

SIDEWAYS PRINTING

Elizabeth Wald's ingenious program sidesteps the Epson HX-20's internal microprinter.

Slipping sideways

The internal microprinter on the Epson HX-20 is a useful device for listing programs and printing results. But it is limited to a maximum of 24 characters per line, which means that program listings are difficult to read and wide program print-out is impossible.

The sideways printing program given in Figure 1 creates the 'sideways printer', which has the device name 'SPT0:' when used in Basic. This uses the microprinter to print out blocks of 16 lines of 80 characters sideways down the paper. A 'cutting line' is printed either side of the text, so that blocks of text may be cut up and pasted together.

The sideways printer is used in Basic by extending the interpreter to include subroutines which control it. Other items, for example disks or a bar-code reader, could also be added in the same way. The collection of subroutines required to control a particular device is known as a device driver.

This actually contains a table known as the device control block. This specifies the addresses of the control subroutines, the device name, and other essential information. The driver is linked to the interpreter via a second table containing the addresses of all the device control blocks. This second table has space for 16 entries of which seven are already defined.

To add any device driver to Basic it is loaded into the lowest end of memory, below MEMSET, and a small loader routine is executed. This loader routine moves all Basic programs and application files down, and copies the device driver into the resulting space at the top end of memory.

Application files are machine code programs permanently linked to the system and are normally linked to the menu. The loader routine then updates various system pointers to protect the driver from being overwritten by Basic, and finally executes a Basic warm start into program area 1. The computer is now in the same state as if Basic had been entered directly from the menu.

When Basic is entered via a warm start it executes an initialisation routine for each linked device in turn. This is contained within the driver, and ensures that the device is marked as being closed, and is also responsible for producing a new 'LOGIN' message. If the sideways printer cannot be initialised, for example if the user attempts to link in more than 16 devices, the normal 'LOGIN' message will be displayed.

Driver subroutines

The sideways printer driver contains subroutines to open and close 'SPT0:' and to transfer a single character to the device from Basic.

