477 Hz tones simultaneously, while pressing a ”5” produces 770 and 1336 HZ tones.

The full DTMF-encoding standard defines four rows and four columns for a total of 16 two-tone combination. Standard telephones use only 12 of these combinations. Depending on the application, these extra codes may be useful. Most tone decoding devices allow a 2 per cent tolerance on DTMF frequencies. This creates a range of acceptable frequencies, which is demonstrated in Table (1.2).

The telephone company prohibits the installation of unapproved equipment on the telephone lines. There is no problem with using the dual-tone, multiple-frequency method of dialing as long as the coupling is done through the microphone of the hand set and not by direct connections to the telephone line.

DIAL PULSES:

When you pick up the receiver on a telephone, an electrical connection is made to the lines leading to the central office. When you replace the receiver on the cradle the connection is broken or interrupted. This applies to both push-button and rotary dial telephones.

By periodically breaking the connections leading to the central office a number can be dialed. The number of interruptions is equal to the digit dialed, with the exception that ten interruptions corresponds to zero. These pulses may be generated at the rate of ten times per second and there should be a 1/2 second delay between each two digits.

The rotary dial on the telephone is a mechanical device which periodically breaks the connection leading to the central office. When the rotary dial is released, as it travels back to its resting position, it breaks the connection at a rate of ten times per

TABLE (1.1)

DTMF DIALING MATRIX

		HIGH GROUP			
		colum 0 1209hz	column 1 1336 hz	column 2 1477 hz	column 3 1633 hz
	Row 0 697 hz	1	2	3	A
LOW GROUP	Row 1 770 hz	4	5	6	B
	Row 2 852 hz	7	8	9	C
	Row 3 941 hz	*	0	#	D

TABLE (1.1)

Table (1.1): The dialing matrix of DTMF (Dual Tone Multiple Frequency) signaling system. The low group frequencies correspond to the matrix row; the high group frequencies correspond to the column. Column 3 is for special applications and is not normally used.

TABLE (1.2)

ACCEPTABLE DTMF FREQUENCIES

DTMF FREQUENCY (HZ)	LOWER DETECTION FREQUENCY LIMIT (HZ)	HIGHER DETECTION FREQUENCY LIMIT (HZ)	ACCEPATABLE FREQUENCY RANGE (HZ)
697	683	711	28
770	755	786	31
852	834	869	35
941	922	960	38
1209	1184	1233	49
1336	1309	1363	54
1477	1447	1507	60
1633	1600	1666	66

TABLE (1.2)

Table (1.2): The standard DTMF frequencies with the minimum and maximum values accepted within the 2 percent tolerance of most digital tone decoding devices (see Ref. 1).

second thus dialing the digit.

Numbers can also be dialed by pushing the cradle switch button at a rate of ten times per second. A solenoid plunger that is mounted to depress and release the cradle switch on the telephone set may be used to dial numbers using the dial pulses technique.

1.3 Implementation Summary

The final outcome of this project, which I would like to refer to as the Computer Telephone Interface (C.T.I.), is a means of simplifying the interface between a host computer and an ordinary telephone set. It is desirable to make the connection between the computer and the C.T.I. as simple as possible. The computer industry's standard on peripheral interfacing is the RS-232. I found it natural to use an RS-232 communication channel to interface the computer to the C.T.I. . The telephone set interfaces with the C.T.I. through an acoustic coupler and a relay that mechanically controls the cradle switch. No direct connection to the telephone line will be required since all connections are purely acoustic or inductive. This adds generality and ease of use to the system. I am assuming that telephone facilities in use are capable of recognizing Dual Tone Multiple Frequency (DTMF) signals.

The C.T.I. is a microprocessor based system. By using a microprocessor, the required hardware is minimized. A great deal of effort has been devoted to the implementation of the desired tasks in software. A brief summary of the methods used to implement the desired functions follows.

Cradle Control: Picking up the receiver and putting it down is realized by a relay (solenoid) that has mechanical control over the cradle switch of the telephone set. The processor controls this relay by sending appropriate signals to a peripheral driver which controls the relay.

Dialing: The desired number can be dialed using two different methods. Tone generation and dial pulses. Both methods are implemented.

Tones: The processor with the help of some minimal external hardware generates the DTMF signals on the voice channel. These tones are recognized by the telephone switching circuitries and the desired number is dialed.

Pulses: As mentioned in section (1.2), numbers can also be dialed by pushing the cradle switch button at a rate of ten times per second. To realize this, the method described under cradle control has been used.

Talking: The computer is able to talk with an almost unlimited vocabulary (limited only by the size and complexity of the program running on the computer). Continuous speech is generated by sequential arrangement of phonemes. The voice generated is intelligible, but has a slight mechanical quality.

Hearing: The computer accepts information by decoding DTMF signals arriving on the voice channel. Entries on the keypad of a a regular Touch-Tone telephone set could generate this information. The 12 keys on the telephone set constitute the incoming vocabulary of the computer.

Details of design and implementation of the above mentioned functions appear in the remaining pages of this document.

Chapter 2

DESIGN

2.1 METHOD

Considering the variety and the complexity of the functions to be performed and the fact that a rather complex controlling structure needs to coordinate the overall activity of the system, a microprocessor based system is ideally suited for this application. In addition to coordinating the functions of the system, the processor itself may be used to simplify the implementation of some of the desired functions or tasks. The use of a microprocessor has the advantage that it provides minimum parts count, thus decreasing the size and cost of the system. Future design changes are quickly and easily implemented primarily by changing the program, thus reducing shop and material costs. Software is more flexible than hardware.

The Top-Down policy approach has been followed in the design of this system. This approach is basically a step-wise refinement. First, the general structure is created. The problem is broken into smaller segments and each one is dealt with individually. This process is repeated until the problem segment in hand is manageable.

The basic functions to be performed by this system could be distinctively divided in the following manner:

1. Cradle Control
2. Talking
3. Listening
4. Dialing
5. Ring Recognition

The host computer can demand the execution of any of the above mentioned functions through the use of available commands. I will be referring to the routine that accepts the commands from the host computer and coordinates the execution of the desired tasks, as the ”Control Structure”. Before getting involved with the description of the control structure, a list of the available commands will be presented.

