the expert

WITH ESM

Multi-Menu V4.2

USER GUIDE

G64 ATTACHED
TO THIS PDF

Contents

Introduction

Welcome to the Expert Multi-Menu V4.2.

The instructions have been supplied on disk so that a new manual does not have to be printed every time we make any improvements or corrections. To get a printout of any of the module instructions use the 'Printout' program on side 2 of this disk.

The strength of this operating system lies in it's modular approach. Till now the expert cartridge could only perform one major operation at a time - sprite steal, picture steal, memory backup etc were all routines contained in seperate modules.

This new operating system bypasses that problem by making it unnecessary. Each module has a specific purpose, and you are able to go between modules via the menu without corrupting the program in memory.

This is a modular operating sytem and can be expanded - please let us know of any kind of utilities you would like to see on future versions.

All modules have been fully tested and are, as far as we can tell, bug-free! If you think you have found something wrong with any of the modules or even if you would just like to make a suggestion please write to us.

NOW Computer Systems
19, Beadon Road,
Bromley,
Kent BR2 9AS

General Information

Memory Management

When using the machine code monitors you will be able to edit, load and save the following sections of memory:-

RAM :	64K
ROM :	16K - BASIC and KERNAL ROMs (A000-BFFF & E000-FFFF) Colour RAM (D800-DBFF) I/O parameter block (DC00-DC6F)

The parameter block contains information from the I/O ROM (D000-DFFF), Status and registers as they were when the program was stopped.

DC00-DC2E = D000-D02E VIC chip
DC2F-DC47 = D400-D418 SID chip
DC48-DC57 = DC00-DC0F CIA #1
DC58-DC67 = DD00-DD0F CIA #2
DC68-DC6F = registers

Registers

DC68 = Status register
DC69 = Program counter lo
DC6A = Program counter hi
DC6B = Accumulator
DC6C = X-register
DC6D = Y-register
DC6E = Stack pointer
DC6F = Q-value

The Q-value is the hibyte of the address which the backup process uses as a work-space when it comes to decompressing the program you have backed-up.

If, for instance, the value is 73 then the workspace address will be 7300. The Expert will automatically find this value but gives you the option to enter your own value. The Q-value must be in the range 02-FF.

In the monitor, the Q-value is preceded by a minus when it can't find sufficient space and is using the screen instead.

The Screen-Viewer

All modules incorporate a screen-viewer which allows you to view the screen as it was when you entered the Expert. In some cases the sprites or even the screen may become corrupted. This is due to the Expert cartridge displaying an image of itself in the top quarter of every video bank:

Bank 0	3000-3FFF
Bank 1	7000-7FFF
Bank 2	B000-BFFF
Bank 3	F000-FFFF

If the screen display has any graphics which reside in these sections of memory then the display will be corrupted. This is much the same as the character image which appears at 1000-1FFF & 9000-9FFF. It doesn't corrupt memory, but you can't display any graphics residing in these sections.

The Multi-Menu

You cannot load the multi-menu if there is a backup in memory. You can clear the backup pointers from the backup module by pressing 'F8'. The backup will still be in memory but can no longer be saved.

To load the multi-menu from any of the modules you press the 'M' key - except the machine code monitor which requires 'MM (RETURN)'.

The bottom line of the screen will then display the following message:-

RESET DRIVE,INSERT MENU DISK,PRESS SPACE

Insert the multi-menu disk and hold down the spacebar - it auto-repeats. The screen will blank and the menu will load and be displayed.

If you wish to quit the menu press stop. The computer will return control to the module currently in memory. To load a module press the appropriate letter. The specified module will be highlighted and you will be given the option to 'LOAD MODULE (Y/N)?'.

N.B. All (Y/N)? prompts will accept 'Y' or 'RETURN' as yes, any other key will be accepted as no.

N.B. All input prompts can be quit by pressing the STOP key.

When you select a module the menu will first check the computer status to see if the module can or should be loaded.

1. If the computer is in picture mode and you try to load the character editor. Will still let you load it. Message 'WRONG GRAPHICS MODE'.

2. If the computer is in character set mode and you try to load the picture editor. Will still let you load it. Message 'WRONG GRAPHICS MODE'.

The screen will blank whilst the module is being loaded and then the menu will be re-displayed, but the loaded module will remain highlighted. If you have loaded the wrong module you can load another module. If you select the loaded module again, you will quit the menu and enter that module.

Do not press any keys or reset whilst modules are being loaded as this may cause the system to crash.

In load-tests, the time for the menu was approximately 4 seconds. The time for the modules may differ but was never more than 15 seconds. If your setup includes a parallel operating system, these times will be reduced quite considerably.

Universal Commands

All modules support the following:

R Restart — The frozen program should resume execution.

↑ Viewer — This command as already mentioned allows you to view the screen. Some modules have extra commands in this mode, but all have the following:

← — Current sprites off/on (only for viewing purposes).

1 — Increment border colour.

2 — Increment screen colour.

STOP — Quit back to module.

N New — Memory is filled with a single byte, the computer then resets.

P Protect — Memory is configured, the computer then resets.

Both the New and Protect commands allow you to specify the fill-byte.

← Reset — The computer is reset back to Basic but memory is not altered. If you hold the shift key down you will do a CBM80 reset - if there is a program in memory which has set the CBM80 vector at $8000 (32768) the computer will jump to this address instead of Basic.

X Exit — This command allows you to exit to Basic.

@ DOS — This opens a channel to the disk drive and allows you to send a command, get the status, get the disk directory and check start/end addresses of a file.

(RETURN)	Get status.
I	Initialise disk in drive and return status.
$	Display directory.
#file	Show start/end addresses of file.

Both the **$** and **#** options can be quit by pressing STOP.

D	DirectoryDisplay directory of disk in drive 0. Spacebar to pause listing, STOP to quit.

M Menu — The bottom screen line displays the message:

RESET DRIVE,INSERT MENU DISK,PRESS SPACE

Insert the menu disk and hold down the Spacebar to load the menu. STOP to quit.

The machine code monitor does not have the Reset (←) or Directory (D) commands. The directory can be displayed either by '@$ (RETURN)' or by pressing 'F7'. Type 'MM (RETURN)' for the menuloader.

Backup

Contained in this module are:

- Superfast backup routine.
- Unique all-memory load/save.
- Disk/tape save of backup.
- Disk turbo.
- Standard screen-viewer.

Key	Command description
B	Backup memory
L	Load all-memory
S	Save all-memory
S	Save backup
T	Tape backup
F1	Disk turbo on/off
↑	Screen-viewer
R	Restart program
N	New memory
P	Protect memory
←	Reset
X	Exit module
@	DOS commands
D	Directory
M	Menuloader

B Backup

This command allows you to compress the whole of memory into a single file which can then be saved. Once compressed and saved, the file can run independantly of the Expert Cartridge.

This is a brand new backup routine which takes less than 15 seconds and is more efficient than that used on the V4.1.

Prior to backing-up you have to set the Q-value. This will be displayed and can be altered. once you have done this the routine will start compacting memory.

After about two seconds the screen will change colour and remain blank until the backup has been completed. If there is insufficient space to make a backup the message "MEMORY OVERFLOW!" will appear.

We have tested this new backup routine on a variety of games and have never had it fail. The maximum time taken to make a backup was twelve seconds!

Once you have made a backup you will not be allowed to use the backup or screen viewer commands again until you use one of the following:

Restart, Reset, Exit, New, Protect

On completion of the backup routine, the restart address will be set to the SYS address of the compacted code.

Whole Memory

The following two unique commands have been incorporated to allow you to make changes to memory without worrying about making a mistake. The whole of memory including colour/IO can be saved to disk in a single file (261 blocks) which can be loaded back if you make a mistake. This is a special file format and the filename is preceded by 'MEM.'.