Figure 1: Disassembled Machine Code listing

<pre> 0A40 8E LDS #04AF 0A43 7B TIM #4078 0A46 27 BEQ 0A50 0A48 FE LDX 0138 0A48 AD JSR X00 0A4D 71 AIM #BF78 0A50 8D BSR 0A52 0A52 32 PUL A 0A53 33 PUL B 0A54 C3 ADD D #00CE 0A57 DD STA D #6A 0A59 CC LDA D #07A0 0A5C DD STA D #68 0A5E D3 ADD D #6A 0A60 DD STA D #6C 0A62 FC LDA D #12C 0A65 DD STA D #62 0A67 FC LDA D #134 0A6A DD STA D #68 0A6C 93 SUB D #68 0A6E DD STA D #64 0A70 18 XDX 0A71 9C CPX #6C 0A73 24 BCC 0A7D 0A75 CC LDA D #0605 0A78 BD JSR FF64 0A7B 28 BRA 0AF9 0A7D 0F SEI 0A7E FF STX 0134 0A81 FC LDA D #136 0A84 93 SUB D #68 0A86 FD STA D #136 0A89 8D BSR 0B00 0A8B DC LDA D #6A 0A8D DD STA D #60 0A8F DC LDA D #6C 0A91 DD STA D #64 0A93 DC LDA D #64 0A95 FD STA D #12C 0A98 8D BSR 0B00 0A9A FE LDX 0134 0A9D 08 INX 0A9E 08 INX 0A9F 08 INX 0AA0 A6 LDA A X00 0AA2 81 CMP A #39 0AA4 27 BEQ 0AAD 0AA6 EE LDX X01 0AA8 09 DEX 0AA9 09 DEX 0AAA 09 DEX 0AAD 28 BRA 0AA0 0AAE 47 STA A #7E 0AB1 FC LDA D #12C 0AB4 C3 ADD D #0003 0AB7 ED STA D X01 0AB9 CE LDX #013C 0ABC A6 LDA A X01 0ABE 81 CMP A #45 0AC0 27 BEQ 0AF1 0AC2 DF STX #6E 0AC4 A6 LDA A X01 0AC6 EE LDX X02 0AC8 8C CPX #FFFF 0ACB 27 BEQ 0AF1 0ACD 4D TST A 0ACE 2B BMI 0AD4 0AD0 18 XDX 0AD1 93 SUB D #68 0AD3 18 XDX 0AD4 18 XDX 0AD5 0E LDX #6E 0AD7 ED STA D X02 0AD9 6D TST X01 0ADB 2A BPL 0ADF </pre>	<pre> 0ADD D3 ADD D #6E 0ADF 18 XDX 0AE0 A6 LDA A X01 0AE2 81 CMP A #45 0AE4 27 BEQ 0AF1 0AE6 4D TST A 0AE7 2B BMI 0AEF 0AE9 EC LDA D X04 0AEB 93 SUB D #68 0AED ED STA D X04 0AEF 28 BRA 0AC2 0AF1 FE LDX 0134 0AF4 FC LDA D #2CE 0AF7 ED STA D X0A 0AF9 0E CLI 0AFA 4F CLR A 0AFB FE LDX #004 0AFE 6E JMP X00 0B00 DE LDX #60 0B02 9C CPX #62 0B04 27 BEQ 0B14 0B06 A6 LDA A X00 0B08 08 INX 0B09 0F STX #60 0B0B DE LDX #64 0B0D A7 STA A X00 0B0F 08 INX 0B10 0F STX #64 0B12 28 BRA 0B00 0B14 39 RTS 0B15 01 NOP 0B16 01 NOP 0B17 01 NOP 0B18 01 NOP 0B19 01 NOP 0B1A 01 NOP 0B1B 01 NOP 0B1C 01 NOP 0B1D 01 NOP 0B1E 01 NOP 0B1F 01 NOP 0B20 39 RTS 0B21 00 *** 0B22 00 *** 0B23 3C PSH X 0B24 8D BSR 0B26 0B26 38 PUL X 0B27 3C PSH X 0B28 18 XDX 0B29 C3 ADD D #006A 0B2C 37 PSH B 0B2D 36 PSH A 0B2E 38 PUL X 0B2F C3 ADD D #0020 0B32 ED STA D X05 0B34 C3 ADD D #007C 0B37 ED STA D X07 0B39 C3 ADD D #FFA0 0B3C ED STA D X0B 0B3E 6F CLR X13 0B40 18 XDX 0B41 CE LDX #0665 0B44 6D TST X00 0B46 27 BEQ 0B53 0B48 08 INX 0B49 08 INX 0B4A 8C CPX #0677 0B4D 26 BNE 0B44 0B4F 38 PUL X 0B50 38 PUL X 0B51 28 BRA 0B5C 0B53 ED STA D X00 0B55 32 PUL A 0B56 33 PUL B 0B57 38 PUL X 0B58 C3 ADD D #0038 0B5B 18 XDX 0B5C 28 BRA 0B20 </pre>	<pre> 0B5E 08 45 78 74 Ext 0B62 65 6E 64 65 ende 0B66 64 20 45 70 d Ep 0B6A 73 6F 6E 20 son 0B6E 42 41 53 49 BAS1 0B72 43 00 0A 77 C w 0B76 69 74 68 20 ith 0B7A 53 50 54 30 SPT0 0B7E 3A 20 62 79 : by 0B84 20 45 20 57 E W 0B86 61 6C 64 00 ald 0B8A 0A 00 25 2C % 0B8E FF 8F 53 50 *SP 0B92 54 30 20 00 T0 0B96 00 00 00 8C 0B9A 70 00 00 8C P 0B9E 70 8C 70 00 P 0BA2 00 00 00 00 0BA6 50 0E 46 80 P F+ 0BAA 00 00 00 00 0BAE 00 00 0BB0 B6 LDA A 068A 0BB3 81 CMP A #20 0BB5 27 BEQ 0BB8 0BB7 C6 LDA B #33 0BB9 7E JMP 8433 0BBC 8D BSR 0C20 0BBE 6D TST X13 0BC0 28 BMI 0BC7 0BC2 62 01M X8013 0BC5 20 BRA 0C19 0BC7 39 RTS 0BC8 00 *** 0BC9 8D BSR 0BB0 0BCB 32 PUL A 0BCC 8D BSR 0C20 0BCE 36 PSH A 0BCF 6D TST X13 0BD1 27 BEQ 0BC9 0BD3 32 PUL A 0BD4 81 CMP A #0D 0BD6 27 BEQ 0C09 0BD8 81 CMP A #0A 0BDA 27 BEQ 0C09 0BDC 81 CMP A #20 0BDE 25 BCS 0C1D 0BE0 36 PSH A 0BE1 A6 LDA A X14 0BE3 C6 LDA B #50 0BE5 3D MUL 0BE6 36 PSH A 0BE7 A6 LDA A X15 0BE9 1B ABA 0BEA 16 TAB 0BEB 32 PUL A 0BEC 89 ADC A #00 0BEE 3C PSH X 0BEF 30 TSX 0BF0 E3 ADD D X00 0BF2 C3 ADD D #0100 0BF5 37 PSH B 0BF6 36 PSH A 0BF7 30 TSX 0BF8 A6 LDA A X04 0BFA EE LDX X00 0BFC A7 STA A X00 0BFE 38 PUL X 0BFF 38 PUL X 0C00 31 INS 0C01 6C INC X15 0C03 E6 LDA B X15 0C05 C1 CMP B #50 0C07 26 BNE 0C1D 0C09 6F CLR X15 0C0B 81 CMP A #0D 0C0D 27 BEQ 0C1D </pre>
--	---	---

