TFOFTHYPERSOFTHYPERSOFTHYPERSOFTHYPERSOFTHYPERZAP!HYPERSOFTHYPER
H S
Y Another great program from Copyright 1985 O
P HYPERSOFT PO Box 51155, Hypersoft. S
E Raleigh, NC, 27609 USA R
R E
S P
O (Press CLEAR to start) Y
F H
T...The Ultimate TRS80 Disk Utility for Models I III & IV...SOFT

Version 3.2

H Y P E R Z A P I N S T R U C T I O N M A N U A L

Rev 3.2 Jan 1st 1985

W A R N I N G !

HYPERZAP is a disk zap program and as such is capable of modifying your disks. It is your responsibility to read the manual and take any precautions neccessary to protect any data you value. At the least, any master disks should always be write protected. You may also modify any part of memory including HYPERZAP itself. You do this at your own risk. Hypersoft and the Author offer this program on an as-is basis only and assume no responsibility for any loss or consequential damage resulting from its use or misuse.

==========

1 . 0 W H A T I S H Y P E R Z A P ?

HYPERZAP is a powerful TRS-80 disk utility which allows you to create, modify and backup part or all of any floppy disk. It will work on any disk that your hardware has the capability of reading even those with sectors of different densities on one track. (Model I Owners must have a doubler for this). It will also work on many non-TRS-80 disks. It works on tracks and sectors and does not care how the informatian is encoded within the sectors. The program is resident in memory and does not use a resident DOS so it will run on any model I III or 4 without requiring the user to have anything special beyond the basic hardware. 48K of memory is required and Model I users should have a doubler if they want to do anything with double density which now includes many of the newer mixed density dual purpose Model I/III disks now on the market. On model I systems the program automatically detects the type of doubler in use if any. It will work with Aerocomp, Holmes, LNW, Percom and Radio Shack doublers except as follows: Doublers with 8 inch capability such as the LNW 5/8 and Holmes DDSD1 change mode by writing to the sector register with the most significant bit set. Because of this you will get unpredictable results if you try to copy sectors with designations above 7F. The Radio Shack doubler changes density and selects write precompensation by writing to the sector register address also, with the most significant data bit set. This program checks for this and will not allow certain sector designations above 80 hex to be read or written using an R/S doubler. In fact, very few disks exist at the present time which make use of sectors labelled 80 and above so this is not a severe restriction.

HYPERZAP is designed with many features to make it easy to analyze or backup an unknown disk. Because of the many ways that information can be laid down on a disk, the user must be prepared to learn something about the way tracks are formatted. Only with this knowledge can the full features of HYPERZAP be taken advantage of. Among the many features of HYPERZAP are:

- Works in SINGLE, DOUBLE and MIXED sector densities so you can work with disks designed to boot on Models I, III and 4 (in the III mode). These disks often have both single and double density sectors on one track.
- DOUBLE SIDED disk drives supported as two single sided drives.
- 80 track drives supported.
- 8 inch drives supported (Model III and 4). A suitably designed disk controller is needed for this. The standard Radio Shack board will not work. Holmes and Micro-Mainframe are among the suppliers of suitable controllers. Owners of the newer versions of the Model 4 (with the green screen) and and the 4P will not be able to use 8 inch drives because the floppy disk controller is integrated on the main CPU board and you cannot fit a different one.
- Analyze a track to determine the format.
- Read/Write tracks sectors.
- Read/Format a whole track.
- Build a directory table describing each tracks statistics.
- Edit memory, sector and track data.
- Move, find and fill memory, calculate CRCs.
- Increment, decrement and complement memory.
- Compare memory regions and find area that matches.
- Backup a disk.
- Autopilot. This exciting new feature allows HYPERZAP to record any sequence of actions for future re-use. Your HYPERZAP disk comes with several examples enabling you to back up some of the most well known hard-to-copy programs.
- Make your own dual Model I/III,4 self-booting disks. If you have developed your own machine language program and you want to put it on a disk to run without a DOS then this feature is for you.
- Screen print for permanent record use.

Backup has many features not the least of which is that special consideration has been given to single drive users. The general backup analyzes the source disk track by track, determining the number of sectors, their kind and their angular position on the disk. This information is put in the directory and after reading the sector data the destination disk is formatted in the same way and the sector data written. Single drive users should note that there is enough free memory to transfer about 6 single or 4 double density tracks per disk swap.

2.0 GETTING STARTED

There are two versions of HYPERZAP, one for the Model I and one for the Model III/4. Both are on your disk and may be removed and made into command files if you wish. The program is supplied as a self booting disk which will auto run when you hit reset. When you run the program it will load and then relocate part of itself down over the top of DOS which is no longer needed. This way the maximum space is available for track and sector data storage. Do not use any part of memory between 4000H and the top of the program for your own purposes (see H command) or strange and unpredictable things will happen !.

Although the program has been provided on a self booting disk you may prefer the flexibility of using it as a command file instead. This has the advantage of being on your DOS and easily transferred from disk to disk. However there are three disadvantages: 1) You cannot use the self booting disk maker feature from a /CMD file version; 2) you will have no access to the supplied autopilot programs unless you insert your Hyperzap master and 3) Hyperzap loads faster from the master disk than from a DOS. A special feature has been built in to make it easy for you to transfer Hyperzap to the DOS of your choice in Model I or III mode.

To make yourself a command file, insert your Hyperzap master in Drive :0 and press reset to load. At the same time, hold down the three keys MJG simultaneously while the program is loading. Instead of the normal greeting coming up you will see a command such as:

DUMP filespec (START=X'9FF8',END=X'D33B',TRA=X'D302')

which is the form of the Dump memory to disk command in TRSDOS. What has happened is that after loading the program a copy was also transterred into high memory together with an appendage. If you were to start running at the address D302 the appendage would copy the entire program back and start running it. At this point you must take your Hyperzap disk out and replace it with a disk containing the DOS of your choice. Now press the reset button on your computer to reboot and then type in the command as above substituting a suitable file name and the exact numbers as they appeared on the screen. The DUMP command is different in some DOSs so check your manual and adapt as neccessary. Note that START is the beginning, END is the end and TRA is the transfer address of your copy of Hyperzap that you must save to DOS.

HYPERZAP has three major screen modes. The main mode which comes up at the program start has a menu of major options most of which are selected by keying a single letter. Also included on this screen are the specifications for the source and destination drives which can be configured for your machine through the parameter change option. The two other main screens can be reached by typing D or I as below.

D nn - Display Track Sector Table.

Pressing "D" followed by a 2 digit number will take you to the Directory screen that displays the sector configuration of a particular track. This is a very important feature which allows you to analyze and understand the makeup of an existing disk or to create a new one. Initially this information screen will contain no data until either you read an existing disk or you enter same sector descriptions yourself using the Append, Insert or Edit options. Given valid data from a real disk you can point to a particular sector in the directory with the cursor arrow, read the sector data into memory, edit it, and rewrite it. Unlike many DOS supplied Zap programs HYPERZAP does not care whether the disk is in a standard format. You can, for example, edit a double density sector residing on an otherwise single density track.

The number you type for the track may be in decimal or hexadecimal (signified by $ sign) thus D 10 and D $0A will both take you to the entry for track decimal 10. Note that 2 digits must always be typed.

I hhhh - Memory Inspect/Edit

Back in the main menu again, pressing "I" followed by a 4 digit hexadecimal memory address, will take you to the third screen which is a screen oriented memory inspect and change subprogram with many useful features for creating, copying and editing disk data. This is also called when you do a whole track read from the Main Menu or a sector data edit from the Directory Screen.

H - Help

Every screen has been provided with a (hopefully) descriptive menu of options and, when that option is selected, additional sub-prompts are provided. A helpful feature is provided in all 3 screens where, by pressing the H key, data will be displayed showing the memory occupied by the program and the various data stores such as the Directory, Sector Data and Track Buffer. This tells the user immediately where everything important is and therefore what free space is available. While in the help mode you can send the contents of the current screen to your printer by pressing the P key.

3.0 Main Menu Options

This chapter describes in detail all the options obtainable from the main menu. Each command line can be edited using the backspace (left arrow) key. No action takes place until the enter key is pressed. Note that the symbols d and h are used below to denote a single decimal (0-9) or hexadecimal (0-9,A-F) character respectively. Note also in the examples your input is shown underlined.

A dd or A $hh Read Address Marks

Typing "A" followed by a track number causes the selected drive to seek that track and then make 8 passes, 3 in single and 5 in double density, each time reading and recording any sector ID address marks and the angular position round the disk is measured from the index hole. The resultant information is then checked and averaged and entered in the directory. To see the result you will have to type D dd to display the directory entry for that track.

Example of use: A 03
Explanation: read track 3 to find any sectors that exist and enter them in the track table in proper order as measured from the index hole.

B Generate a self-Booting disk.

This option gives you the ability to take a machine language program and put it on a disk that will boot automatically and run the program on a Model I,III or 4. To use this you will have to prepare your program using an editor and assembler. Once made, you must place a copy of it in high memory starting at A000 Hex. You may want to produce two versions of the program, one for Model I and one for Model III/4. The Model I version will be placed in the single density sectors of your self booting disk while the Model III/4 version will use the double density sectors.

To produce a self booting disk do the following: prepare your program as a command file and load it into memory at A000 Hex. If your file needs to run at some other location use a block move to prepare an exact copy starting at A000. Run Hyperzap which loads into low memory and call the B function from the main menu. Hyperzap will first ask you for the upper limit of your file. From this it can compute how many formatted tracks you will need. Hyperzap then builds a disk from track 1 to track N with 6 sectors of double density and 6 of single density on each track. Then it formats and writes the sector data to disk. You can make your disk model I or III/4 only by laying down data only in the single or double density sectors respectively. If you want both you will have to make a second pass without formatting. Reply S or D to lay down the sectors in single density for Model I or double for model III/4. Insert your destination disk and answer Y to the format question if this is the first pass. Hyperzap will then take your data, format the disk and write the sector data. The program then adds the boot sectors on track 0 by copying them from your master disk. The load and run addresses are modified to suit your own requirements.

