

SEDORIC DOS
Version 2.0

A U S E R' S G U I D E

based on the French SEDORIC V1.0 manual
by Fabrice Broche and Denis Sebbag

by Jon Haworth and Allan Whitaker

Copyright 1990, 1991, 1992
Issue 3 - First Revision November 1992

C O N T E N T S

INTRODUCTION	1
Conventions and notations	2
Filenames	3
DISC OPERATING COMMANDS	4
Disc directory	5
Loading and saving a file	6
Deleting and modifying a file	8
Initialising and copying of discs	10
Disc configuration	12
EXTENDED BASIC	14
BASIC programming aids	14
Changing the keyboard	17
Function key set-up	19
String commands	21
Error handling routines	22
Text formatting	23
Formatted screen/printing	25
Printer output	26
Graphic instructions	27
User commands	28
Other commands	29
DATA FILES	31
Sequential files	31
Random access files	34
Transfers between buffer and disc	35
Working on the buffer	36
Transfers between fields/variables	37
Disc access	38
Sector reservation	39
INDEX OF DOS KEYWORDS	40
APPENDICES	41
DOS Keyword Addresses and Codes	41
Error Messages	42
Function Key Codes	43
System Variables	44
Disc Structure	45
File Byte Status Coding	46
Keyboard Codes for KEYIF Instruction	46
Switching between RAM and ROM	46
Utility Files on SEDORIC Master Disc	47
Known bugs and limitations - V1.006	49
Developments and solutions - V1.007/V2.0	49

SEDORIC - A USER'S GUIDE

INTRODUCTION

SEDORIC is a fast, sophisticated Disc Operating System (DOS) for the Oric computer. It is capable of reading or writing 24 kilobytes of code in less than 3 seconds. Despite the sophisticated features of the DOS the amount of memory taken away from the user is minimal. Pages 1 and 2 are not used. On booting the SEDORIC DOS disc the DOS code is loaded into memory addresses #C000 to #FFFF in RAM overlay. The DOS makes use of memory locations in page 0, and page 4 is entirely allocated to SEDORIC. The Oric ROM is switched in and out as needed in order to access the DOS or BASIC.

Booting SEDORIC carries out a NEW command on any program in memory, though a BASIC program can be recovered with the OLD command. Before control is passed to BASIC, SEDORIC executes a series of initialisation instructions particular to the disc. These may be modified by the user, to include the loading and execution of your own program to personalise the boot. Two types of discs can be formatted - a 'Master', containing the full DOS, and a 'Slave' which can only be used when the DOS is already in memory. If the DOS is missing, the error message '**WARNING** DOS is altered!' is displayed, requiring the insertion of a Master disc into the system drive if the DOS in memory has been affected.

To display the contents of your SEDORIC disc, type DIR and press RETURN. On the Oric Atmos you can press FUNCT and] together instead. To load a program displayed in the directory, simply type in its name and press RETURN. To exit and reboot SEDORIC, just press the disc drive RESET button.

To save a BASIC program, type SAVE "NAME" and RETURN. A DIR will prove it has been saved to disc. You will note that the directory of the disc displays the file as NAME.COM. The '.COM' is known as the filename. To recall a program, just type its name (as before), i.e. NAME + RETURN. You can also type LOAD "NAME" which has the same result.

If you wish to remove a file from the disc, type DEL "NAME" and RETURN. To rename a program, use REN. Thus REN "NAME" TO "TEST" will change the name of 'NAME' to 'TEST'.

To initialise a blank disc for use by SEDORIC only requires that you type INIT and RETURN. You will probably come across the message "INSERT MASTER DISC IN DRIVE A". Do so, and press RETURN; the routine you are using will then be loaded into memory. This is necessary because the whole of the DOS is too large for the available memory. Certain commands therefore have their routines stored on disc ready to be loaded when required. Once loaded the command can be used repeatedly until another such command routine has to be loaded (e.g. BACKUP) or the computer is switched off. Put in your blank disc, then answer the displayed questions.

