

Advanced Disc Investigator

UVESFUBGPRHIZTM*
COPY A DISC

The Advanced Disc Investigator
(C)1986 ACP LTD. Serial# 00000

DISC:000000000000 TRACK:00 SECTOR:00 BYTES:0256 D/DRIVE:1 96 FM
TRACKS:00 T/ID:000 H/ID:000 S/ID:000 L/ID:001 SECTORS:10
OPERATION: RESULT:000 Successful Completion NORMAL DATA

A VERY POWERFUL DISC UTILITY ROM FOR STANDARD
AND NON-STANDARD DISCS

- BBC, B+, ELECTRON, MASTER SERIES & 
- ACORN 8271 DFS 0.9, DFS 1.20
- ACORN 1770 DFS, ADFS
(INCLUDING ACORN COMPATIBLE DFS's)

D I S C S A N D D R I V E S

RANDOM ACCESS

Those who read this column regularly will know that I have been most impressed with the Advanced Disc Toolkit that ACP released about a year ago. Since then the company has produced several further products for Acorn machines, the most recent being two new ROMs that will make the life of a disc user a great deal easier.

The first of these (and the one that is going to form the bulk of this column) is the *Advanced Disc Investigator*. A unique and most impressive feature of this ROM is that it can use either the 8271 FDC as fitted to the standard BBC B model with DFS, and also the 1770 FDC as fitted to the BBC B+, Electron Plus 3, and the Master. What is more, the ROM is intelligent enough to detect which chip is in use and adjust its operations to match. I find this most remarkable and can recommend it, like ADT, to the user of any Acorn machine.

It comes in the usual ACP fold-over packaging with a 28 page manual bound in black and silver, and at £28.95, you might think that this ROM needs to be good to to justify its price. You won't be disappointed. I think that it is the very best product of its type that I have ever seen.

On using the ADI, the first difference you notice is that it is called as a language and not as a utility or service ROM. This makes a significant difference to the ease of use of the commands and I cannot think why anyone didn't think of this before now. Upon typing *ADI you are presented with the front screen (Figure 1). This comprises two active windows and an identification panel. The panel (top right) contains the ROM name and number whilst the first window (top left) is the menu window and allows you to choose the particular operation that you require. The bottom window is the status display and contains all the present settings and any error or completion messages that are required.

The options that can be selected from the menu window are as follows:

- (C)opy a disc
- (V)erify two discs

Two new ROMs from Advanced Computer Products which are likely to be of interest to all disc users

- (E)dit a sector
- (S)can a disc
- (F)ormat tracks
- (U)nformat tracks
- (M)odify format parameters
- (G)et sectors from disc
- (P)ut sectors to disc
- (R)ead track data
- (W)rite track data
- (B)uffer examine

- (Z)ero buffer
- (T)oggle printer (on/off)
- (I)nitilise status
- (*) Command

These commands are selected using the cursor keys to highlight the command required and the Return key to select. They can also be selected direct from the keyboard by using the correct key (ie S for Scan etc). This means that while you are learning, you can get a readout of the command meaning as you move around using the cursors but once you have memorised the mnemonics, you can get a very fast response by using the keyboard key letters. All of the commands which are likely to affect a disc have an "Are you sure?" type message to prevent accidental errors occurring.

In all the other ROMs of this type I have used you have to issue each request to the ROM as a * command line with all the correct parameters following. The second or lower window makes this type of memory task much simpler. The parameters of the command are all displayed for you to see (Figure 1 makes this clear) and it is these parameters that the command uses when you select it. For example, if you want to format only ten tracks of a disc starting at track 23, you set the track parameter to 23 and the tracks parameter to 10 and the command will do just as you require. These parameters are selected using the left and right cursors with Shift, and are altered by the cursor up and down keys — simplicity itself and much easier than using the normal command line method.