Example of use:

Suppose you have a file that loads at 6000H to 6100H and starts at 6030H and you want to put it on a disk so that it will boot and run on a model I. First you must load it into memory from your DOS and move it up to A000H using a block move utility. Then boot Hyperzap from a self booting disk.

Note: your input is shown underlined.

```
   B
1  Generate Mod I/III/4 self booting disk. You must
   execute Hyperzap from a Self booting disk not a CMD file !
   Insert destination disk.  Object code from A000-A100
2  Enter S for Mod I or D for Mod 3/4 File: S
3  Format or not (Y/N) Y
   Formatting track & writing sectors:
   Track 01H SSSSSS  Verifying: SSSSSS Good
4  Insert original Hyperzap disk & press Enter enter
   Reinsert destination disk & press Enter enter
5  Address at which you want program to load (>43FFH)? 6000
   Program start (transfer) address ? 6030
   *
```

Explanation:
1: At the prompt, enter the upper limit of your file.
2: Enter S for Model I, single density.
3: Enter Y as we are starting with a blank disk.
4: Here Hyperzap copies the boot sectors from your master.
5: Before doing so the program load and run addresses are modified to fit your needs.

C Clear Directory

C clears the track sector table of all previously read or created entries. Whenever you read address marks with the A or E commands, entries are placed in the table and remain there until you either clear them with a C command or use the disk copy command XC. If you have previously read a track from a different disk you must clear the table before doing a read of the same track on your new disk. AlSo, if you are examining double sided disks, clear the table when changing sides or you will get a mixture of front and back sectors in your table.

Example of use: C
Explanation: clear track sector table.

D dd or D $hh Display Directory

D switches the display mode to screen 2, Directory display. The directory display will be entered pointing at the track number specified. From there you can page forwards and backwards through the tracks using the Shift-Up-Arrow and Shift-Down-arrow control keys. See section 4.0 for a detailed description of the functions available.

Example of use: D 03
Explanation: Display the track sector table for track 3.

E dd or E $hh Combined A, S, D command

E gives you the ability in one go to read the address marks of a track, read the sectors and then go and display the result. See the descriptions of the individual commands.

Example of use: E 03
Explanation: Equivalent of A 03 S 03 D 03

H Helpful Information

This shows the memory space currently occupied by the program and all the main data storage areas. You need this if you are going to make use of some spare space to build a sector of data. As a general rule all memory between 4000 hex and the top of the track sector directory should not be overwritten. Space above the start of Sector Data is fair game and above E700 also if you do not intend to do a track format.

While displaying in the H mode you can print the whole screen by pressing the P key. Ihe H command can be called from all screens.

Example of use: H
Explanation: calls help information at bottom of screen.

I hhhh Inspect/Modify Memory

I followed by a 4 digit hexadecimal address switches display to screen 3, Memory Inspect/Edit mode. You must use 4 digits and no preceding $ to specify the address. The screen will show the contents of memory in 11 lines of 16 bytes each starting at the address specified. In this mode you can scroll through memory inspecting and modifying it as desired. See section 5.0 for more details of the sub-commands available.

Example of use: I B800
Explanation: Display memory starting at B800 hex.

J hhhh Jump to Memory

J followed by a 4 digit hexadecimal address and <enter> causes the program to jump (tranfer control) to start executing a program at the address specified.

Example of use: J C000
Explanation: start executing program at C000 hex.

O Port I/O

O will prompt you for a hexadecimal 2-digit port number, it will then print the value obtained by reading that port address and wait for you to enter a 2 digit hex value to be sent to the same address. Any non valid entry will abort without sending anything so you can type <enter> if you just wanted to read the port.

Example of use: O
Port I/O - enter Port # (Hex) : F0 80 81

Explanation:
Inspect contents of port address F0 hex and change from 80 to 81 hex.

P Change Drive Parameters.

The two columns at the right hand side of the screen indicate the parameters of the two drives that will be used as the source and destination drives in any operations selected from the menu. A cursor immediately below one of the columns points to the one that is currently selected for any manual operations. For example if the cursor points to the left hand colum, then that is the drive that will be used in any disk read write commands called using the A,S,Q,R and W options. Typing P allows you to go to the parameter set up mode where you can change the drive selection and any of the individual drive parameters.

To select a drive, use the left or right arrow to move the cursor to point at the appropriate one. To select a parameter use the up and down arrows to move the parameter select cursor. The two cursors jointly point to the parameter of one drive that can be changed. To change the parameter type a space followed by the new value. Most values can be entered as a 1 or 2 digit decimal number but the drive number should be given as a single digit (0-3). The Clear key returns control to the main menu. Note that the 5/8 inch mode is not implemented on the Model I.

Several of the parameters need some explanation.

The stepping rate number should be 00, 01, 02 or 03 for actual disk stepping rates of 3, 6, 12 and 30 milliseconds respectively. Most older drives were slow and take 30 mS. New drives can be as fast as 6 mS.

The track offset number is added to the specified track in any operation. You can use this, for example, to copy the top 40 tracks of an 80 track drive to a 40 track drive by setting the offset of the source to be 40 and the destination to be 0.

The sector skew number is used only where a group of sectors are read or written. It has 2 main uses: it speeds up reading and writing groups of sectors, it allows you to read sectors into memory in a sequence determined by the skew. For instance with 10 sectors numbered 1,2,3,4....8,9,10 a skew of 2 would read in turn sectors 1,3,5,7,9,2,4,6,8,10. This is useful if you are reading sectors from a CP/M disk where CP/M will read them in a sequence determined by a skew table such as the sequence above.

Example of use:	Note
P	1
use arrow keys	2
space key	3
03	4

Explanation:

1: calls parameter change mode.

2: select drive and parameter to change.
3: space key says we want to change selected parameter.
4: value for changed parameter.

Q dd or Q $hh Write Tracks Sectors

Q followed by a track number causes all that tracks sectors to be written to the disk. To do this the track must be already formatted and you must have some valid sectors listed in the track table for that track. Only sectors identified in the track table will be written and even then only those with good data (valid CRC). Also, a sector will not be written to disk if the Data Location pointer in the directory is set to less than 0100. (Normally when you read the address marks of a track, an initial entry is created for each sector but, as the data for that sector has not been read in yet, the Data Pointer is initialized to zero which acts as an indicator that no valid data exists.)

Example of use: Q 03
Explanation: write the sectors of track 3 to disk.

R S dd or R S $hh (or R D dd) Read Track

R S Track number or R D Track number will do a whole track read and load the resulting data into the track buffer starting at address E700 hex. After that, HYPERZAP will automatically switch to the memory inspect/edit screen showing the start of the track's data. S selects single density and D selects double density read. Track read gets all data including the intervening gaps between sectors and the sector address marks. On double density it is usually the case that sector data is not all valid because of the way the floppy disk controller works.

Example of use: RD 03
Explanation: Do a double density track read on track 3,
reading the contents into memory at E700H

S dd or S $hh Read Tracks Sectors

S followed by a track number causes the selected drive to seek that track and read data of all that tracks sectors into memory. You must have preceded this command by an operation which put some entries in the Track sector table because only the sectors identified as being present will be read. Typically you might have done an A dd in the main menu which would have read all the sector address marks on track dd and then you could do an S dd to read the data in the sectors listed. Data will be stored in the sector data storage area and the start of each sectors data will be identified in the directory. Also the address mark will be decoded and entered in the directory. In fact only when the sector data is read can all the missing bits of information be filled in. Now we can determine the length of the sector, whether it is in IBM or non-IBM format and whether the Cyclic Redundancy Check (CRC) is good.

Example of use: S 03
Explanation: read the sectors of track 3 into memory.

Note that sectors are read into memory in an order determined by the skew setting for that drive and the sequence in which they occur in the Track Sector Table. If you look at the parameter options on the main menu you will see that skew is a user changeable function. If, for example you change the skew to 3 then every 3rd sector will be read in. Suppose that the track table has sectors listed as 1,2,3,4,5,6,7,8,9,10 then they will be read in in the order 1,4,7,10,2,5,8,3,6,9.

T dd or T $hh Seek Track.

T gives you manual control over the position of the heads in your drives. Typing T 00 forces a restore to track 0 which is useful if you change the drive selection using the parameter change feature and don't know where the head of the new drive is. The drive selected is the one in the main menu screen pointed to by the cursor below.

Example of use: T 13
Explanation: move the head of the current drive to track 13.

W dd or W $hh Write (Format) a track

W ## will take the sector descriptions for that track from the directory and format the track with all sector data set to E5 hex. Any mixture and sequence of sector lengths and densities can be used provided sufficient clearance is allowed between them.

Example of use: W 03
Explanation: Format track 3 using information from the track table for track 3.

XC Whole Disk Copy

XC is the automatic copy routine which copies a whole disk or group of tracks. Follow the prompts and, for many disks, that is all that is needed. Model III users may not be able to copy all Model I disks because the Floppy Disk Controller can only read and write data marks F8 and FB whereas the Model I can read and write data marks F8, F9, FA and FB. Models III/4 will convert F9 to F8 and FA to FB. Also, Model Is can read and write single density sectors in what is called 'Non IBM format'. This allows a sector to be any length from 16 to 4096 bytes in multiples of 16 bytes. Use of this is very rare except on some early protected Model I disks. Certain disks will have to have some tracks copied manually. To do this you will have to read the address marks (A), read the sectors (S), format the track (W) and write the sectors (Q}. You can also do a track read (R) to examine the makeup of the track, possibly edit the directory (D) and construct special sectors from the information gleaned from the track read. With these tools you can backup or create most disk formats that have been produced to date - plus many that haven't. Known formats that can be copied include TRS80, IBM PC, CP/M and TI 99/4 disks. Formats you cannot copy include Apple, Commodore 64 and others which use non standard proprietary hardware and disk formats.

