Warranty Information

Studio Works is supplied with the following warranty and restrictions:

Warranty:

The program will load and operate on a properly equipped computer for a period of one year from date of purchase.

If the program fails to load or operate within the warranty period (one year from date of purchase), Oblique Triad will replace the program diskette(s) upon receipt of original(s) free of materials charges by pre-paid post. Oblique Triad will not pay shipping charges incurred by user for returning originals, but will pay return postage for replacements.

There is no other warranty, expressed or implied, upon this product.

Restrictions:

There are no warranties of merchantability of fitness for a particular purpose.

In no event will Oblique Triad be liable for damages caused, or losses incurred by use or misuse of the program.

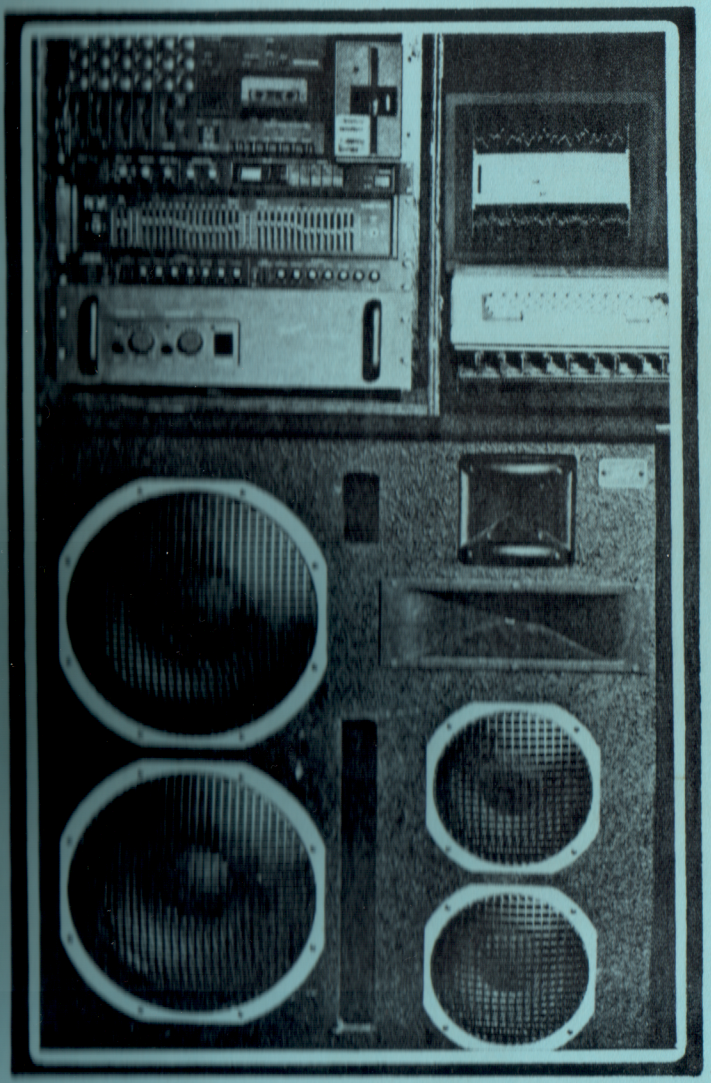
Oblique Triad

32 Church Street,
Georgetown, Ontario,
CANADA. L7G 2A7

Telephone: (416) 877-8149

We accept: AMEX, MasterCard,
Cheque or Money Order
C.0.D. in Canada only.

Studio Works



A Digital Sound Editing System

Studio Works
Version 1.0

7 April, 1989

Oblique Triad
32 Church Street
Georgetown, Ontario
Canada L7G 2A7

Telephone: (416) 877-8149

Trademarks

Colour Computer 3 is a trademark of InterTan Canada Ltd.
Walkman is a trademark of Sony of Canda Ltd.
Maxsound is a trade mark of Gimmesoft U.S.A.

Printed in Canada

Table of Contents

1. Getting started

1.1 System requirements

Studio Works requires a Colour Computer III, 128k (512k recommended), one disc drive, one standard joystick or mouse, and a sound source adaptor (or 'capture') cable.

The cable is supplied with the package if so ordered, or you may use the Maxsound™ cable, or you may build your own using the schematic in section 5.3

1.2 Connecting the capture cable

The capture cable should be connected to the CoCo 3 via the left joystick port on the rear left of your computer. The other end of the capture cable is terminated with a standard 1/8" mono plug. If so ordered, it may have a 1/8 to RCA adaptor, for use with line-level signals from a stereo and/or a 1/8" to 1/4" adapter.

1.2i Portable personal stereos

These small units, such as a Walkman™, will accept the capture cable directly. You may find that some of these units will be incapable of supplying sufficient signal to the computer for an adequate sample.

1.2ii Mid-size portable stereos

Almost all mid-range portables have a headphone jack, which is usually found on the front panel. You will require a 1/8" to 1/4" adaptor plug to use these units with Studio Works.

1.2iii Home stereos:

These units usually present the best signals from their Line Out connexions usually found on the back cover. These connexions are usually made with RCA plugs, and so you may

need the appropriate adaptor before you can use these units with Studio Works.

Make sure the volume control of your sound source is set as low as possible before you start, and be extremely careful when you first turn the volume up. Your best guide is to watch the on-screen VU meter, as you gradually increase the volume. There must, of course, be a signal present when you set the volume controls. See section 2.2 for more details.

1.3 Loading the program.

Insert the Studio Works disc into any drive and type 'LOADM"STUDIO:[drive number]'. The program will auto-execute.

