

---

# DX-1 'ECHO'

---

ECHO AND REAL TIME
SOUND PROCESSING PROGRAM

Decillionix

**NOTICE**

Decillionix reserves the right to make improvements in the product described in this manual at any time and without notice.

**DISCLAIMER OF ALL WARRANTIES AND LIABILITIES**

Decillionix makes no warranties, either express or implied, with respect to this manual or with respect to the software described in this manual, its quality, performance, or fitness for any particular purpose. Decillionix will not be liable for direct, indirect, incidental, or consequential damages resulting from the use of the product described in this manual.

This manual is copyrighted. All rights are reserved. This document may not, in whole or part, be copied, photocopied, reproduced, translated or reduced to any electronic medium or machine readable form without prior consent, in writing, from Decillionix.

The word Apple is a registered trademark of Apple Computer.

Written by Paul Swearingen in conjunction with Decillionix.

DX-1 'ECHO' Rev 1.0

---

**DX-1 'ECHO'  
AND SOUND PROCESSING PROGRAM**

---

TABLE OF CONTENTS

<b>PREFACE</b>	5
<b>INTRODUCTION</b>	7
<b>GETTING STARTED</b>	8
<b>WHAT IS ECHO ?</b>	9
<b>WHAT IS DX-1 'ECHO' ?</b>	10
<b>WHAT ARE THE OUTER, MIDDLE, AND INNER LOOPS ?</b>	12
<b>WHAT DO THE KEYS DO ?</b>	14

**SPECIAL FUNCTION CHARACTERS** 14

<u>APPLE II</u>	<u>APPLE //e or APPLE ///</u>	
ESC	ESC	14
RETURN	RETURN	14
ARROWS	ARROWS	15
SPACE BAR	SPACE BAR	15
/ or ;	/ or ;	15
? or +	? or :	16
, or .	, or .	16
* or =	_ or +	17
: or -	- or =	17
< or >	< or >	17
@	]	18

**NUM and PUN Presets** 20

! thru )	20
l thru 0	20

**LETTERS AND CONTROL CHARACTERS** 21

## PREFACE

You, the user of this program, will find that DX-1 'ECHO' provides the opportunity for you to discover for yourself many different types of sounds that can be created. It would take away from the fun of it for anyone but you to try to classify each of the categories of sounds that can be generated. This manual is written with your best interest in mind and in such a way that only the information necessary to start you on your journey is provided, hopefully in a clear and concise way.

Also enclosed is a quick reference card that is designed to be a summary of all pertinent information in this manual. From my own personal experience with other programs I tend to appreciate and use the quick reference card far more than the manual that came with the program (when it was appropriate for the author to include one). So the quick reference card is designed to be quite thorough.

This exact kind of thing has never been done before on personal computers and my sense is that the domain of possibilities that real time sound processing opens up is unlimited.

Since DX-1 'ECHO' or anything like it has never existed before chances are it will appear to be quite overwhelming on the onset. Actually the routines themselves are quite simple. It's just that when you start to play with sound it doesn't take much to come up with millions of variations. What really happens is that the sound samples themselves (occurring thousands of times a second) become variables that dynamically change to produce unlimited sound qualities.

Most of the time it took to design and write this program was directed toward refining the human to computer interface so as to give you the maximum control with the minimum of effort. This process involved juggling tradeoffs between simplicity, flexibility, speed, accuracy, and program size just to name a few. I welcome and would certainly appreciate any suggestions you may have in increasing the quality and performance of the program or documentation .

Contact:

Paul Swearingen  
3448 Hickerson Drive  
San Jose, Ca. 95127

(408) 258-0233

## INTRODUCTION

DX-1 'ECHO' is a program that runs on an Apple computer that provides you with the opportunity to generate echo and a multitude of different and interesting real time sound effects. In a nutshell this is what the program is with the inclusion of one fringe benefit -- you can also view the text and graphics screens while sounds are being generated in them.

As it turns out, the Apple computer is an ideal choice of a computer to use to bring this new sound processing technology into the hands of thousands of people that could not afford expensive sound processing systems. This program is not meant to be used as a studio quality professional system. I suspect people will create many obscure serious uses for DX-1 'ECHO' but for the most part this program is intended to be used just for your personal creativity, entertainment, and fun.

