TITLE PAGE

IBM PC Voice Mail Card

By

Daniel A. Durbin

Senior Project

ELECTRONIC AND ELECTRICAL ENGINEERING DEPARTMENT

California Polytechnic State University

San Luis Obispo, CA

1989

APPROVAL PAGE

Title:	IBM PC Voice Mail Card
Author:	Daniel Durbin
Date Submitted:	March 15, 1989

Prof. Arthur Dickerson	Signature
Senior Project Advisor	

Dr. James D. Harris	Signature

SP.TXT

TABLE OF CONTENTS

APPENDICES

TABLES

FIGURES

iii
ABSTRACT

The Voice Mail Card (VMC) functions as an enhanced telephone answering machine and is designed as a plug in card for the IBM PC and compatibles. In addition to regular answering machine functions, the VMC features programmable outgoing message selection, response to caller's touch tone signals, and remote programming ability.

The function of the Voice Mail Card is to answer incoming telephone calls, deliver outgoing messages which are programmably selectable from 16 digitized audio messages which are stored on the PC's hard disk, to record incoming messages to the hard disk or optionally to an external cassette tape recorder, to respond to a caller's touch tone signals, and to enter a remote programming mode as a result of a special code sent by the caller.

Audio messages are processed digitally via A/D

and D/A converters which receive and send 8 bit data to and from the IBM PC through a selectable port address. A/D conversion is implemented with the ADC0802 which is operated at a clock rate of 512 Mhz.

At this frequency, the conversion rate of the ADC will be 8 khz yielding a bandwidth of 4 khz which is the limit of the telephone line. This same clock is used for D/A conversion which is implemented with the DAC0830. 8 bit data is read/written to/from the card via software and stored on the hard disk or optionally to an external cassette tape recorder.

Data is transferred to and from the card through I/O port address 0278h which may be changed as necessary by dip switch selection. A software program controls data read and write operations. A full featured pull-down windowing program is provided which controls all features of the VMC and operates in answer mode, program mode, message record, and message playback modes. A/D and D/A conversion, telephone line speech

interface, ring detection, and DTMF decoding are performed by pre-packaged monolithic devices. The entire circuitry resides on a 4-1/2" x 7" printed circuit board which plugs into an expansion slot in the IBM PC.

I. INTRODUCTION

A. Purpose

This project proposes an alternative to the standard telephone answering machine and will offer superior performance as a result of versatile programmability through the use of the IBM Personal Computer (IBM PC). The primary purpose of this project is to provide the designer with design experience interfacing to the IBM PC, interfacing to the telephone line, and digital-to-analog (D/A) and (A/D) analog-to-digital conversion techniques.

B. Problem

To implement this design, a method of recording outgoing messages (OGM), incoming messages (ICM), detecting phone rings, and audio coupling to the telephone line must be devised and implemented.

C. Solution

Messages will be digitized using an A/D converter and stored as 8 bit data on a hard disk. The bandwidth of the telephone line, 4 khz, will be adopted for the bandwidth of recorded messages. This requires a sample rate of 8 khz (by the Nyquist criteria). Therefore, the data rate will be 8 kbytes per second or 480 kbytes per minute. Since incoming messages can

be long and numerous, provision will be made for optionally storing incoming messages to an external cassette tape recorder. Audio will be reproduced using a D/A converter and played back to the telephone line and an external speaker. The digitized data will be read/written by the IBM PC through data buffers on the answering card. Audio will be recorded from a microphone and played back to a speaker. All circuitry to implement this will reside on a 4-1/2" by 7" pc board which will plug into an expansion slot in the IBM PC.

Wherever possible, circuit functions on pre-packaged monolithic devices are used to minimize power consumption, pc board space, cost, and scope of this project.

II. BACKGROUND

A. IBM PC Bus Interface

The type of IBM PC bus interface of interest in this project is 8 bit parallel data read and write operations. The simplest approach is to use port addresses and the port read/write instructions of the 8088 microprocessor. On the

IBM PC, port addresses are numbered from 0200h to 03FFh of which some are assigned by IBM to specific devices. Many of these addresses are unassigned and are available for purposes such as add-ons and expansion. The choice of address to use then depends only on which addresses are free in the particular system being considered. The second parallel printer port is chosen as the default port address, but for adaptability, may be changed by the appropriate selection of dip switch positions. The dip switch may be set to select any of the port addresses allowed by the IBM PC. With this configuration, 8 bit parallel data I/O may be accomplished with software's use of the 8088's IN and OUT instructions. In TurboC, these instructions are referenced with inportb() and outportb() functions.

