ENTERPRISE
COMPUTERS LTD

31-37 Hoxton Street
London N1 6NJ
Telephone 01-739 4282
Telex 22717 ENTER G

ENTERPRISE 64 AND 128.

TECHNICAL DESCRIPTION.

1. PROCESSOR.

Z80 microprocessor running at 4MHZ. Extended addressing allows up to 3.9 megabytes of memory.

2. MEMORY.

User RAM	50K (Enterprise 64)
	113K (Enterprise 128)
ROM	32K internal (EXOS + WP)
	16K cartridge (IS-BASIC)

Memory can be expanded to a total of 3.9 megabytes (RAM and/or ROM). Memory extensions are automatically recognized and used by EXOS. Multiple BASIC programs may exist in memory and can be CHAINED from other programs with parameter passing. Operating system (EXOS) extensions can reside in ROM or RAM.

3. GRAPHICS.

The following graphics and text modes are available.

42 column text mode (2 colour pairs)
84 column text mode (4 colour pairs)
672 x 256 HIRES graphics mode (2 colour)
336 x 256 HIRES graphics mode (4 colour)
168 x 256 HIRES graphics mode (16 colour)
84 x 256 HIRES graphics mode (256 colour)
336 x 256 LORES graphics mode (2 colour)
168 x 256 LORES graphics mode (4 colour)
84 x 256 LORES graphics mode (16 colour)
42 x 256 LORES graphics mode (256 colour)
GRAPHICS ATTRIBUTE mode (16 colour)

In the 2, 4 and 16 colour modes, the colours can be chosen from the palette of 256.

Additionally, the 'interlace' modes give 2 colour, 84 x 50 column text and 672 x 512 graphics resolution. These require the 'interlace driver' to be loaded from the demonstration tape supplied with the computer. A long persistence monitor would be required to minimise the 'flicker' present on the interlaced display.

The graphics on the Enterprise are handled by a custom video processor (NICK) and can easily be manipulated from BASIC.

4. SOUND.

Comprehensive 4-source (3 tone and noise) custom sound generator (DAVE) with full stereo control of all sources. 8 octave range with facilities for ring modulation and filtering.

Sounds can be easily manipulated from BASIC with PITCH, DURATION and ENVELOPE control.

5. KEYBOARD.

The keyboard is full sized, full travel with the ISO standard layout. It has 69 keys plus a joystick for cursor control. The 8 function keys are pre-programmed with functions for BASIC programming and the word processor. The keys can be re-programmed in BASIC to give 16 user defined commands (NORMAL and SHIFTed) or 32 functions (using ALT and CONTROL) for applications programs. The joystick is decoded 8 ways for games use. Editing keys include ERASE, DELETE and INSERT.

6. BASIC.

The Enterprise uses a fully structured BASIC based on the new ANSI standard. It supports DO loops and CASE statements and is fully recursive. Extensions to the ANSI standard allow the easy use of graphics and sound.

7. Large range of interfaces:-

a) 2 joystick control ports, configured as 3 x 5 matrix so could be used for keyboard extensions. Readable from BASIC.

b) Centronics compatible printer port. This is a full 8-bit port.

c) RS423 serial port. Software controlled, baud rates from 75 to 9600, full data format control.

d) Built-in Local Area Network (LAN). This allows 32 Enterprise computers to be linked to each other and/or disk drives, printers, etc.. There is no need for expensive add-ons, as the built-in hardware and software controls the net. Pupil/Teacher cartridges will be produced for specific applications.

e) Two remote control sockets for controlling cassette recorders, plus tape in/out sockets.

f) TAPE OUT socket doubles as a stereo output for headphones with 3.5mm jacks.

g) Monitor Socket allows connection of composite (monochrome) and R,G,B, analogue monitors. (Analogue inputs are required to allow all 256 colours to be displayed), and stereo sound output for hi-fi or monitor with audio.

h) Built-in UHF modulator tuned to channel 36 for use with domestic TV sets (colour or black and white).

i) 9.5v power supply input socket. (Separate transformer supplied).

j) Versatile cartridge port (up to 64K ROM in 16K segments).

k) 66-pin expansion port complete with four extra address lines for almost unlimited expansion. See future products.

l) A disk interface will be available later in 1985. This will allow the use of Cumana or similar drives as used with the BBC microcomputer. The interface is 'double density' and will allow more than 700K on a single 80 track double sided drive.

8. SCREEN EDITOR.

The Enterprise has a full screen editor with reverse scrolling of the 4K editor buffer. Programs may be easily edited using the INSERT, DELETE and ERASE keys. These work in conjunction with the SHIFT and ALT keys to delete letters, words or whole lines.

9. WORD PROCESSOR.

This is an expansion of the screen editor and allows margin setting, justification, centering and paragraph moving. Text can be saved and loaded from tape and sent to the printer or serial device. The word processor works in both 40 and 80 column modes. 'Help' screens are available by pressing a function key and the remaining keys are set up to operate the other word processor commands.

10. OPERATING SYSTEM.

The Enterprise eXpandable Operating System (EXOS) resides in ROM. It allows extensions (either RAM or ROM based) and can access memory up to the full addressing range of the Enterprise (3.9 megabytes). The input/output system is channel based and channels may be OPENed, CAPTUREd, REDIRECTed, etc., and I/O devices specified. Under normal circumstances, this system is transparent to the user with default channels handling keyboard, screen, cassette, printer, etc..

11. PERIPHERALS.

Enterprise have a range of peripherals including a dot matrix printer, R,G,B, colour monitor and joysticks. Cables and adaptors are available to connect the computer to other commercially available compatible devices.

12. FUTURE DEVELOPMENTS.

a) Disk Interface

A disk interface will be available later this year. It will allow the use of SHUGART 410 compatible 5¼" and 3½" drives such as CUMANA supply for use with the BBC micro. The DOS uses a MS-DOS compatible file structure and will be able to run CP/M programs subject to memory availability.

b) Memory Expansion

64K RAM packs will also be available later this year. They also contain a ROM socket for use with system extensions or other languages.

c) Base Unit

A single RAM pack or the disk interface can be connected to a computer using a simple (supplied) adaptor. To connect more than one item, the Base Unit would be required. This will accept up to six expansion packs and has its own power supply.

13. DIMENSIONS.

Width 39 cms.
Depth 26 cms.
Height 3 cms.
Excluding joystick top and BASIC cartridge.
Seperate mains power supply unit.
Weight (packaged) 4.5 kgs.

14. SUPPLIED ACCESSORIES.

Power supply	Cassette leads (2)
BASIC cartridge	TV lead
BASIC programming guide	Setting up guide
Demonstration cassette	Demo. cassette guide

Enterprise Computers Ltd. reserve the right to change specifications at any time without notice.

JULY 1985

ENTERPRISE
COMPUTERS LTD

15.2.85

31-37 Hoxton Street
London N1 6NJ
Telephone 01-739 4282
Telex 22717 ENTER G

APPLICATION NOTE NO. 2

CASSETTE SAVING INSTRUCTIONS

1) Connect grey REM plug to REM socket on cassette recorder (if one is fitted), connect other end to REM2 socket on the computer.

2) Connect black plug to OUT socket on computer and other end to MIC, SAVE, or IN socket on cassette recorder (check your recorder manual if you have difficulty).

3) Put a blank tape into the recorder and wind past the "leader".

4) Press PLAY and RECORD buttons on the recorder and make sure that the PAUSE button (if fitted) is not depressed.

5) Type SAVE "FILENAME" [ENTER]
on the computer where FILENAME is the name of your program.

6) The tape should start turning and a noise will be heard from the speaker. The message SAVING FILENAME should appear on the top of the screen.

7) When the program has been saved, the 'OK' prompt will appear on the screen and the saving message will disappear from the top of the screen.

TO VERIFY THAT A PROGRAM HAS BEEN SAVED CORRECTLY

8) Connect the black plug on the other cassette lead to the IN socket on the computer and the other end to the EAR, LOAD, or OUT socket on the recorder (see your recorder manual for details).

9) Wind the tape back to just before the point at which you started saving.

10) Press PLAY on the tape recorder (be careful, pressing RECORD as well will wipe out your program).

11) Type VERIFY "FILENAME" [ENTER]
The tape should start turning and the messages SEARCHING and then LOADING FILENAME should appear. If the program verifies correctly then the 'OK' prompt will appear and the messages will disappear from the top of the screen.

It may be necessary to adjust the volume level on your tape recorder (see the 'Setting Up Guide' for details). Repeat the above once you have set the volume correctly.

ENTERPRISE
COMPUTERS LTD

31-37 Hoxton Street
London N1 6NJ
Telephone 01-739 4282
Telex 22717 ENTER G

APPLICATION NOTE NO.3

CONVERTING MICROVITEC COLOUR MONITORS FROM TTL TO LINEAR INPUT

The Enterprise computers are capable of producing 256 colours. For all of these colours to be displayed correctly, a monitor with linear (analogue) inputs must be used.

These notes describe how to convert 1431,1441 and 1451 Microvitec monitors from TTL level to linear inputs.

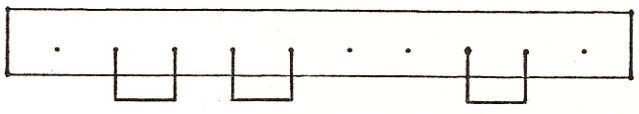
CAUTION - if you are not familiar with this sort of operation, it is wise to seek advice from a qualified television engineer or repairman as lethal voltages are present inside the case. Enterprise computers can accept no responsibility for any accidents or damage caused.

Instructions:-

1. Disconnect monitor from the mains. If the monitor has been used recently, then some internal components may still retain high voltage charges - leave for at least 2 hours before dismantling.

2. If the monitor is of the plastic cased variety, then lay it screen down on a table. Place a cloth or some foam under the screen to prevent damage. The outer case can now be removed by undoing six screws. Metal cased monitors need only have their back panels removed.

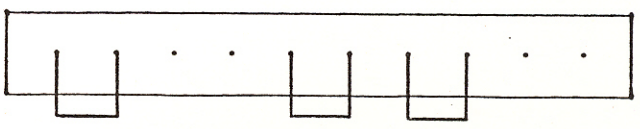
3. Locate the '10 pin in-line' connector (PL 103) on the main PCB.
   If configured for TTL levels, it should look like this:-



Where  represents a wire link.