After this brief introduction, you are recommended to read the section entitled 'Disc operating commands' starting on page 4

of the manual to discover the full potential of SEDORIC. Furthermore, SEDORIC provides you with some 60 additional BASIC instructions, and a comprehensive suite of sequential and random access file routines. You will find plenty to keep you occupied while learning to get the best out of SEDORIC!

Note that the SEDORIC disc contains several utilities in addition to the DOS. These are described in detail in the final appendix, but take particular note of the CONVERT utility to transfer Oric DOS programs to SEDORIC format.

SEDORIC DOS makes full use of the FUNCT key on the Oric Atmos. In conjunction with the other keys it can provide many functions which are used simply by pressing FUNCT and another key or keys. You can set up your own function keys with the KEYDEF and KEYUSE commands. Thus, FUNCT and the / key executes the LIST command; FUNCT and RETURN together provides automatic numbering of program lines.

A new control character - CTRL P - has been added to provide an on/off toggle for the cursor flash. Within programs use PRINT CHR\$(16).

Facilities are provided to avoid inopportune system 'crashes'. SEDORIC controls any BREAK and displays a 'BREAK ON BYTE #XXXX' message. The ESC key provides a way out of all interactive commands such as DIR, DEL and INIT.

Whereas all Disc Operating Systems previously offered for the Oric computers have required keywords to be prefixed with the exclamation mark (!) SEDORIC DOS allows the entry of keywords without this need. The exception is following the use of the QUIT command which makes the prefixing of DOS keywords or filenames (in direct mode) with the exclamation mark mandatory.

Finally, all DOS keywords can be typed in upper or lower case, or even in a combination of the two. BASIC keywords can safely be used in filenames, providing they do not form the start of a filename being loaded in direct mode, e.g. STOPPER. Filenames can also be typed in lower case letters although if BASIC keywords are embedded they must always be typed in upper case, as in 'STOPper'.

Conventions and notations

The following conventions and notations are used in this manual:

A **COMMAND** appears in bold capital letters, general syntax in ordinary capitals. Optional parameters are enclosed in brackets. A command is entered by pressing the RETURN key.

Abbreviations:

Drive - the parameter is a drive number - A, B, C or D (as in MS-DOS). The default is the current drive.

- FN - File name required, must not contain wildcards
- FNA - File name ambiguous - File name required, but can include wild card characters (? and *). It is possible to omit the filename altogether or just to specify the drive name.
- NE - Numeric expression
- AE - Alphanumeric (string) expression
- NV - Numeric variable - a variable name is required
- AV - Alphanumeric (string) variable
- LN - Logical number - refers to a block number used in the control of files.
- FLN - Field name - used in random access file control

The ESC key enables you to exit from all positions where input is required from the user.

DISC OPERATING COMMANDS

(drive)-

On its own, it sets the default (or current) drive.

EXT AE

Specifies the default file name extension to 3 characters.

EXT ?

Displays the current default extension.

EXT

Resets the default extension to .COM

DISC DIRECTORY

DIR FNA

Catalogues the disc. The display can be interrupted by pressing a key; press the space bar to restart, ESC to exit.

LDIR FNA

Catalogues the disc sending the output to the printer.

SEARCH FNA

Checks if a file is on disc.

LOADING AND SAVING A FILE

LOAD FN (,A,NE) (,V) (,J) (,N)

Loads a file. Options:

, A, NE	loads the file at the address specified in NE
, J	joins the file to a BASIC program
, N	stops autorun after loading
, V	shows the file status, but does not load it.

(FILENAME) (,A,NE) (,N) (,J) (,V)

It is unnecessary to use LOAD unless the filename is expressed as a variable, e.g. LOAD A\$. It is enough simply to type the filename to load it - provided the filename does not start with a BASIC keyword or number. If it does, specify the drive, e.g. A-LOADER.

Note that after QUIT is used, the ! must be used as a prefix to the filename.

```
SAVE FN (, A NE) (, E NE) (,T NE) (,AUTO)
SAVEO " " " " " "
SAVEM " " " " " "
SAVEU " " " " " "
```

saves the file with the name FN to disc.

The four save commands only have different effects if the filename already exists on disc.