Port addresses are determined by decoding address bits (A0 - A9, where bit A9 indicates a port is being addressed). A port address on the address bus is valid when address enable (AEN~) is lowered. 8 bit data is then transferred using

the port read (IOR~) and port write (IOW~)

signals. The IBM PC bus signals are tabulated

below in Table II-1.

GND	B1 A1	I/O CH CK~
RESET DRV	B2 A2	D7
+5V DC	B3 A3	D6
IRQ2	B4 A4	D5
-5V DC	B5 A5	D4
DRQ2	B6 A6	D3
-12V DC	B7 A7	D2
NOT USED	B8 A8	D1
+12V DC	B9 A9	D0
GND	B10 A10	I/O CH RDY
MEMW	B11 A11	AEN~
MEMW	B12 A12	A19
IOW~	B13 A13	A18
IOR~	B14 A14	A17
DACK3~	B15 A15	A16
DRQ3	B16 A16	A15
DACK1~	B17 A17	A14
DRQ1	B18 A18	A13
DACK0~	B19 A19	A12
CLK	B20 A20	A11
IRQ7	B21 A21	A10
IRQ6	B22 A22	A9
IRQ5	B23 A23	A8
IRQ4	B24 A24	A7
IRQ3	B25 A25	A6
DACK2~	B26 A26	A5
T/C	B27 A27	A4
ALE	B28 A28	A3
+5V DC	B29 A29	A2
OSC	B30 A30	A1
GND	B31 A31	A0

Table II-1.

B. Telephone Line Interface

1. Speech

The telephone company equipment registers an off-hook condition when a load which draws 20 ma is placed across the two phone lines, TIP and RING. The current is modulated and demodulated to send and receive speech signals through the lines. Factors involved in this process include line impedance balancing, gain control, and amplification. There are many manufacturers who produce telephone speech circuits in monolithic packages which perform these functions. To simplify the design, a choice of one of these circuits is made for the telephone speech interface.

2. DTMF

Dual Tone Multi-Frequency (DTMF) tones are used to represent Touch Tone digits. To decode these tones, a series of band pass filters must be used. Most practical are switched capacitor filters. This method and frequency measuring techniques are used by the M-957-01 DTMF Receiver from Teltone. This is the chip used for the purpose of DTMF decoding.

3. Ring

A ring signal consists of an 86 Vac signal which is on for 2 seconds and off for 4 seconds. The ring detector should present a high impedance to the line. Inexpensive monolithic packages are available to perform the function of ring detect and provide a signal which may be conditioned to deliver a TTL ring detect signal to the IBM PC.

C. Analog/Digital and Digital/Analog Conversion

National Semiconductor offers a wide variety of ADC's and DAC's. The choice of which depends on 3 criteria: bit resolution, conversion rate and interface. In this application, 8 bit data is required at a conversion rate of 8 khz. The data should be easily interfaced to a microprocessor. The ADC0802 and DAC0830 satisfy these requirements with flying colors.

III. REQUIREMENTS

In addition to the IBM PC Voice Mail Card, the following equipment is necessary for full operation:

* IBM PC XT or AT Compatible (10 Mhz max)
* MS DOS v3.0 or above
* 128k memory
* Hard disk (20 meg suggested)
* Telephone line
* External 8 Ohm speaker
* 600 Ohm dynamic microphone
* External tape recorder (optional)

The term "IBM Compatible" has been found to be misleading. Some PC systems claim to be able to run all software written for the IBM PC, but in actuality don't. The Voice Mail Card was designed to operate on an IBM PC XT and was tested on an IBM AT 10 Mhz Compatible using MS DOS 3.2.

The data hold time on the DAC determines the maximum CPU clock frequency of 10 Mhz. The size of the hard disk will determine the amount of digitized data that can be stored. Voice messages may be optionally saved in analog form

to an external tape drive which has a remote start/stop control input.

IV. DESIGN

The VMC consists of the following components:

A. Address Decode
B. Control/Status Registers
C. Data Bus
D. Voice Digitizer
E. Voice Reproduction
F. Audio Input Amplifier
G. Audio Output Amplifier
H. Telephone Interface
I. DTMF Decoder / Ring Detector
J. Software Program

A general block diagram showing the inter-connection of the above is shown in Figure IV-1. Audio may be digitized from two sources - from an external microphone, and from the telephone line. When DATA READY is asserted, the ADC has completed a conversion and the data will be placed on the data lines when the address decode circuit detects address 0278h on the address lines. The CPU must read the data within the sample period to avoid losing the next

conversion. Digitized audio may be written to the DAC by placing 0278h on the address lines and the desired data on the data lines. The CPU should time writes with DATA READY (8 khz). The DAC outputs to the AUDIO OUT amplifier which drives an external speaker and the telephone line.