## GETTING STARTED

The DX-1 'ECHO' package that you are using consists of the following items:

1. Program diskette
2. Feedback Control
3. Quick Reference Card
4. This Manual

Plug the DX-1 circuit board into any one of slots 1 to 4 of the Apple II, Apple //e, or Apple /// Computer. Appropriately connect the microphone and speaker (or any other configuration that you have found to work). Plug the Feedback control into the back connector of the DX-1 circuit board (it should only go in one way). Boot the diskette and select the appropriate slot # (press a Return after pressing 1 to 4). Note: This program will only run in Emulation Mode on the Apple ///. Also, the Apple /// will require a special joystick board. You should now be ready to run the program and adjust the input volume. Follow the instructions you see when the program finishes booting.

## WHAT IS ECHO?

Echo is the repeating of a sound. Often tape recorders are used to produce echo. The tape recorder will make an excellent analogy because the software of DX-1 'ECHO' is set up in much the same way as a tape recorder when configured for echo. In most quality tape recorders there are at least two heads, one for recording and one for playback. It is extremely simple to produce echo with a tape recorder and the concept is also simple to grasp.

In the tape recorder the tape moves across first the record head and then the playback head. An audio signal is connected to the record head and in effect is recorded onto the tape as it moves by. After a short duration of time the recorded signal on the tape will cross the path of the playback head. This duration depends upon the distance between the heads and the tape speed.

Normally, people record sounds and then play them back later. This is not the case when the tape recorder is configured for echo. During echo, recording and playback from the two different heads occurs simultaneously. A portion of the signal coming out of the playback head is mixed back in with the original signal that is being recorded. This is an important point and it is what produces the actual echo in the first place.

The quantity of signal mixed back into into the record head along with the present sound being produced determines the echo intensity and how fast it dies out from a loud sound or builds up from a soft sound. There you have it. The ingredients are:

1. Record and playback heads
2. Combination of the distance between the heads and the tape speed that determines the delay
3. Feedback from the playback head mixed with the source input and inserted back into the record head

## WHAT IS DX-1 'ECHO' ?

With DX-1 'ECHO' the same effect is produced only the sounds are recorded into the Apple memory using an A-to-D (Analog to Digital) converter and played back using a D-to-A (Digital to Analog) converter. In this process sound samples are digitized one by one and are stored into the Apple memory at a speed called the sample rate or simply "SPEED".

In the program the slowest speeds are produced using a large speed number and the fastest speeds are produced using a small speed number. Sounds recorded at a slow speed and played back at a fast speed will sound very high pitched and the opposite is true. Another aspect of the echo that

can be changed is the "LENGTH". This would be like moving the tape heads closer and farther apart on the tape recorder. A small length is represented by a small number and a large length is represented by a large number.

At this time it is best to set the tape recorder analogy aside and take a look at what is actually happening in the program. The echo memory can effectively be represented by a circle. DX-1 'ECHO' gives you the capability to record into and play back from this circle. You can control the speed, length, and beginning location at which the echo is occurring on the circle. Each echo selection from the keyboard provides different variations of speed, length, and beginning locations -- all occurring within this central theme.

Also, you can turn the record process off at any time so you hear only what happens to be in memory at the time over and over again. Plus, you can record and playback in forward or reverse direction -- or alternate directions each time through the main echo loop.

A certain portion of sound signal can be fed back into the input from the output by using the hand controller. This is called feedback so you can refer to this as being the feedback control. This can only occur when you are recording.

#### WHAT ARE THE OUTER, MIDDLE, AND INNER LOOPS ?

First of all, a loop is a section of a program that gets repeated. In creating echo not only does the program loop but the memory space that is being written into and read from also loops.

The outer loop is the outermost echo loop in any given echo selection. The outer loop contains 1 to 146 middle loops. The quantity of middle loops corresponds to the length value. Each middle loop consists of 256 inner loops.

Each inner loop consists of the following:

1. An output sample from memory to the D/A Converter
2. A possible input sample from the A/D Converter to memory (depending upon whether record is turned on or off at the time)
3. A delay in time that corresponds to the SPEED value (actually, this delay is created with yet another loop inside the inner loop)

So, with 146 outer loops there are 146 x 256 actual bytes of memory used in this program (or 37,356 total).

This distinction between the outer, middle, and inner loops is important because the loop boundaries are a logical choice of where to update variables. For instance, on the Apple, a quick and clean place to update variables is every 256 bytes since the Apple is an 8 bit machine. Another good place is at the beginning of each outer loop and another is in the inner loop. As you play the

routines you can listen for differences in where variables are being updated.

When using the quick reference card, unless otherwise mentioned, the echo routines update the variables at the beginning of each outer loop.