If the file already exists:

SAVE	gives a 'File already exists' message
SAVEO	overwrites the existing file unless it is protected
SAVEM	the new file saved is appended to the existing file on disc
SAVEU	the existing file is preserved as a .BAK file. Any existing .BAK file is lost.

ESAVE FN

Saves the current screen (TEXT or HIRES).

DELETING AND MODIFYING A FILE

DEL FNA

Deletes an unprotected file.

DESTROY FNA

DANGEROUS! As for DEL, but with no request for confirmation.

DELBAK (drive)

Deletes all .BAK files.

REN FNA(old) TO FNA

Renames a file.

STATUS FN (,A NE) (,T NE) (,AUTO)

Allows you to change the status of a file without having to load and save it, following the syntax for SAVE. If no option is specified, autorun is removed.

PROT FNA

Protects a file from deletion or overwriting. A letter 'P' is displayed by the filename in the directory.

UNPROT FNA

Removes protection from the file.

INITIALISING AND COPYING OF DISCS

INIT (drive, no. of sectors, no. of tracks ,S or ,D)

To initialise a disc, follow the on-screen instructions (use ESC to abort).

BACKUP (drive) (TO) (drive)

Copies one disc (the source) to another (the target).

COPY (COPYM or COPYO) (FNA) (TO FNA) (,C) (,N)

Copy files - M and O work as for SAVE.

DISC CONFIGURATION

SYS

Displays the DOS configuration in memory (not that on disc).

DSYS (drive)

Displays the DOS configuration saved on the specified disc.

DNAME (drive) AE

Modifies the specified disc name, displaying the old name and then requesting and writing the new one.

INIST (drive) AE

Modifies the initialisation instructions on the specified disc, allowing you to personalise the start-up routines.

DKEY (drive) (,A) (,S)

Sets the disc start-up keyboard configuration - QWERTY if no option is specified, AZERTY with , A , with accents with , S .

TRACK (NEA) (, NEB) (, NEC) (, NED)

Modifies (only in memory) the disc drive configuration by setting the number of tracks for each drive.

DTRACK (drive) (NEA) (, NEB) (, NEC) (, NED)

As the above command, but the change is made on disc and not the DOS in RAM.

DNUM (drive) (,NE) (,NE)

Modifies on the disc the default initialisation values for respectively the first line and step values for the renumbering commands NUM and RENUM and for auto line numbering.

SYSTEM (drive)

Specifies the disc drive on which is the DOS containing transient command files.

EXTENDED BASIC

BASIC PROGRAMMING AIDS

RENUM (NE1) (, NE2) (, NE3) (, NE4)

Renumbers all or a block of a BASIC program.

- NE1 new first line number
- NE2 line increment required
- NE3 and NE4 the first and last line numbers of the program to be renumbered.

MERGE FN (,L)

Merges the BASIC program in memory and the BASIC program specified.

DELETE (NE1) - (NE2)

Deletes a block of lines from a program without losing variable values.

SEEK (AE) (,S) (,M)

SEEK AE lists all program lines in which the string AE exists.

CHANGE AE1 TO AE2

Replaces all occurrences of chain AE1 by the chain AE2 in a BASIC program.

NUM (NE1) (, NE2)

Sets the parameters for automatic line numbering. NE1 is the start line, and NE2 the increment. Each press of FUNCT + RETURN ends the current line and numbers the next.

NUM END

If this command is entered, the DOS automatically finds the number of the last line of a BASIC program already in memory, and starts automatic line numbering at the line following.

CHANGING THE KEYBOARD

KEY SET or KEY OFF

Inhibits (OFF) or authorises (SET) the keyboard.

KEYIF NE GOTO or KEYIF NE THEN

Seeks a keypress set by the argument NE.

QWERTY

Sets the keyboard to normal (usually after an AZERTY) and executes an ACCENT OFF.

AZERTY

Sets to French keyboard.

ACCENT SET / ACCENT OFF

Redefines certain characters to show accents on screen.

FUNCTION KEY SET-UP

A function can be assigned to any key in two ways. The first is by the allocation of a pre-determined command using KEYDEF, the second the allocation of a command defined by the user utilising KEYDEF and KEYUSE.