Example of use:
Note: your input is underlined.

```
1   XC
2   Press Enter when source disk in drive enter
    Press Enter when destination disk in drive enter
3   Pause on non standard tracks ? (Y/N): Y
4   R Regular 10 sectors single /18 double or S special ? S
5   Special; Enter # of sectors expected: 12
6   Any options Y or N ? Y
7   Force track i/d to match physical track # Y/N: Y
8   Enter starting Track No. (Decimal): 1
    Enter No. of tracks (Decimal): 3
9   Analyzing Address Marks and Reading Sector Data:
    SSSDDDSSSDDDSSS 0CH sectors, 0C00H bytes found.
    .... copying procedes.
```

Explanation:
1: Copy all or part of one disk to another.
2: Insert source and destination disks
3: Program will pause if bad track found.
4: R for standard TRS80 disks, S for others.
5: In our example we expect to find 12 sectors per track.
6: Only one option available in present release 3.0
7: If Y then destination sector i/d's are adjusted so track byte matches actual physical track no.
8: Tracks 1,2 and 3 are to be copied.
9: Copying now starts.

There are 3 options you can select if you reply Y to the 'Any options' question.

- You can choose to force the track numbers of the destination disk sector ID to match the source - useful if you are moving tracks with an offset.
- You can enable or disable the verify after write.
- You can use a different skew when writing sectors back. Normally on an XC copy the same skew (that of the source drive) is used on both read and write. However you may want to resequence the sectors. An example of this is the Autopilot program on your disk which allows you to copy the tracks from a Montezuma Micro version 1.30 DS DD disk to a version 2.2 DS DD disk. Here we use the source skew to determine how the sectors are read into memory and the destination skew to set the order they are read back.

Z Autopilot Commands

Z followed by a second letter sets up execution of one of the Autopilot commands. The Autopilot allows Hyperzap to learn a sequence of keystrokes. Once learnt, the sequence is like a program that can be rerun at any time to duplicate the original operation no matter how complicated. All the autopilot commands begin with the letter Z and are called from the main menu screen. The commands are as follows:

ZL turn Autopilot mode on.

Once on, the program records every keystroke from the keyboard no matter what screen you are in. A maximum of 1024 bytes is assigned for storage and when that is exceeded the program will automatically drop out of learn mode.

ZX exit (terminate) learn mode.

Once you exit the learn mode, no more keystrokes are recorded. If you want to see the extent of the auto program you have generated, use the H (help) key to display the storage address limits. If you want to save it to disk you must do the following: use a copy of your Hyperzap master disk and write the contents of the autopilot program zone to a free program sector on that disk. You must know what sectors are free. Suppose you want to save the autopilot program to track 16: starting from the main menu type C to clear and then type A 16 to get up entries for track 16 in the track table and D 16 to show the sector entries. There are 2; move the cursor to the one you want to use, edit the entry so that the data start is indicated as 9800 (the start of the autopilot memory region) and type Z to write the sector.

ZP run Autopilot program.

When you enter the autopilot run mode, whatever is stored in the autopilot program area will be executed, substituting stored characters for commands that would normally have come from the keyboard. The program will abort if an error is encountered. If the source and destination are specified as being the same then the program will stop any time a change between source and destination is needed. This will allow you to change disks and press the enter key to resume.

ZG dd Get a program from disk

Each program occupies one 1024 byte sector. These sectors are provided on your master Hyperzap disk starting at track 13 decimal; There are 2 sectors per track and thus 2 programs. Programs are numbered sequentially starting with 0 and 1 on track 13. Thus ZG 04 will load the first sector from track 15. Appendix 3 gives a list of Autopilots on your disk.

Example of use: First, insert Hyperzap disk in drive 0
and blank disk in drive 1. Then type:
ZG 01 to load Hyperzap self copy
ZP to start program.

If you have only 1 drive, first set the Destination drive to 0. Then proceed as before. Wait for prompts to change disks.

4.0 Using The Directory

When HYPERZAP reads a disk it builds entries in a "Track Directory" which describes exactly how each track is formatted, how many sectors of what density and type and where they are positioned on the track. In the disk copy mode this directory information is used to rebuild an image of the track in the track buffer and subsequent formatting of the destination disk. Note that this directory is NOT the same as the kind of directory you have on your regular DOS disks. Once the destination disk is formatted the sector information can be written.

In the manual mode the directory can be generated by reading address marks from an existing disk or by building entries using the directory edit features. In addition, a sector shown in the directory can be read into memory, editted and re-written to disk either in the same or an alternate position.

4.1 The Directory Display.

The display consists of three parts, the main part being the directory itself. To the left of this is some general information and an indicator giving the current track you are looking at. You can page forward or back to look at other tracks using Shift/down-arrow and Shift/up-arrow respectively.

The directory display consists of a number of columns as follows:

#....Sequential sector number (in decimal) for information only, no relation to the actual sector numbering.

Tk...The logical track number as given in the address mark.

Sp...The spare byte given in the address mark. Sometimes used to indicate the side on two sided disks.

Sc...The logical sector number given in the address mark

Ln...The length code. In IBM 00,01,02 or 03 standing for 128, 256, 512 and 1024 bytes respectively. In N(on) IBM 01,02,03....FF,00 for 16,32,48,...4080,4096 respectively

CRC..Y for yes if the address mark is good as it will be if the directory was read from a disk. May be N if we want to create a special address mark without a valid CRC.

DAM..No swearing now!. Sometimes you might want to with a problem disk in the middle of the night. DAM stands for Data Address Mark and is a byte recorded immediately before the sector data on the disk. Can be F8,F9,FA or FB in single density Model I and F8 or FB in double density Model I/III/4.

Data.This is a 4 digit hexadecimal number specifying where in memory the start of sector data can be found.

Angle...This is a decimal number from 0 to 6250 giving the angular position round the disk of the start of the sectors address mark. Each unit represents one byte (equivalent to 32 microseconds in time) round the disk in double density. (Half a byte in single.) Displacement is measured from the index hole, 0 being the start and 6250 one full revolution.

Type.This is a code specifying the type of sector. The following codes are allowed:

IBM a standard type sector.
NIBM a standard non IBM sector. The length is 16 times the sector length code. Available on Model I single density only.
W special block of data to be included on formatting.
X the sector data does not have a valid CRC.
X the sector address mark does not have a valid CRC.
Z neither data nor address mark has a valid CRC.

CRC..If Y for yes then data read from an existing disk had a valid cyclic redundancy check. If N for no then when a destination disk is next formatted no CRC will be put there.

Dens.Density of sector - S Single, D Double density.

4.2 The Directory Menu

A cursor points to one of the sectors and this is the sector that will be operated on by one of the commands in the menu below the main screen. You can move this cursor by means of the up and down arrows. The options in the menu are as follows:

A - Append a new sector entry after the cursor. The sector parameters must be entered as in the example below.

C - Copy the current track entries up to the next track. Ihe track number will be updated to match the actual track number. This is useful for rapid generation of your own formats. No duplication will take place if entries already exist in the track above.

D - Delete the entry for the sector pointed to by the cursor.

E - Edit the current sector entry. Follow the prompts given. You can skip a reply if you don't want to change the current value, by using the enter key. See example below.

G - Generate a standard track. This allows you to build an entry of an arbitary number of equally spaced sectors. The new sectors will be appended to any already existing in the table for that track so you can build a entry of say 5 sectors of one density followed by 5 of another. Sector spacing is calculated from requirements of sector size and density. The sectors will be numbered according to one of 3 options: 1 count down, 2 count up, 3 skewed (interleaved) count. See example below.

H - Helpful information about memory occupancy.

I - Insert a new sector entry before one currently pointed to by the cursor. See example below.

M - Go to edit memory (Main menu 'I') mode to edit sector data. Pressing M will take you to the memory modify mode starting at the beginning of the data of the sector the cursor was pointing at. The Hi and Lo Limit Indicators will show the bounds of the sector data. You can edit the data and use the screen 3 R(eturn) command to return you to the same place in the directory.

R - Read the current sector to memory. You will be asked for a destination memory address. Use H before you do this if you want to see what space is free. This is a useful command if you want to extract the boot sector from a disk. When you have loaded it you can reboot your DOS, dump it to disk and run it through a disassembler. You can also disassemble it in memory where it is. Generally, anything in high memory will be untouched by booting your DOS or Hyperzap.

T - force the track byte to the same value. Example: T 09 makes all the sectors have 09 as their track byte.

Z - Zap the sector - i.e. write it back to disk at your own risk! The sector pointed to by the cursor will only be written to if the data location is non zero and the sector is an IBM or NIBM type. Other types are only used in formatting.

The W sector definition is not really a sector at all. It gives you a way of defining all or part of a track in any arbitrary way you like. You can build a block of memory using the memory inspect/modify mode and then define it to be a W sector. This will be copied exactly to the track when you do a track format. (W, main menu). Remember though, that the floppy disk controller will replace some bytes. F7 for instance gets replaced by the 2 byte contents of the CRC register.

While in the Directory mode you can page forward and backward through the track entries by using Shift-Up-Arrow and Shift-Down-Arrow.

Edit eample (A, E or I commands)

Here we are going to create a double density regular sector
of 256 bytes numbered 06 on side 0.
Note your entries are underlined.

I
Density - (S)ingle or (D)ouble D
Enter code for sector type: Space = standard IBM, N=Non IBM
W = Special format block, X = No data CRC, Y = No ID CRC,
Z = No data or ID CRC : _ (space entered)
Spare byte: 00
Logical Sector #: 06

Length Code : 01
Start of Sector Data (4 digits Hex): 9000
Sector Angular Position 0-6250 Decimal: 0196

Generate Example

Here we are going to generate a track with 10 sectors using an interleave skew factor of 3. Each Sector will have 256 bytes in double density.