Most of the commands are fairly self explanatory and are implemented in such a way that they are very easy indeed to use. Copy a disc does exactly that and will produce a very faithful copy including any strange and clever formatting techniques. Figure 2 shows the display part way through a copy operation. Verify two discs allows you to check that you have a complete and exact copy of a disc by checking every track and sector on one against those on the other (Figure 3). Edit a sector allows you to view and

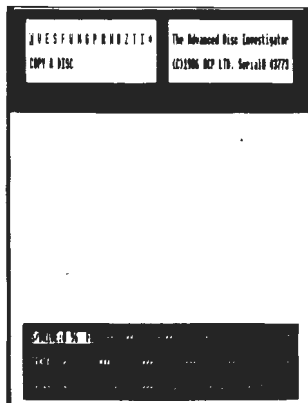


Figure 1 Main menu

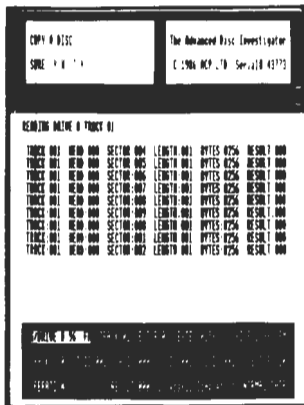


Figure 2 Copying a disc

edit any sector on the disc (Figure 4) and even display the sector as 65C02 assembler language if you choose. Editing can be in hex or ASCII and is changed from one to another by pressing Copy cursor keys. Shift/cursor keys are used to move around the text simply and quickly.

Scanning a disc is essential when you want to understand how a particular disc has been formatted and written. It gives you all the information that was written to the disc surface at that time and is essential when you are copying strange disc formats. It is used by the *Copy a disc* command so you might as well include the option from the menu too. The *Format tracks* command isn't quite what you might expect. Yes it does format tracks as set by the parameter window, but it won't format an entire disc: it doesn't write the catalogue sectors correctly so be careful when using this command. Unformatting is a useful function if you wish to write specially

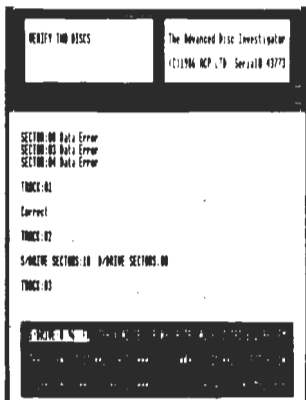


Figure 3 Dual verification

formatted discs or don't possess a bulk eraser. It basically writes a special format to the disc so that it cannot be detected as a correctly formatted track later.

All the commands covered so far are fairly straightforward but *Modify format parameters* is a very strange, and useful, addition that would make the day of those people who really like to mess about with discs. When selected it displays the current skew and field gap settings (see Figure 6) and then allows you to alter them as required. The skew is the difference between the actual sector number and the logical sector number. Every disc has a special index hole at the centre which is used by the drive unit to decide if the disc is spinning at the correct rate and to show where sector 0 starts. Although sector 0 should be the first sector after the hole, it doesn't need to be and by "skewing" the sector numbers slightly you can speed up a disc so that it can be accessed more quickly. The reasoning goes something like this. There is always a slight wait between tracks as the motor steps outwards so if we can make sure that the sector under the head after this stepping operation is sec-

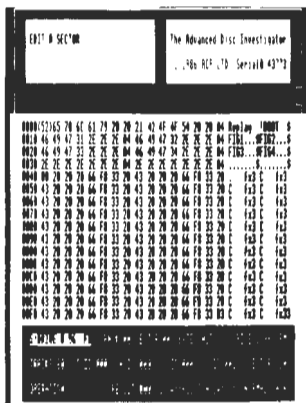


Figure 4 Editing

tor 0, then the disc won't have to be rotated another time in order to find sector 0. By careful calculation you can estimate the actual sector which will be under the head, and so by making that sector logical number 0, you can find the correct skew value. Three is a good value for fast drives formatted with ten sectors per track (ie single density). Gap sizes have been covered recently in this column and so all I will add is that this chip will allow you play to your heart's content without any need to write your own programs!

