# ECHO...ECHO....ECHO.........................

A couple of issues ago, I said that we were going to look at a D/A that would allow those of you with exponential response synthesis equipment to begin playing with the computer software we have been discussing here. Then SEQUE ran longer than I thought it would, and we ran into logistics problems and .... In any case, it's not ready yet. Next time for sure.

Meantime, I've got some quickie code that I think you'll like, It's a program we call ECHO. I'll bet you think that ECHO echoes. It does.

It works in conjunction with an allocation algorithm (POLY from MUS 1 in this case, though something like Bob Yannes' SHAZAM could also be patched in to use this) and "follows" whatever data is being produced from QuASH channel #1, delaying it for a controllable period of time before playing it from a second channel, delaying again before playing on a third channel, and so on.

A convenient conceptual handle that may help you understand the "how-it-works" of ECHO might be a clock face. With only a second hand.

The numbers around the clock face represent memory locations and the second hand represents a pointer to these memory locations which, as it sweeps past each number, writes whatever note happens to be coming out of QuASH channel #1. This is really a funny clock, though, because in addition to the single second hand it has many minute hands that rotate at the same rate as the second hand. If the second hand is a "writing pointer", these funny minute hands are "reading pointers". Within some restrictions that we'll discuss shortly, we can have as many reading pointers as we like; the important feature is that each of these fast minute hands correspond to an additional QuASH channel.

Now as the clock runs, the writing pointer scans merrily through memory, writing the note that's in channel #1. In step behind it are the reading pointers, and as they point to successive memory locations they read them and place the result in the QuASH channel to which they correspond. Presto, echo.

In computerese, this kind of procedure is called a queue.

ECHO has a variety of software control features, and since I don't really know which of them are more important, we'll just plunge into the middle.

While ECHO always pulls the note that it's going to echo from channel #1, the first channel that the echo effect appears on doesn't have to be channel #2. Why? So that some channels can be set aside for polyphonic work while others are producing the echo.

Here's how. One piece of data that every polyphonic allocation subroutine must have is the number of output channels available for its use. POLY established the precedential name OUTS for this datum and set its location in a Paia 8700 as $EA.

Previously, we've always set this variable to represent the number of QuASH channels that were hardware supported. In a system which had a single QuASH, OUTS was set to contain $04 so that all available outputs were used for polyphonic allocation.

But OUTS may be set equal (may I please start saying "equal" instead of "contains"? It's not strictly true, but much less cumbersome.) to a number less than the number of hardware supported channels and the result will be to reserve some channels. In a system with two QuASH (for example) OUTS could be set equal to $05 and the result would be that the upper 3 channels (6 - 8) will not have keyboard activations directly assigned to them. POLY (or whatever) doesn't know they're there.

So we can use them for other things. Like echo channels.

ECHO, in its turn, must know how many channels it has to work with. The location labeled ECCO ($BB) serves this function, and in most cases will be set equal to the number of remaining channels.

To give a final example; if we make OUTS equal to $03 and ECCO equal to $05, we've produced a system which has 3 polyphonic channels (the first three) with channels 4 through 8 echoing, in sequence, the notes that appear on polyphonic channel #1.

I would be less than candid if I didn't forewarn you that successful use of a system which combines both polyphonic and echo channels requires a thorough understanding of the allocation algorithm being used as well as a certain manual and mental dexterity. It's best to start playing with a configuration which has only one channel available to POLY and the remainder used as echo channels. With practice, you can progress from there.

## DELAY CONTROLS

As you certainly know by now, all timing in our system references back to the scan rate of the keyboard, and ECHO has associated with it a variable labeled EDLY ($BC) which regulates how fast (in terms of keyboard scans) the hands in our clock analogy (the reading and writing pointers) advance from one memory location to the next, which in turn contributes to how long the echo delay is.

If we set EDLY equal to $01, the echoing routine is invoked after every keyboard scan (which is variable, but typically will be every 10 to 50 milliseconds). Making EDLY equal to $02 means that the routine is used on alternate scans which, if everything else is equal, will produce an echo delay twice as long.

Notice that this affects only the ECHO and does nothing to alter POLY's allocating channels after every keyboard scan. This is important because when changing the value of EDLY you should be aware that if you skip more than about 8 scans before invoking ECHO, it may miss some keyboard activity in a fast riff. The notes will still play through the polyphonic channels, but won't be echoed.