G
No of sectors 10
Sector count 1 down, 2 up, 3 skewed: 3
Initial sector # Hex: 01
Highest sector # Hex: 0A
Sector skew Hex: 03
Density - (S)ingle or (D)ouble D
Enter code for sector type: Space = standard IBM, N=Non IBM
W = Special format block, X = No data CRC, Y = No ID CRC,
Z = No data or ID CRC : _ (space entered)
Bytes / Sector: 256 (note decimal)

Result: sector sequence 01,04,07,0A,02,05,08,03,06,09.

5.0 Memory Inspect/Modify Mode

This is the third screen. It gives you a scrolling window on memory, all the arrows and shift-arrows do something - try them. The flashing cursor points to a memory location which will be operated on if you try any option key. An indicator top left gives the actual memory address. Other indicators are Marker 1 and Marker 2, L low limit, U upper limit and a continuous length/CRC indicator. This screen allows you to alter any part of memory and to examine previously read track and sector data. Use the H key to see where HYPERZAP is and keep away from this region as modifying it may be hazardous to the health of your disks!. Options available in the present version are is follows:

S Search

Search for a hexadecimal string. Enter the string in hexadecimal character pairs for each byte. Search will be from the current cursor position plus one to Upper Limit. The Upper limit is automatically set by sector and track read commands but you can change it manually (U command). The search string can be defined in ASCII or hex by preceding it with either an apostrophe (') or a $ sign.

Example:	S	(search)
	'text	(find the string: text)
	$0102	(find the 2 bytes 0102)

C Calculate CRCs

Calculate the CRC from Marker 1 to the current cursor position. This is a toggle switch which switches the display back and forth between showing the CRC and showing the distance from marker 1 to the cursor. The latter will always be set if the cursor is at a lower address in memory than Marker 1. The CRC is calculated differently depending on whether the last disk read was in single or double density. To use it correctly, position the cursor over an address mark (F8,F9,FA,FB or FE) and press '1' to set Marker 1. Then as you move the cursor forward in memory the display will show either the CRC or the distance from the address mark. When you reach the last byte of a sector or sector ID the next two bytes should be the CRC which should match the number displayed if you are in CRC display mode. As you move the cursor over the CRC the value calculated will become 0000.

M Modify Memory.

Once in this mode any hex key 0-9,A-F will overwrite the current cursor position. Two cursors show your current position, one in hex and one in the ASCII sections of the display. You can switch between editing in hex and editing in ASCII by typing the ' symbol or back to hex by typing the $ symbol. To enter the $ or ' symbols into memory you must go to hex mode and enter 24 for $ or 27 for '. There are several sub-commands in the memory modify mode.

In either mode there is an additional memory insert/delete feature. This allows you to add or delete bytes to memory, shuffling everything above the cursor up one or down one as you do this. The Lower and Upper limit markers set bounds on the region that is shuffled. If you try and do an insert or delete outside the zone set by L and U then nothing will happen. Insert generates an 00 byte at the current cursor location and moves all memory above up to the Hi limit up by one byte. Delete removes one byte at the current cursor location and moves memory above down by one byte, filling in at the top with 00 bytes. In the hex mode use X to delete and I to insert. In the ASCII mode use Control-D and Control-F. (only available on Models with Control keys such as the LNW and the TRS-80 model 4.)

You can also increment, decrement and complement memory in the hex modify mode using the keys + - and * respectively. The clear key exits the modify mode. To insert the equivalent of the clear key into memory type hex 1F.

B Block Move.

Using this a block move of data can be accomplished. There are two modes, one for a straight move and the other which removes CRCs in the process. As the move proceeds the program checks for the characters that would cause a floppy disk controller to zero its CRC generator and the corresponding program CRC is reset. As the move continues, each successive byte updates the CRC count. If a matching CRC is found in the data then it is replaced by an F7 code. The purpose of all this is to allow you to read a track and move all or part of it to the sector data area, replacing the CRCs with F7s. This can then be defined as a special W type sector and when the track is reformatted using this as direct track data then the Floppy Disk Controller will replace all the F7 bytes with Cyclic redundancy checks. You may think this is a roundabout way of doing things but sometimes it is the only way.

The block move in the memory modify mode has now been improved for operations involving moving sections of data obtained using the track read command. If you select Y for the substitution of F7 bytes whenever the CRC count matches the next two bytes to be moved then an additional substitution takes place. If an F8, F9, FA, FB or FE address or data mark is encountered during the block move then the preceding bytes are filled in automatically as they should be for formatting a track. With single density this means the previous 4 bytes are forced to 00. With double density you get 8 00 bytes and 3 F5 bytes corresponding to the synchronization data essential for a reliable track read.

1,2 Set Markers.

Keying the digits "1" or "2" will set Marker 1 or Marker 2 to the current cursor address. Useful in CRC, Search, Fill and Block Move operations.

L,U Set Limits.

Keying "L" or "U" will allow you to enter a 4 digit hex number for the lower and upper limit markers. These markers do not stop you scrolling or modifying memory. They are simply reminders. The upper limit also acts as a stop when searching memory (S). These two markers are set automatically when you do a track read or a sector data edit (see R below and the M command in section 4.)

The L and U markers also act as bounds for inserting and deleting memory in Modify mode.

R Return to Directory.

Use this if you have previously read a sector in Directory (screen 2) mode and have come to this screen to view or edit the data. R will return you to the same point you left with the cursor still pointing to the sector in question so that you can simply rewrite it to disk by hitting Z. See the M command in section 4.

A New Address.

"A" will allow you to specify an address of a new part of memory that you want to inspect and modify. Enter the address as a 4 digit Hex number when you see the > prompt. The screen will be repainted with data starting in the top left hand corner of the screen at the address you just entered.

F Fill Memory.

"F" allows you to fill memory from Marker 1 to Marker 2 with a specified string of data bytes. Memory is filled from the Marker 1 to Marker 2 with the string from Marker 1 to the current cursor position. The best way to use this is to move the cursor to the end of the space you require filling, press 2 to set Marker 2 and then go to the start of the space to be filled. Press 1 and then M to go into edit mode. Enter the bytes, editing as neccessary and then get out of edit with the Clear key. The cursor should point at the last character of the string you have just entered. Press F and the job is done. Be careful the markers are set correctly. M2 should be higher than M1 or all of memory will be wiped out including Hyperzap!.

Example: suppose you want to fill a region of memory with many copies of the word Hyperzap. Go to the end of the region and press 2 to mark the end. Go to the beginning and press 1 then M to edit. Type 'Hyperzap to insert the word in ASCII mode at the beginning, use Clear to get out of Modify and type F to fill.

V Verify (Compare) Memory.

This command allows you to verify or compare two different regions of memory. You use it by setting marker M1 to point at the 1st region and M2 at

the start of the second region. When you press Z memory at M1 will be compared byte by byte with memory at M2 until a mismatch is found when M2 will be set at one less. The size of the matching region is given by the region M1 to M2 inclusive. If no match is found then M2 = M1-1.

6.0 A Guided Tour of HyperZap

One of the best ways of learning to use the program is by example. The instructions that follow are intended to take you, step by step, through the most important features of Hyperzap. Please refer to the figures at the end of this appendix where neccessary. In general you should see the same on your screen except in the bottom 3 lines which will show the command options on screens 2 and 3 and will be blank on the main menu. For the figures here we had to go to the Help mode (by typing H) to get a screen print.

First we will exercise some of the main menu functions. Figure 1 shows the main menu screen. Most commands are obtained by touching a single key. Those that require a track number or address immediately respond with a space and wait for you to enter the value (decimal for track numbers, hex for addresses). Now, into drive 0 insert your Hyperzap disk or a backup of it (more about that later). Make sure you have a write protect tab on your disk to prevent accidents until you know what you are doing. DO NOT press keys at random. If you have a drive with faulty write protect switches or logic DO NOT use a test disk you value or you may regret it. We are going to read the address marks and data of the sectors on track 0. There are only two, the boot sectors for model I and III.

Type: A 00 (no space between the A and the 00)

There will be a short delay as the disk spins for 8 revolutions while Hyperzap is reading address mark data and averaging the results. Now if you type D 00 you will see the sectors for track 00 listed. However, at this point the sector data has not been read in and the data mark is not neccessarily valid. Return to the main menu by hitting the Clear key and enter:

S 00

The disk will spin again while the sectors are read in the order they appeared in the track sector table. You will see a series of letter S and Ds appear on the screen, one for each sector read, S for single and D for double density. In this case you should just see the letters SD appear as there is 1 single and 1 double density sector. Now the sector data is in memory. Type D 00 again and you should see a screen as in Figure 2. Near the start of the track is a single density sector and its data is in memory starting at address 9C00. About half way round the disk is a double density sector and its data is at 9D00. The angular position in degrees round the disk of the address mark can be got by taking the Ang number given and multiplying by 360/6250.

The cursor in Figure 2 points to the first sector you can see the data that was read into memory. Typing M will take you to Figure 3. Here you are in a memory examine/change mode looking at the data of the 1st sector. To edit the sector data you would have to type M again to go to the modify mode. For now assume that you have edited the data and return to the previous screen by typing R.

Now you are back. If you had actually modified the data you could write it back to disk by typing Z for Zap. An alternative way to get data on individual sectors is, from this screen (2) to type R (for Read). This will allow you to read the data into any arbitary point in memory. You will be asked for the address. You could then edit it and rewrite it as before.

Now return to the main menu (hit the Clear key). We will do a track read in single density.

Type: R S 00

The disk will spin and track 00 will be read in single density into memory starting at address E700. When the index hole comes round again reading will stop and the program will jump to the I mode (screen 3) showing something like Figure 4. You can now see all the features recorded on the track. If there was no single density data on the track you will just see garbage. In line E770 of figure 4 you can see an FE byte. This is the address mark of the sector ID. Following this is

00 00 00 01 F1D3

which says that following this is a sector of track 00, side 00, sector number 00, length code 01 and the CRC is F1D3. To check the CRC move the cursor till it is over the FE byte and touch the 1 key to set marker 1. Now as you advance the cursor the Length/CRC indicator lower left will show the distance or CRC from marker 1. Touch the C key to toggle between these two modes. As the cursor passes over the second byte of the CRC the indicator should go to 0000 if all is correct and you are in the CRC display state.