2.2 AVAILABLE COMMANDS

C.T.I. accepts commands from the host computer and performs the desired task. Available commands are:

`LIFT`	`The receiver is picked up and the telephone set is available for use.`
`DIAL "Number"` `DIALP "Number"`	`The desired phone number is dialed. "Number" is a sequence of up to 16 digits. Two methods of dialing are available. "DIAL" dials the desired number by generating DTMF tones. "DIALP" uses the pulse method.`
`TALK "Phoneme"` `TALK2 "Phoneme"` `TALK3 "Phoneme"` `TALK4 "Phoneme"`	`A sequence of phonemes are uttered by the speech synthesizer. It is the responsibilty of the host computer to organize these phonemes so that English sounding words are pronounced. "Phoneme" is a sequence of up to 70 ASCII characters which correspond to the desired phonemes. Gross variations on the pitch may be accomplished by`

	adding one of the pitch control parameters (2,3,4) to the command "TALK".
HEAR	A code corresponding to one key on the telephone keypad is transmitted to the host computer. This code is an element of the keypad set: {1,2,3,4,5,6,7,8,9,0, *,#,A,B,C,D}.
DOWN	Connections to the telephone facilities are broken. This is equivalent to hanging up the receiver.
RING	If the phone rings after this command has been issued, the message "RING DETECTED" is transmitted to the host computer.

2.3 CONTROL STRUCTURE

C.T.I. is continuously looking for inputs from the host computer. If the input is a legal command, the routine that performs that desired function is called and the function is performed. If it is not, an error message is transmitted.

Figure (2.1) is a block diagram of the software design. The names of the blocks refer to routines that are described in Chapter 4 and listed in Appendix B. CONTROL is the routine that coordinates the execution of the desired tasks. After initializing the system, it calls the REC_COM routine. REC_COM is the routine that compares the input against a list of legal commands and then calls one of the routines immediately below it (see Fig. 2.1) which brings about the execution of the desired function. Details are provided in section 4.7. The methods considered and my choice of implementation for each of the desired functions are discussed in the remaining sections of this chapter.

2.4 CRADLE CONTROL

A simple relay (solenoid) has mechanical control over the cradle of the telephone set. The processor controls this relay through an output port. By sending appropriate signals the processor can effectively push down or release the cradle which is equivalent to picking-up or hanging up the receiver.

2.5 TALKING

Reasonably good electronic voice synthesizers are readily available. A brief review of speech synthesis techniques is provided in the following section.

2.5.1 Methods Considered

Three major techniques are presently used to electronically synthesize the human voice : formant synthesis, linear predictive coding (LPC), and waveform digitization (see Ref. 7).

Formant Synthesis is essentially an electronic modeling of the natural resonances of the human vocal tract. Bands of resonant frequencies in the vocal spectrum, called formants, are generated by excitation sources and then passed through variable filters.

One variation of the formant technique is called phoneme synthesis, in which the spectral parameters are derived from basic sound units that make up words. A phoneme generator circuit is used to produce these sounds. In such a circuit, each phoneme is given a numeric code, and the synthesizer circuit utters phoneme sounds corresponding to codes it receives when it is activated. Words and sentences are assembled by simply stringing the phoneme codes together. The electronic voice so generated is intelligible but has a slight mechanical quality. Continuous speech using phoneme synthesis could typically be generated with a data rate of less than 100 bits per second (bps).

Linear-predictive coding is similar to formant synthesis in that both techniques are based in the frequency domain and use similar hardware to model the vocal tract. The quality of voice is often better than formant or phoneme synthesis, but a higher data rate (1200 to 2400 bps) is needed for continuous speech.

Waveform digitization is the third method of speech synthesis, in which the amplitude characteristics of a vocal waveform are stored and reproduced. The quality of speech is better than the other two methods, but the data rate for continuous speech is very high, and storing sufficient amounts of data conveniently can be a problem. Various schemes of compressing the data have been devised (see Ref. 7).

2.5.2 The Choice

For the purposes of this application an unlimited vocabulary voice synthesizer is desirable. I chose the phoneme synthesis technique to implement the voice synthesizer and in particular used the VOTRAX SC-01 Phoneme Synthesizer which is available from:

```
VOTRAX
A Division of Federal Screw Works
500 Stephenson Highway
Troy, Michigan 48084
tel: (313) 588-2050.
```

By sequencing phonemes that make up English words one can create an unlimited vocabulary. Continuous speech can be generated with a data rate of 100 bps. A phoneme sound is generated when a phoneme code is placed on the control register and is strobed (see section 4.4).

A program that goes over an English written text and finds the phonemes codes corresponding to that word from a vocabulary table may be developed on the host computer. Once the phoneme codes are ready, ”talking” can be realized by simply sending these codes to the speech synthesizer.

2.6 LISTENING:

Ideally it is desirable to create a system capable of accepting information on the telephone voice channel in a given natural language like English. At this point in time the area of speech recognition is not developed enough so that this scheme could easily be implemented. The area of speech recognition is beyond the scope of this project.

The readily available keyboard on the ordinary telephone set is a limited means of transmitting data. Accepting this information may be realized by implementing a DTMF tone decoder. Substantial effort has been devoted to the decoding of DTMF signals and various methods are available.

2.6.1 Methods Considered

A detailed discussion of several different methods for decoding DTMF signals is available in Ref. 1. Here, I will try to mention and briefly discuss some of these methods.

1. Tuned Inductance/Capacitance:
   The Bell System uses a Touch-Tone DTMF receiver consisting of tuned inductance/capacitance circuits and relays. This type of transistorized analog tone decoder is quite accurate but very bulkly.

2. Discrete Filter:
   Seven (The fourth high group frequency is normally not needed) phase-locked-loop frequency detectors that are adjusted to detect the presence of the particular frequencies of current interest are connected in parallel. Decoding simply consists of determining which pair is detecting tones.

3. Integrated Tone-Receiver Chips:

   Several chips are available to perform this task. They are dependable and easy to use. Some worthy of mention are:

   - ITT MSD 3210 Hybrid DTMF Tone-Receiver
   - ITT MSD 3201 CMOS DTMF Tone-Receiver
   - ITT 3044/3045 Group Filter.