The two sector commands (Get and Put) are fairly obvious and allow you to claim sectors from the disc and then re-write them in a different place or in the same place after modification. Those using the B271 FDC will be able to set the sector length sector by sector if they so desire. Track data can be

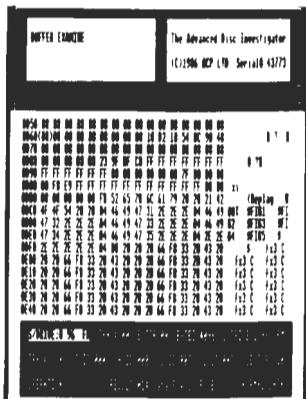


Figure 5 Buffer display

written or read on the 1770 FDC using the Write and Read track commands. This is similar to the sector commands but allows you to read and write the sector identifiers as well. Both these commands require a buffer and this can then be examined or zeroed. In Figure 5 you can see the contents of the buffer after a track read command and some of the gap and track data as well as the catalogue information. Printers can be supported and you can return the parameters to their normal initial setting — an essential feature as this isn't automatic when you end a command by escaping.

That brings us to the end of the main menu features except to mention that the chip contains a few "star" commands as well. These aren't very well explained and in fact if I have any criticism at all, it must be that the documentation could have been better. It is

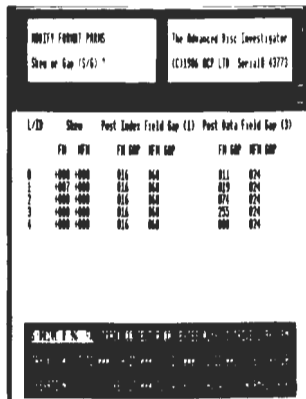


Figure 6 Format parameters

D I S C S A N D D R I V E S

generally adequate, but as the chip is so exceptional, I had expected a better, longer and simpler manual. I know that my readers are hungry for as much information and detail as possible and this manual could have had more.

But back to the star commands. The first of these is *OSWORD7F — the by now familiar "bash the FDC" command. This allows you to pass directly all the information that the FDC requires to act upon a particular command. You will need to have a full list of the commands though and they aren't easy to find. But the author of the chip hasn't left you completely on your own as he has included a most complete listing of the commands inside the chip itself. All you have to do to access them is type *HELP FDC. There are other HELP facilities too. *HELP RESULT gives a read out of all the possible error messages that the DFS system can produce and *HELP TRACK gives a readout of all the normal gap and disc information for both DFS and ADFS format commands. These printouts are shown as Tables 1-3.

In all I have to say that this is an essential ROM to have if you're serious about discs. Get it for your birthday, Christmas or just to celebrate the purchase of a new box of discs. It is exceptional and will bring you a great deal of pleasure.

File too long

I had intended to cover the new Advanced ROM Manager in this issue too but space won't permit more than a summary I'm afraid.

This ROM is more 'normal' than ADI as it uses a number of "star" commands for its operation. They are:

*AUTOROM which will format a program, eg NAME, in such a way that it may be written to an EPROM or to sideways RAM and then executed by typing simply *NAME.

*GOROM which will execute a particular routine in a particular ROM — probably most useful for debugging when writing your own ROMs.

*KILL turns a ROM off.

*MAKEROM creates a ROM Filing System format ROM which can then be used as instant and permanent program storage.

*OFFER routes a certain command to the correct ROM and prevents command clashes.

*RDUMP displays the contents of a ROM in a whole range of possible formats including disassembler.

*REX allows both examination and editing of a ROM or RAM image using the same control keys as those described above under edit.

*RLOAD will load a ROM image into a sideways RAM bank if fitted.

*RMOVE copies all or part of a ROM or RAM image into normal RAM for modification or examination.

*ROMS gives a read out of the currently active ROMs.

*RSAVE is the opposite of *RLOAD but will work on both RAM and ROM.

*RSUM will calculate a checksum and CRC

for a particular ROM. By keeping a record of this almost unique number you can check for alterations etc.