2. Using the program

2.1 Terms and Definitions

Now that the cable is connected to a suitable sound source, and the program is loaded, you should be greeted with this screen layout:

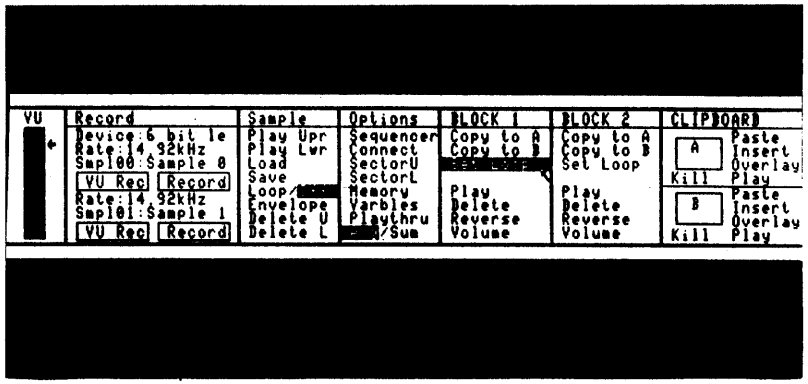


Figure 1-Main Screen

The black areas across the top and bottom are where your digitized sounds will be represented as graphs of amplitude versus time. In the centre is the white area containing all the record/playback/edit function buttons. This area is called the function bar. An arrowhead cursor on the screen is controlled by a standard joystick or mouse plugged into the right joystick port.

When the word sample is used in this manual, it will mean either the action of capturing a sound into memory, or the actual resulting digital recording.

We shall use the following terms when speaking of mouse procedures:

POINT: Move the cursor over an object on the screen.

CLICK: Press the mouse button once and then release.

DRAG: Press and hold the mouse button, then move the mouse until the desired effect is achieved, then release the button.

OK BUTTON: On some dialogue boxes (see below), the OK button is a section of the screen which you should click upon when all the settings in the box are correct, and you wish the function to be executed.

CANCEL BUTTON: On some dialogue boxes, you may click the CANCEL button to return to the main screen without any further action taking place.

Studio Works responds in several ways, which we shall describe as:

PING: A sharp ping sound, indicating a completed action.

BOOP: A short, low boop sound, indicating an error, or that a function has filled all available memory.

DIALOGUE BOX: A section of screen that 'pops up' on top of the screen and asks for more information to complete a function.

INFORMATION BOX: A 'pop-up' section of screen that informs the user of events or errors.

2.2 To Digitize a signal:

The first step in digitizing a signal is to set the correct incoming volume level. The VU meter on the left of the screen indicates this level. The best volume setting is one where the VU meter just touches the top of its range, but only rarely exceeds it. An overly large input signal is signalled by the VU meter going full scale (all the way up) and then blanking off. An excessive input signal can also 'bleed over' from the left joystick port into the right, causing the cursor to jump from left to right. This is another indicator of excessive input levels. Be sure, of course, to set the volume level while

the signal you wish to capture is being presented to the computer.

You should now set your sound source to re-play the desired sound

You now have two options: standard Record, or VU Record.

2.2i Standard Record:

Click and hold over one of the two RECORD buttons on the left side of the screen. You will hear the input signal redirected through the monitor's speaker. You should hold the button down and listen for the beginning of the desired sound, whereupon you should release the button. The play-through will cease and the program will start to record the incoming signal. The VU meter is also blanked at this time since every scrap of the computer's speed is required to accurately capture the signal. When you judge you have recorded long enough, click the mouse button again, and your sound will appear in graphical form ready to be played or edited or saved. Do not worry about digitizing too much sound, since any excess may be easily deleted as outlined later.

2.2ii VU record:

If there is no lead-up to the desired sound, or your reflexes aren't as quick as they used to be, you may let the computer sense when to start recording by itself. Click and hold the VU RECORD button. The program will 'play through' the incoming sound, and wait for the button to be released. When the mouse button is released, the program will re-activate the VU meter, and wait for the incoming signal to exceed a preset level, whereupon it will begin recording. Again, click the mouse button to stop recording. The preset level may be altered by dragging the small arrow to the right of the VU meter up or down.

2.3 Playing the sound

To play, click on the play option under the sample menu in the middle left of the screen. If you wish to play the upper sound, click PLAY UPR, and for the lower sound PLAY LWR.

2.4 Using the block markers.

Superimposed upon each sound you record will appear two vertical lines, each with a box at its upper or lower end. These lines are called the Block Markers and can be used when you wish to move a section of (or the whole) sound to one of the two clipboards A and B. To move a block marker, simply drag the box at its end to the desired point.

To delete a section of sound, for example, place the markers around the offending section, then click on the 'Delete' option in the BLOCK areas of the function bar. A dialogue box will ask you to confirm this action.

To move to the clipboards select either the 'Copy to A' or 'Copy to B' option in the appropriate Block area. A function analogous to the CUT function found in other types of clipboard may be performed by Copying and then Deleting a block.

2.5 The Clipboard

There are three methods of incorporating a clipboard sound into one of the active samples (upper or lower), each of which uses a similar command format. First click on one of the functions. The cursor will change to the first letter of the function: 'P' for paste, 'I' for insert and 'O' for overlay. Now you should click upon the section of the sample where you wish the clipboard sound to be incorporated. If you change your mind after clicking a clipboard function, just click on the function bar to de-activate the function.

Clipboard sounds are not deleted or affected when the clipboard functions, (other than KILL), are used.

2.5i Paste

Paste takes the clipboard sound and stores it directly into the sample. No extra space is made for the clip, and the original sample data 'underneath' the pasted clip is destroyed.

2.5ii Insert

Insert first makes room for the clip, then copies it into the empty space.

2.5iii Overlay

Overlay does not make room, but combines the clip and the sample according to the Avg/Sum setting found at the bottom of the Option column on the function bar. Average will average the two functions together, while Sum adds them.

You will find that Average produces a slightly quieter result than Sum, and should therefore be used on samples of high amplitude. Sum should be used on low-amplitude samples. If used on a high-amplitude signal, Sum will 'clip' the signal vertically to fit within the window (and within the capabilities of the CoCo 3's sound output). This can cause distortion, so ensure that the Avg/Sum flag is set correctly before you use Overlay.