A. Address Decode

The IBM PC system bus has 20 address lines (A0 - A19) of which only the lower 10 are used to decode I/O port or device addresses. The signals may be interpreted as shown below.

AEN A9	A8 A7 A6 A5 A4 A3	A2 A1 A0
port	port select	chip
Address		select
Valid		

Valid port address range is from 0200h to 03FFh. Some of these are used and others are not. The best approach for address selection is a built in dip switch allow to assignment different addresses depending on the particular PC configuration. The author choose the default to be the second printer port (LPT2) addresses 0278h - 027Fh.

The address decode circuitry monitors the IBM PC address lines (A0 - A9), address enable (AEN~) and port read/write (IOR~ and IOW~) and produces control signals which select operation of the card. This method is shown shown in Figure IV-2 and is described in detail.

Address lines (A3 - A8) and address enable (AEN~) are compared with the card address value in S1 (whose default is 0278h) by an 8 bit magnitude comparator which is enabled by the a port I/O read (IOR~) or port I/O write (IOW~) operation. This condition (XIOR + XIOW)~ is decoded by discrete gates and applied to the enable input of the comparator. The comparator then produces card select (CS~) which is used to enable two 1-of-8 decoders. The 1-of-8 decoders further decode address bits (A2 - A0) and are enabled by card I/O read and card I/O write (XIOR~ and XIOW~). The outputs then become register read (RA~ - RH~) and register write (WA~ - WH~) control signals which enable the

various 3-state data registers on the card. The assignment of the registers A - H is tabulated in Table IV-1.

Address	Register	Read	Write
0278h	A	ADC	DAC
0279h	B	-	-
027Ah	C	-	-
027Bh	D	-	-
027Ch	E	-	-
027Dh	F	-	-
027Eh	G	DTMF	-
027Fh	H	STAT	CTRL

Table IV-1.

B. Data Bus

The Voice Mail Card data bus consists of 8 bit data connected to the IBM PC data bus through a three-state buffer which is enabled by card select (CS~) and card I/O read (XIOR~). There are five 3-state devices connected to the Voice Mail Card data bus, the DAC, ADC, Status Register, Control Register, and the DTMF decoder. The status register monitors the state of the ADC clock (DATA READY), DTMF signal decode detect (DTMF), voice signal present

(VOICE), ring detect (RING), and phone off hook (HOOK) signals. The control register holds control signals for the tape drive and phone hook relay.

The Data Bus Schematic diagram is shown in Figure IV-3.

The status and control registers are assigned as register H. From the view of the computer, the status register is a read-only register and is enabled by read register H (RH~). The control register is a write-only register and is enabled by write register H (WH~). These are data registers which hold 8 bit card status and card control signals whose assignment is tabulated below in Table IV-2.

	Control Register	Status Register

BIT	Function	Function
MSB 7	-	-
6	-	-
5	-	-
4	-	-
3	-	DTMF-Detect
2	Mic-On	Voice-Detect
1	Tape-On	Ring-Detect
LSB 0	Off-Hook	Hook-Status

Table IV-2.

C. Voice Digitizer

The selection of an A/D converter is a compromise between price and availability. The ADC0802 was available and chosen. The ADC is clocked at 512 khz which is set by RC components. The conversion rate is then 512 khz / 64 clks/conversion = 8 khz. This will allow a frequency response of up to 4 khz which will cover the telephones bandwidth of 3.5 khz. The ADC is shown in Figure IV-3.

The bandwidth of the telephone line is 3.5 khz. For good results, the card will be designed for a 4 khz bandwidth. According to Nyquist, a sample rate of twice the highest frequency is required to reproduce the original signal. Therefore a sample rate of 8 khz is required. The National Semiconductor ADC0802 digital to analog converter is an 8 bit ADC with a maximum

conversion rate of 10 khz. It requires 64 clock cycles per conversion. Therefore, for an 8 khz conversion rate, a clock of 512 khz is required. This is obtained with RC components connected to the clock inputs of the ADC. The frequency can be given by:

$$f = \frac{1}{1.1\ RC}$$ Let R = 10 kohms,

$$C = \frac{1}{1.1(10k)(512k)} = 177.6\ pF$$

The ADC is set to operate in the free running mode by tying interrupt (INTR~) to write (WR~) which has the effect of initiating another conversion once the last conversion process is completed. The ADC is therefore always

performing conversions. The (INTR~) output will then be the clock frequency divided by the conversion rate and thus will be 8 khz. The state of the INTR~ output of the ADC is monitored by a D FF which is cleared when a read register A (RA~) or write register A (WA~) operation occurs. The output of the D FF is then a DATA READY signal which is monitored by the status register for the CPU to examine. In this manner, the CPU can time read and write operations to the 8 khz clock on the card. The ADC output data will appear on the card data bus when read register A (RA~) is lowered which is tied to read (RD~) on the ADC. This corresponds to address 0278h appearing on the IBM PC bus. The audio input should be a voltage from 0 to 5 Vdc and is produced by the Audio Input Amplifier circuit.