   The price range is around $85 each and they may be purchased from the Micro Mint Inc.

4. Digital Filtering :
   This method is very similar to discrete filtering. Seven digital bandpass filters may be implemented and their output be tested for detection.

5. Software F.F.T.
   Fast Fourier Transform (F.F.T.) is a mathematical procedure that provides information about the frequency spectrum of a time domain signal. An F.F.T. may be performed on the incoming signal and the frequency domain be searched for particular frequencies of interest. More information will be provided in section 2.6.2.1.

2.6.2 The Choice

The method that I first tried was the software F.F.T. The software is very complex and the amount of memory required is substantial. It is also very hard to implement this method in real time. For these reasons this method is not being used. I did not get involved with the implementaion of this method and my effort was limited to the design.

The next choice was the MSD 3201 CMOS DTMF tone receiver. Details of design and implementation are provided in sections 2.6.2.2 and 4.5.

Software F.F.T. Decoding

Since a processor is available and computational power may easily be accessed, a software approach based on an F.F.T. routine seems very attractive. A microphone, mounted close to the speaker of the telephone set, can pick up the signals arriving on the telephone voice channel and deliver them to an Analog-to-Digital converter (A/D). Output of this A/D, which is a binary value, will be passed to the processor through an input port. The processor accepts a predefined number of samples and performs an F.F.T. on the sequence. The result is the frequency spectrum of the input signal. A search will then be made for the dominant frequencies and if they correspond to acceptable DTMF frequencies (see Table 1.1) then the key is decoded.

The highest frequency of interest is 1633 Hz, thus a filter with a cut-off frequency of about 2kHZ would be fine. The sampling rate should then be about (2*2000=4000 HZ). This means that the input needs to be sampled every 250 microseconds.

Truncation of a periodic function at other than a multiple of the period results in a sharp discontinuity in the time domain or equivalently results in sidelobes in the frequency domain. Some type of windowing should therefore be performed. A Hamming window is suitable.

The duration of sampling or the length of the sequence, is the factor that dictates the density of frequency samples in the frequency domain. Assuming that an average of three samples is desired in every acceptable frequency range, (Refer to Table (1.2)), frequency samples should be at least 28/3=9.33 Hz apart. This corresponds to a sequence of length 428. Since F.F.T. is most effective on powers of 2 , a sequence of length 512 appears optimum. This corresponds to a frequency sample every 7.8125 Hz. Sample numbers corresponding to the critical frequencies may then be calculated.

```
Critical Frequency: 697 770 852 941 1209 1336 1477 1633
Critical Sample : 89 99 109 120 155 171 189 209
```

After an F.F.T. has been performed, a search routine should be called. This routine finds out what the dominant frequencies are in each group. In the low group an average of 3 samples may be taken around each critical sample and then compared with the remaining 3 critical averages. In the high group an average of 5 samples may be taken around each critical sample and then be compared against the remaining critical averages.

The 3201 Integrated DTMF Receiver

The MSD 3201 is a complete DTMF receiver detecting a selectable one of 12 or 16 standard digits. No front end filtering is needed. The only externally required components are a 3.58 MHZ oscillator and two low tolerance by-pass capacitors. As described in section 2.4.1 several other integrated tone receiver chips are are available in the market. The reason for the choice of the 3201 was its availability. Details of the operation of the 3201 are covered in section 4.5.

2.7 DIALING:

As mentioned in Chapter 1, two methods of dialing are available: Pulses and Tones. There are several methods of encoding DTMF signals and generating Pulses. Some have been considered.

2.7.1 Methods Considered

1. Hardware tone generation:

   Telephone companies have traditionally used transistor LC oscilators to encode (generate) the DTMF tone pairs. An alternative is to use an integrated tone-encoder component, such as the MM53125 from National Semiconductor and the MK5087 from Mostek. Referred to as integrated tone dialer circuits, these chips divide a 3.579545 MHZ reference frequency into the eight DTMF frequencies. The frequency combinations are selected by a 12 or 16 key matrix keypad connected directly to the chip. The output is a stair-step D/A approximation of the mixture of the high- and low-group tones. No frequency adjustment is necessary to meet standard DTMF specifications, and the average circuit configuration requires little more than the keypad, a crystal, and the integrated circuit. Figure 2.2 shows a block diagram of the MK5087 and a typical DTMF encoder circuit. Radio Shack sells an encoder complete with a 12-key keypad. Using an MM53125, the CEX-4000 tone generating keypad module (catalog number 277-1010) would cost $16.95 .

2. Software Tone Generation:

   Frequency Domain;

   If a Fast Fourier Transform (F.F.T.) routine is available, tone generation may very easily be implemented. F.F.T. and Inverse F.F.T. (I.F.F.T.) are computationaly very similar. Inverse Fast Fourier Transform generates the time domain values of a signal from its frequency domain samples. Only very slight modifications are required to allow the use of a F.F.T. routine to implement an I.F.F.T. (see Ref. 6).

   If in the frequency domain all samples are made zero except those corresponding to the desired tone, then by performing an I.F.F.T. on this sequence, time domain values of the tone will be available. To generate the tones it is sufficient to convert these digital values to analog signals that go into a speaker.

   Time Domain;

   If a sine table is swept at different rates different frequencies are generated. To generate a desired tone the sine table can be swept at rates corresponding to the frequencies associated with the tone and the resulting values superimposed on each other. This is discussed in section 4.3.1.

3. Dial Pulses

   By periodiaclly breaking the connections leading to the central office a number can be dialed. One of the ways that these connections could be broken is through the cradle switch. By pushing the cradle at a rate of ten times per second one can dial a number.

2.7.2 The Choice

I decided to use both, the tone and the pulse methods of dialing. To generate the DTMF signals, a time domain software oriented method is used. The reason for this choice is that in addition to being flexible, the required hardware is minimal. Details of this are covered in section 4.3.2. To generate the pulses, the cradle switch is pushed and released through the cradle control method previously described.