As I hope you can see from even this short listing of features, there is a great deal that this ROM can do and it's likely to be of great interest to all those who want to use sideways RAM in an innovative and useful way. Well I have run right out of room yet again

and I haven't even covered your letters but there will be a few next month when I will also look at the new Advanced 1770 DFS — the very first system to offer complete command compatibility between single and double density discs. If you have any burning questions, drop me a line at the usual address and I will do my very best to feature your letter.

Table 1 *HELP FDC

COMMAND	OPERATION	PARAMETERS
*35	Specify Command	
	Define Drive Characteristics	*0D, Step Rate, H/S Time, H/L Time
	Define Bad Tracks Drive 0	*10, 1st Track, 2nd Track, Track
	Define Bad Tracks Drive 1	*18, 1st Track, 2nd Track, Track
*40	Scan Sector	Track, Sector, L/N, T/S, Length
*44	Scan Deleted Sector	Track, Sector, L/N, T/S, Length
*4A	Write Sector	Track, Sector
*4AB	Write Multiple Sectors	Track, Sector, L/N
*4AE	Write Deleted Sector	Track, Sector
*4AF	Write Multiple Deleted Sectors	Track, Sector, L/N
*452	Read Sector	Track, Sector
*453	Read Multiple Sectors	Track, Sector, L/N
*456	Read Deleted Sector	Track, Sector
*457	Read Multiple Deleted Sectors	Track, Sector, L/N
*45B	Read Track ID	Track, 0, Field Count
*45E	Verify Deleted Sector	Track, Sector
*45F	Verify Multiple Deleted Sectors	Track, Sector, L/N
*463	Format	Track, Gap 3, L/N, Gap 5, Gap 1
*469	Seek	Track
*46C	Read Drive Status	
*47A	Write Special Registers	
*47B	Read Special Registers	
*47D	Scan Sector Number	*06, Data
	Scan Block Number	*14, Data
	Scan Byte Count	*13, Data
	* Current Track Register Drive 0	*12, Data
	* Current Track Register Drive 1	*1A, Data
	Output Port	*23, Data
	Bad Track Register 1 Drive 0	*10, Data
	Bad Track Register 2 Drive 0	*11, Data
	Bad Track Register 1 Drive 1	*18, Data
	Bad Track Register 2 Drive 1	*19, Data
*4E0	Read Track (1770)	Track
*4F0	Write Track (1770)	Track

Table 2 *HELP RESULT

RESULT	MEANING	
*00	Successful Completion	*0E - Data Field CRC Error
*02	Scan Comparison Found	*10 - Drive Not Ready
*04	Scan Range Found	*12 - Disc Write Protected
*06	Not Used	*14 - Track 00 Not Found
*08	Clock Error	*16 - Write Fault
*0A	Late DMA	*18 - Sector Not Found
*0C	ID Field CRC Error	*1A - Not Used
		*1C - Not Used
		*1E - Not Used
		*20 - Deleted Data Found

Table 3 *HELP TRACK

FIELD	PH	VALUE	LENGTH	MPH	VALUE	LENGTH
Post Index Field Gap (1)	*5F	P		-	*4E	P
	-	*00	6			
Sector ID Field				-	*00	12
				-	*A1	3
ID Address Mark	-	*FE	1	-	*FE	1
T/ID	-	Track	1	-	Track	1
H/ID	-	Head	1	-	Head	1
S/ID	-	Sector	1	-	Sector	1
L/ID	-	Length	1	-	Length	1
CRC	-	CRC	2	-	CRC	2
Post ID Field Gap (2)	*5F	P		-	*4E	22
	-	*00	6	-	*00	12
				-	*A1	3
Data Field				-		
Data Address Mark	-	*FB	1	-	*FB	1
Data Bytes	-	Data	Sector	-	Data	Sector
CRC	-	CRC	2	-	CRC	2
Post Data Field Gap (3)	*5F	P		-	*4E	P
	-	*00	6			
Final Gap (4)	-	*5P	-	-	*4E	-