4. Remove wire links.

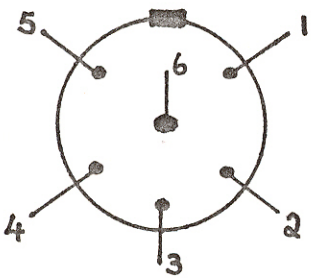
5. Re-instate wire links so that connector looks like this:-



DO NOT OPERATE WHEN DISMANTLED OR BACK REMOVED.

6. Re-assemble case/put back on.

7. The input connections are now as follows (6 pin din):-



1. Red input (RED WIRE)
2. Green input (GREEN WIRE)
3. Blue input (BLUE WIRE)
4. Composite Sync. (BROWN WIRE)
5. OV (BLACK WIRE)
6. Not connected.

IMPORTANT - if using 900-04 video cable - INSULATE ALL OTHER WIRES.

ENTERPRISE
COMPUTERS LTD

31-37 Hoxton Street
London N1 6NJ
Telephone 01-739 4282
Telex 22717 ENTER G

APPLICATION NOTE NO. 4

USE OF THE "REDIRECT" COMMAND

REDIRECT can be used to take an input from a channel and send it elsewhere. The following example shows a REDIRECT from the printer channel (£104) to the screen (£102).

Load any BASIC program.

Type the following in immediate mode:-

```
REDIRECT FROM £104 TO £102
SET £104:SCROLL ON
```

On pressing LLIST (Function Key 2, shifted) the normal printer output will appear on the screen. As this output does not use the editor, it is far faster than a normal screen LIST. Without the SCROLL ON command, the last line would be over-written if the program were more than one screen in length.

You will need to type:-

```
REDIRECT FROM £104 TO £104
```

to obtain normal printer output.

ENTERPRISE
COMPUTERS LTD

31-37 Hoxton Street
London N1 6NJ
Telephone 01-739 4282
Telex 22717 ENTER G

APPLICATION NOTE NO. 5

Connecting an Enterprise 64/128 to a Hitachi CPT 1444 TV receiver.

Use Enterprise monitor cable part no. 900-04.

The following additional parts are required:-

Quantity	
1	7 pin DIN plug.
3	270 ohm 1/8 WATT resistors.
2	100 ohm 1/8 WATT resistors.
	Insulating tape.

NOTE: The YELLOW and VIOLET wires on the 900-04 cable are not used. However, they must be CAREFULLY INSULATED from each other and the other connections.

Connections:-

900-04	Signal	Resistor	DIN plug
Red	Red	270	1
Green	Green	270	2
Blue	Blue	270	3
Brown	Sync.	NONE	4
Black	Common	NONE	5
Orange	Left Audio	100)	6
White	Right Audio	100)	

ENTERPRISE
COMPUTERS LTD

31-37 Hoxton Street
London N1 6NJ
Telephone 01-739 4282
Telex 22717 ENTER G

APPLICATION NOTE NO. 6

ASCII CODES PRODUCED BY KEYBOARD

These are the ASCII codes returned by each key both normally, and with SHIFT, CTRL and ALT. (All values are in hexadecimal.)

Key	NORMAL	SHIFT	CONTROL	ALT
1	31	21	31	31
2	32	22	32	32
3	33	23	33	33
4	34	24	34	34
5	35	25	35	35
6	36	26	36	36
7	37	27	37	37
8	38	28	38	38
9	39	29	39	39
0	30	5F	1F	9F
-	2D	3D	2D	2D
^	5E	7E	1E	9E
@	40	60	00	80
[	5B	7B	1B	9B
;	3B	2B	3B	3B
:	3A	2A	3A	3A
]	5D	7D	1D	9D
\	5C	7C	1C	9C
,	2C	3C	2C	2C
.	2E	3E	2E	2E
/	2F	3F	2F	2F
space	20	20	20	20

A	61	41	01	81
B	62	42	02	82
C	63	43	03	83
D	64	44	04	84
E	65	45	05	85
F	66	46	06	86
G	67	47	07	87
H	68	48	08	88
I	69	49	09	89
J	6A	4A	0A	8A
K	6B	4B	0B	8B
L	6C	4C	0C	8C
M	6D	4D	0D	8D
N	6E	4E	0E	8E
O	6F	4F	0F	8F
P	70	50	10	90
Q	71	51	11	91
R	72	52	12	92
S	73	53	13	93
T	74	54	14	94
U	75	55	15	95
V	76	56	16	96
W	77	57	17	97
X	78	58	18	98
Y	79	59	19	99
Z	7A	5A	1A	9A

ENTER	0D	0D	0D	0D
ESC	1B	1B	1B	1B
TAB	09	09	09	09
DEL	A0	A1	A2	A3
ERASE	A4	A5	A6	A7
INS	A8	A9	AA	AB
STOP	03	03	03	03
joy up	B0	B1	B2	B3
joy down	B4	B5	B6	B7
joy left	B8	B9	BA	BB
joy right	BC	BD	BE	BF
Function 1	.	.	F0	F8
Function 2	.	.	F1	F9
Function 3	.	.	F2	FA
Function 4	.	.	F3	FB
Function 5	.	.	F4	FC
Function 6	.	.	F5	FD
Function 7	.	.	F6	FE
Function 8	.	.	F7	FF

ENTERPRISE
COMPUTERS LTD

31-37 Hoxton Street
London N1 6NJ
Telephone 01-739 4282
Telex 22717 ENTER G

APPLICATION NOTE NO. 7

SCREEN MODES ON THE ENTERPRISE

TEXT MODES	APPROX. MEMORY USAGE.
42 column x 28 lines (2 colour pairs)	4K
84 column x 28 lines (4 colour pairs)	18K

LO-RES GRAPHICS		
42 x 256	256 colour)	
84 x 256	16 colour)	
168 x 256	4 colour)	11K
336 x 256	2 colour)	

HI-RES GRAPHICS		
84 X 256	256 colour)	
168 x 256	16 colour)	
336 x 256	4 colour)	22K
672 x 256	2 colour)	

ATTRIBUTE GRAPHICS

This allows 16 colours with 336 x 256 resolution, only 2 colours per character block are allowed. There are 40 x 25 blocks in a screen. This mode is very memory efficient.

INTERLACE MODES

These require the 'interlace driver' to be loaded from cassette and give 672 x 512 two colour graphics or 84 columns by 50 lines of text. These modes (including the driver) use around 44K of memory. A special 'long persistence' monitor is required for these modes.

SSG/LLB/5/85.

ENTERPRISE
COMPUTERS LTD

31-37 Hoxton Street
London N1 6NJ
Telephone 01-739 4282
Telex 22717 ENTER G

APPLICATION NOTE NO. 8

USE OF THE 'LOOK' COMMAND.

LOOK is a command used in association with graphics modes. Its purpose is to return the palette colour of any specified point on the graphics screen (pixel).

The command takes the form:-

```
LOOK AT X,Y:VALUE
```

X and Y are horizontal and vertical co-ordinates and VALUE is the variable which will hold the palette colour.

Say, for example, that a point is plotted in palette colour 3 at position 500,340 on a graphics screen. Typing:

```
LOOK AT 500,340:V <ENTER>
PRINT V           <ENTER>
```

Will print the number 3.

It should be noted that palette numbers run from Ø to 15. In 256 colours mode, the LOOK function will return the actual INK number which can be any integer from Ø to 255.

Also note that the screen addressing system which is based on a matrix of 1280 x 720 points is not the screen resolution. This means that the LOOK command will generate the same value if the two pairs of co-ordinates actually point to different places within the same pixel.

Here is a program to demonstrate the use of the LOOK command for collision detection. Feel free to experiment with it as you like, and use it in your own programs.

```
100 PROGRAM "bounce"
110 ! By Gerald Morgan
120 GRAPHICS HIRES 16
130 OPTION ANGLE DEGREES
140 SET PALETTE 0,255,RED
150 PLOT 630,360,
160 SET INK 2
170 PLOT ANGLE 180
180 PLOT FORWARD 200,LEFT 90; FORWARD 200,LEFT 90;
    FORWARD 200,LEFT 90; FORWARD 200,LEFT 90;
    FORWARD 200,LEFT 90
190 SET INK 1
200 X=530:Y=260
210 X=X+5
```

CONTINUED/.....2.

```
220 LOOK AT X+5,Y:HIT
230 IF HIT = 2 THEN CALL RETURN
240 PLOT X,Y
250 SET INK O
260 PLOT X-5,Y
270 SET INK 1
280 GOTO 210
290 DEF RETURN
300 SET INK O
310 PLOT X-5,Y
320 PLOT X,Y
330 DO
340 X=X-5
350 LOOK AT X-10,Y:HIT
360 PLOT X,Y
370 SET INK O
380 PLOT X+5,Y
390 SET INK 1
400 LOOP WHILE HIT = O
410 END DEF
```

ENTERPRISE
COMPUTERS LTD

31-37 Hoxton Street
London N1 6NJ
Telephone 01-739 4282
Telex 22717 ENTER G

APPLICATION NOTE NO. 9

MODIFYING THE PERITEL (SCART) CABLE FOR USE WITH CERTAIN TV'S WITH MONAURAL SOUND.

The Enterprise computer produces a stereo sound signal that is output on pin 2 (RIGHT) and pin 6 (LEFT) on the SCART connector. Certain TV receivers do not mix the two signals to produce true mono sound, they just take the LEFT signal and amplify this. The following small modification to the Enterprise cable (part no. 900-02) will produce true mono sound.

Unscrew the cable entry on the SCART connector and dismantle.

Locate the orange and white wires.
Be careful not to break any of the wires.
Connect a 100 ohm 1/8 watt resistor between the two pins.
Re-assemble the connector.

The commands:-

```
          SOUND PITCH 50,LEFT 255,RIGHT 0
and       SOUND PITCH 50,RIGHT 255,LEFT 0
```

should now produce the same audio signal.

ENTERPRISE
COMPUTERS LTD

31-37 Hoxton Street
London N1 6NJ
Telephone 01-739 4282
Telex 22717 ENTER G

APPLICATION NOTE NO. 10.

900-04 COLOUR/SOUND CABLE.

CABLE COLOUR:

RED	RED SIGNAL	(0 to 4V)
GREEN	GREEN SIGNAL	(0 to 4V)
BLUE	BLUE SIGNAL	(0 to 4V)
BROWN	COMBINED SYNC.	(Negative; TTL levels)
BLACK	OV	
VIOLET	COMPOSITE VIDEO	(Monochrome)
ORANGE	LEFT AUDIO	
WHITE	RIGHT AUDIO	
YELLOW	OV	

Colour monitors use the Red, Green and Blue signals and the Combined Sync.. It is important that the monitor used can accept linear signals monitors that use TTL imputs cannot give the full range of colours available from the Enterprise. Microvitec CUB Monitors can accept the correct signals by changing three links inside the cabinet. Instructions for doing this are available from Enterprise or from Microvitec.