You can similarly read the track in double density by, from the main menu typing

R D 00

Try it and you should get something like Figure 5. Since the first part of the track is in single density you will have to advance in memory past the single density region (which will show as garbage) till you come to around F400 in memory in this case.

You can do the same with a double density track read but double density track reads are much more unreliable. Data bytes are often lost and replaced with 00 by the floppy disk controller. You will therefore be unlikely to see an exact replica of the sector data unless it was formatted with some simple constant code and had not yet been written to on a sector by sector basis.

If you want to look at another track, try track 1 which has 12 sectors. 6 in double and 6 in single. Figure 6 shows what you should expect to see. One sector is off the screen and you would have to move the cursor down to see it.

Making a Backup of your Disk.

Although there is an Autopilot prooram on your disk for doing this automatically it is instructive to do it manually. To make a backup of your disk, use the XC whole disk copy feature on the main menu. Insert your original disk in the source drive and, if you have more than one drive, put a blank disk in the destination drive and set the parameter table to indicate which drive is the source and which the destination. To do this type P and set the parameter cursor to the top, if not already there. Select source or destination drive as appropriate using the left/right arrow keys to move the drive cursor. Now change the drive number by typing a space followed by a single digit 0-3. Exit by hitting the clear key.

Now type XC to begin the copy. Copy 27 tracks from 0 to 26. There is no harm in specifying more tracks than there actually are on the original. All that will happen is that blank tracks will be stepped over when the duplicate is made. Track 0 has 2 sectors, tracks 1 through 13 have 12, while the rest have 2 so you may want to copy track 0, 1-12 and 13-26 separately so that you can take advantage of the Sector count check.

Supposing you just want to copy tracks 1 through 12, answer Y to the first and S to the second question for special track check. Then reply 12 to the sector count/track question. Answer 1 to the start and 12 to the Number of Tracks question. If all is well the copy will proceed and you will see the sectors being read in. If you have a 1 drive system you will have to swap disks when asked. If an error occurs you have the opportunity of retrying, continuing or quitting. You should not get this unless you have bad disk or drive problems.

The full sequence for backing up your HyperZap disk (assuming 2 drives) and full checking for errors is as follows (spaces for clarity):

XC enter	(copy a disk)
enter	(source in drive)
enter	(destination in drive)
Y S 2 enter	(pause on error, special, 2 per track)
N 0 enter	(no options, start at 0)
1 enter	(copy one track)
XC enter	(same thing)
enter	
enter	
Y S 12 enter	(but with 12 sectors per track)
N 1 enter	(starting at track 1)
12 enter	(and copying 12 sectors)
XC enter	(finally the Autopilots)
enter	
enter	
Y S 2 enter	(these are 2 sectors/track)
N 13 enter	(starting on track 13)
13 enter	(for 13 tracks)

```
Hyperzap V3.2c Feb-03-85  Copyright 1985  M.J.Gingell  Hypersoft
-------------------------------------------------------------
Screen 1:   *** Command Options *** .     Parameter   Srce Destn
A Read addr mrks   XC Disk Copy     .    Drive Number : 00 : 01
C clr Track Table  B AutoBoot Disk  .    No of tracks : 40 : 40
D display          Z Autopilot      .    Steps/Track  : 01 : 01
I inspect memory   P Change Params  .    Head at track: 00 : 00
J ## jump to #3    + Step & repeat  .    Side         : 00 : 00
S Read sectors     T # seek track   .    Size 5/8 inch: 05 : 05
o Write sectors    E comb. A,S,D    .    Stepping rate: 03 : 03
R S/D read track   H(elpful) facts  .    Track offset : 00 : 00
W Format Track     YX Quit & reboot .    Sector skew  : 02 : 02
-------------------------------------------------------^------
Hyperzap uses  4300 - 8161      Track/sector table   9000 - 90A3
Sector data    9C00 - 9C00      Track buffer         E700 - FFFF
Autopilot      9800 - 9800      P Screen Print         Clear -->
```

Figure 1. The Main Menu Screen

```
Now, type A 00 to read the address marks on track 0   )  Alternatively
          S 00 to read the sector data into memory    )  these 3 commands
          D 00 to show the screen below.              )  can be replaced
                                                         by E 00
```

```
Screen 2:       #   Tk Sp Sc Ln CRC DM Data Ang. TYP CRC Den
 Physical   >   01  00 00 00 01 Y   FB 9C00 0371  IBM Y  S
 track 00       02  00 00 01 01 Y   FB 9D00 3646  IBM Y  D
  Sector
  Table

 Total 02
 sectors

 Drive 00
 05 inch.


Hyperzap uses  4300 - 8161      Track/sector table   9000 - 90A3
Sector data    9C00 - 9C00      Track buffer         E700 - FFFF
Autopilot      9800 - 9800      P Screen Print         Clear -->
```

Figure 2. Sector Table for Track 0

Now type M to go to display of sector 0 data (Fig 3)

```
Screen 3:      0 . . . 4 . . . 8 . . . C . . .    Ascii equiv.
Crsr:9C00 9C00 1806005500550B06F331FF4121EC37ED ..V.V....1.A..7.
          9C10 5B0242CD2242CD69422A0242E9CB4620 [.B."B.iB*.B..F
Limits:   9C20 FCC9CD9B423ED077060010FE7710FECD ....B>.w....w...
  Lo:9C00 9C30 1D423653060A10FECD1D42CD9B427ECB .B6S......B..B..
  Hi:9CFF 9C40 67C821003C11013C01FF033620EDB021 g...<..<...6 ...
          9C50 5E4211103D7EB728FE1223131BF74469 ^B..=..(..#...Di
Markers:  9C60 736B206661756C7400CD9B423A064247 sk fault...B:.BG
  M1:9C00 9C70 3A07424FC5ED53D742060AC5CDA142C1 :.BO..S.B.....B.
  M2:FFFF 9C80 2809ED5BD74210F3C34242C10D20E505 (..[.B...BB.....
Length/   9C90 C8CD9B42C5CD2F42C11BD53E0132E137 ...B../B...>.2.7
CRC :0001 9CA0 C936D0060A10FE060A36D010FECD9B42 .6.......6.....B

Hyperzap uses  4300 - 8161      Track/sector table   9000 - 90B9
Sector data    9C00 - 9E00      Track buffer         E700 - FFFF
Autopilot      9800 - 9800      P Screen Print         Clear -->
```

Figure 3. Track 0, Sector 0 Data

To read track 0 into memory, single density type: R S 00

```
Screen 3:      0 . . . 4 . . . 8 . . . C . . .    Ascii equiv.
Crsr:E7A0 E7A0 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF ................
          E7B0 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF ................
Limits:   E7C0 FFFFB0000000FE00000001F1D3FFFFFF ................
  Lo:E700 E7D0 FFFFFFFFFFFFFFFF01000000000000FB ................
  Hi:F33B E7E0 1806005500550B06F331FF4121EC37ED ..V.V....1.A..7.
          E7F0 5B0242CD2242CD69422A0242E9CB4620 [.B."B.iB*.B..F
Markers:  E800 FCC9CD9B423ED077060010FE7710FECD ....B>.w....w...
  M1:FC00 E810 1D423653060A10FECD1D42CD9B427ECB .B6S......B..B..
  M2:FFFF E820 67C821003C11013C01FF033620EDB021 g...<..<...6 ...
Length/   E830 5E4211103D7EB728FE1223131BF74469 ^B..=..(..#...Di
CRC :EBA1 E840 736B206661756C7400CD9B423A064247 sk fault...B:.BG

Hyperzap uses  4300 - 8161      Track/sector table   9000 - 90B9
Sector data    9C00 - F339      Track buffer         E700 - FFFF
Autopilot      9800 - 9800      P Screen Print         Clear -->
```

Fig 4. Track read in single density

To read track 0 into memory, double density type: R D 00

```
Screen 3:      0 . . . 4 . . . 8 . . . C . . .     Ascii equiv.
Crsr:F546 F530 4E4E4E4E4E4E4E4E00000000000000A1 NNNNNNNN........
          F540 A1A1FE00000101FA0C4E4E4E4E4E4E4E .........NNNNNNN
Limits:   F550 4E4E4E4E4E4E4E4E4E4E4E4E4E4E4E00 NNNNNNNNNNNNNNN.
  Lo:E700 F560 00000000000000000000000000A1A1FB ................
  Hi:FF61 F570 1806005600560B06F331FF42ED5B0243 ..V.V....1.B.[.C
          F580 CD2143CD6D432A0443E9DBF0CB4720FA ..C.mC*.C....G .
Markers:  F590 C9CD9F433ED0D3F0060010FED3F010FE ...C>...........
  M1:F542 F5A0 CD1A433E53D3F0060A10FECD1A43CD9F ..C>S........C..
  M2:FFFF F5B0 43DBF0CB67CB21003C11013C01FF0336 C...g...<..<...6
Length/   F5C0 20EDB021624311103D7EB728F3122313  ...bC..=..(..#.
CRC :FA0C F5D0 18F74469736B206661756C7400CD9F43 ..Disk fault...C

Hyperzap uses  4300 - 8161      Track/sector table   9000 - 90B9
Sector data    9C00 - FF62      Track buffer         E700 - FFFF
Autopilot      9800 - 9800      P Screen Print         Clear -->
```

Fig 5. Track read in double density

```
Screen 2:       #   Tk Sp Sc Ln CRC DM Data Ang. TYP CRC Den
 Physical   >   01  01 00 06 01 Y   FB 9C00 0201  IBM Y  D
 track 01       02  01 00 05 01 Y   FB A200 0514  IBM Y  D
  Sector        03  01 00 04 01 Y   FB 9D00 0850  IBM Y  D
  Table         04  01 00 03 01 Y   FB A300 1175  IBM Y  D
                05  01 00 02 01 Y   FB 9E00 1513  IBM Y  D
 Total 12       06  01 00 01 01 Y   FB A400 1831  IBM Y  D
 sectors        07  01 00 06 01 Y   FB 9F00 2393  IBM Y  S
                08  01 00 05 01 Y   FB A500 2987  IBM Y  S
 Drive 00       09  01 00 04 01 Y   FB A000 3580  IBM Y  S
 05 inch.       10  01 00 03 01 Y   FB A600 4175  IBM Y  S
                11  01 00 02 01 Y   FB A100 4772  IBM Y  S

Hyperzap uses  4300 - 8161      Track/sector table   9000 - 913D
Sector data    9C00 - A800      Track buffer         E700 - FFFF
Autopilot      9800 - 9800      P Screen Print         Clear -->
```

Fig 6. Hyperzap track 1 sectors

Appendix 1: Special Disk Backup

Many disks can be copied without any manual intervention. However some are made with specific errors recorded on them so that regular copying procedures will not be reproduced exactly the same. Typical of these are ones where a CRC error is deliberately introduced in the sector data at the format stage. As an example let us consider disks which use the loader by Paul Brandon. Examples of this come from Med Systems and Melbourne House.