A second variable also interacts with EDLY to detirmine the echo delay. OFST ($BD) controls the offset between the pointers into the echo queue. Going back to the clock metaphore, it detirmines how "far apart" the hands on the clock are. The farther apart they are (the bigger the number in OFST), the greater will be the echo delay.

Like EDLY, there are some caveats that go with OFST. The echo buffer (queue) area of memory is 64 bytes on page 1. You don't want to come up with too many pointers (controlled by ECCO, remember) that are too far apart or they will represent a memory area larger than that set aside. The result of that is far from disastrous, but it will cause things like the high order channels echoing much sooner than you expected, as the reading pointers for those channels "wrap around" past the writing pointer. But, as we've decided here in the past, the difference between noise and a neat effect is often nothing more than a creative mind.

Control of the time delay involved in the echo is important for reasons that you might not first think about, because like any device (or now software) that messes with the subjective flow of time, echo offers a variety of totally different effects depending on how long a time we are talking about.

For example, if the delay is very short, as when both EDLY and OFST are set to $01, the effect will not even be percieved as an echo, but rather as a "thickening" of the voice (voice doubling, actually). It's a lot like phasing or flanging, except that with those techniques the predominant effect is frequently that the subjective flow of time is cyclicly changing.

Longer delays (EDLY = $01 and OFST = $08) produce the types of effects which give ECHO its name. Echoplex type echoing. There is a major difference, though, in that with conventional echo devices you can only echo in a voice that is essentially the same as the starting voice. Here, the echoes can be anything, and there's no way to appreciate the power that this implies without working with it.

When the delays get very long (EDLY = $02 and OFST = $10) you find yourself playing with an instrument that allows you to play rounds with yourself. Also, of course, in different voices.

Because the character of the instrument is so greatly influenced by delay times, and because the different characters can so frequently be used in the same musical performance, we've added a means of quickly switching from one set of operating parameters to another. Four of these presets are provided by pads 0-3 on the command keyboard. Touching one of these pads causes ECHO to get the requested set of parameters from a table that lives in memory $9A - $A9 and place them in the locations referenced by the rest of the program. The pre-sets that are in place in the listing which follows are:

| COMMAND KEY | POLY CHANS | ECHO CHANS | TIME DELAY (KBD SCNS) |
|---|---|---|---|
| 0 | 1 | 7 | 1 |
| 1 | 1 | 7 | 8 |
| 2 | 1 | 3 | 16 |
| 3 | 1 | 3 | 32 |

Notice a couple of things here. First, if you're using a system with only a single QuASH (a P4700/J or its equivalent) it doesn't matter that there are more echo channels than there are hardware channels; the last four iterations simply won't have the hardware to voice them. Secondly, observe that when we got to longer delays we cut back on the number of echo channels so as to circumvent the "too many channels too far apart" problem that we looked at earlier.

You can substitute your own presets for those shown simply by altering or replacing the values shown. Here is a map of locations that will make that a little easier:

| | PRESET # | | | |
|---|---|---|---|---|
| | 0 | 1 | 2 | 3 |
| OUTS | $9A | $9E | $A2 | $A6 |
| ECCO | $9B | $9F | $A3 | $A7 |
| EDLY | $9C | $A0 | $A4 | $A8 |
| OFST | $9D | $A1 | $A5 | $A9 |

With some experimentation you will find echo presets which seem to complement each other particularly well. You will inevitably get to where you use a specific set of presets for each particular song, not only changing presets throughout the song but within a riff or phrase. This can create some neat effects such as having an initially long delay set and, in the middle of the echo chain, hit a faster preset to initiate a burst of echoes. Or, have one preset for the "voice doubling" characteristics we discussed. Then you can switch between echoes for special effects and doubling for use on bass lines or solos.

Actually, there is a lot of power hidden in this program that can be liberated with innovative patching, voicing, and mixing. How about having a chain of voices which are all related but slightly different, such as having higher Q on the filters as the echo is passed on. Or changing envelope times so the first echoes have sharp attacks and delays and later voices have increasingly softer envelopes. Here's a good one- progressively detune each voice so you get a spiraling echo, or the echoes sequence upscale (or downscale). Completely different voices can be used, and this technique really works well on the long delays for doing rounds.