## **WHAT THE KEYS DO ?**

The keyboard can be broken up into three general categories:

1. Special Function Characters
2. NUM and PUN Constants
3. Letters and Control Characters

Following is a description of each of the keys in each category:

### **SPECIAL FUNCTION CHARACTERS**

#### **ESC**

This key allows you to pause or continue at any time during the playing of a routine. Every other time you press ESC you pause or continue.

#### **RETURN**

This key alternately turns the record function on and off. When you are recording something you can at any time "freeze" memory and continue to just play through memory without destroying it.



### LEFT ARROW or RIGHT ARROW

These keys allow you to change the volume at any time. There are 16 volume levels with 0 being all the way off and 15 being maximum volume. The LEFT ARROW key lowers the volume and the RIGHT ARROW key raises the volume one step at a time.

### SPACE BAR

This key clears memory by filling memory with a constant value. What happens next depends upon whether or not the record mode was previously selected. If record was previously on then it will stay on and continue to stay on. If you were not recording previous to pressing the SPACE BAR then the program will still record but just one complete echo loop and then the record will automatically be switched off.

/ or ;

/ lowers the speed mask and ; raises the speed mask. The speed mask is a window within which the speed can possibly fall. If it is 127, for instance, than the speed used in the present echo routine can be between 1 and 127. If it is 3 than the speed will always fall within 1 and 3 (with 1 being the fastest speed possible). It is useful to limit the speed on many occasions to get more consistent speed variations.

The possible speed mask values are shown below:

1 3 7 15 32 63 127

? or + (or: ? or : for  
the Apple //e & Apple ///)

? lowers and + (or :) raises the length mask. This mask operates identically to the speed mask in that the possible length variations can be "masked" off so that when length is random, for instance, and the mask is set to 7 then the length will fall between 1 and 7 blocks long where each block is 256 bytes.

The possible length mask values are shown below:

1 3 7 15 32 64 127

, or .

These let you select the direction the echo routines operate in -- either forward or reverse or you can automatically alternate between the two each time through the routine. It is important to realize that if you record in reverse than reverse will sound normal and forward will sound like it's playing backwards.

\* or = (or:    or + for  
the Apple //e & Apple ///)

These two keys respectively raise and lower the PUN (PUNctuation) variables. This will correspond to whichever quality of the sound the PUN variables are presently controlling.

: or - (or: - or = for  
the Apple //e & Apple ///)

These two keys respectively raise and lower the NUM (NUMBER) variables. This will correspond to whichever quality of the sound the NUM variables are presently controlling.

< or >

These keys let you determine whether the present routine is shrinking, expanding, or alternating between the two each time through the routine. This only pertains to a few of the routines. You can experiment to see what I mean.

@ (or: ] for the Apple //e &  
Apple ///)

This key is used to sequence through 4 different types of displays that are shown on the screen. When the program originally boots up, the Page 1 text screen is displayed showing the main selection options and variables. Every time this character is pressed a new screen appears. The purpose of showing different screens is to allow you to actually "see" sounds being recorded and played in memory. Listed below is the actual sequence of displays.

1. Page 1 text screen showing main selections and variables
2. Page 2 text screen showing interesting text patterns
3. Page 2 low resolution graphics screen
4. Primary page high resolution graphics screen
5. (go back up to step 1 and continue the cycle)

It is important to realize that if RECORD is turned off none of the screens will show any movement because the data in memory at the screen locations is not being updated. It is also important to know where the screens are in memory so you can insure the present echo memory is passing through them. Page 2 text and low resolution graphics start at the beginning of echo memory. That is, when the beginning location is set to MINIMUM any length of echo will update the screen.

This is not the case with the high resolution screen as it starts 24 pages later with each page being 256 bytes long. Although in some routines

the beginning location can be set with the joystick to the beginning of the high resolution screen area and the length can be limited to only this section of memory.

One last comment about the graphics screens is that they are all in color and can be used with color monitors with the exception of the Apple /// low resolution graphics. The Apple /// in Emulation Mode does not support color low resolution graphics.

## NUM and PUN Presets

! thru )

The shift of the top row of keys (shift 1 through shift 9 for the Apple II and shift 1 through shift 0 for the Apple //e and Apple ///) are used to select what are called PUN (PUNctuation) constants. When a variable is controlled by the PUN value, length for instance, then the keys invoke preset lengths you can select from.

! thru 0

The top row of keys (1 thru 0 only) are the NUM (NUMBER) constants. When a variable is controlled by the NUM value, speed for instance, then there are 10 preset speeds you can select from.

## **LETTERS AND CONTROL CHARACTERS**

The quick reference card contains a summary of what type of echo the different letters and control characters produce. This corresponds to the matrix of variables shown on the bottom of the screen.