```
***********************
* Trisyd Video Games *
*   Dynomite Sound   *
*     Digitizer      *
***********************
```


Trisyd Video Games
26 Florence Cres,
Toronto, Ontario.
Canada. M6N 4E4
(416) 769-6446

The Dynomite Sound Digitizer enables Adam computer owners the capability to capture audio signals and convert them to computer readable data. Once edited, the sound data can playback on a normal Adam without use of the digitizer. The use of this device spans area's such as voice mail, special effects and audio applications that were too complex to do before. Since the playback feature does not require use of the digitizer, any Adam with a functioning sound chip can use your creations!

Installation:
The unit comes complete as a game cartridge styled device. Turn Adam's power off then simply plug the digitizer cartridge into your game slot as with any game cartridge. Once installed, turn Adam on and load the driver software from disk/ tape. The game reset switch is not supported as a software loader yet the computer reset switch will still operate in the normal fashion.

The Audio signal connects into the RCA type jack at the cartridge front. It's strength must be no more than 4 volts maximum. Typically it would be the 400 mv audio output from the unamplified section of a radio tuner or tape player. These jacks are readily accessible and do not require any additional wiring. Do not connect directly to speaker output signals as these are too high in power and will damage the unit. Input signal cable is not supplied, it is an unnecessary expense as most persons have ready access to this type of cable.

Concept of Use:
As manuals tend to become too boring to read over, this conceptual overview should get you started to begin using the digitizer. The detailed feature explainations are after this. At software startup, your placed into the sound editor menu. The computer memory is set at maximum, void of any recognizable data and ready for sound data to be loaded from disk or the digitizer cartridge.

Keypress sprites detail the allowable features with an attempt to iconize the function. Load and Save functions are accessed with the L, S or Store/Get key. These icons show a disk, a block and an arrow pointing to or from the disk depending on the data path. The G or P icons dictate that you can Grab sample data from the cartridge or Play the defined block. The remaining icons are for gain set up via the O (Oscilloscope), Graphing output (B), and editing functions. The (plus) + and (minus) - keys set sample rate, block sizing requires use of the arrow keys. These two functions allow you to define what rate you will sample / playback and where the sound data resides.

Initially you want to set the gain control before sampling. This enables you to best fit the input signal. Pressing the Oscilloscope (O) key will invoke another screen containing a real time waveform analyzer. Selecting A or G from within this menu will change the output to Audio or Graphic. Adjust the gain control knob on digitizer cartridge front to arrive at the best possible waveform with the least amount of distortion. Essentially you eliminate the flat spots in Graph view or the "Cluck" noises in Audio mode. Sample rate can be altered using the (plus) + or (minus) - keys. It takes longer to adjust this in this menu due to sample quantities inherent in the software.

Adjust gain for optimum signal then exit this menu using the Undo key. Back in the main editor, select G to begin grabbing the sound and P to here the result. This will get you started, I still recommend reading onward to gain detailed information on the software functions and keypress's not mentioned. BE SURE TO READ THE COMPUTER LOCKUP PARAGRAPH!

Software functions:
The main menu display comprises of 5 distinct areas. These can be named Status Line, Sample Graph, Info Line, Keypress Sprites and Information Window. These area's convey different aspects of the program operation.

Status Line:
Located at the display top. It conveys information such as the start and end sample data parameters (called start block and end block), Block size and Zoom factor. Other program operations will display status messages here too. Typically reserved for status information.

Sample Graph:
This area is 64 lines wide top to bottom and graphically represents the sample in memory. The window size spans the entire screen. The amount of information shown depends upon the Zoom factor (explained later). Two data representation types supported are Bar graph and Dot graph. Toggle between these types using the B key as indicated in the Keypress Sprite area. Yellow and Red bars highlight possible waveform clipping.

Info line:
Located just below the Sample Graph window. It displays 1 line information such as sample rate.

Information Window:
At the lower right hand corner. It is a 13 line display used when detailed information is to be conveyed to the user.