Just playing with the mixing or panning of the normal echo voices can entertain you for hours. Have the echoes pan across the stereo field, or bounce back and forth. Or have the echoes begin to fade out, but set the last or next to last voice at a higher level.

You can also use a multi- voice setup with only a few of the outputs driving voices. Set up the computer to provide (for example) one poly voice and seven echo voices, but only use channels 1, 4, 5, and 8 to drive oscillators. Work with various combinations here; each is a completely different rhythm and could easily provide a rhythmic basis for a whole piece.

Well, by now you are probably ready to dig into the program, so here is the listing.

## LOADING THE PROGRAM

As with other programs that we've examined in the past, ECHO may be hand-loaded using the 8700 computer's monitor, but first set the monitor stack pointer:

```
O-E-D-DISP-F-F-ENT
```

and the user's stack pointer and status register:

```
O-F-E-DISP-F-F-ENT-O-O-ENT
```

and then load the program:

```
O-O-O-DISP-2-O-ENT-2-1-ENT-8-D-ENT- (etc.)
```

and don't forget this data base information:

```
088-  20  21  0D  4C  C0  FF  C9  07
090-  D0  05  A0  5C  20  52  0D  4C
098-  10  10  01  07  01  01  01  07
0A0-  01  08  01  03  02  08  01  03
0A8-  02  10
0B8-  FF  FF  01  03  02  04
0E8-  40  20  01
```

After loading (and before running) the program and data should be dumped to tape (from location $000 to $0EC) using this sequence:

```
0-0-0-0-0-0-E-C-0-1-D-D-TAPE
```