KEYDEF NE

Permits the definition of function keys. The argument is the function code. After entering the command, press the key to which you wish to assign the function.

KEYUSE NE, AE

Only works on the Atmos. It permits the definition of the 16 user commands (with one or more commands each).

KEYSAVE FN

Saves the current function key configuration on disc.

VUSER

Displays the defined command chains for the 16 user commands.

STRING COMMANDS

TKEN AV

Encodes the string variable as a suite of BASIC instructions which can be executed by the use of STRUN.

UNTKEN AV

The inverse of TKEN. Decodes the encoded chain into its original form.

STRUN AV

Executes a chain encoded with TKEN as if it were a BASIC line.

INSTR AE1 , AE2 , NE

Searches for the first occurrence of the string AE2 within the string AE1 , starting at the character position specified in NE.

ERROR HANDLING ROUTINES

ERR SET or ERR OFF

So that the programmer can handle errors generated by the DOS the ERR SET command prevents the interruption of a program by a DOS error.

ERR GOTO NE

Specifies the line number where DOS errors are handled provided an ERR SET has been executed.

RESUME (NEXT)

Restarts execution where it was interrupted (RESUME) or at the following line (RESUME NEXT).

ERROR NE

Generates an error condition with the specified number held at location #4FD.

TEXT FORMATTING

LINPUT (@NE, NE,) (AE,) NE; AV (, E) (, S) (, C) (, J) (, K)
LINPUT (@X, Y,) (Character,) Length; AV...

An instruction to format text input - not numeric variables. It allows entry of text of the length specified, which is assigned to the string variable AV.

This is a very powerful instruction which needs to be used progressively to realise its full potential.

CREATEW (FN)

Creates a screen area of 25 lines by 40 characters, starting at screen line two, for use by the WINDOW command.

WINDOW (FN)

Enables the loading of such text screen areas - not usable in Hires.

FORMATTED SCREEN/PRINTING

USING NE, AE (,AV)

This command formats numbers printed to the screen; the number is the numeric variable NE, its format is defined in the string variable AE.

LUSING NE, AE (, AV)

As for USING, but outputs to the printer.

WIDTH (NE) or WIDTH LPRINT (NE)

Sets the usable screen width (WIDTH) or printer width. NE is the number of characters per line to be printed before a carriage return (CR) and line feed (LF) are automatically generated.

PRINTER OUTPUT

OUT NE

Sends an ASCII code to the printer and is the equivalent of LPRINT CHR\$(NE).

PR SET or PR OFF

Switches the printer in or out.

GRAPHICS INSTRUCTIONS

These commands are constructed around the variable AN, representing the angle to be used. AN is expressed in degrees anticlockwise from EAST. Clockwise values may be expressed as a negative number of degrees.

LINE NE, FB code

Draw NE points in the current direction.

BOX NE1, NE2, FB code

Draws a rectangle, with the current cursor position forming the top left corner. It will be angled to the current value of AN.

LCUR

Returns the horizontal and vertical coordinates of the cursor in Text mode in the variables CX and CY.

HCUR

As above for the Hires screen.

USER COMMANDS

USER x, DEF address (,0)

USER x (, A NE1) (, X NE2) (, Y NE3) (, P NE4)

Allows the call of assembly language sub-routines, with the ability to pass parameters.

] (right bracket)

As for the "!" in non-disc systems, used to call machine code routines from BASIC.

OTHER COMMANDS

QUIT

Resets the pointers used by the DOS, resetting the IRQ and NMI vectors. Disables the FUNCTION keys, and makes the use of "!" obligatory for DOS commands.

The instruction is necessary before running any program that uses Page 4 of memory or modifies the IRQ/NMI vectors itself.

RESET

A command having the same effect as the button on the disc drive.

RESTORE (No. of line)

Places the DATA pointer at the start of the program or at the specified line (which does not have to exist).

MOVE AEstart , AEend , AEtargat address

Moves a block of memory.

OLD

Recovers a BASIC program lost by a BOOT or a NEW.

RANDOM NE

Initialises the random number generator.