L Load All-Memory

This lets you reload a previously saved 'MEM.' file. Because of the way in which the loader works 'MEM.' files saved with the V4.1 cannot be reloaded by V4.2.

S Save All-Memory

This allows you to save a 'MEM.' file.

S Save Backup

If you have used the backup command then this option will replace the one above. You can save the backup as many times as you wish.

T Tape Backup

If you have used the backup command then this command becomes available. This is a TurboTape routine and requires that your tapedeck is properly alligned.

All backups can run independantly of the Expert Cartridge.

F1 Disk Turbo On/Off

The disk turbo works at about five times normal speed. If your setup includes a parallel operating system then you might as well switch the disk turbo off.

These routines have been tested and work with the following drives: 1541, 1541C, Blue Chip & Oceanic OC-118N. If you are having problems loading and saving try switching the disk turbo off.

Monitors

These are full-feature machine code monitors with a few extras:

Whole memory monitor including RAM, ROM, Colour and I/O - load/save anywhere!

Pre-programmed function keys including:

Repeats, directory, load, etc.

Dis-assemble, monitor and text interpret listings.

Bi-directional scrolling in all three listing modes.

Output to either screen or printer. Full unlisted opcodes implementation. Full access to disk drive memory.

When you enter the monitor the screen will clear and the registers will be displayed:

```
 ADDR SR AC XR YR SP IO NV-BDIZC RAM
/E5CD 27 00 00 00 FF 37 00100111 ASC
```

The following values can be altered:

ADDR	The address at which the program was frozen
SR	Status register
AC	Accumulator
XR	X-register
YR	Y-register
SP	Stack pointer
IO	Value in location $0001

The status register is also displayed as seperate flags:

N	Negative
V	Overflow
-	Unused
B	Break
D	Decimal
I	Interrupt
Z	Zero
C	Carry

The status register is displayed here as a binary number. This allows you to see each flag seperately. 0 indicates that the flag is clear and 1 that the flag is set. These flags cannot be altered independantly.

Also displayed with the registers are the following four status flags:

The RAM/ROM status	'RAM' or 'ROM'
Unlisted opcodes on/off	' ' or 'U/L'
ASCII/Screen-pokes mode	'ASC' or 'SCR'
Repeats on/off	' ' or 'REP'

Important

There are a couple of slight differences between the two monitors:

The machine monitor is exactly as above.

The drive monitor as well as the RAM/ROM settings also has R-D/D-D/D-R :

R-D from RAM to drive
D-D from drive to drive
D-R from drive to RAM

These settings allow you to use all the standard commands with drive memory such as transferring from drive to RAM.

The 'K' (Re-locate) command will only work with 'RAM' or 'D-D' to stop you from making mistakes.

You cannot load/save drive memory.

The 'Y' (printer output) command is not implemented.

Commands List

Description	Command syntax
Dec to hex	# 64738
Hex to dec	$ FCE2
'AND'	& 1000 1FFF 7F
Interpret input	' 1000 textstring
RAM/ROM	*
Increment	+ 1000 1FFF 03
Dis-Assemble input	, 1000 4C E2 FC
Decrement	- 2000 2FFF 64
Registers input	/ E5CD 27 00 00 00 FF 37
Monitor input	: 3000 54 52 49 4C
Viewer	↑
Registers	←
DOS	@, @I, @#filename, @$
Assemble	A 1000 LDA #$00
Compare	C 1000 1FFF 2000
Dis-Assemble	D 1000
'EOR'	E 1000 1FFF 73
Fill	F 4000 4FFF 00
Goto	G FCE2
Hunt	H 6000 7FFF 48 49
Interpret	I 1000 1FFF
JSR	J FD15
Locate	K 1000 1FFF 0400 047F
Load	L "filename",08,2000
Monitor	M 1000
New	NN 7F
'OR'	O 1000 1FFF 80
Protect	PP 63
Q-value	Q 08
Restart	R

Save	S "filename",08,1000,2000
Transfer	T 1000 1FFF 3000
Colours	U 678
Verify	V "filename",08,C000
Range	W 1000 2FFF
Exit	X
Output	Y

F1	RAM/ROM toggle
F2	Unlisted opcodes on/off
F3	Registers
F4	ASCII/screen-pokes mode
F5	Load
F6	Repeats on/off
F7	Directory
F8	Cursor bottom screen
CRSR-UP	Scroll listing up
CRSR-DOWN	Scroll listing down

Refer to the Commodore 64 Programmer's Reference Guide for a full description of the AND, OR and EOR commands.

Commands Description

Decimal to hex conversion

Syntax #64738
Converts the decimal number 64738 into it's hex equivalent $FCE2

$ Hex to decimal conversion

Syntax $1000
Converts the hex number $1000 into it's decimal equivalent 4096

& Memory 'AND' XX to memory

Syntax & 1000 1FFF 26
Every location in the specified memory range will be anded with 26.

' Text interpret mode

Syntax '1000 textstring
The textstring will be inserted at the specified address. This command works in both output modes - ASCII or screen-pokes. The values will vary depending on which mode you are in, so be careful when using this command. Also, be careful when generating new lines as this command will even see a blank line as a line of text.

***** RAM/ROM toggle

Syntax *
This dictates whether you are looking at RAM or ROM. If you wish to look at the colours or I/O then you should select ROM. A complete description of the memory mapping of the Expert can be found in the General Information section.

+ Memory plus XX to memory

Syntax + 1000 1FFF 37
Every location in the memory range will have 37 added to it.

, Dis-assemble input

Syntax ,1000 20 00 FE JSR $FE00
When you dis-assemble a section of memory it will be displayed like the example above. If you wish to alter a line you can do this without using the Assemble command. Instead you can either enter the new bytes over the existing ones or you can enter a new instruction over the old one. In either case, you must enter it at it's present screen location.

- Memory minus XX to memory

Syntax - 1000 1FFF 04
Every location in the memory range will have 04 subtracted from it.

/ Change current register values

Syntax /1000 27 00 00 00 FF 37
When you display the registers (←) you can alter the current values by cursoring over them and entering new values - be careful! If you only wish to alter the restart address you could type /4000 and this would change the restart to $4000.

: Monitor input

Syntax :1000 4E 4F 57 20 31 39 38 39
When you display a section of memory with the monitor command it looks like the example above. To alter these values cursor over the bytes and enter the new values. You cannot enter text in this mode.

↑ Screen-viewer

If you wish to view the screen as it was when you froze the program press '↑' and 'RETURN'.

The screen will be displayed with the sprites - if there were any. In this mode you can:

1	Increment border colour
2	Increment screen colour
←	Switch current sprites off/on
STOP	Quit back to monitor

← Display registers

Pressing ← will display the registers as mentioned above. See / command on how to change registers.

@ DOS commands

This allows you to check the status of the disk drive, send commands to the diskdrive, get the disk directory and get start/end address of a file.

@	Gets status
@**command**	Performs command and returns status (I,S,N,V,UI etc).
@**$**	Displays directory of disk currently in drive
@**#file**	Displays start/end address of file. If it is a long file and you just require the start address press STOP

All DOS commands will re-direct output to the screen.

A Assemble machine code instructions

Syntax A 1000 LDA #$00
To enter/alter instructions use the 'A' key followed by the address followed by the command. Once you have entered one line, the code will be assembled to memory and the next address is automatically generated.

C Compare memory with memory

Syntax C 1000 1FFF 2000
This will compare memory specified by the first two numbers with memory starting at the third number. If there are any differences, those locations will be displayed.

D Dis-assemble machine code

Syntax D 1000
This will start dis-assembling memory from the specified address.