When this tape is loaded in the future, it should be loaded from $000 to $0EC so that the presets will be loaded along with the program.

```
                0010  :********************************
                0020  :*                              *
                0030  :*          ECHO 0.31           *
                0040  :*                              *
                0050  :*   POLYPHONIC VOICE QUEUING   *
                0060  :*                              *
                0070  :*             BY               *
                0080  :*        'JOHN SIMONTON        *
                0090  :*                              *
                0100  :*(C) 1979 PAIA ELECTRONICS, INC*
                0110  :*     ALL RIGHTS RESERVED      *
                0120  :*                              *
                0130  :********************************
                0140  :
                0490  :
                0500  :INITIALIZE SYSTEM, CLEAR OUTPUT BUFFERS AND ECHO BUFFER
                0510  :
000-  20 21 0D  0520  STAR JSR INIT     :CALL MUS1 INITIALIZATION
003-  A2 FF     0530       LDX 0FF      :PREPARE TO SET STACK POINTER
005-  9A        0540       TXS          :SET STACK TO TOP OF PAGE
006-  A9 00     0550  EBZR LDA 00       :PREPARE TO ZERO OUT ECHO BUFFER
008-  A2 3F     0560       LDX 3F       :POINTER TO END OF ECHO BUFFER
00A-  9D 00 02  0570  ILP  STA EBUF,X   :ZERO ECHO BUFFER LOCATION
00D-  CA        0580       DEX          :POINT TO NEXT LOCATION
00E-  10 FA     0590       BPL ILP      :NOT DONE YET, LOOP
010-  20 71 0D  0600  ECHO JSR POLY     :CALL MUS1 POLYPHONIC ALLOCATION
                0630  :
                0640  :DETERMINE ADDRESS OF THE FIRST CHANNEL AVAILABLE
                0650  :FOR ECHO USE
                0660  :
013-  A0 0F     0670       LDY 0F       :OFFSET TO FIRST OUT-BUF LOCATION
015-  A6 EA     0680       LDX *OUTS    :NUMBER OF POLYPHONIC CHANNELS
017-  88        0690  LP0  DEY          :POINT TO NEXT OUTPUT CHANNEL
018-  CA        0700       DEX          :ONE LESS POLY CHANNEL
019-  D0 FC     0710       BNE LP0      :ALL POLY CHANS NOT USED, LOOP
01B-  84 EB     0720       STY *OUTT    :SAVE FIRST ECHO POINTER FOR LATER
                0730  :
                0740  :ADVANCE ECHO BUFFER POINTER AND ADJUST IF NECESSARY
                0750  :
01D-  A6 BE     0760       LDX *EPNT    :GET CURRENT ECHO BUFFER POINTER
01F-  C6 EC     0770       DEC *CNTR    :DECREMENT TIMER
021-  D0 09     0780       BNE GETN     :TIME NOT UP, BRANCH
023-  A5 BC     0790       LDA *EDLY    :TIME UP, RE-INIT TIMER VALUE
025-  85 EC     0800       STA *CNTR    :RE-INITIALIZE TIMER
027-  CA        0810       DEX          :POINT TO NEXT
028-  10 02     0820       BPL GETN     :BRANCH IF STILL WITHIN BUFFER AREA
02A-  A2 3F     0830       LDX 3F       :OTHERWISE, RE-INIT POINTER
02C-  86 BE     0840  GETN STX *EPNT    :SAVE NEW POINTER
                0850  :
                0860  :PUT CURRENT CHANNEL 1 NOTE IN ECHO BUFFER AND
                0870  :PREPARE ECHO CHANNEL COUNTER
                0880  :
02E-  A5 DF     0890       LDA *CHN1    :GET CHANNEL 1 NOTE
030-  9D 00 02  0900       STA EBUF,X   :SAVE IN ECHO BUFFER
033-  A5 BB     0910       LDA *ECCO    :GET NUMBER OF ECHO CHANNELS
035-  85 BA     0920       STA *TEMP    :SAVE AS COUNTER
                0930  :
                0940  :CALCULATE SUCCESSIVE ECHO BUFFER LOCATIONS AND
                0950  :ADJUST AS NECESSARY
                0960  :
037-  8A        0970  LP1  TXA          :ECHO BUFFER POINTER TO ACCUMULATOR
038-  18        0980       CLC          :PREPARE FOR ADDITTION
039-  65 BD     0990       ADC *OFST    :CALCULATE NEXT LOCATION
03B-  C9 40     1000       CMP 40       :STILL WITHIN ECHO BUFFER?
03D-  90 03     1010       BCC SAVE     :YES, BRANCH TO CONTINUE
03F-  38        1020       SEC          :NO, SET CARRY FOR SUBTRACTION
040-  E9 40     1030       SBC 40       :AND ADJUST POINTER
042-  AA        1040  SAVE TAX          :PUT POINTER IN PLACE
                1050  :
                1060  :THEN PULL NOTES FROM ROTATED ECHO BUFFER LOCATIONS
                1070  :AND PLACE IN ECHO CHANNELS OF OUTPUT BUFFER (NTBL)
                1080  :
043-  BD 00 02  1090       LDA EBUF,X   :GET NOTE FROM ECHO BUFFER