When you have finished with a clip, you may remove it and free up its memory by selecting the Kill option in the Clipboard column.

2.6 Using Clipboards to create special effects.

2.6i Echo

First Copy the section (or entire sample) to be echoed into a clipboard. Select the Overlay function and place the clip a short distance after the beginning of the copied section. You may Overlay the same sound several times, in order to achieve a repeated echo. If you use the Average function instead of the Sum function (in the Options menu, explained fully above), the echo will be quieter than the original.

2.6ii Stutter

Stuttering may be achieved several ways. If you wish to repeat the first part of a sample, first copy that part to a Clipboard, then paste it directly before the original, as many times as necessary.

You may also create a stutter effect with the Looping controls explained later.

2.7 Using the Sample Archives

The Sample Archives can be used to start working on a different sample while the original lies 'quiescent' in memory. To move to the Sample Archives, click on the 'Smpl' area of either active sample in the Record section of the Function bar.

The sample archives are presented on a different screen (refer to figure 2), with all 64 sample numbers, their names (if assigned) and their Piano Play letters.

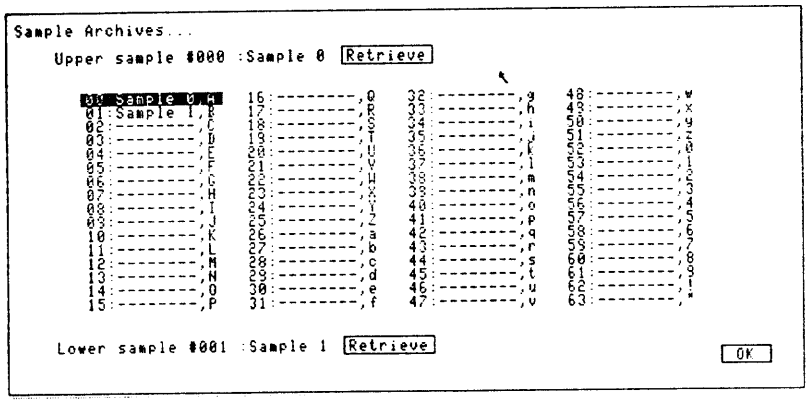


Figure 2-Sample Archive Screen

At the top is listed the upper sample's name and number, and at the bottom the same data for the lower sample. One of the entries in the lists will be highlighted. You may change which is highlighted by clicking on the desired name.

The retrieve function takes the data (including the sample number) from the highlighted entry in the list. You cannot assign both upper and lower samples to the same number.

2.8 Piano Play

On the Sample Archives screen (explained in the previous section), each sample is shown with a corresponding letter to the right of its name. When the Sample Archives screen is shown, you may play any of the 64 samples by pressing its associated key. If the sample has loop points assigned, holding the key will loop-play the sample until the key is released.

2.9 The Sequencer

Clicking on Sequence under the Options menu will move you to the Sequencer screen, refer to figure 3.

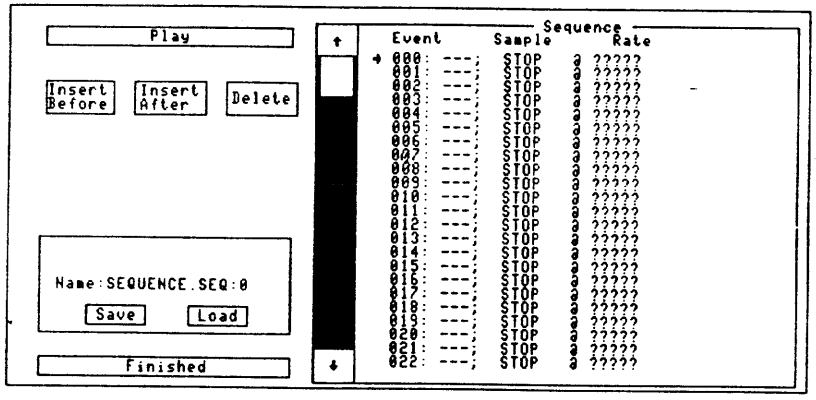


Figure 3 - Sequencer Screen

A sequence is a list of samples and the rates they should be played at. When you click play at the top left of the sequencer screen, the sequencer will go through this list, playing each one of the entries in turn until a STOP entry is reached, or you click the mouse button.

To change which sample an entry represents, click on the number column at that particular entry. Now type in the desired sample number and press enter.

To change the rate, click on the rate column and select from the pop-up menu which appears.

To select a STOP entry, enter a sample number greater than 63.

There are three buttons on the left which will allow you to delete the marked entry, or insert before or after the marked entry.

To change which entry is marked, drag the small arrow pointer on the left of the sequence list.

There is a scroll bar to the left of the sequence list which will allow you to move up or down the sequence by clicking and holding the arrows at either end, or by dragging the position marker in the middle of the scroll bar.

You may change the name of the sequence by clicking on the old name at the bottom left of the sequencer screen.

You may load or save a completed sequence with the load and save options below the name. A dialogue box will ask for confirmation.

When you have finished with the sequencer, click 'Finished' on the bottom left of the screen.

2.10 Saving/Loading

There are two ways to get your sound recorded on disc. Sector dumping allows easy access from machine language, and regular Saving allows you to re-load the sample into Studio Works and also to incorporate it into BASIC.

2.10i Sector Dump

The sector dump option (in the OPTIONS area of the function bar) will produce a dialogue box which will allow you to set the start track, start sector, and sector count for the dump. When you click on the OK button, Studio Works will start to put the sample, sector by sector, onto the disc until the sector count or the end of the sample is reached. Extreme care

must be taken with this function, since it does not check for data being present already on the disc, and will destroy all data on the sectors over which it is invoked.