E Memory 'EOR' XX to memory

Syntax E 1000 1FFF 43
Every loaction in the memory range will be EORed with 43.

F Fill memory with XX

Syntax F 1000 1FFF 80
Every location in the memory range will be filled with 80.

G Goto address

Syntax G 1000
This will cause the computer to start executing code at the specified address using the current register values.

H Hunt memory for values

Syntax H 1000 1FFF 4E 4F 57
Hunt through memory range for byte(s) - as many as 16 bytes.

Syntax H 1000 1FFF 'NOW 1989
Hunt through memory range for ASCII textstring.

Syntax H 1000 1FFF "NOW 1989"
Hunt through memory range for screen- pokes textstring.

Every time a match is found, the address at which the match begins will be shown.

I Interpret memory - text display

Syntax I 1000
Display the specified memory purely as characters. This command works both in ASCII and screen-pokes mode.

J Jump subroutine

Syntax J 1000
This will cause the computer to execute the routine at the specified address but will return control to the monitor on an 'RTS' instruction. The registers will reflect what has occurred during the routine, but the restart address should be the same.

K (Re-) Locate memory references

Syntax K 1000 1FFF 0300 03FF
The specified memory range will be searched for any references to the memory range 0300 to 03FF. Any references will be displayed.

Syntax K 1000 1FFF 0300 03FF 0400
The specified memory range will be searched for any references to the memory range 0300 to 03FF and will alter them to 0400 upwards.

L Load file from disk

Syntax L"FILENAME",08
The specified file will be loaded from disk into memory. The start and end addresses will be displayed. If you wish to load a file into a section of memory that it was not originally saved from use:

Syntax L"FILENAME",08,1000
The address at which it will load.

M Monitor memory + menuloader

Syntax M 1000
Memory starting at the specified address will be displayed:

```
:1000 4E 4F 57 20 31 39 38 39'NOW 89
```

If you wish to alter these values refer to the ':' command.

Syntax MM
This will activate the menuloader. The screen will clear and prompt you to insert the menu-disk and press a key. If you do not wish to load the menu press the STOP key.

N New memory and reset

Syntax NN 0F
Computer memory will be filled with the specified value and then the computer will reset. The number is an option. once you have specified a value it will stay constant. As a safety precaution, you have to press the command key twice.

Syntax N
Returns current new value.

O Memory 'OR' XX to memory

Syntax O 1000 1FFF FD
Every location in the memory range will be ORed with FD.

P Protect memory and reset

Syntax PP F8
Computer memory will be configured using the specified value and then the computer will reset. The number is an option. once you have specified a value it will stay constant. As a safety precaution, you have to press the command key twice.

Syntax P
Returns current protect value.

Q Q-value set

The Q-value specifies the section of memory which will be used as a work area when it comes to compressing memory - see backup module. If you wish to enter a new value use the following

Syntax Q 27

Entering Q without a value will activate the auto-address finder.

R Restart program

Syntax R
This returns control of the computer to the program in memory. If you have altered any registers including the restart address the program will continue with those new settings.

S Save file to disk

Syntax S"FILENAME",08,1000,2000
The specified section of memory will be saved to disk under the specified filename. If you wish to save a file so that it will load in at a different address from which it was saved use:

Syntax S"FILENAME",08,1000,2000,3000
reload address.

T Transfer memory

Syntax T 1000 1FFF 2000
Transfer the memory specified by the first two numbers to memory starting at the third number. This is an intelligent routine and can transfer within itself, ie T 1000 1FFF 1100

U Set user colours

Syntax U789
The border, screen and text colours may be altered with this command. The first number is the border, the second the screen and the third the text. If any command alters any of these colours, they will automatically be set back to these values. The default values are 661 (border blue, screen blue and text white).

V Verify file with memory

Syntax V"FILENAME",08
This command opens the specified disk file, gets the start address and then compares the file with the section of memory starting at the same address. All differences are displayed. If you wish to verify a file against memory not at the same address use:

Syntax V"FILENAME",08,3000

W Auto range set

Syntax W 1000 1FFF
If you are continually working with the same section of memory, it would be easier if you could automatically set the memory range - hence this command. Set the range as above - the default values are 0000 TO FFFF.

This option can be used with: & + - C E F H K O T

Syntax C* 3000 instead of C 1000 1FFF 3000

Syntax H* 'TEXT instead of H 1000 1FFF 'TEXT

Syntax W
Returns current auto-range.

X Exit monitor

Syntax X
This command exits the monitor by performaing a warm start.

Y Screen/printer output

All listings (dis-assemble, monitor and interpret) default to the screen but can be directed to the printer.

Syntax Y
This will toggle between screen and printer output. Any accessing of the disk drive will re-direct output to the screen. The printer cannot display the screen-pokes properly, so when you are in printer mode the output mode switches to ASCII and cannot be changed whilst the printer is active.

Syntax Ytextstring
Sends textstring to printer.

With the Dis-assemble (D), Interpret (I) and Monitor (M) commands, listings can be displayed in the following ways:

Syntax D 1000 list from 1000

Syntax I 1000. list only 1000

Syntax M 1000 1FFF list 1000 to 1FFF

The above three listings can be halted by pressing the STOP key or paused by pressing the spacebar.

Interrupt-Driven Commands

F1 RAM/ROM toggle

This has the same effect as the '*' command but aslo displays the new registers.

F2 Unlisted opcodes on/off

This switches the unlisted opcodes on or off and displays registers. With the unlisted opcodes switched on, you can assemble/dis-assemble using unlisted opcodes. You cannot use unlisted opcodes if this is set to off.

F3 Display registers

This has the same effect as using the '←' command - registers.

F4 ASCII/screen-pokes toggle

This changes all screen text input/ output between ASCII and screen-pokes and displays the registers.

F5 Load

To make loading a file easier you can use this function. List the directory, cursor to the file to load and press 'F5'. This will place 'L' at the start of the line, ',08' after the filename and clear the rest of the line.

```
L "FILENAME",08
```

F6 Repeats on/off toggle

The repeats are switched on or off and the registers are displayed.

F7 Directory

List disk directory.

F8 Cursor bottom-left of screen

This key is used as the opposite of the 'HOME' key and makes it easier when you want to use the scroll routines - see below.

CRSR UP

If the cursor is at the top of the screen and you press this key then the current screen listing will be scrolled up.

CRSR DOWN

If the cursor is at the bottom of the screen and you press this key then the current screen listing will be scrolled down.

Unlisted Opcodes

The following notation applies to this section

A	Accumulator
X	X-register
Y	Y-register
SP	Stack pointer
(HB+1)	(High byte of operand)+1 (HB+1) of FCE2 = FC+1 = FD
M	Location or value specified
#	Hexadecimal value
AND	Logical AND two values
OR	Logical OR two values
EOR	Logical EOR two values

AAX	A AND X, to M
ALR	A AND M, LSR
AMB	A AND M, to A : bit 7 to carry
ASO	ASL M, M OR A, to A
AXH	A AND X AND (HB+1), to M (A, X not affected)
AXS	A AND X, to SP, AXH (A, X not affected)
CRA	Crash processor
DCM	DEC M, CMP with A
HXM	X AND (HB+1), to M (X not affected)
HYM	Y AND (HB+1), to M (Y not affected)
INS	INC M, SBC from A
LAS	LDA, LDX, TXS
LAX	LDA, LDX
LSE	LSR, EOR with A, to A
NPO	NOP replacement
OAX	(A OR #EE) AND M, TAX
RLA	ROL M, M AND A, to A
RRA	ROR M, M ADC A, to A
SAX	A AND X, SBC M, to A
SKB	Skip byte
SKW	Skip word
SUB	SBC replacement
XMA	(A OR #EE) AND M AND X, to A (X not affected)

Addressing Modes

A	Implied	ASL, DEY, BRK
B	Immediate	LDA #$5F
C	(Indirect)	JMP ($1000)
D	(Indirect),Y	LDA ($AC),Y
E	(Indirect,X)	LDA ($AE,X)
F	Zeropage	BIT $FB
G	Absolute	STY $0FE3
H	Zeropage,Y	LDX $10,Y
I	Absolute,Y	LDA $0314,Y
J	Zeropage,X	STA $2B,X
K	Absolute,X	LDA $8000,X
L	Relative	BNE $1FCD

The above mnemonics may not be the same as those that you already know but were fully 'roadtested' and found to be satisfactory.