Keypress Sprites:
Fifteen keypress sprites are installed here, designed in an attempt to iconize the function they represent. Starting from top left to top right then down a line, their functions are as follows.

(O)scilloscope:
Press O and change menu's to the gain adjustment screen. A 128 line waveform analyzer replaces the editor menu. Instructions provided at this screen show the allowable keypress features.

(B)ar/ Dot Graph:
Toggles graph display from Bar type to dot. Sometimes a bar type becomes too cluttered therefore use the dot representation. The graph data fills the entire window when using Dot mode. When in Bar mode, the graph stops after exceeding data memory. When encountering end of memory in Dot mode, a straight line replaces non existent data.

(I)ntelligence:
This feature is not fully supported with release 1.0, future releases will see a scripting type feature that can store multiple block markers. The end use will be for the creation of complex melodies. Such uses may be for storing the entire 64 allophones that make up human speech and then playback to make up entire words. Currently installed is an actual memory location identifier in respect to your start and end blocks. This is more a feature for the person wishing access to their sound data in a way not yet supported by the software.

(G)rab sound:
As implied, this feature "Grabs" the digitized incoming sound fed through the RCA jack and stores it into memory. Your Block markers define the sample size. As mentioned later, sample rate will dictate the quality of signal.

(P)lay sound:
Play the defined block through Adam's sound chip.

(Esc)ape: Abort feature.
Terminates any in progress function except play and grab sound.

(D)elete Block:
Also supported by the Delete key, this feature will remove your sound block from memory and shift the remaining data afterward into the deleted space. Once deleted, your sound data cannot be recovered. This may pose an inconvenience but due to memory limitations, it became necessary. Before the actual delete takes place, your warned through the status line. Press Y to continue and N or the Escape key to abort.

(R)eplace Block: -- Or can be called copy block --
Used to overwrite sound data at the current block pointers. When pressing R, you're presented with a replace block set of keypress sprites. These exhibit the same functions as they do in the editor menu. Once you confirm that your intention is to replace data, the current block pointers are stored, labelled blk #1. You're now prompted to select the data area used to replace the blk #1 area.

Two new sprites are introduced in this menu, they are a pointer hand and ¬P. The pointer alerts you that the (P)lay sound sprite is the keypress used to listen to the current pointers. The ¬P sprite, showing pressing the control key and P together, will play blk #1. It provides an audible confirmation of the initial sound your intending to replace. Otherwise, use the indicated keypress functions to help in locating the sound block to use for replacement data.

The replacement size is the initial block size. If your replacement block is larger than blk #1 then it will be truncated. If smaller then it will not fully overwrite the original data. When you are ready to begin the actual replace feature (Blocks defined), press R.

(M)ix block:
The same idea as replace block except that it preserves your data. The mix feature takes the original data and inserts the secondary data at every second byte. This way it is not feasible to mix data into a previously mixed area. The second mix sequence will eradicate the previous mix. Future revisions may incorporate various mix rates such as every 3rd or 4th etc bytes to resolve this.

(<->) Start Block adjustment:
Use of the right and left arrow keys will adjust the start block pointer. As the adjust takes place, the status line shows the revised pointer. If the graph window view covers the start block area then a marker sprite will appear. This provides a visual aid in identifying sound parameters.

(Cntrl <->) Display graph from start/ end block pointer:
Pressing the control and right or left arrow keys adjusts the graph display window. Control - Left arrow ensures inclusion of start block, Control - Right arrow insures end block. (Uparrow/Dnarrow) End Block adjustment: Same as start block adjustment except that it adjusts the end block pointer. Same holds true for seeing the marker sprite in the sample area.

(Cntrl Uparrow/Dnarrow) Max/ Min block settings:
This is a quick way to maximize your block pointers or to minimize them based upon the zoom value. Maximize block pointers by pressing Control and Up arrow, Minimize block pointers by pressing Control and Down arrow.