2.10ii Saving a sample

The save option (in the sample area) will move you to the disc I/O screen, refer to figure 4

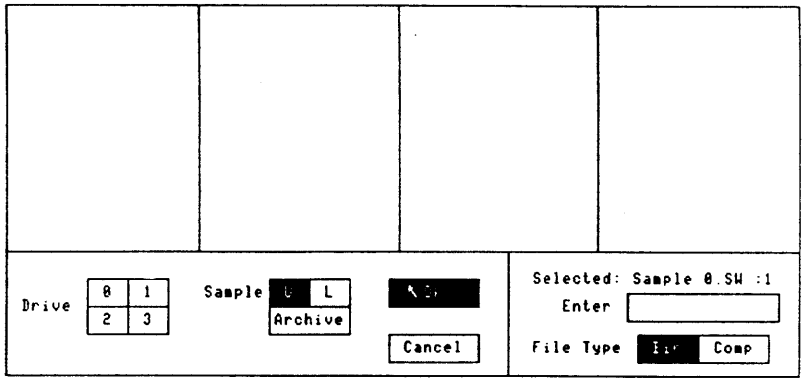


Figure 4-Disc I/O Screen

The first thing to do when you enter the disc I/O screen is to select the active drive. At the lower left are four buttons corresponding to drives 0 through 3. Just click the desired drive number.

When you are saving, you may click on the upr or lwr button in the lower middle of the screen to save the upper or lower active sample. The name in the 'selected' box will change to the name of the sample when you click one of these two buttons. The 'Archive' button is only used when loading.

If you wish to save under one of the names that appears on the disc, click upon that name. The name will be copied to the 'selected' area.

One last way to choose a file name is to type it in yourself. Click the box below the selected name and type in your new name

Note that you may only change the file name, the file extension will be automatically generated.

You should also decide whether to compress the file, or do a straight binary save. Only Studio Works can read a compressed file, While a straight binary file may be read into basic.

When you are satisfied, click the Save button.

If you change your mind, click the Cancel button to return to the main screen.

If you file is larger than the available disc space, Studio Works will ask you to insert another disc when it has filled the current one. The extension of the file name will be modified as new discs are used, such that both Studio Works and yourself will know in what order to load the files.

If the sample can be saved on one disc, the extension will be "SW ", otherwise the extensions will be "SWA" for the first file, "SWB" for the second and so on.

You may decide to abort the saving process when only part of the file is saved by clicking Stop when you are asked to change discs.

2.10iii Loading

To load a sample, click on Load in the SAMPLE AREA of the function bar. The program will switch to the disc I/O screen (shown above). As with Save, select the desired drive and review the directory you see.

Click on the desired sample and its name will be shown in the 'selected' area.

You should also click on which of the upper or lower sounds you wish to load into. Keep in mind that the loaded sample will overwrite the sample in memory that you have selected, so you should use the Sample Archives to select an empty sample number.

The 'Archive' option tells the program to load the sample into the sample number that was recorded with the sample when it was saved. This is extremely useful for loading

a lot of sounds into the correct archive position for use with a pre-recorded sequence. If you load with the 'archive' option on, the program will stay in the disc I/O screen after the load. Keep in mind also, that 'archive' load will erase any sample already in memory under that number, so take care when using the 'archive' function.

If the file has an extension other than "SW ", you will be propted to switch discs in order to load the rest of the sample.

3. Function by Function

3.1 The VU meter

The VU meter gives a representation of the amplitude of the input signal presented to the computer. The preset level of the VU record function is changed by dragging the arrow to the right of the meter up and down.

3.2 The RECORD area

3.2i Device

This button produces a menu from which you may select either 6 or 5 bit samples. A 6 bit sample (the default) will be cleaner than a 5 bit, but a 5 bit sample may be recorded at a higher rate. A higher rate means that you may record samples with higher frequencies in them without quite so much aliasing on the higher notes.. The two exclamation (!) points on the rate menu denote the maximum rates for each method.

3.2ii Rate

The record rate listed here may be changed by clicking upon it and then selecting from the menu that appears. Note that the highest rates at which Studio Works may record through the joystick port are 17.05 kHz for 5 bit samples and14.92 kHz for 6 bit samples. These values are marked with an exclamation point as a reminder of these limits.

3.2iii Smpl

Clicking here will switch to the Sample Archives screen where another sample may be retrieved. This screen also allows the Piano Play function, which lets you Loop-Play any sample by pressing and holding the key listed next to its name.

3.2iv Sample Names

If you click on a sample name, you may type in a new one. Hitting return on a blank line will cancel the operation.

3.2v VU Record

Click and hold here and the program will play-through the incoming sound out the monitor speaker. Release, and the program will wait for the incoming signal to exceed the preset level, then begin recording. Recording will stop when memory is exhausted, or you click the mouse button.

3.2vi Record

Click and hold Record to hear the play-through, then release to begin recording. Click again to halt recording.

3.3 SAMPLE

3.3i Play Upr

Play the sample in the upper window, if defined.

3.3ii Play Lwr

Play the sample in the lower window, if defined.

3.3iii Load

Moves to the disc load screen. See section 2.10.

3.3iv Save

Moves to the disc save screen. See section 2.10.

3.3v Loop/Off

Enables/disables sound looping. See section 3.5iii.

3.3vi Envelope

Allows you to shape an entire sample. First click "Envelope", then click on one of the sample windows. A horizontal line will appear superimposed and centred upon the sample. You may drag this line up or down to increase or decrease the sample's volume at that point. Re-click envelope to execute. Click any other part of the function bar to cancel.

3.3vii Delete U

Delete the upper sample and release its memory for future use. A dialogue box confirms.

3.3viii Delete L

Delete the lower sample and release its memory for future use. A dialogue box confirms.

3.4 OPTIONS

3.4i Sequencer

Moves to the sequencer. See section 2.9.