2.8 Ring Recognition

The method that I decided to use is basically a sound switch. A microphone close to the telephone case ”hears” the ring and sets a bit on an input port (see section 4.6).

Chapter 3

THE SYSTEM

3.1 SYSTEM STRUCTURE:

Figure (3.1) describes how the host computer controls the telephone set through the C.T.I. The host computer communicates with the C.T.I. through an RS-232 interface. ASCII characters are serially transmitted and received. The Baud rate (bits per second) may be adjusted on the C.T.I. site. The interface between the C.T.I. and the telephone set consist of an audio coupler (a speaker and a microphone) and a relay that depresses or releases the cradle switch on the telephone set. I will be referring to this package, the relay, the speaker and the microphone as the Telephone Adapter.

Since no direct connections to the telephone lines are needed, any telephone set may be used and the approval of the phone company is not required. This adds generality to the system and makes the interfacing process as simple as it could be.

The host computer controls the telephone set by sending legal commands to the C.T.I. (See section 2.1). As a simple example let us consider the sequence of commands required to call my house and say "Hello". The host computer needs to send the following sequence of ASCII characters.

```
LIFT [return]

DIAL "3258548" [return]

TALK "[B#X#57?" [return]

([B#X#57?) are the  phoneme  codes  required  for  the
Votrax SC-01 to bring about the utterance of "Hello".
```

3.2 OVERVIEW

C.T.I. is a microprocessor based device. An Intel 8085 microprocessor controls the system. In addition to controlling the overall function of the system, the processor itself plays an important role in the implementation of some of the desired functions. This makes the required external hardware minimal. Figure (3.2) presents a block diagram of C.T.I. . It consists of:

1. SDK-85
2. Data Acquisition System
3. Speech Synthesizer Board
4. Telephone Adapter.

A description of the individual parts of the C.T.I. is provided in the following sections of this chapter.

An enhanced version the Intel System Design Kit-85 (SDK-85), microprocessor board is used in the prototype implementation. SDK-85 accepts commands from the host computer and brings about the desired action by using other components.

The Data Acquisition System (D.A.S.) is connected to a set of I/O ports on the SDK-85. The processor is able to provide a binary value to the D.A.S. and through the use of a Digital-to-Analog converter (D/A) create an analog signal proportional to the input binary value. The D/A is used in the tone generation. The analog signal goes to a speaker that generates the tones. An Analog-to-Digital converter (A/D) enables the system to accept digital values corresponding to analog signals.

Voice synthesis is accomplished through the use of a speech synthesizer board based on the Votrax SC-01 phoneme Synthesizer. The SDK-85 communicates with the speech synthesizer through an output port. Phoneme codes are fed to the speech synthesizer and strobed. Details of this are covered in section 4.4.

The Telephone Adapter which consists of a speaker, a microphone and a relay is connected to J2 of the D.A.S. Only six conductors are required to allow for full use of the telephone adapter.

3.3 COMPONENTS :

3.3.1 S D K - 8 5

An enhanced version of the Intel System Design Kit-85 microprocessor board is being used in the implementation of the prototype. A complete manual on the operation and other details of the SDK-85 is available in Ref. 11.

Elements of the enhanced SDK-85 relevant to this project will now be briefly described.

- Processor: An Intel 8085A 8 bit parallel Central Processing Unit (CPU) with a 6.114 MHZ clock is the heart of SDK-85. Information regarding this processor may be obtained from Refs. 9 & 10.

- I/O ports: The Intel 8255A is a general purpose programmable I/O device with 24 bidirectional lines designed for use with Intel microprocessors. Two 8255s are available on the SDK-85. Data Sheets describing 8255 in detail are provided in Ref. 10.

- USART : Intel 8251, Universal Synchronus Asynchronus Receiver Transmitter allows SDK-85 to communicate with a terminal or other peripheral devices. 8251 is a very flexible USART, Baud rate and other communication characteristics may easily be modified. Detailed characteristics of this chip are described in Ref. 10.

- RAM : The enhanced SDK-85 uses four 2114s to provide two kilo bytes of Random Access Memory (RAM). Figure (A-1) is a diagram of RAM expansion.

- ROM : Four 2708s provide four kilo bytes of Programmable Read Only Memory (PROM). Figure (A-2) is a diagram of PROM expansion.

Table (3-1) presents the memory map and I/O ports of the enhanced SDK-85. This table only describes the elements related to this project. A more detailed version may be obtained from Ref. 12. Figure (A.3) is a schematic diagram of SDK-85 I/O. This diagram contains information about the USART and the two general purpose I/O ports.

SDK-85 communicates with peripheral devices through an 8251 USART. The USART is configured in software to operate at 4800 baud.

The two general purpose I/O ports, available on connectors J10 and J11, allow external devices to talk to the processor through an 8255 programmable peripheral interface. In this application J10 is connected to the speech synthesizer board and J11 is connected to the Data Acquisition System.

3.3.2 D. A. S.

Data Acquisition System (D.A.S.) consists of an A/D and a D/A. By using this device the processor is able to interact with the real analog world. Documents that fully describe the D.A.S. are available in Ref. 12. Figure (A.4) is a schematic diagram of D.A.S. The CD4051 is an analog multiplexer and 7407 is an open collector buffer. The A/D used is a Burr-Brown ADC80 and D/A used is a Burr-Brown DAC80. D/A takes an 8 bit input and the A/D has a 12 bit output.

J1 on the D.A.S. is connected to J11 on SDK-85. This allows an 8255 on SDK-85 to directly talk to the D.A.S. In this project D.A.S. is used for tone generation. Details are described in Chapter 4.

To make the D.A.S. completely suitable for this application, certain modifications were made. Details of this are described in Chapter 4.

TABLE (3.1)

MEMORY MAP AND I/O PORTS

MEMORY MAP		I/O PORT MAP	
ADDRESS IN HEX.	ELEMENT	PORT ADDRESS IN HEX	ELEMENT
0000-7FF	ROM MONITOR	80	USART DATA
2000-20FF	BASIC RAM	81	USART COM/STAT
8000-87FF	USER RAM	84	8255#1 PORTA
9000-9FFF	USER PROM	85	8255#1 PORTB
		86	8255#1 PORTC
		87	8255#1 CONTROL
		88	8255#2 PORTA
		89	8255#2 PORTB
		8A	8255#2 PORTC
		8B	8255#2 CONTROL

TABLE (3.1)

Table (3.1) : This table contains information related only to the elements used in this project. A more detailed version may be obtained from Ref. 11.