Monochrome monitors use the composite video signal.

Hi-fi connections are at normal levels for matching to the 'tape' input on most amplifiers.

IMPORTANT - INSULATE ALL UNUSED WIRES.

ENTERPRISE
COMPUTERS LTD

31-37 Hoxton Street
London N1 6NJ
Telephone 01-739 4282
Telex 22717 ENTER G

APPLICATION NOTE NO. 11.

SERIAL NETWORK CABLE - 900-05.

WIRE COLOUR:

BLUE	DATA OUT
WHITE	DATA IN
BROWN	STATUS OUT (RTS)
GREEN	STATUS IN (CTS)
ORANGE	REFERENCE
BLACK	OV

RS423

Using OV as the return line give approx. +.5V (logical 0) and +12V (logical 1). Certain systems might require a negative voltage for logical 0. In this case use REFERENCE as the return line. This will give approx. -5V (logical 0) and +7V (logical 1).

NETWORK

Connect black wires of all computers together.
Connect both blue and white wires of all computers together to form 'data bus'.
Connect both brown and green wires of all computers together to form 'control bus'

NOTE - ALL COMPUTERS ON THE NETWORK MUST BE SWITCHED ON.

ENTERPRISE
COMPUTERS LTD

31-37 Hoxton Street
London N1 6NJ
Telephone 01-739 4282
Telex 22717 ENTER G

APPLICATION NOTE NO. 12.

PIN CONNECTIONS - 900-10

EDGE CONNECTOR PIN	SIGNAL	COLOUR	7-PIN DIN
A1	GREEN	GREEN	1
B2	OV	BLACK/SCREEN	2
A7	AUDIO LEFT	ORANGE	3
B3	BLUE	BLUE	4
B7	AUDIO RIGHT	WHITE	5
B4	RED	RED	6
B5	/C SYNC	BROWN	7

NOTE: PURPLE CABLE MUST BE INSULATED.

ENTERPRISE
COMPUTERS LTD

31-37 Hoxton Street
London N1 6NJ
Telephone 01-739 4282
Telex 22717 ENTER G

APPLICATION NOTE NO. 13

DUMPING GRAPHICS TO THE PRINTER

The 'dumping' of graphics screens to a printer requires a short BASIC program. A program to do this is included in the form of a procedure.

The computer scans the screen 8 bits at a time, and puts the data into a 180 character buffer. This is then sent to the printer.

Because of the way the graphics screen is encoded, this program will only work in the low resolution graphics modes. However, as LORES 2 has higher resolution than most other computers can manage, this should not be a problem.

To 'dump' an ordinary text screen a program is not required, just type COPY and press ENTER. Here is a sample program to draw a picture and 'dump' it to the printer. The control codes used are for the EP80 and printer, but will work on any Mannesmann Tally printer and most EPSON compatible machines.

GM/LLB/5/6/85.

```
100 PROGRAM "GDUMP_2"
110 GRAPHICS LORES 2
120 SET PALETTE 7,255
130 FOR X=10 TO 100 STEP 12
140   FOR Y=1 TO 1160 STEP 90
150     PLOT Y+X,120+100*SIN(Y/70*PI)+X;
160   NEXT
170   SET BEAM OFF
180 NEXT
190 PLOT 630,360,;ELLIPSE 120,120,PAINT
200 PLOT 30,680,;ELLIPSE 270,270,PAINT
210 PLOT 900,570,
220 FOR X=1 TO 200 STEP 12
230   PLOT ELLIPSE X,200-X,
240 NEXT
250 PLOT 20,400,
260 SET LINE MODE 3
270 PRINT £101:"      Written by Gerald Morgan."
280 !
290 CALL GDUMP
300 !
310 DEF GDUMP
320   NUMERIC DOT(180)
330   LET E$=CHR$(27):LET START=36146:LET FINISH=START+32
340   LPRINT E$;"@";
350   LPRINT E$;"3";CHR$(21);
360   FOR LOOP1=START TO FINISH
370     LPRINT E$;"K";CHR$(180);CHR$(0);
380     LET COUNT=0
390     FOR LOOP2=LOOP1 TO 7199+LOOP1 STEP 40
400       LET DOT(COUNT)=PEEK(LOOP2)
410       LET COUNT=COUNT+1
420     NEXT
430     FOR LOOP3=179 TO 0 STEP-1
440       LPRINT CHR$(DOT(LOOP3));
450     NEXT
460     LPRINT
470   NEXT
480 END DEF
```

ENTERPRISE
COMPUTERS LTD

31-37 Hoxton Street
London N1 6NJ
Telephone 01-739 4282
Telex 22717 ENTER G

APPLICATION NOTE NO. 14

SAVING STRING & NUMERIC ARRAYS

To save numeric arrays:

1. First, create your array e.g. 100 dim x (10).
2. Put what you wish to be committed to tape into the array e.g.:

```
100 for x=1 to 10
110 let t(x)=x*x
120 next x
```

3. Open a channel between 1 and 99 for output e.g.:

```
open £20:"tape:filename" access output
```

The channel number can be in the range of 1 to 254, but some of the channels in this range are used for other purposes such as the graphics screen, etc..

As well as this, only channels between 1 and 99 are closed automatically after use. The file name can be up to 31 characters long (not spaces) or non-existant. If the file name is left out, it is still important to put the colon after the word 'TAPE:'.

4. Straight after the 'OPEN' line you should have the part of the program that writes to the tape. The writing to the tape is achieved with 'PRINT' statements to the relevant channel. So if your 'OPEN' statement looks like this:

```
OPEN £25:"TAPE:FILENAME" ACCESS OUTPUT
```

then you should have a print statement like this:

```
PRINT £25:VARIABLE(K)
```

Where variable (K) is the array you wish to save and K is an element of that variable.

5. Open a channel for input. This need not be the same channel that was used for saving the array. Open the channel like this:

```
OPEN £50:"TAPE:FILENAME" ACCESS INPUT
```

6. Immediately after this line should follow the part of the program that reads the data in:

```
INPUT £50:VARIABLE(K)
```

This is similar to PRINT £50 except that it accepts data from tape instead of pushing it out to tape.

7. Dealing with strings and string arrays.
A dollar sign should be placed after the variable or array name to be saved:

```
PRINT £50:VARIABLE$(K)
```

This is the same for all string operations and is standard BASIC. You should be careful when saving data intending to be shared between more than one array. Unless you have a specific reason to do so do not put more than one print statement on a line if you are printing to tape e.g.:

```
100 PRINT £50:G$(K),P$(K)
```

This will cause out of data errors when you try to read the data back because G$(K) AND P$(K) will be concatenated.

Therefore it is advisable to use more than one line (two in this example) e.g.:

```
100 PRINT £50:G$(K)
110 PRINT £50:P$(K)
```

When loading in the data file it IS acceptable to have the variables for input joined on one line e.g.:

```
INPUT £50:G$(K),P$(K)
```

This is equivalent to:

```
INPUT £50:G$(K)
INPUT £50:P$(K)
```

```
100  PROGRAM "FILE-CREATION"
110  ! AUTHOR G. MORGAN.
120  ! DATE  27/02/85.
130  TEXT
140  STRING PLACE$(9)*20,CAPITAL$(9)*20
150  PRINT AT 5,6:"PRESS RECORD + PLAY ON CASSETTE RECORDER
     THEN HIT THE SPACE BAR"
160  IF JOY(O)=O THEN 160
170  OPEN £99:"TAPE GEO-FILE" ACCESS OUTPUT
180  FOR X=1 TO 8
190  READ PLACE$(X),CAPITAL$(X)
200  NEXT X
210  FOR X=1 TO 8
220  PRINT £99:PLACE$(X)
230  PRINT £99:CAPITAL$(X)
240  NEXT X
250  CLOSE £99
1000 DATA ENGLAND, LONDON, IRELAND, DUBLIN, USA, WASHINGTON,
     SCOTLAND, EDINBURGH, FRANCE, PARIS, ITALY, ROME, HOLLAND
     THE HAGUE, WALES,CARDIFF
```

N.B. TO MAKE THIS DISPLAY NEATER, TRY CHANGING LINE 130 TO:
TEXT 80

```
100  PROGRAM "FILE-READING"
110  ! AUTHOR G. MORGAN.
120  ! DATE  27/02/85.
130  TEXT
140  STRING PLACE$(9)*20,CAPITAL$(9)20
150  OPEN £10:"TAPE:GEO-FILE" ACCESS OUTPUT
160  FOR X=1 TO 8
170  INPUT £10:PLACE$(X)
180  INPUT £10:CAPITAL$(X)
190  NEXT X
200  PRINT:PRINT
210  PRINT TAB(10);".....COUNTRY.....";TAB(40);".....
     CAPITAL....."
220  PRINT TAB(10);"--------------------";TAB(40);"------
     -------------"
230  PRINT
240  FOR X=1 TO 8
250  PRINT TAB(10);PLACE$(X);TAB(40);CAPITAL$(X)
260  NEXT X
270  PRINT TAB(10);"THAT IS THE END OF THIS DATA FILE!"
280  END
```

N.B. TO MAKE THE DISPLAY NEATER, TRY CHANGING LINE 130 TO:
TEXT 80

ENTERPRISE
COMPUTERS LTD

31-37 Hoxton Street
London N1 6NJ
Telephone 01-739 4282
Telex 22717 ENTER G

APPLICATION NOTE NO. 15

SAVING AND LOADING OF GRAPHICS SCREENS

The saving and loading of graphics screens requires two short machine code programs. These programs are included and work in graphics mode 4 only, due to the way in which the screen is organised. The 'save' program draws a picture and then saves it.

Once the program has been entered, save it to tape. Now run the program. If it works, the BASIC part can be NEWed leaving the code in the memory until switch-off. The same applies for the 'LOAD' program.