3.4ii Connect

Toggles the sample displays between connected dots (lines) and simple dots, Set to personal preference.

3.4iii Secdum Up

Sector dump the upper sample. See section 2.10.

3.4iv Secdum Lw

Sector dump the lower sample. See section 2.10.

3.4v Memory

Click and hold this option to see a display of each memory block and what it contains. The block will show a number if allocated to a sample, either CbA or CbB for the clipboards, or 'sys' for system or unavailable memory.

3.4vi Inform

Click this function for a display of the positions of the block markers, given in terms of how may 8k blocks into the sample and how many bytes into the last block. Useful for finding out what values to pass to the play routine when incorporating sounds into BASIC or M/L programmes.

3.4vii Playthru

Click and hold here to hear the signal present at the joystick port through the monitor speaker. The quality is poor due to the need to digitise and re-play the incoming signal.

3.4viii Avg/Sum

Set the Overlay method. See section 2.5iii.

3.5 BLOCK

The operations within both BLOCK function bars are the same.

3.5i Copy to A

Move the marked block to clipboard A without deleting the source block in the sample.

3.5ii Copy to B

Move the marked block to clipboard B without deleting the source block in the sample.

3.5iii Set Loop

When loop points are set for a sample with this function, and looping is enabled by the Loop/Off setting, that sample will be loop-played. When you hold down the button on the play option, the sound will be played up to the second loop.point, whereupon it will be played from the first loop point. The program will continue to cycle from the second point to the first until the button is released, whereupon it will play the sample from the point of release on to the end of the sample. A ping signifies that the loop point was successfully set.

3.5iv Play

Play the marked block at the sample's play rate.

3.5v Delete

Delete the marked block from the sample. A dialogue box confirms.

3.5vi Reverse

Horizontally flip the marked block, to make a section of the sample play backwards.

3.5vii Volume

Amplify or attenuate a section of sound according to the level set in the resulting dialogue box. You may also use this function to "zero-out" or completely deaden a section of the sample by setting the enhance factor to zero.

3.6 CLIPBOARD

The operations within both CLIPBOARD function bars are the same.

3.6i Paste

Copy the clip onto the sample

3.6ii Overlay

Combine the clip with the sample according to the Avg/Sum setting.

3.6iii Insert

Insert the clip into the sample

3.6iv Kill

Delete the clip and free up its memory for future use.

4. Using a digitized sound

4.1 Incorporating sound into BASIC

On the Studio Works distribution disc you should find a program called 'PLYLODSW/BIN'. This program is the interface between BASIC and Studio Works samples. 'PLYLODSW' is relocatable and saved with a zero start address. This means that you can place it anywhere in BASIC's work-space. If you wished the program to load in at $1000 for instance, you would use 'LOADM"PLYLODSW",&H1000' to load the program in. How you must execute the program is covered later.

If you are not going to use CoCo II style low-res graphics (PMODE 0,1,2,3,4) in your BASIC program, PLYLODSW may reside at $E00 (3584 decimal) which is where BASIC stores its low-res graphics screens. Since PLYLODSW is only 640 bytes long, you may use PCLEAR 1 in your program to free up more memory for BASIC.

If you are going to use PMODE type graphics, PLYLODSW should reside at $7D80. Use 'CLEAR200 &H7D7F:LOADM"PLYLODSW",&H7D80' to load the program. The first number after the CLEAR instruction represents how many bytes of string space to reserve (see BASIC manual on how to use this on pages 72 and 232). It is essential that the CLEAR instruction precede the LOADM, or BASIC will crash, since PLYLODSW will overwrite BASIC's stack area.

4.2 Using PLYLODSW to load and play a sample

To load a sample, you must execute PLYLODSW (with the USR function) and pass to it the name of a string that contains this information:

Position	Length	Meaning
0	1	Load/play flag. 0 = load.
1	1	Memory use byte.
2	8	File name, left justified, blank filled.
10	3	Extension, left justified, blank filled.
13	1	Drive number. (actual number, not ASCII).

An example of how to do this from BASIC is as follows:

```
10 A$ = CHR$(0) + CHR$(0) + "SAMPLE  " + "SW" + CHR$(1)
20 DEFUSR = &H7XXX:REM adjust to correct execute address
30 DV$ = USR(A$):REM DV$ is a dummy variable
```

The above program lines would load the sample 'SAMPLE/SW:1' from drive 1. Notice that 'PLYLODSW' must be in memory (id est: previously LOADMed) before this will operate. Notice also that no '/' or '.' or ':' is used in the file name.

The memory use byte gives you a way to control what parts of the computer's memory 'PLYLODSW' will use. There are three sections of lower memory that BASIC sometimes uses. These are the hi-res graphics and hi-res text screens, and the hi-res GET/PUT buffer. If you are using any or all of these in your program, you should set the memory use byte according to this table:

Value	Reserves
1	Hi-res graphics
2	Hi-res text
4	Hi-res GET/PUT

If you wish to use hires text and graphics, for example, you would add the two appropriate values together. The value

for the memory use character of the string would then be CHR$(1+2).

It is important for the string you pass to 'PLYLODSW' to be exactly 14 characters long. If it is otherwise, 'PLYLODSW' will become suspicious and abort the loading process, sounding a BOOP to indicate this.

4.3 Using PLYLODSW to play a sample

You must also pass a string to PLYLODSW in order to play a sample. The string must be in the following format:

Location	Length	Meaning
0	1	Play/load flag. 1 for play
1	1	memory use byte
2	2	play rate (cycles per byte) See Section 5.4
4	1	Block offset into sample for start of play
5	2	byte offset into first block
7	1	block offset into sample for end of play
8	2	byte offset into end block

The memory use byte is defined as with the load function.