The following table lists all opcodes as implemented in the monitors. The addressing mode is defined by the letter underneath the mnemonic (A-L). The '*' signifies that the opcode is an unlisted one.

```
     0   1   2   3   4   5   6   7   8   9   A   B   C   D   E   F
-------------------------------------------------------------------
00  BRK ORA CRA ASO SKB ORA ASL ASO PHP ORA ASL AMB SKW ORA ASL ASO
    A   E   A * E * F * F   F   F * A   B   A   B * G * G   G   G *
-------------------------------------------------------------------
10  BPL ORA CRA ASL SKB ORA ASL ASO CLC ORA NPO ASO SKW ORA ASL ASO
    L   D   A * D * F * J   J   J * A   I   A * I * G * K   K   K *
-------------------------------------------------------------------
20  JSR AND CRA RLA BIT AND ROL RLA PLP AND ROL AMB BIT AND ROL RLA
    G   E   A * E * F   F   F   F * A   B   A   B * G   G   G   G *
-------------------------------------------------------------------
30  BMI AND CRA RLA SKB AND ROL RLA SEC AND NPO RLA SKW AND ROL RLA
    L   D   A * D * F * J   J   J * A   I   A * I * G * K   K   K *
-------------------------------------------------------------------
40  RTI EOR CRA LSE SKB EOR LSR LSE PHA EOR LSR ALR JMP EOR LSR LSE
    A   E   A * E * F * F   F   F * A   B   A   B * G   G   G   G *
-------------------------------------------------------------------
50  BVC EOR CRA LSE SKB EOR LSR LSE CLI EOR NPO LSE SKW EOR LSR LSE
    L   D   A * D * F * J   J   J * A   I   A * I * G * K   K   K *
-------------------------------------------------------------------
60  RTS ADC CRA RRA SKB ADC ROR RRA PLA ADC ROR ALR JMP ADC ROR RRA
    A   E   A * E * F * F   F   F * A   B   A   B * C   G   G   G *
-------------------------------------------------------------------
70  BVS ADC CRA RRA SKB ADC ROR RRA SEI ADC NPO RRA SKW ADC ROR RRA
    L   D   A * D * F * J   J   J * A   I   A * I * G * K   K   K *
-------------------------------------------------------------------
80  SKB STA SKB AAX STY STA STX AAX DEY SKB TXA XMA STY STA STX AAX
    F * E   F * E * F   F   F   F * A   F * A   B * G   G   G   G *
-------------------------------------------------------------------
90  BCC STA CRA AXH STY STA STX AAX TYA STA TXS AXS HYM STA HXM AXH
    L   D   A * D * J   J   H   H * A   I   A   I * K * K   I * I *
-------------------------------------------------------------------
A0  LDY LDA LDX LAX LDY LDA LDX LAX TAY LDA TAX OAX LDY LDA LDX LAX
    B   E   B   E * F   F   F   F * A   B   A   B * G   G   G   G *
-------------------------------------------------------------------
B0  BCS LDA CRA LAX LDY LDA LDX LAX CLV LDA TSX LAS LDY LDA LDX LAX
    L   D   A * D * J   J   H   H * A   I   A   I * K   K   I   I *
-------------------------------------------------------------------
C0  CPY CMP SKB DCM CPY CMP DEC DCM INY CMP DEX SAX CPY CMP DEC DCM
    B   E   F * E * F   F   F   F * A   B   A   B * G   G   G   G *
-------------------------------------------------------------------
D0  BNE CMP CRA DCM SKB CMP DEC DCM CLD CMP NPO DCM SKW CMP DEC DCM
    L   D   A * D * F * J   J   J * A   I   A * I * G * K   K   K *
-------------------------------------------------------------------
E0  CPX SBC SKB INS CPX SBC INC INS INX SBC NOP SUB CPX SBC INC INS
    B   E   F * E * F   F   F   F * A   B   A   B * G   G   G   G *
-------------------------------------------------------------------
F0  BEQ SBC CRA INS SKB SBC INC INS SED SBC NPO INS SKW SBC INC INS
    L   D   A * D * F * J   J   J * A   I   A * I * G * K   K   K *
-------------------------------------------------------------------
     0   1   2   3   4   5   6   7   8   9   A   B   C   D   E   F
```

Refer to the Commodore 64 Programmer's Reference Guide for a full description of the standard opcodes and addressing modes.

Picture Converter

This allows you to convert a standard resolution screen into a picture - try it with the standard commodore screen.

We have supplied one 'we made earlier' on the disk (B.CONVERTED!) just to show you the possibilities of this. Obviously it won't always work as well!

Both the picture editor and the picture formatter have this function.

Extended Screen-Viewer Commands.

←	Sprites on/off
1	Increment border colour
2	Increment screen colour
F1	Hires/multicolour mode
F3	Video bank up
F5	Pixels hi/lo
F7	Colours up
STOP	Quit back to module

When you enter the module it checks that the computer is set up for a picture. If it is then the converter is by-passed.

When you enter the converter section the following message is displayed.

NOT PICTURE SCREEN

Commands List

Key	Command description
R	Restart program
↑	Screen-viewer
N	New memory
P	Protect memory
←	Reset
X	Exit module
B	Background colour
T	Text colour
E	Enter module
C	Convert standard screen
M	Menuloader

E Enter module

This command allows you to enter the module without using the converter. The VIC chip will be set up so that the computer is displaying a picture.

C Convert standard screen

This unique command converts a standard resolution screen into a picture. It can convert from the ROM character sets or from a RAM based set.

In conversion tests that we carried out the maximum time taken for any type of conversion was 8 seconds when using a ROM based character set.

Both the Enter (E) and Convert (C) commands change the restart address so that if you do attempt a restart after having used them the computer will reset because you will have corrupted memory!

Picture Formatter

This module allows you to load, save and convert between all the major formats. It also incorporates a disk turbo and a 2-way picture flipper.

Key	Command
R	Restart program
↑	Screen-viewer
N	New memory
P	Protect memory
←	Reset
X	Exit module
@	DOS commands
D	Directory
L	Load picture
S	Save picture
F1	Disk turbo on/off
H	Horizontal flip
V	Vertical flip
HOME	Cursor top-line
CLR	Cursor bottom-line
CRSRS	Cursor up/down one selection
M	Menuloader

This module supports 11 major picture formats, both hires and multicolour.

The current displayd mode will be highlighted. If you use the screen-viewer the display mode can be altered.

Cursor up/down to select the format you want to load/save. Press 'L' to load or 'S' to save.

When you load a picture the display mode will automatically be set as required. Pictures can be loaded anywhere and will not use any memory other than that used by the picture. This means that if you have stopped a program, save a picture off and edited it you can then load it back again in the same place and continue the program (see 'Mem-Save command - Backup module).

If you are experiencing problems loading or saving try switching off the turbo.

2-Way Picture Flipper

Pictures can be flipped horizontally or vertically without corrupting memory.

H - Horizontal picture flip

V - Vertical picture flip

Make sure that the display mode is right before using the horizontal flip. Because of the bit patterns required for a multicolour picture the routine works differently than when in hires.

