The Digitizer 3+, from D.S.D. Software.
===

Audio digitizing is one of the most interesting things that you can do on your Color Computer 3. By utilizing the capabilities of the 6 bit ADC, (Analog to Digital Converter) in your CoCo, you can sample in sound, through your left joystick port, that will allow you to take full advantage of your CoCo's 6 bit DAC, (Digital to Analog Converter). Six bit sound is 32 times better in quality than the one bit sound that you get from other digitizers that get their input from the cassette port. This has several advantages: You do not need to send sound through the DAC nearly as fast to get good sound, and the sound quality is much improved.

Digitized samples add much to the quality and excitement of computer games. Have you noticed that a lot of new games feature digitized sounds? These sounds are much more impressive than the beeps and bleeps most older games have. BUT -- How do those games use digitized samples without making the game play seem choppy, or freeze every time a sample was played? The CoCo's FIRQ, (Fast Interrupt ReQuest) interrupt is the answer to that question!

Now most 6809 programmers suddenly go silent at the mention of an FIRQ. I don't know why, but less than a year ago, none of the so called "6809" experts seemed to know much about these interrupts. Even William Barden Jr.'s famed TRS-80 Color Computer Assembly Language Programming manual treated even the slightest mention of interrupts as being far too complex for any mere programmer. I have never seen a Rainbow article mention anything about using the FIRQ, and only seen very few articles about linking into usage of the IRQ. Now the FIRQ is a programmable interrupt that can be programmed to go at extremely fast speeds. The 60Hz IRQ has a snails pace compared to the FIRQ's capability of 70ns. Any 6809 programmer knows that digitized sound requires sending data to the DAC at speeds roughly between 5,000 to 10,000 times per second. No problem for the FIRQ, and no problem at all for the Digitizer 3+!

The Digitizer 3+ works with the FIRQ, and multi-tasks
digitizing and playing with the operation of the program. This
allows many features that other digitizers cannot offer: Being
able to display a sound waveform while both digitizing and playing,
being able to sample and play sounds at extremely precise speeds,
and allowing you to multi-task digitized sound with BASIC (or
assembly language). Also, the Digitizer 3+ has a built-in waveform
editor.

The Digitizer 3+ is very easy to use! Most of the main
commands are displayed on screen making it possible to use the
Digitizer 3+ without even reading the documentation!

To Start, make sure you have your DIGITIZER 3+ cable plugged
into your LEFT JOYSTICK PORT in the rear of your CoCo 3, and the
MALE RCA end of the cable plugged into the LINE OUT of your sound
source. Turn your printer or modem off, (to avoid unwanted
interference) and simply type: RUN "DIGI3+" and press [ENTER].

Key Commands.
==============

[R]   —   Records a sample into the current block sequence.

(Left arrow) —   Plays the current sequence in reverse.

(Right arrow) —   Plays the current sequence forward.

[S]   —   Stops playing or recording.

[ALT]   —   Allows changing of block sequence.

[F1]   —   Toggles wave display on or off.

[F2]   —   Toggles continuous play loop on or off.

[W]   —   Turns on wave/volume test. (Press any key to end.)

(Up arrow)   —   Speeds up playback and recording factor.
                       (Shift to increase by larger steps.)
(Down arrow) —   Slows down playback and recording factor.
                       (Shift to decrease by larger steps.)

[D]   —   Go to disk commands. (Load/save.)

[E]   —   Edit sequence waveform.

[Shift-CTRL] —   Go to block commands. (Smooth, mix, reverse, copy)

```
                    Waveform Editor Commands.
                    ===========================

(Auto key-repeat active in edit mode!)

(Left arrow)   -      Move one byte back.
                      (Shift to move 16 bytes back.)

(Right arrow)  -      Move one byte forward.
                      (Shift to move 16 bytes forward.)

(Up arrow)     -      Increase byte by one.
                      (Shift to increase by 16.)

(Down arrow)   -      Decrease byte by one.
                      (Shift to decrease by 16.)

[F1]           -      Move 128 bytes back.   (One screen)

[F2]           -      Move 128 bytes forward.   (One screen)

[ALT]          -      Average byte between previous and next.
                      (Used to smooth wave.)

[CLEAR]        -      "Zero's" byte to value 31.
                      (Used to make silence.)
                      (***Shift to "zero" 128 bytes***)

[ENTER]        -      Exit editor.



o  Pressing [BREAK] from anywhere except the waveform editor will exit you
   to  BASIC.  You can get back to the Digitizer 3+ by typing "GOTO 1000".

o  To make sure you are getting a good waveform, use WAVETEST to see the
   wave, and turn the volume quickly up and quickly back down.  On most
   audio equipment, 40% of the maximum volume is best.  For more powerful
   amplifiers, use your best judgement, (start with a lower volume).
   Sometimes turning the amplifier off and back on helps make the waveform
   better.  Try to get the wave in the middle of the display.  See WAVETEST
   on the next page.
```

