LASER PRINTER CONTROLLER

GENERAL-PURPOSE PROTOCOL

Hamish Dewar
Igor Hansen
September 1983

1. FUNCTION

The general-purpose protocol defines a format for the presentation of text to the laser printer controller. It is designed to give access to the special facilities of the printer through the medium of control sequences embedded in the character stream which constitutes the text to be printed. The protocol allows the full capability of the printing device to be exercised, including precise positioning effects and the creation of arbitrary dot patterns, while permitting ordinary text to be presented with a minimum of protocol overhead.

Because the control sequences are introduced by the ASCII Escape character, this protocol places no restrictions on the use of the full ASCII printing set and standard format effectors in the text itself. The ASCII printing set may be extended to a full 8 bit set, provided that the communication medium permits transfer of arbitrary byte values.

Among the effects which may be achieved by means of control sequences is the selection of one of a number of alternative protocols. These give access to emulation modes and to a text-formatting mode, on a per-document basis. The text-formatting mode provides the facilities of pagination, line-filling and justification, but it also provides access to all the special capabilities of the printer. It is entirely text based and is designed for user created files. By contrast, the general-purpose protocol, based on control sequences, is designed for files created by utilities such as spoolers and formatters.

2. PRINCIPLES OF OPERATION

The basic function of the printer controller is to generate images of pages to be printed in the form of a matrix of pixels in which a pixel is 'on' or 'set' if a dot is to appear in the corresponding position on the physical page and 'off' otherwise. For the Fujitsu printer, the physical page size is A4 and, allowing for mandatory hardware margins, this gives a maximum page area of 7.7" across and 11.3" down. The dot density of the printer is 240 dots to the inch, so that the matrix of pixels is 1848 across and 2712 down.

The print image is created by the sequential execution of actions which are invoked by the character stream sent to the controller. The conceptual model is that of writing or plotting: there is a current operating position which determines where the next information is to be written; the act of writing automatically alters the current position in a precisely defined way; and there are also explicit positioning actions. The effect of writing is cumulative, which means that over-printing effects can readily be achieved. The controller places no restrictions on the order in which actions have to be executed. The current position may be set to any point in the image at any time.

The controller permits a number of 'pages' to be set up on a single sheet of paper, through a facility to define individual page formats within the full print area. In the subsequent description, the term 'sheet' or 'physical page' will be used for the full print area and the term 'page' by itself for a page format within this print area.

In the absence of control sequences, plain text is printed in a fashion similar to the operation of a conventional printer. A default page format and font are used. Consecutive printing characters are placed directly in the page format in adjacent positions. Line breaks are caused by the appropriate ASCII format effectors (see following section). Over-long lines are truncated, not wrapped. Page breaks are caused either by reaching the bottom of the current page or by receipt of an ASCII Form Feed character.

More generally, the principle of operation of the controller is:

actions are taken on the basis of the incoming characters

the effect of each action depends on the current environment as well as the character involved

as well as a printing effect the action usually also modifies part of the environment for subsequent actions

control sequences usually do not have a printing effect but modify aspects of the environment.

The most important components of the environment are:-

the current page format

the currently selected font

the current space increment

the current feed increments

the current position on the page

The current position identifies a particular point within the page format. It is represented by two values, one for the vertical or Y axis and one for the horizontal or X axis. Associated with the current position is a state variable which records whether the position was established implicitly by character interpretation or explicitly by control sequence. The main function of this state variable is to determine whether automatic vertical movement is required at the start of a line of text. Because the page formats include the possibility of printing along the length of the paper as well as across the width, it is important to understand that the terms 'vertical' and 'horizontal' are used in a relative sense. The same applies to the terms 'top', 'bottom', 'left' and 'right'. The interpretation of the positioning commands does not depend on the absolute values of these co-ordinates or the direction of incrementing, so these are not defined.

Within any selected font, an individual printing character denotes a particular pattern of dots. The effect of the occurrence of the character in the text is that the associated pattern of dots is placed in the page image at the point identified by the current position and then the current position is moved to the right by an amount determined by the symbol -- the character width.

Considered at this level, a font is an ordered set of character patterns and associated widths. The ordering corresponds to the sequencing of characters in the character set. The printer pre-supposes either a standard or an extended ASCII coding in the limited sense that the interpretation of the control characters (decimal 0-31) and the space character (32) is fixed. The remaining character codes, from decimal 33 to a maximum of 255, have a graphical interpretation as determined by the font definition. Not all fonts provide a complete set of characters. If a printing character code (33-255) occurs which is not defined in the currently selected font, it is ignored. As well as a width, each character pattern has associated with it two positioning values which control the placing of the character relative to the current position. For all standard fonts, these values are chosen so that the current position is treated as the leftmost point on the Base-Line. That is, the body of the character extends up from the current position, and any descender extends below it.