Try it in the wrong mode - it's fun.

Picture Editor - Main Mode

Screen Layout

The whole of the screen (except for the bottom line) is used as a window on the picture. This displays an 8*8 representation of a 5*3 character square window on the picture.

The picture editor seperates each 8*8 matrix of the picture into a character square to make drawing easier. Each square has it's own colours - when you enter edit mode you will notice that as you move the cursor across squares the colours may change.

Each square has 2 colours in hires mode and 4 colours in multicolour mode. Hires uses colours 2 and 3, Multicolour uses all 4 colours, but colour 1 is the same in every square because it is the screen colour.

Now that I've totally baffled both you and myself I shall continue!

In main mode the bottom line displays the hires/multicolour mode, the video bank number, the pixels location (the actual picture) and the colours location (the colour screen location).

Main Mode Commands List

Key	Command description
R	Restart program
↑	Screen-viewer
N	New memory
P	Protect memory
←	Reset
X	Exit module
B	Background colour
T	Text colour
@	DOS commands
D	Directory
F1	Hires/multi
E	Edit mode - see edit section
F3	Video bank up
F4	Video bank down
F5	Pixels lo/hi
F7	Colours up
F8	Colours down
HOME	Matrix top-left
CLR	Matrix bottom-right
CRSRs	Scroll matrix
C	Clear picture
M	Menuloader
L	Load picture
S	Save picture

This module uses the Vidcom format both for hires and multicolour pictures. If you wish to use different formats use the picture formatter module to convert your pictures to Vidcom format.

Memory Addresses

5800-5BFF	Background colours (D800)
5C00-5FFF	Colour screen
6000-7F3F	Pixels

We have used 5BFF as the location which holds the screen colour, for some reason best know to them, the programmers of Vidcom omitted this!

F5 Pixels Lo/Hi

A picture can sit in 2 locations in each video bank. This toggles between the 2. the pixels address will change and the matrix will display the new 'picture'.

F6 Repeats On/Off

This command just switches the repeats mode on or off.

F7 Colours Up

The colours are moved upwards through the current video bank - there are 16 (decimal) possible locations. The colour address will alter accordingly and the matrix will display the new colours.

F8 Colours Down

This is the opposite of 'F7' and moves the colours downwards.

C Clear Picture

This command will clear the pixels area of the picture - be careful!

L Load Picture

The picture will be loaded into memory. The matrix will display the new window.

S Save Picture

The picture will be saved and can then be edited using Vidcom if you wish.

Both the load and save commands support both hires and multicolour pictures. This allows you to save a picture, edit it with your own program and then reload it into a game - a unique feature.

Both the load and save automatically insert the prefix for you. The load routine only loads into memory which is a part of the picture and will not corrupt any other memory.

Picture Editor - Edit Mode

When you enter edit mode the bottom line will change to display the X/Y positions of the cursor on the picture and the current square's colours.

The X and Y positions are made up of character squares + pixels. i.e. X-03+04 = 3 characters across, cursor at 4th pixel in square.

Edit Mode Commands List

Key	Command description
B	Background colour
T	Text colour
HOME	Cursor top-left
CLR	Cursor bottom-right
CRSRs	Move cursor
SPACE	Draw - set/clear pixels
1-4	Select draw colour
SHIFT 1-4	Set colours
L	Load matrix
S	Save matrix
↑	Viewer
V	Velocity
F1	Hires/multi
F	Fix matrix
F3	Matrix to store
F4	Store to matrix
F2	Swap matrix with store
C	Clear square
SHIFT C	Clear matrix
E	Exit with edits
STOP	Exit without edits

The joystick can be used to move the cursor in all 8 directions. The fire button will set/clear pixels like the spacebar.

As with the character editor, when you edit the matrix you are not editing the actual picture but a workspace in the picture editor. This allows you to make a mistake and rectify it. If you move the cursor off the edge of the matrix causing the matrix to move, any current matrix edits will be transferred to the picture.

The picture editor allows you to 'grab' a small section of a picture and save it to disk. This can then be loaded back afterwards. The file is prefixed with 'MAT.'.

G Goto X/Y Co-ordinates

This allows you to go straight to any picture matrix position by inputting the X/Y co-ordinates. (X 0-23 , Y 0-14).

L Load Matrix

Load a previoulsy save matrix file.

S Save Matrix

Save the current matrix display.

↑ Screen Viewer

This command allows you to go straight to the screen-viewer from edit mode. Any current edits on the matrix will be put into the picture. When you quit the viewer you will return to edit mode.

V Velocity

This allows you to set the velocity at which the cursor moves. The value can be from $00 to $FF. The lower the value the faster the cursor will move.

F Fix Matrix

If you make a mistake while editing, you can fix the matrix from the picture.

F3 Matrix To Store

The picture editor can store a copy of the matrix so that you can move things around the picture or even from picture to picture - this includes the colours. This command puts the current matrix in the store.

F4 Store To Matrix

This is the inverse of 'F3' and will get the store and place it on the matrix.

F2 Swap Matrix With Store

This allows you to swap the matrix with the store-matrix.

C Clear Square

This clears the current square.

SHIFT C Clear Matrix

This clears the matrix.

E Exit With Edits

This will put any edits on the current matrix into computer memory and then return you to the main mode.

STOP Exit Without Edits

This is the inverse of the above command and returns you to the main mode without putting any current edits into computer memory.

Character Editor

The character editor has the following:

- Load and save whole or part character sets.
- Define/edit your own characters or alter existing ones.
- Edit upto 32 character sets at once!

Extended screen-viewer commands:

←	Sprites on/off
1	Increment border colour.
2	Increment screen colour.
F1	Hires/multicolour mode.
F3	Video bank up.
F7	Character set up.
STOP	Quit back to module

When you enter the character editor, it checks whether the current character set is a RAM or ROM based set. If it is RAM based then there is no problem and you will enter the main editor mode. The character editor can obviously not edit a ROM based character set. Because of this, you have the option of transferring the current ROM based set into RAM before entering the editor. When you enter the converter section the following message is displayed:

```
REAL-TIME CHARACTER EDITOR
   ROM CHARACTER SET
```

Commands List

Key	Command description
R	Restart program
↑	Screen-viewer
N	New memory
P	Protect memory
←	Reset
X	Exit module
B	Background colour
T	Text colour
E	Enter editor
C	Convert ROM set to RAM set
M	Menuloader

E Enter Editor

This allows you to enter the main editor mode without using the convert command. If you use the screen-viewer you will have to go through this procedure again unless you change the character set location.

ROM character set image locations - RAM sets not possible in these locations.

BANK0 1000-1FFF and BANK2 9000-9FFF

C Convert

This command will take the ROM character set and put it in RAM at $4000. It will then set the VIC chip so that it is now looking at the new set. Because this command has corrupted memory the restart address is altered so that the computer resets if you use the restart command.

Character Editor - Main Mode

Screen Layout

The bottom 16 lines of the screen contain the main character matrix. This displays an 8*8 representation of the characters displayed at the top-left of the screen. The matrix is laid out so that the top line displays 5 characters starting at char number (top of screen) and below are 5 more characters starting at char+ $20. The character set is split into 8 rows of $20 characters.

You can move freely around the character set using the cursor keys.

The upper-left of the screen displays in formation regarding the location of the characters in memory, hires/multicolour mode and the draw colours.

The upper-right has two small single-character matrices and between them are the two characters they are displaying. The left one is the character you are currently on in the large matrix. The right one is the current store - there are $20 store characters.