D. Voice Reproduction

The DAC chip to be used is the DAC0830. The hookup for this chip is straight forward. The chip is double buffered but in this configuration, this feature is not used. The feedback resistor is used in a current-to-voltage converter in the audio-out circuit. XFER and WR2 are held low while WA~ is used as chip select and XIOW~ clocks the data. Vref = +5 Vdc. The hold time of the data decreases as Vcc increases. For a 10 Mhz CPU, Vcc should be +12 Vdc. +5 Vdc is sufficient for 6 Mhz operation. The DAC is shown in Figure IV-3.

E. Audio Input Amplifier

The audio input amplifier consists of a high gain inverting amplifier, low pass filter, level shifter and activity detector. The sources of audio input signals are from an opto-isolator connected to the telephone interface circuit and a microphone whose impedance is 600 ohms and output signal is about 5 mV at a normal speaking volume and a speaking distance of 3" from the microphone. The input to the ADC0802 should be from 0 to 5 Vdc centered at 2.5 Vdc. An inverting amplifier is used to provide a gain of

5 V / 5 mV = 1000.

The audio signal should also be filtered to contain minimum frequency components above 4 khz. A Sallen and Key two pole low pass filter to provide low pass filtering. The gain and poles are given by:

Av = 1 + R2/R1 f = 1 / 2 pi R C

The gain is set to slightly above one and the poles to 4 khz by selecting R1 = 10 kohms and R2 = 27 kohms. Thus R2 = 12 kohms and C = 1500 pF. The signal is then capacitively coupled to a level shifter whose center is 2.5 Vdc and limited by protection diodes to prevent values greater than 5 Vdc and less than 0 Vdc.

A single-shot retriggerable multivibrator is used as the activity detector and is set to 3 seconds by the timing components. The high time is given by

tw = k Rext Cext, where k ~= 0.45.

Choosing the highest value of Rext yields Cext = 22 uF. The one-shot is connected to trigger on the falling edge of the input signal. Since the

zero or "quiet" value of the audio out signal is 2.5V, the one-shot will trigger when the input falls below 0.8V.

F. Audio Output Amplifier

The Audio Output Circuit is shown in Figure IV-4 and consists of a current-to-voltage converter (since the output of the DAC is a current), a Sallen and Key low pass filter, and an off-the-shelf audio amplifier.

The digitized audio output of the DAC is converted to a voltage by the DAC's internal 15 kohm resistor and the inverting OP-AMP U16. The signal is then low pass filtered by a low pass Sallen and Key filter whose high frequency is set to 4 khz. This removes the discrete digital steps produced by the DAC by smoothing them into an audio signal.

The output of this low pass filter, along with a sample of the audio input low pass filter, is sent to an audio power amp (U17) through a gain control (R32) which acts as a volume control. The output of the power amp is then sent to an external 8 ohm speaker and to an external

cassette tape recorder if available.

Samples of both the audio output and audio inputs circuits are send to the Telephone Interface circuit for coupling to the telephone line.

G. Telephone Interface

The TCM-5700 monolithic speech circuit is used to interface voice audio to the phone line. Optosolators are used to isolate the IBM PC power from the telephone line. The Telephone Interface Schematic Diagram is shown in Figure IV-5.

The phone line is taken off hook when the software address the OFF HOOK bit in the control register. The bridge rectifier BR1 converts the telephone line's AC signal to DC for power and use by the TP5700 speech circuit which is biased and balanced with reference to its data sheet. Since the telephone line must be isolated from the IBM PC power and ground, opto-isolators are used to couple the audio signals to and from the audio circuits. Power from the TP5700 is used to

bias the receive LED and transmit transistor and IBM PC +5 Vdc to bias the transmit LED and receive transistor in the dual opto-isolator.

H. DTMF Decoder / Ring Detector

The DTMF decoder schematic is shown in Figure IV-6 and the Ring Detector in Figure IV-7. The DTMF decoder monitors the received audio from the telephone interface circuit. When the DTMF decoder detects a DTMF signal on the telephone line, it will assert DTMF DETECT which may be monitored by software by reading the status register. If the computer responds by addressing the DTMF decoder, the DTMF decoder will place onto the data bus (XD0-XD3) a 4 bit binary number equivalent to the touch tone signal present on the audio line.