SIDEWAYS PRINTING

◀ 25

The sideways printer requires a 16x80 byte buffer to contain the ASCII codes for one block of text. The buffer is filled when characters are sent to 'SPTØ:' from Basic.

The subroutine which opens 'SPTØ:' fills the 16x80 character buffer with spaces and the device is marked as being open. A horizontal 'cutting line' is then printed. 'SPTØ:' is closed by a subroutine which prints any characters remaining in the buffer and then marks the device as being closed.

The subroutine which transfers a character to 'SPTØ:' places the ASCII code for the character in the buffer. When the buffer contains 16 lines of data, the contents of the buffer are printed out followed by a cutting line. The buffer is then filled with spaces ready for the next block of text.

To print the text block, the sideways printer routine extracts one column of characters at a time from the buffer and converts this to the equivalent 16x6 byte block of dot patterns. This dot information represents six lines of dots on the printer, and these are printed out using a routine in the Epson operating system.

Machine code routines

A listing of the machine code is given in Figure 1. The program is totally relocatable, but for convenience is listed from location &H0A40 onwards. It is important to remember that when the driver loader routine is executed the machine code is copied to the top end of memory, and thus the addresses given will not be the correct ones. The device driver loader functions as follows:

0A40-0A7B Checks that space exists to link in extended Basic. If space exists then the program branches to 0A7D, otherwise it generates a beep and branches to 0AF9 to return to Basic.

0A7D-0A98 Copies all Basic programs and application files down and copies extended Basic into the space created at the top of memory.

0A9A-0AB7 Initialises the 'JMP' instruction to link the extended Basic into the interpreter (warm start hook).

0AB9-0AEF Updates the menu entries for the application files.

0AF1-0AF7 Resets MEMSET to the value before the Basic loader programs was run.

0AF9-0AFE Jumps to Basic warm start routine.

0B00-0B14 Block move subroutine.

The device driver functions as follows:

0B20-0B22 Warm start hook used to link in further extended Basics.

0B23-0B5C Device driver initialisation.

0BB0-0BC7 Opens the device.

0BC9-0C1D Transfers one byte from Basic to the device.

0C20-0C28 Calculates the address of the device control block.

0C2C-0C3C Closes the device.

0C40-0C50 Fills the character buffer with spaces.

Figure 1 (continued)