SWAP Variable, Variable

Exchanges the contents of the specified variables.

DATA FILES

SEDORIC DOS is capable of both sequential and random access to data files on disc. Each file has a logical number (LN) assigned to it to simplify the command syntax. It is necessary to open a file before accessing it, i.e. to allocate to it a logical number and associate it with a filename on disc.

SEQUENTIAL FILES

In sequential access of files the data has to be read in the order in which it was written.

OPEN S, FN, LN

Opens a sequential file, reserves a memory buffer, and sets the pointer to start of file.

CLOSE (LN, LN, ...)

Frees the buffer reserved by OPEN and closes any open files.

PUT LN, list of variables

Writes the variable contents to a file.

TAKE LN, list of variables

Reads from file LN the specified variables.

APPEND LN

Places the file pointer at the end of the specified file.

REWIND LN

Places the file pointer at the start of the specified file.

JUMP LN, number of data items

Moves the file pointer the number of items specified.

BUILD LN

Enables you to build a sequential file by keyboard entry.

TYPE LN

Allows you to display the contents of a sequential file, starting at the current position of the file pointer.

LTYPE LN

As for TYPE, but on the printer.

&(LN)

Returns 0 (false) if at the end of file, and -1 (true) otherwise.

&(-LN)

Returns the type of the next data item to read.

RANDOM ACCESS FILES

Random access is very different from sequential access in that the data is organised within the FILE into RECORDS. Each record contains all the relevant information on the item recorded, and each has its own identity so that it can rapidly be accessed on disc.

Each record contains all relevant information grouped in FIELDS.

The record is the entity that is transferred from the disc to memory and vice-versa. To facilitate file operations, the commands work on a memory buffer holding the whole of the relevant record.

OPEN R, FN, LN (, record length, number of records reserved)

Opens a file. It is essential that you specify the record length and number of records when first opening a file.

CLOSE (LN, LN ...)

Closes the file numbered LN (or all files) and frees the buffer.

TRANSFERS BETWEEN BUFFER AND DISC

TAKE LN, Record no.

Loads the specified record from disc into the buffer.

PUT LN, Record no.

Transfers the specified record from the buffer to the file on disc.

WORKING ON THE BUFFER

To facilitate the classification of data, the record can be further divided into fields, enabling the place within the record where the data will be held to be defined. The fields are rather like variables; you may give them a value and read them as with variables.

FIELD LN, FLN TO type (, ...)

Defines fields within the each record of the file.

TRANSFERS BETWEEN FIELDS AND VARIABLES

RSET FLN < Expression

Writes the given value to a field - both must be of the same type.

LSET FLN < Expression (, ...)

Identical to RSET for numeric and integer fields.

FLN > Variable

Read field of name FLN and put into the specified variable, which must be of the same type.

& (LN)

Display the number of records in file number LN

DISC ACCESS

There is a special kind of file which allows you to modify the disc, sector by sector, directly from BASIC. It enables the experienced programmer to create a filing system particular to the needs of the application.

OPEN D, LN (,Drive)

Reserves the buffer of 256 bytes and opens the file FLN.

CLOSE (LN (,LN....))

Closes the file(s) specified and frees the buffer.

TAKE LN, Track, Sector (,Drive)

Reads the specified sector into the buffer.

PUT LN, Track, Sector (,Drive)

As TAKE, but used to write, not read the sector.

SECTOR RESERVATION

It is possible to search for free sectors on the disc, as well as to free certain sectors. All this information is stored in special sectors (Track 20, Sectors 2 & 3) called BITMAP. To reserve a sector it is necessary to read BITMAP, determine the free the sector using CRESEC, and rewrite BITMAP.

PMAP Drive

Reads the BITMAP into memory.

SMAP Drive

Writes to disc the BITMAP in memory.

CRESEC

Returns in the system variables FP and FS the track and sector address of a free sector.

FRSEC NTrack, NSector

Frees the specified sector and increments the number of free sectors.

APPENDIX - Utility Files on SEDORIC Master Disc

The SEDORIC master disc contains several programs and utilities.

CONVERT