046-  99 D0 00  1100       STA NTBL,Y   :PLACE TO OUTPUT CHANNEL
049-  88        1110       DEY          :POINT TO NEXT OUTPUT CHANNEL
04A-  C6 BA     1120       DEC *TEMP    :ONE LESS ECHO CHANNEL
04C-  D0 E9     1130       BNE LP1      :BUT SOME LEFT, LOOP
                1140  :
                1150  :NOTES ARE PLAYED BY CALLING THE QUASH DRIVER (NOTE).
                1160  :FINALLY, ECHO OUTPUT CHANNELS ARE CLEARED SO AS NOT
                1170  :TO CONFUSE POLY WHEN CALLED
                1180  :
04E-  20 2B 0D  1190       JSR NOTE     :CALL MUS1 QUASH DRIVERS, ETC.
051-  A4 EB     1200       LDY *OUTT    :GET FIRST ECHO CHANNEL POINTER
053-  A6 BB     1210       LDX *ECCO    :GET # OF ECHO CHANNELS
055-  A9 00     1220       LDA 00       :PREPARE TO ZERO
057-  99 D0 00  1230  LP2  STA NTBL,Y   :ZERO ECHO OUTPUT CHANNEL
05A-  88        1240       DEY          :POINT TO NEXT OUTPUT
05B-  CA        1250       DEX          :ONE LESS ECHO CHANNEL
05C-  D0 F9     1260       BNE LP2      :SOME LEFT, LOOP
                1270  :
                1280  :READ COMMANDS. 0-3; PRESETS, 4-INITIALIZE SYSTEM
                1290  :5-CLEAR ECHO, 6-BREAK, 7-TUNE
                1300  :
05E-  20 00 0F  1310       JSR DECD     :READ COMMAND KEYBOARD
061-  C9 04     1320       CMP 04       :IS COMMAND A PRE-SET?
063-  10 1B     1330       BPL NEXT     :NO, BRANCH FOR NEXT TEST
                1340  :
                1350  :THE COMMAND IS TO CALL UP A PRE-SET.  AFTER CALUCLATING
                1360  :THE BASE ADDRESS OF THE PRE-SETS CALLED FOR, THE PRESET
                1370  :VALUES ARE TRANSFERED TO THEIR RESPECTIVE LOCATIONS
                1380  :AS ACTIVE PARAMETERS. NOTE THAT THE NUMBER OF
                1390  :CHANNELS ALLOCATED TO POLY USAGE (OUTS - $00EA) IS IN
                1400  :NON-CONTIGUOUS LOCATION AND MUST BE HANDLED SEPARATELY
                1410  :NOTE THAT THE CONTIGUOUS LOCATION *TEMP IS USED AS A
                1420  :DUMMY VARIABLE AT THIS POINT
                1430  :
065-  8C 20 08  1440       STY DISP     :SHOW PRESET
068-  A9 FF     1450       LDA 0FF      :ONE LESS THAN PRESETS BASE ADDRESS
06A-  18        1460  LP3  CLC          :PREPARE FOR CALCULATION
06B-  69 04     1470       ADC 04       :THERE ARE 4 PRESET VARIABLES
06D-  88        1480       DEY          :POINT TO NEXT PRESET BASE
06E-  10 FA     1490       BPL LP3      :IF NOT THIS PRESET, LOOP
070-  AA        1500       TAX          :PUT POINTER CALCULATED TO X
071-  A0 03     1510       LDY 03       :4 PRESETS, WILL COUNT TO -1
073-  B5 9A     1520  LP4  LDA *PRST,X  :GET PRE-SET DATA
075-  99 BA 00  1530       STA TEMP,Y   :AND PLACE AS ACTIVE PARAMETER
078-  CA        1540       DEX          :POINT TO NEXT PRESET DATA
079-  88        1550       DEY          :AND NEXT ACTIVE PARAMETER
07A-  10 F7     1560       BPL LP4      :IF NOT YET DONE, LOOP
07C-  85 EA     1570       STA *OUTS    :SAVE THE MAVERICK PARAMETER
07E-  30 90     1580       BMI ECHO     :BRANCH ALWAYS
                1590  :
080-  F0 7E     1600  NEXT BEQ STAR     :COMMAND IS FOR CLEAR, BRANCH
082-  C9 06     1610       CMP 06       :IS COMMAND 5 (CLEAR ECHO) OR 6 (BRK)?
084-  30 80     1620       BMI EBZR     :COMMAND IS CLEAR ECHO, BRANCH
086-  D0 06     1630       BNE NXT0     :COMMAND IS NOT BRK, BRANCH
088-  20 21 0D  1640       JSR INIT     :SHUT DOWN SYNTHESIZER
08B-  4C C0 FF  1650       JMP BRAK     :AND RETURN TO MONITOR
08E-  C9 07     1660  NXT0 CMP 07       :IS COMMAND TUNE?
090-  D0 05     1670       BNE BRDG     :A BRANCH TOO FAR
092-  A0 5C     1680       LDY 5C       :PREPARE TO TUNE TO MIDDLE C
094-  20 52 0D  1690       JSR FILL     :SEE MUS 1.0 DOCUMENTATION
097-  4C 10 10  1700  BRDG JMP ECHO     :PLAY ON AND ON AND ON
                1710  :
                1720  :SET-UP VARIABLES FOR MUS1
                1730       .OR 10BA     :INITIAL PRE-SET
                1740       .HS 01030204
                1750       .OR 10E8     :SYSTEM CONTROL AND QUASH DELAY
                1760       .HS 402001   :AND OUTS
                1770  :AND PRESETS
                1780       .OR 109A
                1790       .HS 01070101
                1800       .HS 01070108
                1810       .HS 01030208
                1820       .HS 01030210
                1830  :
                1840  END  .EN
```