Main Mode Commands List

Key	Command description
R	Restart program
↑	Screen-viewer
N	New memory
P	Protect memory
←	Reset
X	Exit module
B	Background colour
T	Text colour
@	DOS commands
D	Directory
F1	Hires/multi
E	Edit mode - see edit section
F3	Video bank up
F4	Video bank down
F7	Character set up
F8	Character set down
HOME	Matrix top-left
CLR	Matrix bottom-right
CRSRs	Scroll matrix
M	Menuloader
L	Load character set
S	Save character set
SHIFT L	Load characters
SHIFT S	Save characters

F7 Character Set Up

This allows you to look at different character sets which may be in memory and can be used in conjuction with 'F3'. It will step upward through the possible 8 character set locations in the current video bank. It is not advisable to use during a game in case you forget where you started.

F8 Character Set Down

This is the opposite of 'F7' and steps down through the current video bank.

Load And Save Routines

All load and save routines will prompt you for a filename. There is no prefix required so you can load or save for any character editor.

The character load/save routines ask you to specify the start and end characters. The start character is automatically set to the character you are currently on. You can of course change this. The end value will then be displayed and this will be the same as the start value - this will save 1 character. Again delete and change if you require. This allows you to load/save anything from 1 to 256 characters.

Throughout the load/save routines STOP will quit.

L Load Character Set

This loads a character set over the one you are currently working on. Irregardless of your current position in the character set it will always load it starting with character 00.

S Save Character Set

This saves the character set. The save routine automatically specifies a load address of $2000.

SHIFT L Load Characters

Allows you to load as many characters as you wish.

SHIFT S Save Characters

Allows you to save as many characters as you wish.

Character Editor - Edit Mode

Key	Command description
B	Background colour
T	Text colour
HOME	Cursor top-left
CLR	Cursor bottom-right
CRSRs	Move cursor
SPACE	Draw - set/clear pixels
1-4	Select draw colour
SHIFT 1-4	Set colours
V	Vertical flip
H	Horizontal flip
+	Roll character up
-	Roll character down
SHIFT +	Roll character left
SHIFT -	Roll character right
R	Rotate character right
SHIFT R	Rotate character left
CTRL R	Reverse character
C	Clear character
F	Fix matrix character
SHIFT F	Fix RAM character
F1	Hires/multi
G	Goto character
*****	Store number up
SHIFT *	Store number down
F7	Character to store
F8	Store to character
F5	Swap character with store
F2	Byte input
E	Exit with edits
STOP	Exit without edits

The joystick can be used to move the cursor in all 8 directions. The fire button will set/clear pixels like the spacebar

When you edit a character on the matrix you are not altering computer memory but a workspace in the character editor. This allows you to make mistakes without losing the original character. If, when using any of the cursor move commands, you cause the matrix to move, any edits which are currently stored in the workspace will be transferred into computer memory.

There are four possible draw colours:

		Bit patterns
1 = Screen	$D020	00
2 = Drawcolour	$D800	11
3 = Multi 1	$D022	01
4 = Multi 2	$D023	10

Bit patterns only apply if you are using multicolour.

Colour 2 is the colour specified in the colour RAM from $D800 to $DBFF. Colours 1,3 and 4 are set automatically when you enter the editor. All colours can be re-set in the editor.

V Vertical Flip

The current character is flipped top to bottom.

H Horizontal Flip

The current character is flipped side to side.

+ Roll Character Up

The current character is rolled up one pixel-row and the top row appears at the bottom.

- Roll Character Down

The current character is rolled down one pixel-rown and the bottom row appears at the top.

SHIFT + Roll Character Left

The current character is roll one pixel-column left and the leftmost row appears on the right.

SHIFT - Roll Character Right

The current character is rolled one pixel-column right and the rightmost column appears on the left.

R Rotate Character Right

The current character is rotated through 90 degrees right.

SHIFT R Rotate Character Left

The current character is rotated through 90 degrees left - inverse of above.

CTRL R Reverse Character

The current character is reversed. (You can also use ctrl-9 which is Rvrs On)

C Clear Character

The current character is cleared, filled with 00.

F Fix Matrix Character

This command fixes the character on the matrix from it's partner in RAM. If you make a mistake you can use this to restore the character.

SHIFT F Fix RAM Character

This command fixes the character in RAM from it's partner on the matrix. If you have defined a character which is ok you can use this to store permanently in RAM.

Both fix commands check the matrix and RAM character. If there is no difference the fix command is ignored.

F1 Hires/Multi

This will alter the display mode between hires and multicolour. All the matrices will change accordingly. In hires, only the first 2 colours are used. In multicolour all 4 colours are used. You can only set colours 3 and 4 when you are in multicolour mode.

G Goto Character

This allows you to move directly to a character. Input the character number and press RETURN.

* Store Number Up

The character editor has $20/32 decimal seperate character stores. You can page upwards through them using this key.

SHIFT * Store Number Down

This is the inverse of the above and pages downwards through the stores.

F7 Character To Store

The current character is put into the current store.

F8 Store To Character

The current store is put into the current character.

F5 Swap Character With Store

The current character and current store are swapped.

F2 Byte Input

The single-character matrix in the top-middle of the screen is a duplicate of the current character in the main matrix to the left of this matrix is the value for each byte making up the character. This command allows you to edit directly using bytes. The bottom screen-line will display

CHARACTER ?? BYTE 0 ??

The question marks are variables. The byte value will correspond to the values at the top of the screen and each byte will be highlighted. As you input the new values, the matrix will change to display the new character. Press STOP to quit at any time.

E Exit With Edits

This will put any edits on the current matrix into computer memory and then return you to the main editor mode.

STOP Exit Without Edits

This is the inverse of the above command and returns you to the main editor mode without putting any current edits into computer memory.

Sprite Editor

The sprite editor allows you to load, save, define and edit sprites anywhere in memory - an amazing maximum of 1024 sprites!

This module uses the standard screen viewer.

Commands List

Key	Command description
R	Restart program
↑	Screen-viewer
N	New memory
P	Protect memory
←	Reset
X	Exit module
B	Background colour
T	Text colour
@	DOS commands
D	Directory
:	Matrix load
;	Matrix save
L	Load sprites
S	Save sprites
SHIFT L	Load 8 sprites
SHIFT S	Save 8 sprites
SHIFT V	View/select/load sprites
M	Menuloader
HOME	Cursor top-left
CLR	Cursor bottom-right
CRSRs	Move cursor
SPACE	Draw - set/clear pixels
1-4	Select draw colour
SHIFT 1-4	Set colours
SHIFT X	X-symmetry on/off
SHIFT Y	Y-symmetry on/off
H	Matrix horizontal flip
V	Matrix vertical flip
+	Roll matrix up
-	Roll matrix down
SHIFT +	Roll matrix left
SHIFT -	Roll matrix right
CBM +	Roll matrix row left
CBM -	Roll matrix row right
A	Matrix 'AND' store
SHIFT A	Matrix 'AND' memory
E	Matrix 'EOR' store
SHIFT E	Matrix 'EOR' memory
O	Matrix 'OR' store
SHIFT O	Matrix 'OR' memory
F7	Matrix to store
SHIFT G	Memory to matrix
F8	Store to matrix
SHIFT P	Matrix to memory

F5	Swap matrix with store
F2	Swap matrix with memory
CTRL R	Reverse matrix
F1	Hires/multi
C	Clear matrix sprite
SHIFT C	Clear memory sprite
<	Scroll sprites left
>	Scroll sprites right
F3	Video bank up
F4	Video bank down
G	Goto memory sprite
*****	Sprites display mode
CTRL 1-8	Auto-sprite get
F6	Repeats on/off

The joystick can be used to move the cursor in all 8 directions. The fire button will set/clear pixels like the spacebar

Screen Layout

Displayed at the top-left of the screen is the sprite on the edit matrix. This sprite is displayed in all four modes - normal, X-expanded, Y-expanded, both. Below this is the store sprite and next to this the first of the 8 sprites. This sprite and the seven below it can either display the memory sprites or the sprites as they were when you froze the program in memory - see '*' command.