The ring detector is connected directly to the phone line through a high impedance RC circuit.

Since the ring detector is powered by the telephone line power, its output must be isolated from the IBM PC. Opto-isolator U23 provides this isolation and produces RING which is then sent to the status register.

I. Software Program

The software program used to operate the voice mail card is written in TurboC v1.5 and controls the reading and writing of digitized speech to and from the card. It also monitors the voice mail card status and takes appropriate action such and picking up the phone in response to a ring, and other control functions. The source code for the program is many thousand of lines long so will not be listed here, however, a partial listing of of the program containing the readADC() and writeDAC() functions is provided

as Appendix G.

The program (called ANSWER.EXE) is a window driven program which provides interface to all the features of the voice mail card. The user may record outgoing messages, play back incoming messages, manually dial a number, program the out going message timer and anything else.

V. DEVELOPMENT AND CONSTRUCTION

The prototype for the VMC was built on a prototype board during design and test. A layout of the proposed circuit board is provided in the appendices. The layout does not include switching capacitors for the logic IC's which should be added. Time did not permit fabrication of the pc board, however, it is the author's intent to construct the pc board version of this project.

VI. TEST RESULTS

A. ADC Test

The first stage to be tested was the ADC. The readADC() function was written to read and store digitized audio from the ADC. A graphing function was written to display the digitized audio on the computer's monitor in high

resolution graphics mode.

B. Audio End-around Test

The audio input circuit was connected directly to the audio output circuit and a frequency response plot was made of the two circuits combined. The combined circuits make up a four

pole band-pass filter with band of 100 hz to 4000 hz. The resultant frequency response data is tabulated in Table V-1 and the corresponding frequency plot is shown in Figure V-1.

R = 12k / (150k + 12k)

Hz	Vi x R	Vo	Vi	dB
70	52.31	0.22	3.875	35.0
100	52.40	0.40	3.881	40.3
200	52.42	0.65	3.883	44.5
400	52.03	0.82	3.854	47.0
700	51.86	0.91	3.841	47.5
1.5k	51.68	1.05	3.828	48.8
1k	51.74	0.96	3.833	48.0
2k	51.72	1.20	3.831	49.9
3k	52.42	1.20	3.883	49.8
3.5k	52.78	0.99	3.910	48.1
4k	52.84	0.78	3.910	46.0
5k	52.51	0.41	3.890	40.5
6k	52.28	0.20	3.873	34.3
7k	52.21	0.10	3.867	28.3

Frequency response raw and calculated data

Table V-1.

C. DAC Test

The second stage to be tested was the DAC. The

writeDAC() function was written to send stored digitized audio to the DAC. A speaker was connected to the audio output circuit and the resulting reproduction of audio was monitored audibly.

D. Telephone Interface

The third stage to be tested was the Telephone interface. A second phone line was used to dial the phone to which the interface was connected. The readADC() function was used to store the digitized audio received from the phone line. The writeDAC() was used to replay the digitized audio.

E. Answering Mode Test

The fourth stage to be tested was the integrated answering mode. For this test, the program was used to monitor the ring signal and respond with an OFF_HOOK command. An outgoing message was then played to the caller. Once finished, the incoming message was recorded until the VOICE signal was no longer asserted. The resultant recorded incoming message was played back through the external speaker.

F. Integrated Test

The bulk of the user interface of the software program was developed during this phase of testing. The readADC() and writeDAC functions were integrated into a pull-down windowing program and support functions were written to provide the programmability of the VMC.

VII. CONCLUSION

This project was successful. It was ahead of schedule, under budget, exceeded specifications, and performed outstandingly.

A. Bugs and Suggestions

The readADC() functions writes data to the hard disk in 1 second blocks (4 kbytes) which takes different amounts of disk access times with different types of hard disks. With a 10 Mhz AT and a 35 ms access time ST-251 hard disk, the data write time takes more than the 1/4000th of a second and results in data lost. A solution would be to provide a data buffering RAM on the card.