If the program does not work:

1. Reset computer.
2. Load program back by typing 'LOAD""'
3. DO NOT RUN PROGRAM!
4. List program and compare with supplied listing.
5. Make any necessary alterations.
6. Repeat instructions in paragraph 2.

```
100 PROGRAM "2:VSAVE"
110 ! To use this routine see the lines 200 to 220
120 ALLOCATE 200
130 CODE VSAVE=HEX$("E5,21,CB,D,AF,6,10,2B,77,10,FC,23,36,20,23,D1,D5,7B,6,2,F
7,B,DF,70,23")
140 CODE =HEX$("71,23,72,23,73,23,E5,ED,62,B2,59,54,19,10,FD,28,13,5D,54,29,29
,29,19,3D,28,A,FE,4")
150 CODE =HEX$("28,6,D3,28,2,29,3E,19,EB,E1,73,23,72,EB,D1,D5,7A,11,BB,D,1,10,
0,F7,8,DF")
160 CODE =HEX$("D1,D5,7B,6,3,F7,B,DF,D1,DB,B2,F5,C5,78,7,7,E6,3,F6,FC,D3,B2,CE
,F8,CB,B0,A,47,7A,D5,F7,7,DF,D1")
170 CODE =HEX$("C1,3,2B,7C,B5,20,E3,F1,D3,B2,C9")
180 GRAPHICS HIRES 4
190 CALL PRETTY
200 OPEN £2:"2:VPAGE" ACCESS OUTPUT
210 CALL USR(VSAVE,256*2+101)
220 CLOSE £2
230 DEF PRETTY
240   PLOT 630,360,
250   LET INK=0
260   FOR X=1 TO 500 STEP 8
270     LET INK=INK+1
280     IF INK>3 THEN LET INK=0
290     SET INK INK
300     PLOT ELLIPSE X,500-X
310   NEXT X
320 END DEF
```

```
  100 PROGRAM "2:VLOAD"
  110 ! To use this routine see lines 170 to 190
  120 ALLOCATE 200
  130 CODE VLOAD=HEX$("E5,11,BB,D,7C,6B,62,1,10,0,F7,6,DF,7E,B7,20,51,23,7E,FE,2
0,20,4B")
  140 CODE =HEX$("23,D1,D5,7B,6,2,F7,B,DF,78,BE,20,41,23,79,BE,20,3C,23,7A,BE,20
,3C,7B,BE,20,37")
  150 CODE =HEX$("23,D1,D5,7B,6,3,F7,B,DF,D1,7E,23,66,6F,DB,B2,F5,C5,78,7,7,E6,3
,F6,FC,D3,B2,CB,F8,CB,B0")
  160 CODE =HEX$("C5,D5,7A,F7,5,DF,78,D1,C1,2,C1,3,2B,7C,B5,20,E1,F1,D3,B2,C9,3E
,EE,DF,3E,DE,DF")
  170 GRAPHICS HIRES 4
  180 OPEN £2:"2:VPAGE"
  190 CALL USR(VLOAD,256*2+101)
  200 CLOSE £2
```

ENTERPRISE
COMPUTERS LTD

31-37 Hoxton Street
London N1 6NJ
Telephone 01-739 4282
Telex 22717 ENTER G

APPLICATION NOTE NO. 16.

SCREEN MODES AND MEMORY REQUIREMENTS

This information refers to the default graphics and text pages. It should be noted that the commands 'TEXT 40' and 'TEXT 80' produce screens of 38 and 78 columns respectively due to two columns being reserved for control codes. Screens with 40 to 42 columns are possible from BASIC using other commands.

MODE	DESCRIPTION	MEMORY USAGE
TEXT 40	40 x 28 lines (2 colour pairs)	Approx. 4K
TEXT 80	80 x 28 lines (4 colour pairs)	Approx.18K
LORES 2	336 x 256 pixels (2 colours)	8628 bytes.
LORES 4	168 x 256 pixels (4 colours)	8628 bytes.
LORES 16	84 x 256 pixels (16 colours)	8628 bytes.
LORES 256	42 x 256 pixels (256 colours)	8628 bytes.
ATTRIBUTE	336 x 256 pixels (16 colours)	8628 bytes.
HIRES 2	672 x 256 pixels (2 colours)	13744 bytes.
HIRES 4	336 x 256 pixels (4 colours)	13744 bytes.
HIRES 16	168 x 256 pixels (16 colours)	13744 bytes.
HIRES 256	84 x 256 pixels (256 colours)	13744 bytes.

Any HIRES or ATTRIBUTE mode in conjunction with 80 columns uses 16784 bytes.

Any LORES mode in conjunction with 80 columns uses 9885 bytes.

The ATTRIBUTE mode only allows 2 colours in a block. There are 1000 blocks arranged as a 40 by 25 matrix.

ENTERPRISE
COMPUTERS LTD

31-37 Hoxton Street
London N1 6NJ
Telephone 01-739 4282
Telex 22717 ENTER G

APPLICATION NOTE NO. 17

GRAPHICS PROGRAMMING IN IS-LISP

LISP stands for LISt Processing. Originally designed for computers without sound and graphics, its prime use was text manipulation. However, IS-LISP is special. Containing over twice as many commands and functions as other versions, it fully encompasses the sound and graphics capabilities of the Enterprise. Here is a sample program which draws 'stars' on the graphics screen. (There are two 'programs' defined as CIRCLE and STAR).

```
(DEFUN CIRCLE(X)(ELLIPSE X(DIFFERENCE 260 X)))

(DEFUN STAR(X Y)
(PLOT X Y)
(SETQ C Ø)
(SETQ A Ø)
(LOOP
(UNTIL(EQ C 27Ø)A)
(CIRCLE C)
(SETQ C(PLUS A C))
(SETQ C(PLUS C 30))))
```

To run this program, enter graphics mode by typing:

(GRAPHICS) then type (STAR X Y) where x and y are the co-ordinates of the centre of the star.

ENTERPRISE
COMPUTERS LTD

31-37 Hoxton Street
London N1 6NJ
Telephone 01-739 4282
Telex 22717 ENTER G

APPLICATION NOTE. NO. 18

NETWORKING ON THE ENTERPRISE

See Application Note No. 11 for connection details.

The Enterprise network system allows the transmission and reception of both programs and data.

It is important to set the serial baud rate to 9600 baud (SET SERIAL BAUD 15) as the operating system on 64K computers defaults to 7200 baud while 128K computers default to 9600.

TO SEND A PROGRAM FROM ONE COMPUTER TO ANOTHER.

Sending computer:-

```
SET SERIAL BAUD 15
SET NET NUMBER 1
SAVE "NET-2:"
```

Receiving computer:-

```
SET SERIAL BAUD 15
SET NET NUMBER 2
LOAD "NET-1:"
```

This provides full error checking.

SENDING MESSAGES BETWEEN COMPUTERS.

This would normally be a 'background' task called by an exception handler. A simple program to send and receive messages is shown below.

Sending computer:-

```
100 PROGRAM "MESSAGE_SEND"
110 SET SERIAL BAUD 15
120 SET NET 1
130 OPEN £110:"NET-2:" ACCESS OUTPUT
140 INPUT "WHAT MESSAGE":X$
150 PRINT £110:X$
160 FLUSH £110
170 CLOSE £110
180 END
```

Receiving computer:-

```
100 PROGRAM "MESSAGE_GET"
110 SET SERIAL BAUD 15
120 SET NET 2
130 OPEN £110:"NET-1:" ACCESS INPUT
140 INPUT £110:A$
150 PRINT A$
160 CLOSE £110
170 END
```

A MORE DETAILED APPLICATION ON THIS SUBJECT WILL APPEAR LATER

ENTERPRISE
COMPUTERS LTD

31-37 Hoxton Street
London N1 6NJ
Telephone 01-739 4282
Telex 22717 ENTER G

APPLICATION NOTE NO. 19

USING "ALLOCATE" ON THE ENTERPRISE 64.

There is a bug in the EXOS ROM of the 64K Enterprise which occurs when making room for machine code programs using ALLOCATE. Sometimes this command will corrupt a BASIC program or 'de-allocate' allowing the code to the accidently over-written. The way around this problem is a little BASIC program which will distinguish between the 64K machine and the 128K machine (which hasn't this bug) and take appropriate action.

An important part of the program is line 40 where DESIRED_SPACE can be changed from 200 to whatever value is required, i.e. the amount of bytes to be reserved for machine code.

Explanation/comments:-

Line 10 holds the name of the program.
Line 30 The variable UNIQUE$ just holds a dummy string which is used later in the program to see if the program has been run before.
Line 40 The variable DESIRED_SPACE holds the amount of bytes to be reserved for machine code.
Line 60 checks what machine is being used (64K or 128K)
Line 70 makes T equal to the current allocation base minus the variable size.
Line 90 compares header string to see if program has already been executed.
Line 130 The string after the CODE statement must be the same as UNIQUE$.
Line 140 This re-runs the program.
Line 180 In conjunction with Line 190 sets BASIC's location counter correctly.
Line 210 is the line that will execute if a 128K machine is in use.

On the next page is the main program with a short test program 'tacked-on' at the end.

CONTINUED...../2/

```
10    PROGRAM "ALLOCATE HEADER"
20    ! LINES 100 AND 230 SHOULD CONTAIN THE SAME STRING.
30    UNIQUE$="ANYTHING"
40    DESIRED_SPACE=200
50    SIZE=DESIRED_SPACE+LEN(UNIQUE$)
60    IF VERNUM=2 THEN
70    T=PEEK(544)+256*PEEK(545)-SIZE
80    FOR X=1 TO LEN(UNIQUE$)
90    IF PEEK (X+T-1)<>ORD(UNIQUE$(X:X))THEN
100   ALLOCATE SIZE
110   POKE 542,PEEK(544)
120   POKE 543,PEEK(545)
130   CODE="ANYTHING"
140   RUN
150   END IF
160   NEXT X
170   X=X+T
180   POKE 540,X BAND 255
190   POKE 541,X/256
200   ELSE
210   ALLOCATE DESIRED_SPACE
220   END IF
230   !*** EXAMPLE ***
240   CODE FRED="RUBBISH"
250   PRINT FRED
260   DEF P(X)=PEEK(X)+256*PEEK(X+1)
270   FOR X=538 TO 544
280   PRINT X,P(X)
290   NEXT X
```

ENTERPRISE
COMPUTERS LTD

31-37 Hoxton Street
London N1 6NJ
Telephone 01-739 4282
Telex 22717 ENTER G

Application note number 20

Sorting strings in IS-BASIC.

One of the most common tasks a computer must carry out is the sorting of strings into alphabetical order. There are various methods of doing this but the one used in this application is known as a bubble sort. The name comes from the way that the 'lowest value' words appear to rise to the top during the sorting process.

The program compares two strings in a list: if the 'lowest value' string of the two is in the highest place the two strings are swapped. This type of sorting is faster than if two adjacent strings were tested because unnecessary swaps are reduced.