You can identify this type of disk because it comes as a dual booting Model I/III disk with one of the tracks, typically number 3 having only 6 sectors the 6th sector giving an error when you try and copy it. Apart from track 3 and track zero all tracks have 10 sectors of single density. Track zero has at least one single and one double density sector.

To make a backup of your disk do a regular disk copy of the entire disk. with the exception of track 3. Some disks such as Asylum only have 12 tracks while others such as Penetrator have as many as 32. Now we have to go back and copy track 3. To do this we must create a specially formatted track which will have space for the first 5 sectors which are normal plus a specially formatted section corresponding to the apparent 6th sector. Then, when the track is formatted we can write in the data for the 5 good sectors.

First, on the original we will read the address marks and sector data into memory. Type:

```
A 03     (program reads address marks)
D 03     (display sector info.)
```

There are 6 sectors apparently but the 6th is there to fool you and will give an error message when you try to read it. Note the angular position of sector 6 and then delete the entry and return to the main menu. Now read in the sector data for sectors 1 to 5 using S 03. Sector 6 is a false sector and we have to replace it with a special block of data to be used to create the same effect at format time. From the main menu type:

```
R S 03   (to read track 03 in single density)
```

The start of the track will be displayed at E700. Now search for the start of sector 6. Type:

```
S 03014E01  (i.e look for the same information
             you saw for sector 6 on Figure 1)
```

You should now see something like Figure 2 (back up about 2 lines using the up-arrow key). The important information extends over about 650 bytes from here. We are going to copy this down in memory to a point where the old sector data was indicated in Fig 1. First add a few preceding lines of FFs to the display. This will help act as a buffer between sector 5 and the new block. Also check your adress mark is preceded by 12 zeros and an FE, sometimes these get corrupted by a track read. If you dont see these use the modify mode to set them up. The loader is very critical about having sufficient zeros present and, if you have problems in can be advantageous to increase the number of zeros before the FE address mark and the FB data mark. Set marker 1 at the beginning of the FFs. Next find the end of the

important data - advance until the length/CRC indicator shows about 0300 and set marker 2. Now type B to do a block move. In this case Fig 1 showed the data for sector 6 starting at 9200 so move the sector data to 9200. Answer Y to the question - do you want CRCs replaced by F7s. If you look at memory at 9200 you should see something like Figure 3.

Now, go to the sector table and edit the entry for sector 6 so it looks like Figure 4. In other words sector 6 is now defined as a type W sector which will only be used at format time. The format subroutine will take the information on the 6 sectors and build an image in memory of 5 blank sectorS plus a zone with a copy of the special block of data for sector 6. Then this will be written to disk and the controller will replace the F7 bytes with CRCs. Once this is done the 5 good sectors can be written onto the track in their respective places using the Q command.

Return to the main menu (using the Clear key) and type:

```
W 03        (to format the track)
Q 03        (to write the first 5 sectors)
```

We are finished. Put a write protect tab on and try booting it.

```
Screen 2:       #   Tk Sp Sc Ln CRC DM Data Ang. TYP CRC Den
 Physical       01  03 00 1A 01 Y   F8 8D00 0067  IBM Y  S
 track 03       02  03 00 19 01 Y   F8 8E00 0665  IBM Y  S
  Sector        03  03 00 22 01 Y   F8 8F00 1265  IBM Y  S
  Table         04  03 00 36 01 Y   F8 9000 1864  IBM Y  S
                05  03 00 29 01 Y   F8 9100 2465  IBM Y  S
            >   06  03 00 4E 01 Y   FB 9200 3065  X   N  S
```

Figure 1. Track 3 sectors information as originally read in.

```
Screen 3:      0 . . . 4 . . . 8 . . . C . . .     Ascii equiv.
Crsr:ECED ECED FFFFFFFFFFFF00000000000000FE03 ................
          ECFD 014E0173FCFFFFFFFFFFFFFFFFFF0000 .N.s............
Limits:   ED0D 000000000000FBE1E12A0544E5C3004C .........*.D...L
  Lo:E700 ED1D 830EE5E5E5E5E5E5E5E5E5E5E5E5E5E5 ................
  Hi:F11F ED2D E5E5E5E5E5E5E5E5E5E5E5E5E5E5E5E5 ................
          ED3D E5E5E5E5E5E5E5E5E5E5E5E5E5E5E5E5 ................
Markers:  ED4D E5E5E5E5E5E5E5E5E5E5E5E5E5E5E5E5 ................
  M1:ECED ED5D E5E5E5E5E5E5E5E5E5E5E5E5E5E5E5E5 ................
  M2:EF6D ED6D E5E5E5E5E5E5E5E5E5E5E5E5E5E5E5E5 ................
Length/   ED7D E5E5E5E5E5E5E5E5E5E5E5E5E5E5E5E5 ................
CRC :0001 ED8D E5E5E5E5E5E5E5E5E5E5E5E5E5E5E5E5 ................
```

Figure 2. Start of Track 3, Sector 6 data. Obtained by doing a single density track read. Note CRC at ED1D (underlined). There are more between here and the end of the significant data. The length of the blocks of zeros has been extended to ensure sync.

```
Screen 3:      0 . . . 4 . . . 8 . . . C . . .     Ascii equiv.
Crsr:9200 9200 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF ................
          9210 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF ................
Limits:   9220 FFFFFFFFFFFF00000000000000000FE03 ................
  Lo:E700 9230 014E01F7FFFFFFFFFFFFFFFFFF000000 .N..............
  Hi:F11F 9240 0000000000FBE1E12A0544E5C3004CF7 ........*.D...L.
          9250 E5E5E5E5E5E5E5E5E5E5E5E5E5E5E5E5 ................
Markers:  9260 E5E5E5E5E5E5E5E5E5E5E5E5E5E5E5E5 ................
  M1:ECED 9270 E5E5E5E5E5E5E5E5E5E5E5E5E5E5E5E5 ................
  M2:EF6D 9280 E5E5E5E5E5E5E5E5E5E5E5E5E5E5E5E5 ................
```

Figure 3: Start of sector 6 data after block move. Note how the CRCs in Fig 2 have been replaced by F7s (underlined).

```
Screen 2:       #   Tk Sp Sc Ln CRC DM Data Ang. TYP CRC Den
 Physical       01  03 00 1A 01 Y   F8 8D00 0067  IBM Y  S
 track 03       02  03 00 19 01 Y   F8 8E00 0665  IBM Y  S
  Sector        03  03 00 22 01 Y   F8 8F00 1265  IBM Y  S
  Table         04  03 00 36 01 Y   F8 9000 1864  IBM Y  S
                05  03 00 29 01 Y   F8 9100 2465  IBM Y  S
              >  06  Block length: 0298 9200 3100  W      S
```

Figure 4: Track 3 sector table after editting sector 6.

Hyperzap's Boot Sectors

Track Zero of your disk is formatted to contain only two sectors. Sector zero is in single density and is for use on model I computers. Sector 1 is in double density and is for Model III/IV machines.

The pages that follow contain the complete source listing for these sectors. Please feel free to use them either as is or modified for your own purposes. You can extract the object code from your disk by the following method:

- Do a read of address marks of track zero by typing A 00
- Do a read of sector data by typing S 00
- Go to the track display page with D 00

You now have a display showing two sectors and their statistics. You can inspect and modify the data by typing M.

The rest of the tracks on your disk have 12 sectors, 6 in double and 6 in single density. When you boot up the appropriate sector on track zero is loaded and then executed. On the Model I it loads and starts running at 4200H and on the Model III/IV at 4300h. The head is stepped to track 1 and loading of the program begins. Now the loader loads the 6 single or double sectors in (in reverse sequence i.e. 6,5,4,3,2,1) and steps to the next track continuing until all tracks are loaded. The 3rd through 7th byte of each sector form a user alterable table specifying the number of tracks, sectors per track, load and start addresses. You can modify these to suit your own programs as neccessary.

