

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 escape sequences cater for.

An escape 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 range 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
- `escC` - Set one palette colour
- `escI<n>` - Set ink colour to <n>
- `escP<n>` - Set paper colour to <n>
- `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 (m-20h) to (n-20h) m <= n