

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.

The information in this Application Note is provided to aid users, software houses and hardware companies in the creation of software or hardware for use with the Enterprise computer. The firmware of the computer, and the documentation herein, are Copyright (1984) Intelligent Software Limited, and no unauthorised reproduction, in any form, is permitted.

The publication of detailed technical information does not imply the release of such information into the public domain, and no waiver is granted to use such information for gain except in connection with the creation of software or hardware for use with the Enterprise computer.

Directors
D.N.L. Levy, R.H. Madge
L.I. Mahtani, D.M. Mirpuri
M.L. Mirpuri, K.J. O'Connell

Registered office
9 Cavendish Square
London W1M 9DD
Registered in England
No. 1674248

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 sequences.

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.

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.

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

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

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.

VIDEO DRIVER SPECIFICATION

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

| EXOS variables area
| (Includes : EXOS variables
| absolute device RAM areas
| line parameter table
EXOS stack)

| Device descriptors and device
RAM areas for external devices.

| Channel descriptors and channel
| RAM buffers.
(May continue into other segments)

The very top of the EXOS variable area contains a few defined values which are guaranteed not to move in future versions of EXOS. They 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 | \ / | These are the contents of the four paging registers when EXOS was last called. |
| OBFFEh | - | USR_P2 | | |
| OBFFDh | - | USR_P1 | | |
| 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. |

KERNEL SPECIFICATION

- 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.