VII. BIBLIOGRAPHY

A. Interfacing to the IBM PC
by Lewis C. Eggebrecht. Howard W. Sams.

B. Introduction to Electronic Speech Synthesis
by Neil Sclater Blacksburg Howard W. Sams

C. Understanding Telephone Electronics
by John L. Fike. Howard W. Sams

REFERENCES

A. Motorola Schottky TTL Data Book

B. National Semiconductor Linear Databook

C. TelTone Corporation DTMF Receiver
M-957-01 Data Sheet

APPENDICES

Appendix A. Specifications

Conversion rate	8	Kbytes/sec
A/D Converter clock rate	512	Hz
Data Rate	480	Kbytes/minute
External Speaker		
impedence	8	ohms
power	2	watts max
Microphone	600	ohms
Audio Gain	48	dB @ 1 Khz

Bandwidth	4	Khz
Phone line	RJ-11	

Appendix B. Parts List

Ref Des	Description	Part #	Spec	Tol	U/I	Price
BR1	Full-Wave Bridge Rectifier	BR84D-ND	2a, 400v		EA	0.77
C1	Capacitor, Electrolytic	2.2u	50V	20%	EA	0.13
C2	Capacitor, Polypropylene	1500p		2%	EA	0.39
C3	Capacitor, Polypropylene	1500p		2%	EA	0.39
C4	Capacitor, Electrolytic	2.2u	50V	20%	EA	0.13
C5	Capacitor, Polyprobylene	4700p		2%	EA	0.39
C6	Capacitor, Polyprobylene	4700p		2%	EA	0.39
C7	Capacitor, Electolytic	2.2u	50V	20%	EA	0.13
C8	Capacitor, Mylar	0.047u	100V	10%	EA	0.09
C9	Capacitor, Electrolytic	22u	20V	20%	EA	0.13
C10	Capacitor, Metallized Polyester	0.01u	400V	10%	EA	0.16
C11	Capacitor, Mylar	0.01u	100V	10%	>10	0.09
C12	Capacitor, Mylar	0.047u	100V	10%	>10	0.09
C13	Capacitor, Electrolytic, Radial	100u	16V	20%	EA	0.45
C14	Capacitor, Mylar	0.22u	100V	20%	>10	0.19
C15	Capacitor, Electrolytic, Tantalu	4.7u	50V	20%	EA	0.14
C16	Capacitor, Mylar	0.1u	100V	20%	>10	0.11
C17	Capacitor, Tantalum	10u	16V	20%	EA	0.13
C18	Capacitor, Metallized Polyester	0.47u	250V	10%	EA	0.36
C19	Capacitor, Electrolytic, Tantalu	10u	50V	20%	EA	0.16
C20	Capacitor, Mica	180p		5%	EA	0.15
C21	Capacitor, Electrolytic	470u	16V	20%	EA	0.38
C22	Capacitor, Electrolytic	2.2u	50V	20%	EA	0.13
C23-C33	Capacitor, Mylar	0.1u	100V	10%	>10	0.11
D1	Switching Diode	1N4148 or eq	Low I, Vf<.8v		>10	0.05

Ref Des	Description	Part #	Spec	Tol	U/I	Price
D2	Switching Diode	1N4148 or eq	Low I, Vf<.8v		>10	0.05
J1, J2	Connector, Phone	RJ-11 DIP			EA	3.72
J3	Connector, Miniature	SJ-441	1/8", 2 cond,		EA	0.53
J4	Connector, Miniature	SJ-441	1/8", 2 cond,		EA	0.53
J5	Connector, Miniature	SJ-441	1/8", 2 cond,		EA	0.53
J6	Connector Subminiture	SJ-M24	1/10		EA	0.39
Q1	Transistor, General Purpose Swit	2N3904	NPN		EA	0.23

Ref Des	Description	Part #	Spec	Tol	U/I	Price
R1	Resistor, Carbon	560	1/4 WATT	5%	200	0.02
R2	Resistor, Carbon	560K	1/4 WATT	5%	200	0.02
R3	Resistor, Carbon	27K	1/4 WATT	5%	200	0.02
R4	Resistor, Carbon	27K	1/4 WATT	5%	200	0.02
R5	Resistor, Carbon	12K	1/4 WATT	5%	200	0.02
R6	Resistor, Carbon	10K	1/4 WATT	5%	200	0.02
R7	Resistor, Carbon	10K	1/4 WATT	5%	200	0.02
R8	Resistor, Carbon	10K	1/4 WATT	5%	200	0.02
R9	Resistor, Carbon	8.2K CF25	1/4 WATT	5%	200	0.02
R10	Resistor, Carbon	8.2K	1/4 WATT	5%	200	0.02
R11	Resistor, Carbon	12K	1/4 WATT	5%	200	0.02
R12	Resistor, Carbon	18K	1/4 WATT	5%	200	0.02
R13	Resistor, Carbon	10	1/4 WATT	5%	200	0.02
R14	Resistor, Carbon	220K	1/4 WATT	5%	200	0.02
R15	Resistor, Carbon	18K	1/4 WATT	5%	200	0.02
R16	Resistor, Carbon	10	1/4 WATT	5%	200	0.02
R17	Resistor, Carbon	620	1/4 WATT	5%	200	0.02
R18	Resistor, Carbon	1.5K	1/4 WATT	5%	200	0.02
R19	Resistor, Carbon	4.7K	1/4 WATT	5%	200	0.02
R20	Resistor, Carbon	56 RSF1A	1 WATT	5%	100	0.12
R21	Resistor, Carbon	1K	1/4 WATT	5%	200	0.02
R22	Resistor, Carbon	8.2K	1/4 WATT	5%	200	0.02
R23	Resistor, Carbon	5.6K	1/4 WATT	5%	200	0.02
R24	Resistor, Carbon	150	1/4 WATT	5%	200	0.02
R25	Resistor, Carbon	15K	1/4 WATT	5%	200	0.02
R26	Resistor, Carbon	820	1/4 WATT	5%	200	0.02
R27	Resistor, Carbon	2.2K	1/4 WATT	5%	200	0.02
R28	Resistor, Carbon	470	1/4 WATT	5%	200	0.02
R29	Resistor, Carbon	4.7K	1/4 WATT	5%	200	0.02