3.3.3 Speech Synthesizer Board.

The speech synthesizer board is fully described in section 4.4.1 of this document. Only a brief summary of its operation will be provided here.

The speech synthesizer board interfaces with the SDK-85. J10 of the SDK-85 is connected to J1 of the speech synthesizer board through an edge connector. The SDK-85 provides the phoneme codes to the speech synthesizer and strobes them. A phoneme code consists of 8 bits. 6 bits are used to represent one of the 64 phonemes and 2 bits are used to select one of the 4 possible pitch variations. A strobe line is used to start the sound generation process. The speech synthesizer internally times the duration of the phonemes for optimal voice quality and by using the Acknowledge/Request (A /R) line notifies the SDK-85 that it is time to sound the next phonemes. It is the responsibility of the SDK-85 to coordinate the delivery of the phoneme codes so that the desired word is properly pronounced.

3.3.4 Telephone Adapter

The telephone adapter consists of a speaker, a microphone and a relay. It is part of an old unoperational MODEM manufactured by Ford Data Inc. No data sheet on this MODEM is available. Ford Data has gone out of business. All of the information presented here comes from physically testing the device. Table (3.2) presents the results of these tests. Some modifications were made to make this device more suitable for the application. They are also shown in Table (3.2).

This telephone adapter is designed for use with the 500-type Bell telephones. A relay mechanically controls the telephone set. Experimentally I found that a 15 volts power supply is sufficient for the activation of this relay. A speaker that is mounted close to the telephone microphone enables the system to ”talk”. A microphone that is mounted close to the earphone of the telephone enables the system to ”hear”. The telephone adapter connects to J2 of the modified D.A.S.

TABLE (3.2)

THE TELEPHONE ADAPTER

ORIGINAL	FUNCTION	MODIFIED	
COLOR	FUNCTION	COLOR	CONFIGURATION
green	relay(-)	green	J2-13
green	speaker(-)	blue	J2-11
purple	speaker(+)	purple	J2-9
orange	relay(+)	orange	J2-3
black	microphone	black	J2-5
red	microphone	red	J2-7

Input resistance of the speaker is 189 Ohms.
Input resistance of the microphone is 1368 Ohms.
Input resistance of the relay is 38 Ohms.

Table (3.2)

Table (3.2) : Physical characteristics of the telephone adapter. Information provided in this table have been obtained by physically testing the device. No data sheets are available.

Chapter 4

IMPLEMENTATION

4.1 Method

4.2 Cradle Control

4.2.1 Hardware

4.2.2 Software

4.2.3 Problems and Possible Improvements

4.3 Dialing

4.3.1 Tones

Hardware

Software

Problems and Possible Improvements

4.3.2 Pulses

Hardware

Software

Problems and Possible Improvements

4.4 Speech Synthesis:

Much of my work on this section is based on article by Steve Ciarcia, available in Ref. 2. The phoneme synthesis technique is used to accomplish an unlimited-vocabulary speech synthesizer. Data sheets describing the SC-01 are available in Appendix C. A brief description of the Votrax SC-01 follows.

Votrax SC-01

The 22-pin Votrax SC-01 integrated circuit, diagrammed in Figure (4.2), contains a digital code translator, or phoneme controller, and an analog of the human vocal tract. The phoneme controller translates a 6 bit phoneme code and a 2 bit pitch code into a matrix of spectral parameters which in turn adjusts the vocal-tract analog to synthesize the phonemes.

In the first part of the vocal-tract section, there are a pair of variable frequency oscillators for simulating vocal-cord produced periodic sounds and a pseudorandom (pink-noise) signal generator that simulates the sound of rushing air. The output signals from these sources are shaped by a bank of four analog band-pass filters that simulate the vocal-tract cavities. The filter outputs, in turn, are directed through a preamplifier to an external amplifier and a speaker.

The SC-01 phoneme synthesizer is a CMOS (Complementary Metal-Oxide Semiconductor) integrated circuit which should be operated within the range of +7 to +14V (Vp). The phoneme-input lines (P0 through P5) are 5V level-compatable and self latching. The two pitch-control lines (I1&I2), on the other hand, must have external latches and must be switched at the same input voltage as the SC-01's power supply. Handshaking with external control circuitry is accomplished through two control lines: strobe(STB) and acknowledge/request (A /R). The STB can be either CMOS or 5 V level, while the A /R line is CMOS level only.

The output pitch of the phonemes is controlled by the frequency of the clock signal, which can be applied from an external source or set internally with a resistor and a capacitor combination. The clock frequency is nominally 720KHz. Two independent pitch control lines, I1 and I2, are available for gross variations in pitch so that the chip can speak with more than one voice.

Listed in Table 4.2 are the 64 phonemes defined for the English language(two produce silent periods of different length; one causes synthesis to stop). A phoneme sound is genertaed when a 6 bit phoneme code is placed on the control register input lines (P0 through P5) and latched by pulsing the strobe (STB) input. Each phoneme is internally timed and has a duration of 47 to 250 milliseconds; some phonemes last longer than others, and variations in the clock frequency affect the phoneme durations. The A /R line goes from a logic 1 to a logic 0 when a phoneme is sounding.

One method to use the SC-01 is to take advantage of the computer system to time the transmission of phoneme codes to the SC-01. This method sends codes to the synthesizer chip through a latched parallel output port and monitors the synthesizer's activities (via the A /R line) through an input port or interupt line. The advantage of this method over other methods described in the data sheet is that it eliminates the extra hardware and doesn't really complicate computer/synthesizer interaction. This is the method that I used.

4.4.1 Hardware:

The Speech Synthesizer Board

The schematic diagram of the speech synthesizer board is shown in Figure (4.3). The phoneme code bits are sent in parallel to the SC-01 (IC3) and buffered through IC1 (a 74LS244 three state octal buffer). Pull-up resistors assure that a logic-1 input to the SC-01 will be at least 4V as required. ENABLE input line allows the controlling circuitry to inhibit the speech synthesizer board. To enable the speech synthesizer board ENABLE should be at logic 0 or grounded.