<pre> 0C0F 6C INC X14 0C11 E6 LDA B X14 0C13 C1 CMP B #10 0C15 25 BCS 0C1D 0C17 8D BSR 0C7C 0C19 8D BSR 0C5B 0C1B 8D BSR 0C40 0C1D 39 RTS 0C1E 00 *** 0C1F 00 *** 0C20 8D BSR 0C22 0C22 38 PUL X 0C23 18 XDX 0C24 83 SUB D #0092 0C27 18 XDX 0C29 39 RTS 0C29 00 *** 0C2A 00 *** 0C2B 00 *** 0C2C 8D BSR 0C20 0C2E 6D TST X13 0C30 27 BEQ 0C3C 0C32 6F CLR X13 0C34 6D TST X14 0C36 26 BNE 0C17 0C38 6D TST X15 0C3A 26 BNE 0C17 0C3C 39 RTS 0C3D 00 *** 0C3E 00 *** 0C3F 00 *** 0C40 8D BSR 0C20 0C42 6F CLR X14 0C44 6F CLR X15 0C46 18 XDX 0C47 C3 ADD D #01D0 0C4A 18 XDX 0C4B CC LDA D #0500 0C4E 36 PSH A 0C4F 86 LDA A #20 0C51 A7 STA A X00 0C53 08 INX 0C54 32 PUL A 0C55 83 SUB D #0001 0C58 26 BNE 0C4E 0C5A 39 RTS 0C5B 3C PSH X 0C5C 37 PSH B 0C5D 36 PSH A 0C5E 86 LDA A #85 0C60 8D BSR 0C6D 0C62 8D JSR FF94 0C65 86 LDA A #20 0C67 8D BSR 0C6D 0C69 32 PUL A 0C6A 33 PUL B 0C6B 38 PUL X 0C6C 39 RTS 0C6D C6 LDA B #18 0C6F FE LDX FFD2 0C72 08 INX 0C73 3C PSH X 0C74 A7 STA A X00 0C76 08 INX 0C77 5A DEC B 0C78 26 BNE 0C74 0C7A 38 PUL X </pre>	<pre> 0C7B 39 RTS 0C7C 8D BSR 0C20 0C7E 18 XDX 0C7F C3 ADD D #01D0 0C82 DD STA D #6E 0C84 C3 ADD D #0500 0C87 DD STA D #6C 0C89 5F CLR B 0C8A 37 PSH B 0C8B 5F CLR B 0C8C 32 PUL A 0C8D 36 PSH A 0C8E 37 PSH B 0C8F 8D BSR 0CF8 0C91 DE LDX #6C 0C93 3A ABX 0C94 B6 LDA A 0190 0C97 A7 STA A X00 0C99 B6 LDA A 0191 0C9C A7 STA A X10 0C9E B6 LDA A 0192 0CA1 A7 STA A X20 0CA3 B6 LDA A 0193 0CA6 A7 STA A X30 0CA8 B6 LDA A 0194 0CAB A7 STA A X40 0CAD B6 LDA A 0195 0CB0 A7 STA A X50 0CB2 33 PUL B 0CB3 5C INC B 0CB4 C1 CMP B #10 0CB6 26 BNE 0C8C 0CB8 5F CLR B 0CB9 37 PSH B 0CBA 86 LDA A #10 0CBC 3D MUL 0CBD D3 ADD D #6C 0CBF C3 ADD D #000E 0CC2 FE LDX FFD2 0CC5 08 INX 0CC6 3C PSH X 0CC7 3C PSH X 0CC8 18 XDX 0CC9 8D BSR 0D18 0CCB 8D BSR 0D18 0CCD 8D BSR 0D18 0CCF 8D BSR 0D18 0CD1 8D BSR 0D18 0CD3 8D BSR 0D18 0CD5 8D BSR 0D18 0CD7 8D BSR 0D18 0CD9 38 PUL X 0CDA 38 PUL X 0CDB 8D JSR FF91 0CDE 33 PUL B 0CDF 25 BCS 0CED 0CE1 5C INC B 0CE2 C1 CMP B #06 0CE4 26 BNE 0CB9 0CE6 33 PUL B 0CE7 5C INC B 0CE8 C1 CMP B #50 0CEA 26 BNE 0C8A 0CEC 39 RTS 0CED C6 LDA B #35 </pre>	<pre> 0CEF 31 INS 0CF0 7E JMP 8433 0CF3 00 *** 0CF4 00 *** 0CF5 00 *** 0CF6 00 *** 0CF7 00 *** 0CF8 37 PSH B 0CF9 36 PSH A 0CFA DE LDX #6E 0CFD 30 TSK 0CFE A6 LDA A X03 0D00 C6 LDA B #50 0D02 3D MUL 0D03 EB ADD B X02 0D05 89 ADD A #00 0D07 E3 ADD D X00 0D09 18 XDX 0D0A A6 LDA A X00 0D0C 38 PUL X 0D0E CE LDX #0190 0D10 8D JSR FF67 0D13 32 PUL A 0D14 33 PUL B 0D15 39 RTS 0D16 00 *** 0D17 00 *** 0D18 3C PSH X 0D19 A6 LDA A X00 0D1B 36 PSH A 0D1C E6 LDA B X01 0D1E 37 PSH B 0D1F 30 TSK 0D20 54 LSR B 0D21 54 LSR B 0D22 54 LSR B 0D23 EE LDX X06 0D25 E7 STA B X00 0D27 33 PUL B 0D28 58 ASL B 0D29 49 ROL A 0D2A 59 ROL B 0D2B 49 ROL A 0D2C 59 ROL B 0D2D E7 STA B X01 0D2F 32 PUL A 0D30 47 STA A X02 0D32 8D BSR 0D40 0D34 8D BSR 0D40 0D36 8D BSR 0D40 0D38 18 XDX 0D39 30 TSK 0D3A ED STA D X04 0D3C 38 PUL X 0D3D 09 DEX 0D3E 09 DEX 0D3F 39 RTS 0D40 E6 LDA B X00 0D42 86 LDA A #20 0D44 97 STA A #68 0D46 4F CLR A 0D47 54 LSR B 0D48 24 BCC 0D4C 0D4A 9A ORA A #68 0D4C 74 LSR 0068 0D4F 24 BCC 0D47 0D51 A7 STA A X00 0D53 08 INX 0D54 39 RTS </pre>
--	--	--