R30	Resistor, Carbon	1M	1/4 WATT	5%	200	0.02
R31	Resistor, Carbon	10K	1/4 WATT	5%	200	0.02
R32	Potentiometer, 15 turn	3006P	10K	10%	EA	1.35

Ref Des	Description	Part #	Spec	Tol	U/I	Price
U1	Octal Bus Transceiver, Non-inver	74LS245			>10	0.69
U2	A/D Converter, uP compatible	ADC0803LCN		1/2 L	EA	4.95
U3	Octal Transparent Latch, 3-state	74LS373			>10	0.79
U4	Octal Transparent Latch, 3-state	74LS373			>10	0.79
U5	D/A Converter, uP compatible	DAC0830LCN			EA	4.75
U6	Quad 2-in NAND	74S00			EA	0.25
U7	Quad Buffer, Low enable, 3-state	74LS125A			>10	0.39
U8	Dual D Flip-Flop	74LS74A			>10	0.25
U9	8 Bit Magnitude Comparator	74LS688N			EA	1.93
U10	8.2 Kohm Resistor Pack DIP	4116R-001-RC1/4 WATT		10%	EA	0.59
U11	8 Switch DIP Switch	16 pin DIP			>10	1.05
U12	1 of 8 Decoder	74LS138			>10	0.39
U13	1 of 8 Decoder	74LS138			>10	0.39
U14	DTMF Receiver	M-957-01 Teltone			dummy	4.00
U15	Quad 2-in NAND	74S00			EA	0.25
U16	Wide Band Input JFET Op-Amp	LF347N			EA	1.49
U17	Low Voltage Audio Power Amplifie	LM386N-1			EA	0.79
U18	DPST Relay	PRMA2A05	5V		EA	4.65
U19	Retriggerable Monostable Multivi	74LS123			>10	0.39
U20	Opto-Isolator, Dual	MCT61GI	If = 50%		EA	2.50
U21	Telephone Speech Circuit	TP5700AN Nat Semi			dummy	2.00
U22	Ring Detector	TCM1520APTI			dummy	0.75
U23	Opto-Isolator	H11A2	If = 20		EA	0.75
U24	Quad Bilateral Switch	CD4016AE			EA	0.29
X1	3.58 MHz Crystal Series	CY3.57	AP18		>10	0.89
ZD1	Zener Diode	1N4746A	18V 100PR	5%	EA	0.25