A disk set up with this format boots very fast. This is because out of each disk revolution, which takes 200 mS, only part is used to read data. Sufficient time remairs to step to the next track before data on that track comes round again. The present disk has 12 tracks including track 0 and so takes 12 times 200 mS or 2.4 seconds to boot.


```
                00010 ; Copyright (c) 1983 Hypersoft, M.J.Gingell
                00020 ;
                00030 ;Model I Bootstrap loader, assumes 6 sectors of
                00040 ;256 bytes, single density on each successive track
                00050 ;Sector sequence 6,5,4,3,2,1
                00060 ;Version 1.2 May-15-83
4200            00070         ORG     4200H
                00080 ;
                00090 ;FDC control registers
37E1            00100 DRIVE   EQU     37E1H ;Drive 0 motor trigger
37EC            00110 COMAND  EQU     37ECH ;Command register
37EC            00120 STATUS  EQU     37ECH ;Status register
37ED            00130 TRACK   EQU     37EDH ;Track number register
37EE            00140 SECTOR  EQU     37EEH ;Sector number register
37EF            00150 DATA    EQU     37EFH ;Data I/O register
                00160 ;
4200  1806       00170 BOOT    JR      BOOT1
4202  0055       00180 LOADAD  DW      5500H       ;Address to start loading
4204  0055       00190 STRTAD  DW      5500H       ;Address to start execution
4206  0C         00200 NTRAKS  DB      12          ;Number of tracks
4207  06         00210 NSECTS  DB      6           ;Number of sectors/track
4208  F3         00220 BOOT1   DI                      ;Disable interrupts
4209  31FF41     00230         LD      SP,BOOT-1       ;Set stack pointer
420C  21EC37     00240         LD      HL,COMAND       ;HL = Command/Status
420F  ED5B0242   00250         LD      DE,(LOADAD)     ;DE is storage pointer
4213  CD2242     00260         CALL    TRACK1      ;Start drive, seek track 1
4216  CD6942     00270         CALL    LOADER          ;Load program
4219  2A0442     00280         LD      HL,(STRTAD)
421C  E9         00290         JP      (HL)            ;Run it
                00300 ;
421D  CB46       00310 NOTBUS  BIT     0,(HL)          ;Wait till not busy
421F  20FC       00320         JR      NZ,NOTBUS
4221  C9         00330         RET
                00340 ;
4222  CD9B42     00350 TRACK1  CALL    RETRIG          ;Select drive 0
4225  3ED0       00360         LD      A,0D0H          ;Force FDC Interrupt
4227  77         00370         LD      (HL),A
4228  0600       00380         LD      B,0             ;Delay while command
422A  10FE       00390 DELAY1  DJNZ    DELAY1          ;takes effect
422C  77         00400         LD      (HL),A          ;Repeat
422D  10FE       00410 DELAY2  DJNZ    DELAY2
422F  CD1D42     00420 STEPIN  CALL    NOTBUS          ;Wait till not busy then
4232  3653       00430         LD      (HL),53H        ;Step out to next track
4234  060A       00440         LD      B,10            ;Delay
4236  10FE       00450 WREADY  DJNZ    WREADY
4238  CD1D42     00460 WRDY0   CALL    NOTBUS          ;Wait till not busy
423B  CD9B42     00470         CALL    RETRIG          ;Retrigger drive
423E  7E         00480         LD      A,(HL)          ;Get status
423F  CB67       00490         BIT     4,A             ;Seek error ?
4241  C8         00500         RET     Z               ;No OK Quit
                00510 ;Clear screen and print error message
4242  21003C     00520 ERROR   LD      HL,3C00H        ;Clear screen
4245  11013C     00530         LD      DE,3C01H
4248  01FF03     00540         LD      BC,3FFH
```

```
424B  3620      00550          LD      (HL),20H
424D  EDB0      00560          LDIR
424F  215E42    00570          LD      HL,DSKERR       ;FAULT message
4252  11103D    00580          LD      DE,3D10H
4255  7E        00590 COPY     LD      A,(HL)          ;Get next char
4256  B7        00600          OR      A               ;Null ends string print
4257  28FE      00610 HANG     JR      Z,HANG          ;Hang up when done
4259  12        00620          LD      (DE),A          ;Copy char to screen
425A  23        00630          INC     HL              ;Advance pointers
425B  13        00640          INC     DE
425C  18F7      00650          JR      COPY            ;Loop till done
425E  44        00660 DSKERR   DEFM    'Disk fault'
      69 73 6B 20 66 61 75 6C
      74
4268  00        00670          DEFB    0
                00680 ;
4269  CD9B42    00690 LOADER   CALL    RETRIG
426C  3A0642    00700          LD      A,(NTRAKS)  ;# tracks to load
426F  47        00710          LD      B,A
4270  3A0742    00720 TRKLOD   LD      A,(NSECTS)  ;# sectors/track
4273  4F        00730          LD      C,A
4274  C5        00740 SECLOD   PUSH    BC
4275  ED53D742  00750          LD      (DETEMP),DE ;Save load address
4279  060A      00760          LD      B,10        ;# tries per sector
427B  C5        00770 SECTRY   PUSH    BC          ;before aborting
427C  CDA142    00780          CALL    RDSEC       ;Try to read the sector
427F  C1        00790          POP     BC          ;Get Try Count back (in B)
4280  2809      00800          JR      Z,SECOK     ;If status not ok then retry
4282  ED5BD742  00810          LD      DE,(DETEMP) ;restore storage poiner
4286  10F3      00820          DJNZ    SECTRY      ;Decrement Try Counter
4288  C34242    00830          JP      ERROR
428B  C1        00840 SECOK    POP     BC          ;Get track/sector count back
428C  0D        00850          DEC     C           ;Decrement sector counter
428D  20E5      00860          JR      NZ,SECLOD   ;Loop till 6 sectors read
428F  05        00870          DEC     B           ;Advance 1 track
4290  C8        00880          RET     Z           ;Loop till all tracks read
4291  CD9B42    00890          CALL    RETRIG
4294  C5        00900          PUSH    BC
4295  CD2F42    00910          CALL    STEPIN      ;Step to next track
4298  C1        00920          POP     BC
4299  18D5      00930          JR      TRKLOD
                00940 ;
                00950 ;Keep drive turning
429B  3E01      00960 RETRIG   LD      A,1
429D  32E137    00970          LD      (DRIVE),A
42A0  C9        00980          RET
                00990 ;
                01000 ;Read next sector #C
42A1  36D0      01010 RDSEC    LD      (HL),0D0H   ;Interrupt controller
42A3  060A      01020          LD      B,10        ;Delay
42A5  10FE      01030 DELAY3   DJNZ    DELAY3
42A7  060A      01040          LD      B,10        ;Repeat
42A9  36D0      01050          LD      (HL),0D0H
42AB  10FE      01060 DELAY4   DJNZ    DELAY4
42AD  CD9B42    01070          CALL    RETRIG      ;Keep drive going
```

```
42B0  79       01080         LD      A,C          ;Get sector number
42B1  32EE37   01090         LD      (SECTOR),A
42B4  01EF37   01100         LD      BC,DATA
42B7  3E88     01110         LD      A,88H        ;Read sector command
42B9  77       01120         LD      (HL),A       ;Issue it to controller
42BA  3E05     01130         LD      A,5          ;Delay
42BC  3D       01140 DELAY5  DEC     A
42BD  20FD     01150         JR      NZ,DELAY5
42BF  7E       01160 GETDRQ  LD      A,(HL)       ;Read status register
42C0  CB4F     01170         BIT     1,A          ;test DRQ bit
42C2  C2D142   01180         JP      NZ,GETBYT    ;Yes get data
42C5  E601     01190         AND     1            ;No test busy bit
42C7  C2BF42   01200         JP      NZ,GETDRQ    ;Keep going while still busy
42CA  CD9B42   01210         CALL    RETRIG       ;else retrigger drive timer
42CD  7E       01220         LD      A,(HL)       ;Test Controller Status: CRC
42CE  E69C     01230         AND     9CH          ;RNF, Lost data, Drive ready
42D0  C9       01240         RET                  ;and exit
42D1  0A       01250 GETBYT  LD      A,(BC)       ;Read Controller Data Reg.
42D2  12       01260         LD      (DE),A       ;and store byte
42D3  13       01270         INC     DE           ;Advance store pointer
42D4  C3BF42   01280         JP      GETDRQ
               01290 ;
42D7  0000     01300 DETEMP  DEFW    0            ;store for DE in case sector
               01310 ;                             read must be restarted
4200           01320         END     BOOT
```

```
                00010 ;   Copyright (c) 1983 Hypersoft, M.J.Gingell
                00020 ;
                00030 ;Model III Bootstrap Loader, expects 6 sectors of
                00040 ;256 bytes, double density on each successive
                00050 ;track. Sector sequence 6,5,4,3,2,1
                00060 ;Version 1.1 May-15-83
4300            00070         ORG     04300H
                00080 ;                     and running program
                00090 ;FDC control register PORT addresses
00F4            00100 DRIVE   EQU     0F4H  ;Drive 0 motor trigger
00F0            00110 COMAND  EQU     0F0H  ;Command register
00F0            00120 STATUS  EQU     0F0H  ;Status register
00F1            00130 TRACK   EQU     0F1H  ;Track number register
00F2            00140 SECTOR  EQU     0F2H  ;Sector number register
00F3            00150 DATA    EQU     0F3H  ;Data I/O register
                00160 ;
4300  1806      00170 BOOT    JR      BOOT1
                00180 ;Table of alterable parameters
4302  0056      00190 LOADAD  DW      5600H   ;Address to start loading
4304  0056      00200 STRTAD  DW      5600H   ;Address to start executing
4306  0C        00210 NTRAKS  DB      12      ;Number of tracks
4307  06        00220 NSECTS  DB      6       ;Number of sectors/track
                00230 ;
4308  F3        00240 BOOT1   DI                      ;Disable interrupts
4309  31FF42    00250         LD      SP,BOOT-1       ;Set stack pointer
430C  ED5B0243  00260         LD      DE,(LOADAD)     ;DE is storage pointer
4310  CD2143    00270         CALL    TRACK1      ;Start Drive, Seek to Tr 1
4313  CD6D43    00280         CALL    LOADER          ;Load program
4316  2A0443    00290         LD      HL,(STRTAD)     ;Run it
4319  E9        00300         JP      (HL)
                00310 ;
431A  DBF0      00320 NOTBUS  IN      A,(STATUS)      ;Test if busy
431C  CB47      00330         BIT     0,A             ;Wait until not busy
431E  20FA      00340         JR      NZ,NOTBUS
4320  C9        00350         RET
                00360 ;
4321  CD9F43    00370 TRACK1  CALL    RETRIG          ;Select drive 0
4324  3ED0      00380         LD      A,0D0H          ;Force FDC interrupt
4326  D3F0      00390         OUT     (COMAND),A
4328  0600      00400         LD      B,0             ;Delay while command
432A  10FE      00410 DELAY1  DJNZ    DELAY1          ;takes effect
432C  D3F0      00420         OUT     (COMAND),A      ;Repeat
432E  10FE      00430 DELAY2  DJNZ    DELAY2
4330  CD1A43    00440 STEPIN  CALL    NOTBUS          ;Wait til not busy
4333  3E53      00450         LD      A,53H           ;Step out to
4335  D3F0      00460         OUT     (COMAND),A      ;next track
4337  060A      00470         LD      B,10            ;Delay
4339  10FE      00480 WREADY  DJNZ    WREADY
433B  CD1A43    00490 WRDY0   CALL    NOTBUS          ;Wait til not busy
433E  CD9F43    00500         CALL    RETRIG          ;Retrigger drive
4341  DBF0      00510         IN      A,(STATUS)      ;Get status
4343  CB67      00520         BIT     4,A             ;Seek error?
4345  C8        00530         RET     Z               ;No OK quit
                00540 ;Clear screen and print error message
```

```
4346  21003C    00550 ERROR   LD      HL,3C00H        ;Clear Screen
4349  11013C    00560         LD      DE,3C01H
434C  01FF03    00570         LD      BC,3FFH
434F  3620      00580         LD      (HL),20H
4351  EDB0      00590         LDIR
4353  216243    00600         LD      HL,DSKERR       ;FAULT message
4356  11103D    00610         LD      DE,3D10H
4359  7E        00620 COPY    LD      A,(HL)          ;Get next char
435A  B7        00630         OR      A               ;Null ends print
435B  28FE      00640 HANG    JR      Z,HANG          ;Hang up when done
435D  12        00650         LD      (DE),A          ;Copy char to screen
435E  23        00660         INC     HL              ;Advance pointers
435F  13        00670         INC     DE
4360  18F7      00680         JR      COPY            ;Loop till done
4362  44        00690 DSKERR  DEFM    'Disk fault'
      69 73 6B 20 66 61 75 6C 74
436C  00        00700         DEFB    0
                00710 ;
436D  CD9F43    00720 LOADER  CALL    RETRIG
4370  3A0643    00730         LD      A,(NTRAKS)
4373  47        00740         LD      B,A         ; # tracks
4374  3A0743    00750 TRKLOD  LD      A,(NSECTS)
4377  4F        00760         LD      C,A         ;6 sectors/track
4378  C5        00770 SECLOD  PUSH    BC
4379  ED53EB43  00780         LD      (DETEMP),DE ;Save store point
437D  060A      00790         LD      B,10        ;# tries per sector
437F  C5        00800 SECTRY  PUSH    BC          ;before aborting
4380  CDA443    00810         CALL    RDSEC
4383  C1        00820         POP     BC          ;Get Try Count back (in B)
4384  2809      00830         JR      Z,SECOK     ;If status not ok, retry
4386  ED5BEB43  00840         LD      DE,(DETEMP) ;restore storage poiner
438A  10F3      00850         DJNZ    SECTRY      ;Decrement Try Counter
438C  C34643    00860         JP      ERROR
438F  C1        00870 SECOK   POP     BC          ;Get trck/sctr counts back
4390  0D        00880         DEC     C           ;Decrement sector counter
4391  20E5      00890         JR      NZ,SECLOD   ;Loop till 6 sectors read
4393  05        00900         DEC     B           ;Advance 1 track
4394  C8        00910         RET     Z           ;Loop till all tracks read
4395  CD9F43    00920         CALL    RETRIG
4398  C5        00930         PUSH    BC
4399  CD3043    00940         CALL    STEPIN      ;Step to next track
439C  C1        00950         POP     BC
439D  18D5      00960         JR      TRKLOD
                00970 ;
                00980 ;Keep drive turning
439F  3E81      00990 RETRIG  LD      A,81H       ;double density, drive 0
43A1  D3F4      01000         OUT     (DRIVE),A
43A3  C9        01010         RET
                01020 ;
                01030 ;Read next sector #C
43A4  3ED0      01040 RDSEC   LD      A,0D0H      ;Interrupt controller
43A6  D3F0      01050         OUT     (COMAND),A
43A8  060A      01060         LD      B,10        ;Delay
43AA  10FE      01070 DELAY3  DJNZ    DELAY3
```

```
43AC  060A     01080          LD      B,10         ;Repeat
43AE  D3F0     01090          OUT     (COMAND),A
43B0  10FE     01100 DELAY4   DJNZ    DELAY4
43B2  CD9F43   01110          CALL    RETRIG       ;Keep drive going
43B5  79       01120          LD      A,C          ;Get sector number
43B6  D3F2     01130          OUT     (SECTOR),A
43B8  2683     01140          LD      H,83H        ;Bit masks
43BA  2E02     01150          LD      L,2
43BC  3E88     01160          LD      A,88H        ;Read sector command
43BE  D3F0     01170          OUT     (COMAND),A   ;Issue it to controller
43C0  3E05     01180          LD      A,5          ;Delay
43C2  3D       01190 DELAY5   DEC     A
43C3  20FD     01200          JR      NZ,DELAY5
43C5  DBF0     01210 RDSEC1   IN      A,(STATUS)   ;Read Status register
43C7  A4       01220          AND     H            ;test DRQ,BUSY READY
43C8  E2C543   01230          JP      PO,RDSEC1
43CB  DBF3     01240 RDSEC2   IN      A,(DATA)
43CD  12       01250          LD      (DE),A       ;Store it
43CE  13       01260          INC     DE           ;Advance pointer
43CF  DBF0     01270 RDSEC3   IN      A,(STATUS)
43D1  A5       01280          AND     L
43D2  20F7     01290          JR      NZ,RDSEC2
43D4  DBF0     01300          IN      A,(STATUS)
43D6  A5       01310          AND     L
43D7  20F2     01320          JR      NZ,RDSEC2
43D9  DBF0     01330          IN      A,(STATUS)
43DB  A5       01340          AND     L
43DC  20ED     01350          JR      NZ,RDSEC2
43DE  DBF0     01360          IN      A,(STATUS)
43E0  CB47     01370          BIT     0,A          ;Still busy ?
43E2  2804     01380          JR      Z,SEXIT
43E4  CB7F     01390          BIT     7,A          ;Not ready ?
43E6  28E7     01400          JR      Z,RDSEC3
43E8  E61C     01410 SEXIT    AND     1CH          ;Test RNF CRC Not Ready
43EA  C9       01420          RET
               01430 ;
43EB  0000     01440 DETEMP   DW      0            ;store for DE in case
               01450 ;                                  sector read must be restarted.
4300           01500          END     BOOT
```