The two manual-inflection inputs (I1 andI2) are also buffered through IC1. The SC-01 can not store these signals, and storage must be provided externally. A 74LS74 type-D flip-flop (IC2) is configured as a two bit latch. It is clocked synchronously with the SC-01's strobe input. Unlike the phoneme inputs, however, the inflection lines are not 5V compatible. Two sections of a 7416 open collector inverter (IC4) are used with pull up resistors to level shift these data inputs to CMOS levels.

The SC-01 can use either its internal clock or an external clock. External clock signals are applied through pin 15 on the SC-01 while pin 16 is grounded. The speech synthesizer board, on the other hand, uses the internal clock-signal generator. The clock frequency is determined by an R/C (resistor/capacitor) combination attached to pins 15 and 16. The frequency is adjusted through potentiometer R8 and nominally set for 720 KHz. Slight adjustments to this control will vary the pitch of speech. The most practical way to set this potentiometer is by ear.

The process of sounding a phoneme begins when the 6 bit phoneme code is latched into the SC-01's control register. Latching occurs on the rising edge of the positive going strobe pulse (STB line). The synthesizer will continue to sound the same phoneme until another phoneme code or a stop code is loaded.

The speech synthesizer board can accept either a normally high or a normally low stobe signal from the controlling device. The SC-01 senses the positive going edge of the strobe pulse. Unlike typical TTL latches which operate in a few nanoseconds, the SC-01 requires some setup time before it can accept the strobe signal. This set up time must meet two requirments:

- The data on the phoneme-input lines P0 through P5 must have been stable for 450 ns before the rising edge of the strobe pulse arrives.
- The logic level on the STB input of the SC-01 (pin 7) must have been low during at least 72 clock periods (approximately 100 microseconds before it goes high for the strobe pulse.

Approximately 500 nanoseconds after the rising edge of the strobe pulse, the A /R line of the SC-01 goes to a logic 0, indicating that the synthesizer chip is busy. Transistor Q1 and IC4 convert the CMOS output of pin 8 to TTL levels. The A /R output can be monitored by the controlling computer in either of two ways: directly through an input port or connected to an interrupt line. In either case, when the A /R line returns to logic-1 level, the SC-01 is ready to receive another phoneme code.

The remaining components on the speech synthesizer board make up the amplifier and filter sections. Capacitors C1 and C2 and resistors R5 and R6 form a simple low pass audio filter. The audio signal is then amplified by an LM386 amplifier (IC5) to drive a speaker. Potentiometer R7 controls the volume.

I wire-wrapped the circuit of Fig.(4.3) on a 16*12 cm wire-wrapping board. The board operates on power supply voltage of +5 and Vp=+12V. The power is applied through 3 female banana plugs. The speech synthesizer board interfaces with J10 of SDK-85 through a 12 conductor cable. Details of interfacing and pin configurations are available in Table (4.3).

4.4.2 Software:

The routine that delivers the phoneme codes to the speech synthesizer board and strobes them is SAY:SAY. Before entering this routine the sequence of phoneme-codes should be present in an array starting at PHONEME_CODE. A description of this follows.

```
PROCEDURE SAY ;

BEGIN

   enable SC-01 ;

   KEEP_LOW ;

   WHILE PHONEME_CODE<>LAST DO

   BEGIN

      get PHONEME_CODE ;

      mask I1 & I2 bits of PHONEME_CODE ;

      get VOICE and combine it with PHONEME_CODE ;

      put PHONEME_CODE on PORTB_2_8255 ;

      take the strobe line high ;

      KEEP_LOW ;

      WHILE (A~/R is low) DO

      BEGIN
```

TABLE (4.3)

Speech Synthesizer- SDK85 Interface

	J1		J10	
color	name	number	number	name
Green	P0	13	17	PB0
Green/White	P1	17	16	PB1
Green/Black	P2	14	15	PB2
Blue	P3	18	14	PB3
Blue/White	P4	15	4	PB4
Blue/Black	P5	19	3	PB5
Orange	I1	20	2	PB6
Orange/Black	I2	16	1	PB7
Red	ENABLE˜	12	19	PC0
Red/white	STB	21	6	PC1
Red/Black	A˜/R	22	21	PC4
Black	GND	11	13	GND

(TABLE 4.3)

Table (4.3):Illustrates how individual pin connections are made. J1 on Speech Synthesizer is connected to J10 on the SDK-85. J10 communicates with SDK-85 through 8255 #2. This chip is set to function in the following mode:

port A	input
port B	output
lower port C	output
higher port C	input

```
          read (A~/R);

          keep strobe line low

        END

     END;

     COMPLETE

  END;

                          * * * * *


    PROCEDURE KEEP_LOW ;

    BEGIN

      bring strobe line low ;

      delay for 100 micro seconds;

      (* so that the strobe signal can be detecetd *)

    END;

                        * * * * *


    PROCEDURE COMPLETE ;

    BEGIN

      put silence code on PORTB_2_8255 ;

      take the strobe line high ;

      KEEP_LOW ;

      WHILE (A~/R is low) DO

      BEGIN
```

```
        read A˜/R

      END;

      bring the enable line low

      (* this disables the speech synthesizer board *)

  END;
```

4.4.3 Problems and Possible Improvements

The speech synthesizer board is operational. Phonemes are correctly uttered. The major problem encountered was hand shaking between the speech synthesizer board and the SDK-85. The method described in the previous section took care of the problem.

The quality of the voice output is very dependent on the organization of the phoneme sequence passed to the speech synthesizer. Without much trouble I have been able to generate recognizable sentences. A limited table of phonemes that pronounce English words are available in Ref. 2. By properly sequencing the phonemes and using the pitch variation lines (I1 & I2) reasonably good quality voice may be generated. Since the quality of voice produced by the SC-01 is totally dependent on the organization of the phonemes, improvements on the quality of voice may only be achieved by devising schemes that find the phonemes corresponding to the desired utterances.