A complete font has associated with it an ascender height and a descender height, which establish the default vertical movements when printing text in that font -- the feed increments. The feed increments may also be set by an explicit control sequence. A font also has a space width, which establishes the horizontal increment to be caused by the ASCII space character (decimal 32) for that font. The space increment may also be set by an explicit control sequence.

The way in which lines of plain text are laid out on the page may now be described more precisely. When a character is received at the start of a page or line, without any explicit vertical movement having been specified, the availability of sufficient space between the current vertical position and the bottom of the page is checked, and if there is not, a Form Feed function is executed. In all cases, the current position is then moved down by an amount equal to the ascender feed increment minus one. A Line Feed function completes the two-stage vertical movement by moving down by an amount equal to the descender height plus one.

3. ASCII CONTROL CHARACTERS

The following control characters have a significance within the general-purpose protocol.

CR   RETURN   (decimal 13)

> CR causes the current position to revert to the leftmost point within the current page format, without change of vertical position (this code is treated as equivalent to the character pair CR LF if the Auto-LF option is set)

LF   Line Feed   (decimal 10)

> This code first causes the vertical position to be moved down by the ascender feed increment minus one, in just those circumstances in which the appearance of a printing character would cause this effect. Then, in all cases, the current position is moved down by the amount of the current descender feed increment plus one. In the absence of Auto-CR, there is no change of horizontal position.
> (this code is treated as equivalent to the character pair CR LF if the Auto-CR option is set)

BS   Back Space (decimal 8)

> BS causes the horizontal position to be adjusted to the left by an amount equal to the current space increment. If the movement would imply a position outside the current page limits, the position is set to the leftmost point within the page.

FF   Form Feed   (decimal 12)

> FF causes termination of the page defined by the current page format and a new page to be started. Depending on the page formats involved, terminating a page may or may not involve printing a sheet of paper. As a safe-guard, the controller discards multiple form feeds which would cause more than one blank sheet to be produced. The current position on the new page is set to the top left printing position.

ESC   Escape   (decimal 27)

> ESC is used to introduce control sequences (see next section).

NUL   Null   (decimal 0)

> The Null character is ignored.

(additional control characters may be defined in due course)
(the presence of an undefined control character in the character stream is a data error)

4. FORMAT OF CONTROL SEQUENCES

All control sequences start with a fixed control sequence initiator, hereafter designated LEADIN. For full compatibility with ANSI control sequence standards, LEADIN should consist of the character pair Escape followed by left square bracket. However, the controller permits omission of the second character, so that Escape alone may be used.

Depending on the individual control sequence, LEADIN is followed by a variable number of numeric parameters, and the control sequence is terminated by a function-code, which is always a letter. In other words, LEADIN and the function-code act as a sandwich for the parameter information. The semi-colon character is used as a separator between multiple parameters.

Numeric parameters are presented in free format decimal. They may, where non-integral values are appropriate, contain a decimal point. When a value is used to specify a measurement on the page, it is by default interpreted as a number of pixels. However, in these cases, the number may be followed by a double-quote symbol to indicate that the value should be interpreted as inches.

5. BASIC OPERATIONS

LEADIN height; width P   start page

The effect of the P control sequence is to define and start a new page
format. The current position when the sequence is executed is taken
as the top left position of the page, and the parameters define the
height and width of the page respectively. They may be specified in
inches rather than pixels, using the double-quote convention (eg
7.25"). Subsequent printing takes place within the page thus defined
until an 'end page' sequence or a Form Feed is encountered. Page
formats may be nested one within the other, to a maximum depth of
eight.

It is legitimate within the protocol for a movement to cause the
current position to move out of the page limits on the right-hand or
bottom edge. Any printing operations specified when the current
position is off-page are ignored. However the current position is not
permitted to go off-page at the left-hand or top edge, and any
movement which would imply this is limited to prevent it.

LEADIN E   end page

The 'end page' control sequence causes the current page to be closed
and the page format active when that format was started to be
re-instated.

LEADIN offset A   move up
LEADIN offset B   move down
LEADIN offset C   move right
LEADIN offset D   move left

These control sequences cause the current position to be modified by
the amount specified by 'offset' in the direction denoted by the
control code 'A' to 'D'.

LEADIN degrees R   rotate

The current direction of printing is altered by rotation through the
number of degrees specified. Only multiples of 90 degrees are valid
for the present controller. The current position does not change,
though its significance is altered. The significance of 'top', 'bottom',
'left' and 'right' are similarly altered.