You might use PLYLODSW from BASIC in the following way:

```
10 A$ = CHR$(1) + CHR$(0):CHR$(0) + CHR$(105) +
CHR$(0) + CHR$(0) + CHR$(0) + CHR$(2) + CHR$(3) +
CHR$(135)
20 DEFUSR = &H7XXX:REM Adjust to actual address
30 DV$ = USR(A$):REM DV$ = dummy variable
```

Note that PLYLODSW must be in memory (ie: must already be LOADMed) before you execute it with a USR function call.

The string must be exactly 10 characters long, or PLYLODSW will sound a BOOP and return to BASIC

When a double byte value is called for, (the two byte offsets), you must break up the value into byte-sized parts. If the value were 1234, you could use this formula to find the two single-byte values that PLYLODSW would recognise:

```
FIRST BYTE = INT(1234/256)  = 4
SECOND BYTE = 1234 AND 255 = 210
```

Since Studio Works' samples are saved in 8k blocks, the play routine in LODPLYSW finds it convenient to be told what to play in terms of these 8k blocks. You could have for instance digitized the words "CASSETTE ON OFF" with Studio Works, then used the block markers and the INFORM function to find out where, in terms of 8k blocks and offsets into the blocks in bytes, the three distinct words began and ended. You could then define three strings, say C$, N$ and F$, whose block and byte start/end values corresponded to the three separate words. Your BASIC program could then 'say' "CASSETTE ON" with the following code fragment:

```
DV$ = USR(C$):DV$ = USR(N$)
```

(Of course, PLYLODSW must be loaded, and the USR function defined with DEFUSR.)

For more information on USR, DEFUSR, and how data is passed back and forth between BASIC and ML, see your "Colour Computer 3 Extended BASIC" manual, on pages 321-233.

4.4 Incorporating sound into M/L

A look at the "PLEQU.ASM" file on the Studio Works Distribution disc will tell most of what you need to know to load and play samples using the PLYLODSW program.

The "PLEQU.ASM" file contains solely EQUate pseudo-ops in standard EDTASM+ ASCII format. If you include this code into your own source code, and adjust the

value of the PLYLOD variable to the first address of the PLYLODSW program in memory, you will be able to access all of PLYLODSW's functions from your own assembled program.

Note that no provision is made for loading PLYLODSW. You must do that either from BASIC before EXECuting your own program, or in any other fashion that suits your style and experience.

The very first entry in "PLEQU.ASM" is BASIC JMP PLYLOD. This routine is the one that handles variables passed from BASIC, and is at the top of the file to facilitate execute address calculations from BASIC. To play or load a sample, you must define a structure, and pass a pointer to it, according to the tables in the previous section. Use the entry point defined in PLEQU.ASM, MLENT, and pass the pointer in the 'X' register. No error reporting is done for load, since an error will return to BASIC since PLYLODSW uses BASIC ROM routines to open and read the files. The Play routine will sound a BOOP if it doesn't like the data you give it. If you attempt to play a sample with good data in the play variables, but no sample was loaded, the Play routine will play the garbage already in memory, which will sound like a high-pitched squeal.

The MEMUSE variable is explained in the previous section, see .

Access to some other useful functions is also provided. These entry points accomplish the following:

SNDON	Turn on 6-bit sound
SNDOFF	Turn off 6-bit sound
BUILD	Return list of available MMU blocks according to MEMUSE byte.

Alternative loading method

The play routine source code is also found on the disc, and also in hardcopy within this manual. Both of these routines are intended for use in your own programmes. You could use the sector dump function of Studio Works to get the data on the disc and Disc BASIC's DSKCON routine to read it

back, if error control were essential to your application. See your "Colour Computer Disk Systems Owner's Manual and Programming Guide" for information on how to use DSKCON.

5. Technical reference

The programmes and files on the distribution disc.

STUDIO.BIN	The Studio Works loader.
SW.BIN	The main Studio Works program.
GFX2.BIN	Graphics support library (for Studio Works' use only)
PLYLODSW.BIN	The BASIC/ML interface program.
PLEQU.ASM	The EQUate pseudo-op file for use with ML.
DEMO.BAS	A BASIC demo program.
SAMPLE.SW	The demo sample.