Computers normally manipulate information that is represented in the form of written text. It is desirable to create means of transforming text to its corresponding phonemes.

Text to Phoneme Transformation :

If English is treated as a basically nonphonetic language then the most natural approach for transforming text into phonemes is via a morpheme dictionary. Depending on the application this dictionary could be large or small. The disadvantage of this method is that it requires lots of memeory.

If English is treated as approximately a phonetic language in which phonetic strings are generated through some pronounciation rules, then the amount of memory required could be extensively reduced. Exception spelings are handled by an exception dictionary. If words are found in this dictionary, the phonemes are taken from the dictionary. Otherwise the pronounciation rules are used. The drawback of this method is that the quality of generated voice is dependent the pronounciation rules used, and to do a good job the programmer should also be a linguist.

The program may recognize individual words from the spacing between them and look them up in the vocabulary dictionary. If the desired word is found, corresponding phoneme codes are recorded and the program proceeds to the next word in the text. If the word does not exist in the vocabulary dictionary, then some primitive pronunciation rules may be used to utter the word so that the program can continue.

4.5 DTMF Decoding

I used a SSI 201 integrated tone receiver (equivalent of ITT MSD 3201) to detect the tones arriving on the voice channel. Data sheets describing the SSI 201 are available in Appendix C.

Figure (4.4) shows the internal structure of this device. After the 60-HZ-reject and band splitting filters, the 3201 uses eight band pass filters to detect the tones by analog means. The digital post-processor times the tone durations and provides the correctly coded digital outputs. Outputs interface directly to standard CMOS circuitry.

4.5.1 Hardawre

Figure (4.5) shows the schematic diagram of the circuitry required to make the DTMF decoder operational. The SSI 201 is confiugured to output a 4 bit binary value representing one of the 16 standard tones. The transistors are used to level shift the output so that they can interface to the 8255#1.

4.5.2 Software

When a tone is detected, DV (pin 18 of the SSI 201) goes high and the code value (A0-A3) is valid. The processor should monitor the DV line and read in the code value when it is valid. The routine that performs this task is HEAR:HEAR. Its logic will now be presented.

```
PROCEDURE HEAR ;

BEGIN

   key_detected := TRUE ;

   WHILE (it has not been 5 seconds) DO

   BEGIN

      read (data_valid);

      IF (data_valid is true) THEN

      BEGIN
```

```
            SEND_CODE ;

            key_detected := TRUE

         END ;

         IF (key_detected is false) THEN say so

      END

   END;
                       * * * * *


   PROCEDURE SEND_CODE ;

   BEGIN

      read in the code value;

      send this value to the host computer

   END ;
                     * * * * *
```

4.5.3 Problems and Possible Improvements

This task is not operational. The major problem encountered is the lack of time. I have not been able to correctly pickup DTMF audio signals from the telephone voice channel. Although not proven, I beleive that the method of implementation is valid. One possible improvement is to get it to work.

4.6 Ring Recognition

A sound switch is being used to detect the ring. Any loud sound close to the microphone can trigger the sound switch. Since the microphone is mounted on the telephone chassis, normally the telephone ring is the only event that can trigger the ring indicator.

4.6.1 Hardware

Figure (4.6) is a schematic diagram of the ring indicator. The LM308 is an amplifier configured to have a constant gain of 500. Th output of the microphone is only a few millivolts. The LM324 is configured as a comparator. By adjusting the 5K potentiometer, the reference voltage can be set. Normally the output of the LM 308 is less than the reference voltage and PA5 is low. When a call comes in, the output of LM308 momentarily becomes larger than the reference voltage and PA5 goes high. The processor is monitoring the activity of this line and detects the ring as soon as PA5 goes high for the first time.

4.6.2 Software

The output of the ring indicator circuitry is monitored. If a signal is detected a "RING DETECTED" message is sent to the host computer. The routine that performs this task is RING:RING. Its logic follows.

```
PROCEDURE RING ;

BEGIN

   WHILE (ring_line is low) DO   read ring_line ;

   write "RING DETECTED"

END;
```

4.6.3 Problems and Possible Improvements

This task is operational. Any movement of the telephone set will trigger the sound switch and cause a false detection of a ring. Since the ring is repeated in known time intervals, this characteristic may be used to eliminate any false detections. A monostable multivibrator and a flip flop can allow for this.

4.7 Control Structure

The routine that accepts the desired commands from the host computer and brings about the execution of these commands is being referred to as the "Control Structure". A list of the available commands is provided in Section 2.1.

The process involved in the implementation of Cradle Control, Dialing Speech Synthesis, DTMF Decoding and Ring Recognition have already been described. The Control Structure Simply finds out what function is to be performed, and calls the routine that performs the desired task.

4.7.1 Software

The input buffer is read and checked against available commands. CONTROL:CONTROL is an endless loop that performs the above mentioned function. REC_COM:REC_COM is the routine that recognizes the desired command and calls the appropriate routine. The logic follows.

```
PROGRAM CONTROL ;

BEGIN

   initialize the system ;

   WHILE TRUE DO

   BEGIN

      read the input ;

      REC_COM

   END

END.
                           * * * * *

PROCEDURE REC_COM (input_buffer) ;

BEGIN

   CASE  input_buffer  OF

      "LIFT" : RECEIVER_UP

      "DOWN" : RECEIVER_DOWN ;

      "HEAR" : HEAR ;

      "DIAL" : DIAL ;

      "TALK" : TALK ;

      "RING" : RING
```

```
    END ;

    ERROR

END;
```

* * * * *

```
PROCEDURE DIAL ;

BEGIN

   get next_letter of the input buffer ;

   CASE next_letter OF

      " ","T" : BEGIN

                   GET_NUMBER ;

                   DIAL_NUM_TONE

                END ;

      "P"  : BEGIN

                GET_NUMBER ;

                DIAL_PULSE

             END ;

   END ;

   ERROR

END ;
```

* * * *

```
PROCEDURE TALK  ;
```

```
BEGIN
   get next_letter of the input buffer ;
   CASE next_letter OF
      " " : TALK1 ;
      "1" : TALK1 ;
      "2" : TALK2 ;
      "3" : TALK3 ;
      "4" : TALK4
   END ;
   ERROR
END;
```

* * * * *

```
PROCEDURE TALK2 ;
BEGIN
   VOICE := VOICE_2 ;
   GET_PHONEME ;
   SAY
END ;
```

And the rest are similar.