The procedure SORT is the actual sorting part of the program and can be used on its own. However it is a good idea to type the whole program in so that you can see how it works.

Parts of the listing such as 'TEXT 80' etc. are not essential to the running of the program but they give a neater display.

```
100 PROGRAM "BUBBLE_SORT"
110 ! Written by Gerald R Morgan.
120 TEXT 80
130 SET £102:PALETTE 7,0,121,121
140 PRINT "How many words to you want to sort (1 to 15)?"
150 INPUT PROMPT ">":AMOUNT
160 IF AMOUNT<2 OR AMOUNT>15 THEN 150
170 DIM ARRAY$(AMOUNT+2)
180 PRINT :PRINT :PRINT "Ok. Enter your" AMOUNT "words (max 14 letters):"
190 FOR X=1 TO AMOUNT
200   PRINT X;
210   INPUT PROMPT ">":ARRAY$(X)
220   IF LEN(ARRAY$(X))>14 OR LEN(ARRAY$(X))<1 THEN 200
230   FOR G=1 TO 15-LEN(ARRAY$(X))
240     LET ARRAY$(X)=ARRAY$(X)&" "
250   NEXT G
260 NEXT X
270 CALL SORT
280 CALL DUMP
290 DEF SORT
300   PRINT :PRINT "Sorting, please wait.":PRINT
310   FOR J=1 TO AMOUNT-1
320     FOR I=J+1 TO AMOUNT
330       LET L=AMOUNT+J-I+1
340       IF ARRAY$(J)>ARRAY$(L) THEN
350         LET M=J
360         LET TEMP$=ARRAY$(L)
370         LET ARRAY$(L)=ARRAY$(J)
380         LET ARRAY$(J)=TEMP$
390       END IF
400     NEXT I
410   NEXT J
420 END DEF
430 DEF DUMP
440   FOR X=1 TO AMOUNT
450     PRINT ARRAY$(X)
460   NEXT X
470 END DEF
```

ENTERPRISE
COMPUTERS LTD

31-37 Hoxton Street
London N1 6NJ
Telephone 01-739 4282
Telex 22717 ENTER G

APPLICATION NOTE NO. 21.

USE OF EXCEPTION HANDLERS.

When writing a program, it is important that it is as 'crash-proof' as possible.

In a graphics program, for example, you may want to prevent the cursor leaving the bounds of the screen, causing the program to stop and display an error message.

This is known on most computers as 'error-trapping'. Some BASICs have a primitive 'error-trapping' facility using a syntax such as:-

```
ON ERROR GOTO XXX
```

IS-BASIC, however, goes several steps further in providing EXCEPTION HANDLING Commands which are well structured and more versatile than the 'ON ERROR' variety. These can not only recognise the error, they can recognise the type of error or interrupt that has occurred and respond differently to each.

This is best described by the following program:-

```
 10   PROGRAM "EXCEPTIONS"
 20   ! BY GERALD MORGAN
100   WHEN EXCEPTION USE THE_HANDLER
110   DO
120   INPUT A
130   SOUND PITCH A
140   LOOP
150   END WHEN
160   HANDLET THE_HANDLER
170   IF EXTYPE=1000 THEN
180   PRINT "THAT SOUND VALUE WAS WRONG"
190   LET A=127
200   RETRY
210   END IF
220   IF EXTYPE=20034 THEN
230   PRINT "ONLY NUMBERS PLEASE."
240   RETRY
250   END IF
260   IF EXTYPE=1001 THEN
270   PRINT "EVEN THE ENTERPRISE CAN'T HANDLE
      NUMBERS THAT BIG!"
280   RETRY
290   ELSE
300   PRINT "OK I'LL STOP THIS PROGRAM."
310   END IF
320   END HANDLER
```

Explanation of program:-

LINE	EXPLANATION
10	NAMES THE PROGRAM
20	THE AUTHOR!
100 & 150	ENCLOSES MAIN PROGRAM WITH THE EXCEPTION HANDLING.
110 & 140	CAUSES THE STATEMENTS BETWEEN THESE TWO LINES TO REPEAT INDEFINITELY.
120	WAITS FOR INPUT FROM THE KEYBOARD.
130	MAKES A SOUND WHOSE PITCH VARIES AS THE INPUT VALUE.
160	START THE HANDLER BLOCK AND CALL IT 'THE_HANDLER'
170 - 210	IF THE ERROR NUMBER IS 1000 THEN WRITE A SUITABLE ERROR MESSAGE ON SCREEN, MAKE THE VARIABLE 'A' EQUAL TO A SUPERSONIC SOUND VALUE SO IT CANNOT BE HEARD AND TRY AGAIN.
220 - 250	IF THE ERROR NUMBER IS 20034 THEN WRITE A MESSAGE AND TRY AGAIN.
260 - 310	IF THE ERROR NUMBER IS 1001 THEN WRITE A MESSAGE AND TRY AGAIN. OTHERWISE WRITE OUT THE DEFAULT MESSAGE (THIS COVERS EVERY OTHER POSSIBLE ERROR) AND STOP THE PROGRAM.
320	FINISH THE HANDLER BLOCK.

NOTES:-

1. The ERROR NUMBER is the value associated with a particular error. For a list of these, see pages 204-208 in the Programming Guide.

2. Just pressing the 'ENTER' key without having entered any data is an error and in this case the default message is primed.

3. EXLINE is a variable that holds the line of the program where an exception occurred. EXTYPE is a variable that holds the number of the most recent exception.

4. Typing PRINT EXSTRINGS$(X) where X is a number between Ø and 30000 will print the exception associated with that number.

5. CAUSE EXCEPTION X will cause the exception with number X to occur.

6. The handler-block can be anywhere in the program, but it is better programming practice to put it at the beginning or the end of a program.

ENTERPRISE
COMPUTERS LTD

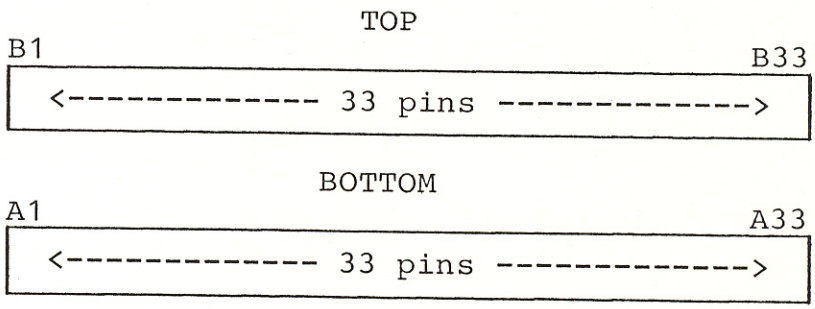
31-37 Hoxton Street
London N1 6NJ
Telephone 01-739 4282
Telex 22717 ENTER G

APPLICATION NOTE NO. 22.

THE ENTERPRISE EXPANSION PORT.

This document gives details of the Enterprise expansion port connections. Great care should be exercised when using this information as damage to the computer could result from mis-use. Such damage is not covered by the Enterprise warranty.

Looking into the Expansion Port. (Double-sided edge connector).



Inputs and outputs in order of pin numbers.

B1	=	LH SOUND IN	A1	=	RH SOUND IN
B2	=	$\overline{WR}$	A2	=	$\overline{RFSH}$
B3	=	$\overline{IORQ}$	A3	=	$\overline{RD}$
B4	=	+17*	A4		+17*
B5	=	$\overline{NM1}$	A5	=	$\overline{MREQ}$
B6	=	A9	A6	=	A8
B7	=	A11	A7	=	A10
B8	=	A13	A8	=	A12
B9	=	A15	A9	=	A14
B10	=	A1	A10	=	A0
B11	=	A3	A11	=	A2
B12	=	A5	A12	=	A4
B13	=	A7	A13	=	A6
B14	=	D1	A14	=	DO
B15	=	D3	A15	=	D2
B16	=	D5	A16	=	D4
B17	=	D7	A17	=	D6
B18	=	$\overline{INT}$	A18	=	$\overline{RESET}$
B19	=	GND	A19	=	$\overline{WAIT}$
B20	=	GND	A20	=	$\overline{M1}$
B21	=	GND	A21	=	1M
B22	=	GND	A22	=	Φ
B23	=	GND	A23	=	8M
B24	=	EC1	A24	=	ECO
B25	=	EC3	A25	=	EC2
B26	=	A16	A26	=	$\overline{EXTC}$
B27	=	A18	A27	=	A17
B28	=	A20	A28	=	A19
B29	=	14M	A29	=	A21
B30	=	VSYNC	A30	=	$\overline{LOCATE}$
B31	=	$\overline{EXP}$	A31	=	GND
B32	=	GND	A32	=	HSYNC
B33	=	+9	A33	=	+9

*Not connected.

Expansion connector assignments by function:-

System control bus connections are:-

A2, B3, A5, A3, B2, A20.

CPU control bus connections are:-

B5, A18, B18, A19.

Data bus connections are:-

A14, B14, A15, B15, A16, B16, A17, B17.

Address bus connections are:-

A6, B6, A7, B7, A8, B8, A9. B9, A10, B10, A11, B11, A12, B12
A13, B13, B26, A27, B27, A28, B28, A29.

Ground (OV) connections are:-

B19, B20, B21, B22, B23, B32, A31, A32.

+9v outputs are:-

A33, B33.

External colour inputs are:-

B24, B25, A24, A25, A26.

Clock outputs are:-

B29, A21, A23.

Video synchronisation outputs are:-

B30, B32.

Sound inputs are:-

A1, B1.

ENTERPRISE
COMPUTERS LTD

31-37 Hoxton Street
London N1 6NJ
Telephone 01-739 4282
Telex 22717 ENTER G

APPLICATION NOTE NO. 23.

HOW TO USE 'CONTROL CODES' AND 'ESCAPE SEQUENCES'.

There are ASCII codes on the Enterprise as on many other computers, which are not used for printing. These codes run from 0 to 31 (decimal). What sets the Enterprise apart from other computers is the wide range of functions that the control codes and excape sequences cater for.

An excape sequence consists of at least 3 bytes of information. These are:-

i) The channel number.
ii) The escape character (chr$(27)).
iii) An ASCII character.
iv) Any number of operands.