5.1 The record routine source

```
00100 *RECORD.ASM          5-bit capture routine from Studio Works
00110 *
00120 *
00130 *      This routine is copyright (C) 1989 by
00140 *                Oblique Triad
00150 *      32 Church Street, Georgetown, Ontario
00160 *            CANADA, L7G 2A7
00170 *              416 877 8149
00180 *
00190 *(Written by Jeff Noyle, Aug '89)
00200 *
00210 *Permission is granted for legitimate owners of Studio Works (C)
00220 *to use within their own programs, in whole or in part, for any
00230 *reason other than commercial gain.
00240
00250
00260
00270
00280 RECORD    LDX    #$8000  Start recording here
00290
00300 *you should also set the delay rate (TRATE)
00310 *somewhere in here. If TRATE=0, the routine
00320 *takes 105 cycles per byte.
00330
00340
```

```
00350             LDA     #$FF     \ Use direct addressing
00360             TFR     A,DP     / for IO, since fastest
00370
00380             CLR     $FFD9    ensure double speed mode
00390
00400             LDA     #$34     Setup
00410             STA     <$23     hardware
00420             STA     <$1      to point DAC
00430             ORA     #$8      and MUX
00440             STA     <$3      to left port
00450
00460
00470 *At this point you must map in the first MMU
00480 *block to be used at $8000. Use therefore
00490 *MMU register $FFA4 for task 0, or $FFAC
00500 *for task 1. If you change the MMU register,
00510 *you must also change the values for the X
00520 *register throughout this routine.
00530
00550
00560 *The number in the comment field represents
00570 *the number of cycles each instruction takes.
00580
00590 RECLOP    LDA     #32*4+2 2        First try in DA
00600             STA     <$20    4        output it
00610
```



```
00620
00630             LDA     #48*4+2 2        assume its higher
00640
00650             LDB     <0      4        is it
00660             BMI     DONE1   3        higher?
00670
00680             LDA     #16*4+2 2        , if not take back assumption
00690                                       and go lower.
00700 DONE1       STA     <$20    4        2nd try
00710             ADDA    #8*4    2
00720             LDB     <0      4
00730             BMI     DONE2   3
00740             SUBA    #16*4   2
00750
00760 DONE2       STA     <$20    4        3rd try
00770             ADDA    #4*4    2
00780             LDB     <0      4
00790             BMI     DONE3   3
00800             SUBA    #8*4    2
00810
00820 DONE3       STA     <$20    4        4th try
00830             ADDA    #2*4    2
00840             LDB     <0      4
00850             BMI     DONE4   3
00860             SUBA    #4*4    2
00870
```

```
00880 DONE4     STA     <$20    4       5th try
00890           ADDA    #4      2
00900           LDB     <0      4       note also that this load
00910                                   prepares for button test
00920           BMI     DONE6   3
00930           SUBA    #8      2
00940
00950 DONE6     STA     ,X+     6       save the value, point to next
00960
00970           LDA     TRATE   5       delay rate preparation
00980 RDLP      DECA            2       \  do
00990           BNE     RDLP    3       /  delay
01000
01010
01020
01030
01040           BITB    #1      2       Test button,
01050           BEQ     RECXIT  3       Out if down
01060
01070
01080 RECOR6    CMPX    #$A000  4       check for end of MMU block
01090           BNE     RECLOP  3       no, so go get more
01100
01110 *In here you must allocate and map in a fresh
01120 *MMU block for the sample to be recorded into.
01130 * This block should be mapped in at $8000,
```



```
01140 *so use $FFA4 for task 0, and $FFAC for task 1.
01150
01160           LDX     #$8000          point to beginning of block
01170           BRA     RECLOP          go get more
01180
01190 RECXIT    RTS                     bye bye
01200
01210 *It is important to store values representing
01220 *how long the sample is. The suggested method is to
01230 *store the number of MMU blocks, and the number of bytes in
01240 *the last block. Find the number of bytes from register
01250 *X - $8000
01260
```

5.2 The play routine source code


```
00100 *PLAY.ASM          Play sample routine from Studio Works
00110
00120
00130 *         This routine is copyright (C) 1989 by
00140 *                    Oblique Triad
00150 *         32 Church Street, Georgetown, Ontario
00160 *                  CANADA, L7G 2A7
00170 *                    416 877 8149
00180 *
00190 *(Written by Jeff Noyle, August, '88)
00200 *
00210 *Permission is granted for legitimate owners of Studio Works (C)
00220 *to use within their own programs, in whole or in part, for any
00230 *reason other than commercial gain.
00240
00250
00260
00280
00290
00300 *You must supply a list of MMU blocks that contain the
00310 *sample, the number of blocks in that list, and the
00320 *number of bytes in the last block.
00330
00340
00350 ENDADD    RMB     2               End address
00360 BLKCNT    RMB     1               # of blocks
```



```
00370
00380 PLAY      LDD     ::::::          User supplied play rate,
00390                                 expressed as # of machine
00400                                 cycles per byte of sample
00410
00420           SUBD    #40             Play routine is 40 cycles
00430                                 long, so find out how
00440                                 many cycles more is the
00450                                 desired speed.
00460
00470           JSR     DIV05           Since delay loop is 5 cycles
00480                                 long, find out how many loops
00490                                 through the delay routine
00500                                 add up to the difference
00510                                 in rates between the desired rate
00520                                 and 40 cycles/sample
00530
00535           INCB                    0 would be 255 loops
00540           STB     TRATE           And store it.
00550           LDA     ******          User supplied block count.
00560           STA     BLKCNT
00570
00580           LDD     ------          User supplied # of bytes
00590           ADDD    #$8000
00600           STD     ENDADD
00610
```