Appendix F. Program listing

```
-----------BEGIN ANSWER.H ------------
/***********************************************
 * File: Answer.h
 * Desc: Include file for the Answering card
 * Date: 12/03/88
 ***********************************************/
/* FILE ADCDAC.C: */

/*  DOS interrupt vectors */

#define Keyboard        0x09            /* keyboard interrupt vector */
#define Timer           0x21
#define D8259           0x20
#define EOI             0x20

/*  Answering card addresses and codes */

#define AnsDAC          0x0278          /* Digital to Analog conv address */
#define AnsADC          0x0278          /* Analog to Digital conv address */

#define AnsStatReg      0x027f          /* Answering Card Status Register */
#define DataReady       0x01            /* Data Ready bit */
#define TouchTone       0x08            /* Strobe from DTMF decoder */
#define Audio           0x20            /* voice present */
#define NotRing         0x40            /* ring */
#define Clock           0x80            /* Clock bit */

#define AnsCtrlReg      0x027f          /* Answering Card Control Register */

#define OffHook         0x01            /* Takes phone off hook */
#define OnHook          0x00            /* Places phone on hook */
#define Mute            0x02            /* Halves receive gain */

#define AnsDTMF         0x027e          /* DTMF decoder */
#define DTMF            0xf0            /* */
```

```
#define Fclock          8000.0          /* 8 Khz conversion rate */

void interrupt keyboard_handler(void);
void read_adc(char *filename, unsigned char limit); /* limit in seconds */
char write_dac(char *filename);
void mask_timer(void);
void unmask_timer(void);
int read_dtmf(void);

-----------END ANSWER.H --------------

---------- BEGIN ADCDAC.C ------------

/***********************************************
 * File: AdcDac.c
 * Desc: Read and write drivers for the ADC and
 *       the DAC at address 0278
 * Date: 12/03/88
 ***********************************************/

#define MENU 1

#include <dos.h>
#include <stdlib.h>
#include <conio.h>
#include <string.h>
#include <io.h>
#include <stdio.h>

#include "answer.h"

#undef  BUFSIZ    /* increase the disk io buffer */
#define BUFSIZ  4096

void fopenerr(char *filename);
void interrupt (*old_keyboard)(void);
char _kbhit;
```

```
/***********************************************
 * Function: Read_ADC()
 * Desc: Reads bytes from the A/D conv at 0x0278h
 ***********************************************/

void read_adc(char *filename, unsigned char limit)
{
    FILE *stream;
    unsigned long size, i;
    char buf[BUFSIZ];
    char ch;

    size = (long)limit * 8000;  /* 8,000 bps */
    if (!(stream = fopen(filename, "wb"))) {
        fopenerr(filename);
        return;
    }
    setvbuf(stream,buf,_IOFBF,4096);
    old_keyboard = getvect(Keyboard);
    setvect(Keyboard, keyboard_handler);
    delay(100);
    if (kbhit())
        getch();
    _kbhit = 0;
    i = 0;
    mask_timer();
    ch = inportb(AnsStatReg);
    while ( !(ch & Audio) && !_kbhit)
        ch = inportb(AnsStatReg);
    while ( (ch & Audio) && !_kbhit) {
        ch = inportb(AnsStatReg);
        if (ch & Clock) {
            fputc(inportb(AnsADC), stream);
            if (i++ > size)
                _kbhit++;
        }
```

```
        if (ch & TouchTone) {
            read_dtmf();
            _kbhit++;
        }
    }
    fclose(stream);
    unmask_timer();
    setvect(Keyboard, old_keyboard);
    if (kbhit())
        getch();
    return;
}
/***********************************************
 * Function: Write_DAC()
 * Desc: Writes bytes to the D/A converter 0x0278h
 ***********************************************/

char write_dac(char *filename)
{
    char buf[BUFSIZ];
    long j;
    long size;
    FILE *stream;
    char dtmf;
    char ch;

    if (!(stream = fopen(filename, "rb"))) {
        fopenerr(filename);
        return(0);
    }
    setvbuf(stream,buf,_IOFBF,4096);
    old_keyboard = getvect(Keyboard);
    setvect(Keyboard, keyboard_handler);
    delay(100);
    if (kbhit())
        getch();
    size = filelength(fileno(stream));
    j = 0;
    _kbhit = 0;
    dtmf = 0;
    mask_timer();
    while (!_kbhit && (j < size)) {
        ch = inportb(AnsStatReg);
        if (ch & Clock) {
            outportb(AnsDAC, fgetc(stream));
            j++;
        }
        if (ch & TouchTone) {
```

```
            dtmf = read_dtmf();
            _kbhit++;
        }
    }
    fclose(stream);
    unmask_timer();
    setvect(Keyboard, old_keyboard);
    if (kbhit())
        getch();
    return(dtmf);
}

/*-------------------*/
/*  F O P E N E R R  */
/*-------------------*/

#ifndef MENU
void fopenerr(char *filename)
{
    printf("Cannot open %s\n", filename);
}
#endif

/*---------------------*/
/*  R E A D _ D T M F  */
/*---------------------*/

int read_dtmf(void)
{
    char dtmf;

    dtmf = 0;
    if (kbhit()) getch();
    while (!dtmf && !kbhit())
        dtmf = (inportb(AnsDTMF) & DTMF) >> 4;
    return(dtmf);
}
/*
 *    Check for keyboard press
 */
void interrupt keyboard_handler(void)
{
    (*old_keyboard)();
    _kbhit++;
}
/*
 *  Mask Timer
 */
```

```
void mask_timer(void)
{
    outportb(Timer, inportb(Timer) | 0x01);
}
void unmask_timer(void)
{
    outportb(Timer, inportb(Timer) & 0xfe);
    outportb(D8259, EOI);
}
#undef     BUFSIZ
#define    BUFSIZ 512    /* set back to normal */
----------------- END OF ADCDAC.C -----------------
```