WAVE TEST

The wavetest feature will allow you to see the waveform of incoming sound so that you can determine if you are going to get a good sample, and if the volume is too low or too high. The waveform should be clean, and the peak of each wave should not be clipped at the top or bottom of the screen. If the volume is too loud, (or clipping is occurring), the blue on the screen will flash red for about half a second. This is handy because sometimes the clipping may not be noticeable until after you record the sample.

BLOCK ALTERATION

The Block Alteration menu will allow you to alter blocks of sound in three different ways. Selection one, "Average/smooth sound block" will "tone down" the sound in a block, and "smooth" out the sound waveform. This is useful when mixing blocks when you want one of the mixed sounds to be more quiet than the other sound. Selection two, "Mix two sound blocks" will actually do a true adding of the two sound waveforms. The first block will be added to the second block, and also divided by two as to retain the same volume level. The third selection, "Reverse a sound block", will flip the sound block around, so that when it is played back, it will sound backwards. This can be used to generate interesting effects. One commonly used effect in music is to mix a sound with itself played backwards. The forth selection, "Copy one block to another", will do just that, copy the first block into the second block. The last selection returns you to the main program.

DIGIPLAY
========

DIGIPLAY.BIN is a short collection of ML routines to allow you to put multi-tasking digitized samples into your own programs. (Basic or machine language.) It includes five ML routines:

PLAY: (EXEC &H6F92) This routine plays the current sequence in memory, beginning at &H7040, terminated with a zero.

AVERAGE: (EXEC &H6F4B) This routine average/smoothes out a block of sound, stored in &H6F4A. (ie: POKE &H6F4A,112:EXEC &H6F4B would average/smooth out ram block #112.)

MIX: (EXEC &H6F02) This routine will mix/add two blocks together. The first block (stored at &H6F00) will be mixed/added into the second block (stored at &H6F01.)

COPY: (EXEC &H6F70) This routine will copy a source block (stored at &H6F00) to a destination block, (stored at &H6F01.)

REVERSE: (EXEC &H6F2A) This routine will reverse a block (stored at &H6F29) so that it will be played backwards.

You can make the sequence be played in a continuous loop by POKING a zero at location &H6F91. To turn this option off, POKE a one at location &H6F91.

Location &H6F90 can be used to stop the playing of a sequence, (by POKING a zero in that location), or it can be used to check if the sequence is still being played, (by PEEKING at location &H6F90 -- If the value is not zero, then a sequence is being played).

DIGIPLAY.BAS has a brief outline of the routines, and SNAP.BAS shows you how to load in a sequence and play it continuously with BASIC.

<center>BLOCK NUMBERS</center>
<center>=============</center>

When the Digitizer 3+ is first loaded, it uses blocks #112 to #119. These are the HSCREEN, HGET/HPUT, WIDTH 80 and secondary stack memory blocks. They are not used by Extended Color Basic, but are used by the CoCo 3 Super Hi-res commands. If you only have 128k, (like me), these are basically the only blocks you can use, (you can also use block #122). If you have 512k, you should have from block #64 to block #111 as well! If you program is completely assembly language, you can eliminate some of the ROM blocks. (#124, #125, #126 and some of #127).

This does not mean that if you only have 128k that all the CoCo 3 commands are useless with DIGIPLAY.BIN! Block #118 is the WIDTH 80 screen, you don't need that for the HSCREEN graphics! And block #119 is unused completely!