As you load or save sprites they will be displayed one at a time. All saves set a reload address of $2000. no prefix is specified, so you can load and save sprites from any sprite editor.

If you attempt to load more sprites than there are in a file, or if you load less sprites than there are in a file, the excess will be ignored.

: Matrix Load

This will load a sprite into the matrix - this doesn't affect computer memory.

; Matrix Save

This will save the matrix sprite.

L Load Sprites

This allows you to load sprites directly into memory. You can specify the number of sprites to be loaded and where they will load.

S Save Sprites

This allows you to save sprites directly from memory. You can specify the number of sprites to be saved and where they should save from.

Both of the above let you specify the start sprite and the end sprite. this can be anything from 1 sprite to the whole bank of sprites.

These two commands are only available when you are viewing memory sprites - see the '*' command.

SHIFT L Load 8 Sprites

This command will load sprites into the locations used by the current 8 screen sprites.

SHIFT S Save 8 Sprites

This command will save the current 8 screen-sprites.

SHIFT V View/Select/Load Sprites

This unique command lets you search a file and select 8 sprites to be loaded. As each sprite is displayed, the load routine will wait for you to press:

RETURN	Load sprite
Any key	Next sprite
STOP	Quit load

The sprites will go in the locations for the 8 current sprites.

SHIFT X X-Symmetry On/Off

This command switches the X-symmetry on or off. The X position flag is reversed when symmetry is on.

SHIFT Y Y-Symmetry On/Off

This command switches the Y-symmetry on or off. The Y position flag is reversed when symmetry is on.

H Horizontal Flip

The matrix is flipped horizontally, side to side.

V Vertical Flip

The matrix is flipped vertically, top to bottom.

+ Roll Matrix Up

The matrix is rolled up one pixel-row and the top row appears at the bottom.

- Roll Matrix Down

The matrix is rolled down one pixel-row and the bottom row appears at the top.

SHIFT + Roll Matrix Left

The matrix is rolled one pixel-column left and the leftmost column appears on the right.

SHIFT - Roll Matrix Right

The matrix sprite is rolled one pixel-column right and the rightmost column appears on the left.

CBM + Roll Matrix Row Left

The row on which the cursor is currently positioned is rolled one pixel left and the leftmost pixel appears on the right.

CBM - Roll Matrix Row Right

The row on which the cursor is currently positioned is rolled one pixel right and the rightmost pixel appears on the left.

The next six commands will probably not be used that much. They perform logical operations on the matrix sprite. Refer to the Commodore 64 Programmer's Reference Guide for a full description of the AND, OR and EOR commands.

A Matrix 'AND' Store

The matrix is ANDed with the store and the result is placed back in the matrix.

E Matrix 'EOR' Store

The matrix is EORed with the store and the result is placed back in the matrix.

O Matrix 'OR' Store

The matrix is ORed with the store and the result is placed back in the matrix.

SHIFT A Matrix 'AND' Memory

The matrix is ANDed with a memory sprite and the result is placed back in the matrix.

SHIFT E Matrix 'EOR' Memory

The matrix is EORed with a memory sprite and the result is placed back in the matrix.

SHIFT O Matrix 'OR' Memory

The matrix is ORed with a memory sprite and the result is placed back in the matrix.

F7 Matrix To Store

This command quite simply copies the matrix to the store.

F8 Store To Matrix

This is the inverse of the above and copies the store to the matrix.

F5 Swap Matrix With Store

This command swaps the matrix with the store.

The following 3 commands allow you to select which sprite you would like to work with.

SHIFT G Memory To Matrix - Get Sprite

This lets you get a sprite from memory and put it in the matrix.

SHIFT P Matrix To Memory - Put Sprite

This lets you put the matrix to memory.

F2 Swap Matrix With Memory

This lets you swap the matrix with the memory sprite.

CTRL R Reverse Matrix

The matrix is reversed - press again to return to normal.

F1 Hires/Multi

This command toggles between hires and multicolour mode. If you are looking at memory sprites these will also be shown in the current mode. In hires only colours 1 and 2 apply.

C Clear Matrix Sprite

The matrix will be filled with 00 which will clear it.

SHIFT C Clear Memory Sprite

The selected sprite will be cleared.

The next five commands only work if you are in memory-sprite mode - see the '*' command.

< Scroll Sprites Left

The sprites will be scrolled left. If you reach the end of the bank, it will wrap round to the other end

> Scroll Sprites Right

The sprites will be scrolled right. If you reach the end of the bank, it will wrap round to the other end

F3 Video Bank Up

The video bank will be incremented. The sprite addresses will reflect the bank.

F4 Video Bank Down

The inverse of the above command. The video bank is decremented.

G Goto Memory Sprite

This command lets you jump directly to a sprite as opposed to scrolling to it.

### *		Sprites Display Mode

1. Memory Sprites - one sprite after another upwards through memory.

2. Game Sprites - the 8 sprites which were being used by the program in memory when you froze it. These sprites will appear as they were on the screen except they will all be standard size.

Use the '*' command to toggle between these two modes.

CTRL 1-8 Auto-Sprite Get

These commands will get the game sprites and put them in the matrix and set the colours and the hires/multi mode.

### F6		Repeats On/Off

This command just switches the repeats mode on or off. When you are scrolling the sprites (< or >) it is easier if the repeats are switched on.

Disk Editor

The disk editor allows you to edit bytes directly on the disk and has some extra features. It supports the standard screen-viewer.

Commands List

Key	Command description
R	Restart program
↑	Screen-viewer
N	New memory
P	Protect memory
←	Reset
X	Exit module
@	DOS commands
D	Directory
HOME	Cursor top-left
CLR	Cursor bottom-right
CRSRs	Move cursor
SHIFT 1-4	Set sprite colours
L	Load sprite
S	Save sprite
F1	Read block
F2	Write block
F3	Next block
F4	Previous block
F7	Put in buffer
F8	Get from buffer
F5	Swap with buffer
F6	Repeats on/off
I	Initialise
J	Jump to link
-	Block down
+	Block up
SHIFT -	Track down
SHIFT +	Track up
*****	Display mode
SHIFT *	Text display mode
F	Find memory
SHIFT F	Find mode
G	Get address
A	Set address
W	Set word
B	Byte edit
T	Text edit
C	Clear bytes
H	Hardcopy
K	Kill block stack
M	Menuloader

Screen Layout

```
TR SC POS VAL  BINARY   WORD ..........
12 01  2D C7  11000111  FB3E .        .
NEXT  ASC SCR TXT FIND* ADDR .        .
12 04  G   G  ASC  PRG  0801 ..........
```

TR SC	Current track and sector
NEXT	Link to next track and sector
POS	Current cursor position
VAL	Value of current byte
ASC	ASCII equivalent of byte
SCR	Screen-poke equivalent of byte
BINARY	Binary equivalent of byte
TXT	Current text-display mode
FIND	Current find mode
WORD	Two-byte word at cursor
ADDR	Address in file

Displayed below this is the block either in byte or text format - see * command. The bottom line is an information/input line.

HOME Cursor Top-Left

Place cursor at position 00.

CLR Cursor Bottom-Right

Place cursor at position FF.

CRSRs Move Cursor

Move cursor around display area.

SHIFT 1-4 Set Sprite Colours

Positioned top-right of the screen is the sprite equivalent of $3F bytes from the current cursor position. As you move the cursor this will constantly change. Use these key to set the colours.

L Load Sprite

This special load command allows you to look through a file and load a sprite.