* * * * *

GET_NUMBER and GET_PHONEME are the routines that transfer the data from input buffer to KEY_SEQUENCE and PHONEME_CODE arrays.

* * * * *

Chapter 5

APPLICATIONS

5.1 The Host Computer

The usefulness of any tool is a direct function of the ability and the knowledge of the user. This statement is universally true, and of course applies to the outcome of this project, the C.T.I. In this case the user is the program that is running on the host computer. If this program is properly designed, it will enable the computer to fully interact with the telephone set.

The word ”computer” does not have a concrete definition and applies to a variety of devices. If the computer has a random access memory on the order of 64k bytes and is mainly used by a hobbyist it is referred to as a ”home-computer.” If the computer has a primary memory on the order of several 100k bytes and is used to run small businesses and the like, it is referred to as a ”mini-computer” If the computer has a tremendous computational power, a large direct memory and is used to run major tasks like organizing the activities of an airline, then it is normally referred to as a ”main-frame” computer. Possible applications of the C.T.I. may best be discussed by categorizing the applications on the ability of the host computer.

5.2 Home_Computer Applications

The follownig may have been the dream of a computer hobbyist.

> ”I want to be able to telephone the computerized home-control system in my house from anywhere in the country, to find out what the conditions are like in and around the house, be informed of problems or messages, and remotely control light and thermostat settings.”

Well, C.T.I. can make this dream come true. The C.T.I. is able to recognize a ring and notify the computer that it is time to work. The computer will then ask the C.T.I.

to lift the receiver, send a greeting message and ask what is to be done. The user may then enter the code for the desired task on the C.T.I. The C.T.I. decodes the tone and passes the result to the computer which in turn activates the desired task.

In the above mentioned example, the dialing capability of the C.T.I. was not used. Here is one possible application for the dialing capability. A sorted file of last names and phone-numbers can replace the computer hobbyists phone-book. The user can dial phone-numbers by using the name of the person to be called. The running program will take the last name and search the phone-book file that treats names as the key and the phone-numbers as the data for the desired name. When the phone-number associated with the desired last name is found, the dialing capability of the C.T.I. will be used to make the phone call.

5.3 Mini-Computer Applications

An example for a possible application of the C.T.I. by a mini-computer may be an Automatic Telephone Interviewing System. The computer, with the aid of the C.T.I., will dial a series of phone-numbers from a given list. After introducing itself, the computer will ask if the person who is being interviewed is willing to continue and if the answer is positive (a certain key stroke may indicate this), continue the conversation through a prepared text. The computer will ask a series of questions wait for answers and guide the converstion acccording to the responses through its logic. When the interview is completed, the computer will thank the interviewee, and repeat this process for the next item on the phone list. After the last item in the phone list has been detected the computer will process and tabulate the gathered information and present the results to the user.

In the above mentioned example, the computer has to ”talk” a lot, and therefore good quality voice output is required. Since the dialogue of the conversation is static, implementation of high quality voice is practical.

5.4 Main-Frame Computer Applications

An enhanced version of the C.T.I. which has a good quality voice output is more suitable for use by a main-frame computer. What is being presented in this section is based on the concept of what the C.T.I. stands for, not the actual outcome of this project.

Problems like directory assistance, credit inquiries, bank balance inquiries and inventory control can comfortably be handled by the C.T.I. In all of these applications the basic premise is that the system is capable of accessing a data bank of information.

For example, suppose that the data bank was an inventory of the quantity of goods produced by a company and available sale and distribution. If the system was accessed by salesmen in the field (through the phone), then each time a sales was made the system could acknowledge the sale, and simultaneously update the inventory data bank. As more goods are manufactured by the company, the inventory could be externally updated as these goods become available for sale. In this example the system not only helps keep track of the inventory, it also prevents the possibility of several salesmen essentially selling the the same item when the inventory of goods is low. It also helps the company keep up-to-the-minute statistics on sales, allowing dynamic variations in the manufacturing of goods.

Another interesting application of the C.T.I. and a main-frame computer can be a Stock Price Quotation System. For this system the data bank is the current price of any stock, as well as the market price at the close of the preceding business day. The mechanism for externally updating the stock prices could an electronic data channel.

A typical scenario for the use of the Stock Price Quotation System is as follows. The user dials the system, which then responds:

> ”This is the Bell laboratories stock price quotation system. Prices are quoted as of the last business day. Please enter market abbreviation of the stock desired.”

```
The user keys in :

                    A-T-T-*
```

```
and the system responds: "American Telephone and Telegraph, 62- and
3/8, up 1/4"
```

Providing such information as stock market prices, without the need for cumbersome teletypes demonstrates the usefulness of what the C.T.I. is representing. Indeed it is clear that the future holds much promise for the wide spread use of this type of system.

Appendix A

Module Name: Ring

```
"8085"
;
;**********************************************************************
;
;
;   Module Name: RING
;   Date: 16/8/82
;   Programmer: Mohsen Banan
;
;   Purpose: This module performs the task of recognizing the ring and
;notifying the host cxomputer of such event. W hen the ring is detected
;(pin A5 ) goes high and the message "RING DETECTED" is transmitted.
;
;
;
;
;--------------------------------------------------------------------
;
               GLB       RING
;
               EXT       MESSAGE_2,WRITE_MES,PORTA_1_8255
;
               PROG
;
;REGISTERS AFFECTED: A,B,C.
;
;CALLS MADE: WRITE_MES
;
RING:
```

```
                IN        PORTA_1_8255  ;THE OUTPUT OF THE COMPARATOR
                ANI       00010000B     ;GOES TO PIN A5 OF 8255#1
                JZ        RING
                LXI       B,MESSAGE_2   ;THE ADDRESS  OF THE START OF THE MESSAGE SHOULD B
                CALL      WRITE_MES     ;B REGISTER BEFORE CALLING THE WRITE_MES ROUTINE.
                RET
;
                END
```