Here's a Block Chart:

```
  BLOCK#            FUNCTION
==================================================================
· 64 to 111         Unused 384K of 512K
    112             HSCREEN 1 - 4
    113             HSCREEN 1 - 4
    114             HSCREEN 2 & 4 ONLY
    115             HSCREEN 2 & 4 ONLY
    116             HGET / HPUT
    117             SECONDARY STACK
    118             WIDTH 40/80
    119             UNUSED BY BASIC
    120             $0000 - $1FFF
    121             $2000 - $3FFF
    122             $4000 - $5FFF
    123             $6000 - $7FFF   (Some of this is WIDTH 40/80)
    124             $8000 - $9FFF   (EXTENDED COLOR BASIC)
    125             $A000 - $BFFF   (COLOR BASIC)
    126             $C000 - $DFFF   (DISK BASIC / CARTRIDGE ROM)
    127             $E000 - $FFFF   (SUPER BASIC, VECTORS AND I/O)
==================================================================
```

Saving and Loading

You can save and load samples one block at a time, or a whole sequence at a time. Each block is saved as a separate file, with the first eight characters of the title as the filename, and the block number as the extension. When loading a block, you can load it into any memory block you like.

Since you can alter your block sequence in any way, make sure that when you save your sequence, that you don't repeat any of the blocks in your sequence, or else those blocks will be saved more than once!

One last note... You will notice that at higher speeds, the digitizing process will reach a peak speed, which is substantially lower than the speed that the playback routine can run. This is simply because it takes a lot more work to read in sound, than to send it out. The play routine consists of about 34 instructions, while the record routine is made up of over 60 instructions. This means that the record routine takes almost twice as much processor time than the play routine, and therefore the play routine can go about twice as fast as the record routine.

Have fun!   -Derek Snider-

DIGITIZER 3+ CABLE PLANS

6PIN DIN PLUG

1 MFD
35VOLTS
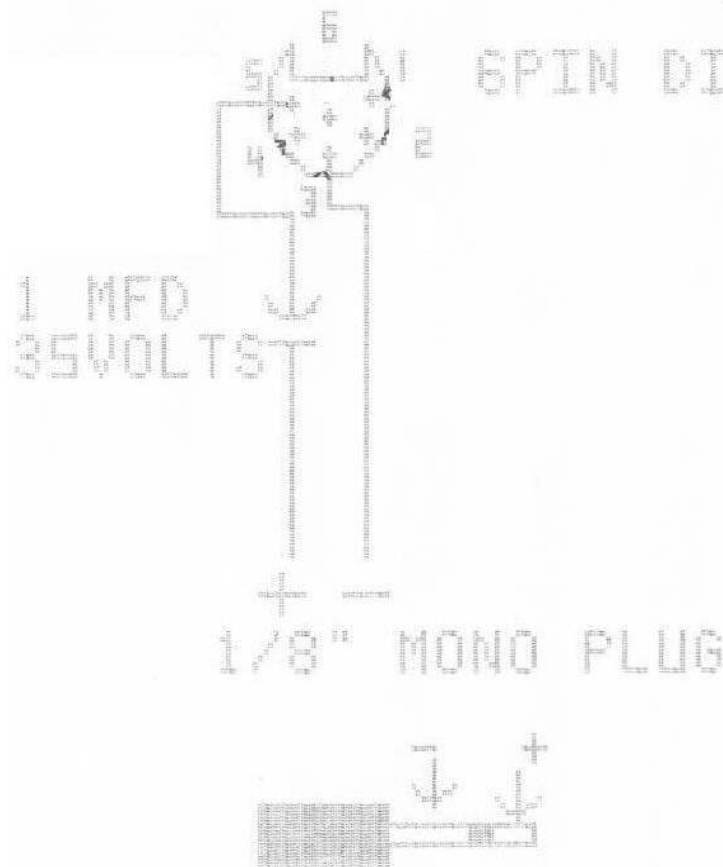
1/8" MONO PLUG

PARTS LIST

6PIN DIN PLUG

1.0 MFD 35 VOLT
TANTALUM
CAPACITOR

1/8 INCH MONO
PLUG

SPEAKER WIRE
(MOST ANY KIND
OF WIRE WILL DO
)