(HOME) Display block pointers/ block size:
Home key brings the current block pointers into the status window area. The format is Start block/ End block and block size (Based in Hexadecimal notation).

(L)oad block from disk:
Transfers data from Hard drive/ Disk or Datapac into the defined block area. Once selected, your prompted for the filename. After this you have three selections. These dictate whether the loaded data will be confined to the defined block pointers; begin from the start block and load file up till memory / file end or load file from start of available memory up till memory end. After loading the file, your block pointers will be adjusted. The Store/Get key in "GET" keypress (No shift or lock key) will activate this also. Load file will overwrite any data in the block area.

(S)ave block to disk:
The same as Load block from disk except the data path is from your defined block into a disk file. Be aware that the software automatically appends an H after the filename. Version 1.0 of the software does not support file renaming or deleting. If you already have a file of that name then expect to encounter an error. File size cannot exceed available disk space. This feature also begins by use of the Store/Get key in the "Store" keypress (activated by using the shift key and store/get).

(E)nlarge view area:
This key will increase the zoom factor value. The result is an increase in the displayable data in the graph window but with less resolution.

(C)ompress view area:
This key will decrease the zoom factor value resulting in a decrease of displayed data but greater resolution.

(Z)oom Factor display:
To display zoom factor without causing alteration then use the Z key. Zoom Factor spans from 1 to 8000 Hex (32768 bytes per pixel). At zoom factor of 1, resolution is such that each graph element represents 1 data byte. With a zoom factor of 2, every second data element is read and placed into the graph window. Therefore a zoom factor of 2 will display 512 bytes in the graph window. The Zoom factor is presented in a Hex number format. Therefore Zoom Factor of 10 is equal to times 16. The max zoom value of 8000h provides an effective view area of 128 64k banks or an 8 megabyte window. Should a ram disk become available which supports a greater memory size then this will be incorporated into future releases.

(Shift-Clear) wipes out entire sound data in memory.

Sample rate:
Pressing either + or - changes the sample rate. The displayed format is a hexadecimal based counter. The greater the count the lower the digitizer's upper frequency thus less quality output. The high sample rates conserve memory yet decrease the highest frequency that can be accurately digitized. Low sample rates enable better representation of high frequency sound.

How it works:
The theory of operation is very simple. Essentially the frequencies of Adam's three sound chip voices are set to maximum. This frequency is inaudible or it appears as a high pitched hiss (Mind your dog). Once set, their volume levels are adjusted to produce the sound waveform. The digitized data represents a volume level. I have devised a table which references the three sound chip volume levels corresponding to the digitizer data. The help file on the distribution disk contains this table. By sucessfully applying the digitized data to the sound chip voices you recreate the waveform.

Sound File format:
This may change in subsequent releases of the program yet to date the format is as follows. Byte #1 is the playback rate, Byte #2 is the input sample rate, Byte #3 is the sound data format type and Bytes #4 to file end are the unmodified data. Right now there is only one data format type but provision now will benefit in later releases. A reasonable cause for change in the data area would come in using a different data range. Instead of the 0 - 46 range you may use something playable on other machines.

Byte #1 ranges from 0 - 31. It is contains a counter to be decremented to zero before sending each successive data byte to the sound chip. Byte #2 is utilized the same as byte #1 except is controls how soon the input data is sampled. Byte #3 is equal to 1 for normal 0-46 range. I am predicting that file format 2 will come available soon. This will be be represented as number 2. The remaining bytes 4 onward, are raw sound data as explained elsewhere.

Sound Table use within your own programs:
Step #1 - Retrieve a sound data byte from your sample.
Step #2 - Use this byte as an offset into the sound data output table.
Step #3 - Retrieve the 3 successive bytes in the table at the offset.
Step #4 - Place these values into the sound port
Step #5 - Repeat Steps 1 - 4 until your sound data is empty.