Appendix 3: List of Autopilots

The following Autopilot programs are on your disk. These routines are provided by way of example and are not guaranteed to work in every case. They have been carefully tested and should work but variations in the quality of the original disks or your hardware may prevent correct operation. To access one, insert your Hyperzap disk (or a copy) into drive 0 and type ZGnn where nn is a number from 00 to 16. It will load into memory at 9800 hex and you can examine it by typing I9800 or run it by typing ZP.

Please don't use Hyperzap to make illegal copies of software. The TRS-80 software world is dying for lack of support. Use Hyperzap only to back up your own legitimately purchased master disks.

00 Directory of Autopilot programs on your disk.

01 Hyperzap self backup program. Put Hyperzap in drive 0 and a blank disk in drive 1 and type ZP.

02 Super Utility 2.2 Backup.

03 Super Utility 3.1 and 3.1a Backup.

04 Brandon Loader disk backup. For game disks using the loader and protection scheme by Paul Brandon. Med Systems disks such as Asylum and The Institute used this loader. Also disks from Melbourne House and Displayed Video may have used it for some of their programs.

05 40 track double sided disk copy example. To make a double sided copy you have to copy all of one side first and then change the side parameter for source and destination to point at the back side. Then you can copy the whole of the second side.

06 Super Utility 3.2 backup.

07 Kaypro II disk formatter. Example of the use of Hyperzap for making disks for use on machines other than the TRS-80.

08 Single Drive Hyperzap. Automatically zaps your Hyperzap so that it boots with both source and destination drives set to 0. Make a copy of Hyperzap first and then run this on it.

09 Single Density only Super Utility 3.2 backup. For Model I computers without a doubler. Copies just the single density part of the disk.

10 PC Visicalc backup. Duplicates your IBM PC Visicalc disk.

11 Powertool backup.

12 Montezuma version 2.2x CP/M 2.2 double sided system disk system tracks copy. Use this together with number 13 to turn your old double sided Montezuma 1.30 disks into Montezuma 2.2x disks. Take a master 2.2 disk and put it in drive 0 and use this AutoPilot to copy front and back of track 0 to a blank disk in drive 1. Then load Autopilot 13, put your

version 1.30 disk in drive 0 and use the autopilot to transfer tracks 1 to 39 from drive 0 to 1.

13 Montezuma version 1.30 double sided to 2.2x double sided copy for tracks 1 to 39. Watch it unscramble the mess of skew and front / back recording difference.

14 Funsoft same disk backup. Works on Apple Panic - may work on others. This is an example of a disk where the program is recorded with a track write. Apart from the loader on tracks 0 and 1 there are no sectors on any other tracks. This Autopilot does a track read, block move (fixing CRC's and cleaning things up) and then a track write.

15 Copycat 3 backup. Similar to Funsoft but double density.

16 Genie self modify. The Genie looks just like a TRS-80 Model 1 except the printer I/O is port instead of memory mapped. This zaps the Model 1 Hyperzap so that the printer output goes to the correct port. Use this on a backup of your Hyperzap disk.