LEADIN font-number F   select font

The font denoted by the specified number becomes the current font.
If the current space increment has not been explicitly set since the
last 'start page' command, it is set to the space-width of the font.
Similarly, if the current feed increments have not been explicitly set
since the last 'start page' command, these are set to the heights of
the font. There is no change in the current position.

If there is no defined font for the number specified, all subsequent
printing characters up to the next valid font selection are ignored.

LEADIN space-width H   set space increment

The current space increment is set to the value specified.

LEADIN ascender-height; descender-height V   set feed increments

The current line feed increments are set to the values specifed.

6. FONT DEFINITION

The controller has available a number of built-in fonts and the protocol permits the definition of additional fonts. A distinction is made between basic fonts and derived fonts. The basic fonts contain direct specifications for the pattern of each character included in the font. Derived fonts specify the characters they contain by selection from the basic fonts. Basic fonts are identified by an alphabetic name, which may be up to 12 characters long. Derived fonts are identified by a number in the range zero to 63.

Typically derived fonts are defined for individual documents, and the derived font numbers may be freely assigned and re-assigned (with the exception of fonts 0 and 1).

Font selection is in terms of derived, not basic, fonts. Hence the numerical parameter to the 'select font' control code. Font zero is always pre-defined in terms of a specific basic font which is used as the default for normal (listing) operation in A4 format, and font 1 is defined as a reduced version of this basic font for use in the double A5 format.

The simplest form of definition of additional derived fonts is to identify them directly with a complete basic font. They may also be defined partly in terms of one basic font, and partly in terms of others. Derived font characters may also be defined as the result of applying an operation to a basic font character, for example scaling, emboldening, or slanting.

The representation of basic fonts takes one of two forms: either a bitmap representation of the dot pattern or a representation as a set of line vectors. The controller supports both forms of representation and corresponding definitions. In general, the line vector representation is more flexible: characters thus represented may be scaled by arbitrary factors, whereas bitmap representations may be scaled only by integral factors; the line vector form is also more economical in storage. On the other hand, the bitmap representation gives completely precise control over the exact final pattern for each character.

There is a single control code ('S') for font specification, the various cases being distinguished by the first parameter. This control sequence continues after the code letter 'S' with the font-name.

Basic Font Definition

(a) bitmap representation

LEADIN 0; ascender-height; descender-height; space-width S font-name

The parameter ascender-height defines the vertical envelope above the Base-line for all the characters in the font and the parameter descender-height defines tenvelope below the Base-line. Ascender-height must be strictly positive. Descender-height may be zero. These values are used by the controller when processing running text (in the absence of explicit commands) to determine line spacing and the position of the Base-Line for characters in that font.

The parameter space-width defines the default value for the width of a space when this font is being used.

The extra parameter font-name, which follows the control code 'S', is the name of the font and may consist of up to 12 alphanumeric characters. Leading spaces are ignored and the name is terminated by LF (or CR if Auto-LF is set).

The line containing the font specification control sequence is followed by a sequence of character definitions. Each of these takes the form of a character definition control sequence followed by hexadecimal values

representing the strips of dots making up the character.

LEADIN char-value; height; down-offset; width; left-offset K

The parameter char-value is the ASCII character value in decimal. It must be in the range 33 to 255. The remaining parameters are all in units of pixels: in this case the option of specifying the values in inches is not available, since the values must be precise. Height is the number of horizontal strips making up this character; down-offset is the number of strips out of the total which are to be displaced below the Base-Line (to form a descender); width is the number of pixels in each horizontal strip, which also defines the effective width of the character for purposes of composition; left-offset is the amount by which the whole matrix of strips is to be displaced to the left of the origin (for characters which may overlap preceding characters).

Following the control sequence comes a sequence of variable length hexadecimal values, height in number, each defining a horizontal strip of dots, width in number. The order of the strips is top to bottom. The number of hexadecimal digits is, at a maximum, the smallest not less than the width specified for the character. The digits correspond to groups of four dots starting at the left of the strip, most signficant bit leftmost, and bit set corresponding to a dot. Trailing zeros need not be included provided that the hexadecimal value is terminated by a space or format effector. If the maximum number of digits is present, there may be such a terminator but need not be; that is, the values can simply be run together.
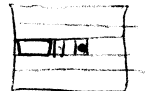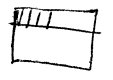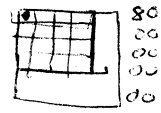
Here is a possible definition of an equals-sign with height 10 and width 19:

```
LEADIN 61;10;0;19;0K
3FFF8 3FFF8 00000 00000 00000 3FFF8 3FFF8 00000 00000 00000
```
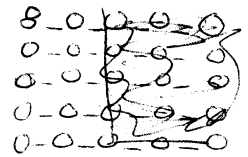
or, more compactly:

```
LEADIN 61;10;0;19;0K
3FFF83FFF80 0 0 3FFF83FFF80 0 0
```

The font definition is terminated by any control sequence other than 'define character' (control code 'K').