The sprite is displayed and the routine will wait for you to press:

RETURN	Load sprite
Any key	Next sprite
STOP	Quit load

S Save Sprite

Save current sprite.

F1 Read Block

Input track and sector to read a block.

F2 Write Block

Input track and sector to write a block.

F3 Next Block

The linking block displayed as next will be read into memory.

F4 Previous Block

Every time you read a block the track and sector are put into a table. This lets you move backwards through a file. 255 blocks can be stored.

F7 Put In Buffer

The buffer can hold 256 bytes (a block). Memory can be stored here and then searched for on the disk - see 'F' Find.

This command allows you to put from 1 to 256 bytes in the buffer.

PUT FROM 07 TO 3E

This would put the bytes from pos 07 to pos 3e inclusive into the buffer. To put just 1 byte in the buffer the from and to values should be the same.

F8 Get From Buffer

Get the bytes previously stored in the buffer back into the block memory.

F5 Swap With Buffer

Swap the bytes preeviously stored in the buffer with an equal number of bytes in the block memory.

When you store bytes in the buffer a '*' will be displayed by find to show that the buffer has contents.

If the '*' is not there you cannot use the 'F' Find command

F6 Repeats On/Off

This command just switches the repeats m ode on or off.

I Initialise Block

The disk is initialised and the current block re-read.

J Jump To Link

The link pointed to by the 2 bytes at the cursor position are set as the next track and sector to be read.

+ Block Up

The next backard sequential block will be read into memory.

+ Block Up

The next forward sequential block will be read into memory.

SHIFT - Track Down

The next forward sequential track will be read into memory.

SHIFT + Track Up

The next forward sequential track will be read into memory.

If the next block to be read is not a legal block then the command is ignored.

* Display Mode

The block can be displayed either as bytes (BYT) or text (TXT).

SHIFT * Text Display Mode

The text display can be altered between ascii (ASC) and screen-pokes (SCR).

F Find Memory

As mentioned above memory can be looked for directly on the disk. This command will move upwards through the disk looking for the memory. If it finds a match the cursor will be placed where the match begins.

SHIFT F Find Mode

The find command can look through a disk in two ways:

1. PRG - Move through a file
2. SEQ - Move through the whole disk

The look command will start from the block it is currently on.

G Get Address

The first four bytes of a file block contain the following:

00	Next track
01	Next sector
02	Load address lo
03	Load address hi

This command will get this address. Obviously if the block you are on is not the first block of a file then this will be wrong. As you move upwards through a file this value will increase accordingly. This allows you to edit a byte at a specific address without using the monitor.

A Set Address

Allows you to set the current address.

W Set Word

A word is a two byte value such as FCE2. As you move the cursor this value will change showing you the new word value. This lets you set a word directly.

B Byte Edit

This allows you to directly edit the bytes. You can still use the cursors.

T Text Edit

This allows you to directly edit the text. You can still use the cursors.

C Clear Bytes

This allows you to fill a set number of bytes with a set value.

CLEAR FROM ?? TO ?? WITH ??

H Hardcopy

The current block details are dumped to a printer. Byte and text displays.

K Kill Block Stack

The block stack can hold 256 blocks in the table before it starts shifting them off the bottom and putting new ones at the top. However, if you wish to start from a set block then this command allows you to zero the block stack so that if you use the previous block 'F4' command you will return to this block.

M Menuloader

This command invokes the menuloader. Press STOP to quit this command.

Hacker/Hardcopy

This module contains the following:

1. Immortaliser
2. Joystick autofire
3. Joystick swap
4. Sprite killer
5. Poker
6. Normal hardcopy
7. Reverse hardcopy

This module also supports a very special screen-viewer which manages to by-pass the problem encounterd by the others. This is because it situates itself outside of the expert and switches off the expert. This means that the expert can not display an image of itself. The reason not all modules have this is because it is very memory-expensive!

Screen-Viewer Commands

←	Sprites on/off
1	Increment border colour.
2	Increment screen colour.
F1	Hires/multicolour mode.
F3	Video bank up.
F5	Pixels lo/hi.
F7	Screen up.
RETURN	Character set up.
STOP	Quit back to module.
SPACE	Return to game.

Commands List

Key	Command description
R	Restart program
↑	Screen-viewer
N	New memory
P	Protect memory
←	Reset
X	Exit module
@	DOS commands
D	Directory
HOME	Cursor top selection
CLR	Cursor bottom selection
CRSRs	Cursor up/down one selection
RETURN	Execute selection
M	Menuloader

Instead of selecting an option with a key this module allows you to select an option with the cursor keys, The option will be highlighted, press 'RETURN'.

Immortaliser

This will attempt to find an infinite-lives poke for you. Input the number of lives you currently have. In response to 'BASE' input 00 and press 'RETURN'. Memory will be inspected and if it finds an instruction which decrements a memory location which holds the lives value it will store it in a table and disassemble the instruction to the screen. When it has inspected all memory or has filled the table it will prompt you to lose a life. Restart the game ('R'), lose one life and re-enter the module.

Run the immortaliser again. It will look at the memory locations in the table to see if they contain the correct value. If it finds a match it will prompt you with (Y/N)? If you wish to install the immortaliser press Y. It will continue doing this until the whole table has been scanned. The table can hold 232 possibilities.

If on the first pass it does not find a decrement instruction the routine will be cancelled.

The 'BASE' value is just a little more complicated. If the lives value is not held in memory as a straight number you can try adding offsets to it by setting this value. 'Terra Cognita' uses a base value of 'F0'.

This routine is not guaranteed to work with everything but worked with over 50% of the software we tested it on.

Joystick Autofire

This inspects memory and any references to the joystick port are re-routed to a small patch routine which will give you an autofire option. Select either joystick 1 or joystick 2.

Joystick Swap

This inspects memory and any reference to the joystick port will be swapped to the other port. Select either '1 to 2' or '2 to 1'.

Sprite Killer

This inspects memory for any reference to the sprite collision locations ($D01E or $D01F) and cancels them. Select either sprite to sprite or sprite to background.

Poker

This quite simply allows you to enter pokes listed in magazines without having to reset to Basic. i.e. Armalyte, this will allow you to make a copy:

```
POKE $DC06,0
POKE $DC07,0
```

Numbers can be decimal or hexadecimal and 1 to 5 digits - POKE 1,$37

Hardcopy

The screen-dump can be printed normally or in reverse. It will not print sprites. If the screen was rastar-split, you may not always see what was displayed. Use the screen-viewer to check this.

If, for any reason, you wish to quit the print routines, press STOP. Do not press the RESET/RESTORE key as this may crash the operating system.

Instructions

This module is really very easy to use. How else did you get this far !?!

Select the topic for instructions, press return and they will be listed. Press STOP at any time to quit to menu. Spacebar to continue listing.

That's all!

Trouble Shooting

If you are having any problems with the software please read this section first. If you are having any problems with the Expert please refer to the supplier.

1. It is possible for the Expert to be activated by generating NMIs. To bypass this leave the Expert in the off position until you decide to freeze.

2. When you use the New function, memory will be filled with a single byte. Some programs may look for this. if this is the case use Protect.

3. It is highly unlikey that the Expert will develop any faults. If you are having problems loading any operating system try removing and re-inserting the Expert. Don't forget, if you have a Commodore 128 you have to hold down the CBM key when you reset.

4. If the disk turbo does not work with your drive switch it off with 'F1'.

5. When in the Expert environment, it is inadviseable to press the RESTORE key or the reset as they may crash it. If you need to quit a mode STOP usually allows you to do this as with the Screen-Viewer and printouts. If the menu fails to load this is the reason unless the disk is faulty.

NOW Computer Systems
19, Beadon Road,
Bromley,
Kent BR2 9AS