Escape sequences use the special 'ESCAPE' non-printing characters. This is:-

```
CHR$(27) e.g.
PRINT £101:CHR$(27);"F"
```

This will activate the FILL routine and is equivalent to PLOT PAINT. Note that the channel number must always be specified regardless of situation. To include the operand(s), turn them into ASCII characters, e.g.:-

```
PRINT £101:CHR$(27);"E";CHR$(45);CHR$(1)
CHR$(145);CHR$(1)
```

This will plot an ellipse 400 pixels high and 300 pixels wide.

Control codes can be used (from BASIC) by simply printing the characters to the appropriate channel, e.g.:-

```
PRINT £102:CHR$(25)
```

will clear to the end of the line on which it is printed.

On the next page is a list of control codes and escape sequences for the video pages. All values in brackets following a control character are in hexadecimal and need to be converted to decimal numbers in the rance 0 to 31.

Control Codes and Escape Sequences.

Character in the range 0 to 31 are control characters and are not printed. Some of these are interpreted by video pages, depending on the mode. Any which are not understood are simply ignored. A special control code is ESCAPE (ASCII 1Bh) which is used to start an escape sequence for carrying out various functions.

Below is a list of the control codes and escape sequences interpreted by the various modes.

Codes Interpreted by Any Video Page

^Z (1Ah) - Clear entire page and home cursor/beam.

^J (0Ah) - Line-feed. Move cursor down to next line (scrolls if at bottom of screen in text mode and scroll is enabled.)

^M (0Dh) - Carriage return. Returns cursor to start of current line

^^ (1Eh) - Cursor/beam home. (ASCII RS)

escK - Define character (see below)

escC - Set all palette colours \ see below for
escc - Set one palette colour / parameters.

escI<n> - Set ink colour to <n> \ See below for details
escP<n> - Set paper colour to <n> / in different modes

esc=<y><x> - Set cursor position (see below)

Codes Interpreted by Graphics Pages Only

escA<xx><yy> - Position beam at co-ordinates (xx,yy) where xx & yy are each 16-bit hex numbers specified low byte first.

escR<xx><yy> - Relative beam movement by amount (xx,yy).

esc@ - Read beam position. (see below)

escS - Set beam on.
escs - Set beam off.

esc.<n> - Set beam to line style <n> - see below.
escM<n> - Set beam to line mode <n> - see below.

esca<n> - Set attribute flags byte to <n>. Only allowed in attribute mode (see below).

escF - Graphics fill - see below.
escE - Plot ellipse - see below.

Codes Interpreted by Text Pages Only

^Y (19h) - Clear to end of line. Does not move cursor.

^H (08h) - Cursor left. (ASCII BS)
^I (09h) - Cursor right. (ASCII TAB)
^K (0Bh) - Cursor up. (ASCII VT)
^V (16h) - Cursor down. (ASCII SYN)

esc? - Read cursor position. Also supported in attribute mode. (see below)

esc.<n> - Set cursor character to character code <n>.

escM<n> - Set cursor to palette colour <n>

escO - Set cursor display on.
esco - Set cursor display off.

escS - Set automatic scroll on
escs - Set automatic scroll off

escU<m><n> - Scroll up lines (m-20h) to (n-20h) m <= n
escD<m><n> - Scroll down lines

ENTERPRISE
COMPUTERS LTD

31-37 Hoxton Street
London N1 6NJ
Telephone 01-739 4282
Telex 22717 ENTER G

APPLICATION NOTE NO. 24.

THE USE OF MULTIPLE PROGRAMS

The Enterprise computers allow more than one independent BASIC program to reside in memory at any one time. Switching between program areas in immediate mode is achieved by the EDIT command. Typing EDIT followed by a number between 0 and 255 will allow you to enter that program area. You can see which area you are in by looking at the status line.

Programs in area 0 can contain up to 48K of code. Programs in other areas can use up to 32K. A BASIC program cannot share a segment with another BASIC program. Therefore a maximum of eight BASIC programs can fit into a 128K machine regardless of whether they are 1K or 16K long. Programs can 'CHAIN' each other and pass variables. The two programs that follow are examples of this. These should be typed in exactly as shown and can be saved to cassette in the normal way so long as the correct program area has been selected.

i) Reset computer.
ii) Type in this program.

```
100  PROGRAM "MAIN_PROGRAM"
110  ! WRITTEN BY GERALD MORGAN.
120  TEXT
130  RANDOMIZE
140  PRINT:PRINT"IF YOU LOOK AT THE 'STATUS LINE' YOU
     WILL SEE THAT THIS IS PROGRAM AREA 0.  THIS IS
     THE PROGRAM AREA THAT IS USED WHEN THE MACHINE IS
     FIRST SWITCHED ON"
150  PRINT AT 12,4:"PRESS THE SPACE BAR TO START..."
160  IF JOY(0)=0 THEN 160
170  TEXT
180  FOR X=1 TO 50
190  SCORE=RND(32767)
200  PRINT AT 10,13:SCORE;"     "
210  NEXT X
220  PRINT AT 10,7:"SCORE="
230  PRINT:PRINT:PRINT:"ENTER YOUR NAME (UP TO 10
     LETTERS):":PRINT
240  INPUT NAME$
250  IF LEN(NAME$)>10 THEN 240
260  PRINT
270  PRINT"PRESS SPACE BAR FOR OTHER PROGRAM."
280  IF JOY(0)=0 THEN 280
290  CHAIN"HI_SCORE"(NAME$,SCORE)
300  END
```

iii) TYPE:EDIT 1 <RETURN>
iv) TYPE THIS PROGRAM IN:

```
100  PROGRAM"HI_SCORE"(NAME$,SCORE)
110  ! WRITTEN BY GERALD MORGAN.
120  TEXT
130  PRINT:PRINT"IF YOU LOOK AT THE 'STATUS LINE' NOW
     YOU WILL FIND THAT THIS IS PROGRAM AREA 1.  THIS
     AREA WAS ENTERED BY THE CHAIN command.  YOUR NAME
     AND SCORE WERE PASSED TO THIS COMPLETELY
     INDEPENDENT PROGRAM";
140  PRINT"IN THE TWO VARIABLES NAME$ AND SCORE."&CHR$(248)
150  PRINT:PRINT:PRINT"YOUR NAME IS";NAME$;"."
160  PRINT:PRINT"YOUR SCORE WAS";SCORE;"."
170  PRINT:PRINT"TO RUN PROGRAM AGAIN TYPE:":PRINT:PRINT"
     EDIT 0 <RETURN> RUN <RETURN>.":PRINT
180  END
```

ENTERPRISE
COMPUTERS LTD

31-37 Hoxton Street
London N1 6NJ
Telephone 01-739 4282
Telex 22717 ENTER G

APPLICATION NOTE NO. 25.

MULTI-COLOURED TEXT ON THE ENTERPRISE.

The Enterprise has two built-in text modes which not only look different, but behave in different ways.

'TEXT 40' is a mode with 38 columns of text each 24 characters deep, (the '40' represents the size of character which could be repeated 40 times across the screen and still be readable on the majority of domestic television receivers. Widths of up to 42 characters are possible using the SET VIDEO X command). This mode allows up to 4 colours to be displayed on the screen at once. The four colours are arranged as two parts of background and foreground colours. When the Enterprise is first switched on, the colours are GREEN on BLACK and RED on BLACK. Translated into palette colours, these become:-

```
0,146,0,73
```

The cursor automatically takes on the colours of the second part in the palette unless instructed otherwise, (SET CURSOR COLOUR X) Try this example:-

1) Reset computer.
2) Type this:
   SET £102:PALETTE 17,27,7,255
3) Type this:
   SET CURSOR CHARACTER 154

The observed results are achieved by redirecting the palette-setting command to the text channel (£102). To switch between the colour pairs in immediate mode without altering the palette, there are two options. The first is to press the CTRL key and FUNCTION 7 simultaneously which will toggle between the two colour pairs for 1 line only. The second is to press the ALT key and FUNCTION 7 which will colour an entire paragraph. To use these two functions in a program, print the codes 246 and 254 respectively, e.g.:

```
PRINT CHR$(246);"HELLO"
```

A more versatile method of changing colours is on a character by character basis using the SET INK command. This must be re-directed to the text page in the same way as the palette command, e.g.:

```
SET £102:INK2
```

will cause all subsequent text to be printed in background colour 2, foreground colour 3.

```
SET £102:INK 3
```

will have exactly the same effect because each ink colour in a text mode is always associated with a defined paper colour.

The second text mode is 'TEXT 80'. This offers a screen format of 78 x 24 text in 8 colours (4 colour pairs). Because of the higher resolution and number of characters, this mode uses much more video RAM than 'TEXT 40' but, instead of colours being toggled, they alternate in a sequence of 4 permutations (it's easier to understand if you try it for yourself). Palette commands can have up to eight arguments in this mode, e.g.:

```
SET £102:PALETTE 1,2,3,4,5,6,7,8
```

Any unspecified colours default to black.

Finally, try this demo-program:-

```
100  PROGRAM "COL_TEXT"
110  ! WRITTEN BY GERALD MORGAN
120  TEXT 80
130  SET £102:PALETTE 7,255,17,27,25,146,73
140  LET SHADE=0
150  DO
160  SHADE=SHADE+1
170  SET £102:INK MOD(SHADE,8)
180  PRINT "*";
190  LOOP
200  END
```

ENTERPRISE
COMPUTERS LTD

31-37 Hoxton Street
London N1 6NJ
Telephone 01-739 4282
Telex 22717 ENTER G

APPLICATION NOTE NO. 26

READING AND WRITING CHARACTERS IN MACHINE CODE

The Enterprise's operating system is called EXOS. EXOS is capable of executing a number of 'functions'. There are over 30 of these and Function 11 is capable of executing many sub-functions, therefore a short program instruction can execute a large number of routines. Just 4 of these functions will be described here as they are all connected. To access these functions from machine code the RST 30h instruction is used. Take, for example, Function 5 (Read character). First of all the number of the channel that is going to be read from must be loaded into the accumulator (A). Next the RST 30h instruction is executed followed by the data item '5'. On a Z80 assembler the code looks like this:-

```
LD A,105 ! Loads keyboard channel number into Accumulator.
RST 30h  ! Calls the function specified by the next byte.
DEFB 5   ! The function number itself.
```

30h => 48

This program will cause a character to be read from the keyboard and the ASCII code of it to be placed in register B. The accumulator will then contain the status. The program can only be entered and executed using a Z80 assembler (not from BASIC). If the need arises for this machine code to be incorporated in a BASIC program, it should be stored in a CODE statement like this:-

```
ALLOCATE 6
CODE FRED=HEX$("3E,69,F7,05,C9")
CALL USR(FRED,0)
```

Here are the 4 read/write functions:-

Function 5 (Read CHARACTER).

```
PARAMETER : A = Channel number.
RESULTS   : A = Status.
            B = Character code.
```

Function 6 (Read BLOCK).

```
PARAMETERS :  A = Channel numbers.
              BC = Total bytes in block (0-65535).
              DE = Buffer address.
RESULTS    :  A = Status.
              BC = Bytes left to read.
              DE = Modified buffer address.
```

Function 7 (Write character).

```
PARAMETERS : A = Channel number.
             B = Status code.
RESULTS    : A = Status.
```

Function 8 (Write block).