## (b) line-vector representation

LEADIN 1; ascender-height; descender-height; space-width; scale S

The first parameter indicates the line-vector type of font specification. The next three parameters have the same significance as for bitmap fonts. The parameter scale is an internal scaling factor which is to be applied to each character in the font in combination with any external scaling factor. A value greater than unity implies a scaling up and a value less than unity a scaling down. For each of the individual characters in the font, there is a control sequence followed by a series of co-ordinate pairs which trace out the shape of the character.

LEADIN char-value; height; down-offset; width; left-offset K

These parameters have the same significance as those for bitmap characters, bearing in mind the possibility of arbitrary scaling being applied. The co-ordinate system used in defining the shape of the character is internal to the character definition and is interpreted (when the character is drawn) relative to the position determined by current position offset by down-offset and left-offset. All co-ordinate values are non-negative and, in accordance with standard graphics practice, the 0,0 point corresponds to the bottom left position.
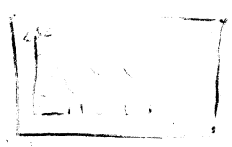
The co-ordinate pairs are interpreted sequentially as tracing out a path as a sequence of point-to-point lines or point-to-point moves (as with plotter 'pen-down' or 'pen-up' modes). The initial position is the 0,0 point. The horizontal co-ordinate is presented first, and the convention is adopted that a move as opposed to a line is indicated by preceding this value by a minus-sign. The last co-ordinate pair is indicated by the convention of preceding the final (vertical) value by a minus sign.

The font definition is terminated by any control sequence other than 'define character' (control code 'K').

Derived fonts

LEADIN font-number; ascender-height; descender-height; space-width T

The effect of this command is, first, to discard any existing definition of the derived font specifed by font-number and then to define the overall attributes for the font, on the same principles as for basic fonts.

Fonts #2....

The 'T' control sequence is followed by a number of derived character specifications.

LEADIN char-value; number; base-char-value; scale; op; amount I font-name

The effect is to define one or more character values in the derived font by reference to the same number of characters in a base font. The parameter char-value determines the first character in the derived font, and base-char-value the first character in the base font. Number indicates how many consecutive character values are involved. Scale denotes a scaling factor, which must be integral if the base font is a bitmap one. The parameter op stands for one of the following operations to be applied to the base character(s) to yield the derived character(s):

1 no operation

2 bold

3 slant by amount degrees

4 rotate by amount degrees

Font-name designates the base font, which must have been defined earlier. The whole command unit is terminated by a line break.

The font definition is terminated by a control sequence other than a character definition (control code 'I').

7. GRAPHICAL DATA

Pictorial information of any kind may be included in a document by including the appropriate direct specification of the data in the text, using a format which closely resembles that used for font definition. The difference is that the shape specified is drawn at the current position rather than being stored away for subsequent use.

(a) bitmap representation

LEADIN 0; height; down-offset; width; left-offset G

The control sequence is followed by hexadecimal values in the same form as for bitmap character specification. The pattern of dots is drawn immediately at the current position.

Mono Curved cursor down by offs-heigh

(b) line vector representation

LEADIN 1; height; down-offset; width; left-offset; scale G   *(NOT YET SUPPORTED*

The control sequence is followed by co-ordinate pairs in the same form as for line vector character specification. The shape described, scaled as indicated, is drawn immediately at the current position, on the same basis of alignment as for line vector characters.

Leadin 1   256   300   0   1   G
0 0 -256 256   0 0 -256 256
d(ot)

note

F 2

X X O X O O O O

sent hex pair

dot pattern
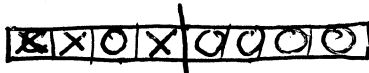
11

8. SERIAL INTERFACE

The serial interface permits the controller and associated printer to be connected as a passive printing device to an RS232 communication line. The baud rate of the interface may be set by switches to any of 1.2k, 2.4k, 4.8k, 9.6k or 19.2k baud. It is clearly desirable to utilise the highest baud rate available from the driving host in order to exploit the throughput capacity of the printer.

The electrical characteristics of the serial interface are those of the Motorola MC6850 ACIA.

The general-purpose protocol permits, but does not require the use of a full 8-bit character set. Switch options are available to cause the high-order bit to be ignored or to be used as a parity check bit.

The controller incorporates an input buffer of at least 8 kilobytes. XOFF/XON control is used on the outgoing channel to provide flow control, with XOFF being transmitted when less than 1 kilobyte of the buffer remains free.