This process is automated by use of the machine language overlay file and basic program supplied on the distribution disk. The machine language overlay resides at memory location 27800. It is set up so that you program the start location of the sound, the output rate and the data length. Then call location 27800 to play that sound. The key memory locations are as follows:

```
  27800        : Call to this location to play sound
  27807        : Sound data location (2 bytes)
  27809        : Sound data length (2 bytes)
  27811        : Output rate (1 byte)
```

The overlay accepts hexadecimal numbering formats. This is in low byte/ high byte format. Example, take a data location of 30728, length of 512 bytes and output rate of 28. The resulting pokes would be 27807,8: 27808,120: 27809,0: 27810,2: 27811,28. The reasoning for these numbers is as follows:

```
Sound Data location at 30728:
            converted to hex: poke 27807,30728-((int(30728/256))*256)
                              poke 27808,int(30728/256)
Sound Data length of 512:
            converted to hex: poke 27809,512-((int(512/256))*256)
                              poke 27810,int(512/256)
Output rate is a straight through poke:
                              poke 27811,28
```

Initiate frequency and sound off commands are called each time you call to the overlay. If you wish to bypass these then refer to the machine language source code. There you will find the hook vectors to change.

Product Limitations:
The intention of the digitizer is to convert audio signals into digital form that can then be used by the computer. In this process there are a few factors that will affect the quality of sound capture/ playback. These are outlined as follows.

Signal Parameters: The maximum input signal can be up to 4 volts. This may not impose a limitation unless you do not have access to the output stage of an unamplified audio signal. The maximum frequency is dictated by the sample rate. Due to the lower clock frequency of Adam (3.58 Mhz) and the wait state implementation generated at each memory cycle, the effect sample rate is significantly reduced. This is within the digitizer software more than can be expected when using a sound player only. When using a sound player, playback speed can be increased when there is no check for memory boundaries where by a simple loop can be set up for playback.

Waveform conversion:
Although there are 8 bits in a byte and therefore an effective conversion from 0 - 255 possibilities, the digitizer only uses numbers 16 - 255 or 240 possibilities. This may seem like a restriction at first but it isn't since there are only 46 output levels used on playback. In effect, with the exclusion of the lower 16 numbers, special effect coding can be incorporated into the samples with little effort. This would be in the form that sample data equal to 15 or less is to be used as an effects code. One effect comes to mind, a repeat loop. The next revision of software may take advantage of such capabilities. There is another method of waveform conversion. This other directly converts the digitized value to a 1-15 value and then output that to all three sound chips.

Block precision:
The Start Block can be altered in steps of 1 byte or 1 data element. The end block is different. End block pointer can only be altered in increments of 256 bytes.

Depending on the actual memory locations the software uses for your data, the minimum block size may be anywhere from 1 byte to 255 bytes. The end block incremental limit is a trade off to gain output rate speed.

File Saving/ Loading:
You must have enough free space on your storage media in order for the sound data to be saved. When loading a sound file, the data will be read from the start and continue until file end. Future revisions may see a rolling buffer that would allow selection of data anywhere within the file.

Computer Lockup:
The input device (be it a tape player, phonograph or radio etc..) usually will be a two prong electrical cord device. I have found that my Adam will lock up during file loading if the input signal coming from one of these device types remains connected to the digitizer.

This problem only occurs with some equipment. I've concluded the reason being that the ground wire in your computer and that of the input signal device are carrying different noise signals which creep into the computer during DMA operations. Before you preform any file input or output, be sure to disconnect the signal from the RCA jack to avoid this lockup. Should you forget and the machine does lockup, reboot the software as most of your sample will still be intact.

Conclusion:
The software may have already been updated since the printing of this document. In that case, a file is enclosed on the disk/ datapac that details the updates. Please forward any comments/ suggestions to Trisyd Video Games for inclusion into future revisions.

Thank you for supporting yet another product by my company.

Disclaimer:
Trisyd Video Games will not accept responsibility for the use of this digitizer device or any malfunctions that may result due to it's use. We offer consultation in these area's.