PARAMETERS : A = Channel number.
BC = Total bytes in block (0-65535).
DE = Buffer address.

Default channels for the various devices are as follows:-

COMMAND input and standard TEXT output - Channel 0
GRAPHICS input (LOOK, etc.) and output - Channel 101
Standard TEXT screen (initially 24x38) - Channel 102
SOUND out (open until explicitly closed) - Channel 103
PRINTER (default for the COPY command) - Channel 104
KEYBOARD operations (this is read only) - Channel 105
File based I/O tasks (default is TAPE:) - Channel 106
Local area NETWORK operations (NET:) - Channel 107

Additional channels can be opened to devices for greater flexibility.

ENTERPRISE
COMPUTERS LTD

31-37 Hoxton Street
London N1 6NJ
Telephone 01-739 4282
Telex 22717 ENTER G

APPLICATION NOTE NO. 27

Using Video from Machine Code Program.

The 'Nick' chip and the EXOS video driver together provide the ENTERPRISE with it's superb graphics capabilities. This document describes the operation of EXOS and it's video driver. Any queries regarding this Application Note should be directed to Enterprise Computers - Technical Support Department, and NOT to Intelligent Software.

VIDEO DRIVER SPECIFICATION.

Introduction

The screen driver handles the display of any number of video "pages" in the various different display modes which the videochip provides although it does not support all the possible modes.

The display is managed in terms of video "pages", with one page corresponding to each channel. Before a channel is opened to the screen driver, the user must specify various parameters, such as a video mode and page size, by setting EXOS variables. A channel can then be opened to device "VIDEO:". If a filename or unit number is specified, then it will be ignored. The video driver will work out how much screen RAM it needs for this video page and obtain that much RAM from EXOS, including enough for the various variables needed.

Once the channel has been set up in this way, the user can read and write characters or blocks of data. This data will be interpreted differently by pages of different modes, particularly control characters and escape sequencies.

At this stage however, the video page will not be visible on the display. A special function call is required to cause a video page to be actually displayed on the screen. It is only at this time that the appropriate line-parameter blocks are set up and the text/graphics will appear. It is possible to display any vertical section of a video page at any vertical position on the screen, covering up anything which was displayed on those scan lines before. If the page width is less than the full screen width, then the margins will be adjusted to display the page in the middle of the screen.

The screen driver has a 128 character font in video RAM which it uses for all character type displays. This is initialised to a standard ASCII character set repeated twice, but any character may be re-defined by the user. Each character is 8 pixels wide and 9 lines deep. These values include the space between characters and between lines.

VIDEO DRIVER SPECIFICATION

Co-ordinate Systems

The co-ordinate system used in specifying graphic positions, etc., is standardised so that giving the same commands to two pages of different resolutions or colour modes will produce a pattern of the same size on the screen. A graphics page of full screen size will be 972 logical pixels high and 1344 pixels wide. This corresponds to twice the maximum horizontal and four times the vertical resolution available. All beam positions, etc., are specified in these co-ordinates, and depending on the colour mode, the actual position will have to be an approximation.

To activate special functions, use EXOS function 11:

Parameters:
- A = Channel number.
- B = Sub-function number.
- C = Unspecified parameter.
- D = Unspecified parameter.

Results:
- A = Status.
- C = Unspecified parameter.
- DE = Unspecified parameter.

MAX X = 672 (42)
" Y = 243 (27)

Basic Control of Video Pages

Each video page is a separate channel. When a channel is opened to the video driver this implies that another video page is to be created. The video driver looks at EXOS variables which specify the page size, page mode and colour mode. These variables must be set up by the user before opening a video channel. From RAM it needs and obtains that much with an EXOS function call ("Allocate channel buffer").

The video driver maintains the line parameter table in a fixed place in its absolute device RAM area. The line parameter table always consists of 28 line parameter blocks of 9 scan lines each for the display area and various other ones to generate the frame sync. and borders. The first line parameter block is reserved for the status line display which is a fixed area of RAM. The other 27 line parameter blocks can display any part of any page, so display is always in vertical units of 9 pixels. All 28 line parameter blocks are initially set up to be blank (i.e. all border colour). The variable LP_POINTER in the EXOS variable area points to the start of the line parameter table.

Display Returns

The display mode is specified by an EXOS variable MODE-VID the allowed values of which are:

```
 0  -  Hardware text mode (up to 42 chars./line).
 1  -  High resolution pixel graphics.
 2  -  Software text mode (up to 84 chars.line).
 5  -  Low resolution pixel graphics.
15  -  Attribute graphics.
```

The three graphics modes correspond to the PIXEL, LPIXEL and ATTRIBUTE modes of the Nick chip.

Colour Modes

As well as the display mode, each video page is of a particular colour mode. The colour mode is specified by an EXOS variable called COLR-VID. The allowed values for this variable are:

```
0 - Two colour mode.
1 - Four colour mode.
2 - Sixteen colour mode.
3 - 256 colour mode.
```

For text modes it is only useful to use two colour mode, unless the characters in the font are re-defined for doing some sort of block graphics. Also attribute mode must always be in two colour mode, although sixteen colours will actually be available.

Page Size

Two EXOS variables, X_SIZ_VID and Y_SIZ_VID, define the size of the page to be created. The vertical size is specified in character rows. It can be any value from 1 to 255, although only 27 rows can be displayed on the screen at one time. The horizontal size is specified in low resolution character widths, and can be any number from 1 to 42.

A special function call is provided to return the size of a video page. It returns the number of lines and the number of characters per line. The characters per line value returned is the actual number of characters per line so, in the case of a software-text mode, it will be double the value in X_SIZ_VID when the channel was opened.

The parameters for this are:

Parameters: A = Channel number (1...255)
B = 2 (Special function code)

Returns: A = Status
B = Number of characters per row.
C = Number of rows.
D = Mode of page (0, 1 or 2, 5 or 15)

Display control

Video pages are not actually displayed on the screen until the user explicitly requests this. This request is done by a special function call. The parameters for this call are:

Parameters: A = Channel number (1...255)
B = 1 (Special function code)
C = First row in video page to display (1...size)
D = Number of rows to display (1...27)
E = Row on screen where first row should display (1...27).

Returns: A = Status

The three row parameters are all given in character row units since the area of screen specified must be a whole number of line parameter blocks. The displayed page will replace anything which was displayed on that part of the screen before. If the channel is subsequently closed then any part of the screen which was displaying that channel will be made border colour (by bringing the margins in the relevant line parameter blocks right in).

A value of 1 for the position on screen parameter (given in register E) refers to the line on the screen directly below the status line. Thus it is not possible to overlay the status line since zero will not be accepted.

If a value of zero is given for the position in the page parameter (register C) then the portion of the screen defined by the other two parameters will be blanked (i.e. made entirely border colour).

VIDEO DRIVER SPECIFICATION

Character Output

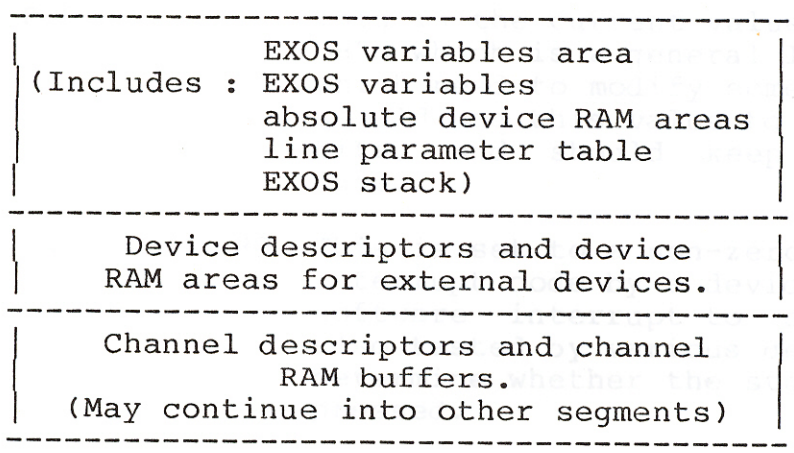
The screen driver supports both the single character write and the block write EXOS function calls. A block write is exactly equivalent to writing all the characters individually, except that it is rather faster as it avoids the overhead of going through EXOS for every character. Block write is implemented using the general purpose WRBLOCK utility routine.

Printing Characters

All characters above 1Fh will be treated as printing characters and will be put at the appropriate place on the video page. All modes have some sort of "cursor" which moves when a character is printed, but the details vary between different modes.

The bit maps for characters are stored in a fixed character font which is initialised to an ASCII character set. Each character is eight bits wide and nine bytes deep. The user can re-define any of these characters by an Esc. sequence (see Application Note No. 23).

KERNEL SPECIFICATION



The very top of the EXOS variable area contains a few defined values which are guaranteed not to move in future versions of EXOS. The are listed here with the address where a device will see them (in Z-80 page-2). If they are to be accessed by an applications program, then the correct segment must, of course, be paged in.

OBFFFh - USR_P3 \
OBFFEh - USR_P2 \ These are the contents of the four paging
OBFFDh - USR_P1 / registers when EXOS was last called.
OBFFCh - USR_P0 /

OBFFA/Bh - STACK_LIMIT Devices which need more than the default 100 bytes of RAM can let their stack down as far as the contents of this variable.

OBFF8/9h - RST_ADDR The address of a warm reset routine which must be in the page-zero segment. If this is zero then a cold reset will be done.

OBFF6/7h - ST_POINTER The Z-80 address of the status line memory. The 42 bytes from this address onwards are the status line (see video driver spec ET11/9).

OBFF4/5h - LP_POINTER The Z-80 address of the start of the line parameter table. The first line parameter block will be the status line one.

OBFF3h	- PORTB5	This is the current value in the output port OB5h which is a general I/O port. Devices which need to modify some bits of this port should use this value to avoid changing other bits, and should keep this variable up to date.
OBFF2h	- FLAG_SOFT_IRQ	This is set to a non-zero software interrupt code by a device to cause a software interrupt to occur. It is also tested by various devices to determine whether the stop key has been pressed.