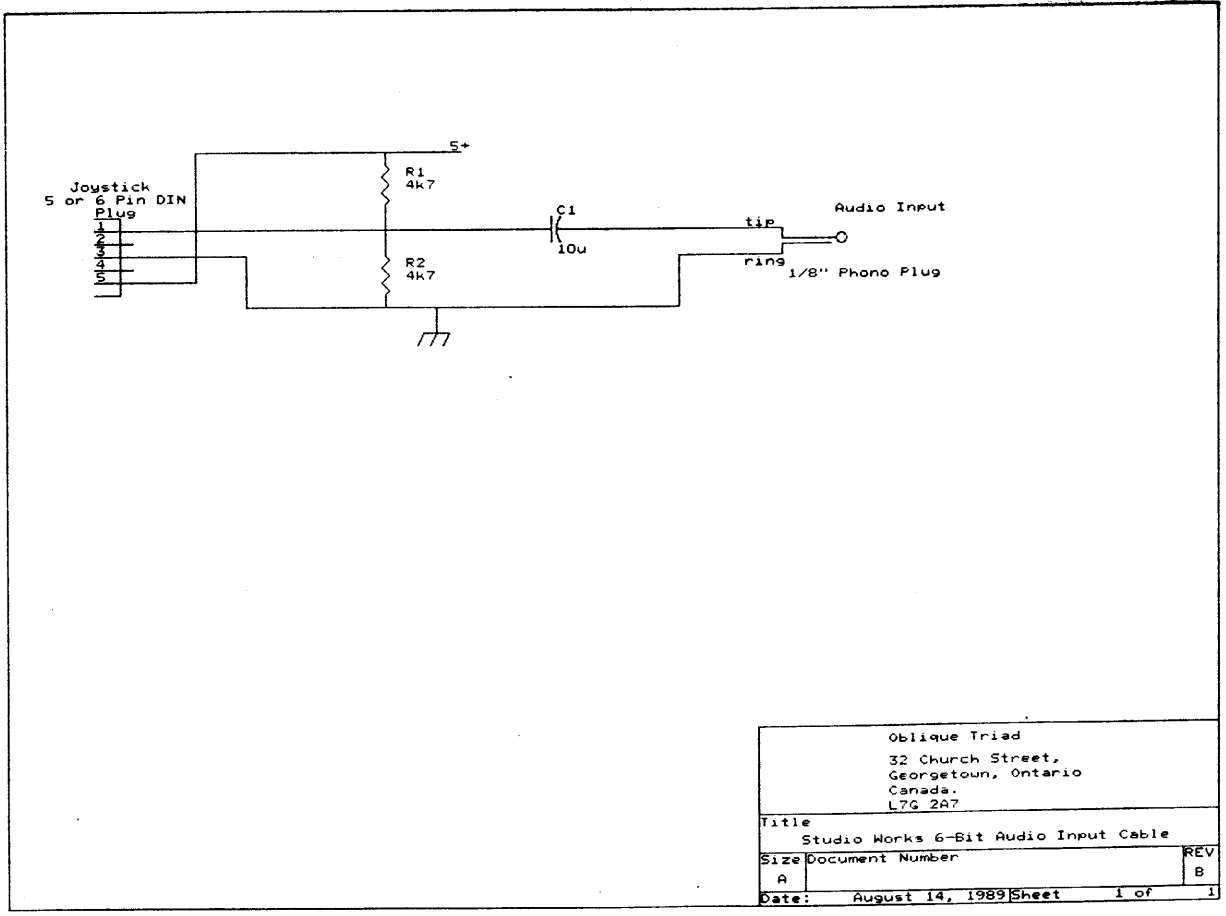
```
00620
00630              JSR     SNDON           Enable 6-bit sound
00640
00650              LDY     #......         address of user supplied list
00660                                  of blocks
00670 JOYP5        LDA     ,Y+             get the first block
00680              STA     $FFA4           and set it up
00690
00700              LDX     #$8000          point to first byte
00710
00720 JOYP3        LDB     ,X+     6       get the value
00730
00740              STB     $FF20   5       put it in DAC
00750
00760
00770              LDB     TRATE   5       fetch the delay,
00780 JOYP4        DECB            2       and...
00790              BNE     JOYP4   3       ....do it
00800
00810              TST     LOPCNT  5       last block?
00820              LBNE    JOYP6   4       no, so go (use LBNE to make
00830                                         40 cycles)
00840
00850              CMPX    ENDADD  7      Last block, so see if up
00860              BNE     JOYP3   3      to last byte in last block
00870
```



```
00880              JSR     SNDOF           Turn off sound
00890              RTS                     and leave
00900
00910 JOYP6        CMPX    LA000   7       Compare X using extended
00920                                  not immediate, so last block
00930                                  will be played at same speed
00940                                  as others
00950              BLO     JOYP3   3       go do more
00960              DEC     LOPCNT          We've done 1 more block
00970              BRA     JOYP5           go do the rest
00980
00990 LA000        FDB     $A000           Dummy variable for end of loop
01000                                  test
01020 *Turn off sound
01030 SNDOF        LDA     $FF23
01040              ANDA    #$FF-8
01050              STA     $FF23
01060              RTS
01070
01080 *Turn on sound
01090 SNDON        LDA     $FF01
01100              ANDA    #$FF-8
01110              STA     $FF01           set up MUX for speaker...
01120              LDA     $FF03
01130              ANDA    #$FF-8
01140              STA     $FF03           ...and DAC.
```

```
01150             LDA     $FF23
01160             ORA     #8
01170             STA     $FF23    enable 6 bit sound
01180             LDA     #$34
01190             STA     $FF21
01200             RTS
01210
01220 *Divide by 5 routine, used for determining delay rate
01230
01240 DIVCNT      FCB     0        Divide routine counter
01250
01260 DIVD5       PSHS    D        Save Dividend
01270             LDA     #8       count
01280             STA     DIVCNT   setup
01290
01300             LDD     ,S++     get divisor back
01310 DIV51       LSLB             prepare for test subtract
01320             ROLA
01330             CMPA    #5       test subtract divisor
01340             BCS     DIV52
01350             SUBA    #5       subtract it
01360             INCB             set bit in quotient
01370 DIV52       DEC     DIVCNT
01380             BNE     DIV51    do 8 bits
01390             CLRA             remainder not needed
01400             RTS              bye bye
```


5.3 The cable schematic



5.4 Play Rates

Play rates (in kHz)	Cycles/ Byte	Max Sample Length (seconds)		
		128k CoCo	256k CoCo	512k CoCo
44.75	40	1.5	4.4	10.3
39.78	45	1.6	4.9	11.5
35.80	50	1.8	5.5	12.8
32.55	55	2.0	6.0	14.1
29.83	60	2.2	6.6	15.4
27.53	65	2.4	7.1	16.7
23.87	75	2.7	8.2	19.2
21.06	85	3.1	9.3	21.8
19.89	90	3.3	9.9	23.1
17.90	100	3.7	11.0	25.6
17.05	105	3.8	11.5	26.9
14.92	120	4.4	13.2	30.7
13.77	130	4.8	14.3	33.3
12.79	140	5.1	15.4	35.9
11.55	155	5.7	17.0	39.9
10.53	170	6.2	18.7	43.6
9.94	180	6.6	19.8	46.2
9.18	195	7.1	21.4	50.0
8.33	215	7.9	23.6	55.1
7.31	255	9.0	28.0	62.8
6.28	285	10.4	31.3	73.0
5.19	345	12.0	37.9	88.4

The Values are calculated with these formulae:

$$f = \frac{1.79*10^{6}cs^{-1}}{C}$$

$$C = \frac{1.79*10^{6}cs^{-1}}{f}$$

Where f is the sample frequency in kHz, and C is the number of 1.79 MHz clock cycles per byte.