The size of the EXOS variable area depends on the amount of RAM and ROM in the system which is determined at startup time and cannot change.

The size of the device RAM area depends on what external and user devices are linked into the system. The size of this can only change when a user device which requests some system RAM is linked into the system.

The channel buffer area is very dynamic since it can change whenever a channel is opened or closed. The channel buffers may occupy any number of segments. The system will, of course, ensure that channel buffers for the video device are kept in the internal video RAM.

To write a character, use EXOS function 7:

Parameters: A = Channel number.
B = Character code.

Results: A = Status

To write a block, use EXOS function 8:

Parameters: A = Channel number.
BC = Byte count.
DE = Buffer address.

Results: A = Status
BC = Bytes left to write.
DE = Modified buffer address.

ENTERPRISE
COMPUTERS LTD

31-37 Hoxton Street
London N1 6NJ
Telephone 01-739 4282
Telex 22717 ENTER G

APPLICATION NOTE NO. 28

Memory map information.

The memory maps of most microprocessors are 'static'. For example, the video RAM might ALWAYS reside at locations A000h to BFE0h inclusive, therefore restricting the displays you can have and chopping that amount of memory off the maximum you can use for your program. As you may be aware, a processor such as the Z80 can usually only access 64K. This means that if a computer has 32K ROM and 64K RAM only 32K of RAM would be available for use. However, the Enterprise is different. By use of a special custom chip called DAVE a colossal 3900K can be accessed. This is achieved by using 'DYNAMIC memory management'. EXOS handles this in such a way as to make the best use of RAM, ROM and the Z80 CPU.

The DAVE chip works in conjunction with EXOS (Enterprise eXpandable Operating System) shifting memory about to the best advantage of all the devices (video, sound, etc.).

This, of course, means that a 'standard' memory map cannot be drawn for the Enterprise because it is constantly changing. To keep track of where everything is, there are 'defined address variables' which NEVER m ove about in memory. A list of these is included as part of this Application Note. On the last page is an example memory map which is of a 64K Enterprise with just 1 cartridge ROM (i.e. BASIC) at the point just after the sign-on flashing 'ENTERPRISE' message is displayed.

Even though the field sizes will differ according to circumstances within the machine, the order in which they appear will always be correct barring deliberate re-arranging by the programmer.

EXAMPLE MEMORY MAP
(64k machine with 1 ROM cartridge at switch-on)

Address	Contents	Size
BFFFh:		
...	Defined address variables (see list).	0017
BFEFh:		
	Internal EXOS system variables.	0267
BEE4h:		
	Device RAM areas (for built in devices).	3212
B258h:		
	Space for EXOS RAM resident code.	0060
B21Ch:		
	The system stack.	1604
ABD6h:		
	RAM segment list, 1 byte per segment.	0004
ABD2h:		
	Extension ROM list, 4 extra bytes per ROM.	0012
ABC6h:		
	RAM areas for extension ROMs.	0000
	Device descriptors for built in devices.	0132
AB42h:		
AB41h:	Start of channel descriptor chain.	

DEFINED ADDRESS VARIABLES

	Address	variable	use
A)	OBFFFh	USR_P3	These variables hold the contents of the four DAVE chip paging registers when EXOS was last called. Needed by devices when given user addresses.
B)	OBFFEh	USR_P2	
C)	OBFFDh	USR_P1	
D)	OBFFCh	USR_P0	
E)	OBFFA/Bh	STACK_LIMIT	Used for stack checking by devices which need to use more than the default amount of stack space.
F)	OBFF8/9h	RST_ADDR	User's warm reset address.
G)	OBFF6/7h	ST_POINTER	This holds the Z80 address of the status line memory. The 42 bytes from this address onwards make up the status line.
H)	OBFF4/5h	LP_POINTER	This holds the Z80 address of the start of the line parameter table. The line parameter table contains the bytes to be displayed on the screen by the NICK chip.
I)	OBFF3h	PORTB5	Holds the current value of general output port 0B5h. Used by various devices which access this port.
J)	OBFF2h	FLAG_SOFT_IRQ	Triggers software interrupts.
K)	OBFF0h	SECOND_COUNTER	16-bit 1Hz incremental counter. This is reduced modulo 65535.
L)	OBFEFh	CRDISP_FLAG	Flag for suppressing the flashing 'ENTERPRISE' sign-on message when the computer is powered up or reset. Can only be used by a ROM.
M)	OBFED/Eh	USR_ISR	Address of user's interrupt service routine. This must be in page-0 and can be set to zero for no routine.

A) 255, 16383
B) 255, 16382
C) 255, 16381
D) 255, 16380
E) 255, 16378/9
F) 255, 16376/7
G) 255, 16374/5
H) 255, 16372/3
I) 255, 16371
J) 255, 16370
K) 255, 16368
L) 255, 16367
M) 255, 16365/6

ENTERPRISE
COMPUTERS LTD

31-37 Hoxton Street
London N1 6NJ
Telephone 01-739 4282
Telex 22717 ENTER G

APPLICATION NOTE NO. 29

Serial Transfers between the IBM PC and the Enterprise.

Construction of the following cable will enable the connection of the serial port built into the Enterprise to the serial card for the IBM PC, thus enabling the transfer of data and programs between the two machines via a specified Enterprise channel opened to its serial port.

Note that due to the nature of the signals on the IBM serial card complete handshaking cannot be implemented. This should cause few real problems in practice as it only affects data transmissions from the Enterprise to the IBM and the IBM has a fairly large buffer. However it may be the cause of unexpected "Write Fault" errors from the serial card driver on the IBM at the beginning of a data transfer if the channels to the Enterprise serial port are not opened and closed at the right moment.

The Enterprise serial handler does no translation of data so it is possible to transmit and receive any bytes of data and thus machine code programs can be transferred. However due to software drivers within the IBM which control serial data reception all "Carriage Returns" sent to the IBM via a device are stripped. Note if two consecutive "Carriage Returns" are found this is taken as the end of the file being sent just as if the end-of-file character (Control Z) had been found. Note that a line feed (Control J) will not be stripped.

Transfer of Files.

To avoid write fault errors being reported on the IBM all commands should be followed in strict sequence.

Enterprise to IBM

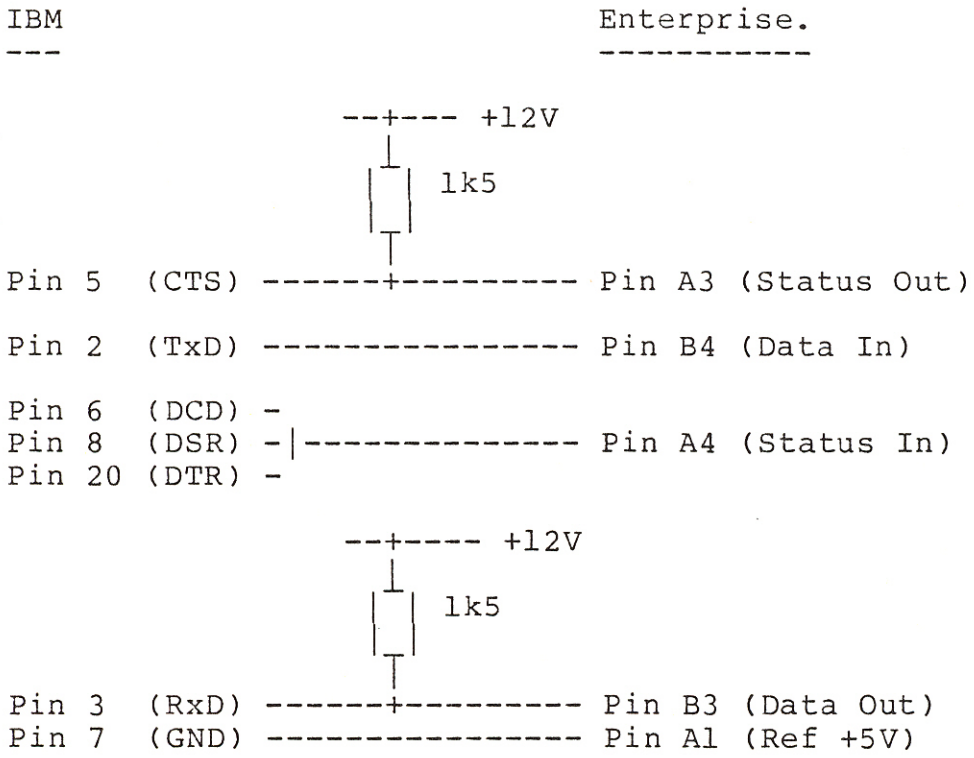
```
(IBM)          copy com1: filename
(Enterprise)   open £111:"serial:" access output
               copy from £10 to £111
```

IBM to Enterprise

```
(Enterprise)   open £111:"serial:"
               copy from £111 to £10
(IBM)          copy filename com1:
```

Notes.
All commands for IBM are DOS ones and for Enterprise are BASIC. All Enterprise channel numbers above 110 are not closed when the BASIC RUN command is given and so the channels used above should be closed if already open before any of the above commands are given.
In the above examples channel £10 is assumed to be already opened to the device required, for a disk the relevant command would have been 'open £10:"disk:filename" access output' and for the tape 'open £10:"tape:filename" access output'.
Since the Enterprise BASIC COPY command does not translate any of the data, if it comes from a serial port then the STOP key should be pressed to terminate the copy from the IBM at the end of the file once the IBM has finished sending and all activity has ceased on the Enterprise.
Before any transfers are attempted the serial baud rate on both machines should be set to the same rate, which can be done for the Enterprise from BASIC eg. for 9600 baud 'SET SERIAL BAUD 15'.

Cable Schematic.



Notes. Two 1k5 pullup resistors are required in order that the Enterprise be able to drive the IBM serial card properly. These should be connected to +12V, with reference to the Enterprise Ground ie. pin B1 of the serial port. It is recomended that the 12V come from an external regulated power supply, although experiments have found the resistors can be connected to pin 20 (DTR) of the IBM serial port for this source instead in which case no connection to pin B1 is required, but this is not

guaranteed to work.

To ensure that the voltage swings obtained by using the ref +5V are not shorted out it is important to make certain that there is no alternative common ground connection between the two machines, such a situation may well arise via the grounded shield on the monitor leads. These problems may be overcome if one machine (and its associated peripherals) is not connected to ground at any point.